
Self-Supervised Pretraining for Instrument-Robust Representations of Scattering Data

Darin Merchant¹ Nofit Segal² Miguel Steiner^{2,3} Rafael Gómez-Bombarelli²

¹ETH Zurich, Zurich, Switzerland

²Massachusetts Institute of Technology, Cambridge, MA, USA

³Technical University of Denmark, Kongens Lyngby, Denmark

Abstract

Machine learning models for scattering data are typically trained on simulation because labeled measurements are scarce. However, instrument effects absent from the simulation cause these models to transfer poorly to measurements. Existing approaches learn task-specific robustness to these effects during supervised training that must be relearned for every facility and labeled dataset. Here, we pretrain an encoder self-supervised on a large dataset of pairs of spectra that share a structure and differ in instrument parameters. Pretraining this way yields a representation in which structural targets remain linearly readable when the training spectra come from a different instrument or material class than the test set, and when labels are few. In the regime that experimental pair distribution function analysis occupies, with few labels and heterogeneous instruments, a frozen encoder and a closed-form probe outperform supervised training. Fine-tuning, where labels permit it, adds accuracy at some cost in robustness. One pretrained encoder serves every task, pointing toward the possibility to train a foundation model for scattering data.

1 Introduction

The pair distribution function (PDF) measures local structural order in real space and characterizes not only periodic crystals but also materials without long-range order, such as nanoparticles [1]. Interpreting it requires expert knowledge of the material, the measurement and the quantity sought. This bottleneck is becoming acute as autonomous laboratories and real-time measurement techniques produce data faster than experts can interpret it [2]. Machine learning can automate this analysis [3, 4], both as a final interpretation step and as the characterization step that closes the loop in self-driving laboratories [5, 6]. The limiting factor is labeled data: measurements are costly, so systematic acquisition rarely extends beyond individual research groups, and the data that exist are fragmented across material classes, facilities and measurement conventions, and often not public.

Simulating spectra from structural databases supplies labels by construction, but imperfect materials and instruments introduce variations which simulations do not capture, so models trained on simulation alone transfer poorly to experiment [7, 8]. Existing approaches close this gap by augmenting the simulated data to resemble experimental measurements [9–13]. In this case, the robustness is learned during supervised training for one task, which must therefore be specified in advance, with the procedure repeated for every task, facility, and labeled dataset.

We instead train a model to learn this robustness through self-supervised pretraining on augmented data independently of the downstream task, an approach that has improved robustness of symmetry classification from powder diffraction patterns [14]. We generate multiple views (slightly varying spectra) of the same structure by varying instrument parameters in the simulator. We train an encoder so that views of the same structure are close in latent space while distinct structures remain separated.

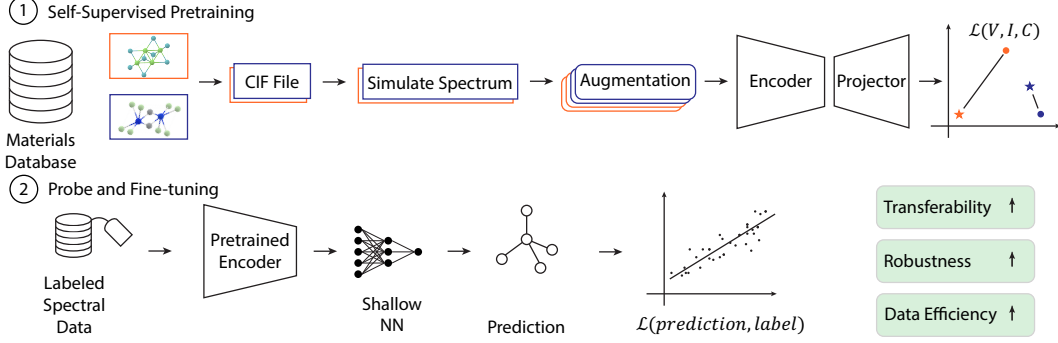


Figure 1: Training procedure. (1) Pretraining: each structure is simulated at randomly drawn instrument parameters, producing multiple views per structure (Figure 3 shows each parameter varied alone). An encoder is trained to represent views of the same structure while spreading different structures across the latent space. (2) Probe and fine-tuning: the encoder is combined with a task-specific head and trained on labeled scattering data.

A single pretrained encoder then serves every downstream task, carrying both the robustness and a structural representation of the PDF.

2 Method

The object of interest in characterization is the structure of the material. The measured PDF depends additionally on the instrument and measurement conditions. We therefore treat every diffraction signal as a view x of a structure through an instrument, and refer to such views as spectra throughout, where c is a crystal structure, $\mathbf{p}_{\text{inst}}$ a vector of instrument parameters, $\mathcal{S}$ a simulator and ν a normalization

$$x = \nu(\mathcal{S}(c, \mathbf{p}_{\text{inst}})). \quad (1)$$

We define a positive pair as two views x_i of the same c under independently drawn $\mathbf{p}_{\text{inst}}$. We pass the view pairs through a CNN encoder and apply a VICReg loss function [15] in the projected space (Appendix A.2.2 and A.2.3).

Data and Simulation. In pretraining, we train on all structures from the Materials Project (133,420 structures) [16] and take $\mathcal{S}$ to be the `diffpy.srreal` PDFCalculator [17]. The PDF $G(r)$ is evaluated on a grid fixed for the whole study, $r \in [0, 50]$ at $\Delta r = 0.01$ (Appendix A.2.1).

We vary four instrument parameters $\mathbf{p}_{\text{inst}}$: U_{iso} sets the thermal peak width, Q_{min} and Q_{max} the low- Q baseline and termination ripples, and Q_{broad} the r -dependent broadening. Two independent draws over the ranges of Table 1 applied to the same structure give a positive pair (Appendix A.1).

3 Results

Evaluation design. Pretraining is meant to give the representation two properties. It should not change when the instrument changes, and it should be rich enough that structural quantities can be read out with a small head and few labels. Three experiments test these properties, each removing something the labeled set would otherwise have to supply on its own. In Section 3.1 we change the instrument between training and testing, in Section 3.2 we shrink the labeled set, and in Section 3.3 we change the material class between training and testing.

All experiments train and evaluate on CHILI-3K [18], a dataset of simulated metal-oxide nanoparticles, which differs from the pretraining domain in three ways. The structures are finite nanoparticles rather than periodic bulk, their PDFs come from a different simulator $\mathcal{S}'$, the `DebyeCalculator` [19], which sums over the particle’s atomic coordinates, and the chemistry narrows to metal oxides. In Section 3.3, we train on CHILI-100K, a larger dataset extending beyond metal oxides.

The instrument shift is constructed by simulating two versions of the spectra for each nanoparticle. The *fixed* spectrum uses CHILI’s own generation parameters, a single instrument realization shared by the entire dataset. For the *varied* spectra we re-simulate the same particle at instrument parameters drawn from the pretraining ranges of Table 1, chosen to cover expected experimental conditions. Within these ranges the representation varies little with the instrument parameters (Appendix A.2.5). Testing always uses the varied test spectra.

We compare three schemes that differ in which components are trained on the labeled set. In the supervised *baseline*, we train the architecture from random initialization, at a learning rate selected on the validation set. The *probe* freezes the pretrained encoder and fits a linear head in closed form. The *fine-tuned* scheme starts from the same encoder and updates it together with the head.

Dataset splits are 80/10/10 and disjoint on reduced formula, so no composition appears in both training and test sets, giving 2530 training, 290 validation and 360 test samples. Table 4 lists the four targets. The two categorical targets are reported as weighted F1 against the *modal floor*, the score of a constant predictor of the most frequent class. Gradient-trained results are means over three seeds (Appendix A.3.3).

3.1 Instrument shift

We train each scheme on the fixed spectra and evaluate on the varied test set, so the only change between training and testing is the instrument. Training the same scheme on varied spectra gives its ceiling, the performance when the shift is absent. The probe and the fine-tuned encoder transfer across the shift, while the baseline does not (Figure 2 top row, per-sample predictions in Appendix A.4.1).

Trained on fixed spectra, every scheme scores below its ceiling, but the gaps differ. The baseline falls to 0.56 and 0.42 on the two categorical targets and to $R^2 = 0.77$ and 0.43 on nanoparticle size and metal-oxygen (M-O) distance. The probe loses at most 0.1 on any target and the fine-tuned encoder at most 0.18. Between the two pretrained schemes, the probe loses less to the shift than fine-tuning does on all targets. This is the expected order, because fine-tuning updates the encoder on the fixed spectra, so part of the pretrained robustness is given up in adapting to the one instrument realization.

At the ceiling, the fine-tuned scheme and the baseline are similar on all targets. The probe matches both on the categorical targets and trails slightly on nanoparticle size. On M-O distance the frozen representation read by a linear head caps below what updating the encoder reaches.

3.2 Label efficiency

All three schemes are trained on varied spectra, so only the label budget varies, the strongest comparison for the supervised baseline. Across a 20-fold reduction of the labeled set, the performance of the probe declines by at most 0.12. Performance of the fine-tuned scheme declines more, by up to 0.29. The baseline sits at or below chance on all the targets until a target-dependent set size and then rises to competitive performance. It draws level with the pretrained schemes only between 632 labels and the full set, depending on the target, which is a training set size out of reach for most experimental data. Below that point, the probe and fine-tuned schemes exceed the baseline on every target.

Below the batch floor of 64 (dotted line in Figure 2, bottom row), only the probe is trained with ridge regression (Appendix A.3.3). The probe decays gradually, reaching $R^2 = 0.50$ on nanoparticle size at 13 labels, while the baseline sits at chance with ten times as many labels (Appendix A.4.2).

