Using deep learning to explore ultra-large scale astronomical datasets

Michael J. Smith

Supervisor: James E. Geach

Second supervisor: Sugata Kaviraj

School of Physics, Engineering, and Computer Science
University of Hertfordshire

A thesis submitted in partial fulfilment
of the requirement of the degree of
Doctor of Philosophy

September 2022

Abstract

In every field that deep learning has infiltrated we have seen a reduction in the use of specialist knowledge, to be replaced with knowledge automatically derived from data. We have already seen this process play out in many ‘applied deep learning’ fields such as computer Go, protein folding, natural language processing, and computer vision. This thesis argues that astronomy is no different to these applied deep learning fields. To this end, this thesis’ introduction serves as a historical background on astronomy’s ‘three waves’ of increasingly automated connectionism: initial work on multilayer perceptrons within astronomy required manually selected emergent properties as input; the second wave coincided with the dissemination of convolutional neural networks and recurrent neural networks, models where the multilayer perceptron’s manually selected inputs are replaced with raw data ingestion; and in the current third wave we are seeing the removal of human supervision altogether with deep learning methods inferring labels and knowledge directly from the data.

§2, §3, and §4 of this thesis explore these waves through application. In §2 I show that a convolutional/recurrent encoder/decoder network is capable of emulating a complicated semi-manual galaxy processing pipeline. I find that this ‘Pix2Prof’ neural network can satisfactorily carry out this task over 100× faster than the method it emulates. §3 and §4 explore the application of deep generative models to astronomical simulation. §3 uses a generative adversarial network to generate mock deep field surveys, and finds it capable of generating mock images that are statistically indistinguishable from the real thing. Likewise, §4 demonstrates that a Diffusion model is capable of generating galaxy images that are both qualitatively and quantitatively indistinguishable from the training set. The main benefit of these deep learning based simulations is that they do not rely on a possibly flawed (or incomplete) physical knowledge of their subjects and observation processes. Also, once trained, they are capable of rapidly generating a very large amount of mock data.

§5 looks to the future and predicts that we will soon enter a fourth wave of astronomical connectionism. If astronomy follows in the footsteps of other applied deep learning fields we will see the removal of expertly crafted deep learning models, to be replaced with finetuned versions of an all-encompassing ‘foundation’ model. As part of this fourth wave I argue for a symbiosis between astronomy and connectionism. This symbiosis is predicated on astronomy’s relative data wealth, and contemporary deep learning’s enormous data appetite; many ultra-large datasets in machine learning are proprietary or of poor quality, and so astronomy as a whole could develop and provide a high quality multimodal public dataset. In turn, this dataset could be used to train an astronomical foundation model that can be used for state-of-the-art downstream tasks. Due to the foundation models’ hunger for data and compute, a single astronomical research group could not bring about such a model alone. Therefore, I conclude that astronomy as a whole has slim chance of keeping up with a research pace set by the Big Tech goliaths—that is, unless we follow the examples of EleutherAI and HuggingFace and pool our resources in a grassroots open source fashion.

Declarations

Copyright

Previous submissions

I declare that no part of this work is being submitted concurrently for another award of the University or any other awarding body or institution. This thesis contains a substantial body of work that has not previously been submitted successfully for an award of the University or any other awarding body or institution.

The following parts of this submission have been published previously and/or undertaken as part of a previous degree or research programme:

Chapter 2: M. J. Smith et al. (2021). 'Pix2Prof: fast extraction of sequential information from galaxy imagery via a deep natural language 'captioning' model'. In: *Monthly Notices of the Royal Astronomical Society* 503.1, pp. 96–105. DOI: 10.1093/mnras/stab424. arXiv: 2010.00622 [astro-ph.IM].

Chapter 3: M. J. Smith and J. E. Geach (2019). 'Generative deep fields: arbitrarily sized, random synthetic astronomical images through deep learning'. In: "*Monthly Notices of the Royal Astronomical Society*" 490.4, pp. 4985–4990. DOI: 10.1093/mnras/stz2886. arXiv: 1904.10286 [astro-ph.IM].

Chapter 4: M. J. Smith et al. (2022). 'Realistic galaxy image simulation via score-based generative models'. In: *Monthly Notices of the Royal Astronomical Society* 511.2, pp. 1808–1818. DOI: 10.1093/mnras/stac130. arXiv: 2111.01713 [astro-ph.IM].

Except where indicated otherwise in the submission, the submission is my own work and has not previously been submitted successfully for any award.

Acknowledgements

There are so many people that have helped me during this PhD. First and foremost I would like to say a huge thank you to Jim Geach, for being the best guide I could have asked for. I look up to Jim both as a researcher, and as a person, and I hope to one day become half the scientist he is. Thanks Jim for all of your guidance and support since our chance meeting in 2017; I am looking forward to working with you at Aspia, and beyond.

A big thanks must also go to Stéphane Courteau. Thanks for welcoming me with open arms into your research group, and for guiding a fledgling student through the ins and outs of academia. You taught me that science is a social endeavour, and I thank you for that. I remember one of our first meetings at Queen's where you warned me how quickly this PhD will fly by. It's passed by even faster than I expected!

I would like to thank John Mooney for introducing me to the world of 'shiny and new' deep learning way back during my MPhys at Leeds in 2016. Your enthusiasm and expertise for the subject got me hooked, and I wouldn't be where I am now without your initial guidance. I would also like to thank all the collaborators I have been lucky enough to accumulate over the years: Nikhil Arora, Tom Bewley, Will Cooper, Ryan Jackson, Niall Miller, Rachael Pirie, Ashley Spindler, and Connor Stone. I learnt so much from all of your unique perspectives, personalities, backgrounds, and ways of thinking, and I am very grateful for that.

Thanks to all my friends, colleagues, and officemates at Herts, Queen's, and the Turing for all the lunches, coffee (procrastination!) breaks, and impromptu pubbing. Alyssa, Ben L., Ben T., Calum, Dhruv, Jaime, John, Kasia, Luke, Maddie, Marina, Matt, Rob, Soumyadeep, Tracy, Vijay—cheers to all of you for making the last four years so much fun.

This thesis is dedicated to my family, and I would especially like to thank them for keeping me sane through two long years of pandemic. Without you all I don't think I would have finished. A big thank you to dad for always being there for advice and reassurance. You have done so much for me and I don't think a thousand paragraphs of acknowledgements would be enough to convey how much it means to me.

Contents

List of Figures

List of Tables

Chapter 1

Astronomical connectionism

> *With machine learning, we are finally able to keep up with the data deluge in astronomy. No longer will we be limited by our human ability to process and make sense of huge amounts of data. Now, we can let the machines do the work for us!*
>
> —GPT-3[1]

Astronomers often seek to compare simulated and empirical data. Both of these types of data are already available in abundance, and the total volume is set to increase exponentially in the coming years. Fig. 1.1 presents a selection of astronomical surveys and their estimated data volume output over their lifetimes (Zhang and Zhao 2015). From this graph we can see an exponential growth in survey volume.

The current scale of astronomical data volume already poses an issue for astronomy as many contemporary data extraction methods rely on human supervision and specialist expertise. The coming growth in data volume will make exploring and exploiting these surveys through traditional human supervised and semi-supervised means an intractable problem. Of most concern is the possibility that we will miss, or substantially delay, interesting and important discoveries due to our inability to accurately and consistently interrogate astronomical data at scale.

Convolutional Neural Networks (CNN; Fukushima 1980), Generative Adversarial Networks (GAN; Goodfellow et al. 2014), and other deep learning techniques (Hochreiter and Schmidhuber 1997; He et al. 2015; Ronneberger, Fischer, and Brox 2015; Vaswani et al. 2017) have recently shown great promise in automating information extraction in various data intensive fields. Therefore, deep learning is ideally poised as a solution to our problem of processing ultra-large scale astronomical data.

[1] This is a quote that was originally generated by prompting OpenAI's large language model 'Generative Pretrained Transformer 3' (Brown et al. 2020, `https://beta.openai.com/playground`). More generated 'quotes' can be found in §A.1.

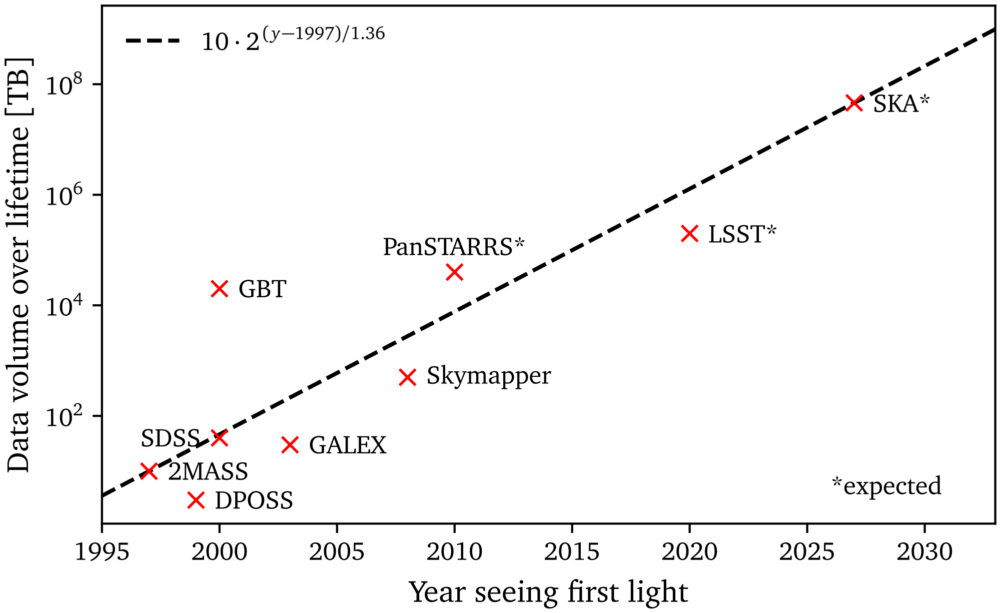


Figure 1.1: Data volume output of a selection of astronomical surveys over their lifetimes. Astronomical survey data volume doubles every 16 months. Data is taken from Zhang and Zhao (2015).

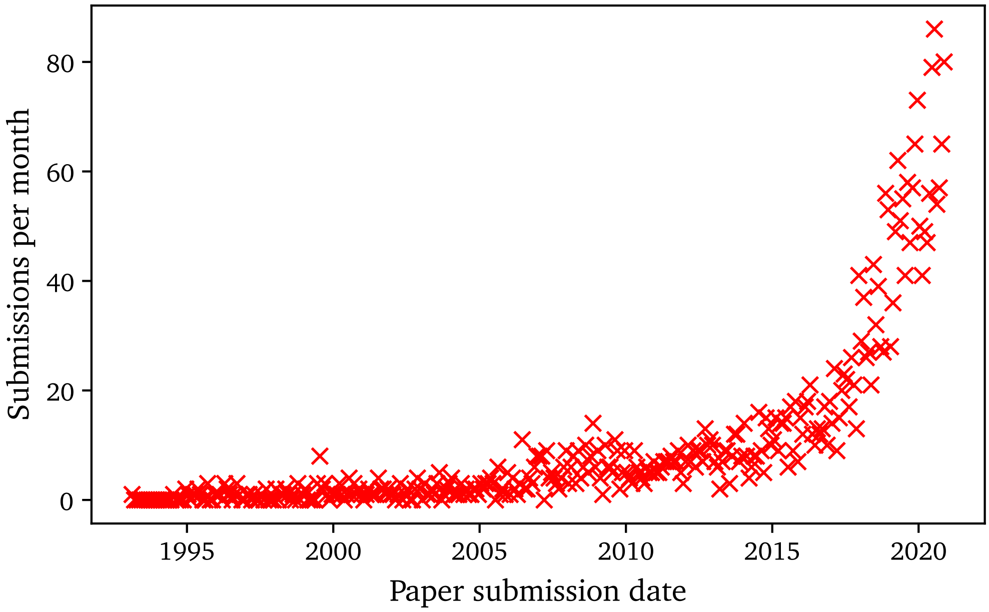


Figure 1.2: Here we see the number of arXiv:astro-ph submissions per month that have abstracts or titles containing one or more of the strings: 'machine learning', 'ML', 'artificial intelligence', 'AI', 'deep learning', or 'neural network'. The raw data is in the public domain and is available at `https://www.kaggle.com/Cornell-University/arxiv`.

As Fig. 1.2 demonstrates, there is already a rich, and rapidly expanding body of literature on the application of artificial neural networks and deep learning to problems in astronomy, and in this thesis I will explore deep learning from the perspective of an astronomer. This chapter will specifically review the history of connectionism, and explore the interconnection between astronomy and deep learning.

1.1 The perceptron

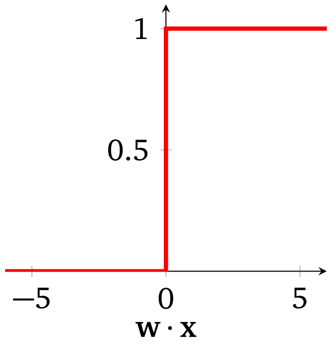
In 1943, McCulloch and Pitts proposed the first computational model of a biological neuron (MP neuron; McCulloch and Pitts 1943). Their model consisted of a set of binary inputs $x_i \in \{0,1\}$ and a single binary output $y \in \{0,1\}$. Their model also defines a single 'inhibitory' input $\mathscr{I} \in \{0,1\}$ that blocks output if $\mathscr{I} = 1$. If the sum of the inputs exceeds a threshold value Θ, the MP neuron 'fires' and outputs $y = 1$. Mathematically we can write the MP neuron function as

$$\mathrm{MP}(\mathbf{x}) = \begin{cases} 1 & \text{if } \sum_{i=0}^{n} x_i > \Theta \text{ and } \mathscr{I} = 0, \\ 0 & \text{otherwise.} \end{cases}$$

The MP neuron is quite a powerful abstraction; single MP neurons can calculate simple boolean functions, and more complicated functions can be calculated when many MP neurons are chained together.

The MP neuron is missing one crucial element—the capacity to learn. Rosenblatt (1958) addressed this, by combining the MP neuron with Hebb's neuronal wiring theory[2] (Hebb 1949). Like the MP neuron, Rosenblatt's perceptron takes a number of numeric inputs (x_i). However, unlike the MP neuron each one of these inputs are multiplied by a corresponding weight (w_i) signifying the importance the perceptron assigns to a given input. As shown in Fig. 1.3, Rosenblatt then sums this list of products and passes it into an 'activation function'. Rosenblatt used the Heaviside step function in his original formulation:

[2] Also known by the mantra 'cells that fire together wire together'.



$$\text{prediction} = H(\mathbf{w}\cdot\mathbf{x}) = \begin{cases} 0 & \text{if } \mathbf{w}\cdot\mathbf{x} < 0, \\ 1 & \text{if } \mathbf{w}\cdot\mathbf{x} \geq 1. \end{cases} \tag{1.1}$$

To concretise exactly how Rosenblatt's perceptron learns we will use an example. Let us say that we want to automatically label a set of galaxy images as either 'spiral' or 'elliptical'. To do this we first need to compile a training dataset of galaxy images. This training set would consist of spiral and elliptical galaxies, and each image would have a ground truth label y—say '0' for a spiral galaxy and '1' for an elliptical. To train our perceptron we randomly choose one image from the training set, and feed it to the perceptron, with the numerical value of each pixel corresponding to an input $\{x_1, \ldots, x_N\}$. These inputs are multiplied by their corresponding weight $\{w_1, \ldots, w_N\}$. Since we do not want our perceptron to have any prior knowledge of the task, we initialise the weights at random. The resulting products are then summed. Finally, our activation function H transforms $\mathbf{w}\cdot\mathbf{x}$ and produces a prediction p. We then compare p to y via a 'loss function', which is a function that measures the difference between p and y. The loss can be any differentiable function, so for illustration purposes we will define it here as the L1 loss: $\mathscr{L}(y,p) = \left|y-p\right|$. Now that we can compare to the ground truth, we need to work out how a change in one of our weights affects the loss (that is, we want to find $\partial\mathscr{L}/\partial\mathbf{w}$). We can calculate this change with the chain rule

$$\frac{\partial\mathscr{L}}{\partial\mathbf{w}} = \frac{\partial\mathscr{L}}{\partial p}\frac{\partial p}{\partial\mathbf{w}}, \tag{1.2}$$

and since $p = H(\mathbf{w}\cdot\mathbf{x})$ and $\partial p/\partial\mathbf{w} = H'\mathbf{x}^T$ we get

$$\frac{\partial\mathscr{L}}{\partial\mathbf{w}} = \frac{\partial\mathscr{L}}{\partial p} \odot (H'\mathbf{x}^T)$$

where $\odot$ is the distributive Hadamard product. Thus we can update the weights to decrease the loss function:

$$\begin{aligned} \mathbf{w}_{\text{next}} &= \mathbf{w} - \eta\frac{\partial\mathscr{L}}{\partial\mathbf{w}} \\ &= \mathbf{w} - \eta\frac{\partial\mathscr{L}}{\partial p} \odot (H'\mathbf{x}^T), \end{aligned}$$

where η is the learning rate[3]. If we repeat this process our perceptron will get better and better at classifying our galaxies!

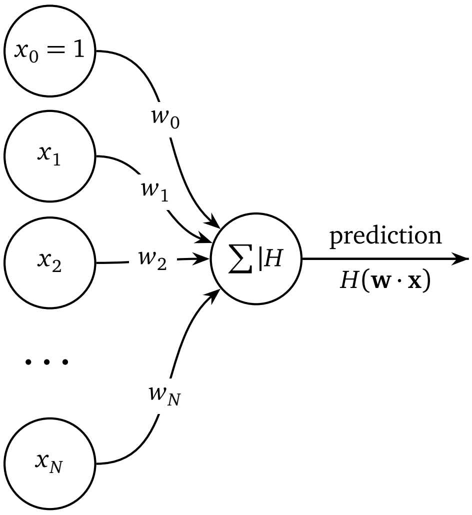


Figure 1.3: A single neuron (or perception) with a bias w_0, inputs $x_1, x_2, \dots, x_N$, and weights $w_1, w_2, \dots, w_N$.

We must go further than training a single layer perceptron; in *Perceptrons: An Introduction to Computational Geometry*, Minsky and Papert (e.g. §13.0; 1969) show that the single layer perceptron is only able to calculate linearly separable functions, among other limitations. Their book (alongside a consensus that AI had failed to deliver on its early grandiose promises) delivered a big blow to the connectionist school of artificial intelligence[4]. In the years following Minsky and Papert (1969) governmental and industry funding was pulled from connectionist research laboratories, ushering in the first 'AI winter'[5].

Yet, as exemplified in Rosenblatt (§5.2, theorem 1; 1962) it was known at the time that multilayer perceptrons could calculate non-linearly separable functions (such as the 'exclusive or'). We can prove intuitively that a set of neurons can calculate *any* function: a perceptron can perfectly emulate a `NAND` gate (Fig. 1.4), and the singleton set {`NAND`} is functionally complete. Since we can combine a set of `NAND` gates to calculate any function, *we must also be able to combine a set of neurons to calculate any function*[6]. Such a group of neurons is known as the multilayer perceptron (MLP). Unfortunately, we cannot simply stack perceptrons together as we are missing one

[3] The eagle eyed reader may have noticed that since the derivative of the Heaviside step function is the Dirac delta function, we will only update the perceptron's weights on an incorrect prediction. If we want to also learn from positive examples, we need to use a smoothly differentiable activation function. This is explored in the next section.

[4] See Metz (2021) for a closer look at the conflicts and personalities that shaped AI.

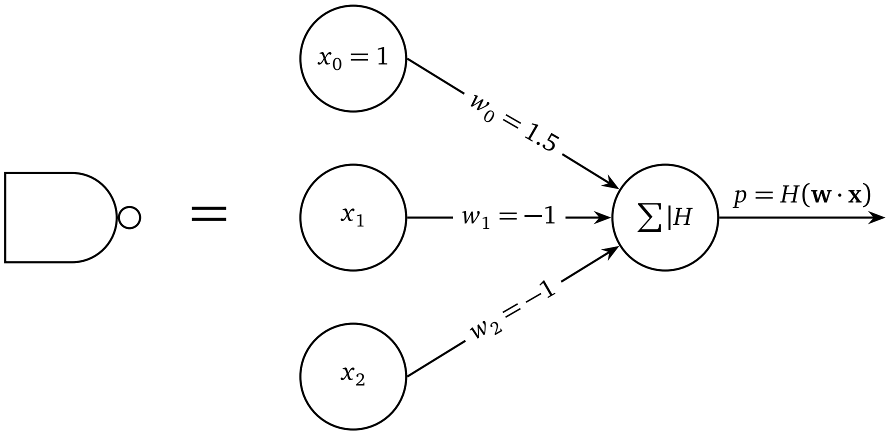
[5] At least, in the Western world. Connectionism continued in earnest in the Soviet Union (Ivakhnenko and Lapa 1965; Ivakhnenko 1971).

[6] More formally, Cybenko (1989) and Hornik, Tinchcombe, and White (1991) prove that an infinitely wide neural network can calculate any function, and Lu et al. (2017) prove that an infinitely deep neural network is a universal approximator.

vital ingredient; a way to train the network! At the time of Minsky and Papert's treatise on perceptrons there was no widely known algorithm (in the West; see Ivakhnenko and Lapa 1965) that could train such a multilayer network. In Minsky and Papert's own words:

> Nevertheless, we consider it to be an important research problem to elucidate (or reject) our intuitive judgment that the extension [from one layer to many] is sterile. Perhaps some powerful convergence theorem will be discovered, or some profound reason for the failure to produce an interesting 'learning theorem' for the multilayered machine will be found.
> (§13.2; Minsky and Papert 1969, on MLPs)

The field had to wait almost two decades for such an algorithm to become widespread. In the next section we will explore backpropagation, the algorithm that ultimately proved Minsky and Papert's intuition wrong.



x_1	x_2	$\neg(x_1 \wedge x_2)$	$p = H(\mathbf{w} \cdot \mathbf{x})$
0	0	1	$H(1.5 + (-1) \cdot 0 + (-1) \cdot 0) = 1$
0	1	1	$H(1.5 + (-1) \cdot 0 + (-1) \cdot 1) = 1$
1	0	1	$H(1.5 + (-1) \cdot 1 + (-1) \cdot 0) = 1$
1	1	0	$H(1.5 + (-1) \cdot 1 + (-1) \cdot 1) = 0$

Figure 1.4: If we define $H(\mathbf{w} \cdot \mathbf{x})$ as in Eq. 1.1 we can set a perceptron's weights so that it is equivalent to the `NAND` gate.

1.2 The multilayer perceptron

Grouping many artificial neurons together may result in something resembling Fig. 1.5. This network consists of an input layer, two intermediate 'hidden' layers, and an output layer. As in the previous section, let us say that we want a classifier that can classify a set of galaxy images into elliptical and spiral types. In an MLP similar to Fig. 1.5 a neuron would be assigned to each pixel in a galaxy image. Each neuron would take the numeric value of that pixel, and propagate that signal forward into the network. The next layer of neurons does the same, with the input being the previous layer's output. This process continues until we reach the output layer. In a binary classification task like our galaxy classifier this layer outputs a value between zero and one. Thus, if we define a spiral galaxy as one, and an elliptical galaxy as zero, we would want the network output to be near one for a spiral galaxy input (and vice versa).

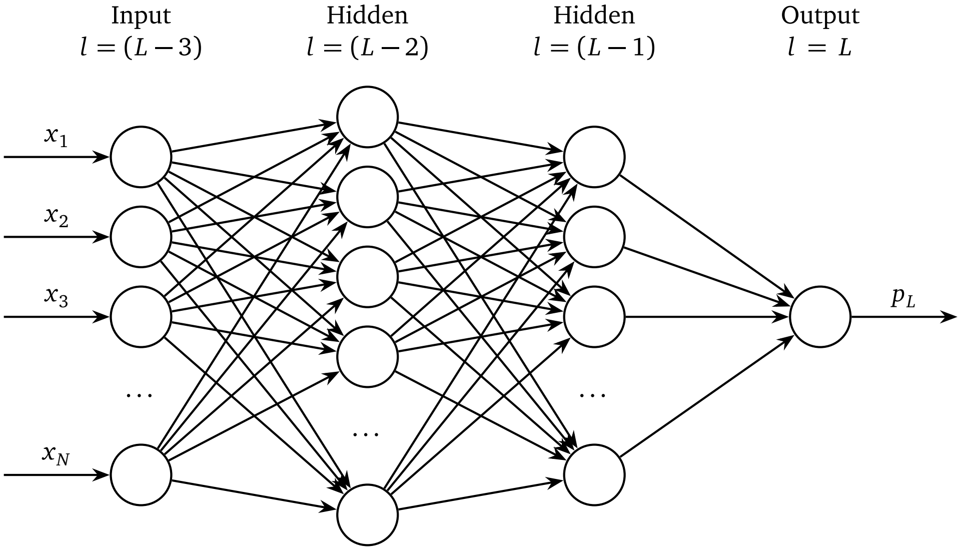


Figure 1.5: The multilayer perceptron, or artificial neural network. The depicted network has two hidden layers. It takes N inputs $x_1, x_2, \ldots, x_N$, and outputs a prediction p_L.

