



Universiteit  
Leiden

The Netherlands

## **Towards the prediction of molecular parameters from astronomical emission lines using Neural Networks**

Barrientos, A.; Holdship, J.R.; Solar, M.; Martín, S.; Rivilla, V.M.; Viti, S.; ... ; Humire, P.

### **Citation**

Barrientos, A., Holdship, J. R., Solar, M., Martín, S., Rivilla, V. M., Viti, S., ... Humire, P. (2021). Towards the prediction of molecular parameters from astronomical emission lines using Neural Networks. *Experimental Astronomy*, 52(1-2), 157-182.  
doi:10.1007/s10686-021-09786-w

Version: Publisher's Version

License: [Licensed under Article 25fa Copyright Act/Law \(Amendment Taverne\)](#)

Downloaded from: <https://hdl.handle.net/1887/3264197>

**Note:** To cite this publication please use the final published version (if applicable).



# Towards the prediction of molecular parameters from astronomical emission lines using Neural Networks

Alejandro Barrientos<sup>1,2</sup> · Jonathan Holdship<sup>3,4</sup> · Mauricio Solar<sup>1</sup> · Sergio Martín<sup>5,1</sup> · Víctor M. Rivilla<sup>6,7</sup> · Serena Viti<sup>3,4</sup> · Jeff Mangum<sup>8</sup> · Nanase Harada<sup>9,10,11</sup> · Kazushi Sakamoto<sup>10</sup> · Sébastien Muller<sup>12</sup> · Kunihiko Tanaka<sup>13</sup> · Yuki Yoshimura<sup>14</sup> · Kouichiro Nakanishi<sup>9,11</sup> · Rubén Herrero-Illana<sup>5,15</sup> · Stefanie Mühle<sup>16</sup> · Rebeca Aladro<sup>17</sup> · Susanne Aalto<sup>12</sup> · Christian Henkel<sup>17,18</sup> · Pedro Humire<sup>17</sup>

Received: 19 May 2021 / Accepted: 20 August 2021 / Published online: 18 September 2021  
© The Author(s), under exclusive licence to Springer Nature B.V. 2021

## Abstract

Molecular astronomy is a field that is blooming in the era of large observatories such as the Atacama Large Millimeter/Submillimeter Array (ALMA). With modern, sensitive, and high spectral resolution radio telescopes like ALMA and the Square Kilometer Array, the size of the data cubes is rapidly escalating, generating a need for powerful automatic analysis tools. This work introduces *MolPred*, a pilot study to perform predictions of molecular parameters such as excitation temperature ( $T_{\text{ex}}$ ) and column density ( $\log(N)$ ) from input spectra by the use of neural networks. We used as test cases the spectra of CO,  $\text{HCO}^+$ , SiO and  $\text{CH}_3\text{CN}$  between 80 and 400 GHz. Training spectra were generated with MADCUBA, a state-of-the-art spectral analysis tool. Our algorithm was designed to allow the generation of predictions for multiple molecules in parallel. Using neural networks, we can predict the column density and excitation temperature of these molecules with a mean absolute error of 8.5% for CO, 4.1% for  $\text{HCO}^+$ , 1.5% for SiO and 1.6% for  $\text{CH}_3\text{CN}$ . The prediction accuracy depends on the noise level, line saturation, and number of transitions. We performed predictions upon real ALMA data. The values predicted by our neural network for this real data differ by 13% from the MADCUBA values on average. Current limitations of our tool include not considering linewidth, source size, multiple velocity components, and line blending.

**Keywords** Molecular astronomy · Molecular parameters · Machine learning · Neural networks · MADCUBA · ALCHEMI

✉ Alejandro Barrientos  
alejandro.barrientos@alma.cl

## 1 Introduction

The study of the molecular composition of objects in space has been a matter of extensive research since the 1930s [16]. Through the use of spectroscopy techniques over 200 molecules have been identified in interstellar or circumstellar medium [20], and the number keeps growing every year. The analysis process to detect these molecules is quite complex and requires effort from observers, astrochemists and laboratory spectroscopists [2]. The construction of astronomical observing facilities like the Atacama Large Millimeter/Submillimeter Array (ALMA) has created the possibility to observe the universe with an unprecedented combination of sensitivity and angular resolution. Such facilities allow astronomers to study the physical and chemical properties of the molecular gas in a variety of sources in the Universe (e.g. Nakajima et al., [12]).

The introduction of ever larger and more sensitive instruments in astronomy and the latest developments in computing performance has generated a need for tools to support the analysis of data sets [1]. This is illustrated by the number of new astronomical facilities that are implementing automated data reduction pipelines such as ALMA [7], Vera C. Rubin Observatory (formerly LSST; Jurić et al., [5]) and E-ELT [8]. Telescopes like the Square Kilometer Array (SKA), that will be deployed in the very near future, will even rely only on fully automatic reduction pipelines, due to the large amounts of data they will produce [4].

Current facilities produce crowded spectra even in sources where previous facilities were only able to detect a limited number of bright transitions of the most abundant species. Line identification and extraction of physical parameters is generally still a manual and time consuming process even making use of state of the art tools briefly described below.

Several tools have been developed over time, to assist in line analysis. These tools have varying degrees of automation. For instance CASA [10] allows the user to connect to Splatalogue<sup>1</sup> catalog via casaviewer. This enables the user to identify lines by overlaying transitions from the catalog on their spectra. In this case, there is no automation. Another tool is the ALMA Data-Mining Toolkit (ADMIT, Teuben, [17]) which enhances CASA with a spectral Line identification algorithm. However, it does not provide estimates of the physical parameters of the molecules identified. Other programs like RADEX [18] and MADEX [2] can generate models, that can be manually compared with data, although this comparison is not automated and requires the tuning of many input parameters. The XCLASS interface [11] contains a program named myXCLASS which contains routines to fit models to observed spectra. The fitting model can be automated with multiple approaches as mentioned in [15]. Another tool is CASSIS [19] which computes synthetic LTE models that can be compared with observations. It contains a tool to perform physical parameter estimations. The last tool we will cover is MADCUBA [9] which is a tool for line analysis and spectroscopic work. It contains a feature called Spectral Line Identification and Modelling (SLIM) which allows the automatic fitting of molecular parameters.

---

<sup>1</sup><https://splatalogue.online>

It allows the possibility to generate synthetic spectra programmatically, given the appropriate parameters, and also to merge several data cubes across a frequency axis. MADCUBA was selected as our primary source for training examples.

Having established that automated tools are necessary, our intention is to contribute by taking a step towards the prediction of molecular parameters using neural networks. For the sake of probing feasibility and as a pilot study, in this paper we constrain our analysis to the prediction of the parameters of excitation temperature ( $T_{\text{ex}}$ ) and column density ( $\log(N)$ ) from molecular spectra. Section 2 presents our pilot study for prediction of molecular parameters, describing the overall design, training data and methods. Section 3 describes and discusses our test results with synthetic data, with a number of parameter combinations for the neural networks. We also present our results with astronomical data, performing predictions for  $\log(N)$  and  $T_{\text{ex}}$  for a spectrum coming from an ALMA Large Program project. Section 4 analyzes the scalability and linear speedup of our solution. Section 5 summarizes our conclusions and comments on future work.

## 2 The *MolPred* prototype model

Given that astronomical spectra can be modelled from their underlying physical parameters, it should be possible to predict those parameters using a regression model. In this work, we use several neural networks which take information from the spectrum as an input. A neural network is an algorithm which learns relationships between some inputs and outputs resulting in a blackbox model [14].

Whilst even simple radiative transfer models require several parameters, we consider only  $\log(N)$  and  $T_{\text{ex}}$  in this pilot study. The column density is an important quantity as it is a measure of how much of a species is present towards an object. The intensity of a line approximately scales with the column density when lines are optically thin. The excitation temperature reflects the relative population of energy levels of a molecule (e.g., following a Boltzmann distribution at local thermal equilibrium) and helps to characterize the conditions of the gas. It primarily determines the relative line intensity among the transitions of a given species.

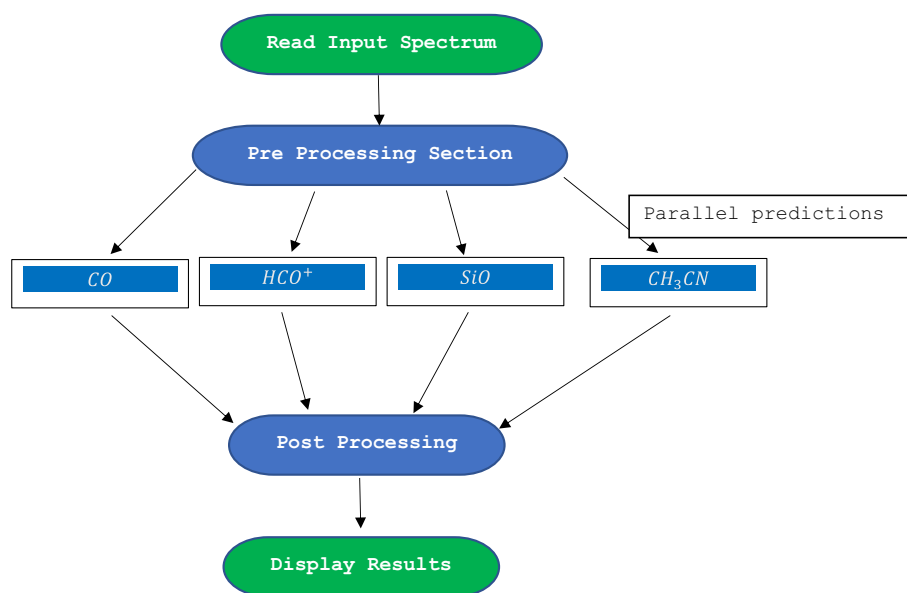
For this purpose, we have developed a tool called *MolPred*, which is capable of predicting said parameters for an initial sample of four molecules, provided an input spectrum. This set can be incrementally extended to more parameters and to the full set of species available in spectroscopic catalogs. The current python code of the tool is available in our GitHub repository <sup>2</sup>

### 2.1 Overall design

An overview of the *MolPred* program is described in Fig. 1. The flow starts by receiving an input spectrum for which *MolPred* will generate predictions of  $\log(N)$  and  $T_{\text{ex}}$ . The input spectrum is pre-processed as explained below, so the predictions for

---

<sup>2</sup><https://github.com/MadScience01/molpred>



**Fig. 1** A diagram showing the *MolPred* workflow. Each box indicates a process and the rectangles showing molecules indicate the neural networks

all the molecules can be done in parallel via individual neural networks. After all predictions are done, the results are collected and handled in a post-processing stage, where the resulting predictions are presented to the user.

