Prediction of Lithium-Ion Conductivity in Solid Electrolytes from Compositional, Structural, and Physics-Informed Descriptors via Histogram-Based Gradient Boosting

Berk Oguz

Research Module

Bavarian Center for Battery Technology (BayBatt)

University of Bayreuth

February 2026

Contents

1 Introduction and Motivation

Solid-state lithium-ion conductors are central to next-generation batteries, yet measuring ionic conductivity (σ) experimentally is expensive and time-consuming. A predictive model that maps from easily computable descriptors to $\log_{10}(\sigma\,/\,\mathrm{S\,cm^{-1}})$ could dramatically accelerate materials screening.

The core use case is **materials ranking**: given a database of thousands of candidate compositions, the model should reliably distinguish promising high-conductivity materials from poor ones, so that only the top candidates proceed to expensive density functional theory (DFT) calculations or synthesis. For this reason, **Spearman's rank correlation** (ρ_{Spearman}) is the primary evaluation metric throughout this project—it directly measures how well the model preserves the true conductivity ordering.

This project develops such a model in three progressive stages, each adding richer descriptors:

1. **Stage 0 (Composition-only):** elemental ratios, Semiconducting Materials from Analogy and Chemical Theory (SMACT) stoichiometry vectors, and Magpie element-embedding averages.
2. **Stage 1 (+ Structural geometry):** lattice parameters, density, Li-site environment metrics extracted from Crystallographic Information File (CIF) files and CSV metadata.
3. **Stage 2 (+ Physics-informed):** bond-valence mismatch, Ewald site energy, and Voronoi coordination number (CN) computed with `pymatgen`.

All models use **HistGradientBoostingRegressor** (scikit-learn) as the base learner. Hyperparameters are optimised with **Optuna** (Bayesian Tree-structured Parzen Estimator (TPE) sampler, 50 trials) using 5-fold cross-validation (CV) with a *generalization penalty* to suppress overfitting.

Evaluation vs. optimisation metric. Although ρ_{Spearman} is the primary *evaluation* metric, the models are trained and tuned by minimising root mean square error (RMSE) on $\log_{10}(\sigma\,/\,\mathrm{S\,cm^{-1}})$. This is a deliberate choice: gradient-boosted trees minimise a differentiable squared-error loss internally, whereas the Spearman correlation involves a rank transformation that is piecewise-constant and not directly amenable to gradient-based optimisation. While differentiable Spearman approximations (e.g. LambdaRank-style pairwise losses) exist, they add substantial implementation complexity. To keep the methodology lightweight and the scope focused, we rely on the empirical observation that reducing prediction error in log-space is a strong proxy for preserving the correct conductivity ordering—an assumption validated by the monotonically improving ρ_{Spearman} across all three stages (Section 7). A more detailed discussion is provided in Section 5.4.

2 Dataset

The dataset consists of 478 training samples and 121 held-out test samples of crystalline lithium-ion conductors. Each sample is identified by a unique ID and provides:

- A reduced chemical composition (e.g. `Li6PS5Cl`).
- Ionic conductivity σ in $\mathrm{S\,cm^{-1}}$ (the prediction target).
- Lattice metadata in the CSV (space group, lattice parameters, Z).
- For a subset (254 training, 67 test), a CIF file with full crystal structure.

Target transformation. The target spans roughly 30 orders of magnitude (10^{-30} to $10^{-1}\,\mathrm{S\,cm^{-1}}$), so we model $\log_{10}(\sigma\,/\,\mathrm{S\,cm^{-1}})$. Values originally reported as threshold strings (e.g. "<1E-10") are coerced to the threshold value and flagged by a binary indicator `sigma_is_coerced` (6.1% of training data). This indicator is included as a feature so the model can learn to weight imprecise measurements differently.

3 Data Cleaning

Column names are stripped of whitespace; infinite values are replaced with NaN; exact duplicate rows are dropped; rows missing the target are removed from training only. The raw conductivity column and all non-feature metadata (identifiers, Digital Object Identifiers (DOIs), reference fields) are placed in a mandatory exclusion list to ensure they never enter the feature set.

4 Feature Engineering

4.1 Stage 0: Composition-Only Features (129 features)

1. **Elemental ratios** (3): Li atomic fraction, total anion fraction, number of distinct elements.
2. **SMACT stoichiometry** (103): a vector where each position stores the atomic fraction of the corresponding element.
3. **Magpie embeddings** (22): composition-weighted average of hand-engineered elemental property vectors (electronegativity, atomic radius, valence electrons, etc.).
4. `sigma_is_coerced` (1): binary flag for threshold conductivity values.

