Draft version May 3, 2025
Typeset using LaTeX **twocolumn** style in AASTeX631

Classifying Merging Galaxies and Estimating Redshifts Using a Dual-Objective CNN

Jenna Kempster-Taylor[1]

[1] *Harvard University*

ABSTRACT

Galaxy mergers are crucial events in the formation of cosmic structure, morphological transformations, star formations bursts and dark matter halo growth (Rodriguez-Gomez et al. 2017). Merger rates are predicted to increase with redshift, making the identification of merger events and their redshifts areas of active research within fields such as galaxy evolution and the ΛCDM cosmological model (Stewart et al. 2009).

This project develops a dual-objective convolutional neural network (CNN) to classify galaxies as merging or non-merging and provide an estimate of their redshift, from image data as the only input. For the baseline model, logistic regression was used for the classification task. This achieved a high overall accuracy (94%) but performed very poorly on mergers (F1-score = 0.23). Linear regression was used for redshift estimates and achieved an R^2 score of 0.37.

The final CNN was trained on a balanced subset of images from the Galaxy10_DECals dataset (Bordelon & Gully-Santiago 2019). Multiple models were tested. The best classification model was a four-layer CNN with separate classification and regression heads during training, though the regression head was later removed to improve classification accuracy. It achieved a validation accuracy of 71% on RGB and 65% on greyscale images. It achieved an F-1 score of 0.67 for mergers and 0.58 for non-mergers. This model had stopped exploring redshift estimates as the two objectives were interfering with one and other's accuracies. Redshift prediction remained challenging, with the best model achieving a mean squared error (MSE) of 0.007964 and R^2 of –0.1608. These results demonstrate the potential effectiveness of CNNs for galaxy classification from imaging data, while highlighting the difficulty of accurate redshift estimation from images alone.

Keywords: galaxies interactions; mergers — methods: data analysis; convolutional neural nets

1. INTRODUCTION

Galaxy mergers are key events in cosmic structure formation, playing a central role in galaxy evolution. They influence star formation, morphological transformations, and dark matter halo growth (Rodriguez-Gomez et al. 2017). Theoretical models predict that merger rates increase with redshift and galaxy mass, particularly for mergers at $z > 3$ (Stewart et al. 2009). Detecting mergers across large imaging surveys—and estimating their redshifts—is therefore vital for understanding the growth and transformation of galaxies within the ΛCDM cosmological framework.

Traditional machine learning approaches have struggled to classify mergers directly from raw pixel data, particularly due to subtle morphological features and class imbalances. In contrast, convolutional neural networks (CNNs) have demonstrated success in galaxy classification tasks by learning hierarchical spatial features directly from images. For example, the DeepMerge CNN achieved up to 76% to 79% accuracy in merger identification at high redshifts (Ćiprijanović et al. 2020).

In this project, we develop a dual-objective CNN to classify galaxies as merging or non-merging, and simultaneously predict their redshift using imaging data. We use a subset of the Galaxy10_DECals dataset (Bordelon & Gully-Santiago 2019) and explore multiple model architectures to assess how model complexity affects classification and regression performance. We compare our CNN against logistic and linear regression baselines to evaluate its performance.

The following paper is organised as follows: Section 2 describes the GalaxyZoo dataset and preprocessing steps used to prepare the input data; Section 3 discusses the different CNN architectures explored and training procedures used; Section 4 presents the results of classification and redshift estimation from the differ-

ent architectures; Section 5 discusses the challenges of this project; Section 6 suggests direction for future work.

2. DATA

2.1. *Dataset Overview*

We used galaxy images from the Galaxy10_DECals dataset, which contains RGB images, galaxy-type classifications, and redshift values.

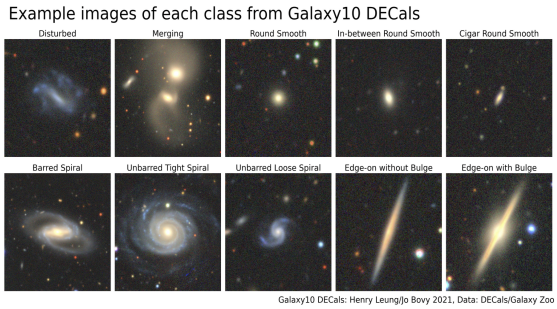

Figure 1. Example Images in the Galaxy10_DECals dataset.

