
MLIP Embeddings Predict Extraordinary Material Properties

Luis Carretero López^{1,2}, Nofit Segal¹, Sathya Edamadaka¹, and Rafael Gómez-Bombarelli¹

¹Materials Science and Engineering, MIT

²Computer Science, ETH Zurich

{luisc440, nofit, sathya, rafagb}@mit.edu

Abstract

Extraordinary materials have properties beyond the range of typical training datasets, so discovering them requires predictors that can extrapolate past the labels they were trained on. Here, we test whether the representations learned by machine learning interatomic potentials (MLIPs) support this out-of-support (OOS) prediction. On seven crystal properties, we hold out the largest 5% of labels and compare a pretrained MLIP, probed or finetuned, against training from scratch and against two classical descriptors under the same predictor. Physics-based descriptors are the conventional choice for extrapolation and led earlier tests, yet we find that MLIP embeddings lead on OOS error on every task. Probed, they correctly identify a median 55% of the held-out materials, against 8% from scratch and 23% for the better descriptor, and their error falls faster with data. Finetuning aligns the property along the first principal component of the embeddings, and the pretraining prior extends that axis only a finite distance past the training range: within it the held-out materials stay ordered, beyond it they scatter into the training mass.

1 Introduction

Much of materials science is a search for extraordinary materials whose properties take extreme values [1].

High-throughput screening and AI-guided generative design both rely on learned property predictors to identify promising candidates [2–6]. The extraordinary candidates often lie outside the label range these predictors were trained on, so discovery depends on the ability to predict beyond it. Labels are also scarce, since each requires an expensive density-functional theory calculation or experimental measurement.

Pretraining is the standard answer to label scarcity, and in molecular property prediction representations learned from large unlabeled corpora already lead the MoleculeNet benchmarks [7, 8]. Inorganic crystals now have pretraining at a comparable scale: machine learning interatomic potentials (MLIPs, or foundation potentials) learn energies and forces from hundreds of millions of density-functional calculations spanning the periodic table [9, 10].

In our setting, the test labels are proxies for extraordinary materials, as these properties lie above the support of the training labels. Regressing these is a task we call out-of-support (OOS) prediction [11, 12]. We compare MLIP embeddings on seven crystal properties, by probing raw embeddings or finetuning, against training from scratch and classical descriptors, and find that the MLIP representation extrapolates best and is the most label-efficient. We then analyze the representations via PCA and ask whether transductive OOS methods, previously successful on composition descriptors [12], still help on the MLIP embeddings.

Related work. The learned representations of MLIPs also transfer to different properties [13–16], but those results use random splits [17] or hold out compositions inside the training label range [18]. A study of molecular and crystal property prediction on out-of-support labels shows that most of the pretraining gain in error vanishes out of distribution [19]. Beyond that range the evidence is ambiguous: scaling training labels is marginal or even adverse out of distribution [20], and linear models on physics-based descriptors perform best where a target range of molecular properties is held out [21].

2 Method

Task. We adopt the extrapolation setup of Segal et al. [12]. Each dataset $\mathcal{D} = \{(x_i, y_i)\}$ pairs structures x_i with scalar property labels y_i , and a task splits it by the label: the largest 5% of property values (y larger than threshold τ) form the OOS test set $\mathcal{D}_{\text{test}}^{\text{oos}}$ (Fig. 1b). The remainder divides into a training set $\mathcal{D}_{\text{tr}}$ (90% of the dataset) and two disjoint hold-outs of 2.5% each, $\mathcal{D}_{\text{val}}^{\text{id}}$ for model selection and $\mathcal{D}_{\text{test}}^{\text{id}}$ for in-distribution (ID) evaluation. We measure each task’s extrapolation distance as the gap between the mean OOS and ID test labels, in units of the training labels’ mean absolute deviation (MAD, App. A.1).

Data. We use seven property datasets, chosen to maximize target diversity. Band gap, bulk modulus, refractive index, and phonon peak come from Matbench [22]. The other three are Debye temperature from AFLOW [2, 23], thermal conductivity from the Phonix database [24], and critical temperature from the superconductor set of Cerqueira et al. [25] (thresholds and label distributions for all seven in App. A.1).

Model and training methods. An MLIP model composes a backbone $f_\theta : x \mapsto \{h_i\}$, which maps a structure to one embedding per atom, with a predictor $r_\psi : \{h_i\} \mapsto \hat{y}$, a shallow MLP around a mean pool over atoms that reads out the property (App. A.5). We use Orb-v3 (direct-inf) [26], pretrained on energies, forces, and stresses from the MPtrj and Alexandria datasets, and take $\{h_i\}$ from its final message-passing layer (App. A.3 and A.5). Training from scratch randomly initializes θ and ψ . The two transfer methods start from the pretrained backbone, where finetuning trains both end to end and probing freezes θ and trains ψ alone on the raw embeddings (App. Fig. 4). The MLIP’s pretraining corpus holds almost every crystal we score but none of our property labels, so only the label lies outside its support (App. A.2).

Descriptor baselines. We compare against two classical structure-based descriptors, the conventional route to extrapolation, from opposite ends of the design space. The smooth overlap of atomic positions (SOAP) [27] expands the local atomic density and carries geometry alone, while classical force-field inspired descriptors (CFID) [28] form a fixed global vector that carries chemistry as well. A fixed descriptor cannot be finetuned, so each is followed with a predictor matched in depth and width to the MLIP probe’s readout (App. A.4 and A.5).

Training and metrics. All models are trained with an L1 loss using Adam with a warmup-cosine schedule, fixed training budgets, and three seeds per reported configuration. The learning rate is the only swept hyperparameter, selected on $\mathcal{D}_{\text{val}}^{\text{id}}$. We report MAE in each task’s label units on both test sets, and OOS recovery, the share of the OOS test set predicted above τ . Metric definitions, learning-rate grids, budgets, and checkpoint selection are given in Appendices A.7 and A.8. Every swept choice is selected in distribution, which against an OOS oracle [29] raises median OOS error by 1.7% but lowers recovery by up to 42 percentage points on single tasks (App. C.4).

3 Results

3.1 Extrapolation and OOS recovery

At the full training set the MLIP probe recovers a median 55% of the OOS set, against 37% finetuned and 8% from scratch (Fig. 1a). Probed with the same readout, raw MLIP embeddings recover more of the OOS set than either classical descriptor on six of the seven tasks, at $2.4\times$ the better descriptor’s median. Phonon peak is the exception, where SOAP recovers slightly more while making

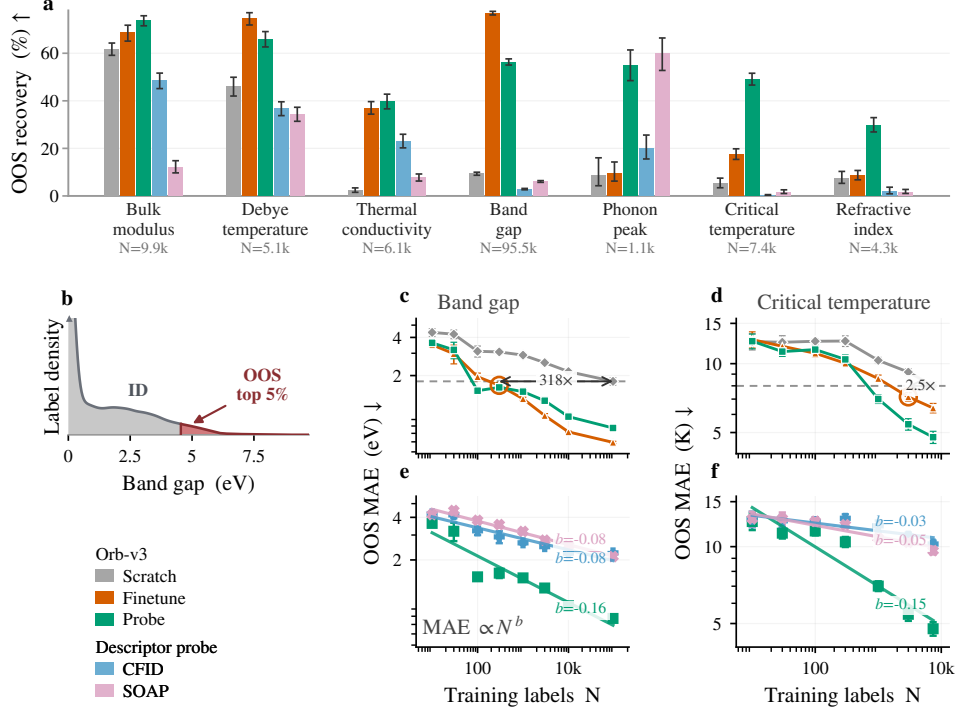


Figure 1: Pretrained MLIP embeddings recover more of the OOS set than any baseline, and their error falls faster with data. **a**, OOS recovery at the full training set, tasks ordered by extrapolation distance. **b**, the label split on band gap, largest 5% held out. **c–d**, OOS MAE against training-set size for the three GNN training methods, with the label saving marked. **e–f**, the same for the three probed representations, with power-law fits and their exponents b . Error bars: ± 1 s.e., Wilson in **a**.

$2.2\times$ the embeddings’ OOS error, so SOAP crosses τ by overprediction (App. B.1). The MLIP probe’s false-positive rate on $\mathcal{D}_{\text{test}}^{\text{id}}$ is a median 0.4% (max 3.2%), so its recovery does not come from overprediction (precision at 5% prevalence in App. Table 4).

Which transfer method leads depends on the extrapolation distance (Section 2). Past an extrapolation distance of about 6 MAD the probe recovers $2.8\times$ to $5.8\times$ what finetuning does, while below it neither method leads consistently (the same threshold divides the tasks on OOS error). On the long-tailed tasks finetuning fits the bulk of the training labels and mis-orders the OOS set, whereas probing keeps the ordering the prior induces (Section 3.2).

Pretraining lowers OOS error and the label cost. Both transfer methods beat training from scratch on every task, the better of the two cutting OOS error by 33% at the median (Fig. 1c–d, App. Table 5; also in distribution, App. B.1). The advantage holds from a few tens of labels to the full training set, and the gain is largest on band gap (67%), which has no direct link to the pretraining objective. Finetuning reaches the best from-scratch OOS error with $3.3\times$ fewer labels, $318\times$ on band gap (App. B.2), and the probe leads or matches finetuning up to a hundred labels on every task.

Raw MLIP embeddings beat both descriptors, and scale better. Among the three probed representations, the MLIP embeddings beat both descriptors on OOS error on every task at the full training set (Fig. 1e–f), by 35% at the median against the more accurate descriptor and by as much as 64% on band gap (App. Table 5; in distribution by 38% at the median, App. B.1). That lead holds at every predictor size (App. C.5). The MLIP embeddings also lead at every smaller training-set size N , and their probed error falls faster with N , on every task and test set (App. B.2), so they hold more extractable information than a fixed descriptor.

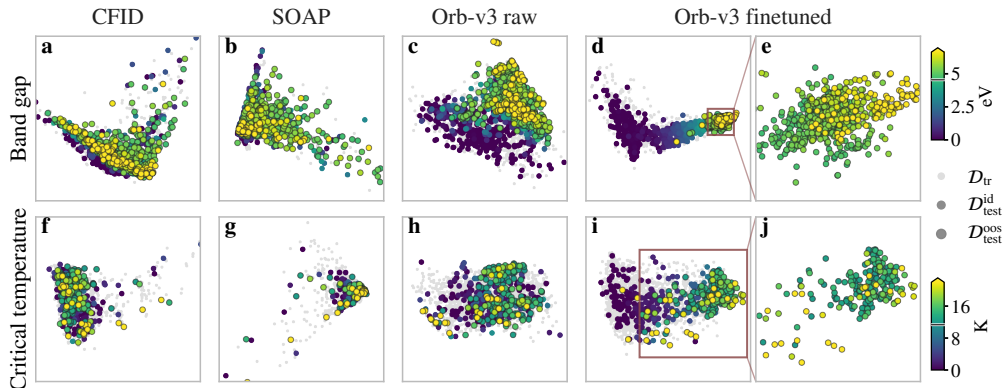


Figure 2: In the raw MLIP embeddings the property varies along the first principal component, more than in either descriptor, and finetuning aligns it almost fully with that component. PCA fitted on the training set, test points colored by label. **e, j**, the boxed region of the finetuned panel, OOS set alone.

3.2 Embedding space geometry

We map every representation by PCA, fitted on the training set, and measure how well its first principal component orders the materials by the property (Fig. 2; all tasks in App. B.3). In the raw MLIP embeddings the property already varies along the first principal component (Fig. 2c, h). Its rank correlation with the label on $\mathcal{D}_{\text{test}}^{\text{id}}$ is 0.54 at the median, against 0.22 on CFID and 0.20 on SOAP, so the pretraining objective alone induces an ordering by the property. Finetuning aligns the property with that component almost completely, above 0.90 on every task (Fig. 2d, i)—an expected consequence of supervised training, which collapses features onto the direction of the label [30, 31]. The alignment is nearly affine, with a linear probe reaching $6.9\times$ lower ID error than the best constant predictor at the median, against at most $3.3\times$ on the other representations (App. B.4). Finetuning’s two-layer readout removes only 8% of the ID error beyond that linear map, so the ordering along that component is nearly the ordering of the predictions.

Under finetuning no supervision signal reaches the OOS set, so where it lies along that component is set by the prior (from scratch, the same labels order the OOS set far less, App. B.3). The reach of the prior past the training range appears finite. On three of the four tasks with the smallest extrapolation distance the OOS set is compact and well ordered at the far end of that component, most visible on band gap (Fig. 2e), with the gradient continuing past the training range. On the long-tailed tasks the ordering reverses, and on critical temperature the highest labels scatter into the training mass (Fig. 2j). The parity plots show the same limit in the predicted values, which past that distance fall inside the training range (App. B.1).

Finetuning absorbs the geometry transduction exploits. Segal et al. [12] reported OOS gains from bilinear transduction (BLT) [11], which predicts a *query* material from a training material, the *anchor*, and the *difference* between their embeddings. The anchor is chosen so that this difference stays within the range seen in training (App. A.6). The error reduction they found on composition descriptors reproduces here with structures as input (tables in App. B.4). On both descriptors and on the raw MLIP embeddings, swapping the MLP predictor for BLT lowers median OOS error by 4% to 6%, usually at some cost in ID error, as in their setting. The gain, however, is uneven across tasks and does not extend to screening, where recovery rises only on CFID and falls on SOAP and on the raw embeddings.

Yet on the finetuned embeddings the swap moves median OOS error by under 1% and BLT reduces to the MLP it replaced (App. B.4). With an affine label the anchor’s contribution cancels against that of the difference, so the prediction can be read off the query alone (Proposition 1, App. B.4). Additionally, the anchor cannot correct an OOS structure the backbone has misplaced, since BLT and the MLP read the same embedding. Finetuning therefore absorbs the geometry that transduction exploits, and the argument extends to any transductive predictor of that form.

4 Conclusion

Across seven properties, pretrained MLIP embeddings outperform the two classical descriptors we test and training from scratch in OOS prediction and label efficiency, inverting prevailing intuition [21]. With matched nonlinear predictors, their advantage over descriptors isolates the learned representation as the source of improved extrapolation. Pretrained MLIP embeddings are therefore the representation to start from for OOS property prediction.

One limitation is the property labels, which come from density-functional calculations, least reliable at the extremes of the property range [24, 25]. We keep those entries, since removing them would edit exactly the OOS set under test (App. A.1). The other is the pretraining corpus, which holds almost every test crystal, though not its label. Four of the seven properties derive from higher derivatives of the energy the MLIP learns and are thus related to its pretraining objective. The largest pretraining gain is nevertheless on band gap, which requires electronic information a potential does not provide. It is also the one property with a test set that a pretraining corpus verifiably excludes, and there we show that an MLIP pretrained on that corpus keeps the advantage (App. C.1). The advantage also holds when four further backbones replace Orb-v3 and when two published crystal GNNs replace the from-scratch model (App. C.2 and C.3). As foundation potentials improve, the label efficiency and OOS recovery we report are likely a lower bound for this setup.

