Robust model benchmarking and bias-imbalance in data-driven materials science: a case study on MODNet

Pierre-Paul De Breuck©, Matthew L Evans© and Gian-Marco Rignanese∗

Université catholique de Louvain (UCLouvain), Institute of Condensed Matter and Nanosciences (IMCN), Chemin des Étoiles 8, B-1348 Louvain-la-Neuve, Belgium

E-mail: gian-marco.rignanese@uclouvain.be

Received 3 February 2021, revised 23 June 2021
Accepted for publication 8 July 2021
Published 29 July 2021

Abstract
As the number of novel data-driven approaches to material science continues to grow, it is crucial to perform consistent quality, reliability and applicability assessments of model performance. In this paper, we benchmark the Materials Optimal Descriptor Network (MODNet) method and architecture against the recently released MatBench v0.1, a curated test suite of materials datasets. MODNet is shown to outperform current leaders on 6 of the 13 tasks, while closely matching the current leaders on a further 2 tasks; MODNet performs particularly well when the number of samples is below 10 000. Attention is paid to two topics of concern when benchmarking models. First, we encourage the reporting of a more diverse set of metrics as it leads to a more comprehensive and holistic comparison of model performance. Second, an equally important task is the uncertainty assessment of a model towards a target domain. Significant variations in validation errors can be observed, depending on the imbalance and bias in the training set (i.e., similarity between training and application space). By using an ensemble MODNet model, confidence intervals can be built and the uncertainty on individual predictions can be quantified. Imbalance and bias issues are often overlooked, and yet are important for successful real-world applications of machine learning in materials science and condensed matter.

Keywords: machine learning, data-driven materials science, uncertainty, bias, property prediction

(Some figures may appear in colour only in the online journal)
on the composition alone [8]. Databases of reported synthesised compositions were screened for new superconductors, yielding 35 non-cuprate and non-ferrous oxides as candidate compositions. These examples, just two among many [4, 5], show the impact of ML techniques in materials science and condensed matter, and as such it is considered as the fourth paradigm of materials science [9].

Many ML approaches are being actively developed to accurately predict compound–property relationships. They differ in the amount, type and richness of descriptors and model complexity, as well as adopting other learning paradigms such as transfer or multi-task learning. As a result, they exhibit very different characteristics, and no universally superior algorithm exists [10, 11]. Firstly, the input type and thus applicability can be very different; some models are architecturally restricted to specific descriptors, e.g., graph models require atomic coordinates to make predictions [12, 13]. Secondly, the ability for models to generalise (i.e., the error on unseen samples) strongly depends on the amount of available data, with different approaches succeeding in contrasting data-poor or data-rich regimes. Often, a model or architecture that performs very well on a large dataset can be disappointing when applied on a small dataset (and vice versa) [14, 15].

Given the number and variety of approaches being adopted, it is important that the field follows a standard benchmarking pipeline, inspired by other fields (e.g., ImageNet for computer vision [16]). Too often, models are tested on only a small number of datasets, with unclear or unrepeatably validation and testing procedures. Significant variation in generalisation performances can be observed for different hold-out sets, as will be shown. To draw a fair comparison between the two competing models, they should not just be benchmarked on the same dataset, but also cross-validated on the same training and testing subsets (typically, a 5-fold cross-validation with fixed random seed). Similarly, hyperparameter choices are often opaque and should instead follow a reproducible pipeline. Every model has advantages and disadvantages and testing a diverse range of tasks shows that no model is universally superior, but rather performance of a given architecture is a function of the amount of data, feature availability, the complexity of the target space and the computational resources available for tuning. These reasons motivate the creation of a standard benchmark. Dunn et al recently released the MatBench v0.1 test suite [15] which consists of 13 materials science-specific prediction tasks, ranging from small (∼300) to large (∼150 000) datasets, both experimental and computational, with target properties covering a wide range of materials phenomena (mechanical, electronic, and thermodynamic), with both regression and classification tasks.

Convinced by this standardised approach, we apply the MatBench v0.1 benchmark to the recently reported Material Optimal Descriptor Network (MODNet) [14] developed by some of the present authors. Overall, the performance of MODNet is quite competitive; out of the 13 tasks, MODNet is able to outperform (by more than 2%) the current leader on 6 tasks or closely match it (within ±2%) on a further 2 tasks. Of the remaining 6 tasks, the 4 largest datasets were not attempted as it has been shown that graph networks will dominate in this regime [14]. For the phonon DOS task, MODNet performs around 5% worse than current leader when trained using automated hyperparameter optimisation, but this can be improved (and reversed) if the models are hand-tuned post hoc. As such, MODNet provides a promising alternative on small to medium-sized datasets to the automated but complex pipelines trained with Automatminer (AM) [15] and the data-intensive graph convolutional neural networks CGCNN [12] and MEGNet [13].

Furthermore, our testing reveals a variance in the generalisation error between cross-validation folds. In the second part of this paper, our aim is to predict this (sometimes abrupt) change in prediction error by using an ensemble of MODNet models, with a discussion of the importance of imbalance and bias in the dataset, and how model performance can be appropriately reported within these different regimes. This paves the way to a better understanding of a model’s applicability for a particular target domain, allowing the estimation of the prediction error of unseen samples.

2. Methods

The MODNet is an open source framework for predicting materials properties from primitives such as composition or structure [14]. It consists of a feedforward neural network fed with a limited number of descriptors, derived from chemical, physical, and geometrical considerations (see appendix A). MODNet was designed to make the most efficient use of data for tasks where large, internally consistent training sets are prohibitively difficult or expensive to obtain; we observe that most existing material datasets are limited in this way, particularly those derived from experimental results. In order to have the best possible performance at low size, three key aspects are needed.