In the preprocessing stage, we extract the intensity of every transition of a given molecule that falls into our working frequency range. Since the networks are trained with a fixed number of transitions, this number of inputs must be supplied for every prediction. However, real astronomical data is often incomplete and transitions may be missing. In those cases, we have opted to zero-pad the input array so that measured transitions are included and zeros replace missing transitions.

For the transitions that are present in the observation window, we include the corresponding Splatalogue frequency and the intensity value for the closest frequency observed in the input array. We consider a transition present if the closest frequency in the observation, with respect to the Splatalogue frequency, is within 10 MHz. Finally, the intensities that will be passed to the predictor are scaled between 0 and 1, which is a typical pre-processing step for neural network training, as all the training examples and labels were scaled for better training results. This scaling was also applied to the values of  $\log(N)$  and  $T_{\text{ex}}$  in our test and training data. The molecules and the number of transitions they have within the frequency range considered in this work are given in Table 1.

In the prediction stage, the neural networks take the input array and produce a scaled prediction for  $\log(N)$  and  $T_{\text{ex}}$ . For each molecule, the neural network used in this stage is the best network from the grid discussed in Section 2.3. Finally, in the post-processing stage, the produced predictions are descaled back to their actual magnitudes.

**Table 1** Molecules considered in this work and the number of rotational transitions from the vibrational ground state which are in the range of frequencies covered by *MolPred*

Species	Name	Number of Transitions (80-400GHz)
CO	Carbon monoxide	3
HCO <sup>+</sup>	Formylium	4
SiO	Silicon monoxide	8
CH <sub>3</sub> CN	Methyl cyanide	437

## 2.2 Training data set

The data used to train the neural networks which make up *MolPred* are a set of synthetic spectra generated by the SLIM LTE spectral model, which is part of the MADCUBA software package [9], which makes use of the spectroscopic parameters from JPL catalog [13]. We generate LTE spectra for each species, with 1 MHz resolution between 80 and 400 GHz using column densities in the range  $\log(N) = 12 \text{ cm}^{-2}$  to  $\log(N) = 19.9 \text{ cm}^{-2}$ , with steps of  $\log(N) = 0.1 \text{ cm}^{-2}$  and temperatures from 10 K, performing multiplicative increments of 30% all the way to 233 K, in this way we increase the coverage at lower temperatures.

In order to produce synthetic spectra, MADCUBA requires other parameters on top of  $\log(N)$  and  $T_{\text{ex}}$  considered in our pilot study. Thus these parameters were fixed as follows: output intensity units were set to Kelvin; line width and velocity were fixed to 150 and 250  $\text{km s}^{-1}$  respectively; an emitting source size of  $10''$  was assumed. These fixed parameters were selected to match those required to fit the actual astronomical spectra from the ALMA Comprehensive High-Resolution Extragalactic Molecular Inventory (ALCHEMI, Martin et. al, in preparation), as discussed below. Since our training datasets are created in rest frequency units, it is therefore agnostic to the velocity parameter. However, our predictor cannot be directly applied to astronomical data with different parameters of line width and source size. An extension of our neural network should be required to make our tool fully useable. All training and validation spectra used in this work were then created by combining these individual molecular spectra and adding Gaussian noise with an rms between 10 and 50 mK, which we consider to be a reasonable range of values for noise in an astronomical spectrum at these frequencies.

## 2.3 Neural network training

We have used the *keras* package from the *tensorflow* library [3] for the creation and training of the neural networks. *Keras* model files are saved after training, for later use in the prediction stages. Neural networks have a large number of hyperparameters. The hyperparameters that were varied are given in Table 2 along with the range of values trialled. For each possible combination of parameters, a model was created. Neural networks were trained for up to 1000 epochs, where an epoch is one pass of

**Table 2** A list of neural network hyperparameters which were varied and the values that were tested

Parameter type	Parameter values
Activation functions	Sigmoid, ReLU, Linear, Tanh, Swish.
Layers	Single, double, triple
Neurons	256, 1024
Molecules	CO, HCO <sup>+</sup> , SiO, CH <sub>3</sub> CN
Training examples	500, 1000, 2000, 4000, 8000, 16000, 32000
Noise levels	0.01 to 0.05 K
Optimizer	Adam
Loss function	MAE
Training patience	10 epochs
Maximum amount of training	1000 epochs

the full training dataset to the network. However, to keep training time low, we implemented an early stopping mechanism where the training would stop if the validation loss did not improve over 10 epochs.

We trained each neural network individually using spectra that contained only noise and emission from transitions of the molecule for which the network would predict. These neural networks could later be loaded together as part of the *MolPred* code to predict from full spectra. In this initial experiment we decided to use a simple sequential neural network with a maximum of 3 densely connected layers of up to 1024 neurons each. We wish to establish an initial set, as more molecules and output classes can be added later. The networks were trained to minimize the mean absolute error (MAE) between the scaled  $\log(N)$  and  $T_{\text{ex}}$  of an input spectrum and the predictions using the Adam optimizer [6]. We use the scaled predictions so that errors in  $\log(N)$  and  $T_{\text{ex}}$  are equally weighted as the unscaled variables differ by many orders of magnitude. To test these trained neural networks, the MAE on the scaled predictions across the entire validation set was calculated for each network. These MAE values were then compared between networks to select the best regressor to include in *MolPred*.

## 2.4 Training performance

With the proposed methods, the next step was to observe the training time differences based on the selection of CPU/GPU(s). We began by comparing training times for a single molecule (CH<sub>3</sub>CN) and model (*relu\_triple\_1024*) with 32000 training examples and 10 mK of noise with a batch size of 32, and configuring *Tensorflow* for using either CPU, one GPU or two non-linked GPUs. We tried different data distribution strategies<sup>3</sup>, to observe the impact on the MAE, the results, ordered by minimum MAE, can be seen in Table 3. For reference, these training times were obtained using a PC with an Intel I7-8700K CPU, 32 GB of RAM, two non linked Zotac GeForce

<sup>3</sup>[https://www.tensorflow.org/api\\_docs/python/tf/distribute/Strategy](https://www.tensorflow.org/api_docs/python/tf/distribute/Strategy)

**Table 3** Data distribution comparison, sorted by MAE

Strategy	Training Time	Epochs	Min MAE
Central storage 2 GPU	29 min 20 sec	100	0.01803
Collective all-reduce 1 GPU	10 min 8 sec	84	0.02096
Mirrored 1 GPU	9 min 43 sec	75	0.02203
Mirrored 2 GPU	21 min 57 sec	66	0.02238
Collective all-reduce 2GPU	20 min 49 sec	53	0.02384
Mirrored CPU	11 min 55 sec	54	0.02386
Collective all-reduce CPU	9 min 41 sec	44	0.02483
Central storage CPU	9 min 28 sec	43	0.0250
Central storage 1 GPU	5 min 3 sec	46	0.02568

GTX 1060 6GB video cards and a Kingston A2000 1 TB Solid State Drive - M.2 2280.

We can observe that in terms of MAE, the error ranges between 1.8% and 2.5% and the training time ranges between 5 and 30 minutes, depending on the data distribution strategy selected. In order to compare the speed of the strategies, we should compare them based on the time they took to reach 43 epochs, which was the minimum number of epochs required to finish the training. The comparison can be observed in Table 4.

It can be observed that the tests where just one GPU is used, performed faster. It is interesting to notice that the instances where the CPU was used for training, were faster than the instances where 2 GPUs were used, this seems reasonable, as both GPUs in the test setup are not linked, and in order to perform the training work, the data needs to be copied to both cards memory, consuming additional time. With this, we believe that our training should be done using just one GPU.

We are able to influence in the training speed, by adjusting the batch size. Table 5 shows the relation between batch size and minimum MAE. We selected the *central*

**Table 4** Strategy performance comparison, sorted by training time

Strategy	Training Time	Epochs	Min MAE
Central storage 1 GPU	4 min 45 sec	43	0.02579
Collective all-reduce 1 GPU	5 min 8 sec	43	0.02553
Mirrored 1 GPU	5 min 35 sec	43	0.02536
Collective all-reduce CPU	9 min 28 sec	43	0.02462
Central storage CPU	9 min 28 sec	43	0.02500
Mirrored CPU	9 min 28 sec	43	0.02536
Central storage 2 GPU	12 min 46 sec	43	0.02601
Mirrored 2 GPU	14 min 15 sec	43	0.02434
Collective all-reduce 2GPU	16 min 55 sec	43	0.02564



**Table 5** Batch size performance comparison, sorted by batch size

Batch Size	Training Time	Epochs	Min MAE
32	5 min 3 sec	46	0.02568
64	2 min 9 sec	38	0.02623
128	2 min 32 sec	75	0.2269
256	44 sec	30	0.03387
512	41 sec	37	0.03658
1024	1 min 11 sec	73	0.03532
2048	1 min 23 sec	99	0.03750

*storage* distribution, with one GPU, and vary the batch size, from 32 examples per batch, to 2048 examples per batch, the noise remains at 10 mK.

To observe the specific relation between batch size, and MAE, we will stop at 30 epochs, the minimum required to complete the training and review the difference in Table 6.

We can observe the trade off between training time versus minimum MAE as given by the batch time. The larger the batch size, the faster the training finishes, at the expense of larger errors. In this test, by increasing the batch size to 2048 examples per batch, the model error increases about 3%, but this represents over 50% of the original error figure, however the training time has been reduced in almost 87%. Having lower training times, opens the possibility of using a larger number of training examples. In order to obtain the best possible predictions, we decided to aim for a lower MAE, rather than a fast training time, thus we will use a batch size of 32 examples per batch for the rest of the following experiments.

