Compact SO(3) Equivariant Atomistic Foundation Models via Structural Pruning

Chen Wang[1,2] **Siyu Hu**[1,*] **Guangming Tan**[1,2,*] **Weile Jia**[1,2,*]
[1]State Key Lab of Processors, Institute of Computing Technology,
Chinese Academy of Sciences, Beijing, China
[2]University of Chinese Academy of Sciences, Beijing, China
[*]Corresponding authors: {husiyu,tgm,jiaweile}@ict.ac.cn

Abstract

SO(3) equivariant graph neural networks have become the dominant paradigm for atomistic foundation models, achieving high accuracy and data efficiency by building rotational symmetry directly into the architecture. Yet the computational cost of their higher-order tensor operations creates a tough trade-off between model accuracy and inference efficiency. In this paper, we propose a structural pruning method for SO(3) equivariant atomistic foundation models to bridge this accuracy-efficiency gap. The pruning is applied along the channel and order dimensions, with each irreducible representation kept or removed as a complete block, thereby retaining SO(3) equivariance. Starting from a large checkpoint, the pruned model substantially reduces the inference cost while retaining higher accuracy than an independently trained small model. The pruned MACE-MP model outperforms the official from-scratch trained small model on 7 of 9 metrics on the Matbench Discovery leaderboard. In terms of efficiency, compressed MACE-MP and MACE-OFF models contain 1.5× to 4× fewer parameters and require 2.5× to 4× less pre-training compute than training a small model from scratch. For downstream applications, fine-tuning the pruned model reduces energy and force errors by 70.1% and 34.4% compared to training task-specific models from scratch across eight representative downstream datasets. We demonstrate that the method generalizes to other SO(3) equivariant architectures (SevenNet, eSCN) and can be combined with quantization and knowledge distillation for further gains.

1 Introduction

Atomistic foundation models have recently emerged as a transformative paradigm in computational chemistry and materials science [1]. These models [2–6] are available at different scales, placing each at a different point on the accuracy-efficiency frontier. Larger foundation models often achieve stronger benchmark accuracy than smaller ones, but their training cost, inference latency [7], and memory footprint become major bottlenecks for practical deployment. This persistent tension between the accuracy of large checkpoints and the efficiency of smaller ones remains a central challenge for atomistic foundation models.

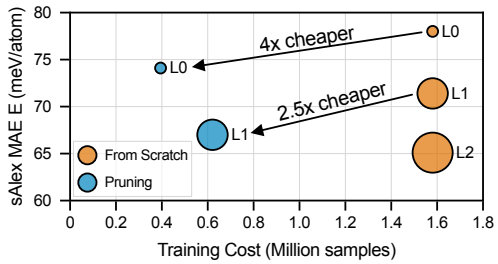

Figure 1: **Pruning results**. Pruned models outperform scratch-trained models of the same size at 2.5×–4× lower training cost. Point size indicates parameter count.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Among atomistic foundation models, SO(3) equivariant architectures have shown particularly strong accuracy on widely used benchmarks by explicitly encoding rotational symmetry [8–10]. This accuracy gain, however, comes with substantial computational overhead: equivariance is commonly implemented through tensor product operations over higher-order geometric features, whose cost scales up to $\mathcal{O}(L_{\max}^6)$, where $L_{\max}$ is the maximum order. This makes the accuracy-efficiency trade-off especially tough for SO(3) equivariant atomistic foundation models.

Model compression is therefore an important route to enhancing model efficiency, with three common paradigms: structural pruning, knowledge distillation (KD) [11], and quantization [12]. As summarized in Table 1, these methods target different aspects of the efficiency problem. Structural pruning removes redundant architectural units from a pre-trained checkpoint, producing a smaller model in the same family with reduced parameters and FLOPs. KD trains a smaller student model to mimic a larger teacher, which can reduce inference cost but typically depends on the chosen student architecture and does not directly reuse the teacher weights. Quantization keeps the architecture unchanged and reduces numerical precision, improving memory and hardware efficiency without reducing the number of parameters or tensor-product paths.

Table 1: **Comparison of model compression paradigms for atomistic foundation models.** Our structural pruning is distinct from and can be combined with Knowledge Distillation and Quantization.

Criterion	**Structural Pruning (Ours)**	**Knowledge Distillation**	**Quantization**
Weight source	Sliced from pre-trained	Random init (student)	Pre-trained (bit-reduced)
Architecture scope	Same family	Often cross-arch	Same arch
Reduces FLOPs & params	✓	✓(smaller student)	×
Reduces bit-precision	×	×	✓
SO(3) equivariance	Exactly preserved	Depends on student	Approx. preserved

Among these compression strategies, we focus on structural pruning because it can directly inherit weights from a large checkpoint and produce a smaller pre-trained model, offering a natural way to bridge this accuracy-efficiency gap. However, pruning strategies developed for large language models [13–16] cannot be directly transferred to SO(3) equivariant models: unlike scalar-featured networks, these models operate on higher-order geometric tensors, so naively removing feature dimensions breaks equivariance. To the best of our knowledge, existing pruning methods for atomistic foundation models remain coarse-grained, typically removing interaction blocks whose marginal accuracy contribution is limited [17, 18]. While such block-level pruning preserves SO(3) equivariance by removing entire equivariant modules, it effectively reduces the receptive field of the model and can be especially harmful for shallow model architectures [19]. This raises a central challenge: **how can we identify removable structures and compress the weights of a large pre-trained SO(3) equivariant model while preserving equivariance and retaining accuracy?**

To address this challenge, we propose a structural pruning framework for SO(3) equivariant atomistic foundation models. Our key contributions are as follows:

- We introduce a structural pruning method for higher-order SO(3) equivariant tensors across both channel and order dimensions while preserving SO(3) equivariance.
- We propose a four-stage framework to compress foundation models for downstream fine-tuning: (1) calibration, (2) pruning, (3) retraining, and (4) fine-tuning.
- We validate our approach on MACE-MP and MACE-OFF, achieving 1.5× to 4× parameter reduction and 2.5× to 4× training cost reduction (Figure 1). The pruned models outperform official models trained from scratch on accuracy and achieve 2.7× inference speedup. When fine-tuned on downstream tasks, they reduce energy and force errors by 70.1% and 34.4% compared to training from scratch.
- We demonstrate the generalizability of our method across architectures (SevenNet, eSCN) and show that structural pruning can be combined with quantization and knowledge distillation for further efficiency and accuracy gains.

Code is available at `https://github.com/HPC-AI-Team/SO3-Pruning`.

2 Related work

Atomistic foundation models. Atomistic foundation models are MLIPs trained on massive chemical datasets to predict interatomic potentials across the periodic table. Unlike earlier MLIPs [20–23] tailored for specific chemical systems, these models [2–6, 24–33] are pre-trained on vast, diverse quantum mechanical datasets [34–39] to learn transferable representations of interatomic interactions. By effectively bridging the accuracy of ab initio methods with the efficiency of classical force fields [40], these foundation models enable accurate simulations at unprecedented scales [41]. In practical applications such as drug discovery and materials design, foundation models can be fine-tuned on domain-specific datasets [42–44]. However, their massive parameter counts and computational costs create a deployment bottleneck, motivating our work to develop efficient compression strategies that make these foundation models practical for downstream applications.

Model compression. Model compression [45] encompasses a suite of techniques, including pruning [46], knowledge distillation (KD) [11, 47–49] and quantization [12, 50], designed to reduce the memory footprint and inference latency of deep networks. While distillation requires training student networks from scratch and quantization reduces bit-precision, pruning directly obtains the compressed model weights from the pre-trained model. In this paper, we focus specifically on pruning, as it offers a direct pathway to reduce computational overhead while maintaining the high numerical precision required for MLIPs.

Structural pruning. Structural pruning involves removing entire structural units rather than individual weights, ensuring acceleration without specialized sparse kernels. This approach has been widely studied in LLMs [51–53, 14]. Notably, existing work in MLIPs has largely focused on layer-level pruning strategies [17, 18], which reduce computational cost by decreasing model depth. In contrast, our work extends structural pruning to the domain of SO(3) equivariant GNNs, addressing the unique challenge of pruning coupled irreducible representations connected by tensor products, which standard channel-pruning methods fail to handle.

3 Method

MLIPs learn a parameterized energy $E = \Phi_\theta(\mathbf{R}, \mathbf{Z})$ over atomic positions $\mathbf{R}$ and species $\mathbf{Z}$, with forces derived as $\mathbf{F}_i = -\nabla_{\mathbf{r}_i}\Phi_\theta$. State-of-the-art MLIPs encode physical symmetries via SO(3) equivariant message passing [8, 9, 23], where node features are higher-order tensors $h^{(t)}_{j,klm}$ indexed by channel k and irreducible representation (l, m), coupled via Clebsch-Gordan products with $\mathcal{O}(L^6_{\text{max}})$ complexity. Structural pruning [54, 53] removes coherent parameter groups to yield dense, hardware-friendly compressed models. However, directly applying it to SO(3) equivariant architectures is non-trivial, as naive removal of feature dimensions breaks the symmetry of the tensor product path (see Appendix A and B for details).

Designing a structural pruning framework for SO(3) equivariant atomistic foundation models requires addressing two key issues: (1) identifying a pruning granularity that strictly preserves the geometric symmetry of the feature space, and (2) establishing a physics-informed importance criterion that captures the contribution of features to both potential energy and forces.

3.1 Equivariant pruning granularity

Standard structural pruning treats channels as independent units. However, in SO(3) equivariant models, the feature space is decomposed into a direct sum of irreducible representations of order $l \in [0, L_{\text{max}}]$. A feature vector at layer t, atom j, channel k, and order l consists of components $h^{(t)}_{j,klm}$ where $m \in [-l, l]$.

The rotation operation in SO(3) inherently couples all components within the m dimension. Treating these components independently would violate the transformation rules of the irreducible representations, thereby breaking the model's equivariance (see Proposition B.2 in Appendix for a formal proof). Therefore, any structural pruning must treat the subspace spanned by the $2l + 1$ components of a specific order l as an atomic unit to avoid breaking SO(3) equivariance. We define our pruning granularity based on the tuple (k, l), representing the k-th channel of order l.

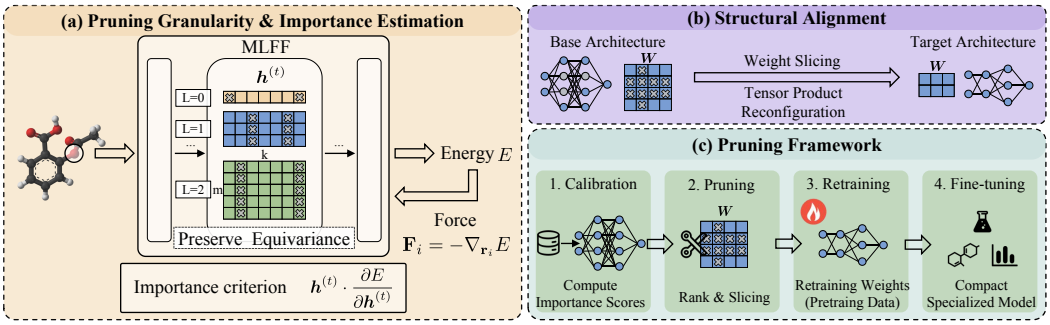

Figure 2: **Overview of the proposed structural pruning framework.** (a) **Pruning granularity & importance estimation:** To preserve SO(3) equivariance, features are pruned as (k, l) blocks based on an energy-force sensitive importance criterion. (b) **Structural alignment:** The model structure is aligned to the target structure via weight slicing and tensor product reconfiguration. (c) **Pruning framework:** The four-stage pipeline (Calibration, Pruning, Retraining, Fine-tuning) effectively compresses the foundation model to a compact specialized model.

We introduce a binary mask $\mathbf{z}^{(t)} \in \{0,1\}^{K\times(L_{\max}+1)}$, where $z_{k,l}^{(t)}$ indicates whether the feature block corresponding to channel k and order l is retained. The pruned feature is formally defined as:

$$\hat{h}_{j,klm}^{(t)} = z_{k,l}^{(t)} \cdot h_{j,klm}^{(t)}. \tag{1}$$

This formulation allows for heterogeneous pruning across orders. The model can retain scalar information ($l = 0$) for a specific channel while discarding its higher-order geometric tensors ($l > 0$), or vice versa. This flexibility is essential for compressing MLIPs, as higher-order interactions are computationally expensive but often sparse in their contribution to the prediction. As shown in Proposition B.4 (Appendix B.4), higher order l corresponds to higher angular frequencies. While these higher-order interactions capture fine geometric details, they are computationally expensive and often sparse in their contribution to the potential energy of stable systems.

3.2 Importance estimation

A robust pruning criterion for MLIPs must account for their dual objective: predicting the potential energy E and the forces $\mathbf{F}_i = -\nabla_{\mathbf{r}_i} E$. According to this, we propose an SO(3) invariant importance criterion based on the sensitivity of the potential energy surface. We approximate the impact of removing a feature block (k, l) on the total energy E using a first-order Taylor expansion. If a feature is pruned (i.e., set to zero), the resulting change in energy ΔE can be approximated as:

$$\Delta E \approx \sum_{j,m} \frac{\partial E}{\partial h_{j,klm}^{(t)}} \cdot (0 - h_{j,klm}^{(t)}) = -\sum_{j,m} \frac{\partial E}{\partial h_{j,klm}^{(t)}} \cdot h_{j,klm}^{(t)}. \tag{2}$$

Here, the term $\frac{\partial E}{\partial h_{j,klm}^{(t)}}$ represents the gradient of the energy with respect to the feature. This product effectively combines the feature's magnitude with the model's sensitivity to that feature.