In §1.1 we found the change we needed to apply to a single neuron's weights to make it learn from a training example. We can train an MLP in a similar way by employing the reverse mode of automatic differentiation (or backpropagation) to learn from our galaxy training data set (Linnainmaa 1976; Werbos 1981; Rumelhart, Hinton, and Williams 1986b)[7]. We want our network to learn when it makes both a

[7] Some controversy surrounds backpropagation's discovery. The Finnish computer scientist Linnainmaa proposed the reverse mode of automatic differentiation and adapted the algorithm to run on computers in their 1970 (Finnish language) thesis (Linnainmaa 1970). They first published their findings in English in 1976. Werbos then proposed applying an adaptation of Linnainmaa's method to artificial neural networks. Rumelhart, Hinton, and Williams (1986b) showed experimentally that backpropagation can generate meaningful internal representations within a neural network, and popularised the method. Here I will err on the side of caution and cite all three manuscripts. For further reading I recommend Schmidhuber (2014) and Baydin et al. (2018).

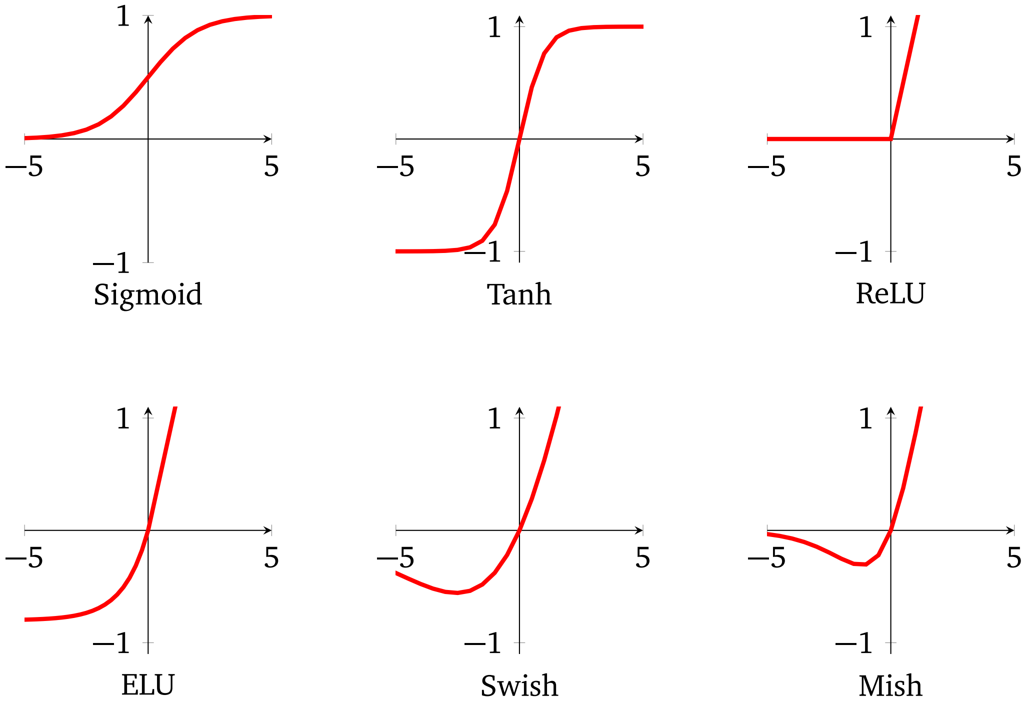


Figure 1.6: A curated selection of activation functions. In all plots, the x axis is the input, and the y axis is the output. The Rectified Linear Unit activation function was first introduced in the context of neural networks in Fukushima (1980) and later rediscovered, named, and popularised in Nair and Hinton (2010). The Exponential Linear Unit, Swish and Mish activations were respectively introduced in Clevert, Unterthiner, and Hochreiter (2016), Ramachandran, Zoph, and Le (2017), and Misra (2019).

correct and incorrect prediction, so we define our activation function as a smoothed version of Rosenblatt's perceptron activation. This ensures that a signal is present in the derivative no matter which values are input. This activation function is known as the 'sigmoid' function, and is shown in Fig. 1.6. As in §1.1 we define a loss function $\mathscr{L}(y,p)$ that describes the similarity between a ground truth (y) and a prediction (p). We also define a neuron's activation function as $\varphi(\mathbf{w}\cdot\mathbf{x})$ where $\mathbf{w}\cdot\mathbf{x}$ is the weighted sum of a neuron's inputs. Following from Eq. 1.2:

$$\frac{\partial\mathscr{L}}{\partial\mathbf{w}_l}=\frac{\partial\mathscr{L}}{\partial\mathbf{p}_l}\frac{\partial\mathbf{p}_l}{\partial\mathbf{w}_l}$$

where l is a layer in the MLP. In the same way as in §1.1 we can calculate an MLP's final layer's $(l=L)$ weight updates in terms of known values:

$$\frac{\partial\mathscr{L}}{\partial\mathbf{w}_L}=\frac{\partial\mathscr{L}}{\partial p_L}\odot\left(\varphi'_L\mathbf{p}^T_{L-1}\right), \tag{1.3}$$

where $\mathbf{p}_{L-1}$ are the outputs from the previous layer. To calculate the $(L-1)$th layer's weight updates we use the chain rule:

$$\frac{\partial\mathscr{L}}{\partial\mathbf{w}_{L-1}}=\frac{\partial\mathscr{L}}{\partial p_L}\frac{\partial p_L}{\partial\mathbf{p}_{L-1}}\frac{\partial\mathbf{p}_{L-1}}{\partial\mathbf{w}_{L-1}}.$$

Likewise for the $(L-n)$th layer:

$$\frac{\partial\mathscr{L}}{\partial\mathbf{w}_{L-n}}=\frac{\partial\mathscr{L}}{\partial p_L}\left(\prod_{i=1}^{n}\frac{\partial\mathbf{p}_{L+1-i}}{\partial\mathbf{p}_{L-i}}\right)\frac{\partial\mathbf{p}_{L-n}}{\partial\mathbf{w}_{L-n}}.$$

Now we can start plugging in some known values. Since $\mathbf{p}_l=\varphi_l(\mathbf{w}_l\cdot\mathbf{p}_{l-1})$, it follows that $\partial\mathbf{p}_l/\partial\mathbf{p}_{l-1}=\varphi'_l\mathbf{w}^T_l$, and $\partial\mathbf{p}_l/\partial\mathbf{w}_l=\varphi'_l\mathbf{p}^T_{l-1}$. So:

$$\frac{\partial\mathscr{L}}{\partial\mathbf{w}_{L-n}}=\frac{\partial\mathscr{L}}{\partial p_L}\odot\left(\prod_{i=1}^{n}\varphi'_{L-i}\mathbf{w}^T_{L-i}\right)\left(\varphi'_{L-n}\mathbf{p}^T_{L-n-1}\right). \tag{1.4}$$

Combining Eq. 1.3 with Eq. 1.4 we get the weight update algorithm for the $(L-n)$th layer of the MLP:

$$\mathbf{w}_{\text{next}}=\mathbf{w}-\eta\begin{cases}\frac{\partial\mathscr{L}}{\partial p_L}\odot\left(\varphi'_L\mathbf{p}^T_{L-1}\right), & \text{for } n=0,\\ \frac{\partial\mathscr{L}}{\partial p_L}\odot\left(\prod_{i=1}^{n}\varphi'_{L-i}\mathbf{w}^T_{L-i}\right)\left(\varphi'_{L-n}\mathbf{p}^T_{L-n-1}\right), & \text{for } n>0.\end{cases} \tag{1.5}$$

With this equation[8] in hand we can use the same technique described earlier in this section and in §1.1 to update the network's weights with each galaxy image to decrease the loss function $\mathscr{L}$. Again, as $\mathscr{L}$ is minimised, our MLP will classify our elliptical and spiral galaxy images with better and better accuracy.

1.3 The first wave of astronomical connectionism

Connectionism was first discussed within astronomy in the late 1980s, after the popularisation of backpropagation (see footnote 7) and the consequent passing of the first 'AI winter'. Two radical studies emerged in 1988 that recognised areas where astronomy could benefit from the use of artificial neural networks (Adorf and Johnston 1988; Rappaport and Anderson 1988). Together, they identified that astronomical object classification[9] and telescope scheduling could be solved through the use of an artificial neural network. These studies were followed by a rapid broadening of the field, and the application of connectionism to many disparate astronomical use cases (Miller 1993, and references therein). In this section, I will outline areas where MLPs found an early use in astronomy.

1.3.1 Classification problems

Odewahn et al. (1992) classified astronomical objects into star and galaxy types. These objects were taken from the Palomar Sky Survey Automated Plate Scanner catalogue (Pennington et al. 1993). To compile their dataset, they first extracted a set of emergent image parameters from the scanned observations. These parameters included the diameter, ellipticity, area, and plate transmission. The parameters were then used to train both a linear perceptron and a feedforward MLP to classify the astronomical objects into stars or galaxies. Odewahn et al. (1992) found that their best performing model could classify galaxies with a completeness of 95% for objects with a magnitude < 19.5. This work was followed by many more studies on the star/galaxy classification problem (e.g. Odewahn et al. 1993; Bertin and Arnouts 1996; Andreon et al. 2000).

Galaxy morphological type classification was explored in the early 1990s. Storrie-Lombardi et al. (1992) describe an MLP that takes an input a selected set of thirteen galaxy summary statistics, and uses this information to classify the galaxy into one of

[8] If we examine Eq. 1.5 carefully, we can see why we add nonlinearities between the MLP layers; without activation functions Eq. 1.5 collapses to the equivilent of a single layer MLP!

[9] Specifically, galaxies were discussed in Rappaport and Anderson (1988) and point sources observed with the Infra-Red Astronomical Satellite (IRAS) were discussed in Adorf and Johnston (1988).

five morphological types. Storrie-Lombardi et al. (1992) report a top one accuracy of 64%, and a top two accuracy of 90%. This pilot study was followed by several studies from the same group that confirmed that MLPs are effective galaxy morphological classifiers (Lahav et al. 1995; Naim et al. 1995a,b, see §1.8 for a continuation of this line of research).

MLPs were also used in other classification tasks; here I select for discussion a few further areas where MLPs were applied. Hippel et al. (1994) classified stellar spectra into temperature types. Chon (1998) described the use of an MLP to search for and classify Muon events (and therefore neutrino observations) in the Sudbury Neutrino Observatory. Quasar classification has been explored in several studies (Carballo, Cofiño, and González-Serrano 2004; Claeskens et al. 2006; Carballo et al. 2008). Seminally, Carballo, Cofiño, and González-Serrano (2004) used an MLP to select quasar candidates, given their radio flux, integrated-to-peak flux ratio, photometry and point spread function in the red and blue bands, and their radio-optical position separation. They found good agreement between their model and that of the decision tree described in White et al. (2000), confirming MLPs as a competitive alternative to more traditional machine learning. As part of the Supernova photometric Classification Challenge (SPCC; Kessler et al. 2010), Karpenka, Feroz, and Hobson (2013) proposed the use of a neural network to classify supernovae into Type-1a/non-Type-1a classes. To classify their light curves, they first used a hand-crafted fitting function, and then trained their MLP on the fitted coefficients. They found that their model was competitive with other, more complex models trained on the SPCC dataset.

1.3.2 Regression problems

MLPs have also been used in regression problems. Angel et al. (1990) applied them first to adaptive telescope optics. They trained their MLP on 250 000 simulated in focus and out of focus observations of stars as seen by the multiple mirror telescope (MMT). From the flattened 13 × 13 pixel observations, their network predicted the piston position and tilt required for each of the MMT's mirrors to bring the stars into focus. After the application of these corrections, the authors were able to recover the original profile. In follow up studies, Sandler et al. (1991) and Lloyd-Hart et al. (1992) proved that Angel et al.'s MLP works on the real MMT.

Photometric redshift estimation was been explored in many concurrent studies (e.g. Firth, Lahav, and Somerville 2003; Tagliaferri et al. 2003; Ball et al. 2004; Collister and Lahav 2004; Vanzella et al. 2004). Firth, Lahav, and Somerville (2003) train a neural network to predict the redshift of galaxies contained in the Sloan Digital

Sky Survey (SDSS) early data release (Stoughton et al. 2002). The galaxies were input to the neural network as a set of summary parameters, and the output was a single float correlating with the galaxy redshift. They found their network attained a performance comparable to classical techniques. Extending and confirming the work by Firth, Lahav, and Somerville (2003), Ball et al. (2004) used an MLP to predict the redshift of galaxies contained in the SDSS's first data release (York et al. 2000). They also showed that MLPs were capable of predicting the galaxies' spectral types and morphological classifications.

Of course, MLPs have been used more widely in astronomical regression tasks. Here I will cherry pick a few studies to show the MLP's early breadth of use. Sunspot maxima prediction was carried out by Koons and Gorney (1990). They found their MLP based method was capable of predicting the number of sunspots when trained on previous cycles. Bailer-Jones et al. (1997) predicted the effective temperature of a star from its spectrum. Auld et al. (2007) and Auld, Bridges, and Hobson (2008) applied MLPs to cosmology, demonstrating that MLPs are capable of computing the cosmic microwave background, given a set of cosmological parameters. Nørgaard-Nielsen and Jørgensen (2008) used an MLP to remove the foreground from microwave temperature maps.

1.3.3 Scaling and expert reliance

There are some issues with MLPs. Primarily they do not scale well to high dimensional datasets. For example, if our dataset consists of images with a 128×128 resolution, we will need 16 384 neurons in the MLP's input layer alone! As we move into the hidden layers, this scaling issue only gets worse. Also, since MLPs must take an unrolled image as an input, they disregard any spatial properties of their training images, and so either need a substantial amount of training data to classify or generate large images[10], or an expert to extract descriptive features from the data in a preprocessing step. We can see this issue writ large in the subsections above—most of the MLP applications described in §1.3 require an expert to extract features from the data for the network to then train on! This drawback is not ideal; what if there are features within the raw data that are not present in these cherry picked statistics? In that case, it would be preferable to let the neural network take in the raw data as input,

[10] At the height of the convolutional neural network architecture's popularity in the mid 2010s these were real problems. However, with the growth of computing power and data in recent years we are seeing a resurgence of the more general MLP model (e.g. H. Liu et al. 2021; Melas-Kyriazi 2021; Tolstikhin et al. 2021; Touvron et al. 2021). This follows the prevailing trend in AI where the removal of human-crafted features and biases ultimately results in more expressive models that learn such features and biases directly from data (Sutton 2019; Branwen 2022).

and then learn which features are the most descriptive. I will discuss neural network architectures that solve both the MLP scaling problem and the expert reliance problem in §1.4–§1.8.

1.4 Convolutional neural networks

Unlike the MLP described in the previous section, convolutional neural networks (CNNs; introduced in Fukushima (1980) and first combined with backpropagation in LeCun et al. (1989)) do not entirely consist of fully connected layers, where every neuron is connected to every neuron in the previous and subsequent layers. Instead, the CNN (such as the one depicted in Fig. 1.7) uses convolutional layers in place of the majority (or all) of the dense layers.

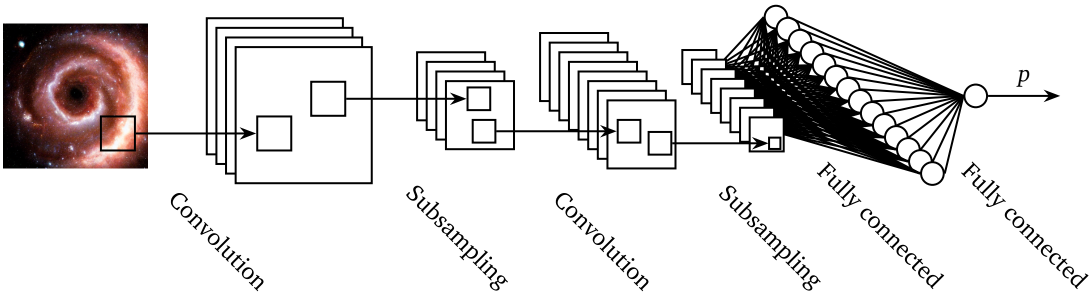


Figure 1.7: A convolutional neural network classifying a spiral galaxy image[11].

We can think of a convolutional layer as a set of learnt 'feature filters'. These feature filters perform a local transform on input imagery. In classical computer vision, these filters are hand crafted, and perform a predetermined function, such as edge detection or blurring. In contrast, a CNN learns the optimal set of filters for its task (say, galaxy classification). Eq. 1.6 shows two different convolution[12] operators

[11] All astronomical objects shown in the neural network diagrams within §1 and §5 are generated via text prompts fed into a latent diffusion neural network model (Rombach et al. 2021).

[12] We must note that in Eq. 1.6 we follow most deep learning libraries and perform a cross-correlation and **not** a convolution. However, since the weights are learnt, this does not matter; the neural network will simply learn a flipped representation of the cross-correlation.

being performed on an array.

$$\begin{bmatrix} 39 & 57 & 86 & 9 & 26 \\ 90 & 74 & 63 & 87 & 98 \\ 79 & 34 & 26 & 16 & 46 \\ 67 & 61 & 96 & 1 & 79 \\ 33 & 47 & 15 & 49 & 29 \end{bmatrix} \star \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 26 & 16 & 46 \\ 96 & 1 & 79 \\ 15 & 49 & 29 \end{bmatrix}$$

$$\begin{bmatrix} 39 & 57 & 86 & 9 & 26 \\ 90 & 74 & 63 & 87 & 98 \\ 79 & 34 & 26 & 16 & 46 \\ 67 & 61 & 96 & 1 & 79 \\ 33 & 47 & 15 & 49 & 29 \end{bmatrix} \star \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 139 & 136 & 219 \\ 220 & 101 & 158 \\ 155 & 179 & 56 \end{bmatrix} \tag{1.6}$$

In the above equation the operation is represented as a matrix. In a CNN the matrix is a set of neuronal weights. As seen in Fig. 1.7 there are multiple feature maps in a convolutional layer, each containing a set of weights independent to the other feature maps, and learning to extract a different feature. As in the MLP described in the previous section, the weights are updated using backpropagation to minimise a loss function. I will discuss astronomical applications of CNNs in §1.8, after I introduce modern CNN architectures.

1.5 Recurrent neural networks

Standard feedforward neural networks like the MLP (§1.2) and CNN (§1.4) generate a fixed size vector given a fixed size input[13]. But, what if we want to classify or generate a variably sized vector? For example, we might want to classify a galaxy's morphology given its rotation curve. A rotation curve describes the velocity of a galaxy's visible stars versus their distance from the galaxy's centre. Fig. 1.8 shows a possible rotation curve for Messier 81. A rotation curve's length depends on the size of its galaxy, and due to this variable length, and the fact that MLPs take a fixed size input, we cannot easily use an MLP for classification. Recurrent neural networks (RNNs), however, can take a variable length input and produce a variable length output.

A RNN differs from a feed forward MLP by having a hidden state that acts as a

[13] As with any rule there are exceptions, such as CNNs containing a global average pooling layer (Lin, Chen, and Yan 2013).

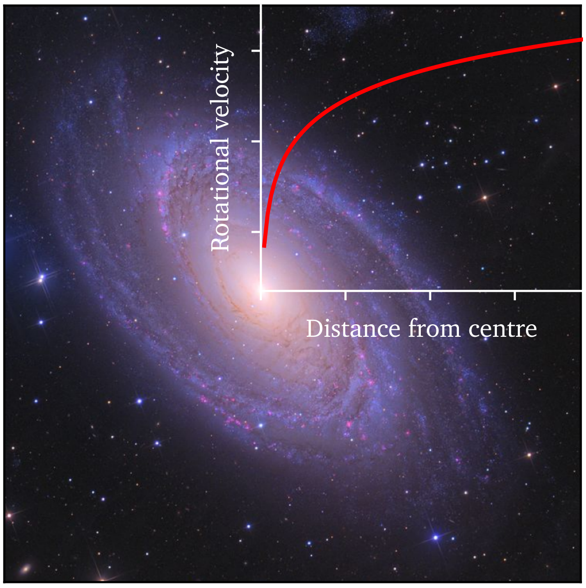


Figure 1.8: An example of a galaxy rotation curve, plotted over an image of Messier 81 (Crawford 2015).

'memory' store of previously seen information. As the RNN encounters new data, its weights are altered through the backpropagation through time algorithm (BPTT; Werbos 1990, and references therein. Also see footnote 7).

We can use an RNN similar to Fig. 1.9 to classify our rotation curves. We take the rotation curve in list form $\{x_1, x_2, \ldots, x_N\}$, with each x being a measurement of the rotational velocity at a certain radius. Then we feed this list into the RNN sequentially in the same way as shown in Fig. 1.9. The RNN will produce an output for each x fed to it, but we ignore those until we feed in x_N, the rotational velocity furthest from our galaxy's centre. When we feed in x_N, the RNN produces a prediction p_N, which we can then compare to a ground truth y_N via a loss function $\mathscr{L}_N$. In our case, y is an integer label representing the galaxy's morphological class. This comparison $\mathscr{L}_N(y_N, p_N)$ is a function that represents the distance between the RNN prediction and the ground truth. We can then reduce $\mathscr{L}_N(y_N, p_N)$ by updating the RNN's weights through BPTT so that the weights $\{\mathbf{w}_x, \mathbf{w}_p, \mathbf{w}_h\}$ follow $\nabla \mathscr{L}_N$ downwards. As we do this, our RNN will improve its galaxy classifications.

BPTT's mathematical derivation is akin to the one we explored in §1.2, and I will quickly derive it here for posterity. Let us first look at the forward propagation equations:

$$\begin{aligned}
\mathscr{L}_n &= \left| y_n - p_n \right|, \\
p_n &= \varphi(\mathbf{w}_p \cdot \mathbf{h}_n), \text{ and} \\
\mathbf{h}_n &= \phi(\mathbf{w}_h \cdot \mathbf{h}_{n-1} + \mathbf{w}_x \cdot \mathbf{x}_n).
\end{aligned}$$

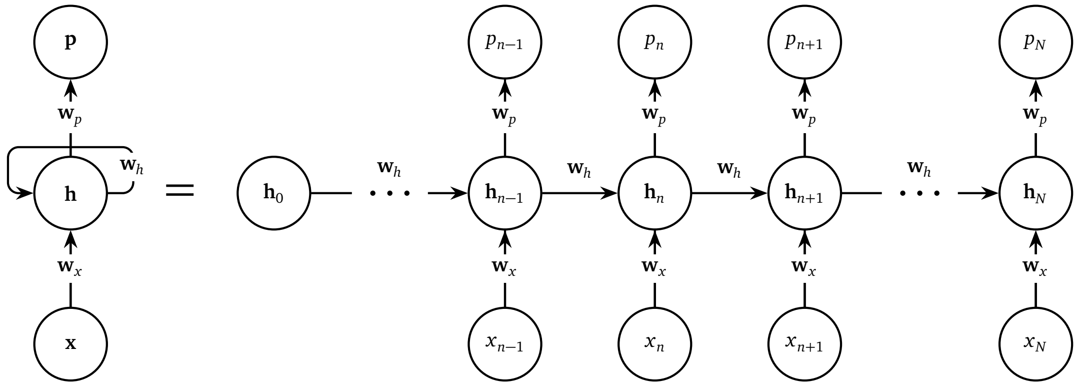


Figure 1.9: A recurrent neural network with weights $\{\mathbf{w}_x, \mathbf{w}_p, \mathbf{w}_h\}$, a hidden state $\mathbf{h}_n$, inputs $\mathbf{x}$, and a prediction $p_{n=N}$ is unrolled into its constituent processes.