Acknowledgments and Disclosure of Funding

We thank Bowen Deng and Ty Perez for helpful discussions. The authors acknowledge the MIT Office of Research Computing and Data for providing high performance computing resources that have contributed to the research results reported within this paper.

References

- [1] Steven K. Kauwe, Jake Graser, Ryan Murdock, and Taylor D. Sparks. Can machine learning find extraordinary materials? *Computational Materials Science*, 174:109498, 2020. doi: 10.1016/j.commatsci.2019.109498.
- [2] Stefano Curtarolo, Wahyu Setyawan, Gus L.W. Hart, Michal Jahnatek, Roman V. Chepurskii, Richard H. Taylor, Shidong Wang, Junkai Xue, Kesong Yang, Ohad Levy, Michael J. Mehl, Harold T. Stokes, Denis O. Demchenko, and Dane Morgan. Aflow: An automatic framework for high-throughput materials discovery. *Computational Materials Science*, 58:218–226, 2012. doi: 10.1016/j.commatsci.2012.02.005.
- [3] Anubhav Jain, Shyue Ping Ong, Geoffroy Hautier, Wei Chen, William Davidson Richards, Stephen Dacek, Shreyas Cholia, Dan Gunter, David Skinner, Gerbrand Ceder, and Kristin A. Persson. Commentary: The materials project: A materials genome approach to accelerating materials innovation. *APL Materials*, 1(1), 2013. doi: 10.1063/1.4812323.
- [4] Anand Babu, Rog rio Almeida Gouv a, and Gian-Marco Rignanese. Towards automated discovery: A review of generative models, multimodal learning and closed-loop workflows in inverse materials design. *CoRR*, abs/2606.02507, 2026. doi: 10.48550/ARXIV.2606.02507.
- [5] Claudio Zeni, Robert Pinsler, Daniel Z gner, Andrew Fowler, Matthew Horton, Xiang Fu, Zilong Wang, Aliaksandra Shysheya, Jonathan Crabb , Shoko Ueda, Roberto Sordillo, Lixin Sun, Jake Smith, Bichlien Nguyen, Hannes Schulz, Sarah Lewis, Chin-Wei Huang, Ziheng Lu, Yichi Zhou, Han Yang, Hongxia Hao, Jielan Li, Chunlei Yang, Wenjie Li, Ryota Tomioka, and Tian Xie. A generative model for inorganic materials design. *Nature*, 639(8055):624–632, 2025. doi: 10.1038/s41586-025-08628-5.
- [6] Evan R. Antoniuk, Peggy Li, Nathan Keilbart, Stephen Weitzner, Bhavya Kailkhura, and Anna M. Hiszpanski. Active learning enables generation of molecules that advance the known pareto front. *npj Computational Materials*, 12(1), 2026. doi: 10.1038/s41524-025-01924-8.
- [7] Jerret Ross, Brian Belgodere, Vijil Chenthamarakshan, Inkit Padhi, Youssef Mroueh, and Payel Das. Large-scale chemical language representations capture molecular structure and properties. *Nature Machine Intelligence*, 4(12):1256–1264, 2022. doi: 10.1038/s42256-022-00580-7.
- [8] Gengmo Zhou, Zhifeng Gao, Qiankun Ding, Hang Zheng, Hongteng Xu, Zhewei Wei, Linfeng Zhang, and Guolin Ke. Uni-mol: A universal 3d molecular representation learning framework. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

- [9] 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, 2024.
- [10] Brandon M. Wood, Misko Dzamba, Xiang Fu, Meng Gao, Muhammed Shuaibi, Luis Barroso-Luque, Kareem Abdelmaqsoud, Vahe Gharakhanyan, John R. Kitchin, Daniel S. Levine, Kyle Michel, Anuroop Sriram, Taco S. Cohen, Abhishek Das, Sushree Jagriti Sahoo, Ammar Rizvi, Zachary W. Ulissi, and Larry Zitnick. UMA: A family of universal models for atoms. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*.
- [11] Aviv Netanyahu, Abhishek Gupta, Max Simchowitz, Kaiqing Zhang, and Pulkit Agrawal. Learning to extrapolate: A transductive approach. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.
- [12] Nofit Segal, Aviv Netanyahu, Kevin P. Greenman, Pulkit Agrawal, and Rafael Gómez-Bombarelli. Known unknowns: Out-of-distribution property prediction in materials and molecules. *npj Computational Materials*, 11(1), 2025. doi: 10.1038/s41524-025-01808-x.
- [13] Han Yang, Chenxi Hu, Yichi Zhou, Xixian Liu, Yu Shi, Jielan Li, Guanzhi Li, Zekun Chen, Shuizhou Chen, Claudio Zeni, Matthew Horton, Robert Pinsler, Andrew Fowler, D. Zugner, Tianyidan Xie, Jake Smith, Lixin Sun, Qian Wang, Lingyu Kong, Chang Liu, Hongxia Hao, and Ziheng Lu. Mattersim: A deep learning atomistic model across elements, temperatures and pressures, 2024.
- [14] Nima Shoghi, Adeesh Kolluru, John R. Kitchin, Zachary W. Ulissi, C. Lawrence Zitnick, and Brandon M. Wood. From molecules to materials: Pre-training large generalizable models for atomic property prediction. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.
- [15] So Yeon Kim, Yang Jeong Park, and Ju Li. Leveraging neural network interatomic potentials for a foundation model of chemistry. *npj Computational Materials*, 2026. doi: 10.1038/s41524-026-02167-x.
- [16] Rogério Almeida Gouvêa, Pierre-Paul De Breuck, Tatiane Pretto, Gian-Marco Rignanese, and Marcos José Leite Santos. Combining feature-based approaches with graph neural networks and symbolic regression for synergistic performance and interpretability. *npj Computational Materials*, 12(1), 2026. doi: 10.1038/s41524-025-01938-2.
- [17] Lingyu Kong, Nima Shoghi, Guoxiang Hu, Pan Li, and Victor Fung. Matteredtune: an integrated, user-friendly platform for fine-tuning atomistic foundation models to accelerate materials simulation and discovery. *Digital Discovery*, 4(8):2253–2262, 2025. doi: 10.1039/d5dd00154d.
- [18] Enda Xiao and Terumasa Tadano. Accurate screening of functional materials with machine-learning potential and transfer-learned regressions: Heusler alloy benchmark. *npj Computational Materials*, 12(1), 2026. doi: 10.1038/s41524-026-02013-0.
- [19] Chengqian Zhang, Duoduo Zhang, Anyang Peng, Mingyu Guo, Yuzhi Zhang, Lei Wang, Guolin Ke, Linfeng Zhang, Tiejun Li, and Han Wang. Multi-task fine-tuning enables robust out-of-distribution generalization in atomistic models, 2026.
- [20] Kangming Li, Andre Niyongabo Rubungo, Xiangyun Lei, Daniel Persaud, Kamal Choudhary, Brian DeCost, Adjai Bouso Dieng, and Jason Hattrick-Simpers. Probing out-of-distribution generalization in machine learning for materials. *Communications Materials*, 6(1), 2025. doi: 10.1038/s43246-024-00731-w.
- [21] Hajime Shimakawa, Akiko Kumada, and Masahiro Sato. Extrapolative prediction of small-data molecular property using quantum mechanics-assisted machine learning. *npj Computational Materials*, 10(1), 2024. doi: 10.1038/s41524-023-01194-2.
- [22] Alexander Dunn, Qi Wang, Alex Ganose, Daniel Dopp, and Anubhav Jain. Benchmarking materials property prediction methods: the matbench test set and automatminer reference algorithm. *npj Computational Materials*, 6(1), 2020. doi: 10.1038/s41524-020-00406-3.
- [23] Cormac Toher, Corey Oses, Jose J. Plata, David Hicks, Frisco Rose, Ohad Levy, Maarten de Jong, Mark Asta, Marco Fornari, Marco Buongiorno Nardelli, and Stefano Curtarolo. Combining the aflow gibbs and elastic libraries to efficiently and robustly screen thermomechanical properties of solids. *Physical Review Materials*, 1(1), 2017. doi: 10.1103/physrevmaterials.1.015401.

- [24] Masato Ohnishi, Tianqi Deng, Pol Torres, Zhihao Xu, Terumasa Tadano, Haoming Zhang, Wei Nong, Masatoshi Hanai, Zeyu Wang, Michimasa Morita, Zhiting Tian, Ming Hu, Xiulin Ruan, Ryo Yoshida, Toyotaro Suzumura, Lucas Lindsay, Alan J. H. McGaughey, Tengfei Luo, Kedar Hippalgaonkar, and Junichiro Shiomi. Database and deep-learning scalability of anharmonic phonon properties by automated brute-force first-principles calculations. *npj Computational Materials*, 12(1), 2026. doi: 10.1038/s41524-026-02033-w. Data: Phonix, DOI 10.57967/hf/8233 (HuggingFace phonix-db/phonix-summary).
- [25] Tiago F. T. Cerqueira, Antonio Sanna, and Miguel A. L. Marques. Sampling the materials space for conventional superconducting compounds. *Advanced Materials*, 36(1):2307085, 2024. doi: 10.1002/adma.202307085. Data: 10.24435/materialscloud:qv-bq.
- [26] Benjamin Rhodes, Sander Vandenhaute, Vaidotas Šimkus, James Gin, Jonathan Godwin, Timothy T. Duignan, and Mark Neumann. Orb-v3: atomistic simulation at scale, 2025.
- [27] Albert P. Bartók, Risi Kondor, and Gábor Csányi. On representing chemical environments. *Physical Review B*, 87(18):184115, 2013. doi: 10.1103/physrevb.87.184115.
- [28] Kamal Choudhary, Brian DeCost, and Francesca Tavazza. Machine learning with force-field-inspired descriptors for materials: Fast screening and mapping energy landscape. *Physical Review Materials*, 2(8), 2018. doi: 10.1103/physrevmaterials.2.083801.
- [29] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.
- [30] Vardan Papyan, X. Y. Han, and David L. Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40):24652–24663, 2020. doi: 10.1073/pnas.2015509117.
- [31] George Andriopoulos, Zixuan Dong, Li Guo, Zifan Zhao, and Keith W. Ross. The prevalence of neural collapse in neural multivariate regression. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.
- [32] 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. doi: 10.1038/s42256-023-00716-3.
- [33] Jonathan Schmidt, Noah Hoffmann, Hai-Chen Wang, Pedro Borlido, Pedro J. M. A. Carriço, Tiago F. T. Cerqueira, Silvana Botti, and Miguel A. L. Marques. Machine-learning-assisted determination of the global zero-temperature phase diagram of materials. *Advanced Materials*, 35(22):2210788, 2023. doi: 10.1002/adma.202210788.
- [34] Shyue Ping Ong, William Davidson Richards, Anubhav Jain, Geoffroy Hautier, Michael Kocher, Shreyas Cholia, Dan Gunter, Vincent L. Chevrier, Kristin A. Persson, and Gerbrand Ceder. Python materials genomics (pymatgen): A robust, open-source python library for materials analysis. *Computational Materials Science*, 68:314–319, 2013. doi: 10.1016/j.commatsci.2012.10.028.
- [35] Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter W. Battaglia. Learning to simulate complex physics with graph networks. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 8459–8468. PMLR, 2020.
- [36] Mark Neumann, James Gin, Benjamin Rhodes, Steven Bennett, Zhiyi Li, Hitarth Choubisa, Arthur Hussey, and Jonathan Godwin. Orb: A fast, scalable neural network potential, 2024.
- [37] Lauri Himanen, Marc O. J. Jäger, Eiaki V. Morooka, Filippo Federici Canova, Yashasvi S. Ranawat, David Z. Gao, Patrick Rinke, and Adam S. Foster. Dscribe: Library of descriptors for machine learning in materials science. *Comput. Phys. Commun.*, 247:106949, 2020. doi: 10.1016/j.cpc.2019.106949.
- [38] James P. Darby, James R. Kermode, and Gábor Csányi. Compressing local atomic neighbourhood descriptors. *npj Computational Materials*, 8:166, 2022. doi: 10.1038/s41524-022-00847-y.
- [39] Logan Ward, Alexander Dunn, Alireza Faghaninia, Nils E.R. Zimmermann, Saurabh Bajaj, Qi Wang, Joseph Montoya, Jiming Chen, Kyle Bystrom, Maxwell Dylla, Kyle Chard, Mark Asta, Kristin A. Persson, G. Jeffrey Snyder, Ian Foster, and Anubhav Jain. Matminer: An open source toolkit for materials data mining. *Computational Materials Science*, 152:60–69, 2018. doi: 10.1016/j.commatsci.2018.05.018.

- [40] Xinwei Shen and Nicolai Meinshausen. Engression: extrapolation through the lens of distributional regression. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 87(3):653–677, 2024. doi: 10.1093/jrsssb/qkae108.
- [41] Logan Ward, Ankit Agrawal, Alok Choudhary, and Christopher Wolverton. A general-purpose machine learning framework for predicting properties of inorganic materials. *npj Computational Materials*, 2(1), 2016. doi: 10.1038/npjcompumats.2016.28.
- [42] Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirremeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, 2025. doi: 10.1038/s41586-024-08328-6.
- [43] Zeyu Wang, Shuya Yamazaki, Martin Hoffmann Petersen, Masato Ohnishi, Tomiya Yamamoto, Wei Nong, Jianghai Wang, Ruiming Zhu, Masatoshi Hanai, Michimasa Morita, Toyotaro Suzumura, Zekun Ren, Junichiro Shiomi, and Kedar Hippalgaonkar. Fast and accurate prediction of lattice thermal conductivity via machine learning surrogates, 2026.
- [44] Woruo Chen, Yao Tian, Nian Liao, Yuchao Deng, Dejun Jiang, and Dongsheng Cao. TabPFN opens new avenues for small-data tabular learning in drug discovery. *Journal of Chemical Information and Modeling*, 66(7):3525–3539, 2026. doi: 10.1021/acs.jcim.5c02823.
- [45] Peter J. Huber and Elvezio M. Ronchetti. *Robust Statistics*. Wiley, 2 edition, 2009.
- [46] Kamal Choudhary and Brian DeCost. Atomistic line graph neural network for improved materials property predictions. *npj Computational Materials*, 7:185, 2021.
- [47] Xavier Bouthillier, Pierre Delaunay, Mirko Bronzi, Assya Trofimov, Brennan Nichyporuk, Justin Szeto, Naz Sepah, Edward Raff, Kanika Madan, Vikram Voleti, Samira Ebrahimi Kahou, Vincent Michalski, Dmitriy Serdyuk, Tal Arbel, Chris Pal, Gaël Varoquaux, and Pascal Vincent. Accounting for variance in machine learning benchmarks. In *Proceedings of Machine Learning and Systems*, 2021.
- [48] Hai-Chen Wang, Silvana Botti, and Miguel A. L. Marques. Predicting stable crystalline compounds using chemical similarity. *npj Computational Materials*, 7(1), 2021. doi: 10.1038/s41524-020-00481-6.
- [49] Janosh Riebesell, Rhys E. A. Goodall, Philipp Benner, Yuan Chiang, Bowen Deng, Gerbrand Ceder, Mark Asta, Alpha A. Lee, Anubhav Jain, and Kristin A. Persson. A framework to evaluate machine learning crystal stability predictions. *Nature Machine Intelligence*, 7(6):836–847, 2025. doi: 10.1038/s42256-025-01055-1.
- [50] Andrew S. Rosen, Shaelyn M. Iyer, Debmalya Ray, Zhenpeng Yao, Alán Aspuru-Guzik, Laura Gagliardi, Justin M. Notestein, and Randall Q. Snurr. Machine learning the quantum-chemical properties of metal–organic frameworks for accelerated materials discovery. *Matter*, 4(5):1578–1597, 2021. doi: 10.1016/j.matt.2021.02.015. Data: QMOF Database, DOI 10.6084/m9.figshare.13147324 (figshare v18).
- [51] Andrew S. Rosen, Victor Fung, Patrick Huck, Cody T. O’Donnell, Matthew K. Horton, Donald G. Truhlar, Kristin A. Persson, Justin M. Notestein, and Randall Q. Snurr. High-throughput predictions of metal–organic framework electronic properties: theoretical challenges, graph neural networks, and data exploration. *npj Computational Materials*, 8(1), 2022. doi: 10.1038/s41524-022-00796-6.
- [52] Ilyes Batatia, Philipp Benner, Yuan Chiang, Alin M. Elena, Dávid P. Kovács, Janosh Riebesell, Xavier R. Advincula, Mark Asta, Matthew Avaylon, William J. Baldwin, and Gábor Csányi. A foundation model for atomistic materials chemistry. *The Journal of Chemical Physics*, 163(18):184110, 2025. doi: 10.1063/5.0297006.
- [53] Cesare Malosso, Filippo Bigi, Paolo Pegolo, Joseph W. Abbott, Philip Loche, Mariana Rossi, Michele Ceriotti, and Arslan Mazitov. High-quality, high-information datasets for universal atomistic machine learning, 2026.
- [54] Duo Zhang, Anyang Peng, Chun Cai, Wentao Li, Yuanchang Zhou, Jinzhe Zeng, Mingyu Guo, Chengqian Zhang, Bowen Li, Hong Jiang, Tong Zhu, Weile Jia, Linfeng Zhang, and Han Wang. A graph neural network for the era of large atomistic models. *npj Computational Materials*, 12(1), 2026. doi: 10.1038/s41524-026-02146-2.
- [55] Keqiang Yan, Cong Fu, Xiaofeng Qian, Xiaoning Qian, and Shuiwang Ji. Complete and efficient graph transformers for crystal material property prediction. In *International Conference on Learning Representations (ICLR)*, 2024.