This gradient-based approach is uniquely advantageous for MLIPs due to the intrinsic relationship between energy and forces. Since calculating forces requires backpropagating the gradient of E to the atomic positions $\mathbf{R}$, the intermediate gradients $\frac{\partial E}{\partial h}$ are computed during the standard force inference. Consequently, our criterion incurs negligible computational overhead. Furthermore, because forces are derived from the energy, features that significantly influence E (high gradient magnitude) are potentially critical for accurate force prediction.

We aggregate this importance score over the representation dimension m and all atoms j, averaged over a calibration dataset $\mathcal{D}$. The final importance score $I_{k,l}^{(t)}$ is defined as:

$$I_{k,l}^{(t)} = \mathbb{E}_{\mathbf{R},\mathbf{Z}\sim\mathcal{D}} \left| \sum_{j,m} \frac{\partial E}{\partial h_{j,klm}^{(t)}} \cdot h_{j,klm}^{(t)} \right|. \tag{3}$$

This criterion is SO(3) invariant (see Appendix B.3 for proof), ensuring that pruning decisions are consistent under rotations of the atomic configuration.

By retaining features with the highest $I_{k,l}^{(t)}$, we ensure the preservation of the structural pathways important for constructing both the potential energy surface and the associated force field. The effectiveness of this criterion over activation-only and magnitude-only alternatives is validated in Appendix D.4.

3.3 Structural alignment

Given a pre-defined target architecture (i.e., the target number of channels for each order l at each layer), we perform pruning by selecting the top-ranked feature blocks within each layer. However, simply masking features is insufficient for acceleration. The model structure must be physically altered to match the reduced feature dimensions. We enforce structural consistency through the following alignment steps:

1. **Tensor Product Reconfiguration:** The tensor product aggregates features via coupled paths. When the feature dimension of an input order l_2 is reduced, the corresponding radial weights $R_{kl_1l_2l_3}^{(t)}$ and Clebsch-Gordan paths must be indexed and sliced to align with the retained active channels. This ensures that the computational graph strictly corresponds to the dense, compacted tensors of the target architecture.
2. **Weight Slicing:** For the linear transformation within the equivariant blocks, the weights are parameterized as $W_{k\tilde{k}l}^{(t)}$, mapping the $\tilde{k}$-th input channel of order l to the k-th output channel. To align with the pruned architecture, we physically slice these weights. Specifically, we retain only the rows corresponding to the active output channels and the columns corresponding to the active input channels. The remaining entries form a smaller, dense weight tensor, effectively reducing the computational cost of the linear projection.

This process ensures that the pruned model is not a sparse version of the original but a dense model with a smaller architecture, ready for efficient inference on standard hardware.

3.4 Pruning framework

Our framework follows the four-stage pipeline illustrated in Figure 2:

1. **Calibration:** We pass a small subset of pre-training data through the pre-trained foundation model. We compute both the forward pass (Energy) and the backward pass (Forces) to accumulate the importance scores $I_{k,l}^{(t)}$.
2. **Target-Driven Pruning:** We define a target compact architecture, specifying the desired number of channels for each order l at each layer. For every layer, we rank the feature blocks by their importance scores $I_{k,l}^{(t)}$ and retain the top blocks to match the target specification. The model parameters are then physically sliced (as described in Sec. 3.3) to instantiate the compact model.
3. **Lightweight Retraining:** To recover from the approximation error introduced by pruning, we perform a one-time retraining phase on a subset of the pre-training data. This allows the remaining weights to adapt to the reduced architecture before task-specific fine-tuning.
4. **Fine-tuning:** Finally, we fine-tune the compact foundation model on a specific dataset to obtain a compact specialized model.

4 Experiments

We evaluate the proposed SO(3) equivariant structural pruning framework primarily on MACE [19], and additionally demonstrate its architectural generalizability on SevenNet [3] and eSCN [10]. Our experiments address four key questions: (1) Do pruned models achieve higher accuracy than same-size models trained from scratch? (2) Do they deliver practical efficiency gains in training cost and inference speed? (3) Do they serve as effective initializations for downstream fine-tuning? (4) Can

structural pruning be combined with other compression methods, such as quantization and knowledge distillation?

4.1 Experimental setup

Foundation models. Our experiments utilize the MACE [19] architecture. We examine two foundation model families: MACE-MP-0 [2] for inorganic materials (pre-trained on MPtrj [34], $\sim$1.5M structures, 89 elements) and MACE-OFF [55] for organic chemistry (pre-trained on SPICE [35], $\sim$1.0M conformations). Both families offer official checkpoints of varying sizes, providing baselines to benchmark our pruning method.

Downstream tasks. For inorganic systems, we evaluate on eight diverse datasets (SSE, H2O, AgAu, AlMgCu, Cu, Ti, V, W) covering materials under varied temperature and pressure conditions. For organic systems, we evaluate on 3BPA [56], AcAc [57], and rMD17 [58]. Dataset details are provided in Appendix D.1.

Implementation details. Further details regarding the calibration dataset construction, specific architectural handling during pruning, retraining hyperparameters, and ablation studies are provided in Appendix D. During the retraining phase, we maintain the same number of training epochs as the baseline models trained from scratch. Consequently, a reduction in training data size translates linearly to a reduction in total computational cost (GPU hours).

4.2 Accuracy

We evaluate whether pruned models produce higher-quality foundation models than those trained from scratch at the same parameter budget. We prune larger MACE-MP and MACE-OFF models into smaller target architectures using the official MACE settings, enabling a direct comparison with from-scratch baselines of identical size under the same evaluation protocol.

Table 2: **Accuracy of pruning methods on MACE-MP.** The table compares the performance of pruned models against baselines trained from scratch. Metrics include MAE for Energy (E, meV/atom), Force (F, meV/Å), and Stress (S, meV/Å^3) on MPtrj, and Energy (E, meV/atom) on the sAlex subset. Data indicates the fraction of training data used. Values in parentheses show the relative change in MAE compared to from-scratch trained models of the same size. Bold values indicate that the pruned model outperforms the from-scratch trained baseline.

Method	Base Model	Pruned Model	Params	Data	MPtrj			sAlex
					MAE E	MAE F	MAE S	MAE E
From Scratch	Layer2 L1 C256	/	15,847,440	100%	/	/	/	81.7
	Layer2 L2 C128		5,725,072		22.6	42.2	1.6	65.1
	Layer2 L1 C128		4,688,656		25.8	46.7	1.7	71.4
	Layer2 L0 C128		3,847,696		29.6	52.3	1.8	78.0
L Pruning	Layer2 L2 C128	Layer2 L0 C128	3,847,696	**25%**	30.9 (+4.4%)	**50.8 (-2.9%)**	1.9 (+5.5%)	**74.1 (-5.0%)**
				100%	**25.5 (-13.9%)**	**47.3 (-9.6%)**	**1.7 (-5.6%)**	**71.9 (-7.8%)**
Channel Pruning	Layer2 L1 C256	Layer2 L1 C128	4,688,656	**40%**	**16.4 (-36.4%)**	**45.8 (-1.9%)**	1.8 (+5.9%)	**67.0 (-6.2%)**
L & Channel Pruning	Layer2 L1 C256	Layer2 L0 C128	3,847,696	**40%**	**18.9 (-36.1%)**	**51.7 (-1.1%)**	1.9 (+5.5%)	**67.2 (-13.8%)**

Notation: Layer$x$ Ly Cz: x message-passing layers, max order y, z channels.

MACE-MP results: Table 2 compares pruned models against from-scratch baselines of the same target architecture. Our method compresses model parameters up to 4$\times$. Pruned models achieve lower Force MAE and substantially better out-of-distribution performance on the sAlex [59, 36, 60] dataset

compared to from-scratch baselines of the same size, indicating that the pruned model effectively inherits the generalized feature representations of the larger parent model.

MACE-OFF results: Table 3 presents similar findings for organic molecules. Across diverse test subsets, the pruned model retrained with 100% of the data outperforms the from-scratch baseline of the same size on all metrics, and already matches it with only 50% of the data, demonstrating that the pruning process yields a higher-quality compact model while preserving broad coverage of chemical space.

Table 3: **Accuracy of pruning methods on MACE-OFF.** The table compares the performance of pruned models against baselines trained from scratch. Metrics include MAE for Energy (E, meV/atom) and Forces (F, meV/Å) across diverse molecular subsets. Data indicates the fraction of training data used. Bold values indicate that the pruned model outperforms the from-scratch trained baseline of the same size.

Method	From Scratch				L & Channel Pruning			
Params	4,707,312		1,428,368		1,428,368		1,428,368	
Data	100%		100%		**50%**		100%	
Subset	**E**	**F**	**E**	**F**	**E**	**F**	**E**	**F**
DES370K Dimers	0.5	6.6	0.6	9.2	0.6	9.8	**0.5**	**9.0**
DES370K Monomers	0.6	6.6	0.6	9.5	0.6	10.1	**0.5**	**9.3**
Dipeptides	0.4	10.2	0.5	14.5	**0.5**	15.3	**0.5**	**14.3**
PubChem	0.9	14.8	0.9	22.0	0.9	23.2	**0.8**	**21.7**
QMugs	0.5	17.0	0.6	24.2	**0.6**	25.0	**0.6**	**23.5**
Solvated Amino Acids	1.0	19.4	1.3	23.8	**1.2**	24.3	**1.2**	**22.6**
Water	0.8	13.6	0.8	16.1	0.8	16.4	**0.7**	**15.5**

Generalization to large-scale benchmarks: To further validate that the accuracy gains extend beyond the pre-training distribution, we evaluate on Matbench Discovery [61], a large-scale screening benchmark across 250,000+ candidate materials. Our pruned MACE-MP model (Layer2 L0 C128) surpasses the official MACE-MP-0 checkpoint (Layer2 L1 C128) on 7/9 metrics using only 25% of MPtrj (Table 4), demonstrating that the pruned model is not only comparable to but can exceed the accuracy of from-scratch-trained foundation models in thermodynamic stability prediction.

Table 4: **Matbench Discovery results.** Our pruned model is compared against the official MACE-MP-0 baseline. Bold values indicate the better result on each metric. ↑: higher is better; ↓: lower is better.

Model	**F1** ↑	**DAF** ↑	**Prec.** ↑	**Recall** ↑	**Acc.** ↑	**MAE** ↓	$\mathbf{R}^2$ ↑	κ_{SRME} ↓	**RMSD** ↓
MACE-MP-0 (Official)	0.669	3.777	0.577	**0.796**	0.878	0.057	0.697	0.682	**0.0915**
Pruned (Ours)	**0.680**	**3.889**	**0.595**	0.794	**0.884**	**0.056**	**0.717**	**0.625**	0.0917

Architectural generalizability: To verify that our framework applies broadly to SO(3) equivariant architectures beyond MACE, we evaluate on SevenNet [3] and eSCN [10]. As summarized in Table 5, the pruned SevenNet surpasses the scratch baseline Force MAE with only 25% of MPtrj, and the pruned eSCN trained on 1M OC20 [62] samples surpasses the scratch baseline trained on 2M, achieving a 2× reduction in data requirements. Full details are in Appendix E.

Table 5: **Generalizability across architectures.** The pruned model matches the from-scratch baseline with a fraction of the training data.

Architecture	Dataset	From Scratch		Pruned	
		Data	Force MAE (meV/Å)	Data	Force MAE (meV/Å)
SevenNet	MPtrj	100%	40	25%	**39**
eSCN	OC20-2M	100%	19.1	50%	**18.4**

4.3 Efficiency

Compact architectures are only practically useful if they deliver concrete reductions in computational cost. We evaluate the efficiency of the pruned models along two axes: end-to-end training cost and inference throughput.

Training cost: As shown in Figure 3, the pruned model consistently outperforms from-scratch training across all data scales, and matches the scratch baseline accuracy with only 25–40% of the data. This translates directly to lower training cost. The full pipeline (calibration, pruning, and retraining on 25% of MPtrj) requires approximately 181 GPU-hours on an A800, compared to ∼750 GPU-hours for pre-training a model from scratch, confirming a **4**× reduction in training cost (Appendix D.10).

Inference efficiency: Using the MPtrj dataset as an example, on an NVIDIA A800 GPU, our pruned models achieve a 2.7× throughput improvement and a 5.7× reduction in peak memory usage compared to the original foundation models. On an RTX 3090, structural pruning yields a 3.0× throughput improvement, confirming that the efficiency gains are hardware-agnostic (Appendix D.11).

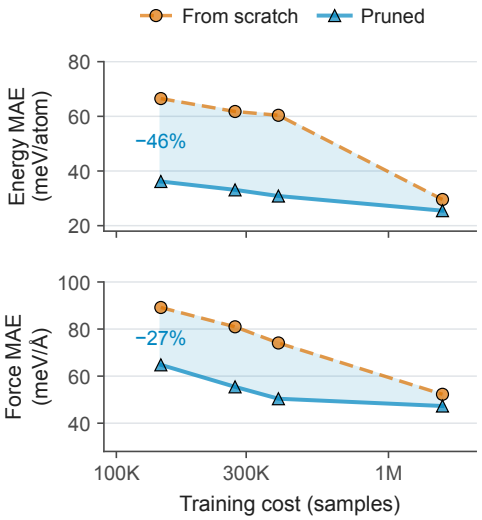

Figure 3: **Energy and force MAE vs. training data size on MACE-MP** (Layer2 L2 C128 → Layer2 L0 C128). Shading indicates the MAE gap between the two models. The pruned model outperforms training from scratch across all data scales.

4.4 Application

We evaluate the utility of the pruned models as initializations for downstream fine-tuning. A key question is whether a pruned-then-retrained model, despite using only a fraction of the pre-training data, can serve as a good starting point.