From these we see that we need to express $\partial \mathscr{L}_n/\partial \mathbf{w}_p$, $\partial \mathscr{L}_n/\partial \mathbf{w}_h$, and $\partial \mathscr{L}_n/\partial \mathbf{w}_x$ as known values to train the network. $\partial \mathscr{L}_n/\partial \mathbf{w}_p$ is relatively easy; via the chain rule, and the fact that $\partial p_n/\partial \mathbf{w}_p = \varphi' \mathbf{h}_n^T$:

$$\begin{aligned}\frac{\partial \mathscr{L}_n}{\partial \mathbf{w}_p} &= \frac{\partial \mathscr{L}_n}{\partial p_n}\frac{\partial p_n}{\partial \mathbf{w}_p},\\ &= \frac{\partial \mathscr{L}_n}{\partial p_n} \odot \varphi' \mathbf{h}_n^T. \end{aligned} \tag{1.7}$$

$\partial \mathscr{L}_n/\partial \mathbf{w}_h$ is more tricky, so we will go step by step. We already know that

$$\frac{\partial \mathscr{L}_n}{\partial \mathbf{w}_h} = \frac{\partial \mathscr{L}_n}{\partial p_n}\frac{\partial p_n}{\partial \mathbf{h}_n}\frac{\partial \mathbf{h}_n}{\partial \mathbf{w}_h}. \tag{1.8}$$

However, we see in Fig. 1.9 that $\mathbf{h}_n$ depends on $\mathbf{h}_{n-1}$, which depends on $\mathbf{h}_{n-2}$ (and so on). We also notice that all the hidden states depend on $\mathbf{w}_h$. We therefore rewrite Eq. 1.8 to make these facts explicit:

$$\begin{aligned}\frac{\partial \mathscr{L}_n}{\partial \mathbf{w}_h} &= \frac{\partial \mathscr{L}_n}{\partial p_n}\frac{\partial p_n}{\partial \mathbf{h}_n}\sum_{j=1}^{n}\frac{\partial \mathbf{h}_n}{\partial \mathbf{h}_j}\frac{\partial \mathbf{h}_j}{\partial \mathbf{w}_h},\\ &= \frac{\partial \mathscr{L}_n}{\partial p_n}\frac{\partial p_n}{\partial \mathbf{h}_n}\sum_{j=1}^{n}\left(\prod_{i=j+1}^{n}\frac{\partial \mathbf{h}_i}{\partial \mathbf{h}_{i-1}}\right)\frac{\partial \mathbf{h}_j}{\partial \mathbf{w}_h}.\end{aligned}$$

We can now substitute in some known values:

$$\frac{\partial \mathscr{L}_n}{\partial \mathbf{w}_h} = \frac{\partial \mathscr{L}_n}{\partial p_n} \odot \varphi' \mathbf{h}_n^T \sum_{j=1}^{n}\left(\prod_{i=j+1}^{n} \phi' \mathbf{w}_{h,i}^T\right)\phi' \mathbf{h}_{j-1}^T. \tag{1.9}$$

Finally, $\partial \mathscr{L}_n / \partial \mathbf{w}_x$ is derived in the same way as $\partial \mathscr{L}_n / \partial \mathbf{w}_h$:

$$\begin{aligned}\frac{\partial \mathscr{L}_n}{\partial \mathbf{w}_x} &= \frac{\partial \mathscr{L}_n}{\partial p_n}\frac{\partial p_n}{\partial \mathbf{h}_n}\sum_{j=1}^{n}\left(\prod_{i=j+1}^{n}\frac{\partial \mathbf{h}_i}{\partial \mathbf{h}_{i-1}}\right)\frac{\partial \mathbf{h}_j}{\partial \mathbf{w}_x},\\ &= \frac{\partial \mathscr{L}_n}{\partial p_n} \odot \varphi' \mathbf{h}_n^T \sum_{j=1}^{n}\left(\prod_{i=j+1}^{n} \phi' \mathbf{w}_{h,i}^T\right)\phi' \mathbf{x}_j^T. \end{aligned} \tag{1.10}$$

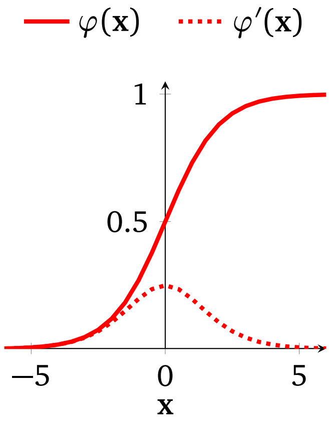
With $\partial \mathscr{L}_n / \partial \mathbf{w}_p$, $\partial \mathscr{L}_n / \partial \mathbf{w}_h$, and $\partial \mathscr{L}_n / \partial \mathbf{w}_x$ in hand we can apply the same update rule shown in Eq. 1.5.

Aside from many-to-one encoding, RNNs can produce many predictions given many inputs, act similarly to an MLP and produce one or many outputs given a single input. I will discuss the application of recurrent neural networks to astronomical data in §1.8, after I introduce gated recurrent neural networks.

1.6 Sidestepping the vanishing gradient problem

In the early 1990s, researchers identified a major issue with the training of deep neural networks through backpropagation. Hochreiter first formally examined the 'vanishing gradient' problem in their diploma thesis (Hochreiter (1991), see also later work by Bengio, Simard, and Frasconi (1994)). Due to the vanishing gradient problem, it was widely believed that training very deep artificial neural networks from scratch via backpropagation was impossible. In this section we will explore what the vanishing gradient problem is, and how contemporary end-to-end trained neural networks sidestep this issue.

First let us remind ourselves of the sigmoid activation function introduced in Fig. 1.6:



$$\varphi(\mathbf{x}) = 1/(1 - e^{-\mathbf{x}}). \tag{1.11}$$

Eq. 1.11 and its accompanying plot shows the output of a sigmoid function φ and its derivative φ', when given an input $\mathbf{x}$.

Now, let us revisit the weight update rule for the $(L-n)$th layer of a feedforward MLP (Eq. 1.4):

$$\frac{\partial \mathscr{L}}{\partial \mathbf{w}_{L-n}} = \frac{\partial \mathscr{L}}{\partial p_L} \odot \underbrace{\left(\prod_{i=1}^{n} \varphi'_{L-i}\mathbf{w}^T_{L-i}\right)}_{\lim_{n\to\infty}\prod_{i=1}^{n}\varphi'_{L-i}\mathbf{w}^T_{L-i}=0} \left(\varphi'_{L-n}\mathbf{p}^T_{L-n-1}\right). \tag{1.12}$$

If φ' is typically less than one (as in Eq. 1.11 and most other saturating nonlinearities) the product term in the above equation becomes an issue. In that case, we can see that the product rapidly goes to zero as n (the number of layers) becomes large[14]. If we study Eq. 1.9, we can see the same problem also plagues RNNs as we backpropagate through hidden states:

$$\frac{\partial \mathscr{L}_n}{\partial \mathbf{w}_h} = \frac{\partial \mathscr{L}_n}{\partial p_n} \odot \varphi'\mathbf{h}^T_n \sum_{j=1}^{n} \underbrace{\left(\prod_{i=j+1}^{n} \phi'\mathbf{w}^T_{h,i}\right)}_{\lim_{n\to\infty}\prod_{i=j+1}^{n}\phi'\mathbf{w}^T_{h,i}=0} \phi'\mathbf{h}^T_{j-1}. \tag{1.13}$$

Let us solidify this issue by reminding ourselves about Eq. 1.5—the weight update rule for a network trained through backpropagation:

$$\mathbf{w}_{\text{next}} = \mathbf{w} - \eta \frac{\partial \mathscr{L}}{\partial \mathbf{w}}. \tag{1.14}$$

Combining Eq. 1.14 and the limits defined in Eq. 1.12 and Eq. 1.13 results in the below weight update rule in the limit $n \to \infty$.

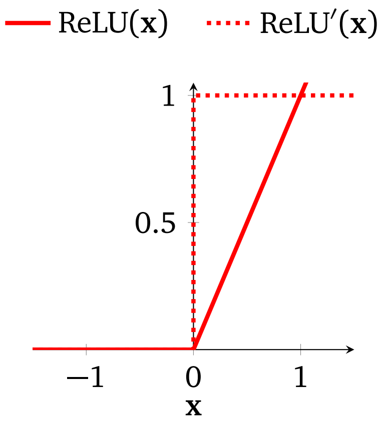
$$\lim_{n\to\infty} \mathbf{w}_{\text{next}} = \mathbf{w}. \tag{1.15}$$

Eq. 1.15 shows that learning via backpropagation slows as we move deeper into the network. This problem once again caused a loss of faith in the connectionist model, ushering in the second AI winter. It took until 2012 for a new boom to begin. In the following three subsections I will explore some of the proposed partial solutions to the vanishing gradient problem and show how they came together to contribute to the current deep learning boom.

[14] Likewise, if φ' is typically greater than one, the product term rapidly 'explodes' to infinity. This is known as the 'exploding gradient' problem, also first identified in Hochreiter (1991).

1.6.1 Non-saturating activation functions

We can see in Eq. 1.13 and Eq. 1.12 that if $\varphi' = 1$ then the product term does not automatically go to zero or infinity. If this is the case, why not simply design our activation function around this property? The Rectified Linear Unit (ReLU; Fukushima 1980; Nair and Hinton 2010) is an activation function that does precisely this[15]:



$$\mathrm{ReLU}(\mathbf{x}) = \max(\mathbf{x}, 0). \tag{1.16}$$

The gradient of ReLU is one if the inputs are above zero, exactly the property we needed to mitigate the vanishing gradient problem. Similar non-saturating activation functions also share the ReLU gradient's useful property, see for example the Exponential Linear Unit, Swish, and Mish functions in Fig. 1.6.

1.6.2 Graphical processing unit acceleration

If we can speed up training, we can run an inefficient algorithm (such as backpropagation through saturating activations) to completion in less time. One way to speed up training is by using hardware that is specifically suited to the training of neural networks. Graphical processing units (GPUs) were originally developed to render video games and other intensive graphical processing tasks. These rendering tasks require a processor capable of massive parallelism. We have seen in the previous sections that neural networks trained through backpropagation also require many small weight update calculations. With this in mind, it is natural to try to accelerate deep neural networks with GPUs.

Oh and Jung (2004) were the first to use GPUs to accelerate an MLP model, reporting a 20× performance increase in inference with their GPU accelerated neural

[15] ReLU′ is always zero if its inputs are < 0, removing any signal for further training. This is known as the 'dying ReLU' problem, but is not as big of an issue as it first seems. Since contemporary deep neural networks are greatly overparameterised (see for example Frankle and Carbin 2018, and other work on the 'lottery ticket hypothesis') backpropagation through the ReLU activation function can act as a pruning mechanism, creating sparse representations within the neural network and thus reducing training time even further (Glorot, Bordes, and Bengio 2011).

network. Shortly after, Steinkrau, Simard, and Buck (2005) showed that backpropagation can also benefit from GPU acceleration, reporting a three-fold performance increase in both training and inference. These two breakthroughs were followed by a flurry of activity in the area (e.g. Chellapilla, Puri, and Simard 2006; Raina, Madhavan, and Ng 2009; Cireşan et al. 2010, 2011), culminating in a milestone victory for GPU accelerated neural networks at ImageNet 2012. AlexNet (Krizhevsky, Sutskever, and Hinton 2012) won the ImageNet classification and localisation challenges (Russakovsky et al. 2015), scoring an unprecedented top-5 classification error of 16.4%, and a single object localisation error of 34.2%. In both challenges AlexNet scored over 10% better than the models in second place. Krizhevsky, Sutskever, and Hinton's winning network was a CNN (Fukushima 1980) trained through backpropagation (Linnainmaa 1976; LeCun et al. 1989), with ReLU activation (Nair and Hinton 2010), and dropout (Srivastava et al. 2014a) as a regulariser. The performance increase afforded by GPU accelerated training enabled the network to be trained from scratch via backpropagation in a reasonable amount of time. The discovery that it is possible to train a neural network from scratch by using readily available hardware ultimately resulted in the end of connectionism's second winter, and ushered in the Cambrianeqsue deep learning explosion of the mid-to-late 2010s and 2020s (Fig. 1.10).

1.6.3 Gated recurrent neural networks and residual networks

The Long-Short Term Memory unit (LSTM; Hochreiter and Schmidhuber 1997; Gers, Schmidhuber, and Cummins 2000)[16] mitigates the vanishing gradient problem by introducing a new hidden state, the 'cell state' ($\mathbf{c}_n$), to the standard recurrent neural network architecture. This cell state allows the network to learn long range dependencies, and I will show why this is the case via a brief derivation[17]. First, as always, let us study Fig. 1.11 and write down the forward pass equation for updating the cell state:

$$\mathbf{c}_n = f(\mathbf{c}_{n-1}, \mathbf{h}_{n-1}, \mathbf{x}_n) + g(\mathbf{h}_{n-1}, \mathbf{x}_n)$$

where $f(\mathbf{c}_{n-1}, \mathbf{h}_{n-1}, \mathbf{x}_n) = \mathbf{c}_{n-1} \odot \varphi(\mathbf{h}_{n-1}, \mathbf{x}_n)$. For brevity I define $\varphi_n = \varphi(\mathbf{h}_{n-1}, \mathbf{x}_n)$.

Like the RNN case (Eq. 1.9 and Eq. 1.10), we will need to find $\partial \mathbf{c}_n / \partial \mathbf{c}_{n-1}$ to

[16] Compare also the Gated Recurrent Unit (GRU; Cho et al. 2014).

[17] Here I loosely follow Bayer (2015, §1.3.4).

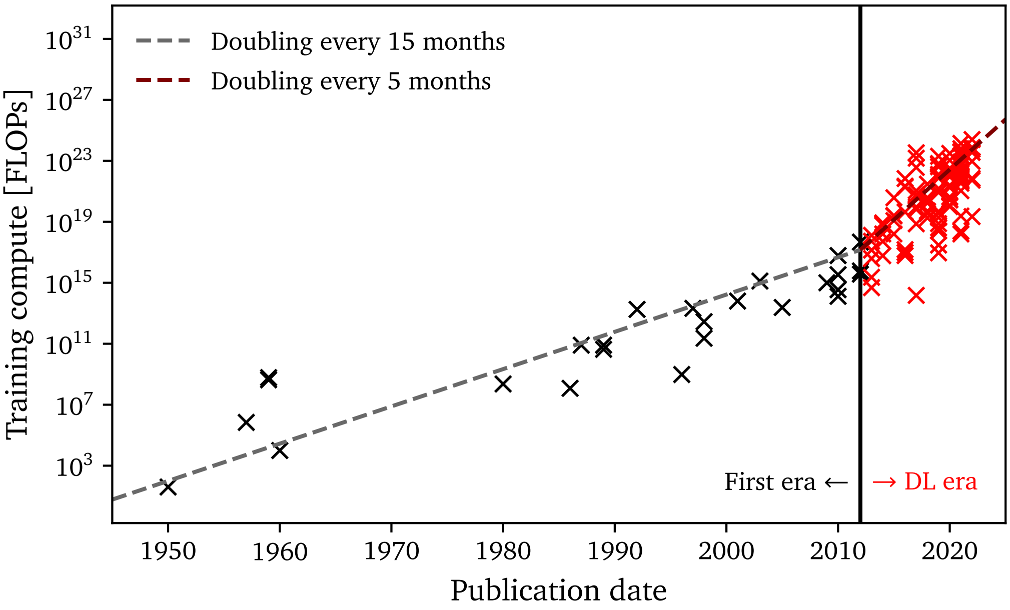


Figure 1.10: If we plot the total number of floating point operations (FLOPs) required to train a neural network model, and compare it to the model's publication date, we can see a change in trend at around 2012. This corresponds to the popularisation of GPU accelerated training of very deep neural networks, with 2012 demarcating AI's 'Deep Learning Era' and the beginning of astronomy's second wave of connectionism (§1.8). Data is taken from Sevilla et al. (2022).

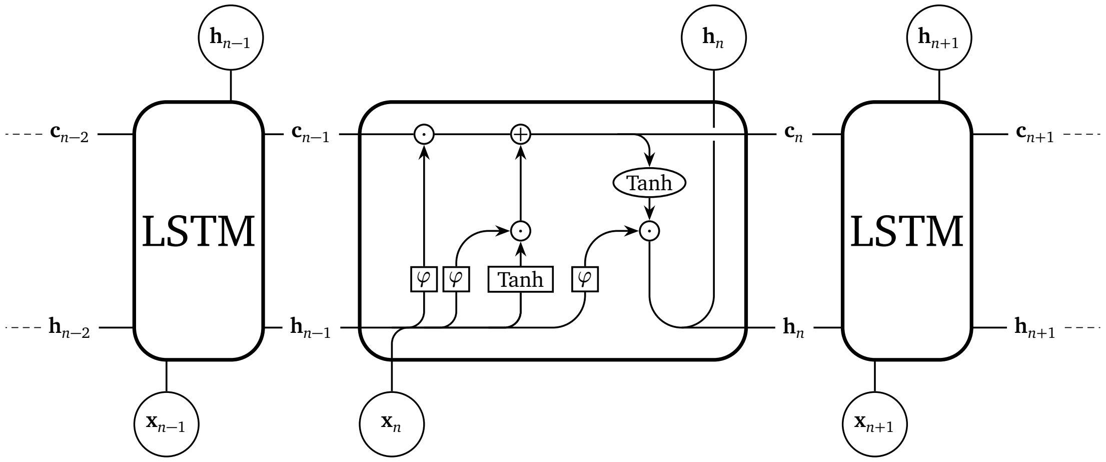


Figure 1.11: A set of sequential data $\mathbf{x}_n$ is input into an LSTM network. Inside the cell ○ denotes elementwise operations and □ denotes neuronal layers. φ is the sigmoid activation function, and Tanh is the hyperbolic tangent activation function. ⊕ is an elementwise addition, ⊙ is the Hadamard product, and line mergers are concatenations. $\mathbf{c}_n$ is the cell state, and $\mathbf{h}_n$ is the hidden state.

calculate $\nabla\mathscr{L}$. Therefore,

$$\begin{aligned}
\frac{\partial \mathbf{c}_n}{\partial \mathbf{c}_{n-1}} &= \frac{\partial f(\mathbf{c}_{n-1}, \mathbf{h}_{n-1}, \mathbf{x}_n)}{\partial \mathbf{c}_{n-1}} + \cancelto{0}{\frac{\partial g(\mathbf{h}_{n-1}, \mathbf{x}_n)}{\partial \mathbf{c}_{n-1}}}, \\
&= \frac{\partial \mathbf{c}_{n-1} \odot \varphi_n}{\partial \mathbf{c}_{n-1}}, \\
&= \mathbf{c}_{n-1} \cancelto{0}{\frac{\partial \varphi_n}{\partial \mathbf{c}_{n-1}}} + \cancelto{1}{\frac{\partial \mathbf{c}_{n-1}}{\partial \mathbf{c}_{n-1}}} \varphi_n, \\
&= \varphi_n.
\end{aligned}$$

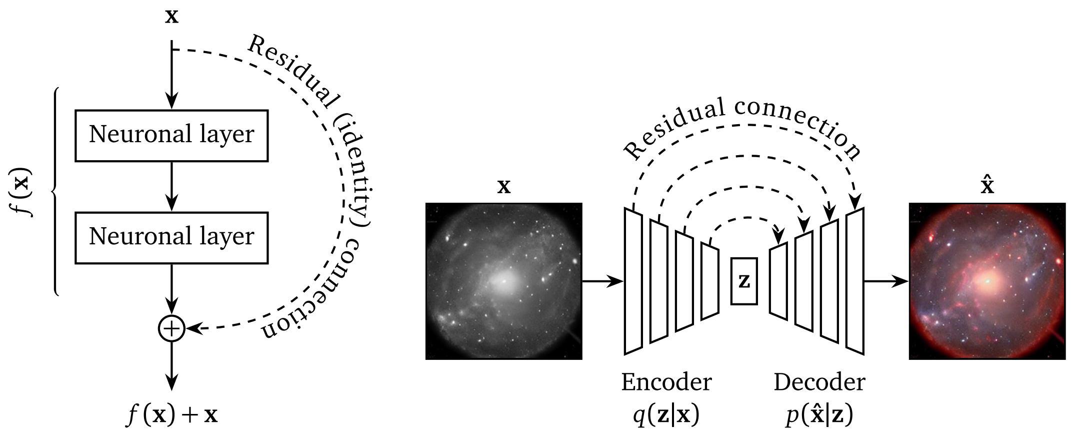
Thus, if we want to backpropagate to a cell state deep in the network we must do

$$\frac{\partial \mathbf{c}_n}{\partial \mathbf{c}_N} = \prod_{i=1}^{n-N} \varphi_i, \quad n > N. \tag{1.17}$$

The product term above does not depend on the derivative of a saturating activation function, and so does not automatically go to zero as N goes to ∞. This means that a gradient signal can be carried through the LSTM cell state without losing amplitude and vanishing[18].

We can use a technique derived from the LSTM to solve our vanishing gradient problem for deep feedforward neural networks (as studied in §1.2). Srivastava, Greff, and Schmidhuber (2015) do this by applying the concept of the LSTM's cell state to their deep convolutional 'Highway Network'. The Highway Network uses gated connections to modulate the gradient flow back through neuronal layers. Later work by He et al. (2015) introduces the 'ResNet' by taking the Highway Network and simplifying its connections. They apply an elementwise addition (or 'residual connection') in place of the Highway Network's gated connection (Fig. 1.12a). One can go even further with residual connections, as Ronneberger, Fischer, and Brox (2015) demonstrate with their U-Net model. The U-Net combines residual connections with an autoencoder-like architecture (Fig. 1.12b). The U-Net has gone on to become the *de facto* network for many tasks that require an input and output of the same size (such as segmentation, colourisation, and style transfer).

[18] Which is great in theory. In practice, LSTMs still have trouble learning very long range dependencies due to their reliance on recurrent processing (Sutskever, Vinyals, and Le 2014). Transformer networks (Vaswani et al. 2017) are an architecture that uses the concept of attention to address this issue. We will discuss transformer networks in §1.7.


x
Residual (identity) connection
Neuronal layer
Neuronal layer
$f(\mathbf{x})$
$f(\mathbf{x})+\mathbf{x}$
Residual connection
x
z
$\hat{\mathbf{x}}$
Encoder
$q(\mathbf{z}|\mathbf{x})$
Decoder
$p(\hat{\mathbf{x}}|\mathbf{z})$

(a) A single residual connection is applied within a neural network.

(b) The U-Net, a network that was originally developed to segment biological imagery uses the residual connection.

Figure 1.12: The left subfigure shows the residual connection as originally introduced in He et al. (2015). The right subfigure shows an application of the residual connection to an autoencoder like architecture (Ronneberger, Fischer, and Brox 2015), in this case colourising an astronomical object.

1.7 Translation, attention, and transformers

Theoretically, Gated RNNs (GRNNs) such as the LSTM can learn very long range dependencies (see Eq. 1.17 and its accompanying text). In practice, GRNNs tend to forget information about distant inputs. This is because the GRNN lacks unmediated access to inputs beyond the immediate antecedent as a consequence of its recurrent architecture. The problem is especially apparent in neural machine translation tasks that require knowledge of an entire sequence to produce an output, such as language to language translation. Fig. 1.13 shows such a sequence to sequence (Seq2Seq; Sutskever, Vinyals, and Le 2014) model. Seq2Seq translates between two sets of sequential data by sharing a hidden state between two GRNN units. In Fig. 1.13 we can see that the shared information is bottlenecked by the hidden state. Therefore, to resolve the GRNN ‘forgetting problem’ we must find a way to avoid any recursion, or serial processing of input and output. We can do this by providing the neural network access to all input while it is calculating an output. This was the primary motivation behind the transformer architecture (Bahdanau, Cho, and Bengio 2014; Vaswani et al. 2017).