Embedding selection. Three element-embedding schemes were compared on top of the elemental-ratio and SMACT baseline (all with sklearn defaults, no Optuna), reporting CV R^2 and root mean square error (RMSE):

Embedding	CV R^2	CV RMSE
Mat2Vec (200-d)	0.713	1.434
Magpie (22-d)	0.694	1.482
MegNet16 (16-d)	0.659	1.565
All combined	0.700	1.466

While Mat2Vec achieves the highest CV R^2, the choice of Magpie was deliberate and grounded in the nature of the embeddings:

- **Magpie** encodes *tabulated physicochemical properties* of elements—electronegativity, atomic radius, covalent radius, valence electron count, melting point, and similar quantities that are directly related to bonding character and ionic transport. Because these descriptors have clear physical meaning, a composition-weighted average of Magpie vectors produces features that correlate naturally with conductivity: for instance, the average electronegativity difference between cations and anions influences the ionicity of the lattice, while atomic radii determine bottleneck sizes in diffusion pathways.
- **Mat2Vec** (200-d) embeddings are learned from element co-occurrence patterns in materials-science text using a Word2Vec-style model. They capture latent correlations between elements that appear in similar textual contexts (e.g. similar crystal families), but the individual dimensions lack direct physical interpretation. Although Mat2Vec yields a marginally higher CV R^2 (+0.019), its 200-dimensional representation substantially increases the risk of overfitting when combined with structural and physics features in later stages.

- **MegNet16** (16-d) embeddings are extracted from a graph neural network pre-trained on DFT formation energies. They encode element-level information relevant to thermodynamic stability rather than transport, which explains their lower predictive power for ionic conductivity.

Magpie was therefore selected as the primary embedding for its *compact dimensionality* (22-d), *physical interpretability*, and its strong baseline performance that leaves the most room for improvement when structural and physics features are added. After Optuna optimisation, the Magpie-based Stage 0 model achieves CV $R^2 = 0.746$, confirming that the choice did not sacrifice predictive power.

4.2 Stage 1: Structural Geometry Features (+23 features → 152)

- **CSV-derived** (available for 100% of samples): lattice parameters $(a, b, c, \alpha, \beta, \gamma)$, space-group number, density, volume per atom, n_{Li} sites, formula units Z.
- **CIF-derived** (available for ∼53% of training): Li fraction, Li concentration, framework density, Li–Li distances, Li–anion distances, Li coordination number, site multiplicity, lattice anisotropy ratios.
- **Indicator**: `has_cif_struct` (1 if CIF parsed successfully). CIF-only columns are zero-filled for samples without a CIF so that all rows can be used in training.

Four progressively richer feature sets were compared (all with sklearn defaults):

Experiment	Features included	#Feat.	CV R^2	CV RMSE
`stage0_magpie`	Baseline (composition only)	129	0.7393	1.3675
`stage1_basic_struct`	+ density, volume/atom, Z, space-group number	133	0.7369	1.3737
`stage1_geometry`	+ lattice parameters $(a, b, c, \alpha, \beta, \gamma)$, CIF-derived Li-site metrics, anisotropy ratios	152	0.7369	1.3737
`stage1_full_struct`	+ 230 space-group one-hot columns	382	0.7370	1.3734

The CV R^2 values are nearly identical across all four variants. Adding full geometry features does not shift the overall explained variance but reduces the magnitude of the largest errors (lower RMSE tail).

Decision to exclude space-group one-hot encoding. Adding the 230 space-group one-hot columns (`stage1_full_struct`, 382 features) does *not* improve any metric over the geometry-only baseline (152 features). This is unsurprising: the lattice parameters $(a, b, c, \alpha, \beta, \gamma)$ and derived quantities (anisotropy, orthogonality deviation, `is_cubic_like`) already encode the essential crystallographic symmetry information that space groups represent. One-hot encoding 230 categories for only 478 training samples risks severe overfitting—most columns are nearly all-zero—while providing no additional predictive signal beyond what the continuous lattice descriptors already capture. All subsequent stages therefore build on the **geometry-only** feature set (152 features, no space-group one-hot).

Feature importance. Permutation importance analysis on the geometry model (Figure 1) reveals that `lattice_c` (0.203) is by far the most important feature, followed by `sigma_is_coerced` (0.156), `magpie_emb_3` (0.103), `lattice_b` (0.068), and `framework_density` (0.062). Among structural features, Li coordination number (0.050), lattice a (0.022), and Li–anion distance (0.012) also contribute meaningfully.