First, our observation showed that chemical and geometrical descriptors can provide more information than a raw graph representation. To this end, MODNet makes use of many of the existing descriptors designed in the community as implemented in the Matminer package [17]. By giving the model descriptors derived from chemical, physical, and geometrical considerations, part of the learning is already done as they exploit existing chemical knowledge. In contrast, more complex models such as graph networks or embedding methods are often initialised with only atomic numbers and bond lengths [12, 13]. Graph models are able to bootstrap this chemical knowledge ex nihilo, and perform highly accurate property predictions but only when a large (∼10^4) and sufficiently diverse dataset is available.

Second, the smaller the dataset, the greater the importance of feature selection for tackling the curse of dimensionality [18]. For instance, a previous application of MODNet to vibrational thermodynamics [14] showed that an average 12% improvement in error can be obtained by applying careful feature selection. For feature selection, MODNet employs an iterative procedure based on a relevance–redundancy (RR) criterion (measured through the normalised mutual information between all pairs of features, and between features and targets), see appendix A.
Third, combined models that integrate other information in the form of transfer learning, multi-task/joint learning, multi-fidelity methods [6, 19], or any other form, can mitigate the lack of data for a given target. Indeed, when a model combines different properties for the same compound, a more general representation can be formed in the hidden layers that more efficiently uses the information provided to the learning algorithm and increasing model performance for a given dataset size. With MODNet, one can learn on multiple properties by using a tree-like architecture, see figure A1 in appendix A. The similarity between target properties can be used to decide where the tree splits, i.e., the layer up to which properties share an internal representation. This approach is particularly effective when learning parametric data that exhibit, for example, temperature or pressure dependence.

MODNet has been trained and tested on both single and multi-property tasks with excellent performance compared to contemporary methods for datasets containing 10 000 samples or fewer. For instance, on the refractive index dataset of Naccarato et al [20] (4040 samples), MODNet achieves a mean absolute error (MAE) of 0.051 (n.b., this dataset is distinct from MatBench refractive index task). A multi-target MODNet was also developed to study temperature-dependent results of the vibrational thermodynamics dataset of Petretto et al [21] (1265 samples). A single model was simultaneously trained on the vibrational entropy, enthalpy, specific heat and Helmholtz energy across a wide temperature range. This model achieved an average error of 0.009 meV/K/atom on the room temperature vibrational entropy ($S^\circ$), four times lower than previous studies, with joint learning contributing to an error reduction of 8% compared to the best MODNet model trained on $S^\circ$ alone [14].

Since the initial release (v0.1.5) of MODNet as reported in reference [14], several improvements to the software framework have been implemented in the version used in this paper (v0.1.10).

First, an ensemble MODNet model has been implemented and used for this work. It consists of a set of MODNet models, all trained on the same task, but with independently optimised hyperparameters, training set and initial weights. In particular, we choose to take the 5 best architectures for each inner training fold. Moreover, multiple copies of each individual architecture are trained via bootstrap resampling of the given training set fold, resulting in a 125-model ensemble. It is based on the deep ensemble framework [22], and the bootstrapping technique, that has been showed to have consistent advantage in out-of-domain epistemic uncertainty calibration and generalization error when applied on the chemical space [23]. This final model has the advantage to predict a distribution of values, from which statistics such as the mean and standard deviation can be computed. The mean over the models is used as final prediction, while the standard deviation is used as an epistemic uncertainty measure. The descriptors used by MODNet are not necessarily numerically robust across the space of all possible compositions and crystal structures and thus could lead to unphysical results for particular rare representations (e.g., unit cells with very acute angles). To mitigate this, predictions from the final model that fall considerably outside of the range, $r = y_{\text{max}} - y_{\text{min}}$ of target values present in the training set (i.e., those outside of $r$ padded by 25%) are replaced with a value drawn from a uniform distribution spanned by $r$. This makes the ensemble more robust; the remapping will have minor impact on the overall ensemble on average, but the variance will still be penalised with the number of pathological cases.

Second, MODNet has been made increasingly flexible in the nature of the material primitives supported (a full structure or just a composition) and on the types of target properties that can be learned. For instance, the user can manually choose which Matminer featureizers [17] to apply to the data, or they can use presets suitable for structural or purely compositional samples. Third, it is now possible to use MODNet to perform binary and multi-label classification tasks, predicting class membership probability in both cases. Finally, an automated procedure for hyperparameter optimisation has been added within the framework of nested cross-validation. Automated hyperparameter selection ensures a correct assessment of the generalisation error and minimises overfitting. That is, hyperparameters should never be chosen to minimise a test error, but rather chosen internally with a validation procedure. The specific hyperparameter optimization process used in this work is described in appendix B.

MODNet is an evolving framework and further improvements are planned. We expect to apply it on even more datasets, as well as applying the trained models to accelerate materials design and discovery. The source code and issue tracker, as well as several featurized datasets and pretrained benchmarks can be found at ppdebreuck/modnet on GitHub [24].

3. Results and discussion

3.1. MatBench benchmarking

The MatBench (v0.1) test suite contains 13 supervised ML tasks, taken from 10 datasets with size varying from 300 to 130 000. Our previous benchmarking of MODNet has shown that it is most effective on small to medium-sized datasets, so efforts were focussed on the 10 tasks with fewer than 15 000 samples. Of the remaining tasks, the largest two consist of the entire Materials Project [25]; the scaling of MODNet’s performance on this dataset was previously studied in detail with the conclusion that graph-based networks outperform MODNet by a factor of 2 when the entire dataset is considered [14].

The MatBench suite covers various observables in materials science driven by different length scales, from microscopic simulations of elastic, electronic, optical and vibrational properties, to macroscopic measurements of steel yield strengths and metallic glass formation. For a detailed description of the contents and curation of each dataset, the reader is referred to the MatBench paper itself [15].