A Method details	9
A.1 Datasets · A.2 Pretraining overlap · A.3 MLIP backbone · A.4 Descriptor construction · A.5 Readouts and predictors · A.6 Bilinear transduction · A.7 Training and model selection · A.8 Metrics and confidence intervals	
B Complete results	15
B.1 Results at the full training set · B.2 Learning curves · B.3 Embedding-space analysis · B.4 Transductive methods	
C Additional experiments	23
C.1 Unseen materials · C.2 Other MLIP backbones · C.3 End-to-end GNN comparison · C.4 Selection rule · C.5 Predictor capacity	

A Method details

A.1 Datasets

The seven tasks draw on public datasets of computed properties, chosen to spread the target across electronic, mechanical, vibrational, optical, and superconducting behavior (Table 1 and Fig. 3). They differ in size and in label distribution, so the extrapolation distance defined below orders them throughout the paper.

Units and filters. Bulk modulus, Debye temperature, and thermal conductivity are modeled and reported in log units, base 10 everywhere in this paper. Matbench ships the bulk moduli in this form, and the logarithm on the other two follows Kauwe et al. [1] and Segal et al. [12]. We filter the thermal-conductivity set on the database’s own convergence diagnostics, dropping structures with imaginary phonon modes or badly fit third-order force constants, as its authors do [24]. The critical-temperature set is restricted by its authors to non-magnetic metals with at most eight atoms per cell, and we apply no further filter [25].

Label reliability. Two of our source papers report the limitation of Section 4 directly, with the thermal-conductivity database reporting unconverged solutions of the transport equation at its highest values [24] and the critical-temperature database reporting soft-mode false positives among its outliers [25]. A random split scatters such entries into training, whereas the largest-5% label split concentrates them in the set being scored. Beyond the thermal-conductivity filter above we apply no exclusions, since a further filter would edit exactly the OOS set under test.

Tail statistics. Write Y_{tr} , $Y_{\text{test}}^{\text{id}}$, and $Y_{\text{test}}^{\text{oos}}$ for the labels of the training set, the ID test set, and the OOS test set. The extrapolation distance (Section 2) is the gap between the mean labels of the two test sets,

$$d_{\text{oos}} = \frac{\text{mean}(Y_{\text{test}}^{\text{oos}}) - \text{mean}(Y_{\text{test}}^{\text{id}})}{\text{MAD}(Y_{\text{tr}})},$$

in units of the training labels’ MAD, their mean absolute deviation around the median (defined with the metrics in App. A.8). The same order follows from the training labels alone, as the gap from their 95th percentile to their maximum in the same unit, so the order is known before any OOS evaluation.

A.2 Pretraining overlap

The backbone we use, Orb-v3 (direct-inf-mpa), is pretrained on MPtrj [32] and Alexandria [33], which together form the MPA corpus of 4.6M relaxed structures, one per material. Corpora of this size are built to cover the known inorganic crystals, so overlap with any downstream property dataset is expected. Essentially every structure in the seven property datasets has been seen, in the training set and in both test sets alike (Table 2, counting the structures the corpus does *not* hold).

Criterion and calibration. Every structure is matched against every corpus entry of its composition with pymatgen’s structure matcher [34] at its defaults, and counts as seen if any matches. Calibrated in both directions, the defaults match 99.5% of the 6,538 thermal-conductivity structures that carry Materials Project identifiers, and falsely match 4.9% of pairs known to be distinct materials of one composition. The net effect slightly overstates exposure, which is the conservative direction here.

Table 1: The seven tasks, ordered by extrapolation distance. Each $|\mathcal{D}|$ column gives a set size, and the ID validation and test sets, equal in size, share one column. The threshold τ is in the task’s unit, and the extrapolation distance d_{oos} is defined above. Where the short names drop a qualifier, the full names are Voigt–Reuss–Hill bulk modulus, lattice thermal conductivity, superconducting critical temperature, and last phonon-DOS peak.

Task	Source	$ \mathcal{D}_{\text{tr}} $	$ \mathcal{D}_{\text{val,test}}^{\text{id}} $	$ \mathcal{D}_{\text{test}}^{\text{oos}} $	Unit	τ	d_{oos}
Bulk modulus	Matbench [22]	9,889	275	549	log GPa	2.4	2.2
Debye temperature	AFLOW [2, 23]	5,087	141	283	log K	2.9	3.5
Thermal conductivity	Phonix [24]	6,066	169	337	log W/mK	1.5	4.0
Band gap	Matbench [22]	95,501	2,653	5,306	eV	4.5	4.4
Phonon peak	Matbench [22]	1,139	31	63	cm^{-1}	1400	7.3
Critical temperature	Cerqueira et al. [25]	7,425	206	412	K	11	7.8
Refractive index	Matbench [22]	4,288	119	238	—	4.4	11.0

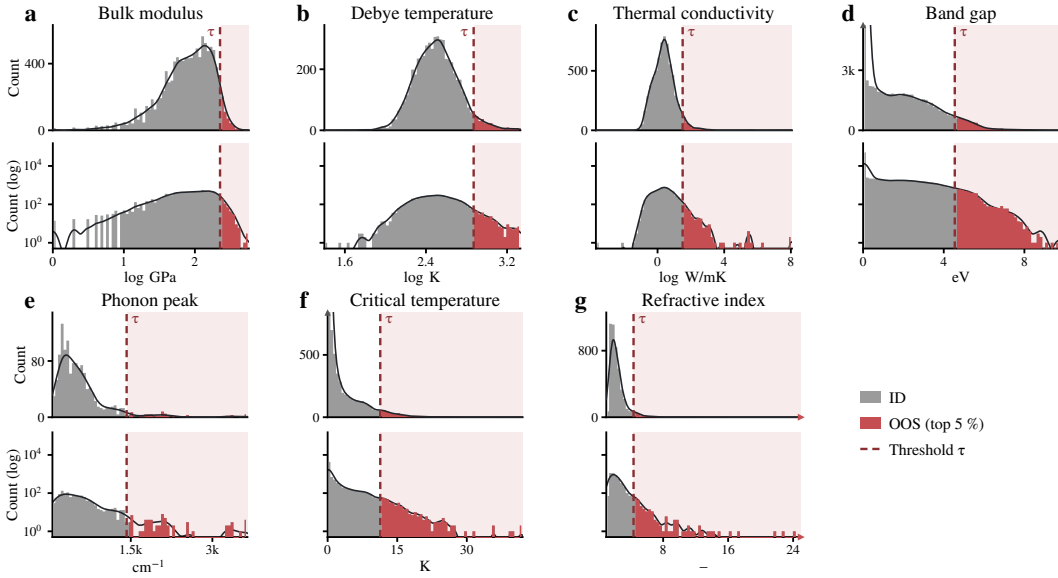


Figure 3: Label distributions of the seven tasks in report units, as per-task counts on a linear axis (top) and on one shared logarithmic axis (bottom), with the threshold τ drawn. The refractive-index view (g) is capped at a refractive index of 25, matching the parity plots (App. B.1), and 9 structures lie above it.

Scope. As far as we are aware, no large dataset with diverse property labels has measured exclusion from a pretraining corpus, so Appendix C.1 tests structure novelty on band gap alone, on a pool the OMat24 corpus verifiably excludes.

Table 2: Share of structures *not* found in the pretraining corpus in each set, with the unseen count in parentheses. Tasks in extrapolation-distance order and set sizes as in Table 1.

Task	$\mathcal{D}_{\text{tr}}$	$\mathcal{D}_{\text{test}}^{\text{id}}$	$\mathcal{D}_{\text{test}}^{\text{oos}}$
Bulk modulus	<1% (1)	0% (0)	0% (0)
Debye temperature	<1% (16)	<1% (1)	<1% (1)
Thermal conductivity	<1% (28)	<1% (1)	1% (5)
Band gap	<1% (433)	<1% (6)	<1% (50)
Phonon peak	0% (0)	0% (0)	0% (0)
Critical temperature	<1% (10)	0% (0)	0% (0)
Refractive index	<1% (2)	0% (0)	0% (0)

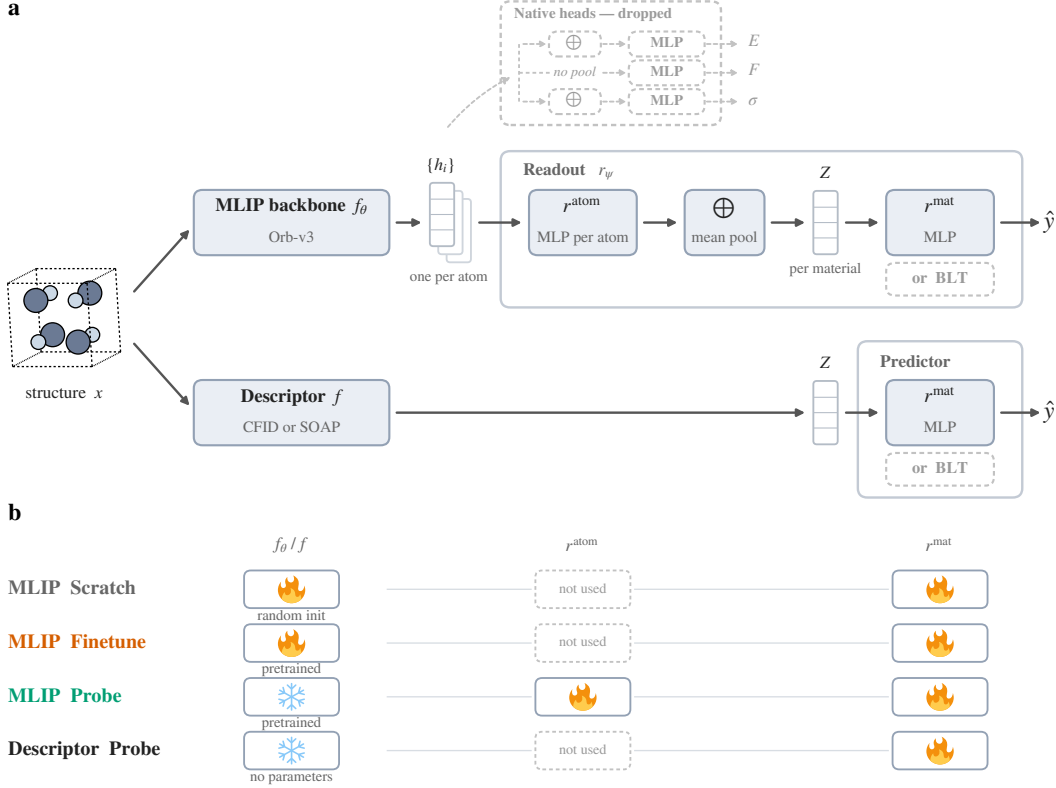


Figure 4: **a**, A structure enters the Orb-v3 backbone, whose native energy, force, and stress heads are dropped, and the per-atom embeddings feed a readout of an MLP per atom, a mean pool, and an MLP to the scalar property. A descriptor takes the lower path to a predictor whose size matches the whole readout. **b**, Finetuning trains backbone and readout, probing freezes the backbone, and from-scratch training runs the same architecture from random initialization.

A.3 MLIP backbone

An MLIP predicts a structure’s energy, the force on each atom, and the cell stress. A conservative model obtains the forces and the stress by differentiating its energy, while a direct model emits all three from heads of its own. The backbone builds a graph over the atoms of a structure x , one node per atom with edges to every neighbor within a cutoff radius, and a stack of message-passing layers updates each node’s embedding at constant width. We drop the native heads and keep the final layer’s embeddings, $f_\theta(x) = \{h_i\}$, as the representation a readout pools (Fig. 4).

Orb-v3 is a graph network simulator [35] with 24M parameters, released in a conservative and a direct variant. We take the direct checkpoint orb-v3-direct-inf-mpa-20250404, whose architecture the Orb papers give in full [26, 36]. Five message-passing layers act on a graph with a 6 Å cutoff, and each edge carries the distance and relative direction between its endpoints in a fixed basis expansion. The architecture is not rotation-equivariant, so its rotational invariance is learned and approximate. The final layer emits one embedding per atom, with each $h_i \in \mathbb{R}^{256}$. Orb-v3 is pretrained in two phases, a self-supervised denoising stage and then supervised training on MPTrj [32] and Alexandria [33], with the direct force and stress predictions distilled from the conservative variant.

A.4 Descriptor construction

SOAP. SOAP describes each atom by the density of its neighbors within a cutoff, expanded in a radial and an angular basis, and its power spectrum makes that expansion invariant to rotation. We compute SOAP with DScibe [37] at its canonical example configuration ($n_{\text{max}} = 8$ radial basis functions, angular frequency limit $l_{\text{max}} = 6$, and a 6 Å cutoff). The full multi-species power spectrum grows quadratically with the number of species [38], to about two million components per atom over

94 elements, so the species dimension must be compressed. DDescribe’s built-in μ_2 compression [38] sums the per-species neighbor densities into one total density before the power spectrum is formed, which cuts the vector to 252 dimensions and drops the identity of the neighbors. Per-atom vectors are then averaged within each element type and concatenated over the fixed 94-element basis, so a block position carries the central atom’s species and a structure becomes one 23k-dimensional vector.

CFID. We use matminer’s implementation of the JARVIS descriptor with all seven blocks on [28, 39]. The result is a 1,557-dimensional global vector whose largest block is 438 element-property statistics and whose remainder holds charge, radial, angular, dihedral, nearest-neighbor, and cell features. It therefore carries chemistry as well as geometry, whereas SOAP carries geometry alone.