Inorganic materials: Figure 4 compares energy and force RMSE after fine-tuning on eight inorganic datasets. We compare three initialization strategies: Random Initialization (From Scratch), an official pre-trained checkpoint (using 100% MPtrj data), and our pruned checkpoint (using only 25% MPtrj data). The results show that the Pruned (25%) initialization significantly outperforms a small model trained from scratch, reducing energy and force errors by 70.1% and 34.4% respectively, while remaining highly competitive with the full pre-training baseline that uses 4× more data. To further validate the deployment readiness of the fine-tuned model, we run MD simulations of liquid water. The fine-tuned pruned model achieves energy drift <0.01 meV/atom/ps and faithfully reproduces structural properties (Appendix D.12).

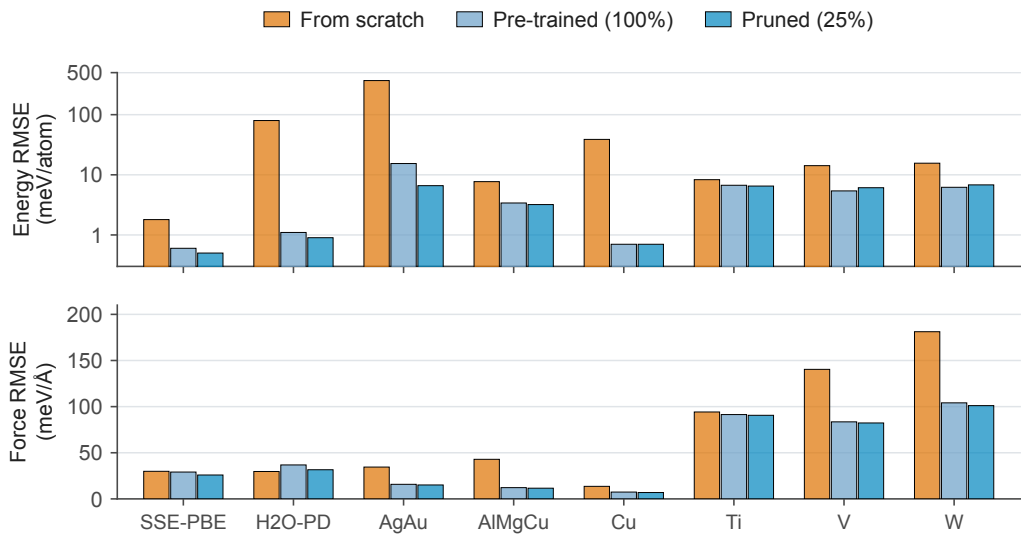

Figure 4: **Downstream fine-tuning performance on eight inorganic datasets.** Bars show energy and force RMSE for three initialization strategies. The energy axis uses a logarithmic scale, while the force axis is linear. Lower values indicate better accuracy. Averaged equally across datasets, pruning reduces energy and force RMSE by 70.1% and 34.4% relative to training from scratch. Percentages in the legend denote the fraction of MPtrj used before fine-tuning. Full RMSE values are reported in Table 21 in Appendix D.13.

Organic molecules: For molecular tasks on 3BPA, AcAc, and rMD17 (see Appendix D.14), pruned MACE-OFF models match or exceed from-scratch baselines across temperature regimes and dihedral configurations, confirming that compact specialized models retain the expressivity needed for complex potential energy surfaces.

Overall, these experiments confirm that our structural pruning framework successfully reduces pre-training cost while retaining transferability. By preserving the critical weights of the large model, the compressed models serve as highly effective, lightweight starting points for downstream applications.

4.5 Combination with other compression methods

As noted in Table 1, structural pruning can be combined with both quantization and knowledge distillation. We verify empirically that these techniques can be combined with our method to achieve compounding gains beyond what each method achieves alone.

Combination with quantization: As structural pruning and quantization reduce different aspects of model cost (FLOPs/params vs. bit-precision), they can be combined for compounding gains. Table 6 reports peak throughput on an RTX 3090 for the base model and pruned models under FP64 and FP32 precision. Structural pruning alone yields a 3.0× speedup; reducing precision to FP32 alone yields 2.0×; applying both together achieves 9.6× combined throughput improvement (2,728 to 26,144 atom/s).

Table 6: **Combined speedup from structural pruning and quantization on RTX 3090.** Speedup is relative to the base FP64 baseline.

Model	**FP64 (atom/s)**	**FP32 (atom/s)**	**Speedup (vs. Base FP64) FP64 / FP32**
Base (Layer2 L2 C128)	2,728	5,576	1.0× / 2.0×
Pruned (Layer2 L1 C128)	4,976	12,268	1.8× / 4.5×
Pruned (Layer2 L0 C128)	8,271	26,144	3.0× / **9.6**×

Combination with knowledge distillation: Structural pruning and knowledge distillation address different aspects of model compression: pruning inherits weights from a large pre-trained model to produce a compact initialization, while KD transfers knowledge from a teacher to a student through soft labels during training. Here we explore the combination of our structural pruning method with Hessian-based KD [48], which uses second-order information from the teacher model to guide student training. Table 7 compares two initialization strategies for training a Layer2 L0 C128 student model with Hessian labels generated from a Layer2 L2 C128 teacher: random initialization (From Scratch) versus pruning-based initialization (Pruned). With 9% of MPtrj, the pruned initialization without KD (Force MAE 64.8) already outperforms random initialization with Hessian KD (68.4). Combining pruned initialization with Hessian KD further reduces it to 64.5. With 25% of MPtrj, pruned initialization alone reaches 50.8, and adding Hessian KD yields 50.7. These results demonstrate that structural pruning and KD are complementary: pruning provides a better starting point, while KD further refines the model through teacher guidance.

Table 7: **Effect of Hessian-based KD combined with structural pruning** (target: Layer2 L0 C128). MAE for Energy (E, meV/atom), Force (F, meV/Å), Stress (S, meV/Å^3) on MPtrj.

Init.	Objective	Data	MAE E	MAE F	MAE S
From Scratch	Standard	9%	66.5	89.2	2.7
From Scratch	+Hessian KD	9%	**37.2**	**68.4**	**2.3**
Pruned	Standard	9%	36.2	64.8	**2.2**
Pruned	+Hessian KD	9%	**32.8**	**64.5**	**2.2**
Pruned	Standard	25%	30.9	50.8	**1.9**
Pruned	+Hessian KD	25%	**23.6**	**50.7**	**1.9**

These results confirm that our structural pruning method can be combined with both quantization and knowledge distillation. Because each technique addresses a fundamentally different axis of model cost (parameter count, bit-precision, and generalization gap, respectively), their benefits compound rather than overlap. This complementarity means practitioners can freely compose these techniques according to their deployment constraints, applying any subset of the three without conflict.

5 Conclusion

We present a structural pruning framework for SO(3) equivariant atomistic foundation models. By pruning at the (k, l)-block granularity with an SO(3) invariant importance criterion, our method produces compact foundation models that strictly preserve SO(3) equivariance and achieve 1.5×–4× parameter reduction. From the accuracy perspective, pruned models outperform same-size models trained from scratch on both in-distribution and out-of-distribution benchmarks. From the efficiency perspective, they reduce training cost by 2.5×–4× and achieve 2.7× inference speedup. From the application perspective, they serve as superior initializations for downstream fine-tuning, reducing energy and force errors by 70.1% and 34.4% compared to training from scratch. Structural pruning can be combined with quantization and knowledge distillation for compounding gains.

The core principle of our method, evaluating the importance of coupled irreducible representations, is broadly applicable to any equivariant architecture that represents features as sums of SO(3) irreducible representations, as validated on MACE, SevenNet, and eSCN. A current limitation is that the target pruned architecture must be specified in advance. Incorporating neural architecture search to automatically discover optimal configurations is a promising direction for future work, enabling more flexible compression tailored to specific deployment constraints.

Acknowledgments

This work is supported by the following funding: National Natural Science Foundation of China (92270206, 62372435, T2125013), and the Innovation Funding of ICT, CAS. Part of the training is performed on the robotic AI-Scientist platform of Chinese Academy of Sciences. The authors declare no competing interests.

References

[1] Bhupalee Kalita, Hatice Gokcan, and Olexandr Isayev. Machine learning interatomic potentials at the centennial crossroads of quantum mechanics. *Nature Computational Science*, pages 1–13, 2025.

[2] 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.

[3] Yutack Park, Jaesun Kim, Seungwoo Hwang, and Seungwu Han. Scalable parallel algorithm for graph neural network interatomic potentials in molecular dynamics simulations. *J. Chem. Theory Comput.*, 20 (11):4857–4868, 2024. doi: 10.1021/acs.jctc.4c00190.

[4] 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.

[5] 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.

[6] Arslan Mazitov, Filippo Bigi, Matthias Kellner, Paolo Pegolo, Davide Tisi, Guillaume Fraux, Sergey Pozdnyakov, Philip Loche, and Michele Ceriotti. PET-MAD as a lightweight universal interatomic potential for advanced materials modeling. *Nature Communications*, 16(1):10653, 2025.

[7] Seung Yul Lee, Hojoon Kim, Yutack Park, Dawoon Jeong, Seungwu Han, Yeonhong Park, and Jae W Lee. FlashTP: Fused, sparsity-aware tensor product for machine learning interatomic potentials. In *Forty-second International Conference on Machine Learning*, 2025.

[8] Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018.

[9] Mario Geiger and Tess Smidt. e3nn: Euclidean neural networks. *arXiv preprint arXiv:2207.09453*, 2022.

[10] Saro Passaro and C Lawrence Zitnick. Reducing SO(3) convolutions to SO(2) for efficient equivariant GNNs. In *International conference on machine learning*, pages 27420–27438. PMLR, 2023.

[11] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International journal of computer vision*, 129(6):1789–1819, 2021.

[12] Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. A survey of quantization methods for efficient neural network inference. In *Low-power computer vision*, pages 291–326. Chapman and Hall/CRC, 2022.

[13] Manish Gupta and Puneet Agrawal. Compression of deep learning models for text: A survey. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 16(4):1–55, 2022.

[14] Mengzhou Xia, Tianyu Gao, Zhiyuan Zeng, and Danqi Chen. Sheared llama: Accelerating language model pre-training via structured pruning. *arXiv preprint arXiv:2310.06694*, 2023.

[15] Saleh Ashkboos, Maximilian L Croci, Marcelo Gennari do Nascimento, Torsten Hoefler, and James Hensman. Slicegpt: Compress large language models by deleting rows and columns. *arXiv preprint arXiv:2401.15024*, 2024.

[16] Xin Men, Mingyu Xu, Qingyu Zhang, Qianhao Yuan, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, and Weipeng Chen. Shortgpt: Layers in large language models are more redundant than you expect. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 20192–20204, 2025.

[17] Yasir Ghunaim, Andrés Villa, Gergo Ignacz, Gyorgy Szekely, Motasem Alfarra, and Bernard Ghanem. Towards faster and more compact foundation models for molecular property prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 48–57, 2025.

[18] Lingyu Kong, Jaeheon Shim, Guoxiang Hu, and Victor Fung. Scalable foundation interatomic potentials via message-passing pruning and graph partitioning. *arXiv preprint arXiv:2509.21694*, 2025.

[19] Ilyes Batatia, David P Kovacs, Gregor Simm, Christoph Ortner, and Gábor Csányi. MACE: Higher order equivariant message passing neural networks for fast and accurate force fields. *Advances in neural information processing systems*, 35:11423–11436, 2022.

[20] Han Wang, Linfeng Zhang, Jiequn Han, et al. DeePMD-kit: A deep learning package for many-body potential energy representation and molecular dynamics. *Computer Physics Communications*, 228:178–184, 2018.

[21] Kristof Schütt, Oliver Unke, and Michael Gastegger. Equivariant message passing for the prediction of tensorial properties and molecular spectra. In *International conference on machine learning*, pages 9377–9388. PMLR, 2021.

[22] 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.

[23] Simon Batzner, Albert Musaelian, Lixin Sun, Mario Geiger, Jonathan P Mailoa, Mordechai Kornbluth, Nicola Molinari, Tess E Smidt, and Boris Kozinsky. E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature communications*, 13(1):2453, 2022.

[24] Chuin Wei Tan, Marc L Descoteaux, Mit Kotak, Gabriel de Miranda Nascimento, Seán R Kavanagh, Laura Zichi, Menghang Wang, Aadit Saluja, Yizhong R Hu, Tess Smidt, et al. High-performance training and inference for deep equivariant interatomic potentials. *arXiv preprint arXiv:2504.16068*, 2025.

[25] Yury Lysogorskiy, Anton Bochkarev, and Ralf Drautz. Graph atomic cluster expansion for foundational machine learning interatomic potentials. *arXiv preprint arXiv:2508.17936*, 2025.

[26] Duo Zhang, Anyang Peng, 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. *arXiv preprint arXiv:2506.01686*, 2025.

[27] 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*, 2024.

[28] Eric Qu, Brandon M Wood, Aditi S Krishnapriyan, and Zachary W Ulissi. A recipe for scalable attention-based MLIPs: unlocking long-range accuracy with all-to-all node attention. *arXiv preprint arXiv:2603.06567*, 2026.

[29] Yuanchang Zhou, Siyu Hu, Xiangyu Zhang, Hongyu Wang, Guangming Tan, and Weile Jia. MatRIS: Toward reliable and efficient pretrained machine learning interatomic potentials. *arXiv preprint arXiv:2603.02002*, 2026.

[30] Yuanchang Zhou, Siyu Hu, Chen Wang, Lin-Wang Wang, Guangming Tan, and Weile Jia. FastCHGNet: Training one universal interatomic potential to 1.5 hours with 32 GPUs. In *2025 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pages 236–246. IEEE, 2025.

[31] Yuanchang Zhou, Hongyu Wang, Yiming Du, Yan Wang, Mingzhen Li, Siyu Hu, Xiangyu Zhang, Weijian Liu, Chen Wang, Zhuoqiang Guo, et al. Breaking the training barrier of billion-parameter universal machine learning interatomic potentials. *arXiv preprint arXiv:2604.15821*, 2026.