3.3 Material shift

In the last experiment we change the materials the labels come from, evaluating on the varied CHILI-3K test set throughout. The *no shift* condition trains on varied CHILI-3K spectra. The *material shift* trains instead on 2,530 CHILI-100K nanoparticles, built from experimentally determined Crystallography Open Database structures [20], with compositions disjoint from the test set, still on varied spectra (Figure 7). The *instrument shift* restricts that training set to the fixed spectra. The categorical targets are dropped because their classes differ between training and test sets (Figure 12, Table 7).

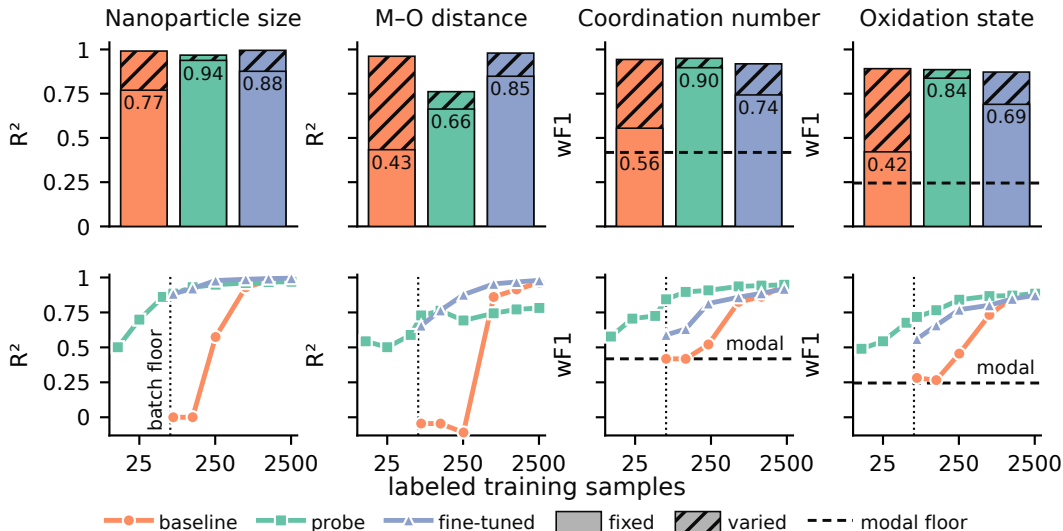


Figure 2: Instrument shift and label efficiency. Top row: test performance on the four targets for each scheme trained on fixed spectra (solid) and on varied spectra (striped), evaluated on the varied test set, with dashed lines at the modal floor. Bottom row: test performance as the labeled training set shrinks from 2530 to 13 samples, with only the probe trained below the batch floor (dotted line). Bars and points are means over three seeds (Tables 5 and 6).

On M-O distance, the probe declines stepwise, from $R^2 = 0.76$ with no shift to 0.70 under the material shift and 0.55 with the instrument added. The baseline drops from 0.96 to 0.78 under the material shift alone and to -0.49 , below the test set mean, with the instrument added, amplifying the failure of Section 3.1. The fine-tuned scheme lands at 0.34, below the probe but well above the baseline, so fine-tuning keeps most of the pretrained robustness, while freezing retains more.

Nanoparticle size stays above 0.86 for every scheme and condition, as it is determined by the decay envelope of $G(r)$, which does not depend on the material class, while M-O distance does.

4 Discussion and Conclusion

In experimental PDF analysis, the labeled sets are small, cover narrow material classes and carry effects from instruments that differ between facilities. In the three experiments that test this setting, the pretrained representation holds where the supervised baseline breaks down. Trained on one instrument realization, the probe carries most of its performance across an instrument shift and the baseline does not. With fewer labels, the probe declines only gradually and is still predictive at 13 labels. In contrast, the baseline matches the pretrained schemes only at label counts beyond the size of most experimental datasets. Under the material and instrument shift, the pretrained schemes degrade but remain predictive, while the baseline drops below chance on the material-dependent target.

Pretraining changes what the labels have to supply. The supervised baseline obtains its representation, readout and robustness to the instrument from the labeled set, and with enough labels it matches both pretrained schemes. The pretrained encoder obtains an instrument-robust representation of the PDF from instrument-varied views of each structure across a broad chemical space before any supervised training, so the labeled data only trains the readout, which needs far less data than a full model. This is why the pretrained schemes hold up when labels are few, narrow or from a different instrument.

All evaluation is on simulated spectra, so the results establish robustness under a change of simulator and structure class within a known instrument model. Natural extensions include broadening the augmentation family and ranges, and repeating the evaluation on measured nanoparticle PDFs.

Nothing in the construction is specific to the PDF: any modality in which a measurement is a view of a structure through an instrument admits the same treatment, with instrument effects designed into the augmentations once during pretraining instead of augmenting task-specific supervised training.

References

- [1] Takeshi Egami and S. J. L. Billinge. *Underneath the Bragg peaks: structural analysis of complex materials*. Number 16 in Pergamon materials series. Elsevier, Amsterdam, 2nd edition edition, 2012. ISBN 978-0-08-097133-9.
- [2] Andy S. Anker, John L. A. Gardner, Louise A. M. Rosset, Andrew L. Goodwin, and Volker L. Deringer. Autonomous interpretation of atomistic scattering data, October 2025. URL <http://arxiv.org/abs/2510.05938>. arXiv:2510.05938 [cond-mat.mtrl-sci].
- [3] Emil T. S. Kjær, Andy S. Anker, Marcus N. Weng, Simon J. L. Billinge, Raghavendra Selvan, and Kirsten M. Ø. Jensen. DeepStruc: towards structure solution from pair distribution function data using deep generative models. *Digital Discovery*, 2(1):69–80, 2023. ISSN 2635-098X. doi: 10.1039/D2DD00086E. URL <https://pubs.rsc.org/dd/article/2/1/69-80/764868>.
- [4] Emil T. S. Kjær, Andy S. Anker, Andrea Kirsch, Joakim Lajer, Olivia Aalling-Frederiksen, Simon J. L. Billinge, and Kirsten M. Ø. Jensen. MLstructureMining: a machine learning tool for structure identification from X-ray pair distribution functions. *Digital Discovery*, 3(5):908–918, 2024. ISSN 2635-098X. doi: 10.1039/D4DD00001C. URL <https://pubs.rsc.org/dd/article/3/5/908-918/846009>.
- [5] Andy S. Anker, Jonas H. Jensen, Miguel Gonzalez-Duque, Rodrigo Moreno, Aleksandra Smolska, Mikkel Juelsholt, Vincent Hardion, Mads R. V. Jorgensen, Andres Faina, Jonathan Quinson, Kasper Stoy, and Tejs Vegge. Autonomous nanoparticle synthesis by design, May 2025. URL <http://arxiv.org/abs/2505.13571>. arXiv:2505.13571 [cond-mat.mtrl-sci].
- [6] Nathan J. Szymanski, Bernardus Rendy, Yuxing Fei, Rishi E. Kumar, Tanjin He, David Milsted, Matthew J. McDermott, Max Gallant, Ekin Dogus Cubuk, Amil Merchant, Haegyem Kim, Anubhav Jain, Christopher J. Bartel, Kristin Persson, Yan Zeng, and Gerbrand Ceder. An autonomous laboratory for the accelerated synthesis of inorganic materials. *Nature*, 624(7990):86–91, December 2023. ISSN 1476-4687. doi: 10.1038/s41586-023-06734-w. URL <https://www.nature.com/articles/s41586-023-06734-w>.
- [7] Pascal Marc Vecsei, Kenny Choo, Johan Chang, and Titus Neupert. Neural network based classification of crystal symmetries from x-ray diffraction patterns. *Physical Review B*, 99(24):245120, June 2019. ISSN 2469-9950, 2469-9969. doi: 10.1103/PhysRevB.99.245120. URL <https://link.aps.org/doi/10.1103/PhysRevB.99.245120>.
- [8] Jerardo E. Salgado, Samuel Lerman, Zhaotong Du, Chenliang Xu, and Niaz Abdolrahim. Automated classification of big X-ray diffraction data using deep learning models. *npj Computational Materials*, 9(1):214, December 2023. ISSN 2057-3960. doi: 10.1038/s41524-023-01164-8. URL <https://www.nature.com/articles/s41524-023-01164-8>.
- [9] Nathan J. Szymanski, Christopher J. Bartel, Yan Zeng, Qingsong Tu, and Gerbrand Ceder. Probabilistic Deep Learning Approach to Automate the Interpretation of Multi-phase Diffraction Spectra. *Chemistry of Materials*, 33(11):4204–4215, May 2021. ISSN 0897-4756. doi: 10.1021/acs.chemmater.1c01071. URL <https://doi.org/10.1021/acs.chemmater.1c01071>.
- [10] Nathan J. Szymanski, Sean Fu, Ellen Persson, and Gerbrand Ceder. Integrated analysis of X-ray diffraction patterns and pair distribution functions for machine-learned phase identification. *npj Computational Materials*, 10(1):45, February 2024. ISSN 2057-3960. doi: 10.1038/s41524-024-01230-9. URL <https://www.nature.com/articles/s41524-024-01230-9>.
- [11] Shuyan Zhang, Jie Gong, Sharon Chu, Daniel Z. Xiao, B. Reeja-Jayan, and Alan J. H. McGaughey. Pair distribution function analysis for oxide defect identification through feature extraction and supervised learning. *APL Machine Learning*, 1(2):026115, June 2023. ISSN 2770-9019. doi: 10.1063/5.0130681. URL <https://pubs.aip.org/aml/article/1/2/026115/2896153/Pair-distribution-function-analysis-for-oxide>.
- [12] Jin-Woong Lee, Woon Bae Park, Jin Hee Lee, Satendra Pal Singh, and Kee-Sun Sohn. A deep-learning technique for phase identification in multiphase inorganic compounds using