Modern transformer architectures consist of a series of self-attention layers, often

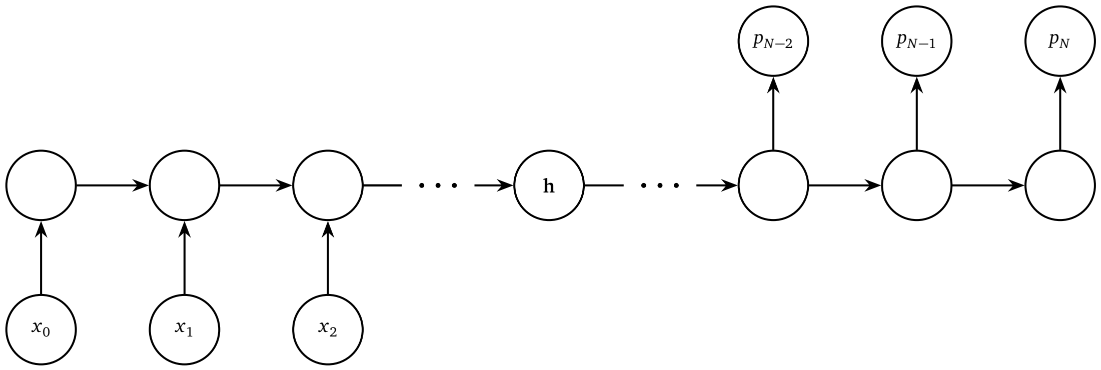


Figure 1.13: A sequence to sequence (Seq2Seq; Sutskever, Vinyals, and Le 2014) model. A sequence **x** is input into a GRNN. The final hidden state (**h**) of the input network is then passed into a second GRNN. The second GRNN then unrolls to predict an output sequence **p**. Due to the hidden state acting as a intermediary, **x** and **p** need not be of equal length. I explore an astronomical use case of this architecture in §2.

interspersed with other layer types[19]. Self-attention as described in Vaswani et al. (2017) is shown in Fig. 1.14. Intuitively, it captures the relationships between quanta within a data input. To perform self-attention we first take an input sequence

$$\mathbf{x} = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix},$$

where **x** can be any sequence, such as a sentence, a variable star's time series, or an unravelled galaxy image[20]. Here we will follow the literature and refer to $[x_1, \ldots, x_n]$ as tokens. As we can see in Fig. 1.14 the input is passed through a trainable pair of weight matrices **Q** and **K**. The output matrices **q** and $\mathbf{k}^\dagger$ are then multiplied together to yield

$$(\mathbf{Q} \cdot \mathbf{x})(\mathbf{K} \cdot \mathbf{x})^\dagger = \mathbf{q}\mathbf{k}^\dagger = \begin{bmatrix} Q_1x_1K_1x_1 & Q_1x_1K_2x_2 & \cdots & Q_1x_1K_nx_n \\ Q_2x_2K_1x_1 & Q_2x_2K_2x_2 & \cdots & Q_2x_2K_nx_n \\ \vdots & \vdots & \ddots & \vdots \\ Q_nx_nK_1x_1 & Q_nx_nK_2x_2 & \cdots & Q_nx_nK_nx_n \end{bmatrix}. \tag{1.18}$$

We can see that Eq. 1.18 describes the relationships between tokens within **x**. For

[19] In the original transformer formulation described in Vaswani et al. (2017), the network consisted of a connected 'encoder' and 'decoder' section much like a Seq2Seq model (Fig. 1.13). Later work has found this to be an unnecessary complication. For example, the Generative Pretrained Transformer (GPT) 2 and 3 models (Radford et al. 2019; Brown et al. 2020) consist of only decoder layers, and the Bidirectional Encoder Representations from Transformers (BERT) model consists of only encoder layers (Devlin et al. 2019).

[20] One can go very general with this, as DeepMind demonstrated with their 'Gato' Transformer model (Reed et al. 2022). Gato can predict sequences for myriad tasks, from operating a physical robotic arm, to completing natural language sentences, to playing Atari games.

example, if x_1 is similar semantically to x_2, we would expect $Q_1x_1K_2x_2$ and $Q_2x_2K_1x_1$ to have a high value. We then normalise $\mathbf{qk}^\dagger$ to mitigate vanishing gradients[21], and apply a softmax nonlinearity so that the maximum weighting (or similarity) is one.

Meanwhile, the input sequence $\mathbf{x}$ is passed through the neuronal layer $\mathbf{V}$, resulting in a weighted representation $\mathbf{v}$:

$$\mathbf{V} \cdot \mathbf{x} = \mathbf{v} = \begin{bmatrix} V_1x_1 & V_2x_2 & \cdots & V_nx_n \end{bmatrix}.$$

$\mathbf{v}$ is multiplied with the similarity matrix $\varsigma(\mathbf{qk}^\dagger/\sqrt{n})$. This process weighs similar tokens within the sequence higher, increasing their relative importance in later neuronal layers.

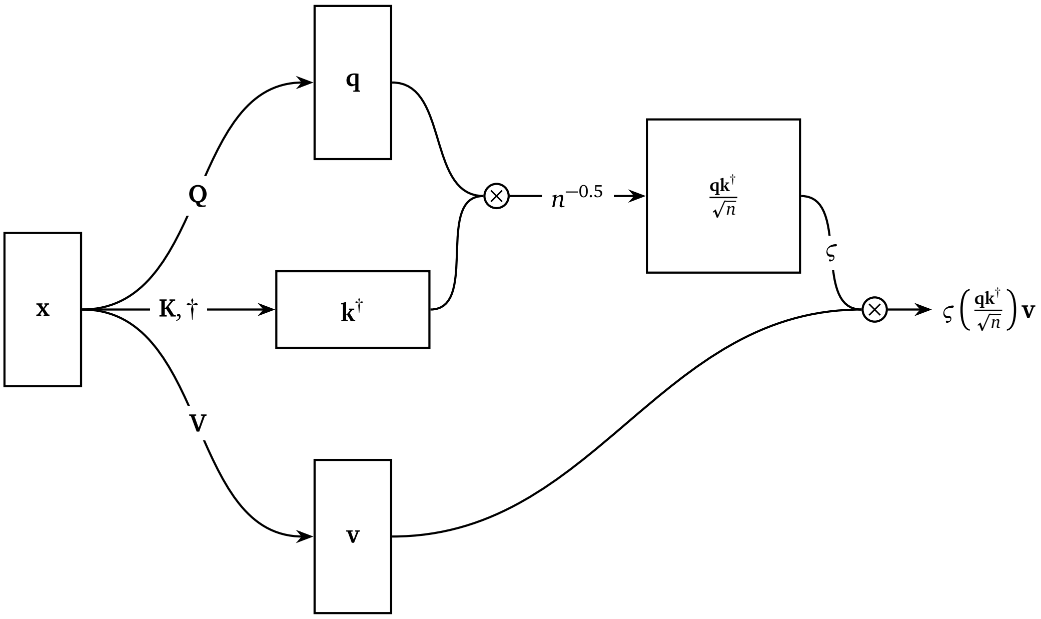

q
Q
x
K, †
k†
V
v
$n^{-0.5}$
$\frac{\mathbf{qk}^\dagger}{\sqrt{n}}$
ς
$\varsigma\left(\frac{\mathbf{qk}^\dagger}{\sqrt{n}}\right)\mathbf{v}$

Figure 1.14: An input ($\mathbf{x}$) is fed into a self-attention mechanism. The weights used to produce the query ($\mathbf{q}$), key ($\mathbf{k}$), and value ($\mathbf{v}$) matrices are learnt via backpropagation. Here the learnt weights are denoted as the capitalised versions of their child matrices. $\mathbf{q}$ and $\mathbf{k}$ are normalised and multiplied together, and a softmax nonlinearity (ς) is applied. Finally, $\mathbf{v}$ is multiplied with output of the upper path and the final output is fed forward to the next neuronal layer. $\otimes$ denotes a matrix multiplication.

1.8 Astronomy's second wave of connectionism

Compared to classical machine learning techniques[22] deep learning as outlined in §1.6.2 does not require an extraction of emergent parameters to train its models.

[21] See Footnote 14.

[22] This includes most MLP applications in astronomy, see §1.3.

RNNs in particular are well suited to observing the full raw information within a time series. Likewise, CNNs are well suited to observing raw information within pictoral-like data. Astronomy is rich with both types of data, and in this subsection I will review the history of the application of RNN, CNN, and transformer models to astronomical data.

1.8.1 Recurrent neural network applications

RNNs were first applied in astronomy to areas very close to home; Aussem, Murtagh, and Sarazin (1994) predicted atmospheric seeing for observations from ESO's Very Large Telescope, and the prediction of geomagnetic storms given data on solar wind was also explored in the mid-to-late 1990s and early 2000s (Wu and Lundstedt 1996, Lundstedt, Gleisner, and Wintoft 2002, and other work from the same group; Vassiliadis et al. 2000).

The first use of RNNs for classification in astronomy was carried out in in a prescient study by Brodrick, Taylor, and Diederich (2004). They describe the use of an RNN-like Elman network (Elman 1990). Their RNN was tasked with the search for artificially generated narrowband radio signals that resemble those that may be produced by an extraterrestrial civilisation. They found that their model had a test set accuracy of 92%, suggesting that RNNs could be a useful tool in the search for extraterrestrial intelligence. More than a decade after Brodrick, Taylor, and Diederich (2004), Charnock and Moss (2017) used an LSTM (Fig. 1.11) to classify simulated supernovae. They describe two classification problems. One, a binary classification between type-Ia and non type-Ia supernovae, and the other a classification between the supernovae types I, II, and III. For their best performing model they report an accuracy of more than 95% for their binary classification problem, and an accuracy of over 90% for their trinary classification. This study cemented the usefulness of RNNs for classification problems in astronomy. Charnock and Moss (2017) was followed by numerous projects studying the use of RNNs for classification of time series astronomical data. A non-exhaustive list of modern RNN use in astronomy includes: stochastically sampled variable star classification (Naul et al. 2018); exoplanet instance segmentation (Gonzalez, Absil, and Van Droogenbroeck 2018); variable star/galaxy sequential imagery classification (Carrasco-Davis et al. 2019); and gamma ray source classification (Finke, Krämer, and Manconi 2021). We must conclude from these studies that RNNs are effective classifiers of astronomical time series, provided that sufficient data is available.

Of course, recurrent networks are not limited to classification; they can also be used for regression problems. First, Weddell and Webb (2008) successfully used an echo state network (Jaeger and Haas 2004) to predict the point spread function of a target object in a wide field of view. Capizzi, Napoli, and Paternò (2012) used an RNN to inpaint missing NASA Kepler time series data for stellar objects. They found that their model could recreate the missing time series to an excellent accuracy, suggesting that the RNN could internalise information about the star it was trained on. As in the classification case, research into the use of RNNs for regression problems picked up massively in the late 2010s, and here I will highlight a selection of these studies that represent the range of RNN use cases. H. Shen et al. (2017) used both an LSTM and an autoencoder based RNN to denoise gravitational wave data, and Morningstar et al. (2019) used a recurrent inference machine to reconstruct gravitationally lensed galaxies. H. Liu et al. (2019) used an LSTM to predict solar flare activity. From these studies, similarly to the classification case above, we can once again conclude that RNNs are effective regressors of astronomical time series.

RNNs have also been used in cases that are a little more unconventional. For example, Kügler, Gianniotis, and Polsterer (2016) used an autoencoding RNN (specifically an echo state network) to extract representation embeddings of variable main sequence stars. They find that these embeddings capture some emergent properties of these variable stars, such as temperature, and surface gravity, suggesting that clustering within the embedding space could result in semantically meaningful variable star classification. We will revisit this line of research when I explore representation learning within astronomy in detail in §1.11. M. J. Smith et al. (2021) use an encoder-decoder network comprising of a CNN encoder and RNN decoder to predict surface brightness (SB) profiles of galaxies. This class of neural network was previously used extensively within natural language image captioning, and by treating SB profiles as 'captions' their model was capable of prediction over 100× faster than the previous method. M. J. Smith et al. (2021) is presented in full in §2.

1.8.2 Convolutional neural network applications

It did not take long after Krizhevsky, Sutskever, and Hinton (2012) established CNNs as the *de facto* image classification network for astronomers to take notice: in 2014 they were applied in the search for pulsars (W. W. Zhu et al. 2014) as part of an ensemble of methods. W. W. Zhu et al. (2014) found that their ensemble was highly effective, with 100% of their test set pulsar candidates being ranked within the top 961 of the 90 008 test candidates. Shortly after, Hála (2014) described the use of one dimensional CNNs for a ternary classification problem. They find that their model is capable of classifying 1D spectra into quasars, galaxies, and stars to an impressive accuracy. CNNs have been also been extensively used in galaxy morphological classification. First on the scene was Dieleman, Willett, and Dambre (2015). They used CNNs to classify galaxy morphology parameters as defined in the Galaxy Zoo dataset (Raddick et al. 2010) from galaxy imagery. They observed their galaxies via the SDSS, and found a 99% consensus between the Galaxy Zoo labels, and the CNN classifications. Huertas-Company et al. (2015) showed that Dieleman, Willett, and Dambre's CNN is equally applicable to morphological classification of galaxies in the CANDELS fields. Likewise, Aniyan and Thorat (2017) showed that CNNs are capable of classifying radio galaxies. The combined work of Dieleman, Willett, and Dambre (2015), Huertas-Company et al. (2015), and Aniyan and Thorat (2017) confirms that CNNs are equally applicable to visually dissimilar surveys, with little-to-no modification. Wilde et al. (2022) used a deep CNN model to classify simulated lensing events. They also applied some interpretability techniques to their data, using occlusion mapping (Zeiler and Fergus 2014), gradient class activation mapping (Selvaraju et al. 2016), and Google's DeepDream to prove that the CNN was indeed classifying via observing the gravitational lenses. We can conclude from the studies described in this paragraph that CNNs are effective classifiers and regressors of raw pictoral-like astronomical data.

Alternative CNN models have also been used, such as the U-Net (Fig. 1.12b). The U-Net was initially developed to segment biological imagery (Ronneberger, Fischer, and Brox 2015). Its first use in astronomy was related: Akeret et al. (2017) use a U-Net (Ronneberger, Fischer, and Brox 2015) CNN to isolate via segmentation, and ultimately remove, radio frequency interference from radio telescope data. Likewise, Berger and Stein (2019) used a three dimensional U-Net (V-Net; Milletari, Navab, and Ahmadi 2016) to predict and segment out galaxy dark matter haloes in simulations, and Aragon-Calvo (2019) used a V-Net to segment out the cosmological filaments and walls that make up the large scale structure of the Universe. Lauritsen et al. (2021) use a U-Net to superresolve simulated submillimetre observations. They found that the U-Net could successfully do this, when using a loss comprising of the L1 loss, and a custom loss that measures the distance between predicted and ground truth point sources.

1.8.3 Transformer applications

Although initially used for natural language, transformers have also been adapted for use in imagery; first by Parmar et al. (2018), and also in Dosovitskiy et al. (2020). Transformers have not yet been applied to astronomical imagery, but they have started to find use in time-series astronomy. Donoso-Oliva et al. (2022) used BERT (Devlin et al. 2019) to generate a representation space for light curves in a self-supervised manner. Morvan et al. (2022) use an encoding transformer to denoise light curves from the transiting exoplanet survey satellite (TESS; Ricker et al. 2015), and show that the denoising surrogate task results in an expressive embedding space. Pan, Ting, and Yu (2022) also use a transformer model to analyse light curves for exoplanets. Transformers have taken the fields of natural language processing and computer vision by storm (§5.1), and so if we extrapolate from trends in other fields I expect to see many more examples of transformers applied to astronomical use cases in the near future. We will revisit the transformer architecture in the context of foundation models (Bommasani et al. 2021, and references therein), and their possible future astronomical uses in §5.1.

1.8.4 A problem with supervised learning

Supervised learning requires a high quality labelled dataset to train a neural network. In turn, these datasets require labourious human intervention to create, and so supervised data is in short supply. One can avoid this issue by letting the deep learning model infer the data labels itself, and project these labels onto a hidden descriptive 'latent space'. In §1.9 and §1.10 I will explore some deep learning models that do not require supervision.

1.9 Deep generative modelling

In this section I discuss generative modelling within the context of astronomy. Unlike discriminative models, generative models explicitly learn the distribution of classes in a dataset (Fig. 1.15). Once we learn the distribution of data, we can use that knowledge to generate new synthetic data that resembles that found in the training dataset. In the following sections I will explore in detail three popular forms of deep generative model: the variational autoencoder (§1.9.1); the generative adversarial network (§1.9.2); and the family of score-based (or diffusion) models (§1.9.3). Finally, in §1.11 I discuss applications of deep generative modelling in astronomy.

1.9.1 (Variational) autoencoders

Autoencoders have long been a neural network architectural staple; in a sister paper to backpropagation's populariser, Rumelhart, Hinton, and Williams (1986a) demonstrate backpropagation within an autoencoder. Fig. 1.16 demonstrates the basic neural network autoencoder architecture. An autoencoder is tasked with recreating some input data, squeezing the input information ($\mathbf{x}$) into a bottleneck latent vector ($\mathbf{z}$)

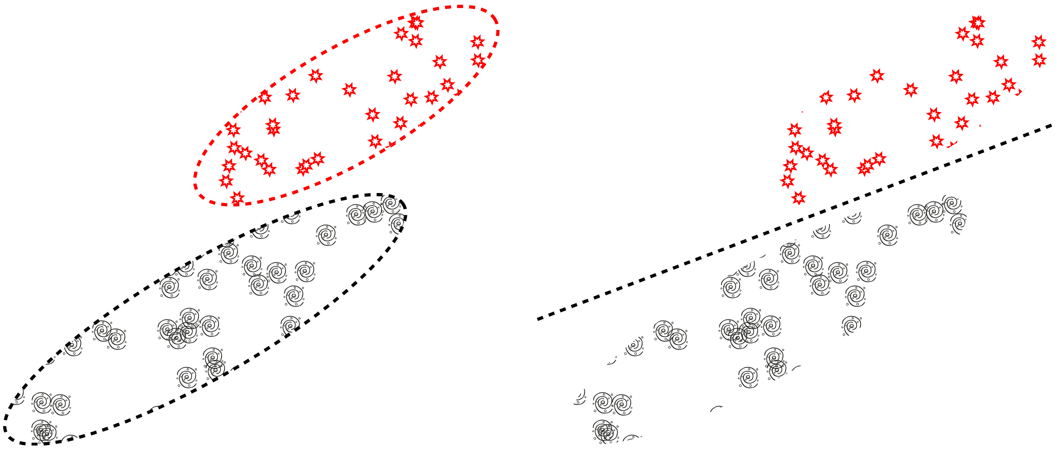


Figure 1.15: On the left we see a generative model attempting to learn the probability distributions of a dataset that contains a set of galaxies, and a set of stars. On the right is a discriminative model, which is attempting to learn the boundary that separates the star and galaxy types.

via a neural network $q(\mathbf{z}|\mathbf{x})$. $\mathbf{z}$ is then expanded to an imitation of the input data ($\hat{\mathbf{x}}$) by a second neural network $p(\hat{\mathbf{x}}|\mathbf{z})$. The standard autoencoder is trained via a reconstruction loss; $\mathscr{L}_R(\mathbf{x}, \hat{\mathbf{x}})$, where $\mathscr{L}_R(\mathbf{x}, \hat{\mathbf{x}})$ measures the difference in pixelspace between $\mathbf{x}$ and $\hat{\mathbf{x}}$.

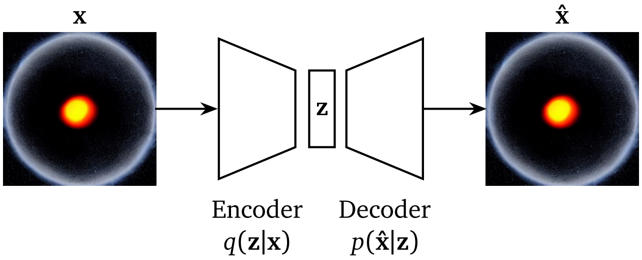


Figure 1.16: An autoencoder (Rumelhart, Hinton, and Williams 1986a) attends to an image of a black hole. $\mathbf{z}$ is a latent vector and $\mathbf{x}$ is a sample from a training set. The encoder, q learns to encode the incoming data into a latent vector while the decoder p takes as input $\mathbf{z}$ and attempts to recreate $\mathbf{x}$.

Naïvely, one would think that once trained, one could 'just' sample a new latent vector, and produce novel imagery via the decoding neural network $p(\hat{\mathbf{x}}|\mathbf{z})$. We cannot do this, as autoencoders trained purely via a reconstruction loss have no incentive to produce a smoothly interpolatable latent space. This means we can use a standard autoencoder to embed and retrieve data contained in the training set, but cannot use one to generate new data. To generate new data we require a smooth latent space, which variational autoencoders (VAEs; Fig 1.17) produce by design (Kingma and Welling 2013).

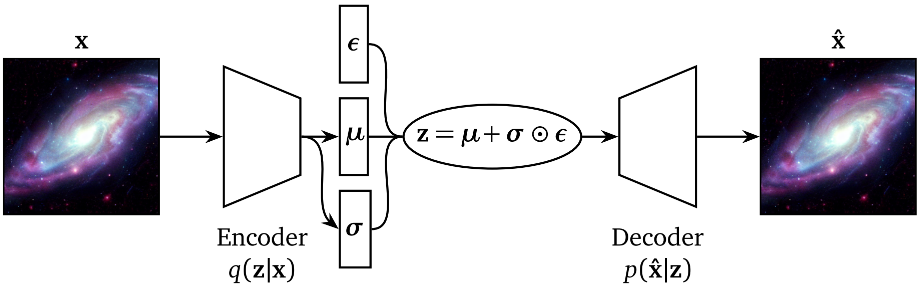

$\mathbf{x}$
$\hat{\mathbf{x}}$
ϵ
μ
σ
$\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}$
Encoder
$q(\mathbf{z}|\mathbf{x})$
Decoder
$p(\hat{\mathbf{x}}|\mathbf{z})$

Figure 1.17: A variational autoencoder (Kingma and Welling 2013) operates on a spiral galaxy. $\mathbf{z}$ is a latent vector and $\mathbf{x}$ is a sample from the training set. The encoder, q learns to compress the incoming data into a latent vector that encodes the normal distribution. The decoder p takes as input $\mathbf{z}$ and attempts to recreate $\mathbf{x}$.

A VAE differs from the standard autoencoder by enforcing a spread in each training set samples' latent vector. We can see in Fig. 1.17 how this is done; instead of directly predicting $\mathbf{z}$ the encoder q predicts two vectors, $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$. $\mathbf{z}$ is then sampled stochastically via the equation

$$\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}, \tag{1.19}$$

where $\odot$ is the Hadamard product, and $\boldsymbol{\epsilon}$ is noise generated externally to the neural network graph[23]. This spread results in similar samples overlapping within the latent space, and therefore we end up with a smooth latent space that we can interpolate through. However, currently there is no incentive for the neural network to provide a coherent, compact global structure in the latent space. For that we require a regularisation term in the loss. This regularisation is provided via the Kullback-Leibler (KL) divergence, which is a measure of the difference between two probability distributions. A standard VAE uses the KL divergence to push the latent distribution towards the standard normal distribution, incentivising a compact, continuous latent space. Hence, the final VAE loss is a combination of the reconstruction loss and KL divergence:

$$\mathscr{L}_{\text{VAE}} = \mathscr{L}_R(\mathbf{x}, \hat{\mathbf{x}}) + \text{KL}(q(\mathbf{x}|\mathbf{z})||p), \tag{1.20}$$

where p is some prior. In a standard VAE $p = \mathscr{N}(\mathbf{0}, \mathbb{1})$.

1.9.2 Generative adversarial networks

Generative adversarial networks (GAN; Goodfellow et al. 2014) can be thought of as a minimax game between two competing neural networks. If we anthropomorphise

[23] To avoid breaking the backpropagation chain the VAE injects noise via an external parameter, $\boldsymbol{\epsilon}$. This is described in Kingma and Welling (2013) as the 'reparameterisation trick'.