[32] Thorben Frank, Oliver Unke, and Klaus-Robert Müller. So3krates: Equivariant attention for interactions on arbitrary length-scales in molecular systems. *Advances in Neural Information Processing Systems*, 35: 29400–29413, 2022.

[33] Adil Kabylda, J Thorben Frank, Sergio Suarez-Dou, Almaz Khabibrakhmanov, Leonardo Medrano Sandonas, Oliver T Unke, Stefan Chmiela, Klaus-Robert Müller, and Alexandre Tkatchenko. Molecular simulations with a pretrained neural network and universal pairwise force fields. *Journal of the American Chemical Society*, 147(37):33723–33734, 2025.

[34] Bowen Deng, Peichen Zhong, KyuJung Jun, Janosh Riebesell, Kevin Han, Christopher J Bartel, and Gerbrand Ceder. CHGNet as a pretrained universal neural network potential for charge-informed atomistic modelling. *Nature Machine Intelligence*, 5(9):1031–1041, 2023.

[35] 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.

[36] 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.

[37] Daniel S Levine, Muhammed Shuaibi, Evan Walter Clark Spotte-Smith, Michael G Taylor, Muhammad R Hasyim, Kyle Michel, Ilyes Batatia, Gábor Csányi, Misko Dzamba, Peter Eastman, et al. The open molecules 2025 (omol25) dataset, evaluations, and models. *arXiv preprint arXiv:2505.08762*, 2025.

[38] Vahe Gharakhanyan, Luis Barroso-Luque, Yi Yang, Muhammed Shuaibi, Kyle Michel, Daniel S Levine, Misko Dzamba, Xiang Fu, Meng Gao, Xingyu Liu, et al. Open molecular crystals 2025 (OMC25) dataset and models. *Scientific Data*, 2026.

[39] Anuroop Sriram, Logan M Brabson, Xiaohan Yu, Sihoon Choi, Kareem Abdelmaqsoud, Elias Moubarak, Pim de Haan, Sindy Löwe, Johann Brehmer, John R Kitchin, et al. The open DAC 2025 dataset for sorbent discovery in direct air capture. *arXiv preprint arXiv:2508.03162*, 2025.

[40] Kristof T Schütt, Stefan Chmiela, O Anatole Von Lilienfeld, Alexandre Tkatchenko, Koji Tsuda, and Klaus-Robert Müller. Machine learning meets quantum physics. *Lecture Notes in Physics*, 2020.

[41] 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.

[42] Mariia Radova, Wojciech G Stark, Connor S Allen, Reinhard J Maurer, and Albert P Bartók. Fine-tuning foundation models of materials interatomic potentials with frozen transfer learning. *npj Computational Materials*, 11(1):237, 2025.

[43] Harveen Kaur, Flaviano Della Pia, Ilyes Batatia, Xavier R Advincula, Benjamin X Shi, Jinggang Lan, Gábor Csányi, Angelos Michaelides, and Venkat Kapil. Data-efficient fine-tuning of foundational models for first-principles quality sublimation enthalpies. *Faraday Discussions*, 256:120–138, 2025.

[44] Chen Wang, Siyu Hu, Guangming Tan, and Weile Jia. ELoRA: low-rank adaptation for equivariant GNNs. In *Proceedings of the 42nd International Conference on Machine Learning*, ICML'25. JMLR.org, 2025.

[45] Lei Deng, Guoqi Li, Song Han, Luping Shi, and Yuan Xie. Model compression and hardware acceleration for neural networks: A comprehensive survey. *Proceedings of the IEEE*, 108(4):485–532, 2020.

[46] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018.

[47] 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: 25761–25792, 2023.

[48] Ishan Amin, Sanjeev Raja, and Aditi Krishnapriyan. Towards fast, specialized machine learning force fields: Distilling foundation models via energy hessians. *arXiv preprint arXiv:2501.09009*, 2025.

[49] Chen Wang, Siyu Hu, Guangming Tan, and Weile Jia. Architecture-agnostic knowledge distillation for machine-learning interatomic potentials. *Neural Networks*, page 109742, 2026.

[50] Alexandre Benoit. Speeding up MACE: Low-precision tricks for equivarient force fields. *arXiv preprint arXiv:2510.23621*, 2025.

[51] Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? *Advances in neural information processing systems*, 32, 2019.

[52] Ziheng Wang, Jeremy Wohlwend, and Tao Lei. Structured pruning of large language models. In *Proceedings of the 2020 conference on empirical methods in natural language processing (emnlp)*, pages 6151–6162, 2020.

[53] Xinyin Ma, Gongfan Fang, and Xinchao Wang. Llm-pruner: On the structural pruning of large language models. *Advances in neural information processing systems*, 36:21702–21720, 2023.

[54] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

[55] Dávid Péter Kovács, J Harry Moore, Nicholas J Browning, Ilyes Batatia, Joshua T Horton, Yixuan Pu, Venkat Kapil, William C Witt, Ioan-Bogdan Magdau, Daniel J Cole, et al. MACE-OFF: Short-range transferable machine learning force fields for organic molecules. *Journal of the American Chemical Society*, 147(21):17598–17611, 2025.

[56] Dávid Péter Kovács, Cas van der Oord, Jiri Kucera, Alice EA Allen, Daniel J Cole, Christoph Ortner, and Gábor Csányi. Linear atomic cluster expansion force fields for organic molecules: beyond rmse. *Journal of chemical theory and computation*, 17(12):7696–7711, 2021.

[57] Ilyes Batatia, Simon Batzner, Dávid Péter Kovács, Albert Musaelian, Gregor NC Simm, Ralf Drautz, Christoph Ortner, Boris Kozinsky, and Gábor Csányi. The design space of E(3)-equivariant atom-centered interatomic potentials. *arXiv preprint arXiv:2205.06643*, 2022.

[58] Anders S Christensen and O Anatole Von Lilienfeld. On the role of gradients for machine learning of molecular energies and forces. *Machine Learning: Science and Technology*, 1(4):045018, 2020.

[59] Jonathan Schmidt, Tiago FT Cerqueira, Aldo H Romero, Antoine Loew, Fabian Jäger, Hai-Chen Wang, Silvana Botti, and Miguel AL Marques. Improving machine-learning models in materials science through large datasets. *Materials Today Physics*, 48:101560, 2024.

[60] Xiang Fu, Brandon M Wood, Luis Barroso-Luque, Daniel S Levine, Meng Gao, Misko Dzamba, and C Lawrence Zitnick. Learning smooth and expressive interatomic potentials for physical property prediction. In *Forty-second International Conference on Machine Learning*, 2025.

[61] Janosh Riebesell, Rhys EA Goodall, Philipp Benner, Yuan Chiang, Bowen Deng, Alpha A Lee, Anubhav Jain, and Kristin A Persson. Matbench Discovery–a framework to evaluate machine learning crystal stability predictions. *arXiv preprint arXiv:2308.14920*, 2023.

[62] 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.

[63] Saurav Muralidharan, Sharath Turuvekere Sreenivas, Raviraj Joshi, Marcin Chochowski, Mostofa Patwary, Mohammad Shoeybi, Bryan Catanzaro, Jan Kautz, and Pavlo Molchanov. Compact language models via pruning and knowledge distillation. *Advances in Neural Information Processing Systems*, 37:41076–41102, 2024.

[64] Jianxing Huang, Linfeng Zhang, Han Wang, Jinbao Zhao, Jun Cheng, et al. Deep potential generation scheme and simulation protocol for the Li10GeP2S12-type superionic conductors. *The Journal of Chemical Physics*, 154(9), 2021.

[65] Linfeng Zhang, Han Wang, Roberto Car, and Weinan E. Phase diagram of a deep potential water model. *Physical review letters*, 126(23):236001, 2021.

[66] YiNan Wang, LinFeng Zhang, Ben Xu, XiaoYang Wang, and Han Wang. A generalizable machine learning potential of Ag–Au nanoalloys and its application to surface reconstruction, segregation and diffusion. *Modelling and Simulation in Materials Science and Engineering*, 30(2):025003, 2021.

[67] Wanrun Jiang, Yuzhi Zhang, Linfeng Zhang, and Han Wang. Accurate deep potential model for the Al–Cu–Mg alloy in the full concentration space. *Chinese Physics B*, 30(5):050706, 2021.

[68] Yuzhi Zhang, Haidi Wang, Weijie Chen, Jinzhe Zeng, Linfeng Zhang, Han Wang, and E Weinan. DP-GEN: A concurrent learning platform for the generation of reliable deep learning based potential energy models. *Computer Physics Communications*, 253:107206, 2020.

[69] Tongqi Wen, Rui Wang, Lingyu Zhu, Linfeng Zhang, Han Wang, David J Srolovitz, and Zhaoxuan Wu. Specialising neural network potentials for accurate properties and application to the mechanical response of titanium. *npj Computational Materials*, 7(1):206, 2021.

[70] Rui Wang, Xiaoxiao Ma, Linfeng Zhang, Han Wang, David J Srolovitz, Tongqi Wen, and Zhaoxuan Wu. Classical and machine learning interatomic potentials for BCC vanadium. *Physical Review Materials*, 6 (11):113603, 2022.

[71] Xiaoyang Wang, Yinan Wang, Linfeng Zhang, Fuzhi Dai, and Han Wang. A tungsten deep neural-network potential for simulating mechanical property degradation under fusion service environment. *Nuclear Fusion*, 62(12):126013, 2022.

[72] Mario Geiger, Emine Kucukbenli, Becca Zandstein, and Kyle Tretina. Accelerate drug and material discovery with new math library nvidia cuequivariance. *URL https://github.com/NVIDIA/cuEquivariance*, 2024.

[73] Zhao Xu, Haiyang Yu, Montgomery Bohde, and Shuiwang Ji. Equivariant graph network approximations of high-degree polynomials for force field prediction. *arXiv preprint arXiv:2411.04219*, 2024.

A Background

Machine-learning interatomic potentials (MLIPs). The primary objective of MLIPs is to approximate the Potential Energy Surface (PES) of an atomic system with the accuracy of *ab initio* quantum mechanical methods but at a fraction of the computational cost. Consider a system composed of N atoms, defined by their Cartesian coordinates $\mathbf{R} = \{\mathbf{r}_1, \ldots, \mathbf{r}_N\} \in \mathbb{R}^{N\times 3}$ and their chemical species (atomic numbers) $\mathbf{Z} = \{z_1, \ldots, z_N\} \in \mathbb{Z}^N$. An MLIP learns a parameterized function $E = \Phi_\theta(\mathbf{R}, \mathbf{Z})$ that predicts the total potential energy E of the system.

Additionally, these models also provide accurate predictions of interatomic forces to drive MD simulations. Since the force field is conservative, the force $\mathbf{F}_i$ acting on the i-th atom is derived as the negative gradient of the energy with respect to its position:

$$\mathbf{F}_i = -\nabla_{\mathbf{r}_i} \Phi_\theta(\mathbf{R}, \mathbf{Z}). \tag{4}$$

Atomistic foundation models extend this framework by training Φ_θ on massive datasets spanning diverse chemical spaces to achieve broad generalizability across unseen systems. To ensure physical consistency and data efficiency, the architecture of Φ_θ is typically designed to respect the geometric symmetries of the 3D space.

SO(3) equivariant message passing. MLIPs typically ensure physical symmetries by constructing an energy function E that is SE(3) (translational and rotational) invariant. Since forces are derived via gradients, the invariance of E naturally guarantees the equivariance of $\mathbf{F}_i$. While early architectures relied on invariant scalars (e.g., distances, angles), recent research indicates that incorporating SO(3) (rotational) equivariant geometric tensors significantly improves data efficiency and accuracy [9, 23].

In this framework, node features are generalized to include higher-order tensors. We denote the feature component of atom j at layer t as $h^{(t)}_{j,klm}$, where k indexes the feature channel, and l, m denote the order and representation index of the SO(3) irreducible representation. The SO(3) Equivariant Message Passing aggregates information from neighbors via a tensor product [8]. The message for atom i at channel k with target order l_3 and representation m_3 is given by:

$$\sum_{l_1 m_1, l_2 m_2} C^{l_3 m_3}_{l_1 m_1, l_2 m_2} \sum_{j\in\mathcal{N}(i)} R^{(t)}_{k l_1 l_2 l_3}(r_{ji}) Y^{m_1}_{l_1}(\hat{\mathbf{r}}_{ji}) h^{(t)}_{j,kl_2m_2}. \tag{5}$$

Here, $h^{(t)}_{j,kl_2m_2}$ is the input feature of the neighbor with order l_2. The interaction couples these features with the spherical harmonics $Y^{m_1}_{l_1}$ of the edge direction using Clebsch-Gordan coefficients $C^{l_3 m_3}_{l_1 m_1, l_2 m_2}$, which enforce the rules ($|l_1 - l_2| \le l_3 \le (l_1 + l_2)$). The learnable radial function $R^{(t)}_{k l_1 l_2 l_3}$ modulates the interaction strength based on the interatomic distance r_{ji} and the specific channel k. This operation has a computational complexity of $\mathcal{O}(L^6_{\text{max}})$, making higher-order interactions computationally expensive.

The tensor product operates in a channel-wise manner, processing each channel k independently. To enable information exchange across different channels, the architecture incorporates learnable linear layers to mix feature channels while strictly preserving SO(3) equivariance by operating only on features of the same order l:

$$\sum_{\tilde{k}} W^{(t)}_{k\tilde{k}l} h^{(t)}_{j,\tilde{k}lm}. \tag{6}$$

Structural pruning. Model pruning compresses models by eliminating redundant parameters. While unstructured pruning creates sparse matrices that often require specialized hardware, structural pruning removes coherent groups, such as entire layers or channels. This approach preserves dense matrix structures, enabling direct acceleration on standard hardware. The core of structural pruning lies in identifying and removing the least significant structures based on a specific importance criterion. Common scoring methods include the magnitude [54], activation [63], or gradient [53].