In Section 4 we will calculate *MolPred*'s prediction speed acceleration or *speedup*, by using (1)

$$S = \frac{\text{Serial execution time}}{\text{Parallel execution time}} \quad (1)$$

**Table 6** Batch size vs training time vs minimum MAE comparison, sorted by batch size

Batch Size	Training Time	Epochs	Min MAE
32	3 min 18 sec	30	0.02836
64	1 min 42 sec	30	0.02746
128	1 min 2 sec	30	0.02971
256	44 sec	30	0.03387
512	34 sec	30	0.03995
1024	30 sec	30	0.04734
2048	26 sec	30	0.05072

### 3 Results and discussion

#### 3.1 Results on test data

Our first task was to determine an appropriate number of training examples for the neural networks. We present the evolution of the MAE as a function of the number of training examples in Fig. 2. We found that the MAE initially improves by doubling the number of training examples but beyond 16000 examples, increasing the number of training examples gives a marginal improvement. As we increase the number of examples, we increase the amount of information we provide the model until the point where further examples just duplicate information we've already provided. At that point, further examples don't improve things. We also included the training times, for evaluation, which we found to be roughly proportional to the number of examples. To balance low MAE values with reasonable training times so we don't affect scalability, we reached a compromise of 32000 examples, which we defined as our baseline for all models shown in this section.

Our networks achieve different accuracies because the relationship between excitation temperature and line intensities varies strongly depending on the molecule's transition properties and collisional rates. Thus, the function the network must learn to represent each molecule is different and the network has varying success in replicating it. We also observe that the more transitions a molecule has within our frequency range, the lower the MAE our network achieves. This is illustrated in Fig. 2 and may indicate that, in general, the more transitions a molecule has, the easier it is for networks of this type to learn a useful prediction strategy.

Once we have established the number of training examples, we wished to observe the behaviour of the MAE in the training stage of each network. Figure 3 presents the evolution of the MAE for the neural network that performed best on each molecule.

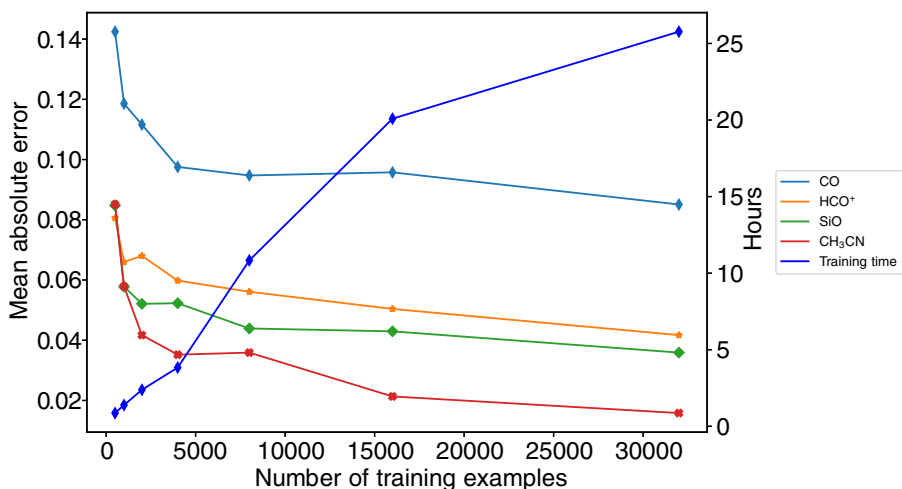
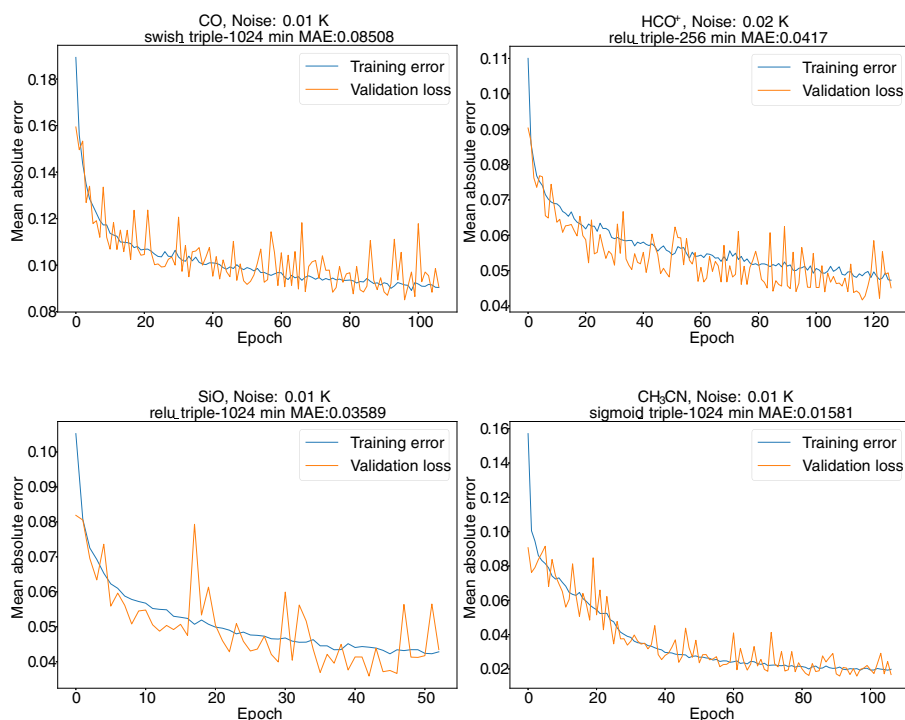


Fig. 2 Mean absolute error and training time in hours as a function of number of training examples

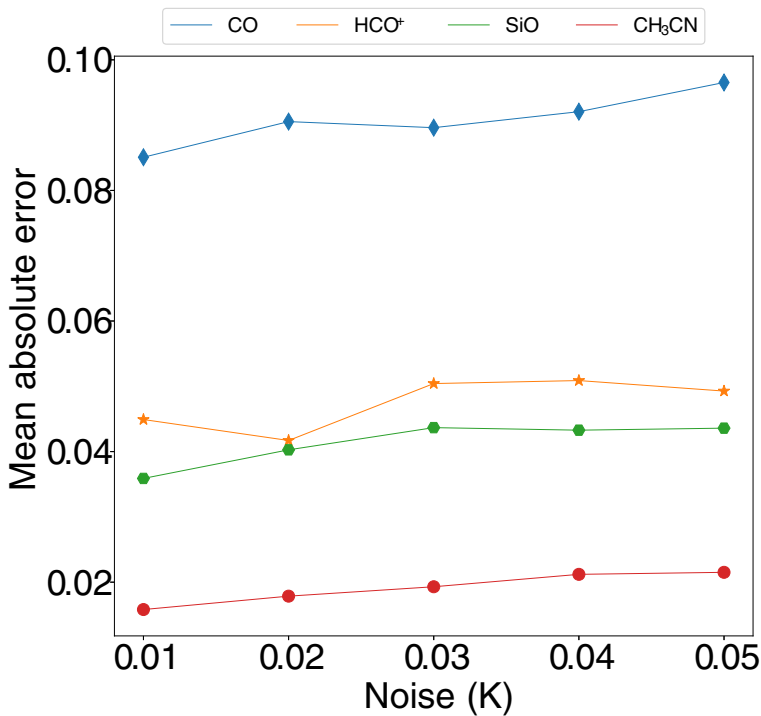


**Fig. 3** Training mean absolute error over Molecule set, each figure contains the molecule name, the best neural network model, the minimum MAE and its respective noise level.  $\text{HCO}^+$  was the only molecule whose best model was with a higher noise level

We allowed the networks to be trained for a maximum of 1000 epochs but stopped training early if the model did not improve for 10 epochs (training patience). In most models, the training stops close to 100 epochs, the lowest validation loss can be seen in the plots approximately ten epochs before the end of the plot.

To obtain a deeper understanding of the behavior of the neural networks, we analyzed how the noise would influence the performance. In Fig. 4 we can observe that the general trend is that the MAE increases with the noise. We assume the best network is the one with the lowest MAE at any noise level for that particular molecule, and use that network in *MolPred* as shown in Table 7. However, since Fig. 4 shows the MAE is only weakly affected by the noise value, one could choose to train their networks with a large noise value to ensure the model is robust to the noise in real spectra.

To verify the quality of the predictions on each neural network, we predicted the column density and excitation temperature for a test set of 6400 testing examples not used during training. We then checked the error distribution of the predictions, which is shown in Fig. 5. The error distribution is centrally peaked which is an important result that our choice of loss function does not guarantee. The strongly peaked distributions mean that the networks typically give small errors and are unlikely to predict an extremely incorrect value.