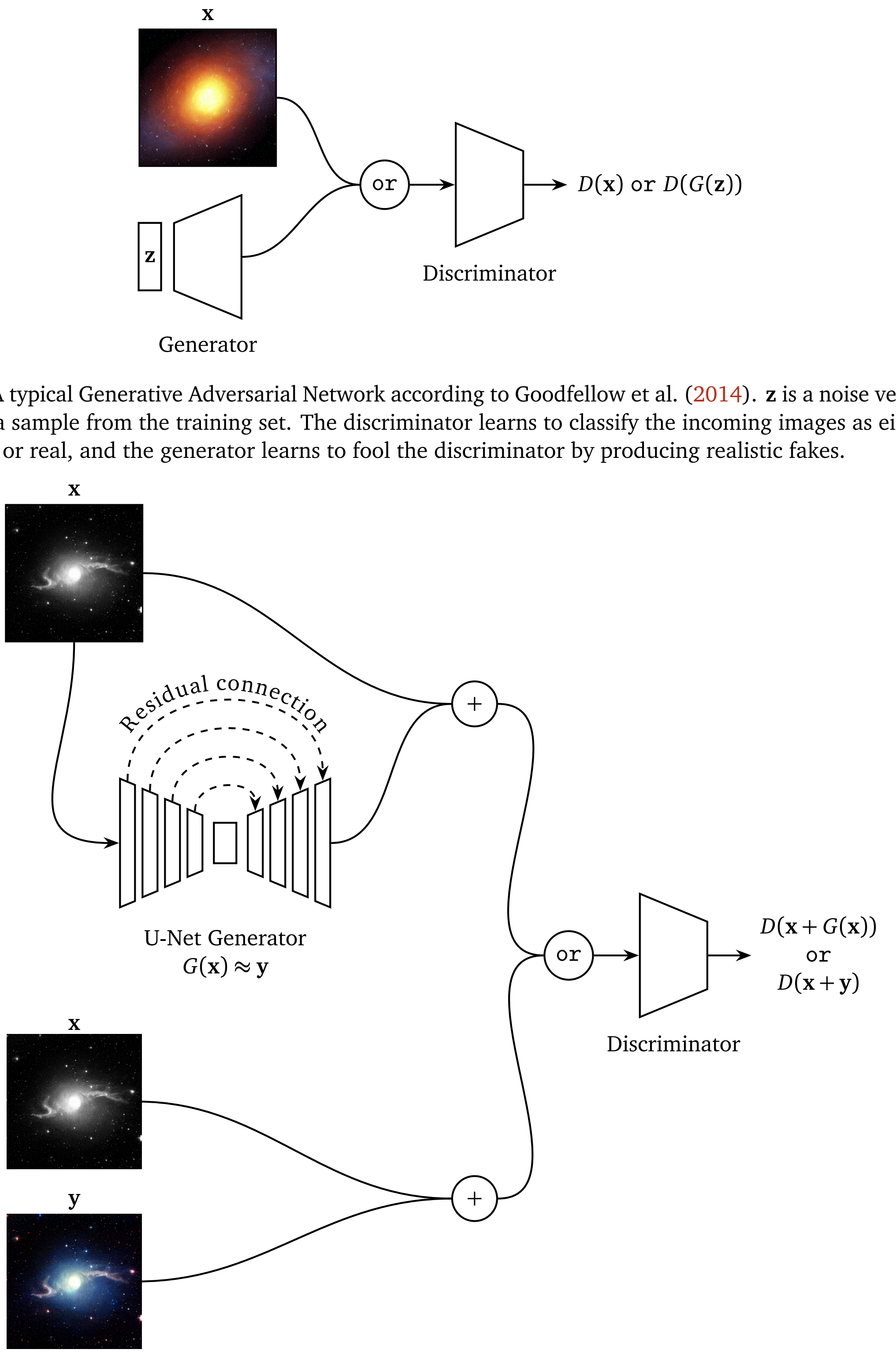
we can gain an intuition for how a GAN learns: let us imagine an art forger, and an art critic. The forger wants to paint paintings that are similar to famous expensive works, and needs to fool the critic when selling these paintings. Meanwhile, the critic wants to ensure that no reproductions are sold and so he needs to accurately determine whether any painting is an original or a reproduction. At first, our forger is a poor painter, and so the critic can easily identify our forger's works. However, the forger learns from the critic's choices and produces more realistic paintings. As the forger's paintings improve, the critic also learns better methods for detecting forgeries. This minimax game incentivises the critic to keep improving his classifications, and the forger to keep improving his painting. If this continues, we get to a point where the forger's works are indiscernible from the real thing—the forger has learnt to perfectly mimic the dataset! In a GAN, we name the critic the discriminator (D), and we name the forger the generator (G).

In Goodfellow et al.'s original GAN formulation (Fig. 1.18a), G and D compete during training in a minimax game where G aims to maximise the probability of D mispredicting that a generated datapoint is sampled from the real dataset. G takes as input a randomly sampled latent vector $\mathbf{z}$, and outputs a synthetic datapoint $G(\mathbf{z})$. D takes either this synthetic datapoint, or a real datapoint $\mathbf{x}$, and outputs $D(G(\mathbf{z}))$ or $D(\mathbf{x})$. This output is the probability that the datapoint is drawn from the real dataset. We compare this probability to a binary label indicating whether the datapoint is real or not, and backpropagate the error through both D and G. The network's weights are updated with each training datapoint to follow $\nabla_{\mathbf{w}}\mathscr{L}$ downwards until the distribution of $G(\mathbf{z})$ closely resembles that of the real dataset. Once trained, G can be used to generate entirely novel synthetic data that closely resembles (but is not identical to) the training set data.

In Fig. 1.18b we see that the GAN adversarial loss can be used to translate between image domains (Isola et al. 2016). In Isola et al.'s Pix2pix model, the generator takes as input an image $\mathbf{x}$, and attempts to produce a related image $\mathbf{y}$. Meanwhile, the discriminator attempts to discern whether the $(\mathbf{x}, \mathbf{y})$ pair that it is given is sampled from the training set, or the generator. Otherwise, Pix2Pix is trained in the same way as the standard GAN.

1.9.3 Score-based generative modelling and diffusion models

Diffusion models were introduced by Sohl-Dickstein et al. (2015) and were first shown to be capable of producing high quality synthetic samples by Ho, Jain, and Abbeel (2020). Diffusion models are part of a family of generative deep learning models that employ denoising score matching via annealed Langevin dynamic sampling (first


x
z
Generator
or
Discriminator
$D(\mathbf{x})$ or $D(G(\mathbf{z}))$
x
Residual connection
U-Net Generator
$G(\mathbf{x}) \approx \mathbf{y}$
+
or
Discriminator
$D(\mathbf{x}+G(\mathbf{x}))$
or
$D(\mathbf{x}+\mathbf{y})$
x
y
+

(a) A typical Generative Adversarial Network according to Goodfellow et al. (2014). $\mathbf{z}$ is a noise vector, $\mathbf{x}$ is a sample from the training set. The discriminator learns to classify the incoming images as either fake or real, and the generator learns to fool the discriminator by producing realistic fakes.

(b) A Pix2Pix-like model with a U-Net generator (Ronneberger, Fischer, and Brox 2015; Isola et al. 2016). The discriminator learns to classify the incoming image tuples as either fake or real. Meanwhile, the generator learns to fool the discriminator by approximating the colourisation function mapping $\mathbf{x} \rightarrow \mathbf{y}$.

Figure 1.18: The GAN and Pix2Pix models.

explored by Hyvärinen (2005) and Vincent (2011). More recent work can be found in Ho, Jain, and Abbeel (2020), Jolicoeur-Martineau et al. (2020), Song and Ermon (2020), Jolicoeur-Martineau et al. (2021), and Song et al. (2021)). This family of score-based generative models (SBGMs) can generate imagery of a quality and diversity surpassing state of the art GAN models (Goodfellow et al. 2014), a startling result considering the historic disparity in interest and development between the two techniques (Dhariwal and Nichol 2021; Nichol and Dhariwal 2021; Song et al. 2021; Ramesh et al. 2022). SBGMs can super-resolve images (Kadkhodaie and Simoncelli 2020; Saharia et al. 2021), translate between image domains (Sasaki, Willcocks, and Breckon 2021), separate superimposed images (Jayaram and Thickstun 2020), and in-paint information (Kadkhodaie and Simoncelli 2020; Song et al. 2021).

Diffusion models define a diffusion process that projects a complex image domain space onto a simple domain space. In the original formulation, this diffusion process is fixed to a predefined Markov chain $q(\mathbf{x}_t \mid \mathbf{x}_{t-1})$ that adds a small amount of Gaussian noise with each step. As Fig. 4.1 shows, this 'simple domain space' can be noise sampled from a Gaussian distribution $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbb{1})$.

Forward process

To slowly add Gaussian noise to our data we define a Markov chain

$$q(\mathbf{x}_{0\ldots T}) = q(\mathbf{x}_0) \prod_{t=1}^{T} q(\mathbf{x}_t \mid \mathbf{x}_{t-1}).$$

The amount of noise added per step is controlled with a variance schedule $\{\beta_t \in (0,1)\}_{t=1}^{T}$:

$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\,\mathbf{x}_{t-1}, \beta_t \mathbb{1}). \tag{1.21}$$

This process is applied incrementally to the input image. Since we can define the above equation such that it only depends on $\mathbf{x}_0$ we can immediately calculate an image representation $\mathbf{x}_t$ for any t (Ho, Jain, and Abbeel 2020). If we define $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$:

$$\begin{aligned}
\mathbf{x}_t &= \sqrt{\alpha_t}\,\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\,\mathbf{z}_{t-1} \\
&= \sqrt{\alpha_t \alpha_{t-1}}\,\mathbf{x}_{t-2} + \sqrt{(1-\alpha_t) + \alpha_t(1-\alpha_{t-1})}\,\bar{\mathbf{z}}_{t-2} \\
&= \sqrt{\alpha_t \alpha_{t-1} \alpha_{t-2}}\,\mathbf{x}_{t-3} + \sqrt{(1-\alpha_t \alpha_{t-1}) + \alpha_t \alpha_{t-1}(1-\alpha_{t-2})}\,\bar{\mathbf{z}}_{t-3} \\
&= \ldots \\
&= \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\mathbf{z},
\end{aligned} \tag{1.22}$$

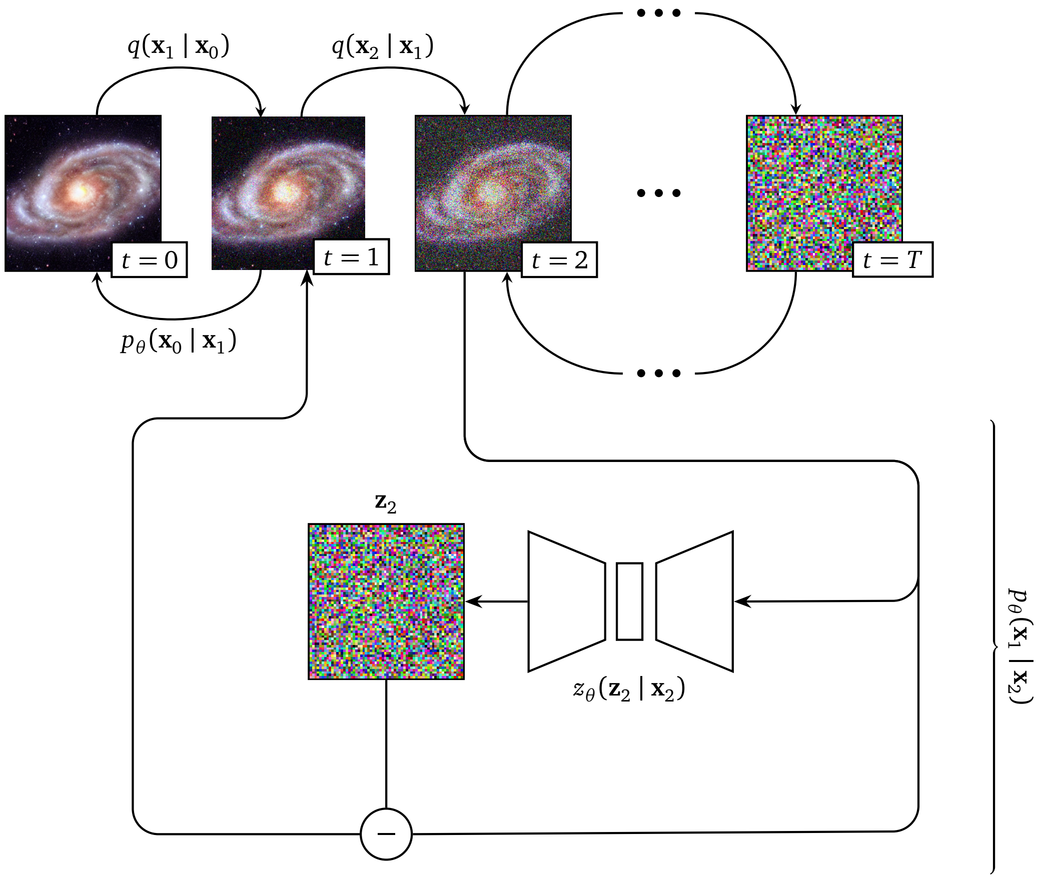

$q(\mathbf{x}_1 \mid \mathbf{x}_0)$
$q(\mathbf{x}_2 \mid \mathbf{x}_1)$
$t = 0$
$t = 1$
$t = 2$
$t = T$
$p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1)$
$\mathbf{z}_2$
$z_\theta(\mathbf{z}_2 \mid \mathbf{x}_2)$
$-$
$p_\theta(\mathbf{x}_1 \mid \mathbf{x}_2)$

Figure 1.19: It is easy (and achievable without learnt parameters) to add noise to an image, but more difficult to remove it. Diffusion models attempt to learn an iterative removal process via training an appropriate neural network, $p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$.

where $\mathbf{z}_t \sim \mathscr{N}(\mathbf{0}, \mathbb{1})$ and $\bar{\mathbf{z}}$ is a combination of Gaussians. Plugging the above expression into Eq. 4.1 removes the $\mathbf{x}_{t-1}$ dependency and yields

$$q(\mathbf{x}_t \mid \mathbf{x}_0) = \mathscr{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbb{1}).$$

Reverse process

Diffusion models attempt to reverse the forward process by applying a Markov chain with learnt Gaussian transitions. These transitions can be learnt via an appropriate neural network, p_θ:

$$p_\theta(\mathbf{x}_{0\ldots T}) = p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t),$$
$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) = \mathscr{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)).$$

While $\boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)$ can be learnt[24], the Ho, Jain, and Abbeel (2020) formulation fixes $\boldsymbol{\Sigma}_\theta$ to an iteration-dependent constant $\sigma_t^2\mathbb{1}$, where $\sigma_t^2 = 1 - \alpha_t$.

By recognising that Diffusion Models are a restricted class of Hierarchical VAE[25], we see that we can train p_θ by optimising the evidence lower bound (ELBO, introduced in Kingma and Welling 2013) that can be written as a summation over the Kullback-Leibler divergences at each iteration step[26]:

$$\mathscr{L}_{\text{ELBO}} = \mathbb{E}_q\Big[D_{\text{KL}}(q(\mathbf{x}_T \mid \mathbf{x}_0)||p(\mathbf{x}_T)) + \sum_{t>1} D_{\text{KL}}(q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0)||p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)) + \log p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1)\Big]. \quad (1.23)$$

In the Ho, Jain, and Abbeel (2020) formulation, the first term in Eq. 1.23 is a constant during training and the final term is modelled as an independent discrete decoder. This leaves the middle summation. Each summand can be written as

$$\mathscr{L}(\boldsymbol{\mu}_t, \boldsymbol{\mu}_\theta) = \frac{1}{2\sigma_t^2}||\boldsymbol{\mu}_t(\mathbf{x}_t, \mathbf{x}_0) - \boldsymbol{\mu}_\theta(\mathbf{x}_t, t)||^2, \quad (1.24)$$

where $\boldsymbol{\mu}_\theta$ is the neural network's estimation of the forward process posterior mean $\boldsymbol{\mu}_t$. In practice it would be preferable to predict the noise addition in each iteration

[24] See for example Nichol and Dhariwal (2021).

[25] Denoising autoencoders (§1.9.1) have an interesting relationship with score-based generative (or diffusion) models. As a taster, Turner (2021) reframe diffusion models as a class of hierarchical denoising VAE, and Dieleman (2022) show through a brief derivation that diffusion models optimise the same loss as a denoising autoencoder.

[26] See Appendix B in Sohl-Dickstein et al. (2015) and Appendix A in Ho, Jain, and Abbeel (2020) for the full derivation.

step ($\mathbf{z}_t$), as $\mathbf{z}_t$ has a distribution that by definition is centred about 0, with a well defined variance. To this end we can define $\boldsymbol{\mu}_\theta$ as

$$\boldsymbol{\mu}_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\mathbf{z}_\theta(\mathbf{x}_t, t)\right), \tag{1.25}$$

and by combining Eqs. 1.24 and 1.25 we get

$$\begin{aligned}\mathscr{L}(\mathbf{z}_t, \mathbf{z}_\theta) &= \frac{1}{2\sigma_t^2}\left\|\frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\mathbf{z}_t\right) - \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\mathbf{z}_\theta(\mathbf{x}_t, t)\right)\right\|^2 \\ &= \frac{(1-\alpha_t)^2}{2\sigma_t^2\alpha_t(1-\bar{\alpha}_t)}\|\mathbf{z}_t - \mathbf{z}_\theta(\mathbf{x}_t, t)\|^2. \end{aligned} \tag{1.26}$$

Ho, Jain, and Abbeel (2020) empirically found that a simplified version of the loss described in Eq. 1.26 results in better sample quality. They use a simplified version of Eq. 1.26 as their loss, and optimise to predict the noise required to reverse a forward process iteration step:

$$\mathscr{L}(\mathbf{z}_t, \mathbf{z}_\theta) = \|\mathbf{z}_t - \mathbf{z}_\theta(\mathbf{x}_t,\, t)\|^2, \quad \text{where } \mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\mathbf{z}_t. \tag{1.27}$$

By recognising that $\mathbf{z}_t = \sigma_t^2 \nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t \mid \mathbf{x}_{t-1})$, we see that Eq. 1.27 is equivalent to denoising score matching over t noise levels (Vincent 2011). This connection establishes a link between Diffusion Models and other SBGMs (such as Song and Ermon 2019; Jolicoeur-Martineau et al. 2020; Song and Ermon 2020).

To run inference for the reverse process, one progressively removes the predicted noise $\mathbf{z}_\theta$ from an image. The predicted noise is weighted according to a variance schedule:

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\mathbf{z}_\theta(\mathbf{x}_t, t)\right) + \boldsymbol{\sigma}_t\mathbf{z}.$$

If we take $p(\mathbf{x}_T) \sim \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbb{1})$, we can use p_θ to generate entirely novel data that are similar, but not identical to, those found in the training set.

Denoising diffusion implicit models

Ho, Jain, and Abbeel's diffusion model performs inference at a rate orders of magnitude slower than single shot generative models like the VAE (§1.9.1) or the GAN (§1.9.2). This is because diffusion models need to sequentially reverse every step in the forward process Markov Chain. Reducing the inference time for diffusion models is an active area of research (Jolicoeur-Martineau et al. 2021; Luhman and Luhman 2021; Watson et al. 2022), and here I will review one proposed solution to the problem; the Denoising Diffusion Implicit Model (DDIM; Song, Meng, and Ermon 2020).

Song, Meng, and Ermon (2020) propose the following decomposition of Eq. 1.22:

$$\begin{aligned}
\mathbf{x}_{t-1} &= \sqrt{\bar{\alpha}_{t-1}}\,\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_{t-1}}\,\mathbf{z}_\theta(\mathbf{x}_t, t) \\
&= \sqrt{\bar{\alpha}_{t-1}}\,\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\,\mathbf{z}_\theta + \sigma_t \mathbf{z}_t \\
&= \underbrace{\sqrt{\bar{\alpha}_{t-1}}\left(\frac{\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\,\mathbf{z}_\theta}{\sqrt{\bar{\alpha}_t}}\right)}_{\mathbf{x}_0\text{ prediction}} + \underbrace{\sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\,\mathbf{z}_\theta}_{\text{vector towards }\mathbf{x}_t} + \underbrace{\sigma_t \mathbf{z}_t}_{\text{noise}}.
\end{aligned}$$

Intuitively, the first term can be thought of as the prediction of the input image $\mathbf{x}_0$, given an iteration step t. The second term can be thought of as a vector from $\mathbf{x}_{t-1}$ towards the current iteration step image $\mathbf{x}_t$. The third term is random noise. If we substitute in $\mathbf{x}_t$ from Eq. 1.27 we make this intuition explicit:

$$\mathbf{x}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\,\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\,\frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0}{\sqrt{1-\bar{\alpha}_t}} + \sigma_t \mathbf{z}_t.$$

If we then set $\sigma_t = 0$, we remove the noise dependency and the forward process becomes deterministic:

$$q_{\mathrm{DDIM}}(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0) = \sqrt{\bar{\alpha}_{t-1}}\,\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_{t-1}}\,\frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0}{\sqrt{1-\bar{\alpha}_t}}. \tag{1.28}$$

This means that DDIMs can deterministically map to and from the latent space, and so inherit all the benefits of this property. For example, two objects sampled from similar latent vectors share high level properties, latent space arithmetic is possible, and we can perform meaningful interpolation within this space. I demonstrate DDIM latent space interpolation in Fig. 1.20.

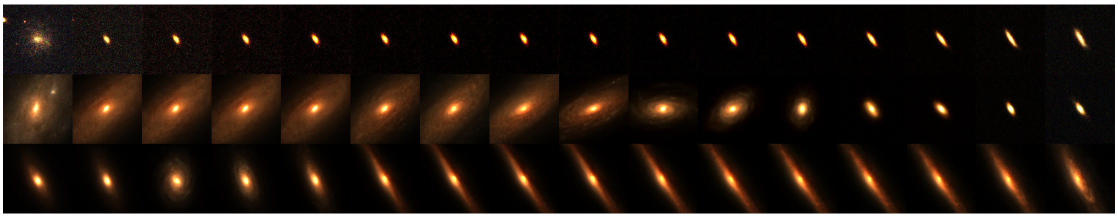


Figure 1.20: Meaningful latent space interpolation via a DDIM model (Song, Meng, and Ermon 2020; M. J. Smith et al. 2022). This property comes 'for free' with most other generative models, however the denoising diffusion probabilistic model (Ho, Jain, and Abbeel 2020) requires a tweak to its sampling scheme (Eq. 1.28).

We can also subsample every τ number of steps at inference time, where τ is a set of evenly spaced steps between 0 and T, the maximum number of steps in the forward process:

$$q_{\mathrm{DDIM}}(\mathbf{x}_{\tau_{i-1}} \mid \mathbf{x}_{\tau_i}, \mathbf{x}_0) = \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_{t-1}} \frac{\mathbf{x}_{\tau_i} - \sqrt{\bar{\alpha}_t}\mathbf{x}_0}{\sqrt{1-\bar{\alpha}_t}}. \tag{1.29}$$

As shown in Song, Meng, and Ermon (2020) this results in acceptable generations with a T/τ inference speed up.

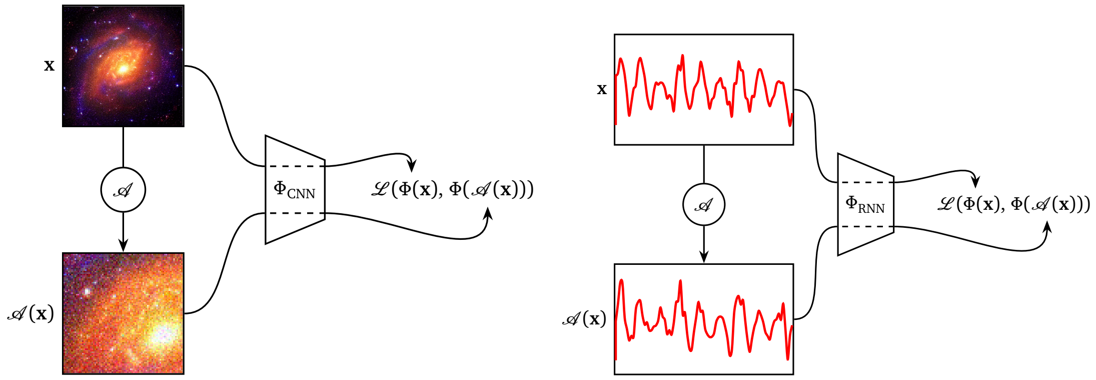
1.10 Representation learning

Self-supervised representation learning has recently exploded in popularity, with a slew of models being developed in rapid succession (e.g. He et al. 2019; T. Chen et al. 2020a,b; X. Chen et al. 2020; Grill et al. 2020). At its core, representation learning attempts to produce semantically meaningful compressed representations (or embeddings) of complex highly dimensional data. Aside from simply being a compression device, these embeddings can also be taken and used in downstream tasks, like clustering, anomaly detection, or classification.

In this section I will describe two approaches to representation learning that are popular within astronomy. The first approach uses contrastive learning as defined by the SimCLR model (T. Chen et al. 2020a,b). The second approach defines and uses a 'surrogate task' (such as autoencoding or next value prediction) to train a deep learning model, and extracts semantically meaningful representations from the subsequent trained network.