Each task provides either a composition or a structural model per sample. The target is always a single property, either continuous or a discrete class label (such as metal or non-metal). Almost all properties are therefore trained in single-target mode, i.e., one target per task. The only exceptions are the elastic properties $\log_{10} K$ and $\log_{10} G$, where both bulk and shear modulus are provided for each sample.
Table 1. The performance of MODNet (v0.1.10) on the MatBench (v0.1) test suite alongside alternative algorithms (AM, RF, CGCNN and MEGNet) reported by Dunn et al [15]. The reported scores are either MAEs for regression or the area under the receiver-operator curve (ROC-AUC) for classification (indicated by †). These scores are averaged over the best models for each of the five cross-validation folds, following the recommendations of MatBench [15]. The hyperparameters are allowed to vary across these folds. Numbers in bold indicate the best performing model for that task, and those within 2% of the best performance relative to the best model. A column is included for a null (‘dummy’) model that yields the mean of the dataset for each prediction for regression tasks, or returns the class frequency for classification tasks.

| MatBench task                | n  | MODNet | AM   | RF   | CGCNN | MEGNet | Dummy |
|------------------------------|----|--------|------|------|-------|--------|-------|
| Steel yield strength (MPa)   | 312| 96.2   | 95.2 | 104  | —     | —      | 230   |
| $E_{\text{exfol}}$ (meV/atom) | 636| 34.5   | 38.6 | 49.9 | 49.2  | 55.9   | 67.3  |
| argmax(PhDOS) (cm$^{-1}$)    | 1265| 38.75  | 50.8 | 68   | 57.8  | 36.9   | 324   |
| Exp. band gap (eV)           | 4604| 0.347  | 0.416| 0.446| —     | —      | 1.14  |
| Refractive index              | 4764| 0.297  | 0.299| 0.421| 0.599 | 0.478  | 0.809 |
| Exp. metallicity†             | 4921| 0.970  | 0.92 | 0.917| —     | —      | 0.495 |
| Glass-forming ability†        | 5680| 0.931  | 0.861| 0.858| —     | —      | 0.495 |
| log$_{10} E$ (log$_{10}$ GPa) | 10987| 0.0548 | 0.0679| 0.081| 0.0712| 0.0712 | 0.289 |
| log$_{10} G$ (log$_{10}$ GPa) | 10987| 0.0731 | 0.0849| 0.104| 0.0895| 0.0914 | 0.293 |
| Perovskite $E_{\text{form}}$ (eV/atom) | 18 928| —     | 0.194| 0.235| 0.0452| 0.0417 | 0.566 |
| Band gap (eV)                 | 106 113| —     | 0.282| 0.345| 0.228 | 0.235  | 1.33  |
| Metallicity†                  | 106 113| —     | 0.909| 0.9  | 0.954 | 0.977  | 0.502 |
| $E_{\text{form}}$ (eV/atom)  | 132 752| —     | 0.173| 0.116| 0.0332| 0.0327 | 1.01  |

Figure 1. Comparison of model performance across regression tasks, relative to a dummy model that predicts the mean of the dataset. The shaded region indicates the range of errors spanned by the MODNet cross-validation folds. An arbitrary vertical offset has been applied to each model to aid visual discrimination.

(4)
tasks relative to the dummy model across all datasets. One can
see that the dummy algorithm is outperformed by all models by
a factor of two to three. The spread between the performance
on individual folds is correlated both with the improvement
relative to the dummy model, but also to the spread between
all competing approaches. It should be noted that hand-tuned
models after automatic hyperparameter optimisation can also
lower this error further. This effect was most prominent for the
steel yield strength task where model performance could be
improved by nearly 20% with manual tuning, indicating that
the hyperparameter optimisation approach could be improved
for smaller datasets where a simple grid search is inherently
noisier.

Detailed MODNet benchmarking plots for each dataset can
be found in appendix C. These plots display the typical regres-
sion between target and predicted properties, and additionally
a regression between targets and prediction error to show any
model biases as a function of target property. As one might
expect, the sample density is often considerably lower for
extremal target property values, so many datasets display a
positive correlation between the errors and targets (i.e., predic-
tions decrease in quality for larger target values). This effect is
most prominent for the refractive index and exfoliation energy
datasets shown in figures C1 and C5.

As a general trend, it can be observed that MODNet per-
forms relatively well on tasks that are limited to composi-
tion data only. We explain this success by the usage of a
limited input neural network where a latent representation is
learned for the different elemental contributions; and moreover
acts as a dimensionality reduction technique. This representa-
tion is much more flexible than the equivalent obtained by
feature engineering for tree-based approaches. Recently pre-
vented deep representation learning approaches should be able
to exploit this effect even further, especially if transfer learning
is used [37, 38].

3.2. Model quality and uncertainty assessment

When predicting material properties with a ML model, one
should keep in mind that the prediction error can vary sig-
ificantly from material to material. The MAE is merely an
aggregate statistic, with sometimes huge differences (lower
or higher) on single predictions. For instance, concerning the
experimental band gap, the errors span 5 orders of magnitude
from $10^{-4}$ to $10^1$. This has two major consequences. First, a
more comprehensive way of assessing a model’s performance
is needed. Particularly, when comparing different models, it is
generally worth having a compilation of metrics or a full dis-
tribution of errors instead of relying on an individual metric.
Second, the ability to provide an error estimate on predictions
a priori is extremely useful, so one could be tempted to seek to
quantify this uncertainty. In other words, given the input ma-
terial, how reliable is the final prediction? This section covers
a solution to the first question, and proposes a simple distance
metric in feature space to address the second question.

3.2.1. Model quality assessment. The results in table 1 are an
average over the 5 cross-validation folds, and significant vari-
ations between folds are often observed, as shown in figure
1. This is caused by the underlying spread in errors over the
individual predictions, and when sampled, a strong discrep-
ancy between folds can be seen, especially when the dataset is
small. This variance over the points contains valuable informa-
tion about the intrinsic model performance, and should, in our
opinion, not be neglected. Too often in the materials science
field, models are assessed using a single metric, which is in
general not enough to fully capture and compare ML models.
Two models could have a similar MAE, but with a very differ-
ent spread. Moreover, models can set their internal parameters
to suit the benchmark metric (e.g., by setting the loss function
the metric of interest), and therefore optimise their model with
respect to a benchmark rather than the domain of application.
A more holistic approach to compare models is clearly needed.
The advantage of using different metrics is that they illuminate
different performance aspects of a model, and taken together,
they give a more comprehensive view of a model’s quality.
Following the work of Vishwakarma et al [39], we provide a
compilation of metrics to assess MODNet performance.