Dimensionality reduction. In a hand-crafted descriptor the magnitude of a component is set by the formula that produces it, so the magnitude says nothing about how informative that component is. We standardize each descriptor component of nonzero training variance first and only then project onto the leading principal components without whitening, both fitted on the training samples alone and applied to every set. In contrast, a learned embedding fits the magnitude of each coordinate to data, so its variance already reflects how much information that coordinate holds, and the MLIP embeddings enter their readouts raw.

The reduction truncates SOAP alone, from 23,688 components to 1,024 (on band gap the standardized spectrum reaches 95% of its variance in about 700 components and 99% in about 1,400). CFID is small enough to go untruncated, so its reduction amounts to standardizing the components with nonzero training variance. In the learning curves every training-set size reuses the same standardization and projection, fitted on the full training set, so the sizes vary the number of labels alone.

A.5 Readouts and predictors

A predictor maps a representation to the scalar property, and every method in this paper uses one of the same form. What differs between them follows from the representation, since an MLIP emits one embedding per atom and the predictor must reduce that set to a single vector, while a descriptor emits such a vector already. The shared form is

$$r_\psi = r^{\text{mat}} \circ \bigoplus \circ r^{\text{atom}}, \quad (1)$$

with r^{atom} an MLP applied to each atom, $\bigoplus$ an aggregation, or pool, over the atoms, and r^{mat} an MLP on the aggregate. Our targets are all intensive, so $\bigoplus$ is an unweighted mean throughout, which is the one place the property’s physics constrains the predictor. We write $Z = \bigoplus_i r^{\text{atom}}(h_i)$ for the per-material embedding the pool emits. All MLPs use SiLU activations and no normalization.

MLIP readouts and descriptor predictors. On an MLIP the input is the set $\{h_i\}$ and the composed map, which the MLIP literature calls a readout, is trained as one. Here, the probe gives each of r^{atom} and r^{mat} two 256-wide layers (1.1% of the backbone), while finetuning and from-scratch training drop r^{atom} , since everything before Z can adapt, so their Z is the mean of $\{h_i\}$ itself and their r^{mat} has two layers. A descriptor’s predictor is an r^{mat} alone (App. A.4), four layers at width 256 with the last emitting the scalar. Two further predictors replace r^{mat} on Z , the linear probe (App. A.7) and bilinear transduction (App. A.6). The three probes therefore share depth and width, so a comparison between them holds the predictor fixed and varies the representation alone, and the predictor capacity ablation brackets every representation with the same four sizes (App. C.5).

Pool position. We ablate the position of the mean pool, moving it through the four-layer stack at a fixed parameter count, on every task at the full training set and on band gap at all eight training-set sizes. No OOS metric moves, and the reported two-and-two split is as often ahead of the other positions as behind.

Why an MLP. We use an MLP as the predictor because the common alternatives in materials property prediction do not extrapolate. Tree ensembles predict an average of training labels [40, 41], so no prediction leaves the training range, and prior-data fitted models such as TabPFN [42–44] place almost all of their mass inside it. Neither can flag a structure past τ , so neither recovers the OOS set whatever representation it reads.

A.6 Bilinear transduction

Bilinear transduction (BLT) [11] was developed for inputs that leave the training support, and Segal et al. [12] applied it to label extrapolation, which is the setting here.

Formulation. The predictor is a bilinear form on two MLP trunks u and v , one reading the difference between query and anchor embeddings and one the anchor,

$$\hat{y}_j = \langle u(Z_j - Z_i), v(Z_i) \rangle, \quad (2)$$

trained on pairs (i, j) of training structures ordered by label, anchor i below query j , so that the difference points toward the large-value tail. At inference the anchor is not given, and it is chosen so that the difference stays inside the training support,

$$a(Z) = \arg \min_{i \in \mathcal{D}_{\text{tr}}} \min_{\delta \in \Delta} \|(Z - Z_i) - \delta\|^2, \quad (3)$$

where Δ is a reference set of differences sampled between the candidate anchors and oriented by label as the training pairs are. Eq. 2 is then evaluated at $(Z - Z_{a(Z)}, Z_{a(Z)})$, following the full treatment of Netanyahu et al. [11]. Any predictor on such a pair is a transductive predictor in their sense, and BLT is the instance whose map is bilinear. Here each trunk is an r^{mat} , two 256-wide layers on the pooled Z , and the form has rank 32 (148k parameters).

Baseline. The matched baseline is an instance of r^{mat} of the same shape, two 256-wide layers on Z (66k parameters), and on the finetuned embeddings it is already the reported readout. The raw MLIP embeddings and both descriptors are reported with larger predictors, but BLT acts on Z alone, so both sides of those comparisons take this smaller one instead (App. C.5). Each side selects its own learning rate under the rule of Appendix A.7, and every run is at the seven tasks and the full training-set size of the main results.

Implementation. Following Segal et al. [12], the predictor outputs the query’s label directly rather than a residual on the anchor’s, and it uses the single nearest anchor instead of an average over the admissible ones. We draw both anchor and reference sets at random once, 1,024 anchors from the training set and 4,096 differences between them. Larger sets, up to the whole training set as anchors and sixteen times as many references, gave no gain on validation error. Both test sets are queried against the same anchor and reference sets, drawn from training structures alone.

A.7 Training and model selection

Training setup. The three GNN training methods and the descriptor probes share one training setup (Table 3), and the linear probes are described below. Batch size 64 is the largest that fits GPU memory when the backbone trains, probes run at 512 with no such limit, and the batch drops to N wherever the training set is smaller. The step targets are set so that every configuration converges inside its budget. Every evaluation reads the EMA-averaged weights, and each run keeps the checkpoint with the lowest MAE on $\mathcal{D}_{\text{val}}^{\text{id}}$. Weight decay is zero throughout, with that checkpoint selection supplying the regularization in its place, following Orb’s own training [36] and the usual practice for probes on a frozen backbone [15, 19, 43].

Seeds and subsampling. A seed sets the parameter initialization and the training subsample, and each subsample comes from the full training set alone, as the first N of a seeded permutation, so the subsamples of one seed are nested, every size a subset of the next.

Learning rate. For every model but the linear probes the learning rate is the only swept hyperparameter. Four points are run per task and method at the first seed, and the lowest MAE on $\mathcal{D}_{\text{val}}^{\text{id}}$ is selected. The sweep runs at the full training set, and the selected rate is held fixed for the reduced- N runs of the learning curves. Probing and finetuning share the grid $\{10^{-4}, 3 \times 10^{-4}, 10^{-3}, 3 \times 10^{-3}\}$, and from-scratch training uses $\{3 \times 10^{-5}, 10^{-4}, 3 \times 10^{-4}, 10^{-3}\}$, one step lower because there the backbone trains at the readout’s rate. Where the selected rate falls on an edge of the grid, the grid is extended by one point past it. Predictors introduced elsewhere in the appendix follow the same grids and rule. The learning rate, the stopping point within a run, and the ridge penalty of the linear probes described below are all selected on $\mathcal{D}_{\text{val}}^{\text{id}}$. Appendix C.4 compares those picks against the ones an OOS oracle would make.

Table 3: The training setup, shared settings above the rule and per-method settings below. The probe column also covers the descriptor probes. A run converts its step target to epochs at its batch size and clamps the result to the epoch range.

	Scratch	Finetune	Probe
Loss		L1	
Optimizer		Adam	
Schedule	cosine, warmup over 5% of epochs, anneal to 1% of peak		
Gradient clipping		global norm 10	
Weight averaging	EMA over a window of 5% of the steps		
Seeds		3	
Step target	20,000	20,000	100,000
Epoch clamp	150–3,000	150–3,000	100–8,000
Batch size	64	64	512
Backbone learning rate	equal to the readout’s	$0.05 \times$ the readout’s	frozen

Linear probes. As the linear probe we use a ridge-penalized Huber regression with an unpenalized intercept, fitted with scikit-learn’s HuberRegressor [45], with the features standardized and the target centered on the training samples. The penalty λ is the only swept quantity, taken from one shared grid of 16 points a decade apart from 10^{-5} to 10^{10} and selected on the MAE of $\mathcal{D}_{\text{val}}^{\text{id}}$, which is held at full size at every training-set size. The grid reaches 10^{10} because the descriptor fits run with more components than training samples at the small training sets, where the strong-penalty endpoint (the intercept-only fit predicting the training mean) is the right limit.

A.8 Metrics and confidence intervals

We report MAE in each task’s report unit (Table 1). MAD:MAE condenses the error into one metric where higher is better [46]. It divides the spread of the test labels, their mean absolute deviation around the median mean $|y - \text{median}(y)|$, by the model’s MAE, so 1 matches the best constant predictor under the L1 loss.

On the OOS set the median lies above every training label, so far smaller MAD:MAE values are expected there. The two test sets are therefore not comparable on any metric, and every comparison the paper draws is between methods within one test set. We center the constant on the median, the optimum under L1, where ALIGNN [46] uses the mean.

Screening. A screen is run to find the extraordinary materials, so the metric of interest is the share of them a model recovers, as in Kauwe et al. [1] and Segal et al. [12]. OOS recovery, the screen’s true-positive rate (TPR), measures it as the share of $\mathcal{D}_{\text{test}}^{\text{OOS}}$ whose prediction clears the threshold τ . The false-positive rate (FPR), the share of $\mathcal{D}_{\text{test}}^{\text{id}}$ predicted above that same τ , rules out a model that gains recovery by predicting everything higher. Spearman ρ measures how well a method ranks the set it scores.

Confidence intervals. The interval on every scalar is ± 1 standard error (Wilson for the two rates, below) and combines the training randomness, measured across the three seeds, with the sampling uncertainty of the finite test set. Both terms come from the standard two-way random-effects decomposition of the per-sample score matrix, estimated by method of moments, following the variance accounting of Bouthillier et al. [47],

$$\text{Var}(\bar{m}) = \sigma_{\text{seed}}^2/S + \sigma_{\text{sample}}^2/n + \sigma_{\text{resid}}^2/(nS),$$

with $\bar{m}$ the reported mean score over S seeds and n test samples, σ_{seed}^2 the variance of the shift a seed applies across all samples, σ_{sample}^2 that of the per-sample difficulty shared by every seed, and σ_{resid}^2 their interaction. MAE enters the decomposition directly, while MAD:MAE and ρ are nonlinear in the samples, so their sample term is a bootstrap with one shared draw per replicate and the MAD recomputed inside it. For the two rate metrics the variance is converted to an effective sample size $n_{\text{eff}} = \bar{m}(1 - \bar{m})/\text{Var}(\bar{m})$ and the interval is Wilson at n_{eff} , which stays correct at a measured zero. The precision at 5% prevalence derived in Appendix B.1 has no estimator of its own, and the two rates’ Wilson edges propagate through its formula.

Two methods on one task share their test samples and seeds, so their contrasts are paired and overlapping intervals are conservative for a difference. One table reports such a difference as its entry and pairs its interval accordingly (Table 7). Every other interval in the paper is marginal, so no difference between two methods’ rates is tested. The linear probes are deterministic and have no seed term.

B Complete results

B.1 Results at the full training set

Three tables report every method at the full training-set size: the screening rates and the precision they imply (Table 4), the MAE on both test sets (Table 5), and the Spearman ρ between prediction and label (Table 6).

In distribution. Finetuning has the lowest ID error of the five methods on every task, and the MLIP embeddings lead both descriptors on every task (Table 5). The best transfer method cuts the from-scratch ID error by 37% at the median, and the MLIP embeddings cut the better descriptor’s by 38%.

Precision at deployment prevalence. Recovery is a screen’s true-positive rate (TPR), and its false-positive rate (FPR) is measured on the ID test set (App. A.8). At a prevalence π of extraordinary materials in a candidate pool, the two rates fix the precision:

$$\text{precision}(\pi) = \frac{\text{TPR} \cdot \pi}{\text{TPR} \cdot \pi + \text{FPR} \cdot (1 - \pi)}.$$

Our two test sets together hold two extraordinary structures per ordinary one, so a precision read off them describes that prevalence alone. The label split of Section 2 defines extraordinary as the largest 5% by label, so we evaluate the precision at $\pi = 5\%$ (Table 4, third block).

At this deployment prevalence the MLIP probe has a precision interval above 0.5 on five of the seven tasks, with lower edges of 0.63 to 0.90.

Parity plots. The predictions themselves are in the parity grids (Fig. 5), with the label threshold τ ruled on both axes so the four quadrants are the confusion matrix of a screen at τ .

OOS predictions compress toward the training range, so the OOS errors of Table 5 are mostly a failure to reach the OOS label range. The exception is SOAP on phonon peak, as in Section 3.1, whose highest recovery in Table 4 comes from overprediction. The compression sets in at a reach past the training range that varies by task and method. Under finetuning or probing the predictions follow the label for a stretch past τ and then level off, furthest on band gap and Debye temperature, where the finetuned predictions track the label at a slope below one, and, probed, on critical temperature. Beyond that reach the OOS structures are predicted at or below τ . Training from scratch levels off at τ on every task, as do the descriptor probes on band gap, thermal conductivity, critical temperature, and refractive index. Section 3.2 reads the same limit off the embeddings.

B.2 Learning curves

The learning curves measure every method at training-set sizes from 10 labels to the full training set (App. A.7). Every curve is fitted as a power law $\text{MAE} \propto N^b$, one least-squares line through the three seeds’ points in log-log space over the whole measured range. The predictor is the reported one at every size, whereas an asymptotic scaling law would resize it to each training-set size, so b is an effective exponent over the range measured.