- synthetic XRD powder patterns. *Nature Communications*, 11(1):86, January 2020. ISSN 2041-1723. doi: 10.1038/s41467-019-13749-3. URL <https://www.nature.com/articles/s41467-019-13749-3>.
- [13] Phillip M. Maffettone, Lars Banko, Peng Cui, Yury Lysogorskiy, Marc A. Little, Daniel Olds, Alfred Ludwig, and Andrew I. Cooper. Crystallography companion agent for high-throughput materials discovery. *Nature Computational Science*, 1(4):290–297, April 2021. ISSN 2662-8457. doi: 10.1038/s43588-021-00059-2. URL <https://www.nature.com/articles/s43588-021-00059-2>.
 - [14] Shubhayu Das, Markus Vorholt, Andreas Houben, and Richard Dronskowski. Learning Self-Supervised Representations of Powder-Diffraction Patterns. *Crystals*, 15(5):393, May 2025. ISSN 2073-4352. doi: 10.3390/cryst15050393. URL <https://www.mdpi.com/2073-4352/15/5/393>.
 - [15] Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning, January 2022. URL <http://arxiv.org/abs/2105.04906>. arXiv:2105.04906 [cs.CV].
 - [16] 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):011002, July 2013. ISSN 2166-532X. doi: 10.1063/1.4812323. URL <https://pubs.aip.org/apm/article/1/1/011002/119685/Commentary-The-Materials-Project-A-materials>.
 - [17] P. Juhás, C. Farrow, X. Yang, K. Knox, and S. Billinge. Complex modeling: a strategy and software program for combining multiple information sources to solve ill posed structure and nanostructure inverse problems. *Acta Crystallographica Section A: Foundations and Advances*, 71(6):562–568, November 2015. ISSN 2053-2733. doi: 10.1107/S2053273315014473. URL <https://journals.iucr.org/a/issues/2015/06/00/ae5008/>.
 - [18] Ulrik Friis-Jensen, Frederik L. Johansen, Andy S. Anker, Erik B. Dam, Kirsten M. Ø Jensen, and Raghavendra Selvan. CHILI: Chemically-Informed Large-scale Inorganic Nanomaterials Dataset for Advancing Graph Machine Learning. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4962–4973, August 2024. doi: 10.1145/3637528.3671538. URL <http://arxiv.org/abs/2402.13221>. arXiv:2402.13221 [cs.LG].
 - [19] Frederik L. Johansen, Andy S. Anker, Ulrik Friis-Jensen, Erik B. Dam, Kirsten M. ø Jensen, and Raghavendra Selvan. A GPU-Accelerated Open-Source Python Package for Calculating Powder Diffraction, Small-Angle-, and Total Scattering with the Debye Scattering Equation. *Journal of Open Source Software*, 9(94):6024, February 2024. ISSN 2475-9066. doi: 10.21105/joss.06024. URL <https://joss.theoj.org/papers/10.21105/joss.06024>.
 - [20] Saulius Gražulis, Adriana Daškevič, Andrius Merkys, Daniel Chateigner, Luca Lutterotti, Miguel Quirós, Nadezhda R. Serebryanaya, Peter Moeck, Robert T. Downs, and Armel Le Bail. Crystallography Open Database (COD): an open-access collection of crystal structures and platform for world-wide collaboration. *Nucleic Acids Research*, 40(D1):D420–D427, January 2012. ISSN 0305-1048. doi: 10.1093/nar/gkr900. URL <https://doi.org/10.1093/nar/gkr900>.

A Appendix

A.1 Simulation & Augmentation

Given a structure and its CIF file, we sample all four augmentations from the ranges in Table 1 to produce a view of the structure. The four are drawn independently and uniformly, in the order $(U_{\text{iso}}, Q_{\text{max}}, Q_{\text{broad}}, Q_{\text{min}})$, and the same draw serves both simulators used in this work. Pretraining views come from $\mathcal{S}$, the `diffpy.srreal` PDFCalculator, applied to the periodic cell parsed from the CIF on $r \in [0, 50]$ Å at $\Delta r = 0.01$. Downstream views of CHILI particles come from $\mathcal{S}'$, the `DebyeCalculator`, applied to the particle’s actual atomic coordinates on $r \in [0, 60]$ Å at the same step.

Table 1: The augmentation ranges $\mathbf{p}_{\text{inst}}$. Each view draws all four parameters independently and uniformly.

Parameter	Symbol	Range	Unit
Thermal Displacement parameter	U_{iso}	[0.001, 0.03]	Å ²
Maximum momentum transfer	Q_{max}	[15, 30]	Å ⁻¹
Peak-broadening coefficient	Q_{broad}	[0.01, 0.06]	Å ⁻¹
Minimum momentum transfer	Q_{min}	[0.5, 2.0]	Å ⁻¹

Two independent draws applied to the same structure c give the positive pair of the pretraining loss,

$$(x_1, x_2) = \left(\nu(\mathcal{S}(c, \mathbf{p}_{\text{inst}, 1})), \nu(\mathcal{S}(c, \mathbf{p}_{\text{inst}, 2})) \right), \quad \mathbf{p}_{\text{inst}, 1}, \mathbf{p}_{\text{inst}, 2} \sim \mathcal{U}(\mathbf{P}_{\text{inst}}). \quad (2)$$

ν is a per-sample min-max rescaling into $[0, 1]$ because the absolute amplitude of a PDF does not transfer between facilities.

PDFCalculator evaluates $G(r)$ as a sum of Gaussian peaks over the interatomic distances r_{ij} of the cell, on a linear baseline set by the number density ρ_0 ,

$$G(r) = \frac{1}{N} \sum_{i \neq j} \frac{b_i b_j}{\langle b \rangle^2} \frac{1}{\sqrt{2\pi} r \sigma_{ij}} \exp\left(-\frac{(r - r_{ij})^2}{2\sigma_{ij}^2}\right) - 4\pi r \rho_0, \quad (3)$$

where b_i are scattering lengths and N is the number of atoms in the cell. U_{iso} and Q_{broad} both enter through the peak width σ_{ij} . Every atom carries the same isotropic displacement parameter, so the distance between any two uncorrelated atoms has variance $2U_{\text{iso}}$, giving an r -independent width $\sigma_0 = \sqrt{2U_{\text{iso}}}$. Q_{broad} enters in the width model of the peaks, with the correlated-motion terms δ_1 and δ_2 set to zero,

$$\sigma_{ij}(r_{ij}) = \sigma_0 \sqrt{1 - \frac{\delta_1}{r_{ij}} - \frac{\delta_2}{r_{ij}^2} + Q_{\text{broad}}^2 r_{ij}^2} \xrightarrow{\delta_1=\delta_2=0} \sqrt{2U_{\text{iso}}} \sqrt{1 + (Q_{\text{broad}} r_{ij})^2}. \quad (4)$$

U_{iso} therefore sets a width shared by all peaks and Q_{broad} a width that grows with r , which is why the Q_{broad} panel of Figure 3 is drawn at large r and the U_{iso} panel at small r .

Q_{min} and Q_{max} are properties of the measured window rather than of the structure, and they enter through the definition of the PDF. $G(r)$ is the sine Fourier transform of the reduced total scattering structure function $F(Q) = Q[S(Q) - 1]$,

$$G(r) = \frac{2}{\pi} \int_0^\infty Q [S(Q) - 1] \sin(Qr) dQ. \quad (5)$$

No measurement supplies $F(Q)$ over that entire domain. A beamstop and the finite width of the direct beam remove the smallest scattering angles, and the detector’s angular reach, together with the decay of the coherent signal at high Q , imposes an upper cutoff, so what is transformed in practice is

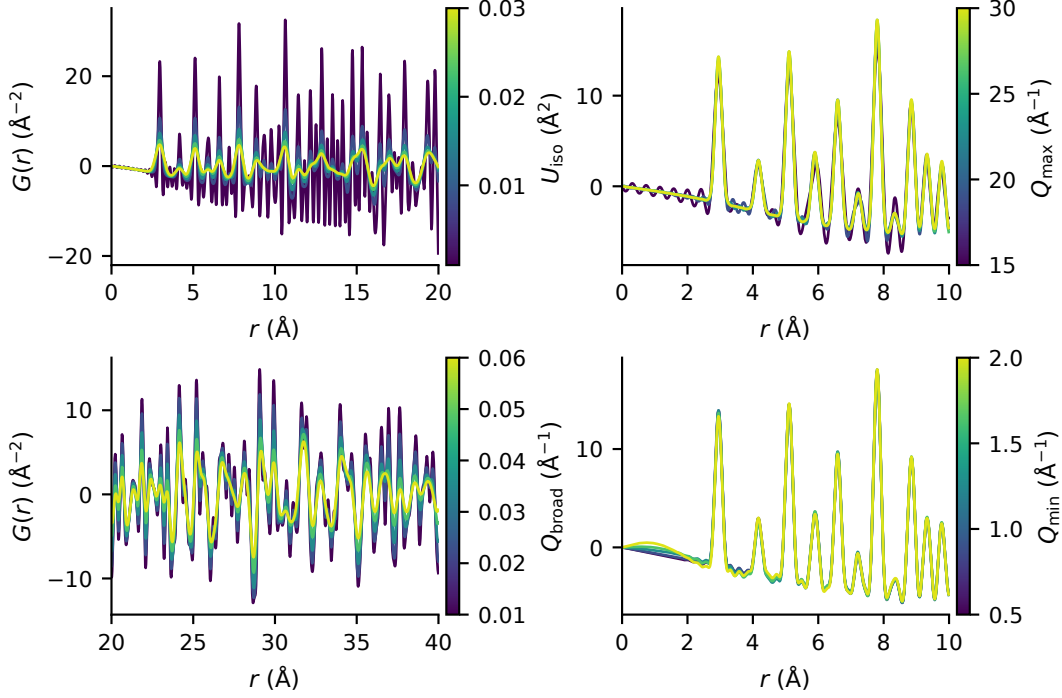


Figure 3: One instrument parameter varied alone. Each panel sweeps one parameter of Table 1 across its pretraining range for one structure, holding the other three at their nominal values, over the r window where that parameter acts.

$$G(r) = \frac{2}{\pi} \int_{Q_{\min}}^{Q_{\max}} Q [S(Q) - 1] \sin(Qr) dQ \quad (6)$$

and the two limits are the two parameters we draw. Truncating at $Q_{\max}$ multiplies $F(Q)$ by a rectangular window and therefore convolves $G(r)$ with that window's transform, producing the termination ripples of period $2\pi/Q_{\max}$ visible in the $Q_{\max}$ panel of Figure 3. Discarding $Q < Q_{\min}$ removes the slowly varying part of $F(Q)$ and distorts the baseline of $G(r)$, which is what the $Q_{\min}$ panel shows. In PDFCalculator the window is not an integration range but a correction applied in Q to the peak sum of Eq. 3.