2.2. *Data Acquisition and Cleaning*

Loading the Dataset: The Galaxy10_DECals dataset was loaded using the HDF5 format. Firstly the arrays were extracted as RGB images, and then later on were extracted to grayscale images. The dataset contains 17,736 samples, each with an associated classification and spectroscopic redshift.

Dataset Cleaning: Initial checks exposed missing redshift estimates so these images were removed from test/train subset. 92 images were removed with NaN redshifts.

Data Normalisation: All image pixel values were converted to `float32` and normalised to the [0, 1]. This step was to stabilise training and help train the CNN.

Merger Extraction: Using the Galaxy10 labels, mergers (class index 0) were revealed to make up on 6.1% of dataset. This lead to a limited training and validation set size, as class balancing was used. These mergers were all extracted to use for either training or validation

Visualisation and Verification: Random samples of merger images were plotted with redshift overlays to visually inspect data quality and help diagnose any future issues.

2.3. *Subset Selection and Preprocessing*

Sampling for Baseline:

For the baseline model, a subset of 1413 was selected and `train_test_split` was used. Flattened RGB images using binary labels (1 = merger, 0 = non-merger), and merger status extracted from one-hot encoded labels were used as inputs. The merging/non-merging split of 1328 non-mergers and 85 mergers led to bias towards the majority (non-mergers). As the focus was on merger galaxies, the RandomOverSampler from imblearn was used to duplicate merger images (Lemaître et al. 2017). The subset split for baseline was 1328 non-mergers and 531 mergers.

Sampling for CNN:

From accuracy results in baseline testing and initial CNN models, a larger subset of 2146 images was used. This contained all available 1073 mergers from dataset and 1073 non-mergers. The decision was made to go 50% mergers and 50% non-mergers to try and optimise training on merging class.

Redshift Removal:

NaN redshifts were removed from this subset.

3. METHODOLOGY

This project aimed to develop a dual-objective convolutional neural network (CNN) that accurately classified galaxies as merging or non-merging and simultaneously estimated their redshift using only image data. The methodology consisted of three main stages: establishing baseline models, constructing an appropriate dataset, and developing and evaluating CNN architectures.

3.1. *Baseline Models and Sampling Strategy*

To evaluate the effectiveness of deep learning approaches, we first established a baseline using logistic regression for merger classification and linear regression for redshift prediction. Images were flattened from their original 256x256x3 RGB format into 1D vectors as input for the linear models. The original Galaxy10_DECals labels were one-hot encoded, and a binary classification task was constructed: mergers (class index 0) were labelled as 1, and all other classes as 0.

Given the extreme class imbalance (mergers made up only 6.1% of the dataset), we applied random oversampling using the `RandomOverSampler` class from the `imbalanced-learn` library (Lemaître et al. 2017). Oversampling was applied only to the training data to avoid data leakage (train data in test data). The final training set for the baseline model included 1328 non-mergers and 531 mergers. To prevent duplicates of train data in test data, the max cosine similarity was monitored and kept below 1.

These preprocessed inputs were used to train baseline logistic regression model for classification, and a

separate linear regression model for redshift prediction. These served as reference points for evaluating the performance of more complex models, and to give us insight into the dataset limitations.

3.2. *Dataset Balancing and Subset Construction*

For CNN training, a larger and more balanced subset of 2,146 images was constructed by combining all 1,073 available merger examples with an equal number of randomly sampled non-mergers. This ensured a 50:50 class distribution, which improved learning stability and allowed the model to better distinguish merger features.

Images were normalised to have pixel values in the range [0, 1] and cast to `float32` for training. Samples with missing redshift values were removed from the dataset. Experiments were conducted with both RGB and grayscale versions of the data to consider the effect of colour information on model performance.

3.3. *CNN Architecture Development*

The core of the project involved designing and iteratively improving a dual-head CNN model. All architectures shared a general structure: a convolutional feature extractor feeding into two fully connected branches—one for classification and one for regression.

The initial version (`DualHeadCNN`) used three convolutional blocks and a shared dense representation, followed by parallel classification and regression outputs. Subsequent versions introduced deeper architectures, regularisation, and optimised training strategies:

- `DualHeadCNNv2`: Added a fourth convolutional layer, batch normalisation after each convolution, and dropout ($p = 0.3$). This version improved classification stability and reduced overfitting. This model achieved the highest classification performance for RGB.
- `DualHeadCNNv3`: Used only two convolutional layers with increased regularisation and stronger augmentation. This served as a control to evaluate model depth vs. generalisation.
- `DualHeadCNNv4`: Five convolutional blocks, including paired convolutions per block (Conv → BN → ReLU → Conv → BN → ReLU), and concluded with an adaptive average pooling layer. Dropout was increased to $p = 0.6$, and learning rate decay with early stopping was used. This model achieved the highest classification performance for greyscale.
- `DualHeadCNNv5`: In the final experiments, the regression head was removed, and the model was trained solely for classification. This resolved observed interference between the two tasks and improved F1-score on merger detection.