Structural pruning generally operates along two dimensions: depth and width. Taking a Multi-Layer Perceptron (MLP) as an example:

$$\mathbf{W}_n \sigma(\ldots \sigma(\mathbf{W}_2 \sigma(\mathbf{W}_1 x))), \tag{7}$$

where layer pruning reduces the depth by removing a weight matrix $\mathbf{W}_k$, effectively shortening the inference path. Conversely, channel pruning reduces the dimension of the matrices by removing specific rows in $\mathbf{W}_k$ (and corresponding columns in $\mathbf{W}_{k+1}$), which shrinks the layer size.

However, applying structural pruning to SO(3) equivariant models is non-trivial. As features are higher-order tensors grouped by order l and representation m, naive pruning can break the symmetry of the message passing path.

B Motivation and theory

B.1 Depth pruning

Message Passing Neural Networks (MPNNs) approximate the potential energy surface of atomic systems by iteratively refining node features through interactions with neighbors. A critical hyperparameter in these architectures is the number of message passing layers, denoted as T. This parameter controls not only the depth of the network but also the physical range of atomic interactions the model can capture—its receptive field.

In the MLIPs architecture, the total energy is a sum of atomic energies, which are determined by the local environment of each atom. The extent of this environment is defined by the receptive field R_{rec}. We formalize the relationship between the network depth and the receptive field with the following proposition:

Proposition B.1. *Let r_c be the radial cutoff distance for a single message passing iteration. For a Message Passing Neural Network with T layers, the total receptive field R_{rec} of an atom i is given by:*

$$R_{rec} = T \times r_c \tag{8}$$

Proof. We proceed by induction on the number of layers T.

Base case ($T = 1$): In the first layer, atom i receives messages from all neighbors j such that the Euclidean distance $d_{ij} \leq r_c$. Thus, the features $\boldsymbol{h}_i^{(1)}$ depend on atoms within a radius $R_{\text{rec}}^{(1)} = r_c$.

Inductive step: Assume that for $T = k$, the features $\boldsymbol{h}_i^{(k)}$ depend on all atoms within a distance $k \times r_c$. In the $(k+1)$-th layer, atom i aggregates messages from neighbors j within distance r_c. Each neighbor j has features $\boldsymbol{h}_j^{(k)}$, which depend on atoms within distance $k \times r_c$ from j by the inductive hypothesis. By the triangle inequality, an atom m influencing j (where $d_{jm} \leq k \times r_c$) is at a distance from i satisfying:

$$d_{im} \leq d_{ij} + d_{jm} \leq r_c + k \times r_c = (k+1)r_c \tag{9}$$

Therefore, the features $\boldsymbol{h}_i^{(k+1)}$ depend on atoms within a radius $R_{\text{rec}}^{(k+1)} = (k+1)r_c$. □

This linear scaling implies that reducing the number of layers (Depth Pruning) directly shrinks the physical volume of the atomic environment that the model can perceive. In contrast, reducing the feature channels or order (Width Pruning) simplifies the mathematical description of the interactions within that volume but preserves the spatial scope of the receptive field. Consequently, we hypothesize that Depth Pruning will result in a more significant loss of accuracy compared to Width Pruning, particularly for systems where long-range interactions are non-negligible.

B.2 Preservation of SO(3) equivariance

Standard pruning methods often operate on individual elements of the feature maps. We show here that such element-wise pruning breaks SO(3) equivariance, whereas our proposed (k, l)-block pruning preserves it.

Proposition B.2. *Let $\mathbf{h}_l \in \mathbb{R}^{2l+1}$ be a feature vector of order l. Let $g \in SO(3)$ be a rotation, and $\mathbf{D}^l(g)$ be the Wigner-D matrix acting on $\mathbf{h}_l$.*

1. ***Element-wise Pruning:** Let $\mathbf{M} = diag(m_{-l}, \ldots, m_l)$ be an arbitrary diagonal binary mask where not all m_i are equal. Applying $\mathbf{M}$ element-wise violates equivariance:*

$$\mathbf{M}(\mathbf{D}^l(g)\mathbf{h}_l) \neq \mathbf{D}^l(g)(\mathbf{M}\mathbf{h}_l) \tag{10}$$

2. ***Block-wise Pruning:*** *Applying a scalar mask $z \in \{0, 1\}$ (our proposed method) preserves equivariance:*

$$z(\mathbf{D}^l(g)\mathbf{h}_l) = \mathbf{D}^l(g)(z\mathbf{h}_l) \tag{11}$$

Proof. **Part 1 (Violation):** The Wigner-D matrix $\mathbf{D}^l(g)$ is a dense $(2l+1) \times (2l+1)$ unitary matrix that mixes all components indexed by $m \in [-l, l]$. Let $\mathbf{h}' = \mathbf{D}^l(g)\mathbf{h}$. The i-th component of the rotated feature is $h'_i = \sum_j D^l(g)_{ij} h_j$. If we apply the mask $\mathbf{M}$ after rotation, the result is $m_i \sum_j D^l(g)_{ij} h_j$. If we apply the mask $\mathbf{M}$ before rotation, the rotated masked feature is $\sum_j D^l(g)_{ij}(m_j h_j)$. For equivariance to hold, we require $m_i \sum_j D^l(g)_{ij} h_j = \sum_j D^l(g)_{ij} m_j h_j$ for all $\mathbf{h}$ and g. This implies $m_i = m_j$ for all i, j where $D^l(g)_{ij} \neq 0$. Since $\mathbf{D}^l(g)$ generally has non-zero off-diagonal entries (mixing components), arbitrary element-wise masking ($m_i \neq m_j$) breaks this equality.

Part 2 (Preservation): Our method treats the subspace of order l as an atomic unit. The mask becomes a scalar $z \in \{0, 1\}$ multiplying the entire vector $\mathbf{h}_l$. Since scalar multiplication commutes with linear transformations:

$$z(\mathbf{D}^l(g)\mathbf{h}_l) = \mathbf{D}^l(g)(z\mathbf{h}_l) \tag{12}$$

Thus, the pruning operation commutes with the group action, satisfying the definition of equivariance. □

Since the irreducible representations of SO(3) inherently mix all m components during rotation, any pruning operation that differentiates between m values within the same order l will inevitably destroy the group structure. Therefore, the proposed (k, l)-block serves as the *minimal pruning granularity* capable of strictly preserving SO(3) equivariance.

B.3 Rotational invariance of the importance criterion

We now prove that our importance criterion $I^{(t)}_{k,l}$ is SO(3) invariant, ensuring that the pruning decisions are consistent under rotations of the atomic configuration.

Proposition B.3. *The importance score $I^{(t)}_{k,l} = \mathbb{E}_{\mathbf{R},\mathbf{Z}\sim\mathcal{D}} \left| \sum_{j,m} \frac{\partial E}{\partial h^{(t)}_{j,klm}} \cdot h^{(t)}_{j,klm} \right|$ is invariant under SO(3) rotations of the input atomic configuration.*

Proof. Let $g \in SO(3)$ be a rotation. Under this rotation, the atomic positions transform as $\mathbf{R} \to g\mathbf{R}$, and the features transform equivariantly: $h^{(t)}_{j,klm} \to \sum_{m'} D^l(g)_{mm'} h^{(t)}_{j,klm'}$, where $\mathbf{D}^l(g)$ is the Wigner-D matrix.

Since the energy E is rotationally invariant by construction, we have $E(g\mathbf{R}, \mathbf{Z}) = E(\mathbf{R}, \mathbf{Z})$. By the chain rule:

$$\frac{\partial E}{\partial h^{(t)}_{j,klm}} \to \sum_{m'} D^l(g)_{mm'} \frac{\partial E}{\partial h^{(t)}_{j,klm'}} \tag{13}$$

The inner product in the importance criterion transforms as:

$$\sum_m \frac{\partial E}{\partial h^{(t)}_{j,klm}} \cdot h^{(t)}_{j,klm} \to \sum_m \left(\sum_{m'} D^l(g)_{mm'} \frac{\partial E}{\partial h^{(t)}_{j,klm'}} \right) \left(\sum_{m''} D^l(g)_{mm''} h^{(t)}_{j,klm''} \right) \tag{14}$$

$$= \sum_{m,m',m''} D^l(g)_{mm'} D^l(g)_{mm''} \frac{\partial E}{\partial h^{(t)}_{j,klm'}} h^{(t)}_{j,klm''} \tag{15}$$

$$= \sum_{m',m''} \left(\sum_m D^l(g)_{mm'} D^l(g)_{mm''} \right) \frac{\partial E}{\partial h^{(t)}_{j,klm'}} h^{(t)}_{j,klm''} \tag{16}$$

For the real representations used here, $\mathbf{D}^l(g)$ is orthogonal, so $\sum_m D^l(g)_{mm'} D^l(g)_{mm''} = \delta_{m'm''}$. Therefore:

$$\sum_m \frac{\partial E}{\partial h^{(t)}_{j,klm}} \cdot h^{(t)}_{j,klm} \to \sum_{m'} \frac{\partial E}{\partial h^{(t)}_{j,klm'}} h^{(t)}_{j,klm'} \tag{17}$$

This shows that the sum over m is invariant under rotation. Since the absolute value and expectation operations preserve this invariance, the importance score $I_{k,l}^{(t)}$ is SO(3) invariant. □

This invariance property ensures that our pruning method produces consistent results under rotations of the molecular system in the calibration dataset.

B.4 Spectral interpretation of feature order L

Understanding the physical meaning of the order l helps justify why pruning higher-order features corresponds to reducing the angular resolution of the model.

Proposition B.4. *The feature order l corresponds to the angular frequency on the sphere. Higher l implies higher spatial frequency, capturing finer angular details of the atomic environment.*

Proof. The angular dependence of the atomic features is encoded using Spherical Harmonics $Y_{lm}(\theta, \phi)$. These functions are the eigenfunctions of the Laplace operator Δ_{S^2} on the unit sphere:

$$-\Delta_{S^2} Y_{lm}(\theta, \phi) = l(l+1) Y_{lm}(\theta, \phi) \tag{18}$$

In harmonic analysis on the sphere, the eigenvalue $\lambda_l = l(l+1)$ is analogous to the squared frequency $|\omega|^2$ in Fourier analysis on Euclidean space. A function $f(\theta, \phi)$ composed of spherical harmonics up to degree L_{max} has a band-limit determined by L_{max}. Increasing l increases the number of nodal lines on the sphere, allowing the function to represent more rapid changes in angular direction. Therefore, pruning features with high l is spectrally equivalent to applying a low-pass filter to the angular representation of the local atomic environment, smoothing out high-frequency geometric details while retaining the coarse structural information. □

To empirically validate this spectral interpretation, we present the feature importance analysis using our method in Figure 5. The experimental results reveal a consistent decay in feature importance as the order l increases. Specifically, the $l = 0$ features (low frequency) exhibit the highest importance scores, whereas higher-order features ($l = 1, 2, 3$) show progressively lower significance. This observation aligns with Proposition B.4, confirming that the model naturally prioritizes coarse-grained geometric information. Consequently, pruning high-l features is physically justified.

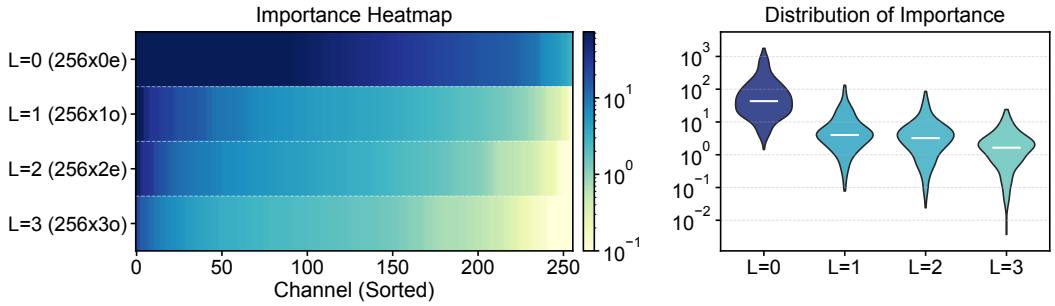

Figure 5: **Feature importance analysis across orders L.** The left panel displays the heatmap of feature importance sorted by channel for each order L, while the right panel shows the corresponding distribution (violin plots).

C Comparison of depth and width pruning

We perform additional comparisons of Depth (reducing the number of message passing layers) and Width pruning (reducing the order L of the internal features) on the MACE architecture.

C.1 Ablation study: non-linear readout substitution in depth pruning

In the standard MACE architecture, the energy is decomposed into contributions from each layer:

$$E_i = E_i^{(0)} + E_i^{(1)} + ... + E_i^{(T)}, \qquad \text{where}$$
$$E_i^{(t)} = \mathcal{R}_t\left(\boldsymbol{h}_i^{(t)}\right) = \begin{cases} \sum_{\tilde{k}} W^{(t)}_{\text{readout},\tilde{k}} h^{(t)}_{i,\tilde{k}00} & \text{if } t < T \\ \text{MLP}^{(t)}_{\text{readout}}\left(\left\{h^{(t)}_{i,k00}\right\}_k\right) & \text{if } t = T \end{cases} \tag{19}$$

where the readout function $\mathcal{R}_t$ is linear for all intermediate layers ($t < T$) and an MLP (non-linear) only for the final layer ($t = T$). This design ensures that the final layer can approximate higher-order residual terms effectively.