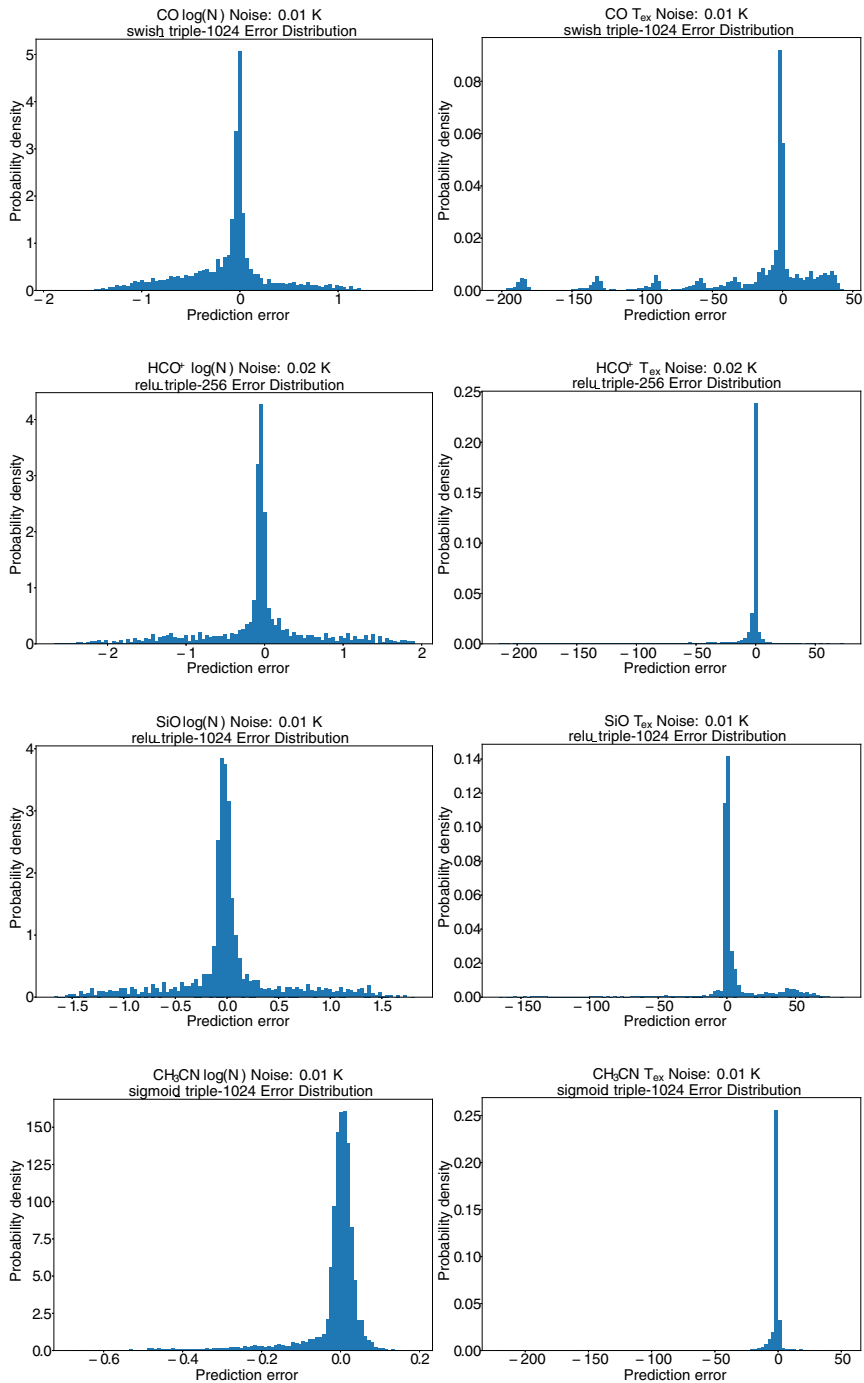


**Fig. 4** Noise vs mean absolute error

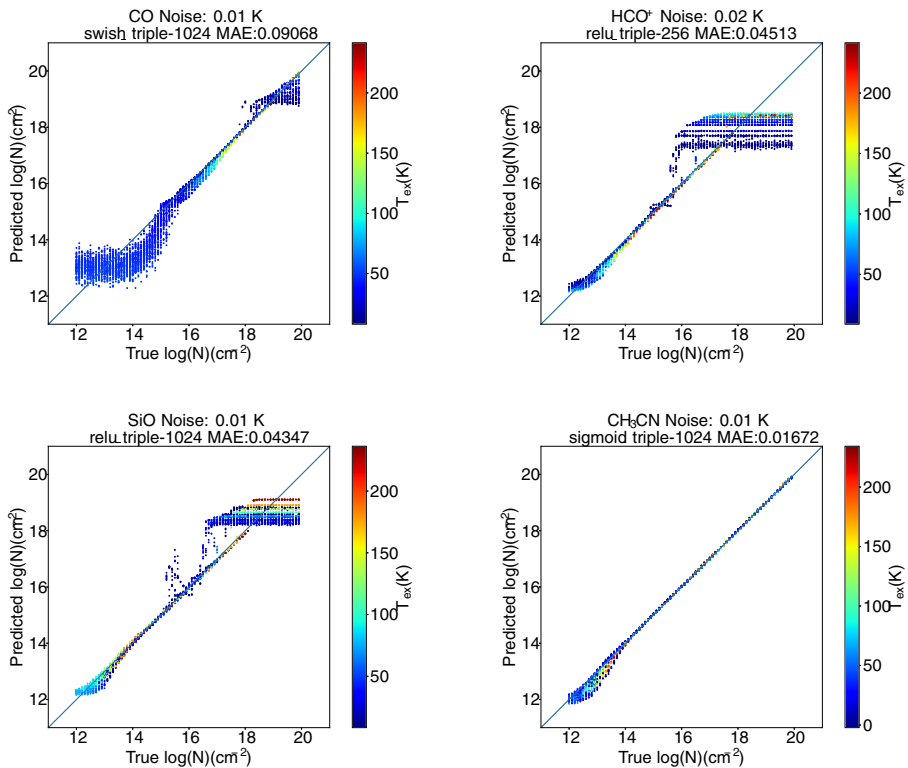
Having obtained the best models for each molecule, we wanted to review the prediction behaviour of the networks, which was plotted in Fig. 6. We see that for all the molecular species in our pilot study, the predictions follow a tight linear correlation with predicted values close to the true values, as also seen in the centrally peaked histograms in Fig. 5. This can be used to obtain a general idea of the relative errors. However, the prediction accuracy drops for both low and high values of  $\log(N)$ . The  $\log(N)$  value at which this deviation occurs is dependent on the molecule and has a physical explanation. For low values of  $\log(N)$  the line intensities become close to or below the noise level, which causes the neural network to predict values that are not dependent on the true value. Moreover, this also has a low dependency on the

**Table 7** The MAE of the best neural network trained for each molecule and the name of the model which indicates the activation function, the number of layers (3), and number of neural per layer

Molecule	Noise (K)	MinMAE	Epochs	Model
CO	0.01	0.08508	96	swish_triple-1024
HCO <sup>+</sup>	0.02	0.04170	116	relu_triple-256
SiO	0.01	0.03589	42	relu_triple-1024
CH <sub>3</sub> CN	0.01	0.01581	96	sigmoid_triple-1024



**Fig. 5** Histograms showing the distribution of errors on the  $\log(N)$  (left column) and  $T_{ex}$  (right column) prediction of each molecule (100 bins). These all peak at zero indicating small errors are most common

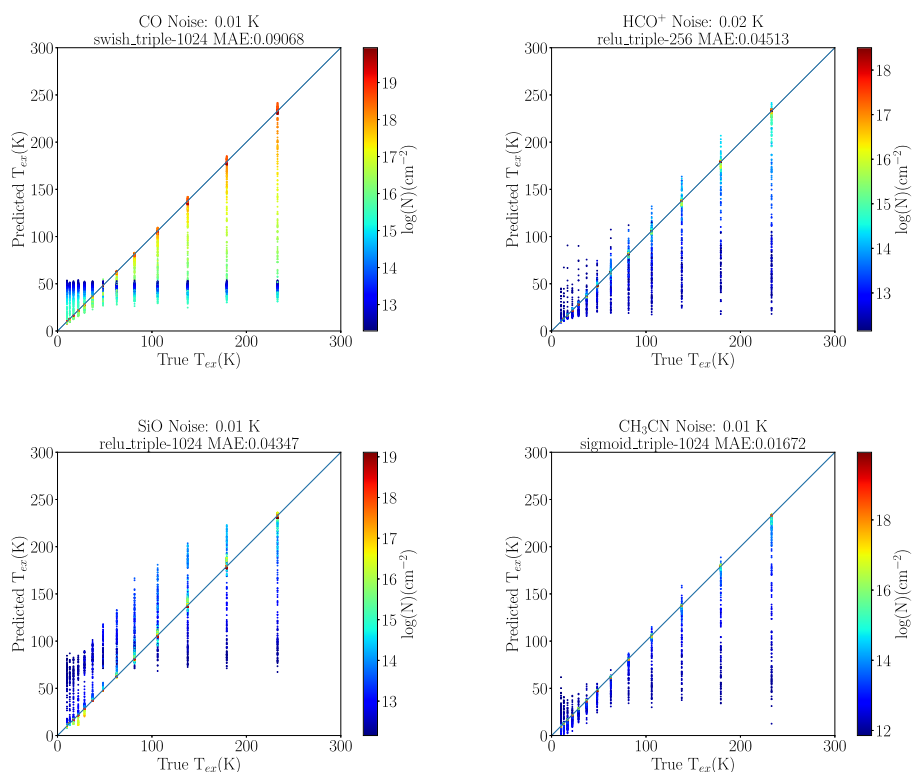


**Fig. 6** Scatter plots showing the predicted  $\log(N)$  against its true value for all spectra in the test data. The colour scale indicates the excitation temperature of the spectrum. Each neural network was trained on spectra with a noise of 10mK for all molecules except for  $\text{HCO}^+$  where the best network was trained using a noise of 20 mK

temperature as seen in the color coding in Fig. 6, as it is purely due to the lack of signal in the input spectra. Essentially, the network cannot distinguish between input spectra below a  $\log(N)$  threshold.

On the other hand, for high values of  $\log(N)$ , the spectral features from the input data are saturated. That is, all transitions reach a saturation flux density once a value of  $\log(N)$  is surpassed (see Martín et al., [9], for a description of line saturation). The  $\log(N)$  value at which saturation occurs has a strong dependency on the value of the temperature as seen in the color coding in Fig. 6. The behaviour also depends on the number of transitions available for each molecule (Table 1). Thus, in the case of  $\text{CH}_3\text{CN}$ , even for high values of  $\log(N)$ , there is still a significant number of low flux density unsaturated transitions and therefore we do not observe the effect of saturation in Fig. 6.