For the CHILI particles, DebyeCalculator evaluates the Debye scattering equation over the particle's atoms in momentum space,

$$I(Q) = \sum_i \sum_j f_i(Q) f_j(Q) \frac{\sin(Qr_{ij})}{Qr_{ij}}, \quad (7)$$

on a grid of step $Q_{\text{step}} = 0.05 \text{ \AA}^{-1}$ running from $Q_{\min}$ to $Q_{\max}$, reduces it to $F(Q)$, and applies the truncated sine transform of Eq. (6). Here, $Q_{\min}$ and $Q_{\max}$ truncate the transform literally, as its integration limits. U_{iso} enters as the isotropic Debye–Waller factor, through $B_{\text{iso}} = 8\pi^2 U_{\text{iso}}$, damping $I(Q)$ at high Q . Q_{broad} has no counterpart in the Debye sum and is applied to the resulting $G(r)$ as a non-stationary Gaussian smear, in which each point is spread with the width that Eq. (4) adds in quadrature at its own r ,

$$\sigma_{\text{add}}(r) = \sqrt{2U_{\text{iso}}} Q_{\text{broad}} r, \quad (8)$$

using a kernel of unit sum, so the smear conserves area. Q_{damp} is held fixed at $0.04 \text{ \AA}^{-1}$, the value CHILI's own spectra were generated with, and is not drawn as its envelope decays with r in the same way a finite particle's $G(r)$ does, so drawing it would overwrite the nanoparticle size target.

Figure 4 shows two views of one structure under this procedure, for Materials Project, CHILI-3K and CHILI-100K.

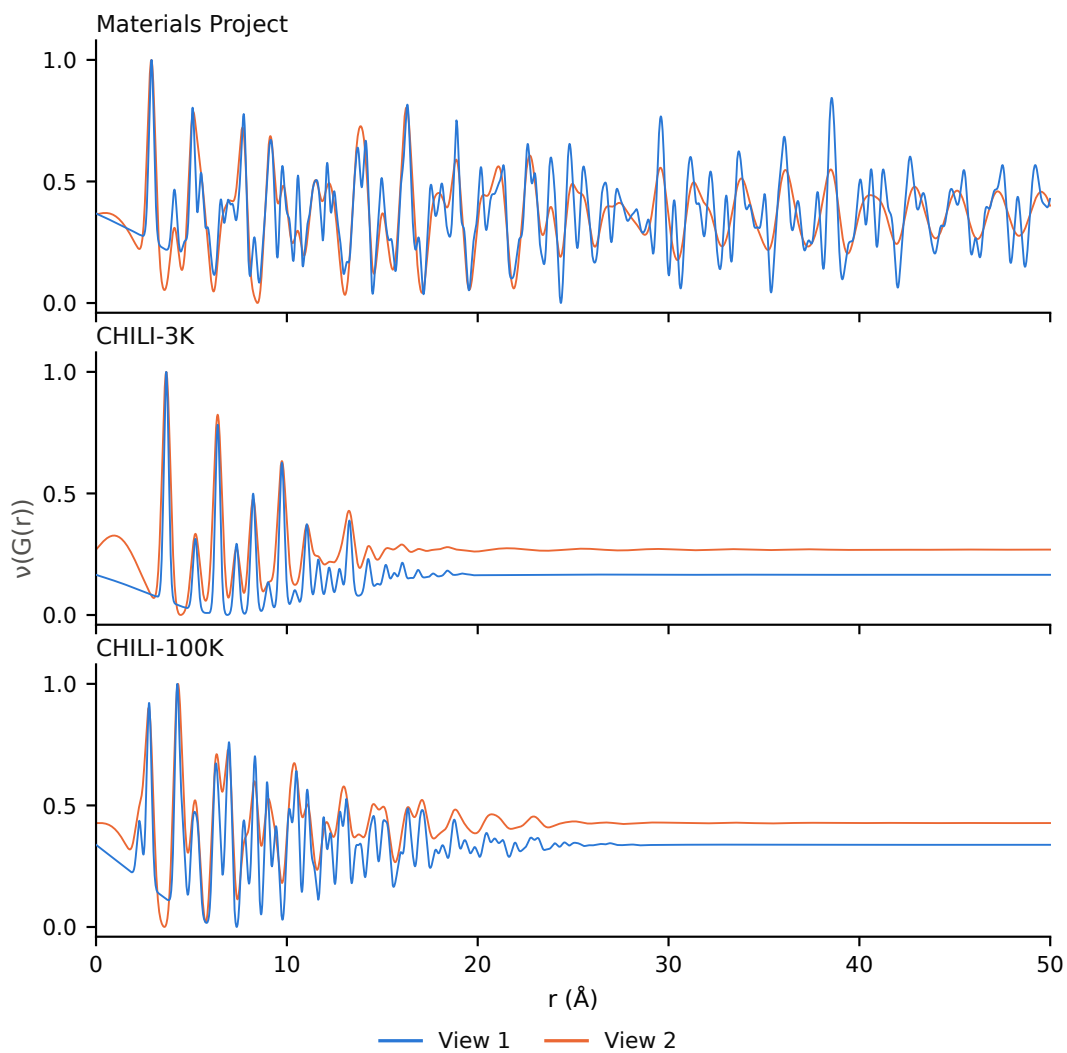


Figure 4: Two views of one structure from each dataset. Views of the same structure differ only in the drawn parameters.

A.2 Pretraining

Given a structure sampled from Materials Project, a view is generated as described above and enters as a single-channel 5000-point vector. A 1D residual convolutional encoder with attention pooling maps it to a 256-dimensional representation, and an MLP projection head maps that to the 512-dimensional vector the pretraining loss reads. All results in this work come from a single pretraining run, whose architecture is given in Table 2 and whose training parameters are given in Table 3.

A.2.1 Data

We train on the Materials Project 2019.04.01 version with 133,420 relaxed ground-state structures and no atom-count cap. Every structure is assigned to the training split, so there is no held-out pretraining set and no pretraining-domain validation number in this work. All $133,420 \times 32 = 4,269,440$ views were simulated once and stored, so the two views drawn each epoch come from $\binom{32}{2} = 496$ distinct pairs per structure.

A.2.2 Architecture

The encoder is a 1D ResNet followed by attention pooling. A stem convolution of width 7 at stride 2, batch normalization, ReLU and a max-pool of width 3 at stride 2 reduce the 5000-point input to 1250 positions at 32 channels. Four residual blocks follow, each at stride 2, widening $32 \rightarrow 64 \rightarrow 128 \rightarrow 256 \rightarrow 256$. A block is two 3-wide convolutions with batch normalization and a 1×1 projected skip where the shape changes. Pooling is a per-position attention logit from a $\text{Linear}(256 \rightarrow 1)$, softmaxed over the sequence and used to take a weighted sum, which collapses the sequence dimension with no positional parameters. The projector is a three-layer MLP, $\text{Linear}(256 \rightarrow 512) \rightarrow \text{BN} \rightarrow \text{ReLU} \rightarrow \text{Linear}(512 \rightarrow 512) \rightarrow \text{BN} \rightarrow \text{ReLU} \rightarrow \text{Linear}(512 \rightarrow 512)$. No architectural sweep was run: every pretraining run in this work uses the parameters in Table 2, so no variant of the encoder’s width, depth or pooling was measured.

Table 2: Encoder and projector. Output shapes are for the $L = 5000$ pretraining grid. Only the encoder is carried downstream.