When performing Depth Pruning (e.g., pruning a 2-layer model down to 1 layer), simply removing the second layer leaves the first layer with its original linear readout. We hypothesize that this significantly limits the expressivity of the pruned model. To address this, we propose a Readout Substitution strategy: when the last layer is pruned, we replace the linear readout of the new last layer with a non-linear MLP, matching the architecture of a standard $T - 1$ layer MACE model.

Table 8 presents the ablation study of this substitution. We observe that retaining the linear readout results in high errors (MAE E: 64.0 meV/atom). However, applying the non-linear readout substitution yields a substantial improvement, reducing the Energy MAE by 26.3% (to 47.2 meV/atom) and Force MAE by 10.7%. This confirms that the non-linearity in the final readout is critical for the MACE architecture to resolve atomic energies accurately.

Table 8: **Ablation of readout substitution in depth pruning on MACE-MP.** Both variants prune Layer2 L2 C128 to Layer1 L2 C128 and use 9% of the training data. We evaluate Energy (E, meV/atom), Force (F, meV/Å), and Stress (S, meV/Å^3) errors on MPtrj.

Readout	Params	MAE E	MAE F	MAE S
Linear	3,453,696	64.0	86.6	2.7
Non-linear	3,455,632	**47.2**	**77.3**	**2.4**

C.2 Comparison results

We further compare the performance of Depth Pruning (with the optimal non-linear readout) against Width Pruning. For Width Pruning, we reduce the irreducible representations of the features from $L = 2$ to $L = 0$, effectively reducing the geometric tensor interactions while maintaining the message passing depth. We compare these pruned models against From Scratch baselines trained directly with the target architecture sizes.

As shown in Table 9, Width Pruning demonstrates superior performance. The width-pruned model (Layer 2 L0) achieves an Energy MAE of 30.9 meV/atom, which is competitive with the baseline trained from scratch (29.6 meV/atom). This suggests that initializing with higher-order equivariant features ($L = 2$) allows the model to learn a robust underlying representation that can be effectively transferred into invariant features ($L = 0$).

Conversely, Depth Pruning fails to match the baseline performance. The depth-pruned model (Layer 1 L2) exhibits an Energy MAE of 44.4 meV/atom, a 50% increase in error compared to the comparable scratch baseline. This disparity highlights a fundamental characteristic of the message-passing architecture: the receptive field is tightly coupled to the number of layers, as established in Proposition B.1. Depth Pruning truncates the message passing iterations, linearly reducing the physical receptive field ($R_{\text{rec}} = T \times r_c$). This loss of long-range spatial information proves critical for accuracy, whereas Width Pruning retains the full receptive field while only simplifying the local feature representation.

Table 9: **Comparison of depth and width pruning methods on MACE-MP.** The table compares the performance of pruned models against baselines trained from scratch. Metrics include MAE for Energy (E, meV/atom), Force (F, meV/Å), and Stress (S, meV/Å^3) on MPtrj, and Energy (E, meV/atom) on the sAlex subset. Data indicates the fraction of training data used. Values in parentheses show the relative change in MAE compared to from-scratch trained models of similar size. Bold values indicate that the pruned model outperforms the from-scratch trained baseline.

Method	Base Model	Pruned Model	Params	Data	MPtrj			sAlex
					MAE E	MAE F	MAE S	MAE E
From Scratch	Layer2 L1 C256	/	15,847,440	100%	/	/	/	81.7
	Layer2 L2 C128		5,725,072		22.6	42.2	1.6	65.1
	Layer2 L1 C128		4,688,656		25.8	46.7	1.7	71.4
	Layer2 L0 C128		3,847,696		29.6	52.3	1.8	78.0
Depth Pruning	Layer2 L2 C128	Layer1 L2 C128	3,455,632	**25%**	44.4 (+50.0%)	63.1 (+20.7%)	2.1 (+16.7%)	99.0 (+26.9%)
Width Pruning	Layer2 L2 C128	Layer2 L0 C128	3,847,696	**25%**	30.9 (+4.4%)	**50.8 (-2.9%)**	1.9 (+5.5%)	**74.1 (-5.0%)**

D Experimental details

D.1 Datasets

Inorganic downstream tasks. We evaluate on eight diverse inorganic datasets: solid-state electrolytes (SSE) [64], H2O [65], AgAu [66], AlMgCu [67], Cu [68], Ti [69], V [70], and W [71]. These datasets cover simulations of diverse inorganic compounds under complex temperature and pressure conditions.

Organic downstream tasks. For organic systems, we evaluate on 3BPA [56], a flexible drug-like molecule tested across temperature regimes (300 K, 600 K, 1200 K) and dihedral angle configurations; AcAc [57] (acetylacetone), which probes complex dihedral potential energy surfaces; and rMD17 [58], a benchmark of small organic molecules with high-quality DFT-computed energies and forces.

D.2 Calibration

To compute the importance scores during the calibration phase, we construct representative subsets of the pre-training data tailored to the specific distribution of each dataset. For the MPtrj dataset, which contains approximately 1.5 million configurations derived from roughly 150,000 unique structures, we employ a structure-based sampling approach to ensure chemical diversity and prevent bias toward trajectories with higher frame counts. Specifically, we randomly select n configurations for each unique structure. Setting $n = 1$, $n = 3$, and $n = 5$ results in calibration subsets covering approximately 9%, 25%, and 40% of the total dataset, respectively. For the SPICE dataset, we perform uniform sampling within each data category to generate the calibration subset.

D.3 Pruning details for MACE architecture

Order pruning and the $L = 0$ target. In our MACE notation, L specifies the maximum order of the hidden features passed between interaction blocks. At $L = 0$, these features are invariant, but higher-order equivariant tensors are still formed within each block and contracted to scalar outputs. To prune Layer2 L2 C128 to Layer2 L0 C128, we remove all inter-block $l > 0$ feature blocks and their associated weights, transfer the remaining $l = 0$ weights and compatible tensor-product paths, and retrain the compact model. All 128 scalar channels are retained, so the target architecture fixes the order-pruning mask. Importance ranking is used when selecting a subset of channels, not when removing all non-scalar inter-block features at unchanged channel width.

Embedding. We investigate the impact of edge embedding dimensions when pruning feature channels. The source model in this ablation has 8 Bessel basis functions, while the standard target

model has 10. The "Small Embedding" variant retains the source basis size. The "Large Embedding" variant uses the target basis size. The Bessel basis has no learnable weights, but its output is the input to the radial MLP in each interaction block. Changing the basis size from 8 to 10 makes the first MLP layer's input dimension incompatible with the source weights. We therefore randomly re-initialize only the first layer of each radial MLP. The remaining compatible weights are transferred through copying or slicing. As shown in Table 10, the Large Embedding variant reduces Energy MAE from 45.3 to 42.7 meV/atom compared with direct inheritance. We adopt this target-compatible initialization in our main experiments.

Table 10: **Ablation of embedding size in channel pruning on MACE-MP.** Both variants prune Layer2 L1 C256 to Layer2 L1 C128 and use 9% of the training data. We evaluate Energy (E, meV/atom), Force (F, meV/Å), and Stress (S, meV/Å^3) errors on the MPtrj. The channels that are pruned here are randomly selected.

Embedding	Params	MAE E	MAE F	MAE S
Small	4,688,400	45.3	68.7	**2.3**
Large	4,688,656	**42.7**	**68.6**	2.4

High body order features. In the MACE architecture, high body-order features are synthesized via a symmetric contraction of atomic basis functions, mathematically realized through Generalized Clebsch-Gordan (GCG) coefficients. Our pruning method extends naturally to this mechanism by associating the importance score $I^{(t)}_{k,l}$ with the output of these contraction steps. When a feature block (k, l) representing a specific many-body interaction is pruned, we physically slice the corresponding dimensions in the CG coupling tensor during the Structural Alignment phase. This effectively removes the computational graph branches responsible for calculating specific higher-order correlations. By pruning these features based on the energy-force sensitivity criterion, the model selectively discards computationally expensive many-body terms.

Residual connections. The presence of residual connections requires strict dimensional consistency between the input and output of a computational block. In MACE, this pathway is introduced by an element-dependent linear projection, formally defined as:

$$\sum_{\tilde{k}} W^{(t)}_{z_i,k\tilde{k}l} h^{(t)}_{i,\tilde{k}lm}, \tag{20}$$

where $W^{(t)}_{z_i,k\tilde{k}l}$ represents the learnable weights specific to the atomic species z_i, mapping input channels $\tilde{k}$ to output channels k for order l. During structural alignment, we slice the input and output channel dimensions according to their respective pruning masks. Furthermore, a unique advantage of this element-wise parameterization is the ability to perform species-specific compression. Since the weights are independent for each element type z_i, parameters corresponding to chemical elements not present in the target downstream dataset can be explicitly removed prior to fine-tuning. This step, combined with channel slicing, results in a highly compact model specialized for both the geometric structure and the chemical composition of the target system.

D.4 Ablation study on importance criterion

We validate the effectiveness of our proposed importance estimation method by pruning a pre-trained MACE-MP model from Layer2 L1 C256 to Layer2 L1 C128 and performing lightweight retraining using 9% of the MPtrj dataset. Table 11 compares our method against three baselines using the same pruning granularity: Random Pruning, Magnitude-based Pruning, and Activation-based Pruning.

Our physics-informed criterion yields the lowest errors across all metrics. This indicates that measuring the sensitivity of the potential energy surface to specific geometric tensors provides a more robust pruning signal than magnitude or activation alone.

Table 11: **Comparison of different importance estimation methods on MACE-MP.** The model is pruned from Layer2 L1 C256 to Layer2 L1 C128 and retrained using 9% of the MPtrj. Metrics include MAE for Energy (E, meV/atom), Force (F, meV/Å), and Stress (S, meV/Å^3) on MPtrj. The best results are in bold.

Method	MAE E	MAE F	MAE S
Random	42.7	68.6	2.4
Magnitude	47.7	68.9	2.5
Activation	36.4	64.4	**2.3**
Our Method	**35.9**	**64.2**	**2.3**

D.5 Validity of first-order importance estimates

First-order estimation at increasing pruning ratios. We evaluate the first-order approximation on 500 MPtrj validation configurations using a frozen MACE-MP checkpoint. We mask complete (k, l) blocks at the tensor-product output of the second interaction block, removing the same fraction of channels within each order $l = 0, 1, 2, 3$. Within each order, we select the lowest-importance blocks using scores computed once in the original model. The same configurations are used for importance estimation, mask selection, and evaluation, without retraining. For each configuration, the actual ΔE is the masked-model energy minus the original-model energy. The one-shot estimate is the signed sum of the first-order contributions of the removed blocks, evaluated in the original model. Table 12 reports mean absolute energy changes per atom and the Pearson correlation between the signed predicted and actual changes across configurations. Estimation accuracy decreases as the pruning ratio increases, but the first-order estimates remain strongly correlated with the actual changes.

Table 12: **First-order estimation at increasing pruning ratios.** Predicted and actual changes are mean absolute ΔE in meV/atom. Pearson correlation uses signed per-configuration changes.

Pruning ratio	First-order estimate	Actual	Pearson
25%	1.947	2.010	0.9968
50%	12.616	14.093	0.9719
75%	37.490	46.625	0.8755

One-shot versus iterative pruning. We also perform iterative pruning in steps of 6.25 percentage points. At each step, we recompute importance scores in the currently masked model, remove the lowest-scoring remaining blocks within each order, and accumulate the stepwise first-order estimates. Table 13 compares the two procedures. Since they can select different blocks, their actual energy changes are reported separately. At higher pruning ratios, iterative estimates are closer to the actual changes, while the actual perturbations are similar to those of one-shot pruning. Considering the additional cost, we use one-shot estimation in our pruning experiments.

Table 13: **One-shot versus iterative pruning.** Values are mean absolute ΔE in meV/atom, measured without retraining.

Pruning ratio	One-shot		Iterative	
	Estimate	Actual	Estimate	Actual
25%	1.947	2.010	2.501	2.579
50%	12.616	14.093	14.636	14.450
75%	37.490	46.625	45.703	46.633

D.6 Effect of calibration-set size

We evaluate how the size of the calibration subset affects importance rankings using the same pre-trained MACE-MP checkpoint. Table 14 compares rankings computed from the 9%, 25%, and 40% MPtrj subsets constructed as described in Appendix D.2, using the 40% subset as the reference. The importance rankings are highly consistent across the tested data fractions, with Spearman correlations of 0.99976 and 0.99997 for the 9% and 25% subsets, respectively.

Table 14: **Effect of calibration-set size on importance rankings.** Spearman correlations are computed relative to the rankings obtained from the 40% MPtrj subset.

Data fraction	Configurations	Spearman correlation
9%	145,923	0.99976
25%	394,397	0.99997
40%	620,741	1.00000

D.7 Ablation study on retraining

To validate the necessity of the Lightweight Retraining stage, we conduct an ablation study on the AgAu dataset. We compared three strategies: training the compact architecture from scratch with random initialization, directly fine-tuning the pruned model without retraining, and our full pipeline, which includes lightweight retraining on 25% of the pre-training data.

As shown in Table 15, training From Scratch yields poor convergence (RMSE E: 369.1 meV/atom), demonstrating the value of initializing from a pre-trained foundation model. Furthermore, skipping the retraining phase (Pruning – Finetuning) results in higher errors compared to our proposed method. By incorporating Lightweight Retraining, the model effectively recovers from pruning-induced approximation errors, significantly reducing the Energy RMSE from 16.9 to 6.6 meV/atom and Force RMSE from 18.4 to 15.1 meV/Å. This confirms that a brief adaptation phase is crucial for maximizing the performance of the compact model.

Table 15: **Ablation study of retraining.** We evaluate RMSE for Energy (E, meV/atom) and Force (F, meV/Å) on the AgAu dataset.

