Report 3

Weaknesses

> 1. rather simple use of ML methods as in older image recognition methods, no ResNet, no transfer learning, etc.

We thank the referee for these suggestions. We included the possibilities of using other, more advanced deep learning models in the future in the outlook of the paper.

> 2. language used to quantify ML recognition accuracy uses standard ML terms (MAE, MSE), but not physically useful parameters (% of deviations, errors, etc)

We modified our definition of MSE to give it a meaningful physical interpretation. More details are given our reply below and in the updated manuscript around Eq.(4.2).

> 3. usefulness of method in "real" situations is left somewhat unclear as ML aspect of data creation is not detailed

We expanded and clarified the description of training data generation and explicitly included the time is needed in practice to train a CNN for a given set of thermodynamic data sets available to the experimentalist. Additional sentences were added to Secs.3A and B.

"One training sample therefore consists of five (nine) different sets of thermodynamic data. A complete training sample for $J = 4$, $\mathcal{G} = \mathrm{m\bar{3}m}$ and $x_0 = 20$~K, $x_1 = 0.5$ is shown in Fig.1(b, c).
To obtain the training data set, we draw the Stevens parameters randomly from a uniform distribution and, for each of these sampled values, compute the aforementioned observables. To generate sufficient training data for the network, the process takes 2-3 hours."

"Let us finally describe the resource cost of training the network. Using 10^5 training examples and 1.5×10^4 validation and 1.5×10^4 testing examples with a batch size of $N_{\mathrm{batch}} = 64$, the network converged after around 100 epochs. With the available GPU (Nvidia Volta V100S, 32 GB), training the network took around 70 seconds per epoch. Fully training the network thus takes around 1-2 hours."

Once the CNN model is trained, it is extremely fast to test new experimental examples.

Report

> The manuscript applies machine learning (ML) approaches, specifically deep learning (DL) methods such as convolutional neural networks (CNN) used predominantly in image recognition, for material sceience/physics. As such, it "open(s) a new pathway in an existing or a new research direction, with clear potential for multipronged follow-up work". The manuscript is also written in a clear and intelligible way, contains a detailed abstract and introduction explaining the context of the problem, and provides sufficient detail.

We thank the referee for this positive evalution of our work, as well as the constructive criticism that we address below and in the modified manuscript.

> The wavelet scaleograms are an ingineous way to harness the power of modern DL image recognition programs for a material science study of material parameters. It was interesting to note that page 8, RH column, that this 2D approach seems to work even better than a simple fully connected 1D neural network structure. A bit more detail might be useful here.

At the beginning of our research, we first tested simpler network architectures, specifically (i) a 1D CNN that was fed with a 1D stacked vector of the raw data and (ii) a simple feedforward deep neural network (DNN). The advantage of these models is that they are less complicated than the 2D CNN described in the paper and it thus takes less time to train. However, we never got fully satisfactory results, even in the simplest case of cubic symmetry, where only two Stevens parameters exist.

We describe details of the performance of these two networks here. We have expanded the paragraph in the manuscript that details the network comparison. It reads

"To make a fair comparison, we created architectures that had approximately the same number of parameters as the 2D CNN and were trained on the same data. Applying to the cubic case with two Stevens parameters, x_0 and x_1, we find the 1D CNN a factor of 2 worse for x_0 and a factor of 7 worse for x_1 than the 2D CNN. The feed-forward deep neural network performed a factor of 1.25 worse for x_0 and a factor of 9 worse for x_1 than the 2D CNN. It is expected that the performance difference is enhanced in the lower symmetry cases with more Stevens parameters to predict, which is why we chose to use the 2D CNN. "

Here in the reply, we give further details on the architecture of the other networks we tested. Since the manuscript is already quite long and we do not include any results of the other networks, we have opted to not include all these details in the manuscript (and we hope the referee agrees with our decision):

Specically, to make a fair comparison, we tried to create architectures that had approximately the same number of parameters as the 2D CNN, $\sim 1.3 \times 10^7$. Each of the models were trained with the same data, although hyperparameters were changed in an attempt to further optimize them. After around 30 epochs and a batch size of 64, the networks converged.