We suggest the following metrics for regression: MAE, med-
ian absolute error, root-mean-squared error (RMSE), mean
absolute percentage error (MAPE), maximum absolute
error (MaxAE), and the Pearson correlation ($R$) of an ordi-
ary least squares regression between the target and predicted
properties. The corresponding values for the different tasks on
MODNet are given in table 2 (excluding MAPE).

Comparing the MAE with the RMSE offers insights on
the variability of prediction errors. Large fluctuations in errors
(outliers) will result in a significantly higher RMSE. All tasks
in table 2 have a higher RMSE than MAE, indicating a right
skewed distribution of the absolute errors. This is especially
the case for the refractive index and exfoliation energy.

The MAPE provides an error relative to the ground truth,
and complements thus the MAE and RMSE. It should be
noted, however, that it is best avoided to properties having val-
ues equal or close to zero, as the MAPE diverges. It is therefore
omitted in 2.

The MaxAE gives the worst error in the test set, and is thus
related to the spread in errors. When making finite-cost deci-
sions based on predictions from a model, it is crucial to have
an idea of the worst case error. As can be seen from the table,
all tasks have a MaxAE that is orders of magnitude larger than
the MAE among the test samples. This aspect and its relation
uncertainty is discussed hereafter.

The $R$ value measures how close predictions are to the
ground truth by measuring their linear relationship, and has the
advantage to be bounded between −1 and 1. A value of 1 indi-
cates a perfect positive correlation. This enables to compare
performance not only between models but also between tasks.
Best performing tasks are found to be the phonons and elas-
tic constants. In contrast, the refractive index and exfoliation
energy are both seen to yield a low $R$ (0.43 and 0.65 respec-
tively). This is mainly caused by outliers, as the $R$-value metric
is very sensitive to them.

Finally, an alternative solution to the ensemble of metrics,
is to provide a full probability distribution of errors, by for
instance applying a Gaussian kernel density estimation on the
Both the ROC and PR curves are given in appendix C for the quality of the model for a given threshold. A confusion matrix is a simple yet powerful way of visualizing MODNet. The corresponding ROC-AUC and average performance at different class probability thresholds. Providing the full ROC curve is also good practice. However, for imbalanced datasets, precision-recall (PR) curves have greater utility [40]. Both the ROC and PR curves are given in appendix C for the experimental metallicity and glass-forming ability tasks using MODNet. The corresponding ROC-AUC and average precision scores are given in Table 2. For multi-class classification, a confusion matrix is a simple yet powerful way of visualizing the quality of the model for a given threshold.

3.2.2. Uncertainty prediction: the bias-imbalance issue. Beyond benchmarking a model, it is equally important to assess the uncertainty of novel predictions. Given an unseen sample, how reliable is the model’s prediction? This aspect is closely related to the applicability and generalisation capabilities of a model. This is not a straightforward problem in materials science and it should be given significant attention, in the same way much effort is spent on optimizing an error metric. For instance, a MODNet model was created on the MODNet [14].

Concerning classification, the previously mentioned ROC-AUC is an overall good metric for binary classification provided the labels are balanced, combining the model’s performance at different class probability thresholds. Providing the full ROC curve is also good practice. However, for imbalanced datasets, precision-recall (PR) curves have greater utility [40]. The concepts of imbalance and bias are important in this context. A training set that has two or more distinct classes (i.e., clusters in the material space) that are sampled in different proportions is imbalanced. For instance, 84% of the refractive index dataset is comprised of oxides and therefore constitutes an imbalanced training set. A training set is biased when it covers only a specific region (i.e., not all classes are covered) of the material space, a more extreme case of imbalance. Both imbalance and bias have consequences to the generalisation of a model as they express potential dissimilarities between training and target domain.

In order to visualise imbalance-bias and the discrepancy in error between oxides and non-oxides for the refractive index dataset of Naccarato et al [20], we performed a PCA decomposition. The first three components together account for 25% of the variance. Their detailed description are provided in appendix D, with the list of descriptors and corresponding weights in the feature space. The first feature roughly corresponds to a mean of all features and therefore identifies outlier compounds in the feature space. They are not of

| Regression task         | MAE  | Median AE | RMSE | MaxAE | R  |
|-------------------------|------|-----------|------|-------|----|
| Steel yield strength (MPa) | 96.2 | 60.8      | 151.9| 931   | 0.87|
| $E_{\text{el}}$ (eV/atom) | 34.5 | 9.05      | 102.8| 1535  | 0.65|
| Refractive index        | 0.297| 0.0612    | 1.90 | 58.9  | 0.43|
| Exp. band gap (eV)      | 0.347| 0.080     | 0.75 | 9.9   | 0.86|
| log_{10} K (log_{10} GPa) | 0.0548| 0.028 | 0.104| 1.5   | 0.96|
| log_{10} G (log_{10} GPa) | 0.0731| 0.049 | 0.110| 1.7   | 0.96|
| argmax(PhDOS) (cm$^{-1}$) | 38.75| 20.7      | 79.4 | 1032  | 0.99|

| Classification task     | ROC-AUC | Average precision |
|-------------------------|---------|-------------------|
| Exp. metallicity        | 0.970   | 0.968             |
| Glass-forming ability   | 0.931   | 0.966             |

*As in table 1, one numerically unstable prediction was replaced with the dataset mean before reporting the metrics for the log_{10} K and log_{10} G tasks.
Figure 2: PCA decomposition of the refractive index dataset (Naccarato et al. [20]) for the second and third component. The second component is linked to the bond-length, while the third component is linked to the ionicity of the compound. Data points corresponding to oxides and non-oxides are coloured blue and red respectively. Compounds having at least one oxygen atom tend to have shorter mean bond lengths and increased ionicity. An imbalance appears due to sampling: the dotted blue line divides the sample space in denser and coarser regions. The compounds in the well-sampled centre region yield on average the lowest error, regardless of whether they are an oxide. PCA components are provided in detail in appendix D.