GNN training methods. On the seven-task learning curves behind the main-text pair (Fig. 6, extending Fig. 1c–d) the ordering of Section 3.1 holds along the learning curves, a transfer method leading from-scratch training at every size beyond a few tens of labels on both test sets. Finetuning takes its ID lead over the probe within a few hundred labels on every task. On the OOS set the division of the tasks by extrapolation distance repeats, with finetuning overtaking the probe at one to three thousand labels on three of the four short-tailed tasks, while the probe leads the three long-tailed

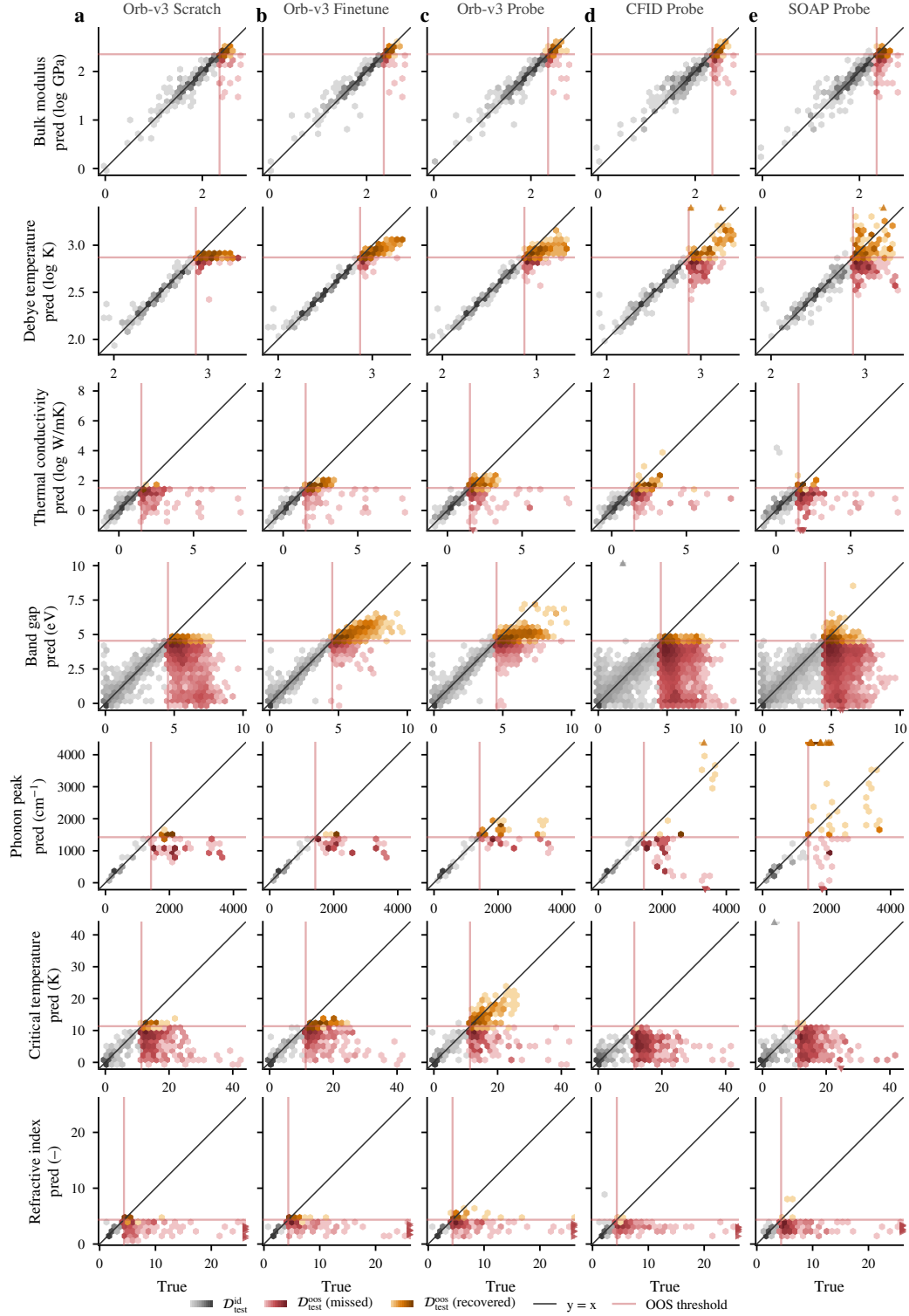


Figure 5: Parity plots of all five methods, one per column (a–e), on the seven tasks as hexagonal density tiles, ID in grey and OOS in red, with the label threshold τ ruled on both axes. Each tile is shaded by the logarithm of its point count, scaled within its panel. The upper-right quadrant holds the recovered OOS structures, drawn in orange, and their share of the OOS points is the recovery of Table 4. The upper-left holds the false positives, whose share of the in-distribution points is the FPR. Points outside the drawn window are marked as triangles on the frame. Single seed.

Table 4: Screening rates and derived precision at the label threshold τ and the full training-set size, with recovery, the false-positive rate and the intervals as defined in Appendix A.8. The third block is the precision the two rates imply at 5% prevalence. Brackets: Wilson intervals, propagated through the derivation for the precision block. Read the FPR entries as upper bounds, so only the recovery block marks best (bold) and second best (underlined). *A zero-count FPR gives a precision point of 1.00, so the interval is the number to read. †Phonon peak’s ID test set holds 31 structures, too few to resolve its FPR.

Task	Orb-v3			CFID	SOAP
	Scratch	Finetune	Probe	Probe	Probe
<i>Recovery (TPR, %), OOS test set</i>					
Bulk modulus	61.7 [59.1, 64.3]	68.5 [65.1, 71.8]	73.7 [71.5, 75.8]	48.4 [45.1, 51.7]	12.0 [9.7, 14.8]
Debye temperature	45.9 [42.0, 49.9]	74.6 [71.9, 77.1]	<u>66.0 [62.7, 69.1]</u>	36.6 [33.8, 39.6]	34.3 [31.4, 37.3]
Thermal conductivity	2.3 [1.5, 3.4]	37.0 [34.4, 39.7]	39.7 [36.6, 42.8]	22.9 [20.2, 25.9]	7.6 [6.3, 9.2]
Band gap	9.3 [8.7, 10.0]	76.9 [76.1, 77.6]	56.3 [55.0, 57.6]	2.9 [2.6, 3.2]	6.1 [5.7, 6.4]
Phonon peak	8.5 [4.3, 16.0]	9.5 [6.2, 14.3]	<u>55.0 [48.5, 61.4]</u>	20.1 [15.5, 25.6]	59.8 [52.8, 66.4]
Critical temperature	5.1 [3.4, 7.5]	<u>17.5 [15.4, 19.8]</u>	49.1 [46.6, 51.6]	0.2 [0.0, 0.6]	1.6 [1.0, 2.6]
Refractive index	7.4 [5.3, 10.3]	<u>8.5 [6.8, 10.7]</u>	29.8 [26.9, 32.9]	1.8 [0.9, 3.7]	1.7 [1.0, 2.7]
<i>False-positive rate (%), ID test set</i>					
Bulk modulus	0.36 [0.14, 0.94]	0.12 [0.02, 0.70]	0.36 [0.14, 0.94]	0.36 [0.12, 1.08]	0.12 [0.02, 0.70]
Debye temperature	0.0 [0.0, 0.70]	0.0 [0.0, 0.70]	0.0 [0.0, 0.70]	0.0 [0.0, 0.70]	0.7 [0.3, 1.8]
Thermal conductivity	0.0 [0.0, 0.58]	1.2 [0.6, 2.3]	1.8 [0.9, 3.4]	0.39 [0.11, 1.35]	1.2 [0.6, 2.3]
Band gap	0.09 [0.04, 0.18]	0.08 [0.04, 0.15]	0.20 [0.13, 0.31]	0.09 [0.04, 0.18]	0.28 [0.17, 0.45]
Phonon peak†	0.0 [0.0, 3.1]	0.0 [0.0, 3.1]	3.2 [1.2, 8.1]	3.2 [1.2, 8.1]	3.2 [1.2, 8.1]
Critical temperature	0.0 [0.0, 0.48]	0.0 [0.0, 0.48]	0.49 [0.19, 1.26]	0.0 [0.0, 0.48]	0.32 [0.09, 1.11]
Refractive index	0.0 [0.0, 0.82]	0.0 [0.0, 0.82]	0.0 [0.0, 0.82]	0.8 [0.3, 2.2]	0.0 [0.0, 0.82]
<i>Precision at 5% prevalence, derived*</i>					
Bulk modulus	0.90 [0.77, 0.96]	0.97 [0.83, 0.99]	0.91 [0.80, 0.97]	0.88 [0.69, 0.96]	0.84 [0.42, 0.97]
Debye temperature	1.00 [0.76, 1.00]	1.00 [0.84, 1.00]	1.00 [0.83, 1.00]	1.00 [0.72, 1.00]	0.72 [0.47, 0.88]
Thermal conductivity	1.00 [0.12, 1.00]	0.62 [0.44, 0.78]	0.54 [0.36, 0.71]	0.75 [0.44, 0.92]	0.25 [0.12, 0.45]
Band gap	0.85 [0.71, 0.93]	0.98 [0.96, 0.99]	0.94 [0.90, 0.96]	0.63 [0.44, 0.79]	0.54 [0.40, 0.67]
Phonon peak†	1.00 [0.07, 1.00]	1.00 [0.10, 1.00]	0.47 [0.24, 0.72]	0.25 [0.09, 0.52]	0.49 [0.26, 0.74]
Critical temperature	1.00 [0.27, 1.00]	1.00 [0.63, 1.00]	0.84 [0.66, 0.94]	1.00 [0.01, 1.00]	0.21 [0.05, 0.59]
Refractive index	1.00 [0.25, 1.00]	1.00 [0.30, 1.00]	1.00 [0.63, 1.00]	0.10 [0.02, 0.37]	1.00 [0.06, 1.00]

tasks from a thousand labels at the latest. Critical temperature has a larger training set than two of the three where finetuning overtakes, and the probe still leads finetuning there.

The label saving, the factor by which finetuning needs fewer labels to reach the best from-scratch error, is present on the ID test set too, at a median $2\times$ (OOS the median saving is $3.3\times$, Section 3.1). Every saving is quantized to the measured training-set sizes and is a lower bound. On band gap that ID saving is $9.6\times$ against the $318\times$ OOS.

On the OOS set the probe has the steepest median exponent, $b = -0.16$ against -0.13 finetuned and -0.10 from scratch, while in distribution finetuning descends fastest, at -0.26 . The fits are tight, at a median R^2 of 0.94 over the 42 fits (three methods, seven tasks, two test sets).

Probed representations. On all seven tasks and both test sets (Fig. 7, extending Fig. 1e–f) the ordering holds without a crossing, the MLIP embeddings leading both descriptors at every measured size apart from two 10-label points inside the seed spread.

The MLIP embeddings descend faster in each of the 28 comparisons of task, descriptor, and test set (Section 3.1). The difference exceeds its standard error in 27, the exception being thermal conductivity against CFID OOS. These fits are as tight as those of the GNN training methods, at a median R^2 of 0.93 over the 42 fits.

B.3 Embedding-space analysis

The maps of Section 3.2 extend to all seven tasks and all five representations (Fig. 8). The embedding vector for the three GNN training methods is the mean of the backbone’s final per-atom features, while the probe applies r^{atom} before the pool (App. A.5), and its column shows the mean of $\{h_i\}$

Table 5: MAE in report units, both test sets, at the full training-set size. Each task is in its own unit, so entries are read along a row and not down a column. Intervals: ± 1 s.e. The best method per task and test set is bold, the second best underlined.

Task	Unit	Orb-v3			CFID	SOAP
		Scratch	Finetune	Probe	Probe	Probe
<i>ID test set</i>						
Bulk modulus	log GPa	0.060 ± 0.005	0.055 ± 0.006	0.060 ± 0.005	0.089 ± 0.006	0.089 ± 0.006
Debye temperature	log K	0.032 ± 0.004	0.021 ± 0.004	<u>0.025 ± 0.003</u>	0.041 ± 0.005	0.054 ± 0.006
Thermal conductivity	log W/mK	0.16 ± 0.01	0.12 ± 0.01	<u>0.13 ± 0.01</u>	0.18 ± 0.01	0.25 ± 0.03
Band gap	eV	0.152 ± 0.006	0.081 ± 0.004	<u>0.112 ± 0.004</u>	0.345 ± 0.011	0.302 ± 0.010
Phonon peak	cm^{-1}	45 ± 9	28 ± 9	<u>36 ± 9</u>	50 ± 11	100 ± 21
Critical temperature	K	0.82 ± 0.09	0.52 ± 0.07	<u>0.64 ± 0.08</u>	1.34 ± 0.12	1.35 ± 0.23
Refractive index	—	0.10 ± 0.03	0.04 ± 0.02	<u>0.06 ± 0.01</u>	0.25 ± 0.06	0.20 ± 0.02
<i>OOS test set</i>						
Bulk modulus	log GPa	0.069 ± 0.005	<u>0.050 ± 0.005</u>	0.048 ± 0.004	0.087 ± 0.005	0.144 ± 0.008
Debye temperature	log K	0.148 ± 0.007	0.091 ± 0.005	<u>0.110 ± 0.005</u>	0.170 ± 0.012	0.199 ± 0.009
Thermal conductivity	log W/mK	0.98 ± 0.06	0.72 ± 0.06	<u>0.76 ± 0.06</u>	0.84 ± 0.05	1.10 ± 0.06
Band gap	eV	1.80 ± 0.12	0.594 ± 0.012	<u>0.772 ± 0.010</u>	2.243 ± 0.032	2.113 ± 0.024
Phonon peak	cm^{-1}	1117 ± 94	1010 ± 97	751 ± 91	1000 ± 120	1650 ± 470
Critical temperature	K	7.99 ± 0.33	<u>6.40 ± 0.30</u>	4.77 ± 0.29	10.24 ± 0.31	9.65 ± 0.33
Refractive index	—	4.09 ± 0.49	<u>3.77 ± 0.50</u>	3.69 ± 0.50	4.12 ± 0.49	4.34 ± 0.49

Table 6: Spearman ρ on both test sets at the full training-set size. A negative entry ranks the OOS set backward. Intervals: ± 1 s.e. The best method per task and test set is bold, the second best underlined.

Task	Orb-v3			CFID	SOAP
	Scratch	Finetune	Probe	Probe	Probe
<i>ID test set</i>					
Bulk modulus	0.96 $\pm$ 0.01	0.97 $\pm$ 0.01	<u>0.97 $\pm$ 0.01</u>	0.94 $\pm$ 0.01	0.94 $\pm$ 0.01
Debye temperature	0.98 $\pm$ 0.01	0.99 $\pm$ 0.00	<u>0.98 $\pm$ 0.00</u>	0.96 $\pm$ 0.01	0.94 $\pm$ 0.01
Thermal conductivity	0.89 $\pm$ 0.03	0.93 $\pm$ 0.02	<u>0.93 $\pm$ 0.02</u>	0.87 $\pm$ 0.03	0.81 $\pm$ 0.04
Band gap	0.90 $\pm$ 0.01	0.92 $\pm$ 0.01	<u>0.91 $\pm$ 0.01</u>	0.82 $\pm$ 0.01	0.84 $\pm$ 0.01
Phonon peak	0.99 $\pm$ 0.01	0.99 $\pm$ 0.01	<u>0.99 $\pm$ 0.01</u>	0.98 $\pm$ 0.02	0.94 $\pm$ 0.04
Critical temperature	0.85 $\pm$ 0.02	0.94 $\pm$ 0.01	<u>0.91 $\pm$ 0.02</u>	0.61 $\pm$ 0.05	0.73 $\pm$ 0.04
Refractive index	0.94 $\pm$ 0.05	<u>0.96 $\pm$ 0.04</u>	0.96 $\pm$ 0.03	0.88 $\pm$ 0.05	0.90 $\pm$ 0.03
<i>OOS test set</i>					
Bulk modulus	0.60 $\pm$ 0.04	0.69 $\pm$ 0.03	<u>0.66 $\pm$ 0.03</u>	0.48 $\pm$ 0.04	0.12 $\pm$ 0.05
Debye temperature	0.48 $\pm$ 0.07	0.84 $\pm$ 0.02	<u>0.68 $\pm$ 0.04</u>	0.46 $\pm$ 0.06	0.15 $\pm$ 0.07
Thermal conductivity	-0.19 $\pm$ 0.05	0.39 $\pm$ 0.06	0.19 $\pm$ 0.06	<u>0.25 $\pm$ 0.06</u>	-0.06 $\pm$ 0.06
Band gap	-0.43 $\pm$ 0.02	0.75 $\pm$ 0.01	<u>0.51 $\pm$ 0.02</u>	-0.25 $\pm$ 0.02	-0.22 $\pm$ 0.01
Phonon peak	-0.14 $\pm$ 0.14	-0.44 $\pm$ 0.11	-0.09 $\pm$ 0.14	<u>0.08 $\pm$ 0.16</u>	0.10 $\pm$ 0.15
Critical temperature	-0.22 $\pm$ 0.05	<u>-0.05 $\pm$ 0.05</u>	0.20 $\pm$ 0.05	-0.19 $\pm$ 0.05	-0.17 $\pm$ 0.05
Refractive index	-0.23 $\pm$ 0.06	-0.49 $\pm$ 0.06	-0.35 $\pm$ 0.06	-0.38 $\pm$ 0.06	<u>-0.24 $\pm$ 0.06</u>

itself. For the descriptors it is the vector their predictors receive (App. A.4), and projections are fitted on $\mathcal{D}_{\text{tr}}$ throughout.