1D CNN: The raw numerical thermodynamic data is stacked to form a two dimensional input with shape (64, 5) (64 temperature/magnetic field values, 5 thermodynamic observables). It is then fed though a number of 1D convolution, max-pooling layers, and eventually a sequence of fully-connected layers. With the fully-connected layers, we apply a dropout of 0.4 to help prevent overfitting. We similarly apply batch normalization and the ReLU activation function to hidden layers, ending with a fully-connected output layer of width 2 with a linear activation function. Applying to the cubic case with two Stevens parameters, this model is able to predict the x_0 coefficient with a mean absolute error (MAE) of $\mathrm{MAE}(x_0) = 0.585$ K. The x_1 coefficient is predicted with $\mathrm{MAE}(x_1) = 0.087$ units. This is about a factor of two (seven) worse than the MAE of x_0 (x_1) we found for the 2D CNN that is fed with the 2D wavelet scaleograms of the thermodynamic data, where we found $\mathrm{MAE}_{\text{2D CNN}}(x_0) = 0.321$ K and $\mathrm{MAE}_{\text{2D CNN}}(x_1) = 0.012$

Feedforward deep neural network: This architecture is the most naive approach to building a model. The input data is reduced to a single one dimensional array containing all 384 data points in sequence. In this network we only apply a sequence of fully-connected layers, each with batch normalization and dropout. We again end with an output layer of width 2 with a linear activation function. This model is able to predict the x_0 coefficient with a mean absolute error (MAE) of $\mathrm{MAE}(x_0) = 0.388$ units. The x_1 coefficient is predicted with $\mathrm{MAE}(x_1) = 0.103$ units. While the network predicts x_0 as good as the 2D CNN, the MAE for x_1 is about a factor of nine worse that of the 2D CNN.

It is expected that the performance difference is enhanced in the lower symmetry cases with more Stevens parameters to predict, which is why we chose to use the 2D CNN.

> Also, is there an intuitive way to understand why a CNN might work better than a simple deep neural net? Why is the convolution/neighbor structure to much better?

Yes, a CNN can pick up specific features that appear in the frequency domain if a Fourier decomposition is applied to the data (here the time domain corresponds to temperature or magnetic field). This ability stems from the fact that the CNN has pooling layers and convolutional layers. They are designed to recognize specific features in images, i.e., specific features in the frequency domain in the 2D wavelt scaleograms. Convolutional layers can use localization of (ω, T) features to classify images (= scaleograms). This information cannot be accurately resolved using simple deep neural nets or 1D neural nets, where the image data is flattened and this spatial information is lost.

> I was surprised not to see a ResNet structure used. Or is the LeNet implementation you use now a ResNet?

The LeNet is not a ResNet. We chose a LeNet to see how well a relatively simple and well known architecture would perform on this problem. It would indeed be interesting to use other CNN architectures and compare their performance on this problem. We appreciate the referee's suggestions and leave this for future work.

> I also did not see that you are using batch normalization? Is that not needed since already included in the CWT construction such that an overall norm is being used? Please comment.

The input data scaleogram is centered, but not normalized. Batch normalization is used in each of the layers, which involves both centering and normalizing the input of that layer.

> The DL approach here is a multidimensional regression. More modern methods could use GANs or even variational auto encoders. Is there a reason why this is not done here?

It would indeed be interesting to use other DL approaches (in particular GANs) for this problem. Here, we chose a well established multidimensional regression approach for simplicity. Since we were able to obtain satisfactory results with this method, we did not need to increase the complexity of the DL method. However, future work, in particular for lower symmetry materials, may very well benefit from more sophisticated DL methods, and we very much appreciate the referee's suggestion.

> Overall, the methods seems to work well since in the end, only very few Stevens parameters are "predicted". It would be good to gain an understanding for how many Stevens parameters one might see that the method still works. What is the theoretically possible such number, given the available crystal fields/materials?