1.10.1 Contrastive learning

Fig. 1.21 describes a simple contrastive learning model similar to SimCLR (T. Chen et al. 2020a). This model takes as input a sample $\mathbf{x}$ from the training set, and augments it to produce $\mathscr{A}(\mathbf{x})$. This augmentation is performed in such a way that $\mathscr{A}(\mathbf{x})$ shares enough semantically meaningful data with $\mathbf{x}$ to belong to the same class. In the contrastive learning literature $(\mathbf{x}, \mathscr{A}(\mathbf{x}))$ is known as a positive pair. This positive pair is passed to a Siamese neural network Φ, which projects the high dimensional input data onto a lower dimensional 'embedding space'. All other training set samples are assumed to belong to a different class to $\mathbf{x}$, and so can be combined with $\mathbf{x}$ to produce 'negative pairs'. Once we produce some embeddings we need to define a loss that clusters similar samples together, while simultaneously pushing away dissimilar samples. Hadsell, Chopra, and LeCun (2006) propose such a loss—the maximum



(a) Possible application to imagery.

(b) Possible application to sequential data.

Figure 1.21: A simple contrastive learning model is applied to both imagery and sequential data. $\mathscr{A}$ is an augmentation pipeline. For imagery, $\mathscr{A}$ could consist of random crops, noise addition, and colour jitter. For sequential data, $\mathscr{A}$ could consist of noise addition, stochastic temporal shifting, and random data deletion. Φ is a function approximator that projects inputs onto an embedding space. Φ is typically a neural network: when processing imagery, Φ could take the form of a CNN, and when processing sequential data Φ could be an RNN. The loss $\mathscr{L}$ measures the distance between the embeddings $\Phi(\mathbf{x}) \equiv \mathbf{z}_i$ and $\Phi(\mathscr{A}(\mathbf{x})) \equiv \mathbf{z}_j$, and we train by attempting to minimise this distance while maximising the distance between dissimilar samples.

margin contrastive loss:

$$\mathscr{L}(\mathbf{z}_i, \mathbf{z}_j) = \delta_{ij}\, \mathbf{z}_i^T \mathbf{z}_j + (1 - \delta_{ij}) \max(0, m - \mathbf{z}_i^T \mathbf{z}_j),$$

where δ is the Kronecker delta, $\mathbf{z}_i$ and $\mathbf{z}_j$ are embedding vectors[27], and m is the margin. If $\mathbf{z}_i$ and $\mathbf{z}_j$ are a positive pair, the loss pulls the embeddings closer, and if they are a negative pair the loss pushes the embeddings away from each other. The margin imposes an upper distance bound on dissimilar embeddings.

While useful, the maximum margin contrastive loss does not take into account the embedding space beyond the pair it is attending to in each training step. This limitation ultimately results in a less expressive embedding space. The triplet loss (Chechik et al. 2010) solves this issue by taking into account the broader embedding space and simultaneously attracting a positive pair while repulsing a negative pair with each training step:

$$\mathscr{L}(\mathbf{z}_i, \mathbf{z}_j, \mathbf{z}_k) = \max(0,\ \mathbf{z}_i^T \mathbf{z}_j - \mathbf{z}_i^T \mathbf{z}_k + m) \tag{1.30}$$

where $\mathbf{z}_k$ is a sampled from a different class to $\mathbf{z}_i$, and $\mathbf{z}_j$ is sampled from the same class as $\mathbf{z}_i$.

[27] All embeddings in this subsection are normalised.

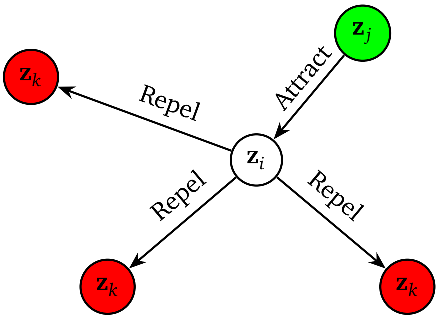


Figure 1.22: The triplet (Eq. 1.30) and NT-Xent (Eq. 1.31) losses simultaneously incentivise attraction between embeddings sampled from the same class ($\mathbf{z}_i$ and $\mathbf{z}_j$), and repulsion between embeddings sampled from different classes ($\mathbf{z}_i$ and $\mathbf{z}_k$).

If we study Eq. 1.30 we see that it is possible to generalise our loss even further, taking into account an arbitrary number of negative samples. The Normalized Temperature-scaled Cross Entropy loss (NT-Xent; Sohn 2016) does precisely this:

$$\mathscr{L}(\mathbf{z}_i, \mathbf{z}_j) = -\log\left(\frac{\exp(\mathbf{z}_i^T \mathbf{z}_j / \mathscr{T})}{\sum_{k=1}^{2N}(1-\delta_{ki})\exp(\mathbf{z}_i^T \mathbf{z}_k / \mathscr{T})}\right), \tag{1.31}$$

where $\mathbf{z}_i$ and $\mathbf{z}_j$ are a positive embedding pair, and $\mathbf{z}_i$ and $\mathbf{z}_k$ are a negative pair. $\mathscr{T}$ is a 'temperature' hyperparameter introduced in T. Chen et al. (2020a) to help the model learn from hard negatives (negatives closer to the anchor than the comparison positive, see Fig. 1.23).

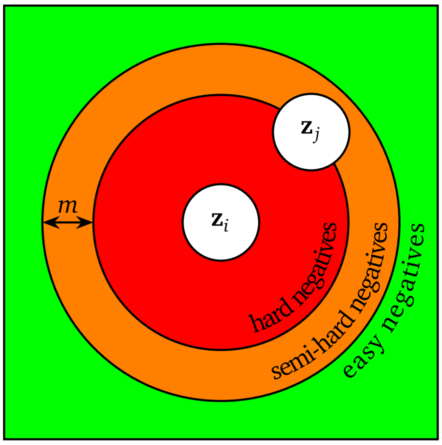


Figure 1.23: Types of negative embeddings. $\mathbf{z}_i$ and $\mathbf{z}_j$ form a positive embedding pair. If a negative is closer than the current positive it is considered a hard negative, if it lies within the margin it is considered a semi-hard negative, and if it is beyond the margin it is considered an easy negative.

1.10.2 Learning representations via a surrogate task

One can also learn representations via a surrogate task. A surrogate task is any task that is unrelated to the network's final use. However, in the process of learning to perform the surrogate task, the network learns what is important, and what is unimportant about data within the training set. This information can then be extracted in the form of learnt representations. If the surrogate task is general enough, these representations will contain useful semantic information about the items in the dataset, and can then be used for downstream applications (i.e. clustering, classification, anomaly detection).

Let us concretise this process by revisiting an example that we previously discussed in §1.5. Let us imagine we have a large set of galaxy rotation curves that we want to extract embeddings from. We could train an LSTM model (Fig. 1.24) on the task of predicting the next item in the rotation curve, with the model only having access to the previous items in the profile. Once the LSTM model is trained on this task, we can feed in a full, new rotation curve, and repurpose the final hidden state as a representative embedding. Note that this set up does not rely on any external labels, only on the rotation curve itself[28].

We can generate embeddings via an autoencoding task. Again, let us use an astronomical example to specify this, and say that we want to extract embeddings from a set of galaxy observations. We could repurpose a variational autoencoder for this, training it as normal as described in §1.9.1. However, once the model is trained we would discard the decoding part of the network, and only consider the encoder. To generate embeddings we would then simply pass in our galaxy images to the trained encoder. The same process can be carried out by a GAN (§1.9.2). In the GAN case, we would discard the generator after training, and use the discriminator's penultimate layer outputs as our embeddings.

Supervised networks can also be used to generate embeddings. If a network has been trained in a supervised manner to classify or regress data, it will have learnt some properties about that data that helps it to carry out its task. We can access these learnt representations by taking the outputs from a trained network's penultimate layer as an embedding[29].

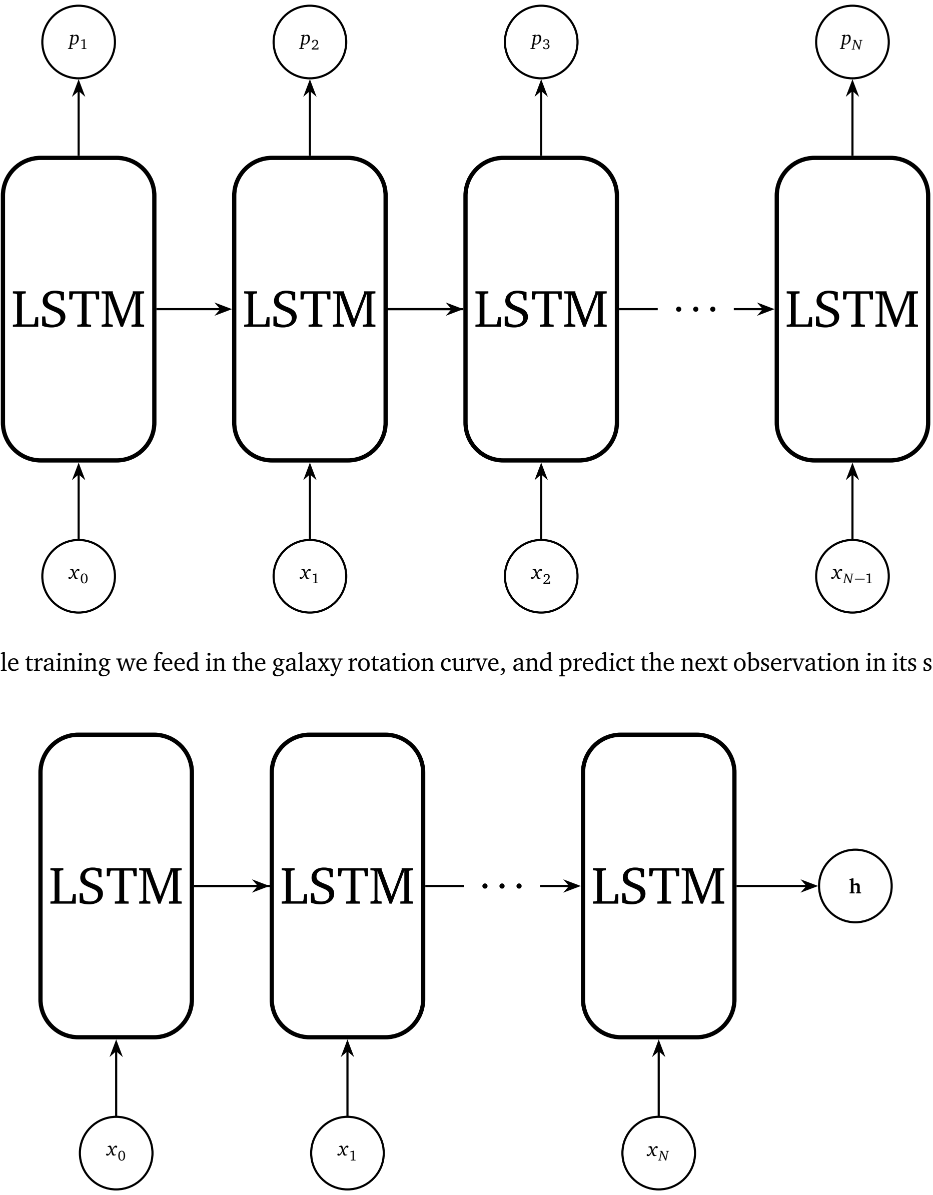
1.11 A third wave: non-supervised astronomical connectionism

Since its astronomical debut in the mid-2010s (Regier, McAuliffe, and Prabhat 2015)[30], deep generative modelling has become a popular subfield within astronomical connectionism. This popularity is driven by its inherent scalability; the lack

[28] This self-supervised training set up is similar to that used to train autoregressive foundation models. These models will be explored in detail in §5.1.

[29] Interestingly, this process is used in the calculation for the Fréchet Inception Distance (FID) (Heusel et al. 2017; Seitzer 2020). The FID acts as a measurement of the visual similarity between two datasets. The FID works by taking the penultimate layer representations from a trained Inception-v3 model (Szegedy et al. 2016) for each dataset and calculating the distance between them.

[30] Also compare its companion paper (Regier et al. 2015).



(a) While training we feed in the galaxy rotation curve, and predict the next observation in its sequence.

(b) While inferring we feed in the full galaxy rotation curve, and extract the LSTM hidden state as a compressed representation embedding of the curve. Otherwise, we ignore whatever output (i.e. $\{p_1, \ldots, p_N\}$) the LSTM generates.

Figure 1.24: A hypothetical surrogate task for extracting rotation curve representations is shown. $\{x_0, \ldots, x_N\}$ is a set of observations from a galaxy rotation curve, in order of radial distance from the galactic centre. $\{p_1, \ldots, p_N\}$ is the LSTM's corresponding set of predictions for $\{x_1, \ldots, x_N\}$. $\mathbf{h}$ is the LSTM hidden state vector. See Fig. 1.11 for more about the internal workings of the LSTM.

of a need for labelled data allows the methods to be repurposed for any dataset that might be at hand. Self-supervised connectionism has been around for longer, but again has recently exploded in popularity due to its usefulness in wrangling enormous unlabelled datasets. This section is split into two major parts; I will first outline the history of deep astronomical generative modelling in §1.11.1, and the history of contrastive representation learning will be discussed in §1.11.2. Although representation learning is the explicit goal for only the studies described in §1.11.2, it must be stressed that representations *can also be extracted from all the deep generative models described in §1.11.1*.

1.11.1 Deep astronomical generative modelling

Seminally, Regier, McAuliffe, and Prabhat (2015) proposed the use of a VAE to model galaxy observations. They trained their network on downscaled 69 × 69 crops of galaxies from a SDSS-sampled dataset containing 43 444 galaxies. They trained their network in the same way as described in §1.9.1, and find that the network is capable of generating galaxies similar to those found in the training set. They also find that their network produces semantically meaningful embeddings, noting that their galaxies are clustered by orientation and morphological type. This same line of enquiry was followed by Ravanbakhsh et al. (2016), who showed that VAEs could be used to generate galaxies conditionally. Ravanbakhsh et al. (2016) also pioneered the use of GANs to generate galaxy imagery. Spindler, Geach, and Smith (2020) used a VAE combined with a Gaussian mixture model prior (see Eq. 1.20 and accompanying text) to generate and cluster galaxy images into morphological types. The previous studies in this paragraph used low resolution observations in their training set. Fussell and Moews (2019) and Holzschuh et al. (2022) demonstrated that GANs are capable of generating large high fidelity galaxy observations. Fussell and Moews (2019) achieved this with a stacked GAN architecture (Zhou et al. 2016), and Holzschuh et al. (2022) use the related StyleGAN architecture (Karras, Laine, and Aila 2018) to the same end. Relatedly, M. J. Smith et al. (2022) use a diffusion model to generate large high fidelity galaxies. They trained their network on two datasets comprised of galaxies as observed by the Dark Energy Spectroscopic Instrument (DESI; Dey et al. 2019). One, a set of 306 006 galaxies catalogued in the SDSS Data Release 7 (York et al. 2000; Abazajian et al. 2009; Wilman, Zibetti, and Budavári 2010), and the other a set of 1962 late-type galaxies, as catalogued in the Photometry and Rotation curve OBservations from Extragalatic Surveys (PROBES; Stone and Courteau 2019) dataset. PROBES contains well resolved galaxies that exhibit spiral arms, bars, and other features characteristic of late-type galaxies. They found that their model produces galaxies both qualitatively and physically indistinguishable from those in the training set, proving that diffusion models are a competitive alternative to the more established GAN and VAE models for astronomical simulation. M. J. Smith et al. (2022) is presented in full in §4. From all of these studies we can conclude that deep generative models can internalise a model capable of physically and morphologically describing galaxies.

Generative models have also been used to simulate data in areas beyond galaxies. In a use-case tangential to galaxy generation, Smith and Geach (2019) show that a Spatial-GAN (Jetchev, Bergmann, and Vollgraf 2016) can simulate wide field surveys. They train on the Hubble eXtreme Deep Field, and find that galaxies generated within their model's synthetic deep fields are statistically indistinguishable from the real thing. Smith and Geach (2019) is presented in full in §3. Cosmological simulations have also been explored, with Rodriguez et al. (2018) using a GAN to generate cosmic web simulations at pace, and Mustafa et al. (2019a) generating weak lensing convergence maps at a pace faster than classic simulations. Beyond GANs, Remy et al. (2020)[31] trained a SBGM on simulated maps from MassiveNus (J. Liu et al. 2018), and found that their model was capable of replicating these maps. They also demonstrated that their model was capable of producing a likely spread in the posterior predictions. Finally, they demonstrate that a SBGM is capable of predicting the dark matter map of the real Hubble Cosmic Evolution Survey (COSMOS) field (Scoville et al. 2007), finding that their SBGM can produce an extremely high quality dark matter map.

The image domain translation abilities of GANs in a Pix2Pix-like formulation (Isola et al. 2016, see Fig. 1.18b.) is particularly useful in astronomy. Schawinski et al. (2017) demonstrated this use first by training a Pix2Pix-like model to denoise astronomical data. They trained their network on 4550 galaxies sampled from the SDSS. The galaxies were convolved to increase the seeing, and speckle noise was added. The GAN was tasked with reversing this process. They found that their method outperformed both blind deconvolution, and Lucy-Richardson deconvolution. Generative models are also capable of separating sources, as Stark et al. (2018) demonstrate by using a Pix2Pix model to deblend quasar point source light and that of their host galaxy. Reiman and Göhre (2019) use a similar model to Stark et al. (2018) to deblend overlapping galaxies.

At the time of writing there are only three examples of score-based modelling in the astronomy literature (Remy et al. 2020, 2022; M. J. Smith et al. 2022). It is surprising that these studies are the only examples of score-based modelling in

[31] This preliminary work has been subsequently extended in Remy et al. (2022).

astronomy, as SBGMs produce generations that rival that of state-of-the-art GAN models, without drawbacks present in other models (like blurring in the case of VAEs, or mode collapse and training instability in the case of GANs). SBGMs also have some natural uses in astronomical data pipelines. For example, an implementation similar to Sasaki, Willcocks, and Breckon (2021) could be used for survey-to-survey photometry translation similarly to Buncher, Sharma, and Carrasco Kind (2021). The source image separation model described in Jayaram and Thickstun (2020) has the obvious application as an astronomical object deblender (i.e. Stark et al. 2018; Reiman and Göhre 2019; Arcelin et al. 2021). SBGMs are ripe for exploitation by the astronomical community, and I hope to see much interest in this area in the coming years.

1.11.2 Self-supervised astronomical representation learning

Very recently there has been some work produced applying self-supervised contrastive learning models to galaxy image clustering. Hayat et al. (2021) trained SimCLR (T. Chen et al. 2020a) on multi-band galaxy photometry from the SDSS (York et al. 2000). They show that the resulting embeddings capture useful information by directly using them in a training set for a galaxy morphology classification model, and a redshift estimation model. Similarly, Sarmiento et al. (2021) trained SimCLR on integral field spectroscopy data captured from galaxies in the Mapping Nearby Galaxies at Apache Point Observatory survey (MaNGA; Bundy et al. 2014). Again, they find that SimCLR produces semantically meaningful embeddings. These studies show that contrastive learning is applicable to galaxy imagery, but questions remain about its effectiveness regarding other types of astronomical data, such as time series. I am currently co-leading a study that will attempt to address this gap in the literature (Miller, M. J. Smith et al. in prep.).

1.12 Deep learning for large astronomical datasets

The following three chapters describe three models that leverage recent developments in deep learning, and astronomy's recent abundance of data. First, §2 demonstrates that it is possible to perform a classical semi-manual astronomical information extraction pipeline via a Seq2Seq style neural network (§1.7). §3 and §4 describe two generative models that have been repurposed as astronomical simulators. §3 uses a generative adversarial network (§1.9.2) to simulate the Hubble eXtreme Deep Field. Meanwhile, §4 uses score-based modelling (§1.9.3) to generate hyper-realistic galaxy observations. All of these chapters have been published previously in the Monthly

Notices of the Royal Astronomical Society, and the contents of each chapter have been kept as similar to their published counterparts as possible. In §5 I draw this thesis to an end by extrapolating astronomy's future from other applied deep fields, and outlining a coming 'fourth wave' of astronomical connectionism.

Chapter 2

Pix2Prof: fast extraction of sequences from galaxy imagery via a deep natural language ‘captioning’ model

We present ‘Pix2Prof’, a deep learning model that can eliminate any manual steps taken when extracting galaxy profiles. We argue that a galaxy profile of any sort is conceptually similar to a natural language image caption. This idea allows us to leverage image captioning methods from the field of natural language processing, and so we design Pix2Prof as a float sequence ‘captioning’ model suitable for galaxy profile inference. We demonstrate the technique by approximating a galaxy surface brightness (SB) profile fitting method that contains several manual steps. Pix2Prof processes ∼1 image per second on an Intel Xeon E5 2650 v3 CPU, improving on the speed of the manual interactive method by more than two orders of magnitude. Crucially, Pix2Prof requires no manual interaction, and since galaxy profile estimation is an embarrassingly parallel problem, we can further increase the throughput by running many Pix2Prof instances simultaneously. In perspective, Pix2Prof would take under an hour to infer profiles for 100 000 galaxies on a single NVIDIA DGX-2 system. A single human expert would take approximately two years to complete the same task. Automated methodology such as this will accelerate the analysis of the next generation of large area sky surveys expected to yield hundreds of millions of targets. In such instances, all manual approaches – even those involving a large number of experts – will be impractical.

This chapter has been previously published as M. J. Smith et al. (2021). ‘Pix2Prof: fast extraction of sequential information from galaxy imagery via a deep natural language ’captioning’ model’. In: *Monthly Notices of the Royal Astronomical Society* 503.1, pp. 96–105. DOI: 10.1093/mnras/stab424. arXiv: 2010.00622 [astro-ph.IM]. It has been presented as as a contributed talk at *IAP-ML 2021*, Paris, France.

2.1 Introduction

Large astrophysical surveys such as the Sloan Digital Sky Survey (SDSS; York et al. 2000), the Panoramic Survey Telescope and Rapid Response System (Pan-STARRS; Chambers et al. 2016), the Hyper Suprime Cam (HSC; Aihara et al. 2017) Subaru Strategic Program Survey, or the upcoming Vera C. Rubin Observatory Legacy Survey of Space and Time (LSST; Ivezić et al. 2019), carry an inherent scaling problem. The SDSS has observed over 35 per cent of the sky, cataloguing over 1 billion astronomical objects (Ahumada et al. 2019) with a data rate of raw multiband imagery approaching 200 GB per night. In January 2019, Pan-STARRS second Data Release totalled 1.6 PB of imaging data. A precursor to LSST, HSC's 990 megapixel camera has already produced over 1 PB of imaging data (Aihara et al. 2019). These surveys will be dwarfed by the upcoming LSST project. LSST's 3.2 gigapixel camera will be the largest ever made, and will survey the entire visible sky twice per week, generating ~500 PB of imaging data over its decade-long mission. The 'firehose' of data from surveys such as LSST will require correspondingly efficient and fully automated procedures to curate and analyse the data, enabling new astrophysical findings and making unanticipated discoveries.

In this study, we are concerned with the automated direct analysis of galaxy imagery towards estimating galaxy properties such as size, luminosity, colour, and stellar mass. To calculate these properties, one typically applies a photometric analysis that involves extracting and characterising the spatial distribution of a galaxy's light, described by a surface brightness (SB) profile. The galaxy structural parameters as reflected by the SB profile can be used to infer a suite of other important characteristics such as light concentration, age, star formation rate, and assembly history (e.g. Strom et al. 1976; Bell et al. 2003; S. Shen et al. 2003; Bernardi et al. 2005; Fernández Lorenzo et al. 2013; Trujillo, Chamba, and Knapen 2020).