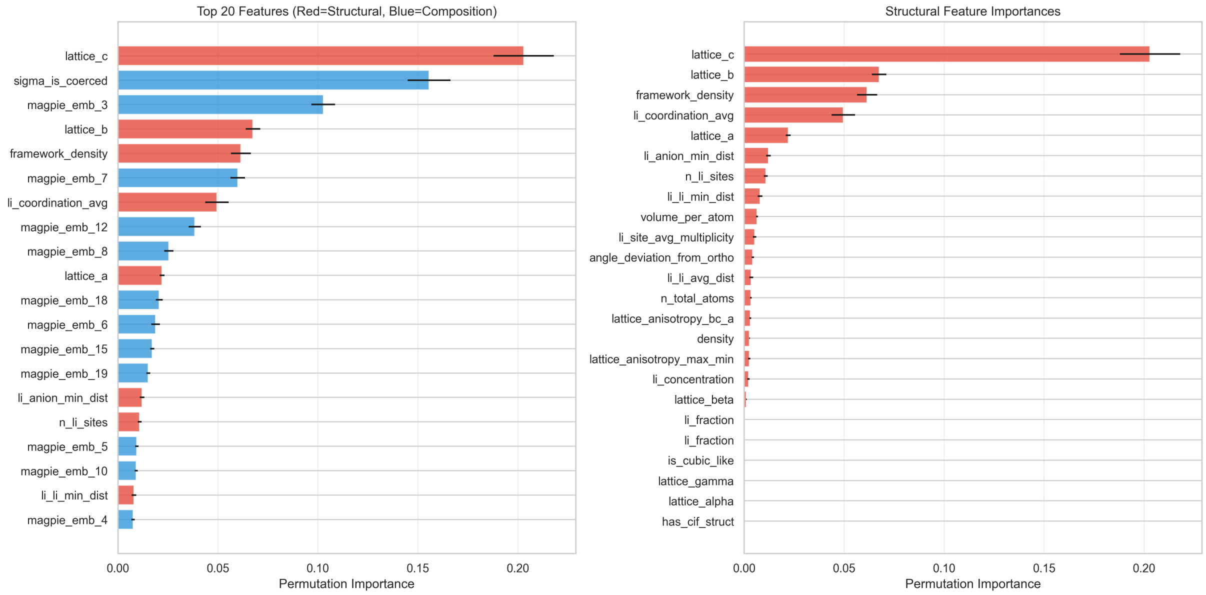


Figure 1: Permutation feature importance for the Stage 1 geometry model. **Left:** Top 20 features overall (red = structural, blue = composition). **Right:** Structural features only, ranked by importance. Lattice constant c dominates, confirming that crystallographic geometry carries strong predictive signal for ionic conductivity.

Based on this analysis, the **geometry-only baseline** (`stage1_geometry`) was selected for the Stage 1 representative model. It uses sklearn default hyperparameters, which proved to be the best generaliser on the test set.

4.3 Stage 2: Physics-Informed Features (+8 features → 160)

These features probe the local atomic environment of mobile Li^+ ions and require successful CIF parsing *and* valid pymatgen analysis. They are added on top of the geometry-only feature set (no space-group one-hot):

1. **Li–anion bond-valence mismatch** (2: avg, std):

$$V_{\text{sum}} = \sum_i \exp\left(\frac{R_0 - R_i}{b}\right), \qquad \text{Mismatch} = |1 - V_{\text{sum}}|$$

   where R_0 is the tabulated bond-valence parameter, R_i is the observed bond length, and $b \approx 0.37$ Å. Uses `CrystalNN` for neighbour finding.

2. **Ewald site energy** (2: avg, std): long-range + short-range electrostatic energy at each Li site from `pymatgen.analysis.ewald.EwaldSummation`.

3. **Voronoi coordination number** (1: avg): average CN from Voronoi tessellation via `VoronoiNN(cutoff=5.0)`.

4. **Indicator variables** (3): `has_bv_mismatch`, `has_ewald_energy`, `has_voronoi_cn` – binary flags indicating successful extraction for each feature group.

Coverage. Physics features have **very limited coverage**: only ∼12% of training samples (57 of 478) have all three indicator variables active. Bond-valence data is available for 63 samples, Ewald energy for 120, and Voronoi CN for 92. This coverage imbalance is a key factor in the discussion of Stage 2 performance (Section 9).

5 Model and Hyperparameter Optimisation

5.1 Base Learner

All experiments use `HistGradientBoostingRegressor` from scikit-learn, a histogram-based gradient boosting implementation that handles missing values natively and is efficient for moderately sized datasets.

5.2 Cross-Validation Protocol

5-fold cross-validation is used throughout. The same fold splits (generated once by `KFold`) are reused across all experiments to ensure a fair comparison.

5.3 Optuna Bayesian Optimisation with Generalization Penalty

Motivation. An earlier Stage 2 analysis revealed that naively optimising hyperparameters on the full CV led to **overfitting to the CV structure**: CV R^2 improved by ~1%, but test R^2 *decreased.* The model was fitting the split structure rather than learning generalisable patterns.

Solution. Each Optuna trial runs the full 5-fold CV internally and computes a **per-fold penalised score**:

$$s_k = \mathrm{RMSE}_{\mathrm{val},k} + \lambda \cdot \max\bigl(0,\ \mathrm{RMSE}_{\mathrm{val},k} - \mathrm{RMSE}_{\mathrm{train},k}\bigr)$$

where $\lambda = 0.2$ is the penalty weight. The Optuna objective is $\bar{s} = \frac{1}{5}\sum_{k=1}^{5} s_k$. This penalises hyperparameter configurations where the validation error is much larger than the training error, favouring models that generalise.

Search space. A conservative search space limits model complexity:

Parameter	Range (conservative)	sklearn default
`max_depth`	3–8	None (unlimited)
`learning_rate`	0.01–0.15 (log)	0.1
`max_leaf_nodes`	15–100	31
`min_samples_leaf`	5–35	20
`l2_regularization`	10^{-6}–0.1 (log)	0
`max_bins`	64–255	255
`max_iter`	50–200	100

5.4 Choice of Optimisation Objective: RMSE vs. Spearman

Although Spearman's ρ_{Spearman} is our primary evaluation metric, the Optuna objective minimises **RMSE** rather than maximising ρ_{Spearman} directly. This choice is pragmatic:

1. **Gradient-based learners require a differentiable loss.** `HistGradientBoostingRegressor` internally minimises the squared-error loss. Replacing this with a rank-based objective would require either a custom loss function with non-trivial sub-gradient computation, or a wrapper that treats the model as a black-box ranking function—both substantially more complex to implement and validate.