The plane of the first two components keeps 33% of the in-distribution variance for the raw MLIP embeddings at the median and 31% for CFID, but only 1% for SOAP. The standardization of Appendix A.4 leaves SOAP with a flat spectrum, so its panel is a thin slice of a near-isotropic cloud. The raw MLIP embeddings are never trained on the property, yet their leading component orders it in distribution (Section 3.2), while on the OOS set its rank correlation is +0.04 at the median against +0.05 on CFID and -0.08 on SOAP.

The alignment finetuning produces extends to the OOS set only in part. The same correlation falls to +0.27 at the median on the OOS set, from +0.75 on Debye temperature to -0.45 on phonon peak, and the plane keeps 45% of the in-distribution variance against 15% for the OOS set. Of the four shortest-tailed tasks three are ordered outside the training range and thermal conductivity is not, as Section 3.2 reads off the maps.

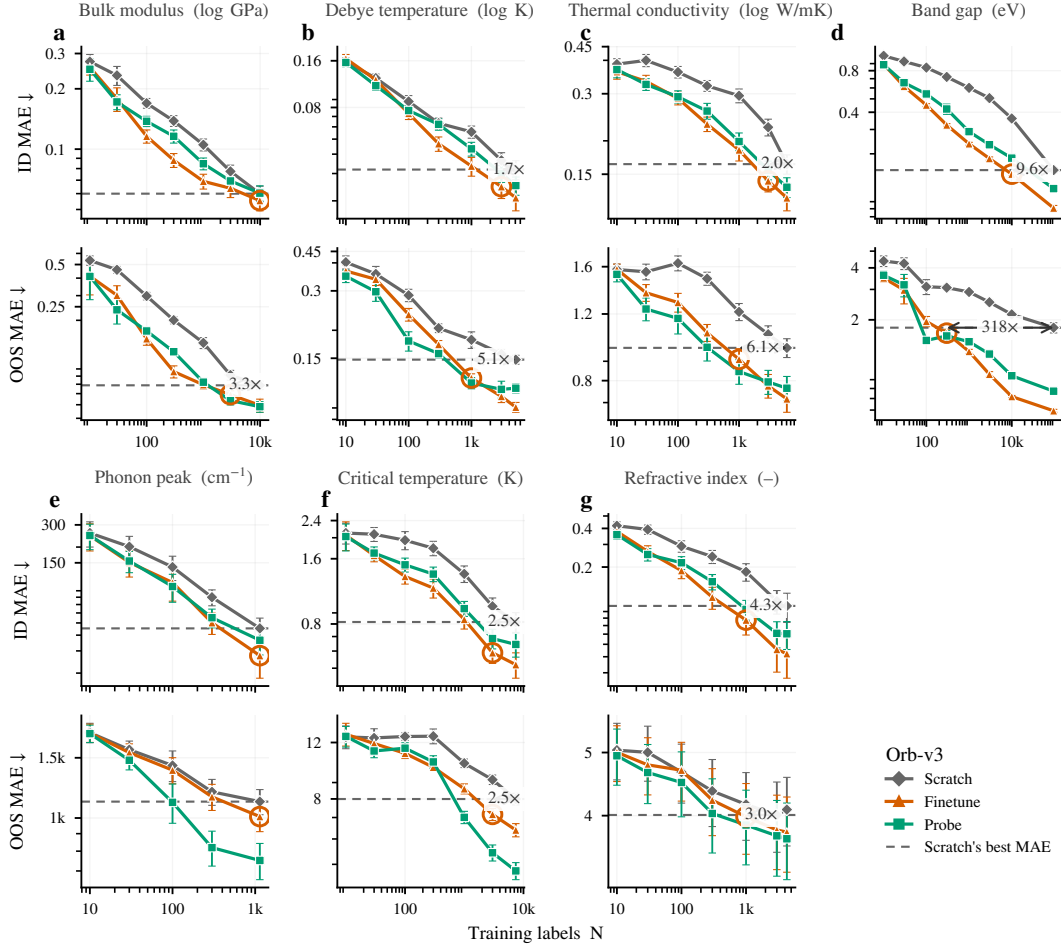


Figure 6: Learning curves of the three GNN training methods on all seven tasks, one panel per task (a–g) with the ID curve above the OOS curve, MAE in each task’s unit. Annotations give the label saving over the from-scratch model per test set. Error bars: ± 1 s.e.

From-scratch training, the same objective without MLIP pretraining, aligns the property in distribution almost as well as finetuning (0.92 against 0.94) but not out of support (-0.07 against $+0.27$). Both train on the same labels, so the OOS ordering comes from the MLIP pretraining alone.

B.4 Transductive methods

Each comparison below pairs BLT against the matched MLP baseline of Appendix A.6 on one representation at the full training-set size, so the bilinear form is the only difference in the pair.

Raw and finetuned representations. Across the three raw representations the swap moves OOS error in transduction’s favor, lowering OOS MAE on five of seven tasks on each of them at medians of 4% to 6% (Table 7). The size matches the error reduction Segal et al. [12] report against their strongest model, a few percent of OOS MAE. Single task-and-representation entries swing far past the medians in both directions, from a 52% reduction to a 107% increase in OOS MAE. The gains cost ID error at the median on CFID and on the raw MLIP embeddings, by 14% and 21% (upper block). On SOAP the ID changes go both ways, at a median absolute change of about 6%, and on Debye temperature the swap lowers error on both test sets, by 18% in distribution and 52% out of support. On every representation both predictors are selected on $\mathcal{D}_{\text{val}}^{\text{id}}$, so the trade is not a selection artifact.

The effect of the swap on OOS recovery differs by representation. On CFID the swap raises OOS recovery on all seven tasks, by a median 5.1 points and by 14.7 on refractive index (sign test

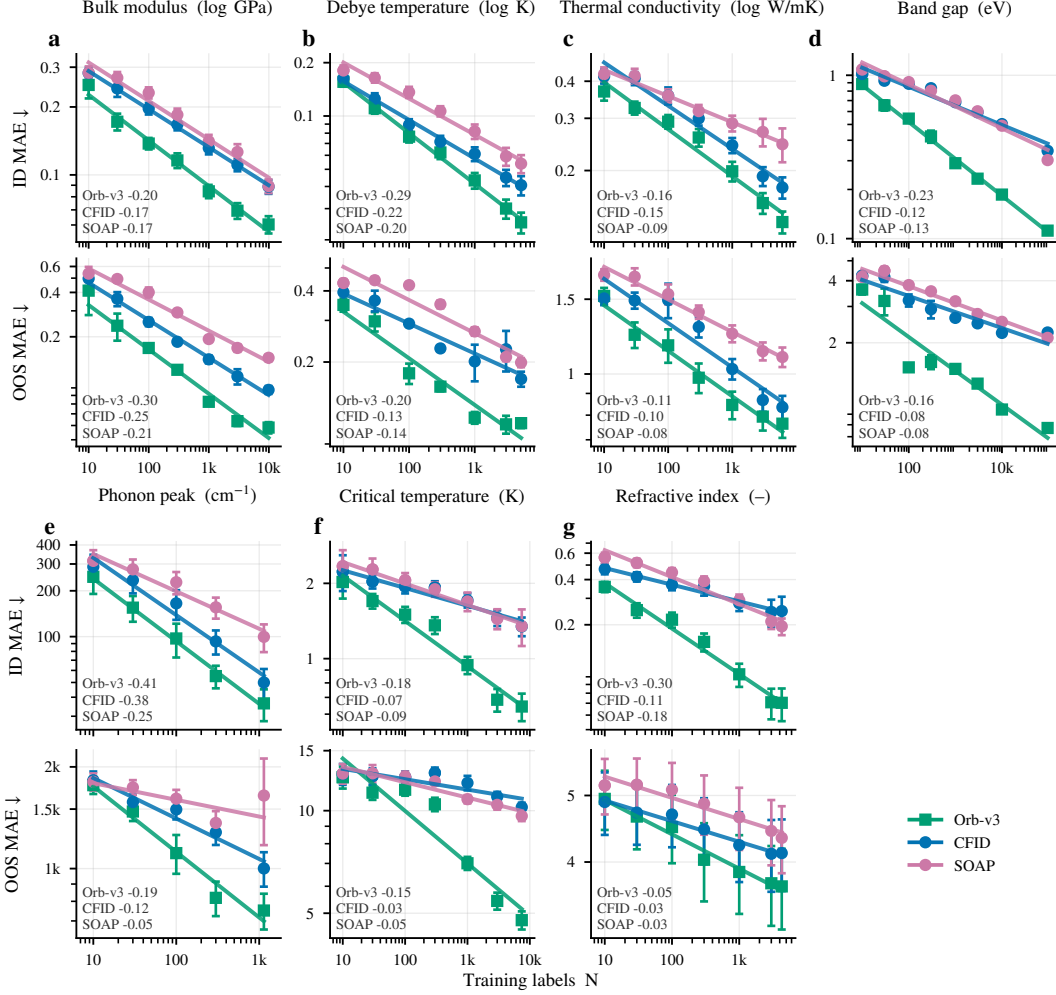


Figure 7: Log-log fits of probed error against the number of training labels for the three representations, one panel per task (a–g), ID above OOS, with the fitted exponents b per fit. Hollow markers flag points at or above the best constant predictor’s error. Error bars: ± 1 s.e.

$p = 0.016$), while on SOAP and on the raw embeddings it lowers recovery on five of the seven, by a median 3.2 points on each. Median OOS error falls on all three regardless.

The finetuned embeddings have the lowest ID error of the four representations and are the one representation shaped by the task itself, so transduction is expected to work best there, yet BLT does not improve on the matched MLP on any task by more than a few percent. The OOS MAE shifts stay within 2.4% in either direction, the ID shifts within 3.7%.

Degeneracy under an affine label. Finetuning drives the per-material embedding Z (App. A.5) toward a representation in which the label is close to affine (Section 3.2), and the readout explains why. Above Z lies only a two-layer MLP, small and nearly linear, so driving the training loss down forces the label toward linearity in Z itself, which leaves the transductive construction nothing to do.

Proposition 1 (Anchor invariance under an affine label). *Let the label be affine in the embedding on the training set, $y_i = w^\top Z_i + c$ for every $i \in \mathcal{D}_{\text{tr}}$, and let $T_0(\delta, Z_i) = w^\top \delta + w^\top Z_i + c$, the affine increment plus the anchor’s label. Then for any query embedding Z and any anchor $i \in \mathcal{D}_{\text{tr}}$, $T_0(Z - Z_i, Z_i) = w^\top (Z - Z_i) + y_i = w^\top Z + c$, so T_0 fits every pair of $\mathcal{D}_{\text{tr}}$ with zero error, and it is the bilinear form of Eq. 2 with two-dimensional trunks $u(\delta) = (w^\top \delta, 1)$ and $v(Z_i) = (1, w^\top Z_i + c)$. Let $T(\delta, Z_i)$ be any predictor of a difference and an anchor, applied at a query as $T(Z - Z_i, Z_i)$, and let $\eta = \sup_{Z, i} |T - T_0|$ be the largest gap between the two over the test queries and over every anchor the rule of Eq. 3 selects from. Then any two anchors give predictions within 2η of each other.*

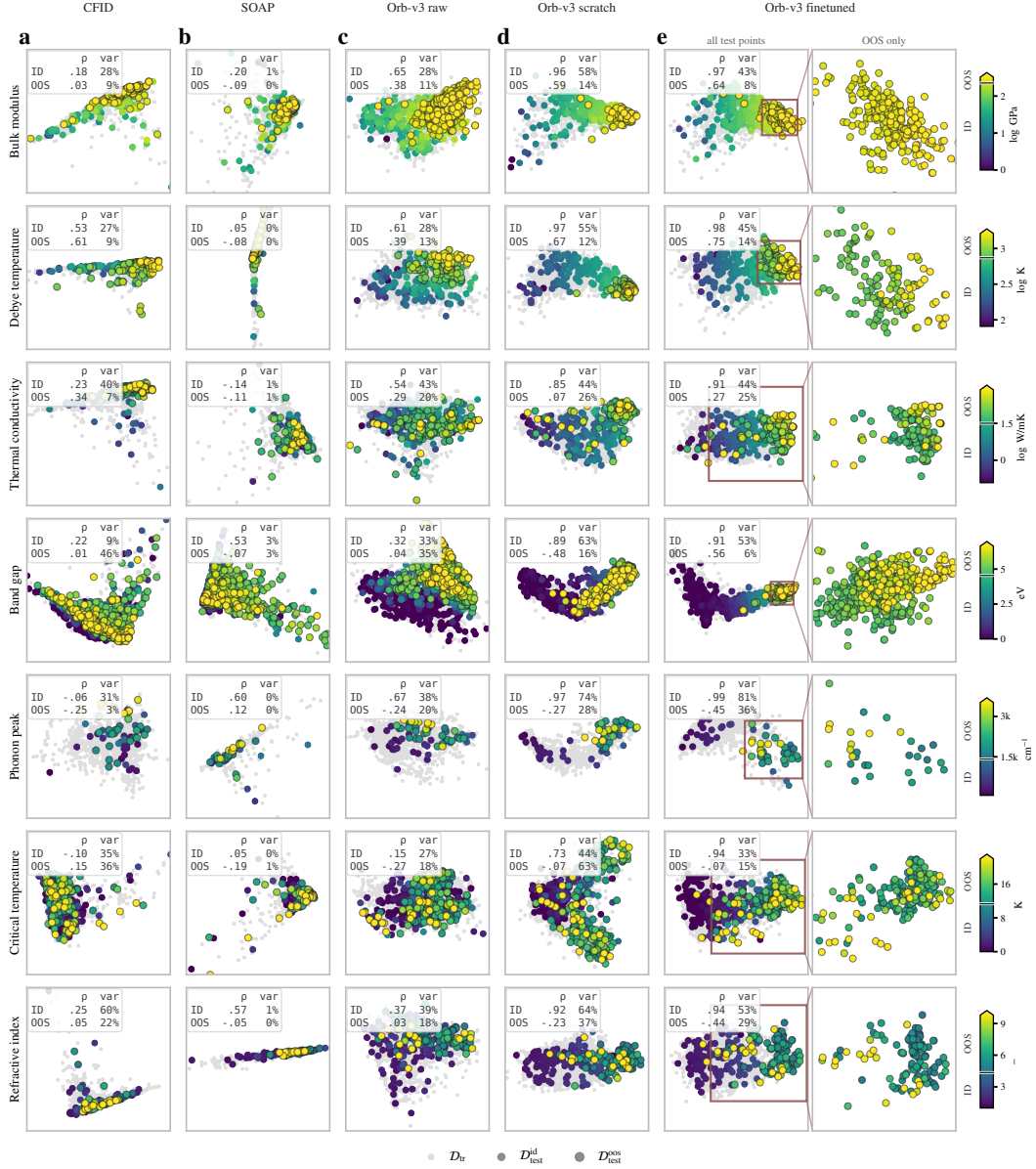


Figure 8: PCA maps as in Figure 2 for all seven tasks and all five representations. The horizontal axis is the leading principal component and the vertical the second, both fitted on the training set, with signs oriented so higher labels point toward positive. Each panel's box gives, per test set, the Spearman correlation between the leading component and the label and the share of that test set's own variance the plane keeps, both computed on the full test set while the panel shows the middle 99% of the points. Only the finetuned column carries an OOD-only enlargement, since its OOD set collapses too tightly to read at full scale. Single seed.

Table 7: Change in MAE from replacing the matched MLP predictor with bilinear transduction, as a percentage of the MLP’s MAE at the full training-set size, on the ID (top) and OOS (bottom) test sets. Negative favors transduction. Intervals: ± 1 s.e. per Appendix A.8, a paired bootstrap over the shared test samples plus the seed term. The last row of each block is the median. The shaded column is the finetuned embeddings, where the swap moves the median by under 1% on either test set.