Numerous catalogues of galaxy structural properties already exist (Jedrzejewski 1987; Courteau 1996; Brinchmann et al. 2004; Blanton et al. 2005; Hall et al. 2012; Gilhuly and Courteau 2018). Unfortunately, the methods used in these compilations are either time consuming, requiring human supervision, or fast but unreliable since they require *a priori* assumptions about the shapes of galaxy components and other features. Similarly to the study detailed in this chapter, Tuccillo et al. (2018) describe a fully automated neural network based technique (named 'DeepLeGATo'; Deep

Learning Galaxy Analysis Tool) designed to replicate the GALFIT model-dependent algorithm. DeepLeGATo is a fine example of an effective application of deep learning, providing faster and possibly more accurate analysis than its parent method, GALFIT (Peng et al. 2002). However, DeepLeGATo inherits from its similarity to GALFIT the same issues stated previously; namely, the hard-coded assumption of galaxy profile shapes, and other features. Furthermore, DeepLeGATo can only produce single float outputs, and so cannot infer an SB profile directly. This means that DeepLeGATo must rely on an intermediary model to generate a complete SB profile. Therefore, even with semi-automated methods, the accurate extraction of *all* the useful information from existing surveys would take years. With the data volume expected to grow significantly in the coming years, this becomes an intractable problem. Of great concern is the possibility that important discoveries and insight could be missed or delayed significantly due to the technical challenges imposed by the unprecedented data volume. Clearly, there is a pressing need for entirely new and efficient automated methods that significantly reduce, and ideally circumvent, human interactions. Machine learning is ideally suited for this task, and we apply it in this chapter towards the specific problem of extracting SB profiles from multi-band imaging data. Our approach takes advantage of a set of SB profiles already determined via classical, interactive methods (Courteau 1996; Gilhuly and Courteau 2018). We describe the classical method used to produce the training data set in the next section. The remainder of the chapter is organised as follows: Section 2.2 introduces our approach; our results and validation are presented in Section 2.3; Section 2.4 addresses our global findings, and concludes with suggestions for broader application of the algorithm.

2.2 Method

2.2.1 The classical surface brightness profile extraction algorithm

In the surface photometry of galaxies (e.g. Courteau 1996, and references therein), the spatially resolved light profile of a galaxy is extracted by fitting progressively larger isophotes about a common centre. The fitting technique assumes that projected isophotes are well represented by ellipses. A galaxy's centre is found by identifying the brightest pixel in a manually selected region. Given a manually defined galaxy centre, the classical algorithm determines the parameters needed to define each ellipse via a least squares optimisation. The algorithm then generates isophotal solutions at each radius well into the faint outskirts of the galaxy. In these regions of lower signal-to-noise, where fitting algorithms are challenged, the algorithm radially extends the last fitted isophote in the previous operation with a set of concentric isophotes out to an arbitrarily large radius, usually taken to be the edge of the image.

The isophotes may vary in ellipticity and position angle as a function of galactocentric radius. This can become problematic when fitting to non-axisymmetric structures in galaxies, such as bars and spiral arms that can cause large twists in the fitted isophotal map. This issue can be corrected by manually applying a smoothing function to some portions of the image. The latter consists of manually smoothing the contour fits (i.e. uncrossing twisted isophotes), and replacing poorly fitted data with a polynomial smoothing function. Note that, prior to applying isophotal fitting to galaxy images, some pre-processing is also required: the galaxy centre must be identified as described above; the 'sky' background must be estimated and removed from the image; nuisance foreground objects (such as unassociated galaxies or foreground stars) must be identified and masked. These steps add to the manual supervision of the task.

Besides the assumption that galaxies are circular when viewed face-on, and thus generally of elliptical appearance when projected on to the plane of the sky, the algorithm purposefully avoids using *a priori* knowledge of galactic disc profile shapes and other features such as bars, rings, and spiral arms. This avoids biasing the isophotal solution to any pre-determined, and possibly incorrect, shape – a problem especially acute in the faint outer edges of a galaxy.

Table 2.1: A summary of the Courteau (1996) surface brightness profile fitting algorithm's processes. An approximate wall time per galaxy is given for the manual sections. The automated sections' time contributions are negligible.

Process	Automated?	Wall time (s/gal)
Identify galaxy centre	No	5
Estimate & remove sky background	Yes	–
Remove foreground objects	No	120
Fit contours	Yes	–
Extend contours to galaxy extent	No	30
Smooth isophotes	Yes	–
Interpolate poorly fitted data	No	120

While the semi-automated steps outlined above yield high quality SB profiles, the process of obtaining a single profile is slow and systematic variations may exist between different profile extraction methods. The interactive nature of certain steps may indeed give rise to marked profile differences, especially in low SB regimes where the isophotal solutions are less robust. The low SB regimes will always remain the bane of galaxy image analysis, whether automated or interactive, but the elimination of subjective steps goes a long way towards reducing systematic differences between profiles. Therefore, it becomes desirable to eliminate all interactive steps while retaining all the benefits of classic algorithms such as Courteau (1996) described above. In this chapter, we present a fully automated solution that incorporates the extant knowledge base of SB profile fitting methodology, but avoids human interaction.

2.2.2 Borrowing from automated image captioning

In recent years, the field of automated image captioning has benefited greatly from advances in deep learning. We were strongly influenced by these developments when designing the architecture of our 'Pix2Prof' profile estimator. In this section, we briefly review pure recurrent neural network (RNN) based encoder–decoder architectures, or models that only use a single encoder and decoder to generate captions. A comprehensive review on deep learning methods applied to image captioning can be found in Hossain et al. (2018).

We primarily draw inspiration from gated RNN based encoder–decoder architectures, as seen in Sutskever, Vinyals, and Le (2014) for sequence-to-sequence translation, and in Vinyals et al. (2014), Jia et al. (2015), and Wang et al. (2016) for image-to-sequence translation. Sutskever, Vinyals, and Le (2014) uses an LSTM (Hochreiter and Schmidhuber 1997) network to encode a given sentence to a latent descriptive vector, and a second LSTM network to decode the descriptive vector into a different feature space. One can use this technique to translate text between two different languages, for example.

Vinyals et al. (2014), Jia et al. (2015), and Wang et al. (2016) all use a convolutional neural network (CNN; Fukushima 1980; LeCun et al. 1989) to first encode an image to a latent descriptive vector, and then use an LSTM network to decode this vector into a text description (caption) of a given image. Xu et al. (2015) use a CNN encoder, and an LSTM that attends over the CNN output. Attention allows this approach to link each word in the caption with an associated part of the image. This approach works well for images that are crowded with multiple objects, but a simpler approach is preferred for our case where each image is dominated by a single, central galaxy.

A galaxy profile can be thought of as analogous to a text caption describing that galaxy. Both a text caption and galaxy profile can be encoded as a list of floats or integers, and both have a length and content dependent on the context of the conditioning image. Both also need to terminate once a complete sentence or profile

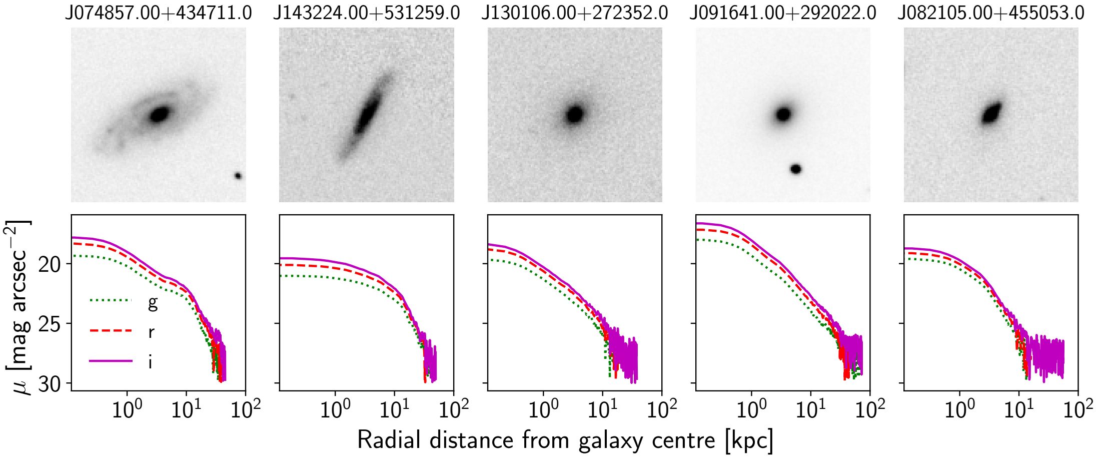


Figure 2.1: SDSS images of sample galaxies in the g band (top row), and corresponding surface brightness 'ground truth' profiles (bottom row). μ is the surface brightness. The galaxy names above each image refer to their J2000 celestial coordinate.

is generated, a subjective task well suited to a machine learning solution. RNNs learn where to terminate a given caption or profile empirically from the training set. Additionally, galaxy profiles and text captions can both be approximated with an appropriate RNN. RNNs also produce spatially consistent captions as a consequence of the architecture, a property not guaranteed in a straight one-shot CNN. With these benefits in mind, it is natural to consider an encoder–decoder network for the specific task of estimating challenging galaxy profiles. Importantly, since the proposed model directly learns the transformation between an unprocessed galaxy image, and the galaxy's corresponding SB profile, it eliminates all of the manual steps described in Section 2.2.1 and Table 2.1.

While we develop Pix2Prof within the context of galaxy SB profile extraction, the model is equally applicable to any array → float sequence translation task.

2.2.3 Training set

We initially populate our data set with 10 arcmin × 10 arcmin g, r, and i band image crops extracted via the SDSS DR10 (York et al. 2000; Eisenstein et al. 2011; Ahn et al. 2014) online mosaic interface. Each image is centred on a galaxy. From these images we extract an SB profile via the method described in Section 2.2.1. Fig. 2.1 presents a random sample of training set galaxies, and their corresponding, manually extracted SB profiles. The 1953 galaxy image–SB profile pairs in each of the g, r, and i bands yield a total of 5859 pairs. This full data set is then divided into training, validation, and testing sets. There are 5367 galaxy image–profile pairs in the training set, 192 galaxy image–profile pairs in the validation set, and 300 galaxy image–profile pairs in the test set. The sets are randomly assigned, with the condition that a given galaxy's three photometric bands are kept within the same set. The subset sizes are chosen to maximise the training set efficacy while retaining most of the training set variance in the test set.

The only destructive pre-processing performed on the galaxy imagery is a 99.9th percentile clipping. This clipping mitigates the issue of single bright (i.e. 'hot') pixels reducing image contrast when the galaxy images are normalised, which would reduce training efficacy. To this end, we apply a fixed min-max normalisation, defined as

$$\bar{x} = \frac{x - A}{B - A}, \tag{2.1}$$

where $A = 2.0$ nanomaggies is the floor of the minimum value in the training set, and $B = 30.0$ nanomaggies is the ceiling of the 99.9th percentile value in the training set.

To reduce information sparsity in the training set images, we further crop each galaxy image to a shape of [256, 256] pixels before passing the image to Pix2Prof. We train using the full 32-bit depth of the original data as measured. Good quality data are paramount when training a neural network, and we therefore cut the profiles when the signal-to-noise ratio reaches a quality threshold. We terminate the profile when the signal-to-noise ratio of a 1D convolution with a length of 40 pixels ($= 8$ arcsec) reaches a threshold of 4. We define signal-to-noise so that it is equivalent to the ratio of the power of a signal to the power of background noise: $(\mu/\sigma)^2$, where μ is the mean and σ is the standard deviation of the convolutional window.

2.2.4 Network architecture

We write our model in PyTorch (Paszke et al. 2019), using a ResNet-18 (He et al. 2015; Srivastava, Greff, and Schmidhuber 2015) encoder, and a GRU (Cho et al. 2014) decoder architecture. This architecture takes an arbitrarily sized single channel image input, and outputs a sequence of floats of arbitrary length. These floats are spaced along a galaxy's semimajor axis at a spacing of 0.2 arcsec per step (the same as the Courteau (1996) technique). The same network is trained on images in the g, r, and i bands, and therefore can produce a SB profile in any one of these bands. Fig. 2.2 shows a representation of the architecture used.

We use the standard ResNet-18 architecture as described in He et al. (2015). The GRU is stacked to three layers. We apply a rectified linear unit (Nair and Hinton 2010, ReLU) activation and a dense neural layer after the three layer stack to reduce the number of output values to one. As a regularising measure, we apply dropout at

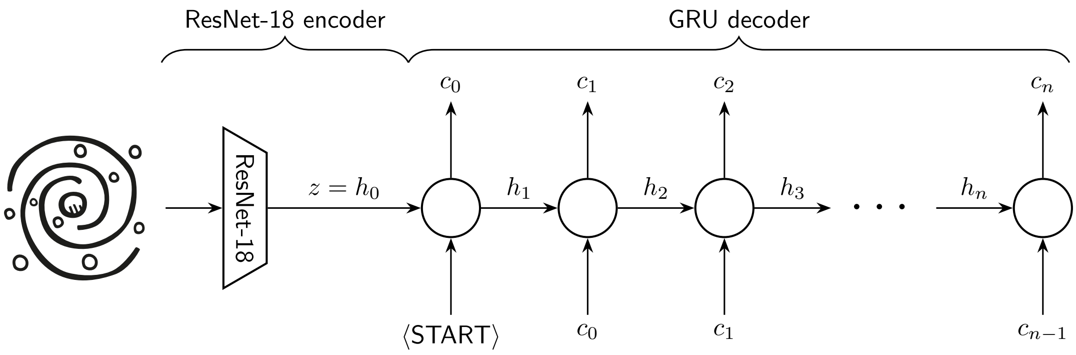


Figure 2.2: The ResNet → GRU encoder–decoder architecture used in this chapter. The hidden state h_i is the internal state of the GRU, and is dependent on both the galaxy latent encoding z, and the previous profile predictions c_i.

a 20 per cent rate (Srivastava et al. 2014a). The ResNet first encodes the incoming galaxy image to a latent space vector z of length 512. This vector is then used as the initial hidden state h_0 of a GRU. In this way, Pix2Prof encodes and passes relevant information from the image to the GRU. The GRU then unrolls to estimate properties of the galaxy from z. In this chapter's case, we demonstrate this process by using z to estimate a galaxy's SB profile.

To start estimation, the GRU is fed a start of sequence token. This token is set as an array of zeros. In place of an end of sequence token, the GRU is programmed to halt after 100 predictions are output that have a standard deviation of 0.01 or less. This ensures that the GRU halts estimation once it encounters the background sky.

We use the Adam optimiser (Kingma and Ba 2015) to train Pix2Prof via gradient descent (Robbins and Monro 1951). Using manual search, we set the learning rate as 2×10^{-4}. Due to the logarithmic nature of magnitude, we want to penalise large deviations from our ground truth SB profiles at a higher rate compared to small deviations, and so we use the mean squared error loss:

$$\mathrm{MSE} = \frac{1}{b}\sum_{i=1}^{b}(y_i - p_i)^2, \tag{2.2}$$

where b is the batch size, y is the ground truth, and p is a prediction.

2.2.5 Training the model

We augment the galaxy images by applying a 'wobble'. This wobble is a random small shift in the centre of the image. Each band is treated independently. We do this to encourage the network to work with the slightly off-centre galaxies that will be encountered in real data. This is required to make Pix2Prof robust to poorly centred galaxy images. We also exploit the rotational axisymmetry of galaxies and further augment the data by randomly rotating an input image through 90, 180, and 270 degrees.

We train the model for 100 000 global steps on a single NVIDIA TESLA V100 GPU. Training takes approximately 20 minutes per epoch of 500 galaxy images, a rate of 0.4 galaxies per second.

2.3 Results and validation

We validate the model during training once per epoch using the validation set. We test the trained model on 100 randomly sampled observed galaxies in the g, r, and i bands (for 300 total image–profile pairs) drawn from the data set and which are set aside entirely during training. We use the model with the lowest validation loss; epoch 160. We run an entirely automated inference on an Intel Xeon CPU E5-2650 v3 CPU at a rate of 0.9 galaxies per second.

Fig. 2.3 shows a random selection of 25 Pix2Prof inferred test set SB profiles superimposed on to the Courteau (1996) SB profiles. Since we have trained Pix2Prof to directly infer a SB profile from a galaxy image we do not produce intermediate steps (such as the galaxy centre, ellipticity profile, or position angle profile). However, one could estimate these values if Pix2Prof is explicitly trained to reproduce them. Fig. 2.4 shows the error distribution of the test set as well as the test set error per distance in physical units from the galaxy centre. We define error as the absolute of Fig. 2.3's residual, the absolute deviation:

$$\eta = \left| y - p \right|, \tag{2.3}$$

where p is a prediction, and y is measured via Courteau (1996). The units of SB call for additional care in defining our errors. Since SB values are defined on a logarithmic scale, equation (2.3) is really a form of fractional error:

$$\eta = \left| 2.5 \log_{10} \frac{I_p}{C} - 2.5 \log_{10} \frac{I_y}{C} \right| = \left| -2.5 \log_{10} \frac{I_y}{I_p} \right|, \tag{2.4}$$

where C is a constant reference brightness. $\{I_p, I_y\}$ are brightnesses in linear units.

We take the median of this error per galaxy profile to produce the violin plots in Fig. 2.4a, and we take the median of this error across profiles to produce the line plot in Fig. 2.4a. Fig. 2.4a's line plot shows that the error increases with radius away from the galaxy centre towards regions containing less signal, as expected. We find that the

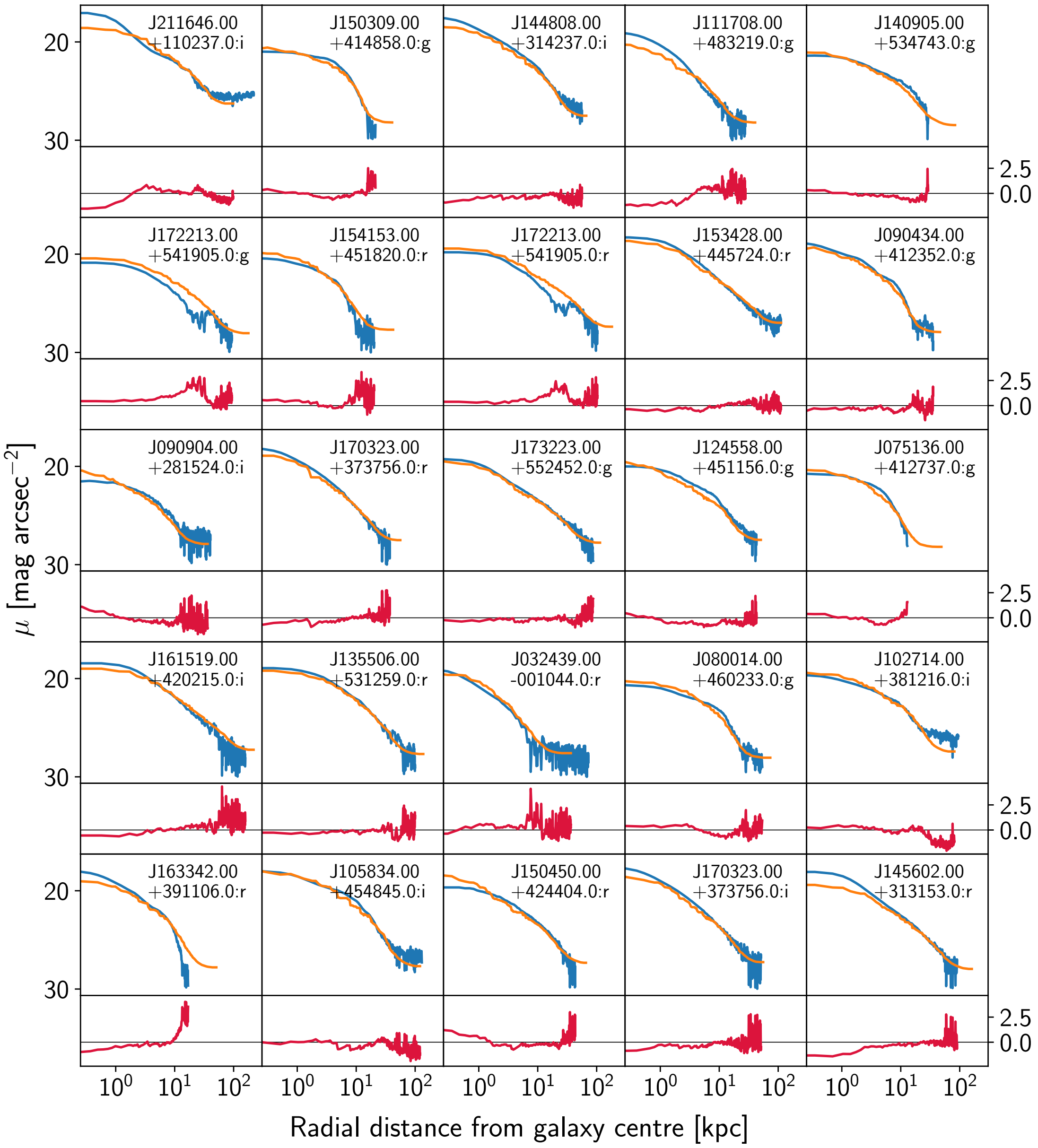


Figure 2.3: Randomly sampled test set predicted SB galaxy profiles (orange) superimposed onto SB profiles measured via the Courteau (1996) method (blue). μ is the surface brightness. Distances from centre are in log scale to emphasise divergences in the high signal-to-noise ratio region closer to the galaxies' centres. Below each SB profile plot is the residual defined as res $= y - p$, where y is the profile as measured according to Section 2.2.1, and p is the prediction. The galaxies' J2000 celestial coordinates and spectral bands are indicated at the top right of each graph.

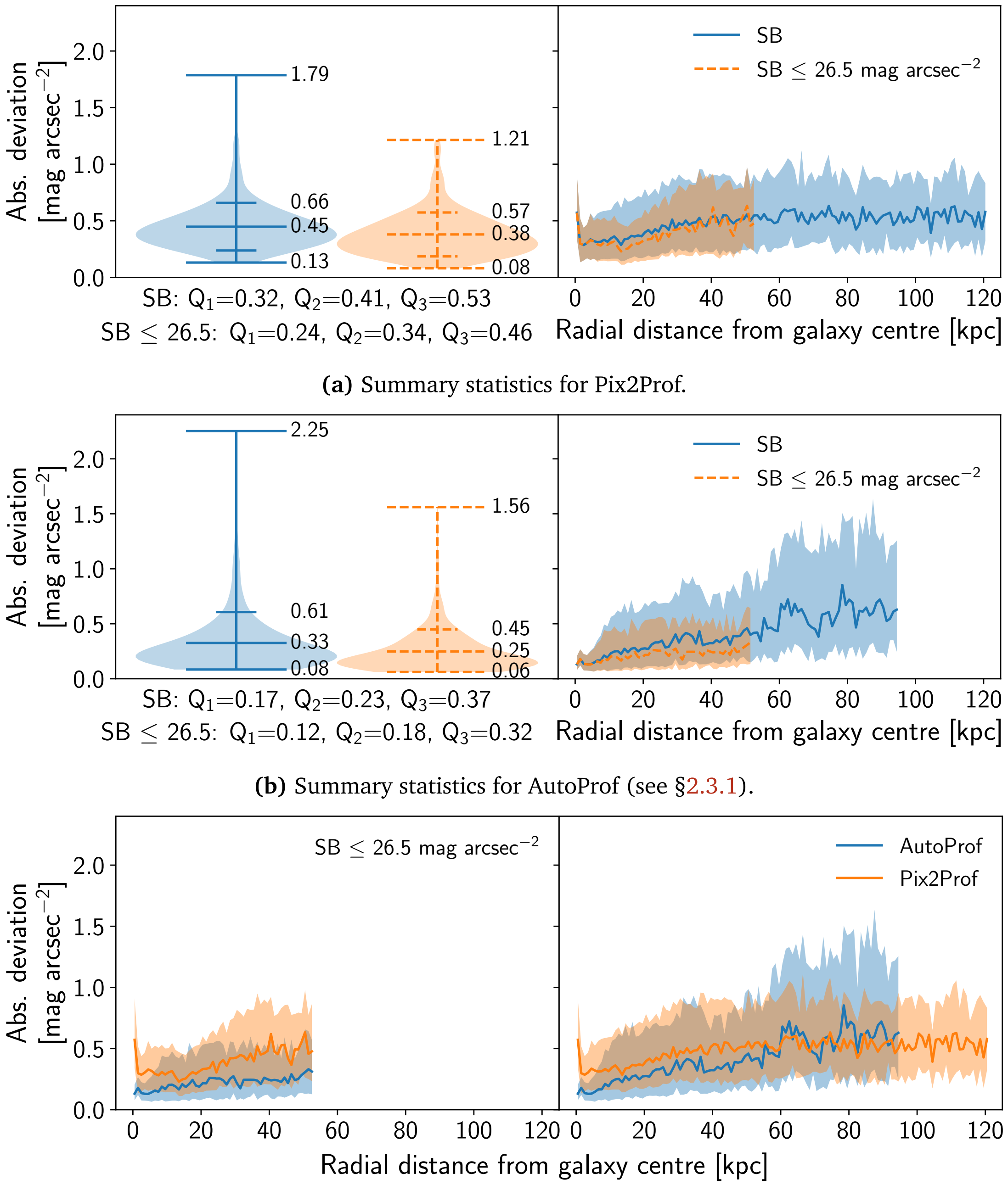
median test set absolute deviation is 0.41 mag arcsec^{-2} with an interquartile range of 0.21 mag arcsec^{-2}. We also find that the median test set absolute deviation for y values brighter than the SDSS limiting SB (26.5 mag arcsec^{-2}) is 0.34 mag arcsec^{-2}, with an interquartile range of 0.22 mag arcsec^{-2}. Errors of this scale mean that profiles generated via Pix2Prof will be immediately useful for rough searches; it would be possible to categorise galaxies roughly by brightness, isophotal radius, scale length, and other structural parameters. Further refinement of the model may reduce error, enabling more sophisticated processing and analysis of generated SB profiles. Possible refinements are described in Section 2.4.