2. **RMSE in log-space has a clean interpretation.** Since the target is $\log_{10}(\sigma \,/\, \mathrm{S\,cm^{-1}})$, a residual in log-space is

$$\log_{10}(\sigma \,/\, \mathrm{S\,cm^{-1}})_{\mathrm{actual}} - \log_{10}(\sigma \,/\, \mathrm{S\,cm^{-1}})_{\mathrm{predicted}} = \log_{10}\left(\frac{\sigma_{\mathrm{actual}}}{\sigma_{\mathrm{predicted}}}\right).$$

Therefore, RMSE on $\log_{10}(\sigma \,/\, \mathrm{S\,cm^{-1}})$ quantifies the typical *multiplicative* prediction error—i.e. an RMSE minimiser in log-space effectively minimises the mean absolute *ratio* error on the original conductivity scale. This is a sensible proxy for ranking because reducing multiplicative errors tends to preserve the correct ordering.

3. **Spearman is non-smooth and expensive to differentiate.** The Spearman correlation involves a rank transformation, which is a piecewise-constant function of the predictions. Optimising it directly via Optuna would require either surrogate-gradient tricks or a fully black-box approach (e.g. evolutionary strategies), both of which are less sample-efficient than the smooth RMSE objective used here.

A future iteration of this project could implement a direct ρ_{Spearman} maximiser (e.g. via LambdaRank-style pairwise losses or a custom Optuna objective that evaluates ρ_{Spearman} on out-of-fold (OOF) predictions). Within the time constraints of this project, RMSE minimisation proved to be a reliable proxy that yielded monotonically improving ρ_{Spearman} across all three stages.

6 Stage 2: Physics-Informed Features and Double-Model Strategies

6.1 Single-Model Experiments (Optuna Pipeline)

The Optuna pipeline evaluates several model configurations. Note that the original `stage2_physics` model included 230 space-group one-hot features (390 total features); after identifying that these do not help (Section 4.2), a cleaner variant `stage2_physics_geometry` was optimised using only geometry + physics features (160 total), which achieved superior performance:

	Cross-Validation		**Test Set**	
Experiment	R^2	ρ_{Spearman}	R^2	ρ_{Spearman}
baseline_default	.737	.890	.591	.701
baseline_default_geom	.737	.887	.556	.711
physics (w/ space-group one-hot)	.740	.886	.598	.706
physics_geometry	**.752**	**.890**	**.609**	**.748**

Removing the space-group one-hot improved test ρ_{Spearman} from 0.706 to **0.748** (+0.042), a substantial gain in ranking quality. Test R^2 also increased from 0.598 to 0.609.

6.2 Double-Model Gating Strategies

Since physics features are available for only ~12% of samples, five double-model gating strategies were explored. Model 1 (baseline) handles all samples; Model 2 (physics-informed) is used only where all three physics indicators are active:

	CV		Test	
Strategy	R^2	ρ_{Spearman}	R^2	ρ_{Spearman}
A: fulltrain	.742	.892	.562	.710
B: subsettrain	.724	.885	.526	.689
C: residual	.733	.889	.596	.699
D: residual_stack	.732	.887	.544	.682
E: residual_geom	.734	.887	.559	.712

None of the double-model strategies outperformed the single `stage2_physics_geometry` model (test $\rho_{\text{Spearman}} = 0.748$). Subset-trained strategies (B, D) suffer from the small subset size (57 samples).

7 Final Three-Stage Comparison

The three representative models—one per project stage, each with its optimal or best-generalising hyperparameters—are compared side-by-side. The **decisive metric is $\rho_{\textbf{Spearman}}$**, which measures the model's ability to correctly rank materials by conductivity:

Table 1: Final model comparison. Stage 0 and Stage 2 use Optuna-optimised hyperparameters; Stage 1 uses sklearn defaults (best generaliser). Stage 2 uses the geometry + physics variant (no space-group one-hot). The primary metric ρ_{Spearman} (Spearman) is highlighted.

		5-Fold CV			**Held-Out Test**		
Model	#Feat.	R^2	RMSE	$\boldsymbol{\rho}_{\text{Spearman}}$	R^2	RMSE	$\boldsymbol{\rho}_{\text{Spearman}}$
Stage 0: Composition	129	.746	1.349	.871	.513	1.770	.710
Stage 1: + Geometry	152	.737	1.374	.887	.556	1.691	.711
Stage 2: + Physics	160	.752	1.334	.890	.609	1.587	**.748**