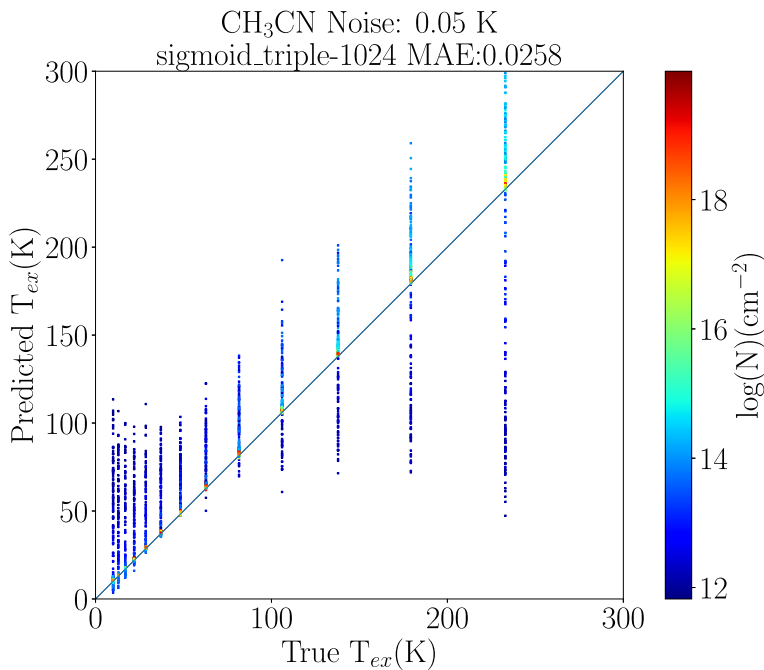
The same physical explanation applies to the predictions of temperature parameter. Figure 7 presents the test predictions of  $T_{\text{ex}}$  against their true values. For all molecules, we can observe that for all values of  $T_{\text{ex}}$ , predictions appear dependent on the value of  $N$ . As can be seen in the figure, the predicted temperatures are closer to



**Fig. 7** Predicted  $T_{\text{ex}}$  from the best neural networks against the true value for all spectra in the test set. The column density of the test spectra is given in the colour scale. The training noise was 10 mK for all molecules except HCO<sup>+</sup> which was 20mK

their real values for higher  $N$ . This has a similar explanation to the neural networks' poor performance on the column density prediction for low column densities. At low  $N$ , the intensities of the transition decrease and come closer to the noise. The noise then dominates the prediction, making the recognition of the transition impossible if the transition is below the noise level. We can compare the CO results with the results of other molecules, particularly with CH<sub>3</sub>CN which has the largest number of transitions of the molecules in our set. These results suggest that the accuracy in the prediction of the temperature could be dependent on the number of transitions available to construct a better characterized model of the molecule. However this would require additional testing, which escapes the original scope of this work.

If we increase the noise to 0.05 (50mK) we can observe that the predictions of temperature spread over a wider range, rather than forming small groups, as in the case with lower noise. Still, the predictions remain accurate for the higher column density values (yellow/red values in Fig. 8), where the spectral features are clearly identified above the noise level.



**Fig. 8** Similar to Fig. 7 but only showing predictions for CH<sub>3</sub>CN with an rms noise of 50 mK so it can be compared with lower right panel of Fig. 7

### 3.2 Results on astronomical data

To further test *MolPred* we used data from ALCHEMI, which was one of the ALMA Large Programmes in Cycle 5. The astronomical target was the central molecular zone of the starburst galaxy NGC253. The project consists of a full spectral line survey continuously covering ALMA Bands 3 to 7. We used the low resolution spectrum from the ALMA Compact Array (Morita Array) which covers the frequency range of 125 GHz to 373.2 GHz. It therefore contains an almost complete set of transitions from the sample molecules used in this study. The continuous coverage and uniform sensitivity makes this dataset an ideal test sample for our study. This wide band spectrum was shift to rest frequency assuming a Doppler shift of 250 km s<sup>-1</sup> and used as an input to *MolPred*. *MolPred* then generated  $\log(N)$  and  $T_{ex}$  predictions for each molecule following the flow of Fig. 1.

The *MolPred* predictions are displayed in Table 8. For comparison, we also include the fitted values using the MADCUBA AUTOFIT packaged used for the actual spectroscopic analysis of the ALCHEMI data. The *MolPred* predictions for  $\log(N)$  are within 1.5% of the ones from MADCUBA, and for  $T_{ex}$  are within 25% from the ones by MADCUBA. Thus the predictions agree very well except in the case of the excitation temperature of CH<sub>3</sub>CN. However, we note that MADCUBA fitting algorithm did not converge unless taking into account the contribution from other molecular species blended to the CH<sub>3</sub>CN. Despite the fact *MolPred* did not include

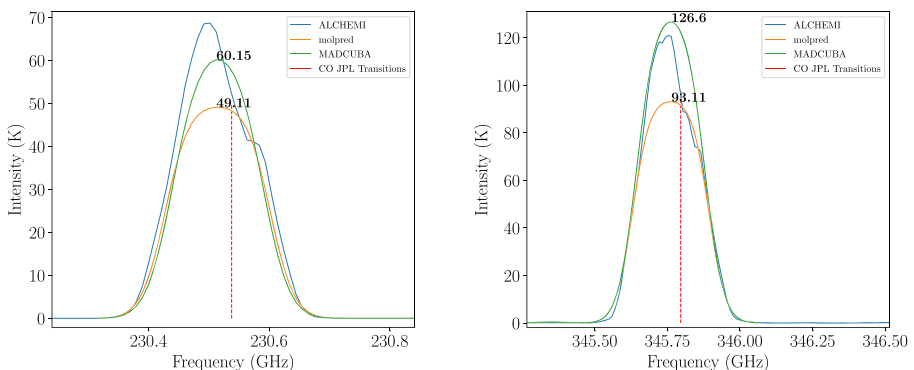


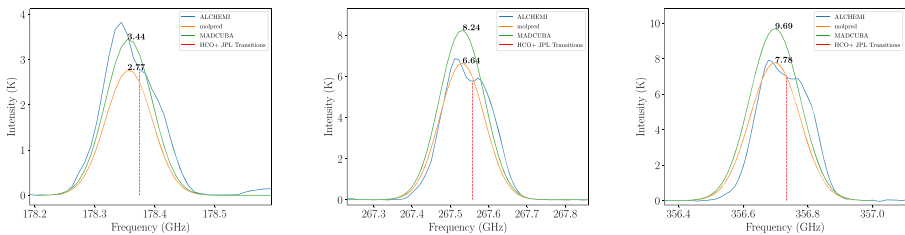
**Table 8** The  $\log(N)$  and  $T_{\text{ex}}$  predicted by Molpred for the ALCHEMI data alongside the values obtained from the MADCUBA fit

Molecule	<i>MolPred</i> $\log(N)$	<i>MolPred</i> $T_{\text{ex}}$	MADCUBA $\log(N)$	MADCUBA $T_{\text{ex}}$
CO	18.51	17.45	18.44	19.48
HCO <sup>+</sup>	13.98	16.55	14.08	15.30
SiO	13.17	17.65	13.18	13.98
CH <sub>3</sub> CN	13.39	58.95	13.27	38.95

information on other molecular “contaminants”, the predictions were still reasonably close to the value predicted by MADCUBA. We now examine these results by plotting our predictions together with the closest examples to the MADCUBA predictions from our training files.

We start the analysis with CO, the simplest molecule of our set. The *MolPred* prediction gives a very similar value of  $\log(N)$  to the MADCUBA fit but has a slightly lower excitation temperature. The result of this difference can be seen in Fig. 9 where we plot the intensities of MADCUBA spectra generated using the *MolPred* and MADCUBA predictions alongside the ALCHEMI data. Whilst both models underfit the data, the *MolPred* spectrum has a somewhat lower intensity than the others due to the low excitation temperature. Based on our analysis of the test data, we can expect the saturation effect seen in Fig. 7 has affected the accuracy of predictions at these high column densities. Further, the value of the intensity passed to the *MolPred* predictor was taken at the rest frequency of the transition and does not match the peak value due to imprecise Doppler shifting. The peak intensity of the *MolPred* prediction is close to the value of the ALCHEMI data at the transition frequency for both CO transitions.

**Fig. 9** CO line profiles from the ALCHEMI Data plotted in blue. Overplotted are the spectra generated from the *MolPred* predictions (orange) and MADCUBA predictions (green). For reference, we included a red vertical line to indicate the frequency of the closest transition according to the JPL Catalog. The ALCHEMI data set used in this test, only has data for 2 out of the 3 transitions in our training data, we can see the detail of the CO(2-1) in the left figure and CO(3-2) transitions in the right figure



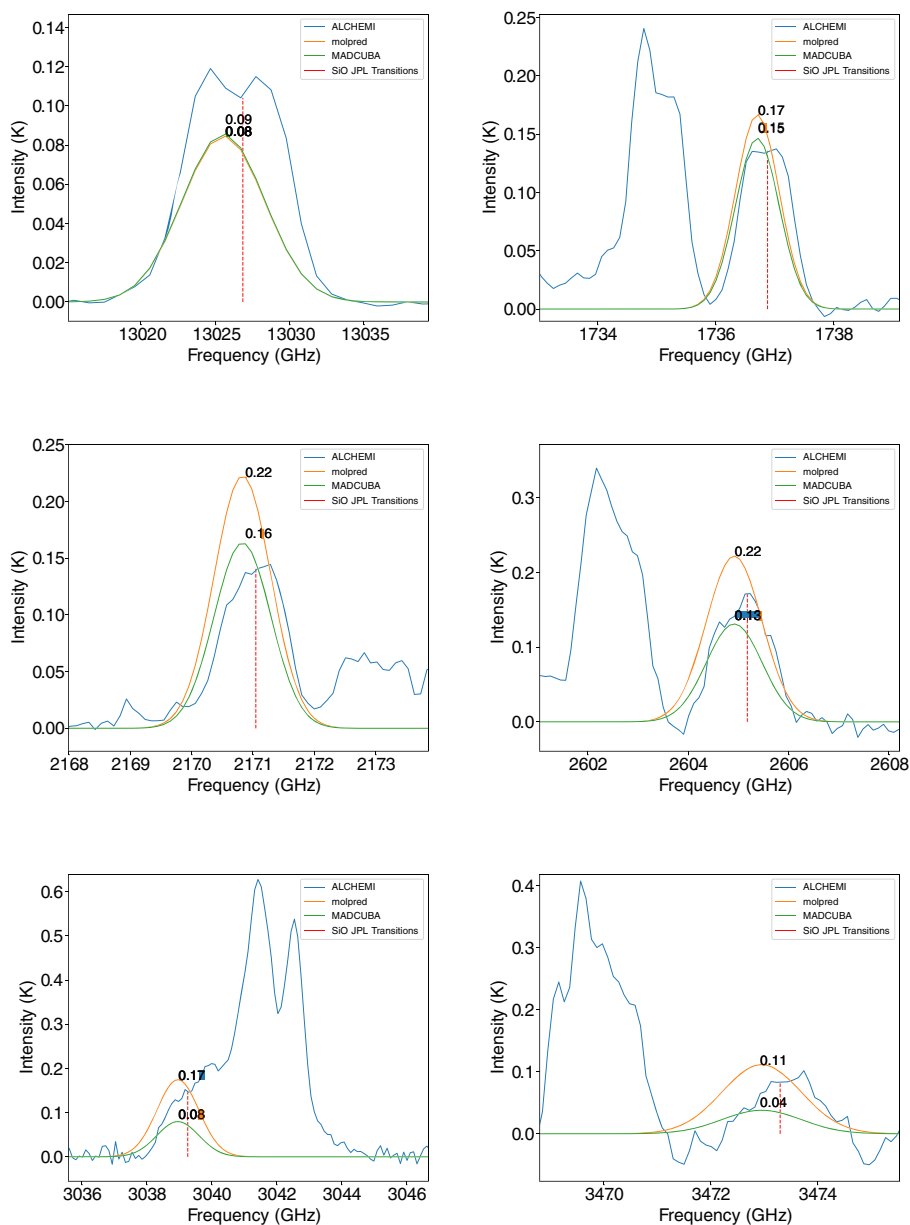
**Fig. 10** Similar to Fig. 9 for  $\text{HCO}^+$