Figure 3 represents the MAE as a function of the confidence percentile for different MatBench tasks. The MAE of a confidence percentile $p$ is computed by removing the $p\%$ most uncertain predictions, following different strategies. Randomly removing points (in red) forms a baseline, while a perfect ranked strategy based on the absolute errors forms a lower limit. The uncertainty provided by the ensemble MODNet model is depicted in blue. In addition, a strategy based on distances in the feature space is added. The idea is to determine a model’s applicability domain via the distance of a query point and the $k$-nearest neighbours (KNN) in the training set, $d_{KNN}$. The space of optimal descriptors, as selected by MODNet is used. Here, after various tests, the cosine distance is used for simplicity with $k = 5$. Other distances are possible, although one should be cautious with the high dimensionality [42].

Both the ensemble’s uncertainty and $d_{KNN}$ distance are adding value to distinguish error magnitudes with respect to the random procedure. The ensembling techniques is seen to be performing particularly well, especially at the lower end of the confidence percentiles. Moreover, assuming Gaussian distributions, 95% confidence intervals can be built for each prediction from the predicted standard deviation. Figures in appendix C depict confidence intervals for the different predictions as well as calibration curves, and various error related scatter plots. In particular, good agreement between predicted and observed test-error distribution are found. This shows that the ensemble MODNet model is effective at providing uncertainty estimates. Note that recalibration of the model (i.e., remapping the predicted standard deviations based on the observed errors) could further improve the accuracy.

The improvement of $d_{KNN}$ w.r.t. random ranking confirms that feature space imbalance is indeed an important factor in assessing uncertainty. However, it has clear limitations. First, it is clear that it does not perform as well as the probabilistic ensemble method. Especially, for the experimental band gaps a poor performance is found. Second, the $d_{KNN}$ method cannot provide any quantitative uncertainty estimate such as confidence intervals. Therefore, the ensembling technique is preferred for uncertainty estimations.

Finally, it should be noted that beyond ensemble-based methods, there are other model architectures that intrinsically...
Figure 3. Confidence error curves for six different regressions task found in MatBench: steel yield strength, 2D exfoliation energy, refractive index, experimental band gap, phonon DOS peak and bulk modulus. Each curve represents how the MAE changes when test points are sequentially removed following different strategies. A baseline (red) is generated by randomly removing points one at a time (as if all points had equal confidence), with the shaded area showing the deviation across 1000 trials. The randomly ranked error (red) forms a baseline with the shaded area representing the standard deviation over 1000 random runs. The error ranked curve (green), where the highest error is removed sequentially, represent a lower limit. The std-ranked strategy follows the uncertainty predicted by the ensemble MODNet, while the $d_{KNN}$ ranking is based on the 5-nearest neighbour cosine distance between each test point and training set.

4. Summary

To summarise, we report benchmarks of the Materials Optimal Descriptor Network (MODNet) on the MatBench v0.1 test suite. MODNet is a universal model, in regard to both the inputs (material structure or composition) and the target type. MatBench is a standardised benchmarking pipeline, with fixed procedures on training and testing, containing various material prediction tasks of differing sizes and types. Our results show that MODNet performs well on small to medium-sized datasets. In particular, it was found to outperform or match the current leaders on the experimental band gap, experimental metallicity, glass formation ability, 2D exfoliation energy, elastic moduli (bulk and shear), and phonon DOS peak estimation (see table 1). We provide metrics beyond MAE and ROC-AUC scores and encourage the reporting of multiple metrics and we make available all benchmarking data for future model comparisons.

We show that depending on the test set sampling (or fold), significant variation in error can be measured. This is due to the high dimensionality of the feature space spanned by materials and the associated sparsity of a given dataset. Bias or imbalance is easily introduced in the training set by this sampling. Therefore, applicability through uncertainty assessment is a crucial aspect for future developments. We show that dataset imbalance can be examined through dimensionality reduction.
techniques (such as PCA) and that uncertainty can be quantified by ensemble-based methods. This offers the possibility to set an confidence bound on individual predictions; uncertain predictions could be flagged or removed.

Finally, we emphasise that this work only forms a basis to a better practice in model design and performance assessment. Uncertainty quantification, and particularly for extrapolations, is a difficult task in general. We showed some options to this regard, but further development and benchmarking is necessary.

We hope that this work encourages future developments on the topics of metrics, bias and uncertainty that are crucial in materials science for robust, transparent testing and better understanding of a model applicability in a wider context. By providing all of our benchmarking data, we hope that the differences between competing approaches can be learned from and used to improve real-world model performance.

Acknowledgments

PPDB and GMR are grateful to the F.R.S.-FNRS for financial support. MLE and G-MR acknowledge support from the European Union’s Horizon 2020 research and innovation programme under the European Union’s Grant Agreement No. 951786 (NOMAD CoE).

Computational resources were provided by the supercomputing facilities of the Université catholique de Louvain (CISM/UCL) and the Consortium des Equipements de Calcul Intensif en Fédération Wallonie Bruxelles (CÉCI) funded by the Fond de la Recherche Scientifique de Belgique (F.R.S.-FNRS) under convention 2.5020.11 and by the Walloon Region.

Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: https://github.com/ml-evs/modnet-matbench.

Appendix A. The MODNet model

This appendix briefly summarizes the MODNet model. For further details, the reader is referred to reference [14].

The model, named MODNet and illustrated in figure A1, consists in a feedforward neural network with an reduced set of optimal descriptors (based on a criterion detailed further). Moreover, we propose an architecture that, if desired, learns on multiple properties, with good accuracy. This makes it easy to predict more complex objects such as temperature-, pressure-, or energy-dependent functions.