	CFID		SOAP		Orb-v3	
Task	Probe		Probe		Probe	Finetune
<i>Change in ID MAE (%)</i>						
Bulk modulus	+23.1 ± 7.7		+13.0 ± 12.6		+24.3 ± 8.3	+1.6 ± 1.2
Debye temperature	+2.3 ± 6.2		−17.6 ± 7.4		+10.7 ± 5.9	−1.1 ± 1.5
Thermal conductivity	+5.4 ± 5.7		−6.4 ± 10.1		+20.2 ± 6.3	+0.7 ± 0.5
Band gap	+29.3 ± 3.2		+33.4 ± 2.2		+67.9 ± 6.6	+3.7 ± 0.4
Phonon peak	+14.1 ± 13.8		+0.2 ± 14.7		+20.7 ± 32.1	+2.8 ± 5.6
Critical temperature	+4.4 ± 4.2		−1.0 ± 8.8		−2.2 ± 7.6	−1.3 ± 0.9
Refractive index	+16.3 ± 13.1		+4.5 ± 8.4		+24.3 ± 14.4	+0.1 ± 1.7
Median	+14.1		+0.2		+20.7	+0.7
<i>Change in OOS MAE (%)</i>						
Bulk modulus	−6.2 ± 3.1		−12.5 ± 2.4		+3.0 ± 3.0	+1.1 ± 1.3
Debye temperature	+106.7 ± 70.0		−51.6 ± 4.9		−6.1 ± 3.1	−2.4 ± 1.1
Thermal conductivity	−5.7 ± 6.3		+2.1 ± 1.8		−7.6 ± 1.7	+0.4 ± 0.2
Band gap	−24.8 ± 0.8		−6.4 ± 1.0		−8.1 ± 1.4	+1.7 ± 0.8
Phonon peak	+33.2 ± 19.2		−7.1 ± 10.4		−2.6 ± 1.2	+0.9 ± 0.6
Critical temperature	−2.6 ± 4.9		+6.6 ± 4.6		+4.9 ± 4.2	+0.1 ± 0.4
Refractive index	−4.1 ± 0.8		−3.2 ± 1.7		−6.3 ± 1.3	+0.1 ± 0.2
Median	−4.1		−6.4		−6.1	+0.4

Furthermore, for any predictor r that reads Z alone, with $\eta' = \sup_Z |r(Z) - w^\top Z - c|$ over the same queries, $|T(Z - Z_i, Z_i) - r(Z)| \leq \eta + \eta'$ at every query.

Proof. The first equality of the identity is the hypothesis at the anchor, the second cancels Z_i . For a pair (i, j) of $\mathcal{D}_{\text{tr}}$, the identity at $Z = Z_j$ and the hypothesis at j give $T_0(Z_j - Z_i, Z_i) = w^\top Z_j + c = y_j$, and $\langle u(\delta), v(Z_i) \rangle = T_0(\delta, Z_i)$. By the identity and the definition of η , every prediction lies within η of the one value $w^\top Z + c$, so two predictions lie within 2η of each other. The last bound is the triangle inequality through $w^\top Z + c$. $\square$

The last bound makes the two predictors agree wherever both stay near the affine map, and says nothing about whether either is right. Affinity is assumed over the training set alone, so beyond the training range both predictors extend the same affine map and inherit the same error in it. The MLP gain of Table 8 indicates the size of η' in distribution, and the anchor sweep below measures the spread that the bound caps at 2η . The guarantee of Netanyahu et al. [11] does not cover this setting, since their Theorem 1 assumes a square loss and an averaged anchor rule that we do not deploy (App. A.6).

Two in-distribution statistics, the linearization conditions of Table 8, measure how close a representation comes to affinity. Linear decodability, the MAD:MAE of the linear fit of Appendix A.7, measures how close to affine the label is in Z , and the trained MLP’s gain over that fit indicates how much of the label the linear map leaves unexplained. Neither statistic alone is decisive: SOAP has the smallest MLP gain of the four yet the decodability furthest from affine, and BLT still swings single tasks by 33% ID there. Band gap has the finetuned row’s median decodability yet keeps a 43% MLP gain, and on the finetuned embeddings it is the task where BLT and the MLP part furthest, the largest of the finetuned shifts that resolve. The finetuned embeddings are the one representation that meets both conditions and the one where BLT degenerates.

We measure the trained trunks u and v of Eq. 2 directly, and they behave as the argument predicts. Each responds to the anchor on its own, yet the two responses anti-correlate at -0.98 and -0.99 (bulk modulus and band gap) on the OOS test set and cancel, where the two terms of T_0 in Proposition 1 give exactly -1 . Sweeping the anchor across the whole label range moves the OOS prediction by 2% and 11% of the model’s own error on band gap and bulk modulus (the two tasks measured).

Table 8: The linearization conditions and the size of the BLT effect, per representation, at the full training-set size, both conditions as defined in the text. Decodability and MLP gain are medians over the seven tasks, higher decodability being better. The largest shift is the largest single-task MAE change from swapping the matched MLP for BLT, on each test set. The shaded row is the finetuned embeddings, the one representation that meets both conditions and the one where BLT degenerates.

Representation	Linear decodability (MAD:MAE)	MLP gain (%)	Largest BLT shift (%)	
			ID	OOS
SOAP	1.5	6	33	52
CFID	1.6	19	29	107
Orb-v3 raw	3.3	34	68	8
Orb-v3 finetuned	6.9	8	3.7	2.4

C Additional experiments

C.1 Unseen materials

Every result above is measured on structures the backbone’s pretraining corpus already holds, with only the labels new (App. A.2). No decontaminated dataset with diverse property labels exists to repeat the comparison on unseen structures. This subsection repeats the comparison in the scope that does exist, with a second Orb-v3 checkpoint, pretrained on a different corpus, and two pools outside that corpus, one of them excluded by the corpus’s own documentation.

Setup. The comparison runs on three band gap tasks, the MP band gap task and two extension tasks. The WBM search [48] substitutes chemically similar elements into Materials Project prototypes in successive rounds, five in the release distributed with Matbench Discovery [49]. WBM-1 is the search’s first substitution round, with PBE band gaps (61k crystals, 55.3k training and 3,073 OOS). QMOF band gap holds relaxed metal-organic frameworks [50, 51] (20k, 18.3k training and 1,019 OOS). WBM-1’s training labels are 80% exactly zero against about half on the MP band gap task of the main results. QMOF’s structures are metal-organic frameworks of up to 300 atoms per primitive cell, a shift in structure as well as in label, since the pretraining corpus holds inorganic crystals alone.

OMat24 is the one pretraining corpus that documents the exclusion of a labeled pool: generated from relaxed Alexandria structures, it excludes every structure whose parent carries a WBM prototype label (5.3M structures in all) [9]. We therefore switch to `orb-v3-direct-inf-omat-20250404`, identical in architecture to the MPA checkpoint and pretrained on OMat24 instead of MPtrj and Alexandria. Each of OMat24’s 9.4M `rattled-relax` records ships the prototype label of its Alexandria parent, a Wyckoff-position key our overlap audit computes alongside the geometric rule of Appendix A.2. The shipped labels confirm the exclusion of WBM-1 up to one label match in the training split (Table 9).

The training setup of Appendix A.7 is held fixed, the descriptor reductions are refit per task as in Appendix A.4, and each checkpoint is calibrated on its own learning-rate grid, where the optimum moves with the corpus in four of six task-and-method combinations (three tasks, two transfer methods). The checkpoint change itself costs accuracy: on MP band gap the OMat24 checkpoint makes $1.2\times$ the MPA checkpoint’s OOS error finetuned and $1.3\times$ probed, at matching ID error.

Results. On WBM-1, the pool OMat24 excludes, the decontaminated checkpoint reproduces the method ordering of Section 3.1 (Fig. 9). Both transfer methods beat the from-scratch model and both descriptor probes on error in distribution and on the OOS set, and on recovery. They also order the OOS set forwards, at Spearman ρ of $+0.71$ finetuned and $+0.22$ probed, where the descriptor probes and the from-scratch model rank it backwards. The MLIP embeddings stay ahead of both descriptors as they do in Section 3.1, the better descriptor making about $2\times$ the MLIP probe’s OOS error on either extension task, and no method on either task exceeds a 0.2% false-positive rate. On QMOF, outside both corpora, the error ordering holds with compressed margins, and on screening the probe recovers $6.9\times$ what finetuning does. The comparison the paper draws therefore does not depend on the overlap.

Table 9: Share of structures unseen per split under MPtrj and Alexandria, by the geometric rule of Appendix A.2, and under OMat24, with the unseen count in parentheses. [†]OMat24 is audited for WBM-1 alone, by the parent prototype labels it ships, with one label match in the training split. The other two rows of the OMat24 block are not audited and count a structure as unseen if its prototype label is absent from Alexandria, which OMat24 is generated from. Those shares are lower bounds, since OMat24 holds a subset of Alexandria, loosened by the matches the label rule misses (5 points on WBM-1 against the geometric rule).

Task	MPtrj and Alexandria			OMat24 [†]		
	$\mathcal{D}_{\text{tr}}$	$\mathcal{D}_{\text{test}}^{\text{id}}$	$\mathcal{D}_{\text{test}}^{\text{oos}}$	$\mathcal{D}_{\text{tr}}$	$\mathcal{D}_{\text{test}}^{\text{id}}$	$\mathcal{D}_{\text{test}}^{\text{oos}}$
MP band gap	<1% (433)	<1% (6)	<1% (50)	22% (20,552)	22% (575)	25% (1,342)
WBM-1 band gap	1% (302)	1% (9)	1% (21)	100% (55,319)	100% (1,537)	100% (3,073)
QMOF band gap	99% (18,239)	100% (507)	96% (980)	100% (18,288)	100% (507)	98% (997)

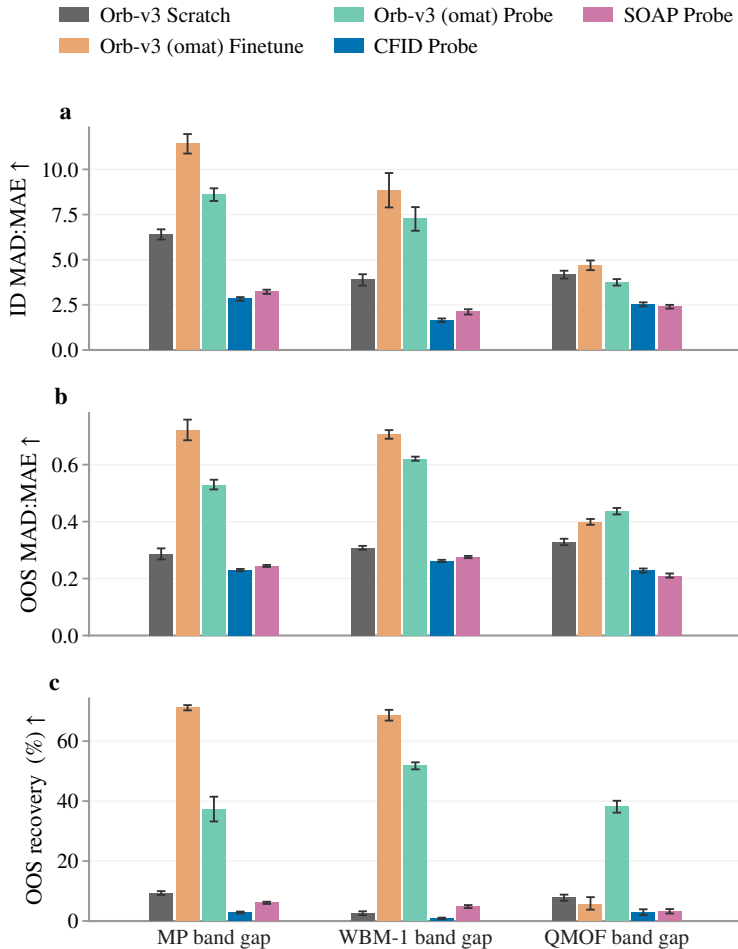


Figure 9: Three band gap tasks at the full training-set size, for the two transfer methods on the OMat24 checkpoint against the from-scratch model and the two descriptor probes. Error bars: ± 1 s.e., Wilson in c.

C.2 Other MLIP backbones

The main paper shows results for one backbone. This subsection repeats the comparison of the three GNN training methods on band gap across five pretrained backbones, to test whether the ordering the paper reports is specific to Orb-v3.

Table 10: The five backbones by architecture. Forces is the route each released checkpoint deploys, width is the per-atom feature width the probe reads, and size counts the backbone alone. The architecture values are read off the loaded checkpoints, and Released is the month each checkpoint’s weights became public.

Backbone	Released	Rotation	Forces	Layers	Width	Size
Orb-v3 [26]	2025-04	augmented	direct	5	256	24M
MACE [52]	2024-12	equivariant	cons.	2	256	9.1M
UMA [10]	2026-03	equivariant	cons.	4	128	291M
PET [53]	2026-02	augmented	cons.	3	1024	26M
DPA-3 [54]	2025-12	invariant	cons.	24	128	4.7M

Setup. The comparison keeps Orb-v3 (direct-inf-mpa) and adds MACE-MPA-0-medium, UMA-S 1.2, PET-MAD-1.5-S, and DPA-3.2-5M (Table 10). In the Rotation column of Table 10, an equivariant backbone constrains its features to transform correctly under rotation, an invariant one admits only rotation-invariant quantities such as distances and angles, and an augmented one learns approximate invariance through rotation augmentation during pretraining. The Forces column is the conservative or direct route of Appendix A.3. The probe keeps its architecture, with the first layer sized to each backbone’s feature width, and finetuning and from-scratch training keep their readouts (App. A.5). Every other element of the training setup of Appendix A.7 is held fixed, with three seeds per backbone-and-method combination. Each backbone is calibrated on its own learning-rate grid. Compute limits the training set to 10k labels, against band gap’s full training set of 95.5k, so no bar in Figure 10 is comparable with the results of Section 3.1 at the full training set. The two classical descriptors of Section 2 are probed on the same 10k labels with their reported predictor (App. A.5).

MACE trains on the same MPTrj and Alexandria lineage as Orb-v3 (App. A.3), UMA-S and DPA-3 train multi-task across corpora spanning materials, catalysis, and molecules, and PET-MAD-1.5 is finetuned from an OMat24-pretrained checkpoint onto the all-electron r^2 SCAN MAD (Massive Atomic Diversity) set, so the five share little beyond pretraining itself.

Results. Both transfer methods beat training from scratch on every backbone, in distribution and on the OOS set (Fig. 10), as they do on every task in Section 3.1. Finetuning cuts OOS error against training from scratch by a median 59%, from 49% on MACE to 77% on DPA-3. Every from-scratch model ranks the OOS set backwards, at a median Spearman ρ of -0.12 and below zero on all five, as Orb-v3’s does on band gap at the full training set (-0.43 , Table 6). The finetuned models reach a median $+0.59$ (0.21 to 0.64). No backbone and method combination exceeds a 0.3% false-positive rate on the ID test set. MACE gains the least from finetuning, 18 percentage points of recovery over its from-scratch model against 37 to 55 on the other four. Both descriptors trail both transfer methods on OOS error and recovery on every backbone. They recover 0.26% (CFID) and 2.4% (SOAP) of the OOS set against 26% to 49% for the MLIP probes, and their OOS MAD:MAE (0.23 and 0.20) lies inside the from-scratch range (0.14 to 0.28). Hence the ordering of Section 3.1, embeddings over descriptors over training from scratch, holds on band gap for every backbone.

C.3 End-to-end GNN comparison

We read the from-scratch model as a stand-in for a conventional end-to-end crystal GNN. Yet the from-scratch model carries an architecture built for interatomic potentials, at four to six times the size of a typical crystal GNN, and is trained by us, so we add two published crystal GNNs to the seven-task comparison.