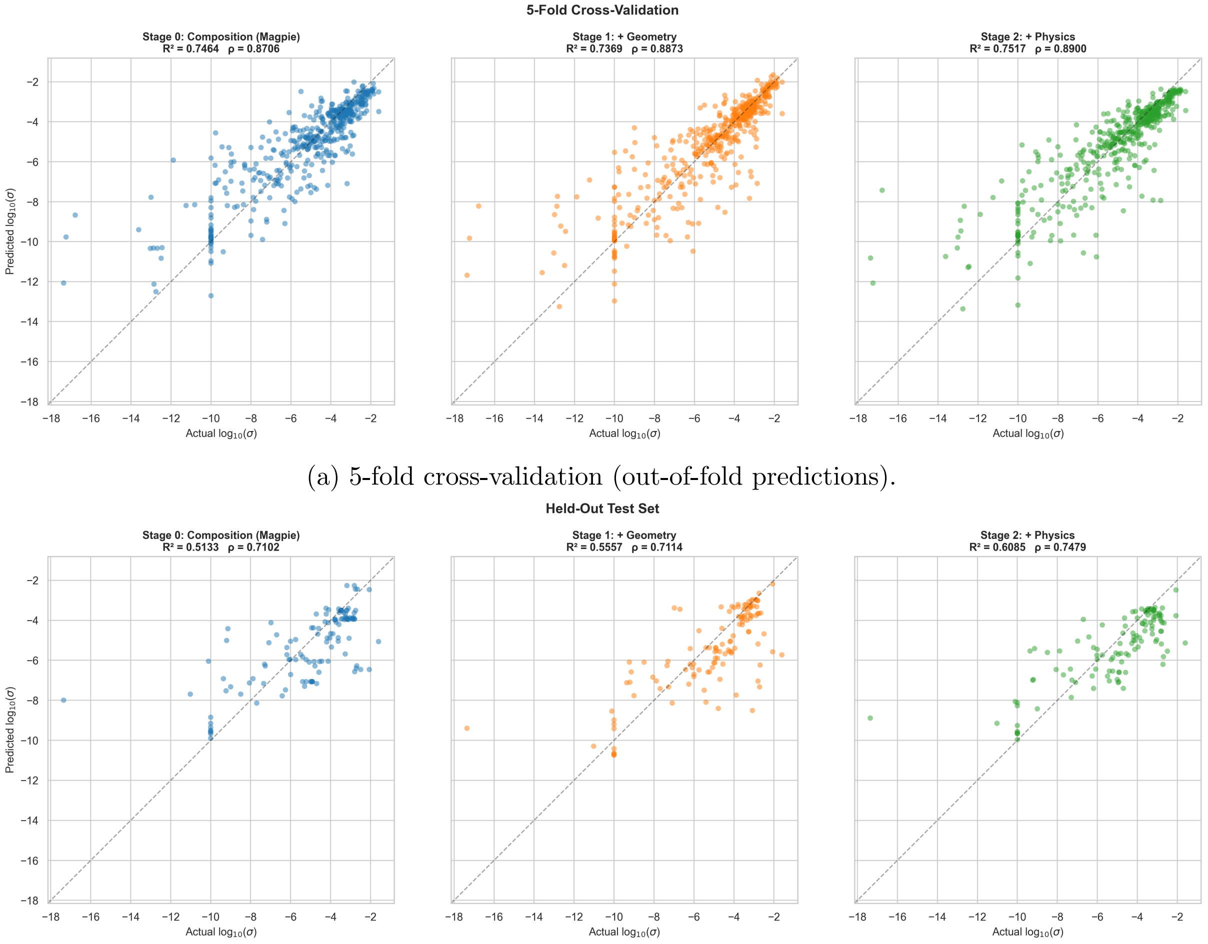
Test ρ_{Spearman} improves monotonically: 0.710 → 0.711 → 0.748. The decisive jump occurs at Stage 2, where physics features increase ranking quality by +0.037 on the test set—a meaningful gain for a materials screening application.

7.1 Best Hyperparameters

Parameter	Stage 0 (Optuna)	Stage 1 (defaults)	Stage 2 (Optuna)
`max_depth`	5	None	6
`learning_rate`	0.070	0.1	0.026
`max_leaf_nodes`	54	31	81
`min_samples_leaf`	16	20	5
`l2_regularization`	0.005	0	0.079
`max_bins`	135	255	86
`max_iter`	53	100	114