For  $\text{HCO}^+$ , the analysis was done following the same process as used for CO. For this molecule, the ALCHEMI dataset contains data for 3 out of 4 transitions in the working frequency range. It is interesting that despite the missing transition, in most cases the *MolPred* predictions for  $\text{HCO}^+$  are closer to the ALCHEMI data, than those obtained with MADCUBA. We can see the prediction differences per transition in Table 11 and visually observe them in Fig. 10. In Fig. 6, we can see the saturation limit beyond which the column density cannot be predicted is  $14 \text{ cm}^{-2}$ . Therefore, we might expect the accuracy to be affected by this since the column density predicted by *MolPred* is close to this value. Despite this, the spectra from the *MolPred* predictions are good fit to the ALCHEMI dataset.

The ALCHEMI dataset contains 6 out of the 8 SiO transitions in the working frequency range. Similarly to  $\text{HCO}^+$ , we can see the prediction accuracy is good in a couple of transitions. Intensities predicted for SiO transitions can be examined on Table 12 and seen in Fig. 11. The SiO lines are quite weak and the *MolPred* fits give a small  $\log(N)$ . As a result, the temperature prediction suffers for SiO as discussed in Section 3.1. When used to generate an LTE spectrum, the high temperature predicted by *MolPred* for SiO, results in line intensities that are often too large.

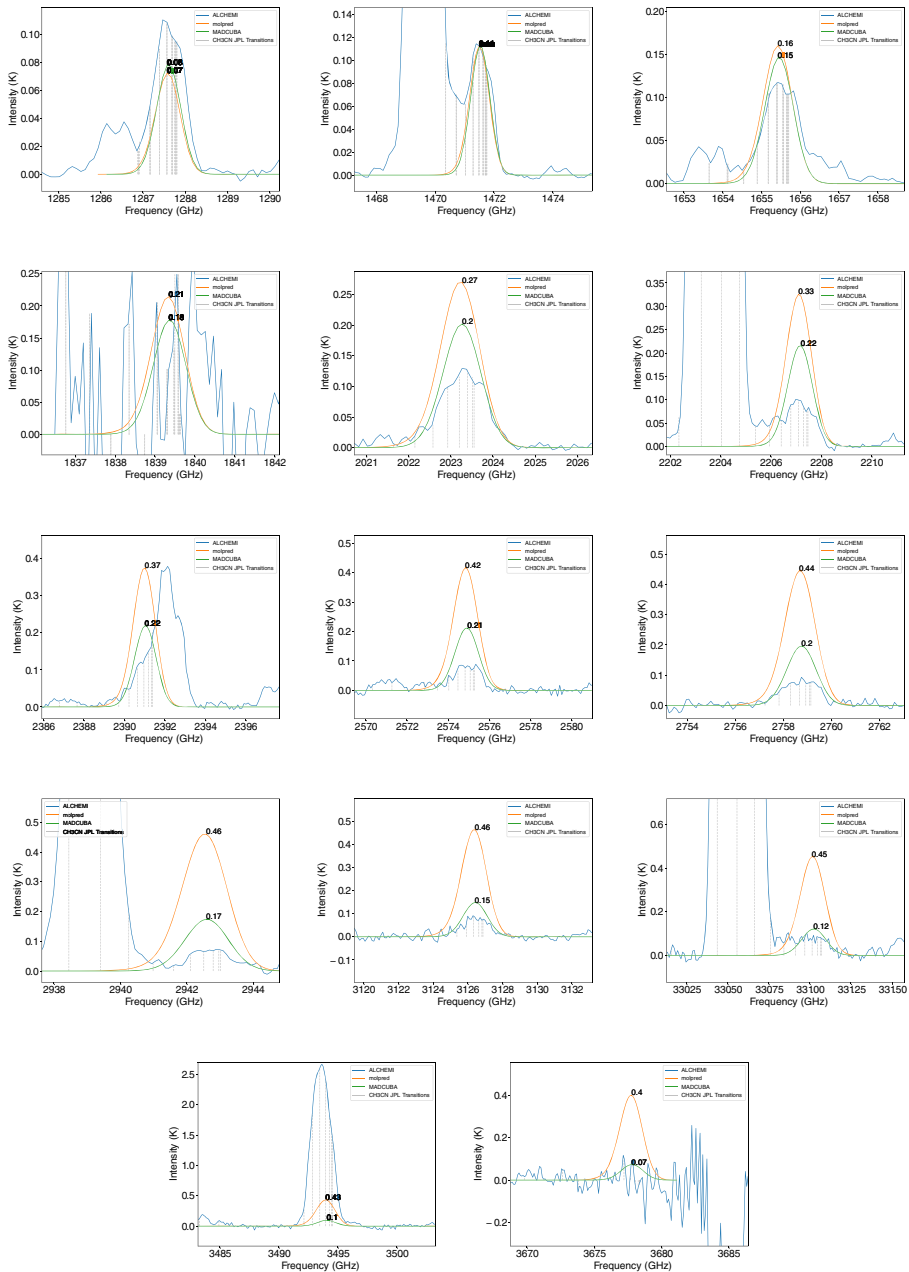
Finally we move to the molecule with the largest number of transitions in this exercise,  $\text{CH}_3\text{CN}$ . Since there are 14 groups of transitions with the same J quantum number, we will zoom in on each group to observe the behavior in Fig. 12. Predicted intensities generated for each group are described in Table 13. Since the molecule contains many transitions per group, we decided to select a group representative frequency (GRF) as the frequency that is closer to the *MolPred* and MADCUBA's prediction peaks. We included in the table, the ALCHEMI intensity at that frequency. *MolPred* predictions for  $\log(N)$  are close to  $13.4 \text{ cm}^{-2}$ , at which point the noise starts to strongly affect the predictions (see Fig. 6). This may explain why the temperature predicted by *MolPred* for  $\text{CH}_3\text{CN}$  differs so strongly from the MADCUBA prediction. We can see the effect of this inaccuracy on the intensities of several transitions, where the intensities derived from the *MolPred* prediction, are higher than the ALCHEMI intensity.

Figure 13 summarizes the prediction differences seen in previous tables. In many cases, *MolPred* does very well, giving predicted intensities that are closer to the data than MADCUBA. Where the results appear significantly different, these differences can be understood based on the way the parameters are obtained. In the case of MADCUBA, the entire spectrum is used to fit a group of Gaussian profiles at

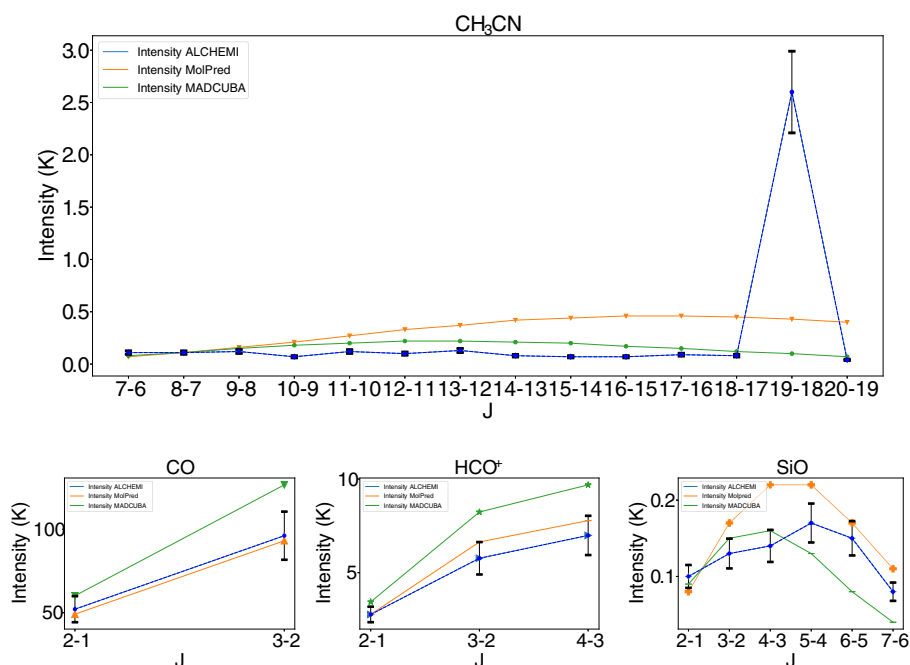


**Fig. 11** Similar to Fig. 9 for the SiO transitions in the ALCHEMI data

the frequencies of the molecular transitions. This includes a simultaneous fit to the width of the line, which can help constrain the effect of opacity mentioned in Section 3.4 of [9], and the fit may be more robust to individual channel variations due to line shape or noise. On the other hand, our neural networks are trained with single



**Fig. 12** Similar to Fig. 9 for the  $\text{CH}_3\text{CN}$  transitions in the ALCHEMI data. The transitions are grouped by the  $J$  quantum numbers and the frequency of each transition is indicated by silver lines



**Fig. 13** Real and predicted intensities as a function of  $J$  from MolPred (orange), MADCUBA (green) and the ALCHEMI data (blue). Intensities are at the rest frequency of the transitions. ALCHEMI collaboration has agreed on a 15% error bar, due to calibration uncertainties

intensity values for each transition. We can see how in the particular cases of CO and  $\text{HCO}^+$ , the prediction from *MolPred* follows very closely the intensity at the reference frequencies.