Method	RMSE E	RMSE F
From Scratch	369.1	34.5
Pruning – Finetuning	16.9	18.4
Pruning (25%) – Finetuning	**6.6**	**15.1**

D.8 Retraining hyperparameters

We adopt the official training configurations provided by MACE for the retraining process. To ensure reproducibility, the detailed hyperparameters are listed in Table 16.

The from-scratch baselines and the corresponding pruned models use the same training hyperparameters and epoch schedules: 200 epochs for MACE-MP and 190 epochs for MACE-OFF, without additional pruning-specific tuning. The MACE-MP baselines show small changes in error near the end of training. Over the final 25 epochs, Energy and Force MAE change by 2.0% and 1.3%, respectively, for the from-scratch Layer2 L0 C128 model, and by 1.5% and 1.1% for the from-scratch Layer2 L1 C128 model. These small late-stage changes support convergence of the two baselines.

Table 16: **Retraining hyperparameters.**

Hyperparameter	MACE-MP	MACE-OFF
Epochs	200	190
Loss	universal	ef
E weight	1	40
F weight	10	1000
S weight	100	/
Learning rate	0.005	0.01
Weight decay	1e-8	5e-10
Patience	5	20
EMA	0.995	0.99
Gradient clipping	100	1

D.9 Feature importance stability after retraining

A natural concern is whether the importance distribution across (k, l) blocks shifts significantly after retraining, which could render the initial pruning criterion suboptimal in hindsight. To address this, we analyze the evolution of importance scores for the 128 retained channels in a model pruned from Layer2 L1 C256 to Layer2 L1 C128, comparing scores computed immediately after pruning against those computed after retraining.

Figure 5 (Appendix B.4) shows the overall importance distribution across orders L before pruning. To quantify stability after retraining, we compute the Spearman rank correlation between the pre- and post-retraining importance scores across all 128 retained channels, obtaining a value of **0.81**. This strong positive correlation indicates that the global relative ranking of feature importance is largely preserved. Furthermore, we examine the overlap of the top-16 channels (top 12.5%) before and after retraining: **14 out of 16 channels** (87.5%) remain identical. This demonstrates that the core structural pathways identified by our gradient-activation criterion remain the most heavily utilized by the retrained model, validating the reliability of the one-shot pruning criterion.

D.10 End-to-end pipeline cost

Table 17 compares GPU hours and wall-clock time for our structural pruning pipeline and training a compact foundation model from scratch on NVIDIA A800 GPUs. Pre-training and retraining use four GPUs, while calibration, pruning, and fine-tuning use one GPU.

Table 17: **End-to-end training cost comparison on NVIDIA A800 GPUs.** Our pruning pipeline reduces total GPU hours by $\sim 4\times$ relative to pre-training the same compact architecture from scratch.

Strategy	Stage	GPU Hours	Wall-Clock (h)
From Scratch	Pre-training	750	188
	Fine-tuning	35	35
	Total	**785**	**223**
Pruning (Ours)	Calibration & Pruning	1	1
	Retraining	180	45
	Fine-tuning	35	35
	Total	**216**	**81**

The pruning-retraining phase requires only 181 GPU hours (1 + 180), compared to 750 GPU hours for pre-training the same compact architecture from scratch, directly validating our claimed $\sim 4\times$ reduction in training cost. The fine-tuning cost is identical in both pipelines. In total, our pipeline reduces end-to-end GPU hours from 785 to 216 and wall-clock time from 223 hours to 81 hours.

Table 18 reports the training cost of each MACE-MP architecture. All runs use 200 epochs. The from-scratch models use the full MPtrj dataset, while the pruned L1 and L0 models use 40% and 25%, respectively. The L2 model is the source foundation model and has no pruned counterpart in

this comparison. Its pre-training cost is shown separately from the cost of training the compact target models.

Table 18: **MACE-MP training cost by architecture.** Values are approximate training GPU-hours, with the fraction of MPtrj in parentheses. Calibration and pruning are excluded.

Architecture	From scratch GPU-hours (data)	Pruned GPU-hours (data)
Layer2 L2 C128	∼5,500 (100%)	N/A
Layer2 L1 C128	∼1,800 (100%)	∼650 (40%)
Layer2 L0 C128	∼750 (100%)	∼180 (25%)

For each compact target architecture, the from-scratch and pruned models use the same hyperparameters and epoch count, giving the same training schedule per sample. Total training compute is lower for the pruned models because they use less data. Thus, their accuracy gains do not result from receiving more target-model training compute. For the L0 target, calibration and pruning, including importance computation, add approximately one GPU-hour to the 180 GPU-hours of retraining, giving the 181 GPU-hours reported above.

The nominal 2.5×–4× MACE-MP training-cost reduction follows from the data fractions at a fixed target architecture and epoch count: $100/40 = 2.5$ and $100/25 = 4$. The measured L0 cost ratio, including calibration and pruning, is $750/181 \approx 4.1$. Parameter reductions use the source-to-target parameter ratios in Table 2. The endpoints are $5{,}725{,}072/3{,}847{,}696 \approx 1.49$ and $15{,}847{,}440/3{,}847{,}696 \approx 4.12$, reported as approximately 1.5×–4×.

D.11 Throughput and memory usage

To validate the efficiency gains achieved through our structural pruning approach, we evaluate the inference throughput and memory consumption of models with varying sizes on an NVIDIA A800 GPU. We conduct two sets of experiments: one with a batch size of 1 to measure latency, and another to maximize the batch size to evaluate peak throughput. The experiments are conducted using the MPtrj dataset, sampling one configuration from each of the approximately 150,000 unique structures. For the maximum batch size tests, we utilize cuEquivariance [72] v0.8.1, which accelerates computation and reduces memory footprint in such memory-bound scenarios.

Table 19 presents the performance comparison. The results demonstrate that structural pruning significantly reduces computational costs. Compared to the baseline, the pruned models achieve a memory reduction of up to 5.7× and a throughput improvement of up to 2.7×.

Both reported gains compare Layer2 L2 C128 with Layer2 L0 C128. The throughput ratio is $38{,}043/14{,}031 \approx 2.7$ at each model's maximum batch size. The memory ratio is $10.3/1.8 \approx 5.7$ at batch size 1.

Table 19: **Throughput and memory usage comparison of models with different sizes.** We evaluate performance in two regimes: single-sample inference (Batch Size = 1) to measure latency, and maximum batch size to measure peak throughput.

Model	Batch Size = 1		Max Batch Size		
	Throughput (atom/s)	Memory (GB)	Batch Size	Throughput (atom/s)	Memory (GB)
Layer2 L2 C128	1,204	10.3	64	14,031	58.6
Layer2 L1 C128	1,532	4.7	128	23,879	62.4
Layer2 L0 C128	2,017	1.8	256	38,043	50.1

D.12 MD simulation validation

While energy and force MAE are standard surrogates for MLIP quality, the ultimate test of a potential is its behavior in actual MD simulations, where small force errors can accumulate over long

trajectories. To validate that our pruned models produce stable and physically accurate dynamics, we conduct MD simulations on liquid water at 300 K using the pruned MACE-MP model (25% data), fine-tuned on the H2O dataset, and compare against the full pre-trained model.

We run two types of simulations: (1) a **10 ps NVE** simulation to assess energy conservation (energy drift), and (2) a **20 ps NVT** simulation to evaluate structural properties, specifically the radial distribution functions (RDFs). Results are summarized in Table 20.

Table 20: **MD simulation validation of pruned vs. full model on liquid water at 300 K.** RDF peak positions are reported for O–O, O–H, and H–H pairs. The pruned model maintains near-identical energy conservation and structural properties.

Model	**NVE Energy Drift**	**RDF 1st / 2nd Peak (Å)**		
	(meV/atom/ps) ↓	**O–O**	**O–H**	**H–H**
Pre-trained (100%)	< 0.01	2.72 / 4.42	0.98 / 1.75	1.56 / 2.29
Pruned (25%)	< 0.01	2.72 / 4.52	0.98 / 1.77	1.56 / 2.31

Both models achieve near-perfect energy conservation (< 0.01 meV/atom/ps), confirming a smooth potential energy surface. The pruned model faithfully reproduces the first-shell RDF peak positions for all three pair types (Figure 6), indicating that the local liquid structure is well preserved. These results confirm that structural pruning does not destabilize MD trajectories and produces dynamics that are physically consistent with the full model.

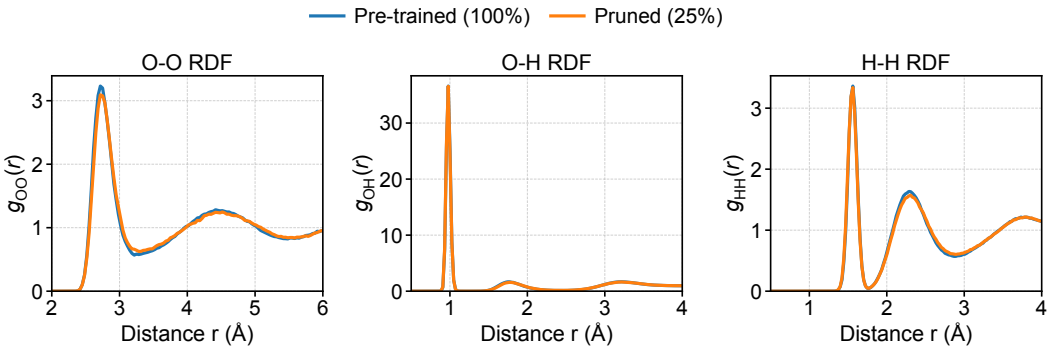

Figure 6: **Radial distribution functions from NVT MD simulation of liquid water at 300 K.** O–O, O–H, and H–H RDFs are shown for the pre-trained model (100%, blue) and the pruned model (25%, orange), both fine-tuned on H2O. Percentages indicate the fraction of MPtrj used for pre-training or recovery training. The two curves overlap closely, confirming that structural pruning preserves the accuracy of local liquid-state structure.

D.13 MACE-MP downstream fine-tuning

Table 21 reports the full RMSE values corresponding to Figure 4. Percentages indicate the fraction of MPtrj used before downstream fine-tuning.

Table 21: **Effectiveness of pruned MACE-MP model as initialization for downstream fine-tuning.** RMSE for Energy (E, meV/atom) and Force (F, meV/Å) across 8 inorganic datasets. Bold: best per column.

Init.		SSE-PBE	H2O-PD	AgAu	AlMgCu	Cu	Ti	V	W
From Scratch	E	1.8	79.9	369.1	7.7	38.8	8.3	14.2	15.6
	F	29.9	**29.7**	34.5	42.9	13.6	94.2	140.4	181.2
Pre-trained (100%)	E	0.6	1.1	15.4	3.4	**0.7**	6.7	**5.4**	**6.2**
	F	29.1	36.8	15.8	12.2	7.4	91.4	83.5	104.1
Pruned (25%)	E	**0.5**	**0.9**	**6.6**	**3.2**	**0.7**	**6.5**	6.1	6.8
	F	**25.9**	31.6	**15.1**	**11.6**	**6.9**	**90.6**	**82.3**	**101.1**

For energy and forces separately, we calculate the reported downstream error reduction as an equal-weight average over these eight datasets: $\frac{1}{8}\sum_{d=1}^{8}(1 - e_d^{\text{pruned}}/e_d^{\text{scratch}}) \times 100\%$. Here, e_d^{pruned} and e_d^{scratch} are the RMSE values for the Pruned (25%) and From Scratch rows in Table 21. This gives reductions of 70.1% for energy and 34.4% for forces.

D.14 Organic molecule benchmarks

We evaluate the pruned MACE-OFF model on three organic benchmarks covering downstream fine-tuning and ultra-low data regimes.

Downstream fine-tuning on 3BPA and AcAc. Tables 22 and 23 report RMSE for energy and forces on 3BPA [56] (a flexible drug-like molecule tested across temperature regimes and dihedral configurations) and AcAc [57]. Using only 50% of the SPICE pre-training data, the pruned model matches or exceeds the from-scratch baseline in most conditions, confirming that compact specialized models retain the expressivity needed for complex organic potential energy surfaces.

Table 22: **Effectiveness of pruned MACE-OFF model as initialization for 3BPA fine-tuning.** We evaluate RMSE for Energy (E, meV) and Force (F, meV/Å). The training set is collected at 300 K. Standard deviations are computed over three runs and shown in parentheses. The best results are in bold.

Dataset		Initialization		
		From Scratch	Pre-trained (100%)	**Pruned (50%)**
300 K	E	**3.0 (0.2)**	3.3 (0.03)	**3.0 (0.08)**
	F	8.8 (0.3)	**7.8 (0.01)**	**7.8 (0.07)**
600 K	E	9.7 (0.5)	7.3 (0.04)	**7.1 (0.12)**
	F	21.8 (0.6)	**16.6 (0.05)**	16.7 (0.13)
1200 K	E	29.8 (1.0)	**20.3 (0.17)**	21.7 (0.96)
	F	62.0 (0.7)	**48.7 (0.56)**	**48.7 (0.7)**
Dihedral Slices	E	7.8 (0.6)	**7.3 (0.28)**	7.6 (0.25)
	F	16.5 (1.7)	12.3 (0.10)	**11.8 (0.11)**

rMD17. To further evaluate data efficiency in the ultra-low data regime, we benchmark on rMD17 [58] following [19], training each model on only 50 molecular conformations. Table 24 compares our pruned MACE-OFF model (retraining using 50% data) against ACE [56], NequIP [23], PACE [73], and a MACE model trained from scratch. The pruned model consistently outperforms from-scratch baselines, indicating that structural pruning retains robust feature representations that generalize to extremely limited fine-tuning data.

Table 23: **Effectiveness of pruned MACE-OFF model as initialization for AcAc fine-tuning.** We evaluate RMSE for Energy (E, meV) and Force (F, meV/Å). The training set is collected at 300 K. Standard deviations are computed over three runs and shown in parentheses. The best results are in bold.