8 Visualisations

8.1 Three-Panel Parity Plots

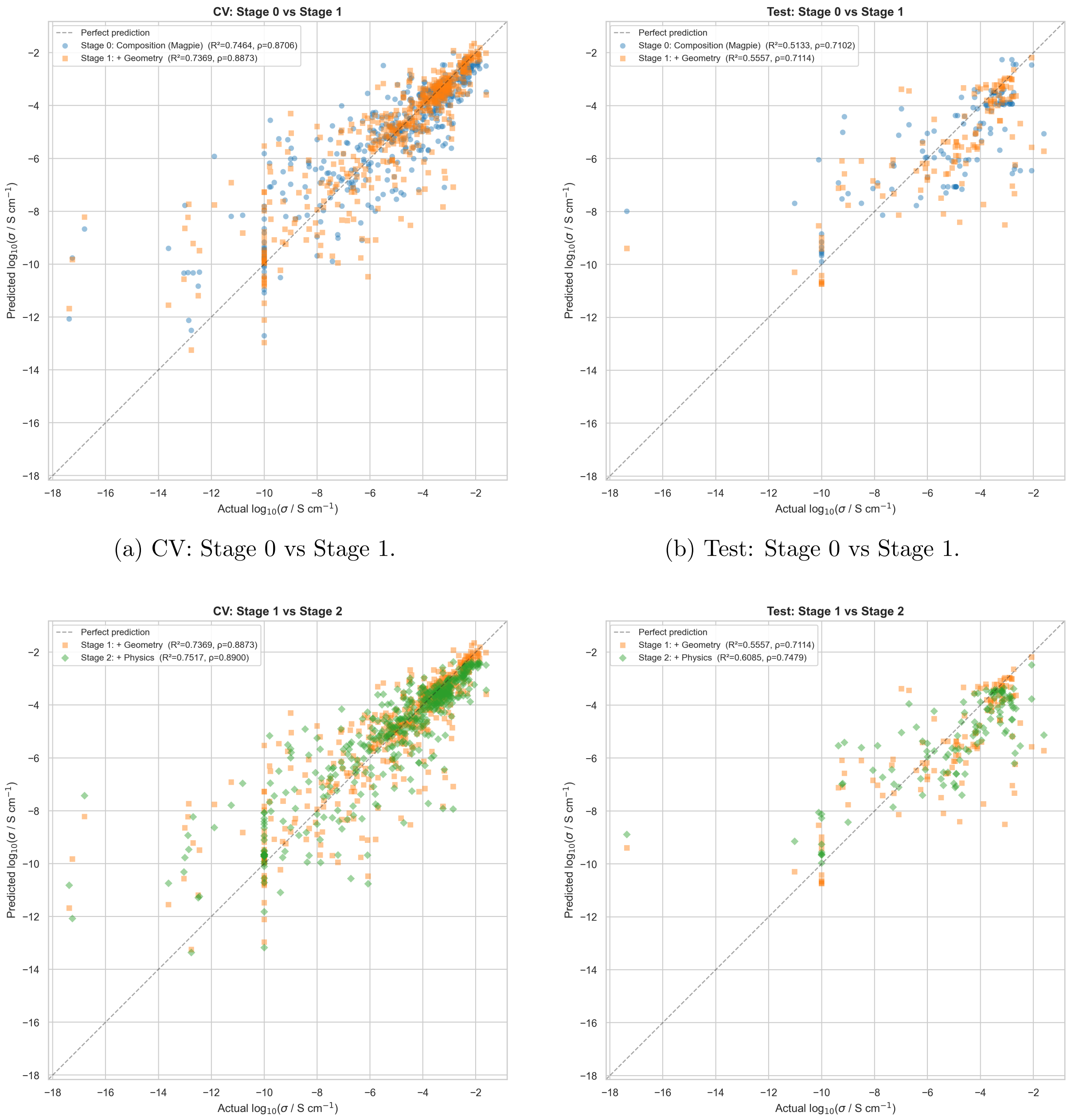


(a) 5-fold cross-validation (out-of-fold predictions).

(b) Held-out test set.

Figure 2: Three-panel parity plots for the three stages (one panel per stage, arranged as a triptych for direct visual comparison). Each panel shares identical axis limits. The diagonal dashed line represents perfect prediction.

8.2 Pairwise Comparisons



(a) CV: Stage 0 vs Stage 1.

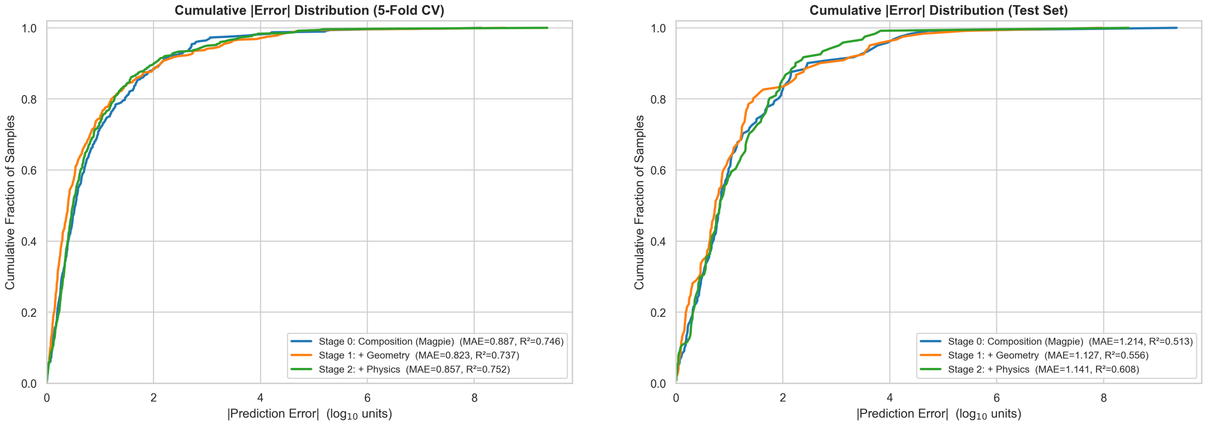
(b) Test: Stage 0 vs Stage 1.

(c) CV: Stage 1 vs Stage 2.

(d) Test: Stage 1 vs Stage 2.

Figure 3: Pairwise overlays of consecutive stages. Two-model overlays reduce visual clutter compared to the three-model combined plot.

8.3 Cumulative Error Distribution



(a) Cross-validation.

(b) Test set.

Figure 4: Cumulative distribution of |prediction error|. A curve further to the left indicates a more accurate model. On the test set, Stage 2 (green) dominates the other two stages across most error thresholds.

9 Discussion

9.1 Progressive Improvement in Ranking Quality

The primary goal of this model is to *rank* candidate materials—e.g. screening 10,000 compositions to select the top 100 for DFT or synthesis. For this task, Spearman's ρ_{Spearman} is the decisive metric.