The structure (or composition) is first transformed to descriptors based on physical, chemical, and geometrical properties. They fulfill a number of constraints such as rotational, translational and permutational invariances. Moreover, features driven by physical and chemical intrinsically contain knowledge (w.r.t. more flexible graph representations) that facilitates learning when dealing with limited datasets. We rely on a large amount of features previously published in the literature, that were centralized into the Matminer project [17]. In order to reduce redundancy and therefore limit the curse of dimensionality [18] we rely on a feature selection process based on the normalized mutual information (NMI). It is defined as,

$$\text{NMI}(X, Y) = \frac{\text{MI}(X, Y)}{(\text{H}(X) + \text{H}(Y))/2} \quad \text{(A.1)}$$

with MI the mutual information, computed as described in reference [45] and H the information entropy (H(X) = MI(X,X)). The NMI, which is bounded between 0 and 1, provides a measure of any relation between two random variables X and Y. It goes beyond the Pearson correlation, which is parametric (it makes the hypothesis of a linear model) and very sensitive to outliers.

Given a set of features $\mathcal{F}$, the selection process for extracting the subset $\mathcal{F}_S$ goes as follows. When the latter is empty, the first chosen feature will be the one having the highest NMI with the target variable $y$. Once $\mathcal{F}_S$ is non-empty, the next chosen
Figure B1. Graphical depiction of the network architectures sampled during hyperparameter optimisation. Absolute values correspond to the layer depths for a 1000-dimensional feature space (before feature selection).

feature \( f \) is selected as having the highest RR score:

\[
RR(f) = \frac{\text{NMI}(f, y)}{\max_{f',c \in F} \left(\text{NMI}(f', f_0)\right)} + c
\]

where \((p, c)\) are two hyperparameters determining the balance between RR. In practice, varying these two parameters dynamically seems to work better, as redundancy is a bigger issue with a small amount of features.

The selection proceeds until the number of features reaches a threshold which can be fixed arbitrarily or, better, optimized such that the model error is minimized. When dealing with multiple properties, the union of relevant features over all targets is taken.

Secondly, we take the advantage of learning on multiple properties simultaneously. This could be used, for instance, to predict temperature-curves for a particular property.

In order to do so, we use the architecture presented in figure A1. Here, the neural network consists of successive blocks (each composed of a succession of fully connected and batch normalization layers) that split on the different properties depending on their similarity, in a tree-like architecture. The successive layers decode and encode the representation from general (genome encoder) to very specific (individual properties). Layers closer to the input are shared by more properties and are thus optimized on a larger set of samples, imitating a virtually larger dataset. These first layers gather knowledge from multiple properties, known as joint-transfer learning [46]. This limits overfitting and slightly improves accuracy compared to single target prediction.

Appendix B. Notes on featurization and training

All datasets were featurized using the DeBreuck2020 Featurizer preset that is bundled with MODNet, though some datasets only made use of compositional descriptors. After featurization, the normalised mutual information was computed between all pairs of descriptors across the entire dataset and was referred back to when training each model.

Hyperparameter optimization was performed via 5-fold cross-validation, with 85% of each fold being used for training in the inner loop. For each fold, a grid search over batch sizes, learning rates, number of training features \((N)\) and hidden layer depths was performed, and the hyperparameters with the lowest MAE on the remaining 15% of the fold were used to fit a new model on the entire fold. Feature selection is key to the performance of MODNet; the features were selected per fold based on a RR criterion and the top \( N \) features were used for training. Some typical architectures are illustrated in figure B1 for a 1000-dimensional feature space.

The outer loop was performed in parallel, yielding five models per task. After featurization, training can be efficiently performed on commodity hardware, with a full grid search (20–60 hyperparameter combinations) requiring no more than a 2 h per outer fold when run on 4-cores of an AMD EPYC 7742 (64-core) CPU.

All benchmarking data and the scripts used to featurize and train models for each dataset are available on GitHub at ml-evs/modnet-matbench [47].

Appendix C. Detailed benchmark plots

C.1 matbench_dielectric

See figures C1 and C2.

C.2 matbench_expt_gap

See figures C3 and C4.
Figure C1. Results for the matbench_dielectric dataset.

Figure C2. Uncertainty results for the matbench_dielectric dataset.

C.3. matbench_jdft2d
See figures C5 and C6.

C.4. matbench_log_gvrh
See figures C7 and C8.

C.5. matbench_log_kvrh
See figures C9 and C10.

C.6. matbench_phonons
See figures C11 and C11.
Figure C3. Results for the matbench_expt_gap dataset.

Figure C4. Uncertainty results for the matbench_expt_gap dataset.

C.7 matbench_steels

See figures C13 and C14.

C.8. matbench_expt_is_metal

See figure 15.
Figure C5. Results for the matbench_jdft2d dataset.

Figure C6. Uncertainty results for the matbench_jdft2d dataset.
Figure C7. Results for the matbench_log_gvrh dataset.

Figure C8. Uncertainty results for the matbench_log_gvrh dataset.
Figure C9. Results for the matbench_log_kvrh dataset.

Figure C10. Uncertainty results for the matbench_log_kvrh dataset.
Figure C11. Results for the matbench_phonons dataset.

Figure C12. Uncertainty results for the matbench_phonons dataset.
Figure C13. Results for the matbench_steels dataset.

Figure C14. Uncertainty results for the matbench_steels dataset.
Figure C15. Results for the matbench_expt_is_metal dataset.

Figure C16. Results for the matbench_glass dataset.