Stage	Operation	Output	Parameters
Input	—	1×5000	—
Stem	$\text{Conv}(1 \rightarrow 32, k7, s2)$, BN, ReLU, $\text{MaxPool}(k3, s2)$	32×1250	288
Block 1	$\text{ResBlock}(32 \rightarrow 64)$, stride 2	64×625	20,864
Block 2	$\text{ResBlock}(64 \rightarrow 128)$, stride 2	128×313	82,688
Block 3	$\text{ResBlock}(128 \rightarrow 256)$, stride 2	256×157	329,216
Block 4	$\text{ResBlock}(256 \rightarrow 256)$, stride 2	256×79	460,288
Pooling	$\text{Linear}(256 \rightarrow 1)$, softmax, weighted sum	$h \in \mathbb{R}^{256}$	257
Encoder subtotal (the downstream representation)			893,601
Projector	3-layer MLP $256 \rightarrow 512 \rightarrow 512 \rightarrow 512$	$z \in \mathbb{R}^{512}$	658,944
Total trainable			1,552,545

A.2.3 Loss Function and Hyperparameters

The encoder maps a view x to its representation $h(x) \in \mathbb{R}^{256}$, and the projection head maps $h(x)$ to $z(x) \in \mathbb{R}^{512}$. The projection head is discarded after pretraining, so h is the representation every downstream readout receives. The VICReg loss is applied to the projected views of a positive pair, (z_1, z_2) , each a batch of B embeddings of dimension D , as three terms. The invariance term pulls the two views of a structure together,

$$\ell_{\text{inv}}(z_1, z_2) = \frac{1}{BD} \sum_{b=1}^B \|z_{1,b} - z_{2,b}\|^2. \quad (9)$$

The variance term hedges each dimension’s standard deviation at 1, which prevents the embedding from collapsing to a point,

$$\ell_{\text{var}}(z_1, z_2) = \frac{1}{D} \sum_{j=1}^D \left[\max(0, 1 - \sigma(z_1^j)) + \max(0, 1 - \sigma(z_2^j)) \right], \quad \sigma(z^j) = \sqrt{\text{Var}(z^j) + \varepsilon}. \quad (10)$$

The covariance term penalizes the off-diagonal of each view’s feature covariance, which decorrelates the dimensions,

$$\ell_{\text{cov}}(z_1, z_2) = \frac{1}{D} \sum_{k \in \{1, 2\}} \sum_{i \neq j} [C(z_k)]_{ij}^2, \quad C(z) = \frac{1}{B-1} \sum_{b=1}^B (z_b - \bar{z})(z_b - \bar{z})^\top, \quad (11)$$

and the objective is their weighted sum,

$$\mathcal{L} = \lambda_{\text{sim}} \ell_{\text{inv}} + \lambda_{\text{var}} \ell_{\text{var}} + \lambda_{\text{cov}} \ell_{\text{cov}}. \quad (12)$$

We use $\varepsilon = 10^{-4}$, $B = 1024$, $D = 512$ and $(\lambda_{\text{sim}}, \lambda_{\text{var}}, \lambda_{\text{cov}}) = (25, 25, 50)$.

No validation loss is recorded during pretraining, and the total of Eq. (12) scales with λ_{cov} by construction, so the pretraining loss does not rank checkpoints across the sweep.

A.2.4 Training

In training, we apply AdamW as optimizer with weight decay 10^{-2} , with a linear warmup to a peak learning rate of 1.4×10^{-4} over 8 epochs followed by a cosine decay to a floor of 10^{-5} , stepped once per iteration. Gradients are clipped at norm 5 and training runs in mixed precision on a single V100, taking 2 h 31 min for the full 300 epochs. Of the run’s 39,000 optimizer steps, 24 were discarded by the gradient scaler as non-finite. Each term moves in one direction after warmup: invariance falls from 0.0885 at epoch 8 to 0.0151 at epoch 300 and variance from 1.064 to 0.743, while covariance rises from 0.085 to 0.153 and the weighted total settles from 33.06 to 26.60 (Figure 5). The checkpoint carried into every downstream experiment is the one with the lowest mean training loss, which falls at epoch 295.

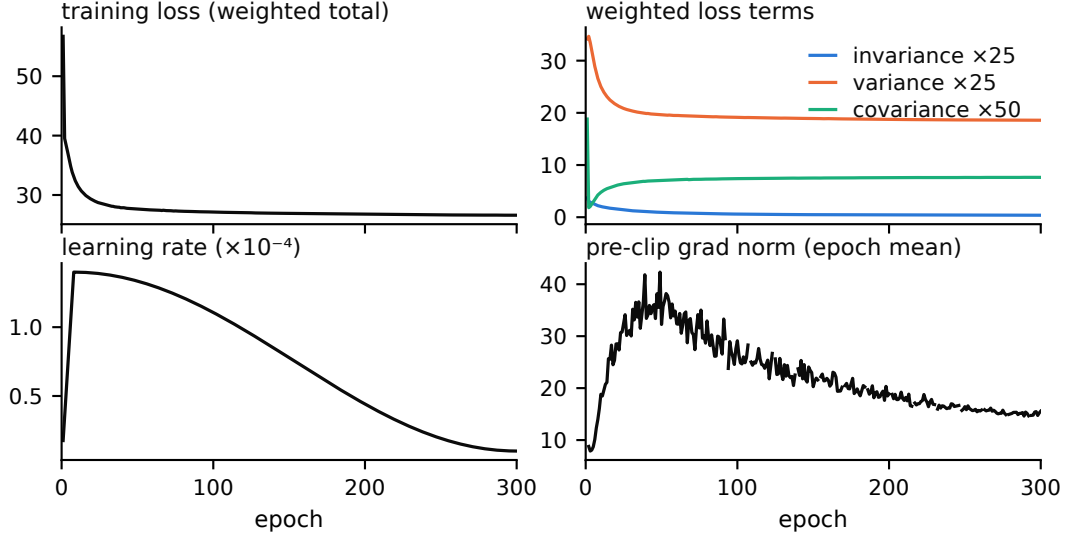


Figure 5: Pretraining diagnostics over 300 epochs. Weighted total loss, the three weighted loss terms, the learning rate schedule and the mean pre-clip gradient norm.

Table 3: Pretraining parameters.

	Parameter	Value
Data	Corpus	Materials Project, 133,420 structures
	Views banked per structure	32
	Signal length	5000
Model	Representation dimension	256
	Projection dimension	512
Loss	Objective ($\lambda_{\text{sim}}, \lambda_{\text{var}}, \lambda_{\text{cov}}$)	VICReg (25, 25, 50)
Optimizer	Algorithm	AdamW
	Weight decay	10^{-2}
	Peak learning rate	1.4×10^{-4}
	Warmup	8 epochs, linear from $10^{-3} \times \text{peak}$
	Decay	cosine to 10^{-5} , per iteration
	Gradient clipping	5.0
Schedule	Batch size	1024
	Steps per epoch	130
	Epochs	300 (39,000 steps, 1,040 warmup)
	Precision	mixed, fp16 with loss scaling
	Seed	42

A.2.5 Augmentation Ranges

We sweep the augmentation ranges to measure how the representation changes when one instrument parameter is shifted. We freeze the encoder and sample 256 structures from the Materials Project. For each parameter of Table 1, we sweep 16 points spanning its physical bounds, eight inside the pretraining range and eight outside it, holding the other three parameters at $U_{\text{iso}} = 0.005 \text{ \AA}^2$, $Q_{\text{max}} = 22.5 \text{ \AA}^{-1}$, $Q_{\text{broad}} = 0.03 \text{ \AA}^{-1}$ and $Q_{\text{min}} = 1.0 \text{ \AA}^{-1}$. Each point yields a view x of structure c as in Eq. (1).

The reference $\bar{h}(c)$ is the mean of $h(x)$ over the eight in-range views of structure c , and the displacement of a view is the cosine distance

$$d(x) = 1 - \cos(h(x), \bar{h}(c)). \quad (13)$$

The same displacement computed on the spectra $\nu(G(r))$ measures how far the parameter moves the signal itself. The two spaces have different global scales, so each curve is divided by its own space’s negative-pair distance, the mean cosine distance between different structures at the nominal parameters. A displacement of 1 therefore means a parameter has moved a structure as far as two different structures lie apart. The distance between the two views of a positive pair, drawn as in Eq. (2), lies at 0.014 of the negative-pair distance in latent space and at 0.115 in input space. Figure 6 shows the median over the 256 structures with a band of one geometric standard deviation.

Inside the pretraining range the median latent displacement stays below 2% of the negative-pair distance on every parameter, while the input displacement reaches 21% on U_{iso} . The varied spectra of Section 3 are drawn from these ranges, so a varied spectrum and a fixed spectrum of the same particle receive nearly the same representation, which is the mechanism behind the instrument-shift results. Outside the pretraining range the order reverses on U_{iso} , Q_{max} and Q_{min} . At $Q_{\text{max}} = 5 \text{ \AA}^{-1}$ the latent displacement reaches 0.91 of the negative-pair distance and the input displacement 0.29. The robustness is a property of the ranges the encoder was trained on.

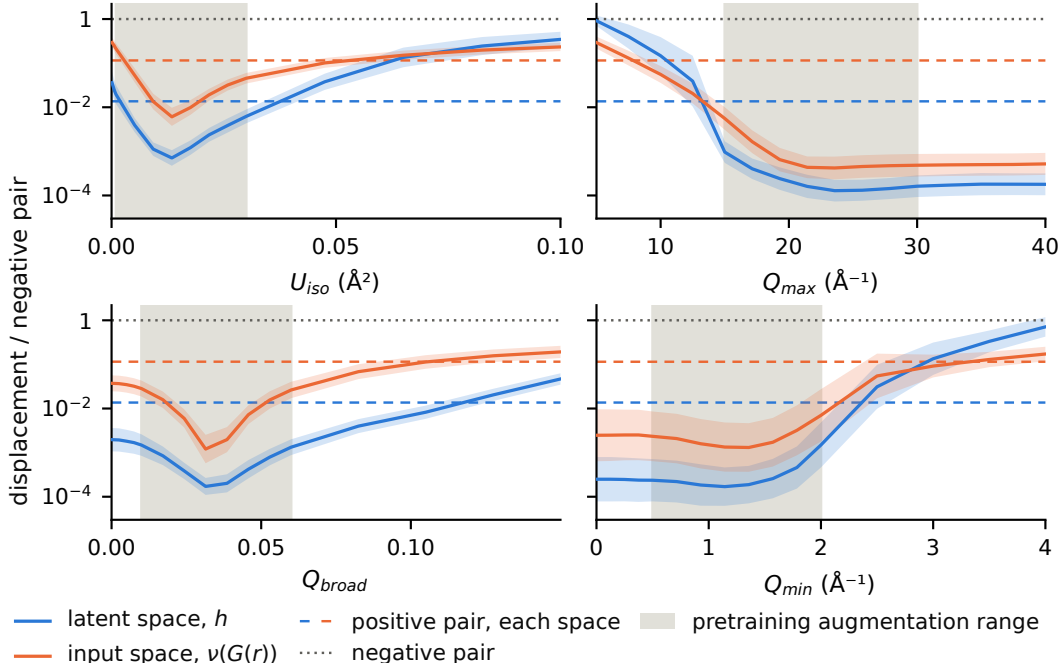


Figure 6: Displacement of the representation under an instrument sweep. Each panel shows the displacement against one instrument parameter, normalized to the negative-pair distance of its space, with medians over 256 structures, bands of one geometric standard deviation, dashed lines at the positive-pair distances and shading over the pretraining augmentation range.