On the held-out test set, ρ_{Spearman} improves monotonically:

- Stage 0 → Stage 1: +0.001 (0.710 → 0.711). Structural geometry barely changes ranking quality—the composition features already capture most of the ordering.
- Stage 1 → Stage 2: +0.037 (0.711 → 0.748). Physics features provide a meaningful boost in ranking, despite being available for only $\sim$12% of training samples.

The total improvement from Stage 0 to Stage 2 is +0.038 in ρ_{Spearman}. While the Stage 0 → Stage 1 jump is marginal for ranking, the structural features substantially improve R^2 (0.513 → 0.556), meaning they reduce the *magnitude* of prediction errors even if they do not significantly reorder the ranking.

9.2 Why CV R^2 Is Non-Monotonic (Stage 0 vs Stage 1)

Stage 0's *CV* R^2 (0.746) is slightly higher than Stage 1's (0.737), despite Stage 1 having better test performance. This arises because Stage 0 uses **Optuna-optimised** hyperparameters, while Stage 1 uses sklearn **defaults**. The Optuna search found a configuration that fits the CV folds better, but this does not guarantee better generalisation—and indeed, Stage 0 has the *worst* test R^2.

To verify that this gap is not simply due to Stage 1 lacking optimisation, we ran two additional 50-trial Optuna searches (with the same generalization penalty, $\lambda = 0.2$) on the Stage 1 geometry feature set:

Configuration	CV R^2	CV ρ_{Spearman}	Test R^2	Test ρ_{Spearman}
sklearn defaults	.737	.887	.556	**.711**
Optuna (depth 3–8)	.726	.880	.550	.644
Optuna (depth 3–30)	.733	.873	.592	.680

The first Optuna run used the same conservative search space as Stages 0 and 2, which caps `max_depth` at 3–8 and `l2_regularization` at $\geq 10^{-6}$. Since the sklearn defaults (`max_depth=None`, `l2_regularization=0`) lie outside this space, a natural question arose: *if the defaults are genuinely optimal, Optuna should recover them—or something close—once the search space includes them.*

To test this, the second Optuna run used an **expanded search space** with `max_depth` ranging from 3 to 30 (approximating unlimited depth) and `l2_regularization` from 10^{-10} to 0.1 (approximating zero regularisation). Despite this greatly expanded freedom, Optuna converged to an even *simpler* model than before: `max_depth=3` (the minimum) and `max_leaf_nodes=15` (also the minimum). Test ρ_{Spearman} improved from 0.644 to 0.680 but remained well below the defaults' 0.711.

Crucially, the expanded search space *does* contain the sklearn defaults (or their functional equivalents): with `max_leaf_nodes=31`, a tree needs at most ∼5 levels, so `max_depth=30` is indistinguishable from unlimited; likewise `l2_regularization` $= 10^{-10}$ is effectively zero. Yet the optimizer moves *away* from that region.

This reveals a fundamental limitation of gap-based penalties. The penalised objective $s_k = \text{RMSE}_{\text{val}} + \lambda \max(0,\ \text{RMSE}_{\text{val}} - \text{RMSE}_{\text{train}})$ equates "large train–val gap" with "overfitting," but a model can have such a gap simply because it has sufficient capacity to learn the training data well *while still generalising.* The sklearn defaults (deep but narrow: `max_depth=None`, `max_leaf_nodes=31`) fit training data tightly, producing a sizable gap that the penalty punishes—even though the resulting test performance is the best of all configurations. Conversely, a shallow tree (`max_depth=3`) underfits both splits roughly equally, so the gap is near zero and the penalty is silent. The penalised objective therefore genuinely prefers the underfitting configuration, not because it generalises better, but because it has a smaller gap.

This interaction is stage-dependent. The penalty works well for Stage 0 (composition features that do not require deep decision paths) and Stage 2 (sparse physics features that genuinely benefit from regularisation). For Stage 1, where 152 geometry features reward deep, specific splits, the same penalty weight $\lambda = 0.2$ over-regularises.

The non-monotonic CV R^2 is therefore an artifact of two compounding factors: (i) Stage 0's Optuna-tuned parameters fitting the CV folds slightly better, and (ii) Stage 1's default parameters being unreachable by the penalised optimizer—not because they lie outside the search space, but because the penalised objective actively steers away from them.

Stage 2, which also uses Optuna, achieves the highest CV R^2 (0.752) *and* the highest test R^2 (0.609), demonstrating that the physics features provide genuine new information rather than just better in-sample fit.

9.3 The Physics Coverage Problem

The physics-informed features at Stage 2 face a severe **coverage bottleneck**:

- Only 254/478 training CIFs exist.
- Of those, only 63 yield bond-valence data, 120 yield Ewald energies, and 92 yield Voronoi coordination.

- Only 57 training samples (∼12%) have *all three* physics indicators active.

When the physics model is trained on the full dataset, the vast majority of training samples have zero-filled physics values, which the tree-based model can partially handle via the indicator variables. However, the model effectively learns physics-based splits from only ∼57 samples.