All models used ReLU activations, max pooling, and the Adam optimiser (Kingma & Ba 2017). Classification loss was computed using cross-entropy, and when included, regression loss used mean squared error. A composite loss function of the form:

$$\mathcal{L}_{\text{total}} = \alpha \cdot \mathcal{L}_{\text{classification}} + \beta \cdot \mathcal{L}_{\text{regression}}$$

was used for dual-task training, with α and β set to emphasise classification (e.g., $\alpha = 10$, $\beta = 1$). When training for classification only, $\mathcal{L}_{\text{classification}}$ was used exclusively.

4. RESULTS

4.1. *Baseline Model Results*

- Classification Accuracy:
  Table 1 shows the performance of a baseline logistic regression classifier trained to assign merging and non-merging galaxies using oversampled image data. While overall accuracy is high at 94 percent, this is driven by strong performance on the majority class (non-mergers) (typical for logistic regression) . The classifier achieves very high precision (0.95) and recall (0.99) for non-mergers, resulting in a high weighted F1-score.
  However, the model struggles to correctly identify mergers, achieving a low recall of 0.14 and an F1-score of only 0.23. This imbalance is also seen in the confusion matrix, where only 3 out of 21 actual mergers are correctly classified This indicates that the classifier frequently misclassifies mergers as non-mergers despite oversampling.
  The large class imbalance and the subtlety of merger features in raw pixel data likely contribute to this underperformance, highlighting the need for models catered towards image data such as convolutional neural networks (CNNs).
- Redshift Baseline Linear Regression:
  The model achieved a mean squared error (MSE) of 0.00098 and an R^2 score of 0.365, indicating moderate accuracy. Despite the simplicity of the model, the predictions were reasonably aligned with true redshift values. 81.8% of redshift predictions on the test set fell within one standard deviation of the true values, suggesting that the model is somewhat accurate at predicting redshift. These results highlight the potential for redshift estimation based on imaging data alone and provide a solid baseline for comparison against more

Table 1. Classification Report for Baseline Logistic Regression Model

Class	Precision	Recall	F1-Score	Support
Non-Merger (0)	0.95	0.99	0.97	333
Merger (1)	0.60	0.14	0.23	21
Accuracy		0.94		
Macro Avg	0.77	0.57	0.60	354
Weighted Avg	0.93	0.94	0.93	354

Table 1. Confusion Matrix for Baseline Logistic Regression Model

		Predicted	
		Non-Merger (0)	**Merger (1)**
Actual	**Non-Merger (0)**	331	2
	Merger (1)	18	3

complex convolutional architectures in the dual-objective CNN developed later.

4.2. *Classification Performance Results and Workflow*

- DualHeadCNNv1 to DualHeadCNNv2:
To address overfitting observed in the initial CNN (v1), dropout was introduced with probability 0.3 (Srivastava et al. 2014). As seen in Figure 2, training accuracy increased rapidly, reaching over 90% by epoch 10, while validation accuracy plateaued near 59%. The gap between training and validation performance—visible in both accuracy and loss curves—indicates the model was overfitting training examples rather than generalising. Early-stopping was introduced, where if validation accuracy had not improved over 10 epochs, then training was finished. Adding dropout ($p = 0.3$) and early stopping allowed the accuracy in CNN v2 to stabilise and delay overfitting. Another convolutional layer was added too to increase model depth and assess whether performance was better. Batch normalisation was also introduced to prevent overfitting (Ioffe & Szegedy 2015). Input data was augmented (via flips and rotations) to increase model generalisation via `torchvision.transforms`.

  CNNv2 peaked at a classification performance of 71% and proved much more accurate than baseline logistic regression models. It was more effective at classifying non-mergers than mergers.
  CNNv2 was explored further with more layers, different learning rates, different constants in the loss function, yet the accuracy began to decrease with further models. Therefore, the decision was made to explore the performance of CNN with greyscale images.