Table D1. 20 highest contributing features of PC1 with corresponding weights.

| $w_{ij}$ | Feature $f_{ij}$            |
|----------|-----------------------------|
| -0.0890  | ChemEnvSiteFingerprint mean SC:12 |
| -0.0890  | ChemEnvSiteFingerprint mean SH:11 |
| -0.0890  | ChemEnvSiteFingerprint std dev S:10 |
| -0.0890  | ChemEnvSiteFingerprint mean DD:20 |
| -0.0890  | ChemEnvSiteFingerprint mean H:10 |
| -0.0890  | ChemEnvSiteFingerprint std dev CO:11 |
| -0.0890  | ChemEnvSiteFingerprint mean S:12 |
| -0.0890  | ChemEnvSiteFingerprint mean S:10 |
| -0.0890  | ChemEnvSiteFingerprint mean CO:11 |
| -0.0890  | ChemEnvSiteFingerprint std dev SH:11 |
| -0.0890  | ChemEnvSiteFingerprint std dev S:12 |
| -0.0890  | ChemEnvSiteFingerprint std dev H:10 |
| -0.0890  | ChemEnvSiteFingerprint std dev DD:20 |
| -0.0890  | ChemEnvSiteFingerprint std dev SC:12 |
| -0.0890  | ChemEnvSiteFingerprint mean H:11 |
| -0.0890  | ChemEnvSiteFingerprint mean HD:9 |
| -0.0890  | ChemEnvSiteFingerprint mean SH:13 |
| -0.0890  | ChemEnvSiteFingerprint std dev H:11 |
| -0.0890  | ChemEnvSiteFingerprint mean PCPA:11 |
| -0.0890  | ChemEnvSiteFingerprint mean TBSA:10 |
Table D2. 20 highest contributing features of PC2 with corresponding weights.

| $w_{2,i}$ | Feature $f_{2,i}$                                                                 |
|----------|---------------------------------------------------------------------------------|
| -0.1070  | GaussianSymmFunc|| mean G2_20.0                                                   |
| -0.1053  | AGNIFingerPrint|| mean AGNI eta=1.23e+00                                         |
| -0.1049  | GeneralizedRDF|| mean Gaussian center=1.0 width=1.0                           |
| 0.1047   | ElementProperty||MagpieData mean Row                                                 |
| 0.0996   | VoronoiFingerprint|| mean Voro dist minimum                                  |
| -0.0993  | GeneralizedRDF|| mean Gaussian center=0.0 width=1.0                           |
| -0.0973  | AGNIFingerPrint|| std_dev AGNI eta=1.23e+00                                      |
| 0.0966   | AverageBondLength|| mean Average bond length                                    |
| -0.0965  | AGNIFingerPrint|| mean AGNI eta=1.88e+00                                       |
| 0.0962   | ElementProperty||MagpieData mean Number                                           |
| -0.0949  | GeneralizedRDF|| std_dev Gaussian center=0.0 width=1.0                           |
| -0.0948  | GaussianSymmFunc|| std_dev G2_20.0                                               |
| 0.0948   | ElementProperty||MagpieData mean CovalentRadius                                        |
| 0.0945   | ElementProperty||MagpieData mean AtomicWeight                                       |
| -0.0877  | GaussianSymmFunc|| mean G4_0.005_4.0_-1.0                                       |
| -0.0876  | AGNIFingerPrint|| std_dev AGNI dir=y eta=1.23e+00                                 |
| -0.0864  | AGNIFingerPrint|| std_dev AGNI dir=x eta=1.23e+00                                 |
| -0.0859  | GaussianSymmFunc|| mean G2_4.0                                                |
| -0.0848  | AGNIFingerPrint|| std_dev AGNI dir=y eta=1.88e+00                                 |
| -0.0845  | GeneralizedRDF|| std_dev Gaussian center=1.0 width=1.0                           |

Table D3. 20 highest contributing features of PC3 with corresponding weights.

| $w_{3,i}$ | Feature $f_{3,i}$                                                                 |
|----------|---------------------------------------------------------------------------------|
| -0.1194  | IonProperty|| avg ionic char                                                    |
| -0.1174  | ElementProperty||MagpieData avg_dev Electronegativity                                  |
| -0.1159  | LocalPropertyDifference|| mean local difference in Electronegativity                |
| -0.1023  | IonProperty|| max ionic char                                                  |
| 0.1020   | ElementProperty||MagpieData mean MendeleevNumber                                       |
| -0.1005  | ElementProperty||MagpieData avg_dev CovalentRadius                                     |
| -0.0995  | DensityFeatures|| packing fraction                                               |
| -0.0990  | ElementProperty||MagpieData range Electronegativity                                  |
| -0.0959  | ElectronegativityDiff|| mean EN difference                                    |
| -0.0957  | ElementProperty||MagpieData avg_dev MendeleevNumber                                      |
| -0.0955  | ElementProperty||MagpieData avg_dev SpaceGroupNumber                                |
| -0.0950  | ElementProperty||MagpieData avg_dev Column                                         |
| 0.0950   | ElementProperty||MagpieData minimum Electronegativity                                   |
| 0.0939   | ElementProperty||MagpieData minimum Column                                         |
| 0.0936   | ElementProperty||MagpieData mean NpValence                                       |
| 0.0936   | ValenceOrbital|| avg p valence electrons                                     |
| 0.0935   | ElementProperty||MagpieData minimum MendeleevNumber                                   |
| -0.0922  | ElectronegativityDiff|| maximum EN difference                                   |
| 0.0898   | ElementProperty||MagpieData mean Column                                         |
| -0.0896  | ElementProperty||MagpieData range Column                                         |
C.9. matbench_glass
See figure C16.

Appendix D. PCA

D.1. First component (PC1)

$$PC_1 = \sum w_1 f_{1,i}$$

See table D1.

D.2. Second component (PC2)

$$PC_2 = \sum w_2 f_{2,i}$$

See table D2.

D.3. Third component (PC3)

$$PC_3 = \sum w_3 f_{3,i}$$

See table D3.