It is important to indicate that the training examples were calculated for a velocity of  $250 \text{ km s}^{-1}$  which is slightly different from the source velocity and thus the channel closest to each transition's rest frequency is not the peak intensity. This misalignment affects our neural network predictions since it uses the intensity of a single channel instead of the integrated emission.

Figure 12 also shows the strong impact of contamination from brighter transitions from other species on the fit results. This is severely affecting 19 – 18 transition of  $\text{CH}_3\text{CN}$ .

Despite its limitations, the *MolPred* predictions for CO and  $\text{HCO}^+$  are within the error boundaries of the ALCHEMI data set. We believe that its performance can be further tuned moving to a model which uses more information from the spectrum such as the full line profile or an integrated intensity.

## 4 Scalability and linear speedup

In order to observe the effects of the parallelization of the predictions, we devised a test where we would test four different scenarios based on the number of molecules

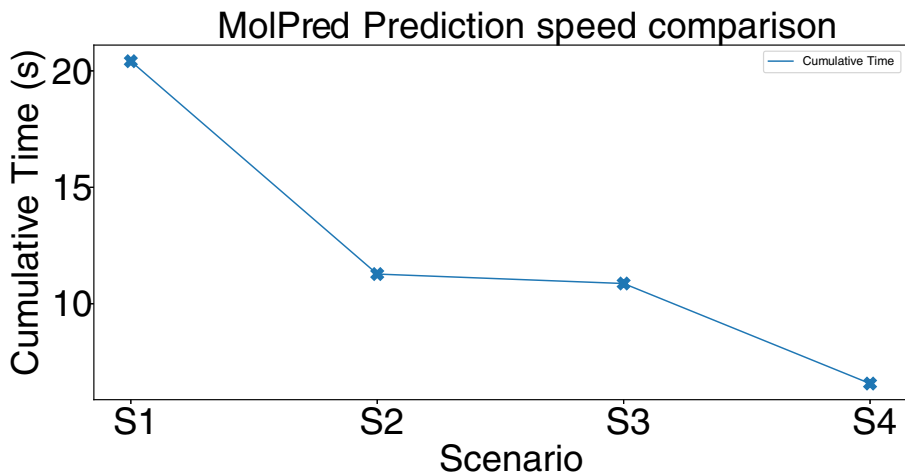
**Table 9** Prediction speed test, average times in seconds after 30 executions

	CO	HCO <sup>+</sup>	SiO	CH <sub>3</sub> CN	Cumulative
S1	5.122	5.066	5.246	4.981	20.415
S2	5.671		5.604		11.275
S3	5.888			4.981	10.869
S4	6.573				6.573

passed to *MolPred*. Each value was obtained by executing *MolPred* thirty times, and averaging the execution times. Scenario number one (S1) was to run *MolPred*, passing one molecule each time. Scenario number two (S2) consisted in passing pairs of molecules on each run, CO and HCO<sup>+</sup> in the first run, and SiO and CH<sub>3</sub>CN in the second run. Scenario number three (S3) consisted in passing three molecules CO, HCO<sup>+</sup> and SiO in one run. We kept the time for CH<sub>3</sub>CN from its individual pass. Scenario number four (S4), consisted in passing all four molecules in a single run.

We can observe the times in Table 9 where Scenario number one shows a linear cumulative time of twenty seconds. Checking scenario number two, we can observe a 45% decrease in time, as expected from running in parallel. Furthermore, we can observe that scenario number four displays a 68% time reduction, agreeing with the expected behaviour.

We can observe in Fig. 14 that the *MolPred* design, allows to have faster predictions for a larger number of molecules than if they were calculated in a serial way.

**Fig. 14** *MolPred* prediction speed comparison, average times in seconds after 30 executions

We can calculate the prediction *speedup* by applying (1), to Scenario 2, as shown in (2) and Scenario 4 in (3).

$$S(S2) = \frac{20.415}{11.275} = 1.810x \quad (2)$$

$$S(S4) = \frac{20.415}{6.573} = 3.105x \quad (3)$$

## 5 Conclusions and future work

We described a software package called *MolPred* which is able to extract the peak intensities of molecular transitions from spectra and use them to estimate the column density and temperature of the molecule. It is able to do this for CO, HCO<sup>+</sup>, SiO and CH<sub>3</sub>CN with a mean error of 1-9% on the predicted values when evaluated on synthetic data.

Molpred was also shown to perform well on real astronomical data. A spectrum from NGC 253 was processed by *MolPred* and the values obtained were similar to those found using the MADCUBA software package for the cases of CO and HCO<sup>+</sup>. This was despite the fact that some molecular transitions required for the networks were missing from the data. The predicted values of  $\log(N)$  and  $T_{ex}$  from *MolPred* were within 13% of those found using MADCUBA on average. The differences with MADCUBA are understood, and mostly related to the fact that *MolPred* is using single values per transition training and spectrum analysis rather than the whole spectral profile.

We concluded that the best way to train our networks, was by the use of a single GPU, and observed that increasing the batch size, allowed lower training times even achieving 87% faster training times, at the expense of a larger MAE, with an overall increase of 3%, however the relative increase was over 50% which motivated us to remain using lower batch sizes to achieve lower error levels. *MolPred's* parallel prediction design allowed to train all the molecules in 68% less time (or 3x faster) than what it would have taken to train them one after another.

Further work should include increasing the number of molecules and predicting a broader range of physical parameters such as the line width and source size, for which the whole line profile should be used. The number of molecules should be extended, and the capacity to predict highly blended spectrum should be explored. Another parameter to explore would be multiple velocity components. However, this will require further tuning of the networks as three of the four best neural networks in this work had three layers of 1024 nodes. This would be prohibitively large if many molecules were considered, each needing their own neural network. Therefore, either these networks must be greatly reduce or networks must be trained which can solve for more than one molecule.

## Appendix A

**Table 10** Peak intensities from spectra generated using the *MolPred* and MADCUBA predictions for CO

Molecule	J	Intensity <i>MolPred</i> (K)	Intensity MADCUBA (K)	Intensity ALCHEMI @ Transition Freq (K) @ (GHz)
CO	2-1	49.11	60.15	52.10 @ 230.5380
CO	3-2	93.11	126.60	96.14 @ 345.7959

**Table 11** Peak intensities from spectra generated using the *MolPred* and MADCUBA predictions for HCO<sup>+</sup>

Molecule	J	Intensity <i>MolPred</i> (K)	Intensity MADCUBA (K)	Intensity ALCHEMI @ Transition Freq (K) @ (GHz)
HCO+	2-1	2.77	3.44	2.77 @ 178.3750
HCO+	3-2	6.64	8.24	5.77 @ 267.5576
HCO+	4-3	7.78	9.69	6.99 @ 356.7342

**Table 12** Peak intensities from spectra generated using the *MolPred* and MADCUBA predictions for SiO

Molecule	J	Intensity <i>MolPred</i> (K)	Intensity MADCUBA (K)	Intensity ALCHEMI @ Transition Freq (K) @ (GHz)
SiO	3-2	0.08	0.09	0.10 @ 130.2686
SiO	4-3	0.17	0.15	0.13 @ 173.6884
SiO	5-4	0.22	0.16	0.14 @ 217.1050
SiO	6-5	0.22	0.13	0.17 @ 260.5180
SiO	7-6	0.17	0.08	0.15 @ 303.9270
SiO	8-7	0.11	0.04	0.08 @ 347.3306



**Table 13** Peak intensities from spectra generated using the *MolPred* and MADCUBA predictions for CH<sub>3</sub>CN. The ALCHEMI intensity at the Group Representative Frequency is also shown

Molecule	J	Intensity <i>MolPred</i> (K)	Intensity MADCUBA (K)	Intensity ALCHEMI @ GRF (K) @ (GHz)
CH <sub>3</sub> CN	7 – 6	0.07	0.08	0.11 @ 128.7577
CH <sub>3</sub> CN	8 – 7	0.11	0.11	0.11 @ 147.1499
CH <sub>3</sub> CN	9 – 8	0.16	0.15	0.12 @ 165.5415
CH <sub>3</sub> CN	10 – 9	0.21	0.18	0.07 @ 183.9320
CH <sub>3</sub> CN	11 – 10	0.27	0.20	0.12 @ 202.3204
CH <sub>3</sub> CN	12 – 11	0.33	0.22	0.10 @ 220.7090
CH <sub>3</sub> CN	13 – 12	0.37	0.22	0.13 @ 239.0968
CH <sub>3</sub> CN	14 – 13	0.42	0.21	0.08 @ 257.4830
CH <sub>3</sub> CN	15 – 14	0.44	0.20	0.07 @ 275.8678
CH <sub>3</sub> CN	16 – 15	0.46	0.17	0.07 @ 294.2514
CH <sub>3</sub> CN	17 – 16	0.46	0.15	0.09 @ 312.6336
CH <sub>3</sub> CN	18 – 17	0.45	0.12	0.08 @ 331.0143
CH <sub>3</sub> CN	19 – 18	0.43	0.10	2.6 @ 349.3450
CH <sub>3</sub> CN	20 – 19	0.40	0.07	0.04 @ 367.0777

**Acknowledgements** A.B. wishes to thank Dr. Diego Mardones for his contribution to the early stages of this work. Also, to acknowledge support from the Federico Santa María Technical University General Directorate for Research and Postgraduate Studies (DGIP). JH and SV are funded by the European Research Council (ERC) Advanced Grant MOPPEX 833460. V.M.R. acknowledges support from the Comunidad de Madrid through the Atracción de Talento Investigador Modalidad 1 (Doctores con experiencia) Grant (COOL: Cosmic Origins Of Life; 2019-T1/TIC-15379; PI: V.M. Rivilla).