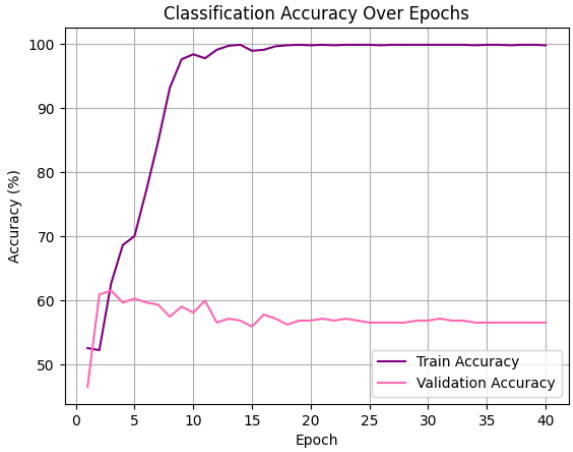

Figure 2. Classification accuracy over epochs for CNN v1. The widening gap between training and validation accuracy indicates overfitting.

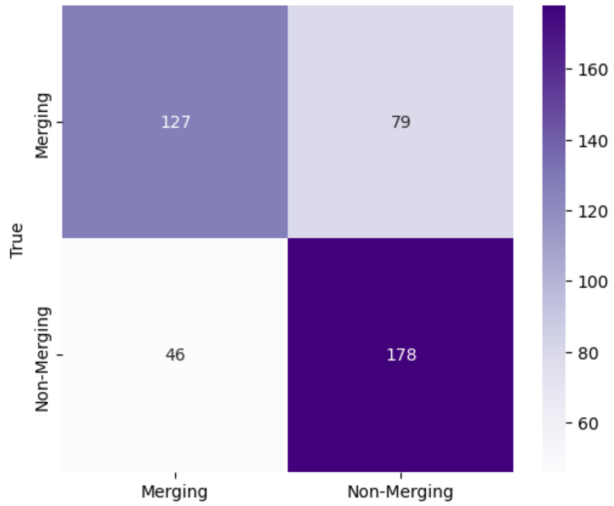

Figure 3. Confusion matrix for DualHeadCNNv2 with RGB input. Shows strong merger recovery and better class balance than the baseline.

- Blocking Regression Head: Despite using different weightings in loss function, regression head would interfere with classification accuracy. It seemed to be a limiting factor to the model and upon pivoting to greyscale classification was prioritised. Redshift estimates were unblocked for CNN v4 but were unreasonable and not useful.

- DualHeadCNNv2 to DualHeadCNNv3:
  While CNN v2 achieved the best overall performance using RGB input, it remained prone to overfitting despite dropout regularisation. To explore the generalisation of the model, greyscale was then used as an input. CNN v3 had a simple architecture of 2 layers and greyscale images were augmented. Performance dropped to 57%, which indicated that the CNN was not deep enough to represent the data effectively.

- DualHeadCNNv3 to DualHeadCNNv4:
  To explore the impact of depth on classification accuracy, CNNv4 had a deeper structure with five convolutional blocks and paired convolutions per stage. Batch normalisation was applied at each convolutional layer and dropout was increased to $p = 0.6$. CNNv4 acheived 67% validation accuracy and a merger's F1-score of 0.67. The F1 score is the same as CNNv2 despite using greyscale images, suggesting model architecture can compensate for less input information.

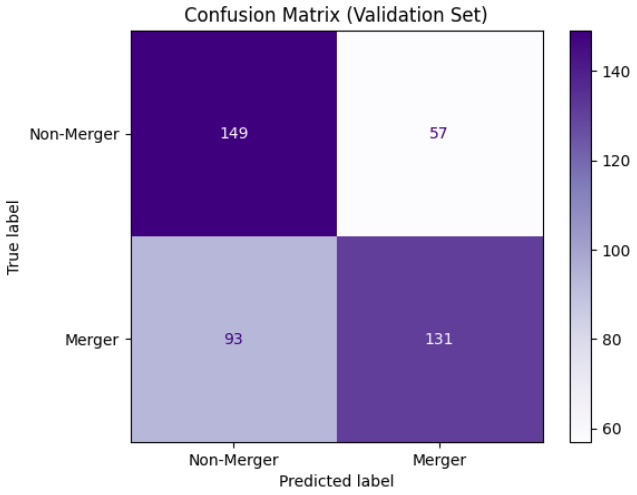

Figure 4. Confusion matrix for DualHeadCNNv4 with greyscale input. Performance is slightly lower than RGB but stronger on non-mergers.