A.3 Downstream Readout

A.3.1 Data

CHILI-3K. Downstream evaluation uses CHILI-3K, 3180 metal oxide nanoparticles built from 53 metals, 12 structure prototypes and 5 particle sizes. Each prototype carries exactly one stoichiometry. RockSalt, Wurtzite, ZincBlende, NiAs and CsCl give MO, Rutile, Fluorite, CdCl₂ and CdI₂ give MO₂, AntiFluorite gives M₂O, Spinel gives M₃O₄ and ReO₃ gives MO₃. The set is enumerated rather than observed experimentally, so it contains compositions that are not known phases, such as Al₂O₃, AuO₃ and B₃O₄. Every sample carries the periodic parent cell it was carved from, reconstructed from the atomic numbers, fractional coordinates and cell parameters, together with CHILI’s own simulated xPDF on 6000 points over $r \in [0, 60]$ Å at stepsize $\Delta r = 0.01$.

Splits. Splits are 80/10/10 and disjoint on reduced formula. The distinct formulas are shuffled at a fixed seed and sliced by those fractions, with the test split taking the remainder. Grouping on formula rather than on structure keeps every polymorph of a composition, and therefore all particle sizes of every parent structure, within one split. The 3180 samples form 265 formulas, which split 212/26/27 into 2530 training, 290 validation and 360 test samples. The split is not stratified. CHILI’s published split is stratified by crystal system, so the balance across crystal systems differs from the released one. Every experiment evaluates on this test set.

CHILI-100K. CHILI-100K contains nanoparticles built from 20,882 experimentally determined structures of the Crystallography Open Database [20], each at up to five particle sizes. For the experiments of Section 3.3 we keep the binary metal oxides, for which the four targets of Table 4 are defined, and remove every composition that appears in the CHILI-3K test set. The remaining 110 compositions are split by the same procedure, and from the resulting 4,025 training samples we sample 2,530 uniformly at a fixed seed, matching the size of the CHILI-3K training set. Figure 7 compares the structures of the two training sets. CHILI-100K spans seven crystal systems, 58 space

groups and larger unit cells, against four crystal systems and nine space groups in CHILI-3K, and its experimentally refined cells carry distortions that CHILI-3K’s idealized prototypes lack.

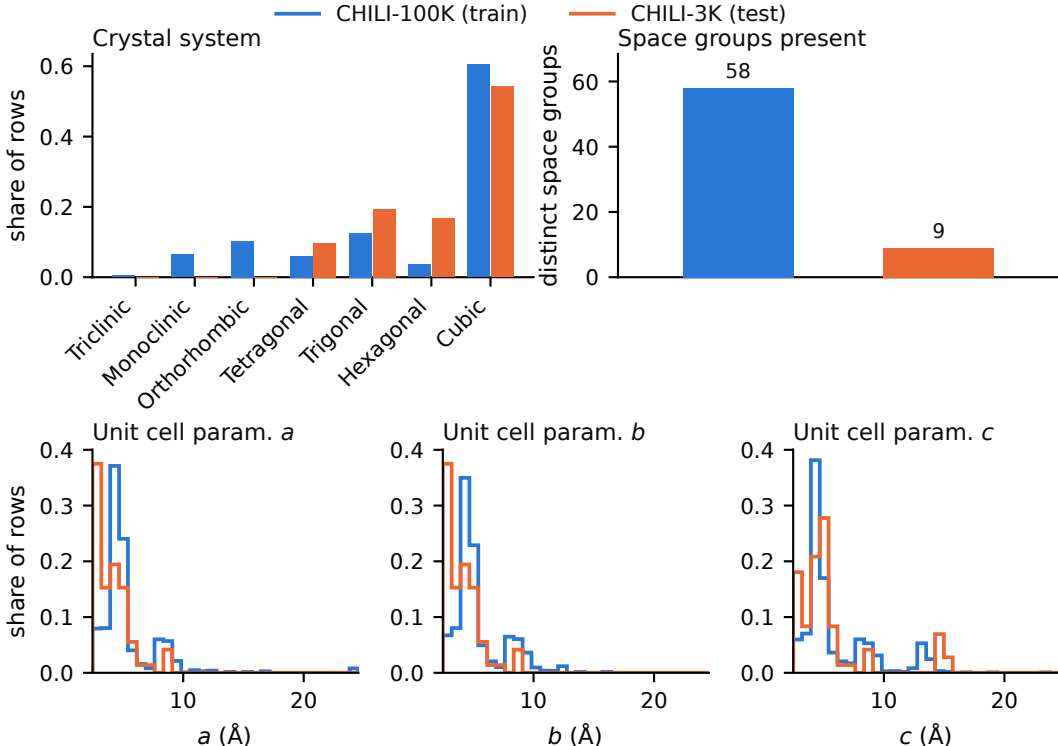


Figure 7: Structural coverage of the CHILI-100K training set and the CHILI-3K test set: crystal systems, number of distinct space groups, and unit-cell edges. CHILI-100K spans a broader range on every axis shown.

A.3.2 Targets

Every label is derived from the structure and none from the spectrum, so a label is unchanged by re-simulating a particle at new instrument parameters. Table 4 lists the four targets and Figure 8 their distributions over CHILI-3K. Nanoparticle size is the dataset’s own `np_size` field, in Å. The five sizes are nominal classes and the realized diameter is continuous, taking 2236 distinct values over 7.36 to 56.69, Å. M-O distance is the minimum metal-oxygen separation in the parent cell, taken from the distance matrix, and spans 1.455 to 3.316 Å. Coordination number is the number of neighbors that `CrystalNN` assigns to every metal site. Oxidation state is the formal metal charge implied by stoichiometry with oxygen at -2 , so that M_xO_y gives $2y/x$. The $8/3$ class is M_3O_4 and is a stoichiometric average rather than a valence at any single site.

The metal sites of a cell can disagree on coordination number, in which case the sample is left unlabeled for that target and retained for the other three. Only the Spinel prototype disagrees, at $\{4, 6\}$, which leaves 265 samples unlabeled and reduces coordination number to 2310 training and 340 test samples. The values that occur are $\{4, 6, 8\}$.

A.3.3 Training

In the probe scheme, we freeze the encoder and embed each spectrum once. Regression targets are read out by a closed-form least squares fit of the 256-dimensional representation plus a bias term, solved in double precision. Categorical targets are read out by a multinomial logistic head on standardized embeddings, with the scalar fitted on the train split alone. The probe never reads the validation split, having no hyperparameter to select on it.

Table 4: The four downstream targets.

Target	Metric	Definition
Nanoparticle size	R^2	Diameter of the nanoparticle, Å.
M-O Distance	R^2	Minimum metal-oxygen distance, Å
Coordination number	F_1	Number of oxygen neighbors shared by every metal site, 3 classes {4, 6, 8}
Oxidation state	F_1	Formal metal charge from stoichiometry, 5 classes {1, 2, 8/3, 4, 6}

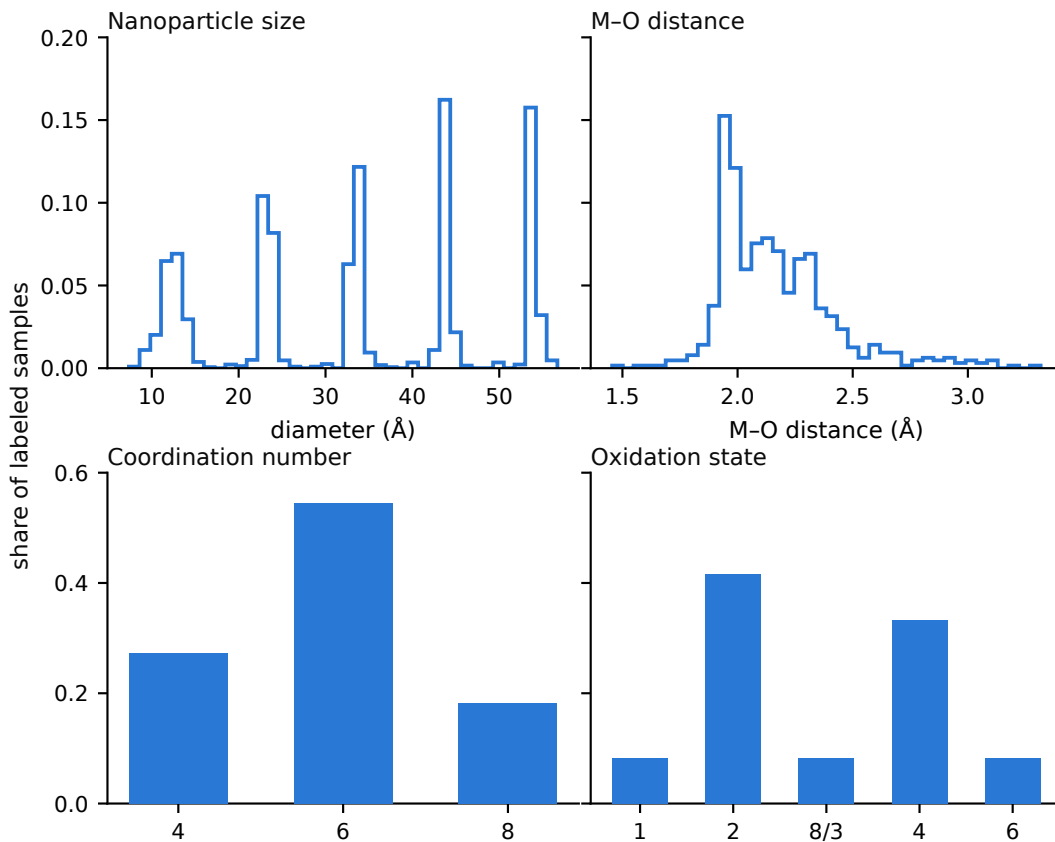


Figure 8: Distributions of the four downstream targets over CHILI-3K.