There are 32 different site point symmetry groups in three dimensions. The number of Stevens parameters for f electron systems range from 2 (highest cubic symmetry) to 26 (lowest symmetry = no symmetry or just inversion symmetry). The different number of Stevens parameters are 2 (cubic), 4 (hexagonal D_{6h}), 5 (tetragonal, D_{4h}), 6 (tetragonal C_{4h}), 8 (hexagonal C_3), 9 (D_{2h}), 14 (C_{2h}), 26 (C_i). See for example, Ref.[17] (Walter, 1984) of our paper.

It is an interesting follow-up work to investigate the lower symmetry groups and test how well the deep learning approach works there (possibly using more advanced models as suggested by the referee). In this manuscript, we wanted to focus on the experimentally more relevant case of higher symmetry, as studies of higher symmetry compounds are more common than, e.g., low symmetry triclinic systems.

Requested changes

> 1 clearly indicate in the paper (not just in the caption to Fig. 1) that your wavelet construction leads to image of size 64x64.

We added a comment mentioning this explicitly in the training data generation section.

> 2 The discussion of what a CWT is, appears somewhat ad hoc on page 8 after CWTs have already been discussed. I think it might be useful to move this paragraph surrounding Eqs. (3.3) and (3.4) somewhere else or earlier. Or, alternatively, to introduce subheading in section III.A, or, perhaps a new section on just CWTs as section III.B (or some such label).

We introduced subheadings as suggested by the referee.

> 3 I am overall worried that you only use ML-based accuracy measures (section IV, e.g. caption to Fig. 4 and others). Clearly, MAE and MSE are useful image recognition/classification measures, but for a physics context, I would have expected to see errors and accuracy measures expressed in physical units. For example, an MSE of 10^-3 is somewhat meaningless while a % RMSE, i.e. (4.1) or (4.2) divided by, say O(x_true), would allow the contruction of a % error/deviation, averaged over the used Stevens coefficients. Indeed, one can also cmpute such measure for each Stevens parameter.

Regarding the MSE, we agree with the referee and have updated the manuscript to now define the MSE for normalized and dimensionless operator differences (see updated Eq.(4.2)). The MSE is now physically meaningful. We have also updated the figures accordingly and added the following sentence below Eq.(4.2): "To account for the differences in size and units between observables, we first normalize each dataset by their mean and perform Eq.(eq:4.2) on the resulting dimensionless quantities."

We think that the MAE is an appropriate measure for the quality of the Stevens's coefficients predictions. The coefficients x_i with $i \neq 0$ are all dimensionless and range from $[-1, 1]$ and a MAE thus describes how well they are predicted. The coefficient x_0 has units of Kelvin (it is the overall scale of the Stevens parameters) and it ranges from $[0.5, 50]$ K in practice (this covers the full range of typical x_0 values found experimentally). The MAE of x_0 thus is a measure of how well we can predict the overall scale.

> 4 page 9, LH column, "clearly recognized". Sorry, but I cannot see this, never having looked at a CWT. Please either indicate in the figure or reword my clearly.

We have expanded our discussion in the manuscript how some broad features of the original data can be recognized in the scaleograms. We have removed the word "clearly".

> minor points:
> 5 page 12, RH column, "In this section, we demonstrate this ..." I am not sure what is meant by the 2nd "this", i.e. circumstances, parameters, custem CNN, each case? Please rewrite.

We have reformulated the paragraph to make this point more clear. In the sentence, "this" refers to "The ultimate application of the presented CNN algorithm is to extract CF parameters from real experimental data".

> 6 conclusions: what is meant be "net work performs ... well ... algorithm ... works well"? Could you please be quantitative and specific? How much faster and how much more accurate?

We provide more details in the modified manuscript on how long the generation of training data and how long training the network takes. We also refer more clearly to the MAE and MSE as a quantitative measure of how well the network performs. We feel that we do not want to repeat all these details in the conclusion of the paper, as it can be found in the corresponding sections, but we are more precise what we take as quantitative performance measures.