## References

- Berriman, G.B., Groom, S.L.: How will astronomy archives survive the data tsunami? Commun. ACM **54**(12), 52–56 (2011). <https://doi.org/10.1145/2043174.2043190>
- Cernicharo, J.: Laboratory astrophysics and astrochemistry in the herschel/alma era. EAS Publ. Ser. **58**, 251–261 (2012). <https://doi.org/10.1051/eas/1258040>
- Chollet, F., et al.: Keras. <https://github.com/fchollet/keras> (2015)
- Farnes, J., Mort, B., Dulwich, F., Salvini, S., Armour, W.: Science Pipelines for the Square Kilometre Array. arXiv:1811.08272 (2018)
- Jurić, M., Kantor, J., Lim, K., Lupton, R.H., Dubois-Felsmann, G., Jenness, T., Axelrod, T.S., Aleksić, J., Allsman, R.A., AlSayyad, Y., Alt, J., Armstrong, R., Basney, J., Becker, A.C., Becla, J., Bickerton, S.J., Biswas, R., Bosch, J., Boutigny, D., Carrasco Kind, M., Ciardi, D.R., Connolly, A.J., Daniel, S.F., Daues, G.E., Economou, F., Chiang, H.-F., Fausti, A., Fisher-Levine, M., Freemon, D.M., Gee, P., Gris, P., Hernandez, F., Hoblitt, J., Ivezić, Ž., Jammes, F., Jevremović, D., Jones, R.L., Bryce Kalmbach, J., Kasliwal, V.P., Krughoff, K.S., Lang, D., Lurie, J., Lust, N.B., Mullally, F., MacArthur, L.A., Melchior, P., Moeyens, J., Nidever, D.L., Owen, R., Parejko, J.K., Peterson, J.M., Petravick, D., Pietrowicz, S.R., Price, P.A., Reiss, D.J., Shaw, R.A., Sick, J., Slater, C.T., Strauss, M.A., Sullivan, I.S., Swinbank, J.D., Van Dyk, S., Vujčić, V., Withers, A., Yoachim, P., LSST Project, f.t.: The LSST Data Management System (2015)
- Kingma, D.P.: Ba, J. Adam: A method for stochastic optimization (2017)

7. Lightfoot, J., Kosugi, G., Wyrowski, F., Zapata, L., Muders, D., Boone, F., Tsutsumi, T., Davis, L., Wilson, C., Shepherd, D.: ALMA Pipeline Heuristics. In: Argyle, R.W., Bunclark, P.S., Lewis, J.R. (eds.) *Astronomical Data Analysis Software and Systems XVII*. Vol 394 of *Astronomical Society of the Pacific Conference Series*, p. 573 (2008)
8. Mach, M., Köhler, R., Czoske, O., Leschinski, K., Zeilinger, W.W., Kausch, W., Ratzka, T., Leitzinger, M., Greimel, R., Przybilla, N., Schaffenroth, V., Güdel, M., Brandl, B.R.: Data reduction software for the Mid-Infrared E-ELT Imager and Spectrograph (METIS) for the European Extremely Large Telescope (E-ELT). In: *Software and Cyberinfrastructure for Astronomy IV*, vol. 9913, p. 991327. SPIE (2016)
9. Martín, S., Martín-Pintado, J., Blanco-Sánchez, C., Rivilla, V.M., Rodríguez-Franco, A., Rico-Villas, F.: Spectral line identification and modelling (slim) in the madrid data cube analysis (madcuba) package. *Astron. Astrophys.* **631**, A159 (2019). <https://doi.org/10.1051/0004-6361/201936144>
10. McMullin, J.P., Waters, B., Schiebel, D., Young, W., Golap, K.: Casa architecture and applications. In: *Astronomical Data Analysis Software and Systems XVI (ADASS XVI)*, pp. 127–130 (2007)
11. Möller, T., Schilke, P.: Manual for xclass-interface. [https://www.astro.uni-koeln.de/wd-schilke/myXCLASS/XCLASS-Interface\\_No-NR\\_Linux\\_version\\_1.1.6.zip](https://www.astro.uni-koeln.de/wd-schilke/myXCLASS/XCLASS-Interface_No-NR_Linux_version_1.1.6.zip) (2015)
12. Nakajima, T., Takano, S., Kohno, K., Harada, N., Herbst, E., Tamura, Y., Izumi, T., Taniguchi, A., Tosaki, T.: A multi-transition study of molecules toward NGC 1068 based on high-resolution imaging observations with ALMA. *Publ. Astron. Soc. Jpn.* **67**(1), 8 (2015)
13. Pickett, H., Poynter, R., Cohen, E., Delitsky, M., Pearson, J., Müller, H.: Submillimeter, millimeter, and microwave spectral line catalog. *J. Quant. Spectrosc. Radiat. Transf.* **60**(5), 883–890 (1998). <https://www.sciencedirect.com/science/article/pii/S0022407398000910>
14. Rosenblatt, F.: The perceptron: A probabilistic model for information storage and organization in the brain. *Psychol. Rev.* **65**, 386–408 (1958)
15. Schilke, P., Möller, T., Comito, C., Sánchez-Monge, Á., Schmiedeke, A., Zernickel, A.: Taming the dragon: automatic line fitting of ALMA data. In: Iono, D., Tatematsu, K., Wootten, A., Testi, L. (eds.) *Revolution in Astronomy with ALMA: The Third Year*. Vol. 499 of *Astronomical Society of the Pacific Conference Series*, p. 195 (2015)
16. Swings, P., Rosenfeld, L.: Considerations regarding interstellar molecules. *Astrophys. J.* **86**, 483–486 (1937)
17. Teuben, P.: Admit 0.5.2 documentation. <http://tinyurl.com/osbu75s> (2015)
18. van der Tak, F.F.S., Black, J.H., Schöier, F.L., Jansen, D.J., van Dishoeck, E.F.: A computer program for fast non-lte analysis of interstellar line spectra\* - with diagnostic plots to interpret observed line intensity ratios. *A&A* **468**(2), 627–635 (2007). <https://doi.org/10.1051/0004-6361:20066820>
19. Vastel, C., Bottinelli, S., Caux, E., Glorian, J.-M., Boiziot, M.: CASSIS: a tool to visualize and analyse instrumental and synthetic spectra. In: Martins, F., Boissier, S., Buat, V., Cambrésy, L., Petit, P. (eds.) *SF2A-2015: Proceedings of the Annual Meeting of the French Society of Astronomy and Astrophysics*, pp. 313–316 (2015)
20. Woon, D.: Lists of interstellar and circumstellar molecules. <http://tinyurl.com/m69l6qq> (2020)

## Affiliations

**Alejandro Barrientos**<sup>1,2</sup>  · **Jonathan Holdship**<sup>3,4</sup>  · **Mauricio Solar**<sup>1</sup>  · **Sergio Martín**<sup>5,1</sup>  · **Víctor M. Rivilla**<sup>6,7</sup>  · **Serena Viti**<sup>3,4</sup>  · **Jeff Mangum**<sup>8</sup>  · **Nanase Harada**<sup>9,10,11</sup>  · **Kazushi Sakamoto**<sup>10</sup>  · **Sébastien Muller**<sup>12</sup>  · **Kunihiko Tanaka**<sup>13</sup>  · **Yuki Yoshimura**<sup>14</sup>  · **Kouichiro Nakanishi**<sup>9,11</sup>  · **Rubén Herrero-Illana**<sup>5,15</sup>  · **Stefanie Mühle**<sup>16</sup>  · **Rebeca Aladro**<sup>17</sup>  · **Susanne Aalto**<sup>12</sup>  · **Christian Henkel**<sup>17,18</sup>  · **Pedro Humire**<sup>17</sup> 

<sup>1</sup> Federico Santa María Technical University, Vicuña Mackenna 3939, San Joaquín, Santiago, Chile

<sup>2</sup> Joint ALMA Observatory, Alonso de Cordova 3107, Vitacura, Santiago, Chile

<sup>3</sup> Leiden Observatory, Leiden University, PO Box 9513, 2300 RA Leiden, The Netherlands

<sup>4</sup> Department of Physics and Astronomy, University College London, Gower Street, WC1E 6BT, London, UK

<sup>5</sup> European Southern Observatory, Alonso de Cordova 3107, Vitacura, Santiago, Chile

<sup>6</sup> Centro de Astrobiología (CSIC-INTA), Ctra Ajalvir km 4, 28850, Torrejón de Ardoz, Madrid, Spain

<sup>7</sup> INAF-Osservatorio Astrofisico di Arcetri, Largo Enrico Fermi 5, 50125, Florence, Italy

<sup>8</sup> National Radio Astronomy Observatory, 520 Edgemont Road, Charlottesville, VA 22903-2475, USA

<sup>9</sup> National Astronomical Observatory of Japan, 2-21-1 Osawa, Mitaka, Tokyo 181-8588, Japan

<sup>10</sup> Institute of Astronomy and Astrophysics, Academia Sinica, 11F of AS/NTU Astronomy-Mathematics Building, No.1, Sec. 4, Roosevelt Rd, Taipei 10617, Taiwan

<sup>11</sup> Department of Astronomy, School of Science, The Graduate University for Advanced Studies (SOKENDAI), 2-21-1 Osawa, Mitaka, Tokyo, 181-1855 Japan

<sup>12</sup> Department of Space, Earth and Environment, Onsala Space Observatory, Chalmers University of Technology, SE-43992 Onsala, Sweden

<sup>13</sup> Department of Physics, Faculty of Science and Technology, Keio University, 3-14-1 Hiyoshi, Yokohama, Kanagawa 223–8522 Japan

<sup>14</sup> Institute of Astronomy, Graduate School of Science, The University of Tokyo, 2-21-1 Osawa, Mitaka, Tokyo 181-0015, Japan

<sup>15</sup> Institute of Space Sciences (ICE, CSIC), Campus UAB, Carrer de Magrans, E-08193 Barcelona, Spain

<sup>16</sup> Argelander-Institut für Astronomie, Universität Bonn, Auf dem Hügel 71, D-53121 Bonn, Germany

<sup>17</sup> Max-Planck-Institut für Radioastronomie, Auf dem Hügel 69, 53121 Bonn, Germany

<sup>18</sup> Astronomy Department, Faculty of Science, King Abdulaziz University, P. O. Box 80203, Jeddah 21589, Saudi Arabia