Plain least squares is used wherever the training set is larger than the representation dimension. This covers the experiments in Sections 3.1 and 3.3, both of which fit 2530 samples against 256 dimensions. The label efficiency sweep of Section 3.2 reaches down to 13 labeled samples, where the system is underdetermined and least squares returns the minimum norm solution, which interpolates the training set. Every point of that sweep is therefore fitted with ridge regression. The penalty is chosen by generalized cross validation over 97 log-spaced multiples of the design matrix’s mean squared singular value, with the effective degrees of freedom capped at half the number of training samples. The cap removes a degenerate branch of the criterion at small penalties and is inert whenever the training set is larger than the representation.

In the baseline scheme, we train the encoder and head together from random initialization with Adam, at one learning rate for both. The rate is searched over $\{3 \times 10^{-5}, 10^{-4}, 3 \times 10^{-4}, 10^{-3}, 3 \times 10^{-3}\}$ and selected on validation loss for nanoparticle size on the fixed spectra, with the test split never read. The selected rate is 10^{-3} and the minimum is interior to the grid. Training runs for at most 300 epochs at batch size 64 and stops early on validation loss after 15 epochs without improvement, restoring the best validation state rather than the last. No cell of the search reaches the epoch ceiling. The

same rate is used for the Section 3.3 baselines. Repeating the search on the CHILI-100K training sets selects 10^{-3} throughout, except for M–O distance on the varied spectra, where it selects 3×10^{-3} .

In the fine-tuning scheme, the encoder is initialized from the pretrained checkpoint and trained at a fixed learning rate of 1×10^{-5} , with the freshly initialized head at 10^{-4} . Epoch budget, batch size, patience and seed match the baseline. Neither rate was selected on this data. Both are carried unchanged from prior work, which keeps the comparison non-circular and points the search asymmetry against the pretrained scheme, so wherever fine-tuning wins, tuning it could only have widened the margin.

Every gradient-trained entry in Tables 5, 6 and 7 is the mean over three seeds, with the sample standard deviations in the lower block of each table. The seed varies the head initialization, the batch order and, where the labeled set is a subsample, the draw. Every learning-rate search runs at one seed and the selected rate is carried to the other two seeds. The pretraining run behind the probe and the fine-tuned scheme is a single run, so no entry resolves variation in the encoder. The probe is deterministic wherever it fits on the full labeled set, because a closed-form fit on a frozen encoder is fixed by its inputs. These entries are single runs, marked by a dash in the standard-deviation tables, and where the probe fits a subsample its spread comes from the draw alone.

A.3.4 Latent Space

Figure 9 shows the two leading principal components of the varied spectra of all 3,180 CHILI-3K particles, in three representation states: the normalized input $\nu(G(r))$ the encoder reads, the representation h of the frozen pretrained encoder, and that encoder fine-tuned on the fixed spectra as in Section 3, once per target, at the learning rates of Appendix A.3.3. Each panel is its own principal-component analysis, colored by one of the four targets of Table 4, with the variance explained by the two components printed in its corner. The input and pretrained states involve no labels, so their projection repeats down every column and only the coloring changes, while the fine-tuned projection is specific to its panel’s target and changes from row to row. The 265 samples without a coordination-number label (Appendix A.3.2) are drawn in gray in that row.

In the input state the two components carry 39% and 8% of the variance and order none of the targets. The pretrained state spreads the variance more evenly, at 18% and 13%, as the covariance term of Eq. 11 encourages, and an ordering on all targets is visible. After fine-tuning, each panel separates its own target more sharply. On nanoparticle size the first component alone carries 75% of the variance and splits the particles into the five nominal sizes.

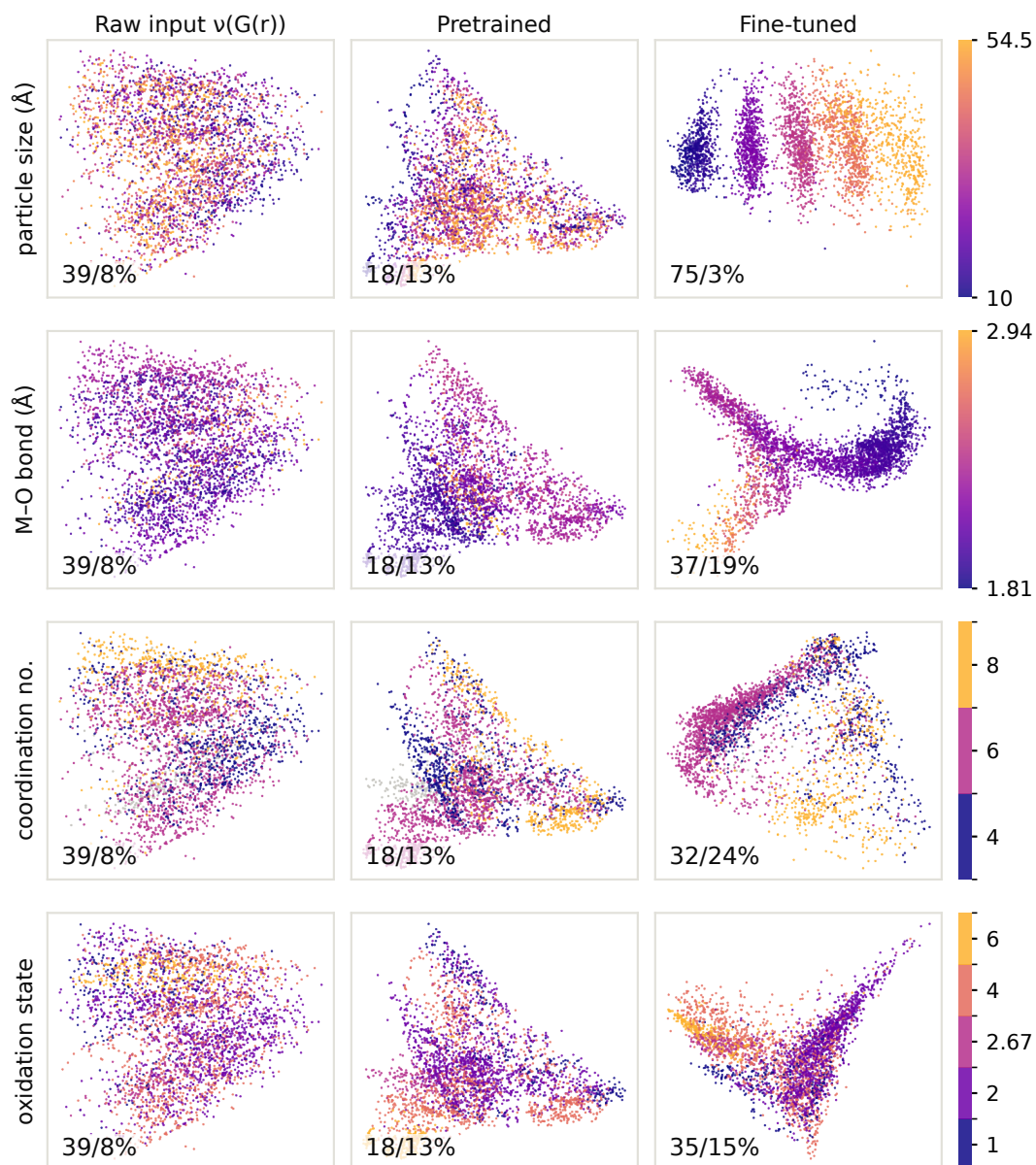


Figure 9: Three representation states of the varied CHILI-3K spectra. Each panel shows the two leading principal components of the same 3,180 particles, colored by the four targets of Table 4, as raw input (left), under the frozen pretrained encoder (center), and after fine-tuning on the panel's own target (right).

A.4 Results

A.4.1 Instrument shift

Figures 10 and 11 expand the top row of Figure 2 to the level of individual test samples. No downstream run stored per-sample output, so the predictions were generated again at a single seed. The probe reproduces its entries in Table 5 exactly. The baseline and fine-tuned panels are single training runs, so their annotations differ from the three-seed means in Table 5. The differences lie within the standard deviations reported there. Each panel’s annotation is recomputed from the points plotted in it, so it is exact for the data shown.

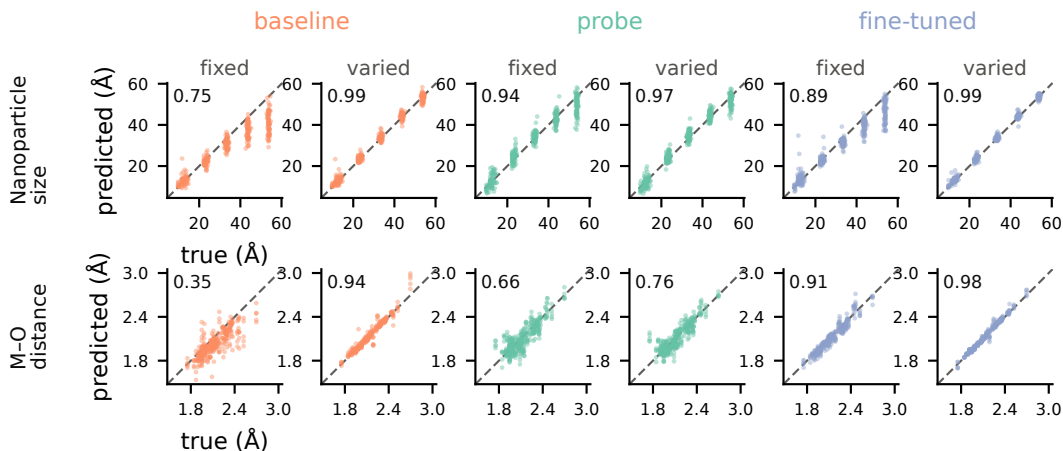


Figure 10: Parity for the regression targets. Predicted against true value for each scheme. Fixed and varied name the training set, with varied CHILI-3K as evaluation set throughout.