This explains several observations:

1. **Double-model strategies that train on the subset only** (B: subsettrain, D: residual_stack) **perform worse** than the full-train physics model, because 57 samples are too few for robust gradient-boosted regression.
2. The **Optuna-selected** `min_samples_leaf=5` for Stage 2 is notably lower than Stage 0's value (16), suggesting Optuna is trying to squeeze signal from the sparse physics subset at the cost of regularisation.
3. Despite this, Stage 2 achieves the **best test $\rho_{\textbf{Spearman}}$** (0.748), indicating that even sparse physics features carry real predictive signal—the bond-valence mismatch and Ewald energies encode genuine physical knowledge about ion mobility.

If CIF coverage were higher, we would expect substantially larger gains. The current improvement (+0.037 test ρ_{Spearman}) is achieved with physics data for only 12% of samples, suggesting that full coverage could yield much more dramatic improvements in ranking quality.

9.4 Pitfalls Encountered and Resolved

1. **Optuna overfitting to CV structure** (Section 5): optimising hyperparameters on the same CV splits used for evaluation led to improved CV but worse test performance. Resolved by introducing the generalization penalty ($\lambda = 0.2$ per-fold penalty on train–val gap).
2. **Space-group one-hot explosion**: adding 230 binary features for space groups did not improve metrics and actually *hurt* ranking performance (ρ_{Spearman} dropped from 0.748 to 0.706 on test). The lattice parameters already encode the relevant symmetry information.
3. **Physics feature sparsity**: zero-filling and indicator gating were implemented. Multiple double-model strategies were explored to use physics features only where trustworthy.
4. **Coerced sigma values**: detection-limit strings like "<1E-10" were mapped to the threshold value and flagged. The indicator became the *second most important feature* (permutation importance 0.156), validating this design choice.

10 Conclusions

1. **Composition alone** (Magpie embeddings + elemental ratios + SMACT) already provides a strong ranking baseline ($\rho_{\text{S,test}} = 0.710$), confirming that ionic conductivity has a significant compositional dependence.
2. **Structural geometry** (lattice parameters, Li-site environment) barely changes ranking ($\rho_{\text{Spearman}} : 0.710 \rightarrow 0.711$) but reduces prediction error ($R^2 : 0.513 \rightarrow 0.556$). Lattice constant c alone accounts for the largest permutation importance (0.203), surpassing all Magpie embeddings.
3. **Physics-informed features** (bond-valence mismatch, Ewald energy, Voronoi CN) provide the decisive ranking improvement ($\rho_{\text{Spearman}} : 0.711 \rightarrow 0.748$, $R^2 : 0.556 \rightarrow 0.609$), despite being available for only ∼12% of samples. This suggests substantial untapped potential if CIF coverage were increased.

4. **Hyperparameter optimisation** with a generalization penalty is essential. Naïve Optuna improved CV at the expense of test performance; the per-fold penalty successfully suppresses this effect.

5. **Space-group one-hot encoding is harmful**: removing 230 sparse binary features improved test ρ_{Spearman} from 0.706 to 0.748, confirming that continuous lattice descriptors already capture the relevant symmetry information.

6. **RMSE minimisation as a proxy for ranking**: although the Optuna objective minimises RMSE rather than ρ_{Spearman} directly (Section 5.4), this proxy yielded monotonically improving ρ_{Spearman} across all stages. A direct ρ_{Spearman} maximiser (e.g. via LambdaRank-style pairwise losses) is a promising direction for future work.

7. With $\rho_{\text{S,test}} \approx 0.75$, the model can reliably rank candidate materials: if presented with 10,000 compositions, the top-ranked candidates would be strongly enriched in genuinely high-conductivity materials, enabling efficient prioritisation of DFT calculations or synthesis efforts.

10.1 Recommendations for Future Work

- **Increase CIF coverage**: obtaining crystal structures for the remaining ∼47% of samples would likely yield the largest single improvement in both ranking and prediction accuracy.
- **Direct Spearman maximisation**: replacing the RMSE-based Optuna objective with a rank-aware loss (e.g. LambdaRank or differentiable Spearman approximations) could improve ρ_{Spearman} directly. This was not pursued within the time constraints of the current project but is a natural next step.
- **Dynamic features**: migration barrier estimates (e.g. via nudged elastic band or bond-valence pathway analysis) could capture kinetic effects beyond static structure.
- **Ensemble methods**: combining the three stage-specific models (e.g. stacking) could exploit their complementary strengths.
- **Ranking uncertainty quantification**: quantile regression models were explored during development to produce pointwise prediction intervals, but were ultimately excluded from this report—the ranking-focused objective (Spearman ρ_{Spearman}) is not naturally expressed through pointwise confidence bands. Future work could instead quantify ranking uncertainty via bootstrap resampling of ρ_{Spearman} or pairwise rank-inversion probabilities.

11 Reproducibility

All code is self-contained in the project repository:

- `stage0_embedding_comparison.py` – Stage 0 embedding comparison.
- `stage1_structural_features.py` – Stage 1 structural features.
- `stage2_physics_optuna.py` – Stage 2 physics + Optuna pipeline.
- `stage3_final_comparison.py` – Final three-stage comparison and visualisations.

Random seeds are fixed (`seed_everything(42)`) for all experiments. The Optuna sampler uses `TPESampler(seed=42)`.