ORCID iDs
Pierre-Paul De Breuck https://orcid.org/0000-0002-3173-2058
Matthew L Evans https://orcid.org/0000-0002-1182-9098
Gian-Marco Rignanese https://orcid.org/0000-0002-1422-1205

References

[1] Magee C L 2012 Complexity 18 10–25
[2] Lejaeghere K et al 2016 Science 351 31dd0000
[3] Himanen L, Geurts A, Foster A S and Rinke P 2019 Adv. Sci. 6 1900808
[4] Butler K T, Davies D W, Cartwright H, Isayev O and Walsh A 2018 Nature 559 547–55
[5] Schmidt J, Marques M R G, Botti S and Marques M A L 2019 npj Comput. Mater. 5 83
[6] Chen C, Zuo Y, Ye W, Li X and Ong S P 2021 Nat. Comput. Sci. 1 46–53
[7] Oliynyk A O, Antonovich S, Sparks T D, Ghadbeigi L, Gaultois M W, Meredig B and Mar A 2016 Chem. Mater. 28 7324–31
[8] Stanev V, Osse C, Kusne A G, Rodriguez E, Paglione J, Curtarolo S and Takeuchi I 2018 npj Comput. Mater. 4 29
[9] Agrawal A and Choudhary A 2016 APL Mater. 4 053208
[10] Wolpert D H and Macready W G 1997 IEEE Trans. Evol. Comput. 1 67–82
[11] Wolpert D H and Macready W G 2005 IEEE Trans. Evol. Comput. 9 721–35
[12] Xie T and Grossman J C 2018 Phys. Rev. Lett. 120 145301
[13] Chen C, Ye W, Zuo Y, Zheng C and Ong S P 2019 Chem. Mater. 31 3564–72
[14] De Breuck P P, Hautier G and Rignanese G-M 2021 npj Comput. Mater. 7 83
[15] Dunn A, Wang Q, Ganose A, Dopp D and Jain A 2020 npj Comput. Mater. 6 138
[16] Deng J, Dong W, Socher R, Li L, Li K and Fei-Fei L 2009 ImageNet: a large-scale hierarchical image database 2009 IEEE Conf. on Computer Vision and Pattern Recognition pp 248–55
[17] Ward L et al 2018 Comput. Mater. Sci. 152 60–9
[18] Jovicic A, Bricic K and Bogunovic N 2015 A review of feature selection methods with applications 2015 38th Int. Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO) (IEEE) pp 1200–5
[19] Kauwe S K, Welker T and Sparks T D 2021 Integr. Mater. Manuf. Innov. 9 213–20
[20] Naccarato F, Ricci F, Suntivich J, Hautier G, Wirtz L and Rignanese G-M 2019 Phys. Rev. Mater. 3 044602
[21] Petretto G et al 2018 Sci. Data 5 180065
[22] Lakshminarayanan B, Pritzel A and Blundell C 2017 Simple and scalable predictive uncertainty estimation using deep ensembles Proc. of the 31st Int. Conf. on Neural Information Processing Systems NIPS’17 (Curran Associates Inc.) pp 6405–16
[23] Scalia G, Gantos R, Pernici B, Li Y-P and Green W H 2020 J. Chem. Inf. Model. 60 2697–717
[24] De Breuck P-P, Clevans M L 2021 MODNet v0.1.10 https://github.com/ppdebreuck/modnet
[25] Jain A et al 2013 APL Mater. 1 011002
[26] Mechanical properties of some steels https://citrination.com/modnet-matbench
[27] Choudhary K, Kalish I, Beams R and Tavazza F 2017 Sci. Rep. 7 5179
[28] Zhuo Y, Mansouri Tehrani A and Brogich J 2018 J. Phys. Chem. Lett. 9 1668–73
[29] Petousis I et al 2017 Sci. Data 4 160134
[30] Karabas M, Masutomo T, Tsai A P, Yu J Z and Aihara T J 1997 Nonequilibrium Phase Diagrams of Ternary Amorphous Alloys (Landolt-Börnstein - Group III Condensed Matter 37A) (Berlin: Springer)
[31] Ward L, Agrawal A, Choudhary A and Wolverton C 2016 npj Comput. Mater. 2 16028
[32] de Jong M et al 2015 Sci. Data 2 150009
[33] Castelli E I, Landis D D, Thysegen K S, Dahl S, Chorkendorff I, Jaramillo T F and Jacobsen K W 2012 Energy Environ. Sci. 5 9034
[34] Breiman L 2001 Mach. Learn. 45 5–32
[35] Faber F, Lindmaa A, von Lilienthal O A and Armiento R 2015 Int. J. Quantum Chem. 115 1094–101
[36] Olson R S, Bartley N, Urbanowicz R J and Sparks T D 2016 Evaluation of a tree-based pipeline optimization tool for automating data science Proc. of the Genetic and Evolutionary Computation Conf. 2016 GECCO ’16 (New York: ACM) pp 485–92
[37] Goodall R E and Lee A A 2020 Nat. Comput. Sci. 9 721–35
[38] Wang A Y-T, Kauwe S K, Welker T and Sparks T D 2021 npj Comput. Mater. 7 77
[39] Vishwakarma G, Sonpal A and Hachmann J 2021 Trends Chem. 3 146–56
[40] Davis J and Goadrich M 2006 The relationship between Precision–Recall and ROC curves Proc. of the 23rd Int. Conf. on Machine Learning ICML ’06 (New York: Association for Computing Machinery) pp 233–40
[41] George J and Hautier G 2021 Trends Chem. 3 86
[42] Aggarwal C C, Hinneburg A and Keim D A 2001 On the Surprising Behavior of Distance Metrics in High Dimensional Space Database Theory—ICDT 2001 Lecture Notes in Computer Science Vol 7737 ed J Van den Bussche and V Vianu (Berlin: Springer) pp 420–34
[43] Abdar M et al 2021 Inf. Fusion 76 243
[44] Coulston J W, Blinn C E, Thomas V A and Wynne R H 2016 Photogramm. Eng. Remote Sens. 82 189–97
[45] Kraskov A, Stgbauer H and Grassberger P 2004 Phys. Rev. E 69 066138
[46] Li Z and Hoiem D 2018 IEEE Trans. Pattern Anal. Mach. Intell. 40 2935–47
[47] Evans M L and De Breuck P-P 2021 modnet-matbench https://github.com/ml-evs/modnet-matbench