Setup. ALIGNN [46], a line graph network with explicit bond angles (four line-graph and four graph layers), and iComformer [55], an invariant graph transformer (four node-wise and one edge-wise attention layer), are two published supervised crystal property predictors (sizes in the caption of Fig. 11). Both read the same splits as every other method, train on each task’s full training set under the loss, optimizer, schedule, and step budget of Appendix A.7, and carry three seeds. The learning rate is again the only swept hyperparameter, selected on $\mathcal{D}_{\text{val}}^{\text{id}}$ from a grid per architecture and task.

Results. The three networks trained on labels alone behave alike (shaded group, Fig. 11). In distribution they lie within 20% of each other on five of the seven tasks. On the OOS set the published

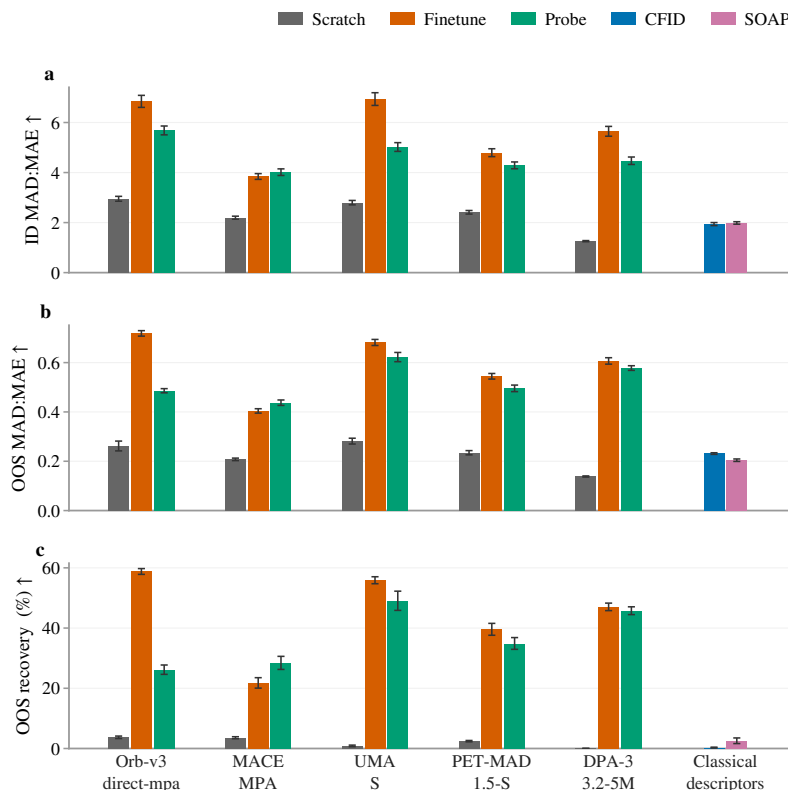


Figure 10: Five pretrained backbones under the three GNN training methods on band gap at a reduced training set of 10k labels, and the two classical descriptors probed on the same labels with their reported predictor. Error bars: ± 1 s.e., Wilson in c.

GNNs are slightly ahead, the better of the two beating the stand-in on six of the seven tasks by 3% to 42% of MAD:MAE and losing on bulk modulus by 18%. The stand-in therefore behaves as an end-to-end crystal GNN does, so the comparisons of Section 3.1 read as comparisons against that class of model. Neither published architecture correctly orders the OOS set, at an OOS Spearman ρ below zero in ten of the fourteen task-and-architecture combinations as the from-scratch model’s is, so training on the task’s labels alone leaves the OOS set mis-ordered whichever network does it.

Over the same seven tasks, the better transfer method beats the better published GNN on every task, cutting OOS error by a median 36% against the 33% it cuts of our own from-scratch model (Fig. 11b). The two medians fall on different tasks, and on bulk modulus, where both GNNs make more OOS error than the from-scratch model, the cut against them is 42% against 30%. The better transfer method recovers more of the OOS set than either GNN on every task, from $1.4\times$ on bulk modulus to $10\times$ on phonon peak, and no GNN exceeds a 0.8% false-positive rate in distribution (Fig. 11c). On band gap we also measure the OOS learning curves, where ALIGNN tracks the from-scratch model to within 3% at every size below the full training set and iComformer stays within 18% of it, ahead at most sizes. At the full 95.5k labels iComformer stays behind the OOS error the probe reaches at 3k.

C.4 Selection rule

In-distribution selection can disagree with an OOS oracle (the pick with the lowest OOS MAE, which no reported run uses), as in domain generalization [29]. Every validation label lies below the threshold τ , so a model is scored in distribution alone, and a ranking obtained there can invert on the OOS set. Table 11 measures the disagreement over the learning-rate grids of Appendix A.7.

The two criteria pick different learning rates in 46 of 63 task-and-method combinations. Over all 63 the ID-selected pick costs a median 1.7% (at most 160%) in OOS error, against a 1.5% median cross-seed spread in the reported runs, and a median 0.2 percentage points of recovery (at most 42).

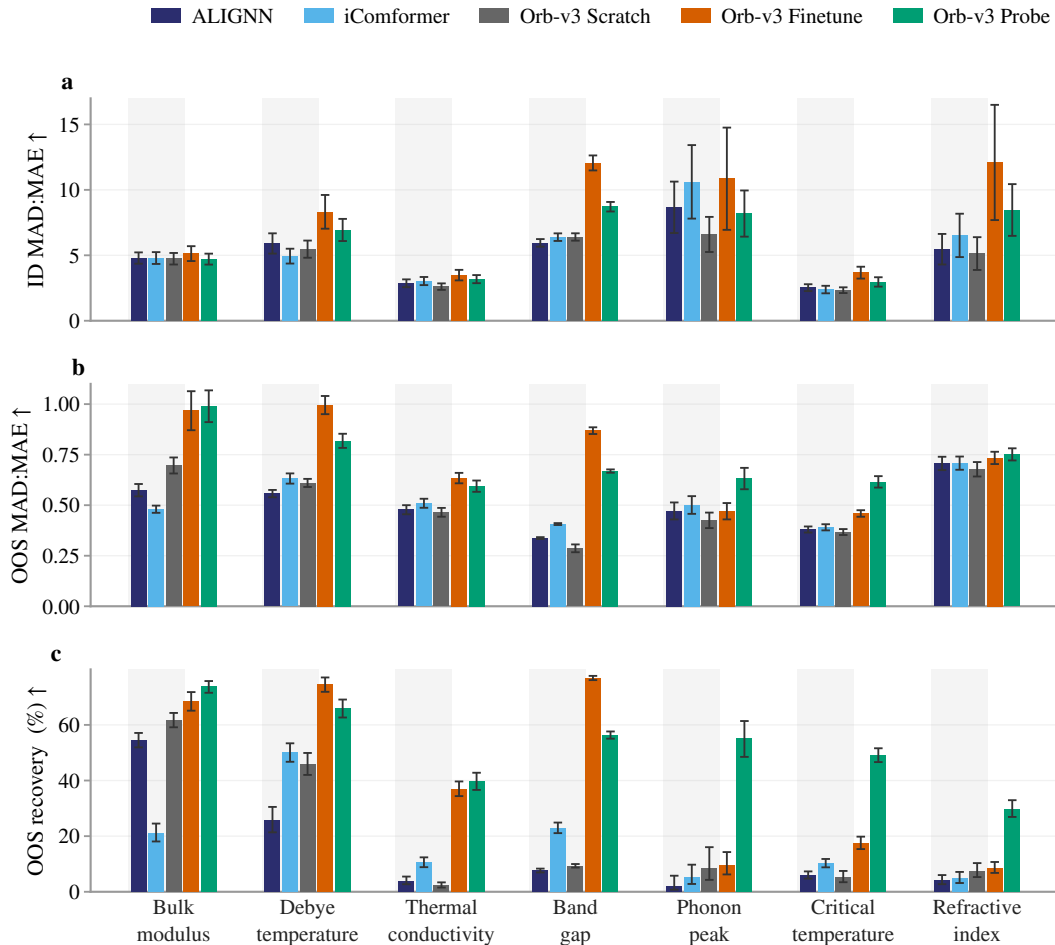


Figure 11: Two published end-to-end crystal GNNs beside the three GNN training methods on all seven tasks at the full training set. The shaded group holds the three networks trained on the task’s labels alone (Orb-v3 from scratch at 24M parameters, ALIGNN at 4.0M, and iComformer at 5.0M). Error bars: ± 1 s.e., Wilson in c.

Table 11: The cost of the ID-validation pick against the pick an OOS MAE oracle would make, in OOS MAE and in OOS recovery, over the learning-rate grids on the seven tasks at the reported predictors. Cells counts the task-and-method combinations, and Differ those where the two picks disagree.

Comparison	Cells	Differ	MAE penalty (%)		Recovery penalty (pp)	
			Median	Max	Median	Max
GNN training	21	14	3.9	15	0.0	13
Descriptor probes	14	12	3.0	160	3.6	42
Bilinear	28	20	1.1	41	0.0	9.5
Overall	63	46	1.7	160	0.2	42

The MAE penalty is an upper bound, since the grid points are single-seed and their best is a maximum over four to six of them, while the recovery penalty, read at the same pick, can be negative.

C.5 Predictor capacity

The reported configuration matches predictor size across representations (App. A.5), and here we vary that size on each raw representation, with the splits and the training setup held fixed, from the linear probe of Appendix A.7 to the trained sizes of Figure 12.

Setup. Every trained probe is an instance of Eq. 1. Medium is the reported configuration of Appendix A.5, and Small is the readout that finetuning and from-scratch training use and the matched MLP baseline of the transduction comparison (App. A.6). Large widens the hidden layers to 1024 (caption of Fig. 12), and the linear probe of Appendix A.7 is fit on the same Z .

Nonlinear probe capacity. Every representation carries four predictor sizes at the full training set (Fig. 12), and the learning curves repeat the comparison at every training-set size. A larger descriptor predictor lowers ID error and leaves OOS error nearly unchanged. Across the learning curves (not shown) the large predictor has a lower ID MAE than the small one on 94 of the 96 task, descriptor, and training-set size combinations (seven tasks, two descriptors, five to eight sizes each), and a lower OOS MAE on 37. The small descriptor predictor also recovers more than the medium one (Fig. 12c), but with more false positives, its ID false-positive rate exceeding the medium predictor’s on 12 of the 14 task-and-descriptor pairs, so its extra recovery comes from predicting high. On the MLIP embeddings the small predictor costs a median 14% ID MAE against the medium one, while ρ , recovery and the false-positive rate are as often ahead as behind. The large readout moves neither experiment.

ID validation selects the large or the medium predictor on all 21 task-and-representation combinations and ranks the linear probe last on 20 of them. Thus the small descriptor predictor’s recovery margin is out of reach in deployment.

The descriptor comparison of Section 3.1 holds at every MLP predictor size. At the small, medium, and large sizes the MLIP probe leads both descriptors on OOS error on all seven tasks and on recovery on six, with phonon peak the exception as in Section 3.1. Only the median per-task recovery ratio to the better descriptor changes, from $1.3\times$ to $1.8\times$.

Linear probe. In distribution the linear probe trails on every task, at a median $1.5\times$ the medium predictor’s MAE and $4.1\times$ on band gap with the MLIP embeddings, with a gap that widens as the linear curve saturates at the larger training-set sizes while the trained curves keep descending (learning curves not shown). On the OOS set it narrows the gap on error to a median $1.2\times$, eight of the 21 task-and-representation combinations within 5% or better, while on screening it recovers far less than the MLP predictors on the MLIP embeddings, a median 21% against 49% to 57% (Fig. 12c). Even that recovery overstates the linear probe on CFID’s Debye temperature and phonon peak, where a few of its OOS predictions leave the label range by orders of magnitude and clear τ trivially. The MLIP embeddings’ median margin over the descriptors on OOS error narrows at the linear size from $1.54\times$ to $1.30\times$ but stays above 1 on every task.

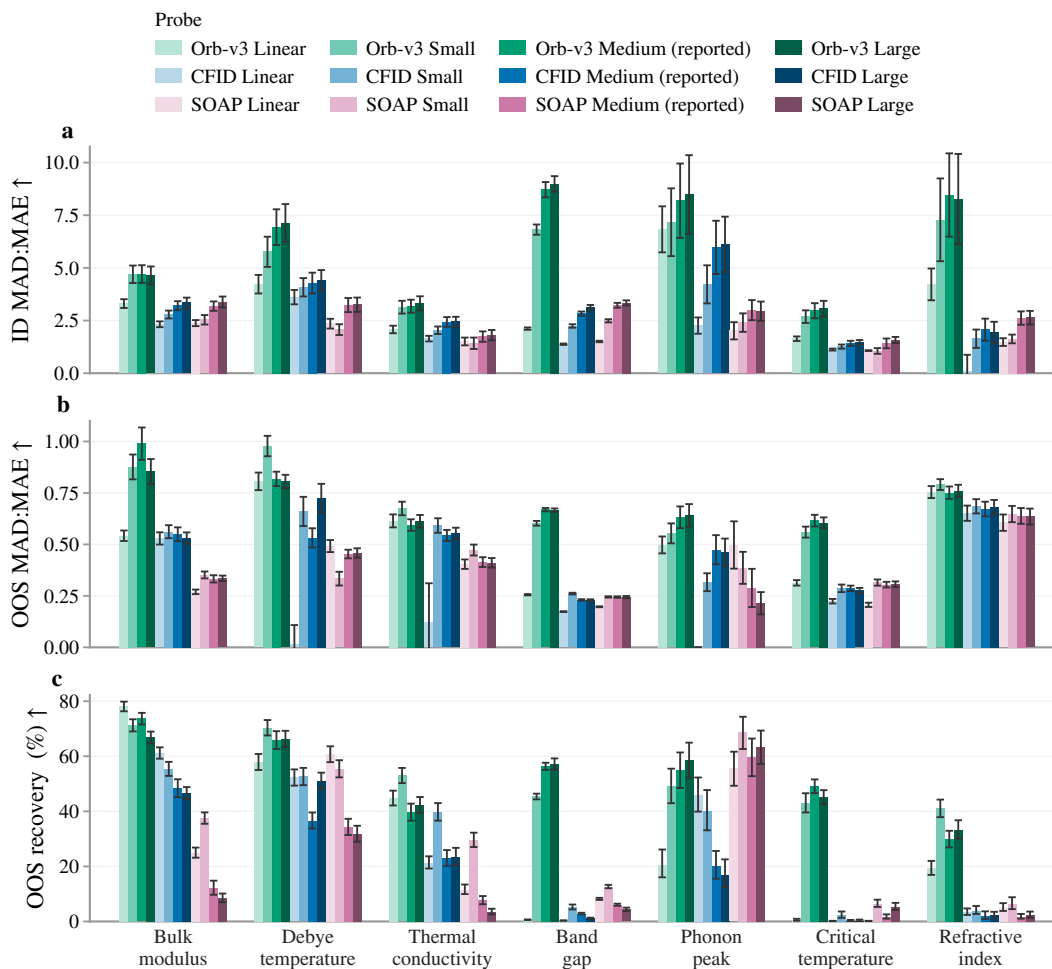


Figure 12: Each predictor size on the three raw representations at the full training set. Small is a 2×256 MLP after the pool (66k parameters on the MLIP embeddings, 263k on SOAP and 392k on CFID). Medium, the reported size, adds a 2×256 per-atom MLP on the MLIP embeddings (263k) and deepens the descriptor probe to 4×256 (390k and 520k). Large widens both to 1024 (3.4M on the MLIP embeddings, 3.1M and 3.7M on the descriptors). Counts cover the MLPs of Eq. 1 alone, the scalar output layer excluded, and Linear is the probe of Appendix A.7. Error bars: ± 1 s.e., Wilson in c.