Dataset		Initialization		
		From Scratch	Pre-trained (100%)	**Pruned (50%)**
300 K	E	**0.9 (0.03)**	1.0 (0.02)	**0.9 (0.02)**
	F	**5.1 (0.10)**	**5.1 (0.07)**	5.3 (0.07)
600 K	E	**4.6 (0.3)**	5.8 (0.28)	5.6 (0.02)
	F	22.4 (0.9)	**16.4 (0.70)**	**16.4 (0.22)**

Table 24: **MAE of energy (E, meV) and force (F, meV/Å) on the rMD17 dataset.** Each model is trained using only 50 molecular conformations per molecule. The best results are in bold.

Dataset		From Scratch				Pruned (50%)
		ACE	NequIP	PACE	MACE	MACE
Aspirin	E	26.2	19.5	15.7	17.0	**10.8**
	F	63.8	52.0	37.4	43.9	**26.4**
Azobenzene	E	9.0	6.0	6.7	**5.4**	5.5
	F	28.8	20.0	17.5	17.7	**16.6**
Benzene	E	**0.2**	0.6	0.6	0.7	0.6
	F	**2.7**	2.9	3.3	**2.7**	2.9
Ethanol	E	8.6	8.7	6.3	6.7	**3.0**
	F	43.0	40.2	25.4	32.6	**15.7**
Malonaldehyde	E	12.8	12.7	11.5	10.0	**6.7**
	F	63.5	52.5	57.3	43.3	**25.2**
Naphthalene	E	3.8	**2.1**	**2.1**	**2.1**	**2.1**
	F	19.7	10.0	9.7	9.2	**8.9**
Paracetamol	E	13.6	14.3	10.1	9.7	**6.7**
	F	45.7	39.7	29.3	31.5	**22.1**
Salicylic acid	E	8.9	8.0	7.0	6.5	**4.5**
	F	41.7	35.0	29.2	28.4	**19.5**
Toluene	E	5.3	3.3	2.7	3.1	**2.1**
	F	27.1	15.1	12.0	12.1	**9.6**
Uracil	E	6.5	7.3	5.9	4.4	**3.1**
	F	36.2	40.1	26.8	25.9	**16.0**

D.15 Other pre-training datasets

To verify the transferability of our method to different pre-training data, we conduct an additional experiment using the MACE-OMAT model. The base model was pre-trained on the large-scale OMat24 [36] dataset (containing over 100 million structures). We apply our pruning method to this pre-trained model, reducing its equivariant features from $L = 1$ to $L = 0$, and subsequently retrain it on the official OMat-1M subset.

Table 25 compares the pruned model against a baseline trained from scratch on OMat-1M with the same architecture. The results show that the pruned model achieves lower errors than the from-scratch baseline, with a 13.0% reduction in Energy MAE and a 7.1% reduction in Force MAE. This observation suggests that our pruning approach effectively preserves valuable structural priors. These

preserved features serve as a superior initialization compared to random weights, validating the potential of our method in other datasets.

Table 25: **Data efficiency of pruning methods on MACE-OMAT.** Both models use Layer2 L0 C128 with 8,222,244 parameters and are trained on OMat-1M. L pruning uses Layer2 L1 C128 as the source model. We report Energy (E, meV/atom) and Force (F, meV/Å) MAE on OMat-1M, with relative changes from the from-scratch baseline in parentheses and improvements in bold.

Method	MAE E	MAE F
From Scratch	26.9	106.6
L Pruning	**23.4 (-13.0%)**	**99.0 (-7.1%)**

E Applicability of the proposed method

E.1 SevenNet architecture

To verify the transferability of our method, we conduct experiments on the SevenNet [3] foundation model, which is based on the NequIP [23] architecture. We prune the pre-trained SevenNet-l3i5 model to a compact model with the same layer count and channel configuration as the official SevenNet-0 model ($L_{\max} = 2$, five interaction layers). Here, SevenNet-0 is the official model name, and the suffix "0" does not denote the maximum tensor order. We denote the target architecture as Layer5 L2 C128 in Table 26. The following describes our pruning strategy for the non-linear layers involved in the NequIP architecture and the results.

Non-linear activation. NequIP employs a specific equivariant non-linearity strategy to process geometric tensors. While scalar features ($l = 0$) are processed via standard activation functions, higher-order tensors ($l > 0$) rely on a Gated Non-Linearity to preserve equivariance. This mechanism involves generating auxiliary scalar values that act as multiplicative gates for the higher-order features. Consequently, this introduces a strict structural dependency: a higher-order feature block cannot exist without its corresponding scalar gate.

We address this by enforcing a coupling constraint during the mask generation phase. The pruning decision for the gate scalars is not determined by their individual importance scores but is derived deterministically from the masks of the higher-order tensors. Specifically, if a feature block (k, l) with $l > 0$ is retained, we strictly enforce the retention of its associated scalar gate channel in the preceding layer.

Results. We evaluate our approach by pruning the pre-trained SevenNet-l3i5 model (configured with 5 layers and a maximum order $L_{\text{max}} = 3$) down to a target architecture of $L_{\text{max}} = 2$. Table 26 compares our pruning method against a baseline model with the identical architecture ($L_{\text{max}} = 2$) trained from scratch.

The results demonstrate significant data efficiency. The model trained from scratch requires 100% of the dataset to achieve a Force MAE of 40 and an sAlex MAE E of 59.1. In contrast, our Pruning method achieves a better Force MAE of 39 and a lower sAlex MAE E of 58.9 while utilizing only 25% of the training data. With the full dataset, the pruned model further improves to a Force MAE of 35 and an sAlex MAE E of 53.6, outperforming the from-scratch baseline across all metrics. This indicates that our method effectively transfers structural knowledge from the higher-order foundation model to the compressed model, reducing the data required for convergence compared to standard training.

E.2 eSCN architecture

To further validate the generalizability of our method beyond MACE and NequIP-based architectures, we conduct experiments on the eSCN (Equivariant Spherical Channel Network) [10] foundation model. Like MACE and SevenNet, eSCN constructs geometric tensors via Clebsch-Gordan tensor products and operates on coupled $(2l + 1)$-dimensional irreducible representations. This shared

Table 26: **Data efficiency of pruning methods on SevenNet.** The table compares the performance of pruned models against baselines trained from scratch. Metrics include MAE for Energy (E, meV/atom), Force (F, meV/Å), and Stress (S, meV/Å^3) on MPtrj, and Energy (E, meV/atom) on the sAlex subset. Data indicates the fraction of training data used.

Method	Base Model	Pruned Model	Params	Data	MPtrj			sAlex
					MAE E	MAE F	MAE S	MAE E
From Scratch	Layer5 L3 C128	/	1,171,144	100%	/	/	/	51.2
	Layer5 L2 C128		842,440		11	40	1.7	59.1
L Pruning	Layer5 L3 C128	Layer5 L2 C128	842,440	**25%**	11	**39**	2.1	**58.9**
				100%	**9**	**35**	**1.7**	**53.6**

Table 27: **Data efficiency of pruning methods on eSCN.** All models use the target architecture eSCN-L4-M2-Lay12. Metrics are Force MAE (meV/Å, lower is better) and EFwT (%, higher is better) on the OC20 validation set (ID split). Bold values indicate that the pruned model outperforms the from-scratch baseline.

Method	Data	Force MAE (meV/Å) ↓	EFwT (%) ↑
From Scratch	200k	33.1	0.20
L Pruning	200k	**30.3**	**0.24**
From Scratch	2M	19.1	2.55
L Pruning	1M	**18.4**	**2.86**
L Pruning	2M	**17.5**	**3.36**

mathematical basis means our (k, l)-block pruning principle applies to eSCN without any substantive modification.

Setup. We prune a pre-trained eSCN-L6-M2-Lay12 ($L_{\max} = 6$) model to the target architecture eSCN-L4-M2-Lay12 ($L_{\max} = 4$). We evaluate data efficiency by training both a randomly initialized model and the pruned model on the OC20 200k and 2M subsets for the same number of epochs. All evaluated models share the same target architecture (eSCN-L4-M2-Lay12).

Results. Table 27 reports Force MAE and the Energy-Force within Threshold (EFwT) metric on the OC20 validation set (ID split). The pruned model consistently outperforms the from-scratch baseline at every data scale: on 200k samples, it reduces Force MAE from 0.0331 to 0.0303 eV/Å (+20% relative EFwT), and on 2M samples from 0.0191 to 0.0175 eV/Å. Notably, the pruned model trained on only 1M samples (50% of 2M) achieves a Force MAE of 0.0184 eV/Å, which is comparable to the from-scratch model trained on the full 2M dataset, demonstrating a 2× reduction in training data requirements.

Inference efficiency. We compare the source eSCN-L6-M2-Lay12 model with the pruned eSCN-L4-M2-Lay12 checkpoint retrained on OC20-2M. Both models are evaluated in FP32 on an NVIDIA A100 80GB GPU using the same batch of 24 configurations from the OC20 validation ID split. Inference includes graph construction and energy and force prediction. Table 28 reports latency and peak allocated GPU memory. Pruning provides a 1.55× inference speedup and reduces peak memory by 36.4%. These results show that structural pruning can provide additional efficiency gains for eSCN, which already uses efficient SO(2)-based convolutions.

Table 28: **Inference efficiency of eSCN on OC20.** Measurements use an A100 GPU and batch size 24.

Model	Parameters (M)	Latency (ms)	Peak memory (GiB)
eSCN-L6-M2-Lay12	51.84	991.8	4.89
eSCN-L4-M2-Lay12 (pruned)	36.11	641.7	3.11

F Licenses for datasets and models

Table 29 provides license information for all datasets and pre-trained models used in this work. All assets are used in accordance with their respective licenses.

Table 29: **Licenses for datasets and models used in this work.**

Asset	License	Reference
Pre-training Datasets		
MPtrj	MIT	[34]
SPICE	CC0	[35]
OC20	CC BY 4.0	[62]
OMat24	CC BY 4.0	[36]
Downstream Datasets (Inorganic)		
SSE, H2O, AgAu, AlMgCu, Cu, Ti, V, W	LGPL-3.0	[64–71]
sAlex	CC BY 4.0	[59, 36]
Matbench Discovery	CC BY 4.0	[61]
Downstream Datasets (Organic)		
3BPA, AcAc	CC BY 4.0	[56, 57]
rMD17	CC0	[58]
Pre-trained Models		
MACE-MP-0	MIT	[2]
MACE-OFF	ASL License	[55]
MACE-OMAT	ASL License	[2]
SevenNet	GPLv3	[3]
eSCN	CC BY 4.0	[10]

G Broader impacts

This work reduces the computational cost of atomistic foundation models, lowering barriers to computational materials discovery and drug design, which benefits researchers with limited resources and reduces energy consumption. There is no direct path to negative societal impacts: atomistic simulation tools are domain-specific scientific instruments not applicable to surveillance, disinformation, or fairness-sensitive decision making, and our compression method does not change the underlying capabilities or introduce new misuse risks beyond those already inherent to molecular simulation.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state all key contributions: the pruning framework, 1.5×–4× parameter reduction, 2.5×–4× training cost reduction, 2.7× inference speedup, 70.1%/34.4% downstream error reductions, and composability with quantization and knowledge distillation. These claims are supported by experimental results in Section 4 (Accuracy, Efficiency, Application, and Combination subsections).

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The conclusion discusses a key limitation: the target pruned architecture must be specified in advance, which limits flexibility. Neural architecture search is identified as a future direction to address this.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Proposition B.2 (SO(3) equivariance preservation under block-wise pruning), Proposition B.3 (SO(3) invariance of the importance criterion), and Proposition B.4 (spectral interpretation of feature order) are provided with complete proofs in Appendix B.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Appendix D provides full experimental details including dataset descriptions, calibration procedure, pruning configurations, retraining hyperparameters, and hardware specifications. Code is available at `https://github.com/HPC-AI-Team/SO3-Pruning`.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code is available at `https://github.com/HPC-AI-Team/SO3-Pruning`. All datasets used are publicly available and properly cited.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4.1 describes the foundation models, datasets, and downstream tasks. Appendix D describes calibration, pruning configurations, and MACE retraining settings. Appendix E provides the SevenNet and eSCN experimental setups.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Standard deviations over three independent runs are reported for the downstream fine-tuning experiments on 3BPA and AcAc (Tables in Appendix D). For the large-scale pre-training experiments (MACE-MP, MACE-OFF, SevenNet, eSCN), error bars are

not reported as multiple runs are computationally prohibitive (hundreds of GPU-hours per run). Results are instead validated across multiple datasets and architectures.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix D.10 reports MACE-MP training costs and GPU configurations. Appendices D.11 and E.2 describe the hardware and measurement settings for the MACE-MP and eSCN inference benchmarks, respectively.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This work involves purely computational methods for compressing atomistic foundation models. It uses only publicly available datasets and open-source model checkpoints. The research conforms to the NeurIPS Code of Ethics in all respects.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Broader impacts are discussed in Appendix G. The work has positive societal impact by reducing computational costs and enabling more accessible materials discovery. There is no direct path to negative societal impacts, as atomistic simulation tools are domain-specific scientific instruments not applicable to misuse scenarios such as surveillance or disinformation.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Atomistic foundation models for materials simulation pose no high risk for misuse. No special safeguards are required beyond standard open-source release practices.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All datasets and pre-trained models are properly cited with their original papers. Specific license information for each dataset and model is provided in Appendix F.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The pruning code is available at `https://github.com/HPC-AI-Team/SO3-Pruning` with documentation covering installation, usage, and reproduction of the main experiments.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve human subjects research and therefore requires no IRB approval.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not used as any component of the core methodology. No declaration is required.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.