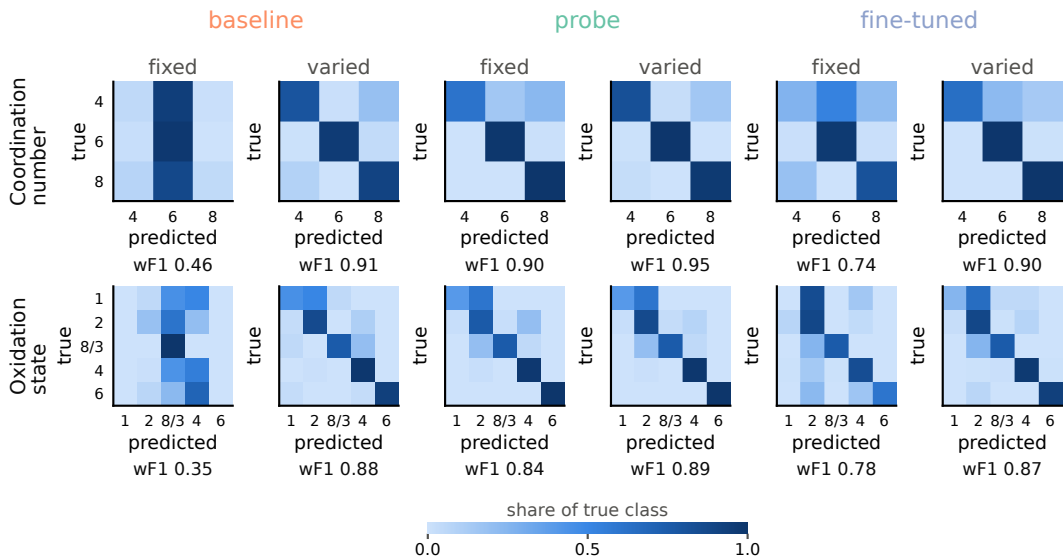


Figure 11: Confusion matrices for the categorical targets. True class by row, predicted class by column. Fixed and varied name the training set, with varied CHILI-3K as evaluation set throughout. Cells are normalized by true class.

Table 5: Exact numbers behind the top row of Figure 2. Test performance of each scheme trained on fixed and on varied spectra, as means over three seeds (upper block) with their standard deviations (lower block). A dash marks a deterministic single fit.

Scheme	Training labels	R^2		weighted F1	
		NP size	M–O dist.	Coord. no.	Oxid. state
baseline	fixed	0.770	0.434	0.555	0.421
	varied	0.991	0.962	0.943	0.891
probe	fixed	0.938	0.663	0.897	0.838
	varied	0.968	0.762	0.950	0.886
fine-tuned	fixed	0.877	0.849	0.743	0.690
	varied	0.995	0.980	0.919	0.872
<i>Standard deviations</i>					
baseline	fixed	0.003	0.213	0.092	0.077
	varied	0.002	0.030	0.008	0.024
probe	fixed	—	—	—	—
	varied	—	—	—	—
fine-tuned	fixed	0.006	0.065	0.006	0.073
	varied	0.000	0.002	0.022	0.002

A.4.2 Label efficiency

The probe trains a linear map on top of a frozen representation in which the targets are already linearly readable (Appendix A.3.4), so the labels only set the coefficients of that map, and ridge regularization keeps the fit stable when there are fewer samples than features. The baseline trains the encoder and the head at once, and at small label counts it scores at chance until the set is large enough to train both.

Table 6: Exact numbers behind the bottom row of Figure 2. Test performance as the labeled training set shrinks, as means over three seeds (upper block) with their standard deviations (lower block). Entries below the batch floor of Section 3.2 exist only for the probe.

scheme	labeled training samples								
	13	25	51	70	126	253	632	1265	2530
<i>Nanoparticle size</i>									
baseline	—	—	—	0.000	0.000	0.574	0.930	0.975	0.991
probe	0.501	0.699	0.861	0.885	0.931	0.949	0.960	0.967	0.970
fine-tuned	—	—	—	0.883	0.921	0.978	0.987	0.992	0.995
<i>M–O distance</i>									
baseline	—	—	—	−0.045	−0.045	−0.109	0.860	0.915	0.962
probe	0.543	0.501	0.589	0.728	0.763	0.693	0.744	0.772	0.782
fine-tuned	—	—	—	0.656	0.765	0.877	0.954	0.968	0.980
<i>Coordination number</i>									
baseline	—	—	—	0.418	0.418	0.521	0.825	0.861	0.943
probe	0.578	0.706	0.725	0.844	0.897	0.908	0.936	0.942	0.948
fine-tuned	—	—	—	0.589	0.631	0.814	0.859	0.887	0.919
<i>Oxidation state</i>									
baseline	—	—	—	0.281	0.266	0.455	0.732	0.854	0.891
probe	0.490	0.544	0.676	0.718	0.766	0.841	0.867	0.871	0.884
fine-tuned	—	—	—	0.560	0.658	0.769	0.801	0.847	0.872
<i>Standard deviations</i>									
<i>Nanoparticle size</i>									
baseline	—	—	—	0.001	0.001	0.500	0.015	0.002	0.002
probe	0.049	0.063	0.024	0.016	0.006	0.004	0.003	0.000	0.000
fine-tuned	—	—	—	0.034	0.057	0.002	0.002	0.001	0.000
<i>M–O distance</i>									
baseline	—	—	—	0.038	0.036	0.034	0.044	0.028	0.030
probe	0.065	0.234	0.255	0.081	0.061	0.095	0.033	0.027	0.000
fine-tuned	—	—	—	0.269	0.197	0.010	0.016	0.012	0.002
<i>Coordination number</i>									
baseline	—	—	—	0.000	0.000	0.178	0.060	0.026	0.008
probe	0.052	0.099	0.082	0.028	0.038	0.025	0.004	0.006	0.002
fine-tuned	—	—	—	0.156	0.194	0.037	0.029	0.016	0.022
<i>Oxidation state</i>									
baseline	—	—	—	0.088	0.036	0.223	0.021	0.014	0.024
probe	0.063	0.059	0.024	0.034	0.040	0.028	0.013	0.007	0.002
fine-tuned	—	—	—	0.030	0.025	0.062	0.019	0.019	0.002

A.4.3 Material shift

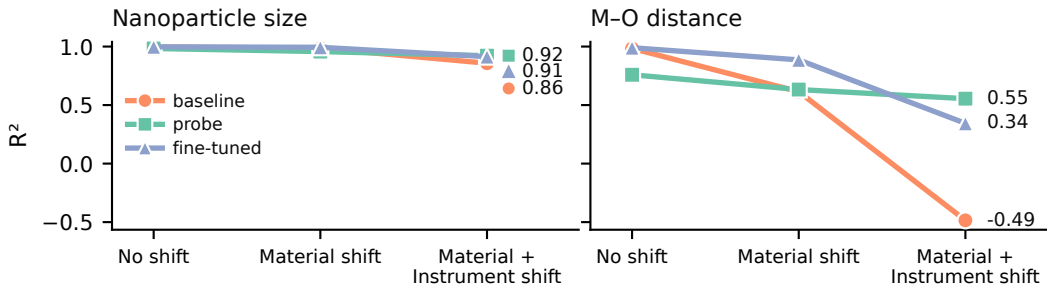


Figure 12: Material shift. Test R^2 on nanoparticle size and M–O distance on the varied CHILI-3K test set for the three training sets of Section 3.3. Points are means over three seeds (Table 7).

Table 7: Exact numbers behind Figure 12. Test performance on the two regression targets under its three training conditions, as means over three seeds (upper block) with their standard deviations (lower block). A dash marks a deterministic single fit.

Training scheme	Nanoparticle size (R^2)			M–O distance (R^2)		
	No shift	Material	+ Instrument	No shift	Material	+ Instrument
baseline	0.991	0.984	0.857	0.962	0.779	−0.485
probe	0.968	0.911	0.922	0.762	0.698	0.555
fine-tuned	0.995	0.990	0.914	0.980	0.790	0.344
Standard deviations						
baseline	0.002	0.005	0.042	0.030	0.052	0.295
probe	—	0.005	0.004	—	0.015	0.021
fine-tuned	0.000	0.002	0.009	0.002	0.079	0.183

A.5 Software and Data Availability

All datasets and simulation software used in this work are publicly available. Structures for pretraining come from the Materials Project [16], downstream nanoparticles from CHILI-3K and CHILI-100K [18], and the CHILI-100K parent structures originate from the Crystallography Open Database [20]. Pretraining spectra are simulated with the `diffpy.srreal` PDFCalculator [17] and downstream spectra with the `DebyeCalculator` [19]. Appendix A.1 specifies the simulation and augmentation pipeline, Tables 1 to 3 list every parameter, including the seed, needed to regenerate the training data, and the pretraining run. Code, configuration files and the pretrained encoder weights are available at <https://github.com/darinmerchant/MMXRay-thesis> under the MIT license.