Table 3. Performance of CNN Model Variants for Merger Classification and Redshift Regression

Model	Accuracy	F1 (Merger)	F1 (Non-Merger)	MSE	R^2	Notes
Baseline (Logistic Reg.)	0.94	0.23	0.97	–	–	Flattened RGB input, oversampled
DualHeadCNNv1 (RGB)	0.59	0.59	0.59	0.00796	–4.13	3 conv layers, dual-output model
DualHeadCNNv2 (RGB)	0.71	0.67	0.74	0.00180	–0.16	4 conv layers, batch norm, dropout=0.3
DualHeadCNNv3 (Simpler, Gray)	0.57	0.53	0.61	–	–	2 conv layers, strong greyscale augmentation
DualHeadCNNv4 (Deeper, Gray)	0.66	0.67	0.64	–	–	5 conv blocks, no regression head
Best RGB (CNN v2)	**0.71**	**0.67**	**0.74**	**0.00180**	**–0.16**	RGB input with dual-task loss (best overall)
Best Gray (CNN v4)	**0.66**	**0.67**	**0.64**	–	–	Greyscale-only, classification-focused

4.3. *Redshift Regression Performance*

- Redshift regression was explored as a secondary objective in the dual-head CNN model. CNNv2 with loss weight: $\alpha = 10$ and $\beta = 1$ achieved some alignment between predicted and true redshift. The mean squared error (MSE) was 0.00180 and R^2 of –0.16. This indicates that the CNN is not extracting any useful information from input data (RGB in this case). The resulting scatter plot (Figure 5) demonstrated a weak positive trend but mostly emphasises the inaccuracy of prediction. At higher redshifts, the estimate is worse.

- Since a weighted loss function was used, adjusting the weights could improve performance. Yet when this was explored, performance was worse. This suggests the model performs very poorly at redshift estimation, and given it more weight just collapsed the network to predicting a narrow range. This could be slightly seen in Figure 5 but is much more obvious in Figure 6.

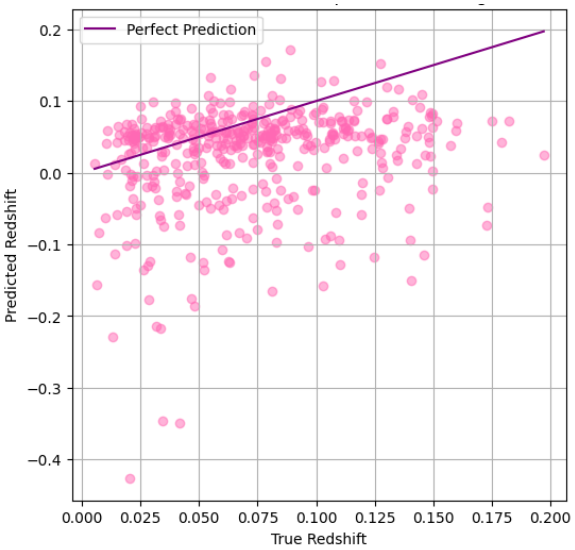

Figure 5. Redshift prediction scatter for CNN v2 ($\alpha = 10$, $\beta = 1$). Some correlation is present, but variance remains high.

- Classification was the primary goal and regression did not demonstrate any positive direction, so the decision to block the regression head to explore classification was made.

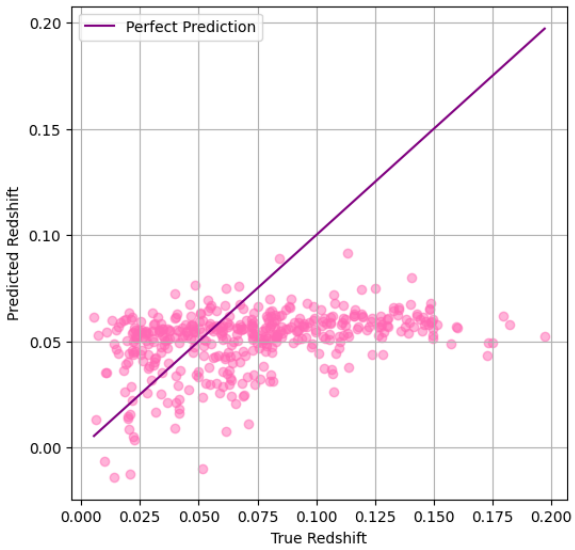

Figure 6. Redshift prediction scatter for CNN v2 with increased regression weight ($\alpha = 8$, $\beta = 4$). Predictions collapsed to a narrow range.

5. DISCUSSION AND CONCLUSIONS

This project explored using dual-head convolutional neural networks to classify merging galaxies and estimate redshifts from imaging data alone. Multiple CNN architectures were developed and evaluated, progressing from a baseline logistic regression model to increasingly complex deep learning models. While classification performance improved significantly through different model complexities, regularisation, and class balancing, redshift regression remained a major challenge.

The best classification results were achieved by CNN v2, which used RGB inputs and four convolutional layers with dropout regularisation. This model achieved 71% validation accuracy and an F1-score of 0.67 for mergers. Comparable performance was achieved on greyscale images with CNN v4, which suggests that strong architecture can compensate for the absence of colour data.

Despite the improvement from baseline, classification accuracy was lower than expected. This is likely due to the following factors;

Limited dataset: Mergers were only 6.1% of the Galaxy10_DECals dataset and using an oversampled split was key to classification performance on mergers. This restricted the amount of data that could be trained on and made it harder to generalise.

Morphological Similarities in Images: From examining the Galaxy10_DECals dataset, some images of mergers and non-mergers labelled "disturbed" look very

similar, which could be leading to misclassification. It is important to note that it is hard to establish a point in which two galaxies are considered merging in images. There could be galaxies close to one and other in an image but not interacting. These morphological properties and image similarities could lead to a ceiling on the model's training.

As expected, estimating redshift proved even more difficult than the classification task. The network collapsed to a narrow range of estimates which were very inaccurate. This aligns with scientific expectations as redshift is not entirely represented in morphological structure and it is very difficult to calculate from image data alone. Without emission lines or spectral features, regression is very inaccurate.

To conclude, this project demonstrated that CNNs can classify mergers from both RGB and greyscale image data with reasonable accuracy. Key factors to its success is minimising overfitting and augmenting data.

Code Availability:
All code used in this project is available on this colab notebook:https://colab.research.google.com/drive/14z3U6VNkb5bcXKwU0hpuSrqbwLNwRlRW?usp=sharing

6. FUTURE WORK

This project concludes with several directions to be explored.

Firstly, redshift estimation from imaging alone is fundamentally limited. Further models could incorporate additional data such as spectra and photometric data across different bands. This data is influenced heavily by redshift and more information can be extracted to then predict redshift. For example, emission lines from spectra could be used.

Secondly, the limitations of the dataset size strongly influenced the classification accuracy. While it is not certain whether this was the strongest limitation, it would be interesting to explore the models on a larger dataset.

Thirdly, the classification between mergers and non-mergers may be too vague and exploring the task of classifying between two distinct classes may be simpler. For example "Merging" images and "Round Smooth" images could be used. This would reduce ambiguity surrounding training labels and may yield higher performance.

Finally, future models could explore more focused classification tasks. For example, rather than distinguishing mergers from all other classes, the model could be trained to distinguish between specific well-separated morphologies, such as mergers versus undisturbed ellipticals. This approach could reduce ambiguity in the training labels and lead to more robust model predictions.

More advanced techniques such as self-supervised learning or transfer learning from larger astronomy-focused models could also prove to be effective and yield interesting results.

REFERENCES

Bordelon, D., & Gully-Santiago, M. 2019, Galaxy10 DECals Dataset, https://astronn.readthedocs.io/en/latest/galaxy10.html

Ioffe, S., & Szegedy, C. 2015, in Proceedings of Machine Learning Research, Vol. 37, Proceedings of the 32nd International Conference on Machine Learning, ed. F. Bach & D. Blei (Lille, France: PMLR), 448–456. https://proceedings.mlr.press/v37/ioffe15.html

Kingma, D. P., & Ba, J. 2017, Adam: A Method for Stochastic Optimization. https://arxiv.org/abs/1412.6980

Lemaître, G., Nogueira, F., & Aridas, C. K. 2017, Journal of Machine Learning Research, 18, 1

Rodriguez-Gomez, V., et al. 2017, Monthly Notices of the Royal Astronomical Society, 467, 4739

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. 2014, Journal of Machine Learning Research, 15, 1929. http://jmlr.org/papers/v15/srivastava14a.html

Stewart, K. R., Bullock, J. S., Barton, E. J., & Wechsler, R. H. 2009, The Astrophysical Journal, 702, 1005

Ćiprijanović, A., Snyder, G. F., Nord, B., & Peek, J. E. G. 2020, Astronomy and Computing, 32, 100390, doi: 10.1016/j.ascom.2020.100390