In Fig. 2.5, the three bands' median errors are separated as a function of galactocentric radius. Close to the galaxy, there is little difference in the three bands' median predictions. However, as we proceed outwards, the r-band's error is higher than the g-band's, and the i-band's error is higher still. This is due to a difference in the instrumental noise between the three bands, as evidenced in the difference in the spectral bands' median galaxy image signal-to-noise ratios: $\mathrm{SNR}_g = 41.6$; $\mathrm{SNR}_r = 35.8$; $\mathrm{SNR}_i = 28.4$.

Fig. 2.6a shows each test set galaxy's ellipticity against the galaxy profile's median absolute deviation. The ellipticity is defined as the final isophote ellipticity for a profile calculated using the Courteau (1996) method. Fig. 2.6b shows each test set galaxy's semimajor axis radius for the first isophote whose value is greater than or equal to 23.5 mag arcsec^{-2}. In both of these cases, we run a linear regression and find no significant correlation, suggesting that Pix2Prof's predictions are equally robust when inferring across galaxies with a range of sizes and ellipticities.

Figs 2.3 and 2.4 show that Pix2Prof can successfully approximate a complicated astrophysical image processing pipeline with low deviation (0.34 mag arcsec^{-2} averaged over the test set). Processing 0.9 galaxies per second on an Intel Xeon E5-2650 v3 CPU, Pix2Prof improves on the speed of the classical image analysis method of Courteau (1996) by more than two orders of magnitude. For comparison, an astronomer trained to use the Courteau (1996) method can typically process $\sim$150 galaxies in a full eight hour working day (or $\sim$0.005 galaxies per second). However, even astronomers must rest and so the true working rate for a human would be $\sim$150 galaxies per 24 hours, or $\sim$0.002 galaxies per second.

As Table 2.2 shows, Pix2Prof eliminates any manual interaction from SB profile inference, alleviating the issue of subjectivity in the different methods developed for such tasks; Pix2Prof will infer the same profile every time for a given galaxy image, whereas a human may not. The full automation of Pix2Prof enables a complete parallelisation, and thus significant gain in parallel throughput of galaxy profile



(a) Summary statistics for Pix2Prof.

(b) Summary statistics for AutoProf (see §2.3.1).

(c) Median error per kpc from galaxy centre comparison between Pix2Prof and AutoProf.

Figure 2.4: Approximation errors as defined in equation (2.3). Fig. 2.4a shows summary statistics for Pix2Prof. For comparison, Fig. 2.4b depicts the same statistics for AutoProf. In all of the above images/pix2prof we define absolute deviation as relative to the Courteau (1996) test set profiles. The leftmost violin plot in Fig. 2.4a and 2.4b shows the distribution of median test set errors. The rightmost violin plot in Fig. 2.4a and 2.4b shows the same distribution for only SB values below the SDSS limiting SB of 26.5 mag arcsec^{-2}. The maximum, minimum, mean, and (mean + standard deviation) are labelled. Below the violin plots are their distribution quartiles. The line plots show the median error per kpc from the galaxy centre, with the interquartile range shaded. To reduce the effect of small sample size variability, the line plots are terminated once 90 per cent of the SB profiles reach their galaxy's extent. Fig. 2.4c compares on the same axes the median errors per kpc from the galaxy centre for Pix2Prof and AutoProf.

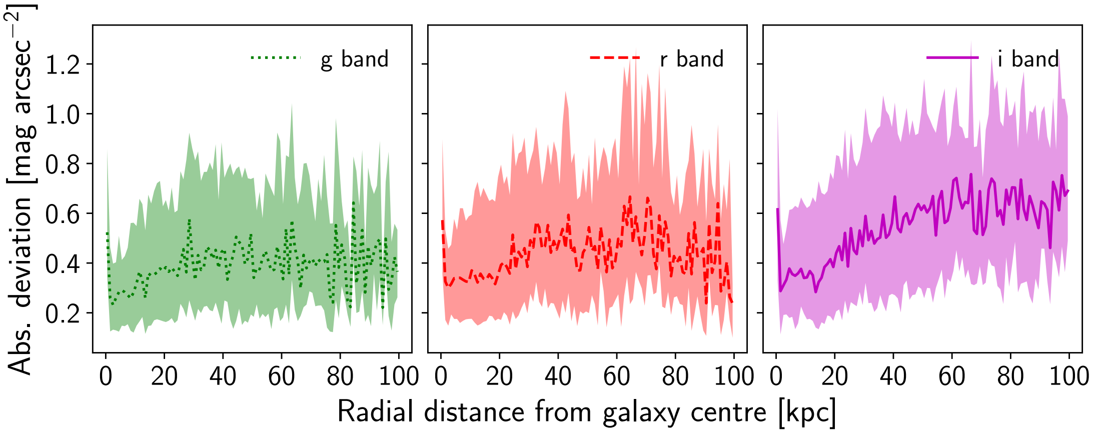


Figure 2.5: Median test set error per kpc from the galaxy centre, with the interquartile range shaded, split into the three bands present in the test set.

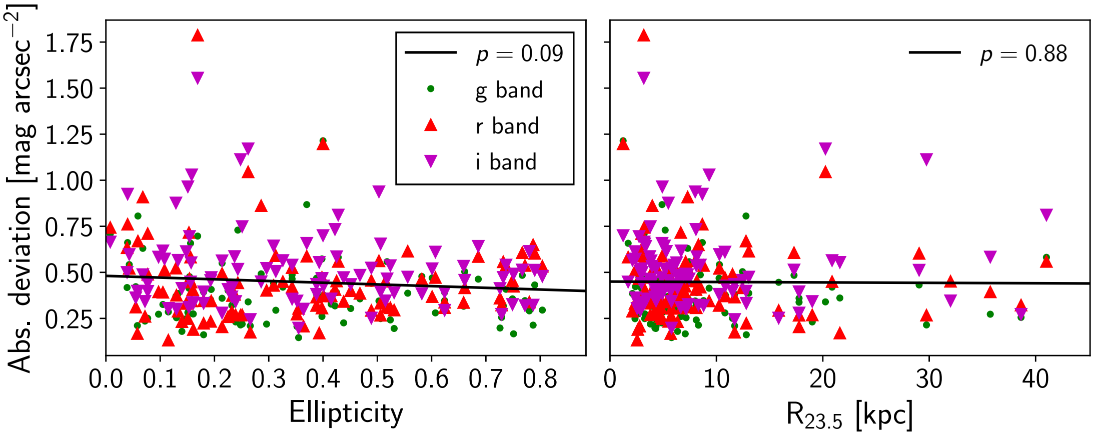


Figure 2.6: Median test set error over each galaxy predicted profile, plotted against each galaxy's ellipticity (left, 2.6a), and size at $R_{23.5}$ (right, 2.6b). p values obtained via a linear regression are stated in the legends.

estimation.

2.3.1 Comparison with AutoProf

AutoProf (Stone, Courteau, and Arora 2021) is a sibling method to Pix2Prof that uses a more standard astronomical pipeline to tackle the problem of non-parametric automated SB profile inference. A combination of standard image analysis packages from PHOTUTILS (Bradley et al. 2019) and novel techniques are used to construct a robust isophotal pipeline. Initial image analysis such as determining the PSF, centre finding, star masking, and sky background subtraction are performed using PHOTUTILS. Next, AutoProf simultaneously fits an ellipticity and position angle profile by minimising low order FFT coefficients along each isophote plus a regularisation

Table 2.2: Pix2Prof eliminates all interactive steps in the Courteau (1996) algorithm, alleviating subjectivity and accelerating inference significantly.

	Automated in:	
Process	Courteau (1996)?	Pix2Prof?
Identify galaxy centre	No	Yes
Estimate & remove sky background	Yes	Yes
Remove foreground objects	No	Yes
Fit contours	Yes	Yes
Extend contours to galaxy outskirts	No	Yes
Smooth isophotes	Yes	Yes
Interpolate poorly fitted data	No	Yes

term (Shalev-Shwartz and Ben-David 2014). The regularisation term penalises neighbouring ellipticity and position angle values that deviate significantly, ultimately favouring smooth profiles. Once the profiles have converged, AutoProf extracts the median SB along each isophote and determines the error with a 68.3 per cent quartile range. A curve of growth is determined by appropriately integrating the SB profile and propagating errors.

Fig. 2.4 compares the performance of Pix2Prof and AutoProf. AutoProf is found to produce SB profiles that are a slightly closer match to those produced via Courteau (1996) than Pix2Prof. The difference in absolute deviation between the two methods' profiles is typically around 0.1–0.2 mag arcsec^{-2}. However, AutoProf's accuracy comes with a time penalty; AutoProf takes on average 490 s to produce a profile on an Intel Xeon E5-2650 v3 CPU, a rate of 0.002 galaxies per second. This rate is roughly equivalent to the throughput of a human running the Courteau (1996) method. Pix2Prof processes 0.9 galaxies per second on the same hardware. Pix2Prof also offers more flexibility; it can be retrained to recreate any (semi) manual galaxy profile fitting pipeline and is therefore not limited to automation of the Courteau (1996) method.

AutoProf is presented in Stone, Courteau, and Arora (2021), and its code is available at https://github.com/ConnorStoneAstro/AutoProf.

2.4 Discussion and conclusions

While Pix2Prof can rapidly and accurately produce profiles of arbitrary length, there are some limitations to this technique. Principally, any profile produced will be biased to the training set. For instance, if Pix2Prof is trained on primarily nearby galaxies, it may not yield accurate profiles for more distant systems whose images will be poorly resolved. Similarly, if the model is trained on galaxy image–profile pairs as produced by numerical simulations the model will encode any flaws, incompleteness, or bias inherent to each simulation and will not encode instrumental effects (e.g. read-out noise) unless properly included. The same issue will occur if we train on galaxy image–profile pairs sampled from one survey and deploy the trained model on a dissimilar survey, for example SDSS, and LSST (York et al. 2000; Ivezić et al. 2019). It may be possible to mitigate this problem with an image domain translator (i.e. Isola et al. 2016; J. Zhu et al. 2017; Choi et al. 2018) that could transform observations so that they match a given survey. Of course, the Courteau (1996) measured profiles may also not entirely reflect the 'true' SB profile, due to modelling assumptions, human bias, and inherent noisiness in measurement. As neural networks typically require very large data sets, our relatively small data set is likely not reflecting the true potential of the model. Therefore, a larger set of training data could improve the results presented here. Generating a larger data set from simulated galaxies for training Pix2Prof will be a future project.

As described in Jia et al. (2015), due to the vanishing gradient problem an LSTM or GRU may 'forget' an image encoding as it unrolls. For Pix2Prof, this will manifest in a loss of accuracy at larger galactocentric radius. We see this effect in Fig. 2.4a's line plot and Fig. 2.5, but we cannot disentangle the individual contributions from image noise and the model architecture. However, assuming that the noise is significantly caused by GRU 'forgetfulness', future Pix2Prof models could imitate Jia et al. (2015) and counteract the noise by reinjecting the image encoding into the GRU's hidden state periodically as it unrolls. Another solution could involve adopting an architecture that suffers less from the vanishing gradient problem, such as the Transformer (Vaswani et al. 2017). The non-sequential nature of a Transformer would also allow us to parallelise output at inference time, reducing processing time even further.

In Section 2.1 we stressed the need for efficient and fully automated methods for timely analysis of ultra-large scale astrophysical imaging survey data. We believe that Pix2Prof addresses this challenge. Pix2Prof can predict any galaxy profile, given the right simulated or observed data set. Training Pix2Prof on simulated galaxy images offers additional benefits; the model could be trained on information that is only inferred indirectly in observations. For instance, Pix2Prof could train on sets of galaxy image–mass profile pairs directly in order to predict dark matter halo profiles, as mass profiles cannot be recovered classically by direct imaging observations. Furthermore, Pix2Prof has the potential to automate any galaxy profile fitting routine and be ported to other forms of galaxy image analysis that may not rely on isophotal analysis, but still produce a float sequence given a multidimensional array. These analyses could include galaxy component decompositions, the characterisation of galaxy interactions and distortions, pixelised stellar population synthesis, inference of galaxy mass distributions, and more (e.g. Eneev, Kozlov, and Sunyaev 1973; Vazdekis 2001; Peng et al. 2002). In a future study, we will demonstrate how Pix2Prof can be used to recover simultaneously the galaxy SB profile as well as the ellipticity profile and curve of growth of a galaxy.

An exciting future investigation involves building a system that can predict properties of unseen classes of objects. This could be achieved by building up a 'prior' that encodes known objects into a latent space and interpolates between their latent spatial representations at inference time. A generative model like the GAN (Goodfellow et al. 2014) or VAE (Kingma and Welling 2013) could achieve this (i.e. Spindler, Geach, and Smith 2020). Such a model could quickly identify astrophysically 'interesting' objects in a large field survey. The ability to search for rare objects in large unstructured data sets will become increasingly more important as new large scale astronomical surveys come online (Chambers et al. 2016; Aihara et al. 2017; Ivezić et al. 2019).

In summary, we have introduced a fully automated deep learning model for the extraction of sequential data from galaxy imagery. We have tested this model by applying it to the specific problem of estimating galaxy SB profiles, a process that previously required manual, time-consuming human intervention. We have tested our model on unseen galaxy images and found that our model has an average absolute deviation of 0.34 mag arcsec^{-2} with an interquartile range of 0.22 mag arcsec^{-2}, while inferring SB profiles over two orders of magnitude faster than the classic (interactive) algorithm it automates.

Data and Code Availability. The code and trained model used in this chapter is available at `https://github.com/Smith42/pix2prof`. The profile data set used to train the network is documented in Arora et al. (2021).

Carbon Emissions. The training of deep learning models requires considerable energy, contributing to carbon emissions (Lacoste et al. 2019; Strubell, Ganesh, and McCallum 2019). The energy used while training Pix2Prof on a single NVIDIA V100 GPU is estimated to be ∼20 kWh (5.54 kg CO_2 e.) according to the Machine Learning Impact calculator described in Lacoste et al. (2019). To counteract further emission from redundant retraining, we follow the recommendations of Strubell, Ganesh, and McCallum (2019) and make available the fully trained model, as well as the code to run it. Also, we will make available trained models for any improvements that we make to Pix2Prof in the future.

Chapter 3

SAGAN: Synthetic Astronomy Generative Adversarial Network

GANs are a class of artificial neural network that can produce realistic, but artificial, images that resemble those in a training set. In typical GAN architectures these images are small, but a variant known as Spatial GANs (SGANs) can generate arbitrarily large images, provided training images exhibit some level of periodicity. Deep extragalactic imaging surveys meet this criteria due to the cosmological tenet of isotropy. Here we train an SGAN to generate images resembling the iconic *Hubble Space Telescope* eXtreme Deep Field (XDF). We show that the properties of 'galaxies' in generated images have a high level of fidelity with galaxies in the real XDF in terms of abundance, morphology, magnitude distributions and colours. As a demonstration we have generated a 7.6-billion pixel 'generative deep field' spanning 1.45 degrees. The technique can be generalised to any appropriate imaging training set, offering a new purely data-driven approach for producing realistic mock surveys and synthetic data at scale, in astrophysics and beyond.

This chapter has been previously published as M. J. Smith and J. E. Geach (2019). 'Generative deep fields: arbitrarily sized, random synthetic astronomical images through deep learning'. In: "*Monthly Notices of the Royal Astronomical Society*" 490.4, pp. 4985–4990. DOI: 10.1093/mnras/stz2886. arXiv: 1904.10286 [astro-ph.IM]. It has been presented as a contributed talk at *The Big Data Era in Astronomy 2020*, Online.

3.1 Introduction

Synthetic, or mock, data plays an important role in the interpretation of observations, as it provides a means to test a theoretical framework, as a tool to explore biases or systematics in data analysis, or to help design future experiments. In astrophysics, like many fields, a standard route to generating synthetic data is to use an analytic or semi-analytic model, or numerical simulation, to generate synthetic data that mimic the observations e.g. Cole et al. (1998), Obreschkow et al. (2009), and Mandelbaum et al. (2012). The most common form of observation in astrophysics is digital imaging, and deep extragalactic imaging surveys have transformed our understanding of the Universe (Williams et al. 1996; Scoville et al. 2007).

Current methods to generate synthetic deep fields include the projection of volume- and resolution-limited hydrodynamical cosmological simulations e.g. Vogelsberger et al. (2014) and Schaye et al. (2015) into lightcones (Vogelsberger et al. 2014), requiring a treatment for the transport of radiation through the volume and modelling of a particular instrument response (Jonsson 2006; Trayford et al. 2017). Alternatively, mock deep fields can be created by taking an input catalogue containing the positions and properties of fake galaxies and applying models to describe their light profiles (Bertin 2009; Dobke et al. 2010; Bergé et al. 2013; Peterson et al. 2015; Rowe et al. 2015). Mock blank fields can be created entirely parametrically, as in the case of the Ultra Fast Image Generator (UFIG; Bergé et al. 2013), and these have the advantage of having a very well-defined input model. However, analytic and semi-analytic models require explicit encoding of astrophysical properties such as light profiles, and are therefore limited in realism. This is important when considering objects in surveys that are poorly described in this way, for example interacting/merging or high redshift galaxies. Therefore, if the physical model is incomplete or flawed or too simplistic, simulations produced by the model may not be representative of real data. Furthermore, additional processing is required to produce a realistic synthetic observation, for example an understanding of the noise properties and convolution with a point spread function. Letting a machine infer both the astrophysical *and* the instrumentational properties of a data set sidesteps these issues: with a large representative data set it is possible to use empirical data to construct new, realistic, but synthetic observations, at the expense of model transparency.

GANs are a type of deep learning algorithm that can generate new samples from a probability distribution learnt from a representative training set (Goodfellow et al. 2014; Radford, Metz, and Chintala 2016; Salimans et al. 2016). The adversarial aspect of the algorithm refers to the use of two neural networks – a generator, G and discriminator, D – that compete during training. G tries to estimate the probability distribution of the input data by producing samples that aim to trick D, which is estimating the probability that the generated sample came from the training set. Training is a 'minimax' game where G is trying to maximize the likelihood that D

predicts the generated samples are from the real data. The generator transforms a 'latent' vector z, into an output $G(z)$. Meanwhile, the discriminator takes either the output of the generator $G(z)$, or a real data example x, and transforms this input into an output, $D(G(z))$ or $D(x)$. The output can be thought of as the probability that $G(z)$ is indistinguishable from x. The networks are trained through gradient descent (Robbins and Monro 1951) until $G(z)$'s distribution closely matches the distribution of x. After training, the generator can produce convincing images resembling those in the training set. The current state of the art is capable of producing very convincing imagery (Karras et al. 2017; Brock, Donahue, and Simonyan 2018; Karras, Laine, and Aila 2018; Karnewar, Wang, and Iyengar 2019). These generated images can be thought of as random draws from the probability distribution estimated by the generator that describes the distribution from which pixels in the training image were sampled.

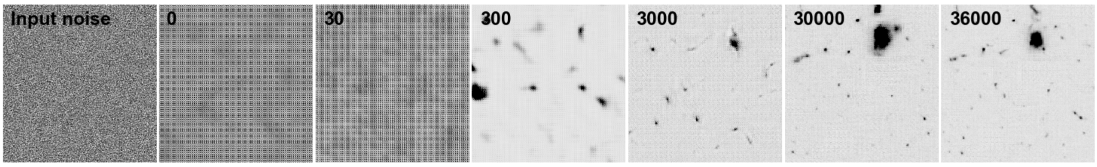


Figure 3.1: Evolution of SGAN training. The images show a 256×256 pixel generated image for the F775W channel after 0, 30, 300, 3000, 30,000 and 36,000 epochs using the same input latent noise vector, z. The distribution of input (Gaussian) noise before training is shown in the first panel. Each image is linearly scaled with a 0.5–99.5% percentile clip. Structure becomes apparent after a few hundred training epochs and is continuously refined as training progresses. The total training time to 36,000 epochs is 1,280 hours wall time on an NVIDIA Tesla K40c GPU.

One limitation of the GAN technique is its instability during training (Kodalg et al. 2017; Roth et al. 2017). We mitigate this somewhat by incorporating the relativistic discriminator introduced in Jolicoeur-Martineau (2018) (see section 3.2.3). Also, a GAN model is inferred completely on training data. Therefore the trained model is only as accurate as the given data, and preferably a lot of data is required for training. Model interpretability can also be an issue with large neural networks.

In this work we present a method exploiting GANs to generate realistic, but random, extragalactic deep field images of arbitrary size. In §3.2.3 we describe the 'Spatial GAN' variant, and explain how it is trained to generate fake images. In §3.3 we describe our results, comparing 'galaxies' detected in the fake images to galaxies detected in the training image. We discuss and summarise the results and highlight limitations and scope for improvement and future work in §3.4. Magnitudes are all quoted on the AB system.

3.2 Method

3.2.1 Spatial GANs

A Spatial GAN (SGAN; Jetchev, Bergmann, and Vollgraf 2016) is a fully convolutional GAN that uses variably sized 2D latent vector arrays as the generator input z. This is in contrast to a standard GAN, which uses a 1D latent vector. In a standard GAN a dense neural layer is used to connect and reshape the latent vector so that a convolutional layer can operate on it. This lack of a fully-convolutional architecture means that a standard GAN can only produce images of a single, fixed shape. An SGAN replaces all dense neural layers in both its generator and its discriminator with convolutional layers. This allows input of variably sized image–latent vector array pairs. In an SGAN the latent vector arrays are upsampled using deconvolving layers in the generator, and both generated and real images are downsampled in the discriminator via convolving layers. Since z can be varied, an SGAN can be used to create an image of any size, even one much larger than seen in the training set. If the training images exhibit periodicity, the SGAN's generator will also learn to exhibit the same periodicity (Jetchev, Bergmann, and Vollgraf 2016).

Note that the presence of periodicity in the training data is not a requisite for producing arbitrarily sized output images, but it is a requisite for producing realistic output images. An SGAN trained on an image of a cluster of galaxies, for example, will only produce output images that contain galaxy clusters, regardless of size. However, cosmic isotropy means that SGANs trained on deep 'blank field' extragalactic images, can in principle produce output images of arbitrary size that resemble the real Universe. The important caveat and limitation is that the generated images will only contain objects similar to those present in the training image; if the training image is too small to contain a representative sampling of the galaxy population (e.g. the rare massive clusters), then these types of object will not appear in the output. The SGAN cannot extrapolate in this manner, and this should be an important consideration when training and using such networks.

3.2.2 Training set

We use a training set comprised of the F814W, F775W and F660W bands of the *Hubble Space Telescope* eXtreme Deep Field (XDF; Illingworth et al. 2013), with all images aligned and sampled on the same 60-mas grid[32] The only other preprocessing is a channel-wise clip at the 99.99th percentile. Unlike many GAN approaches that use

[32] https://archive.stsci.edu/prepds/xdf.

training images scaled to an 8-bit depth per channel, we train using the full floating point dynamic range of the data, with 32-bit depth per channel. Training images are sampled from the full XDF image by cropping the image at random positions with crop sizes of 64, 128 or 256 pixels, with the size fixed for each batch. Corresponding noise arrays of sizes 4, 8, or 16 pixels, are sampled from a Gaussian distribution with a zero mean and unity variance, and passed to the generator. Each noise array has a channel axis of size 50.

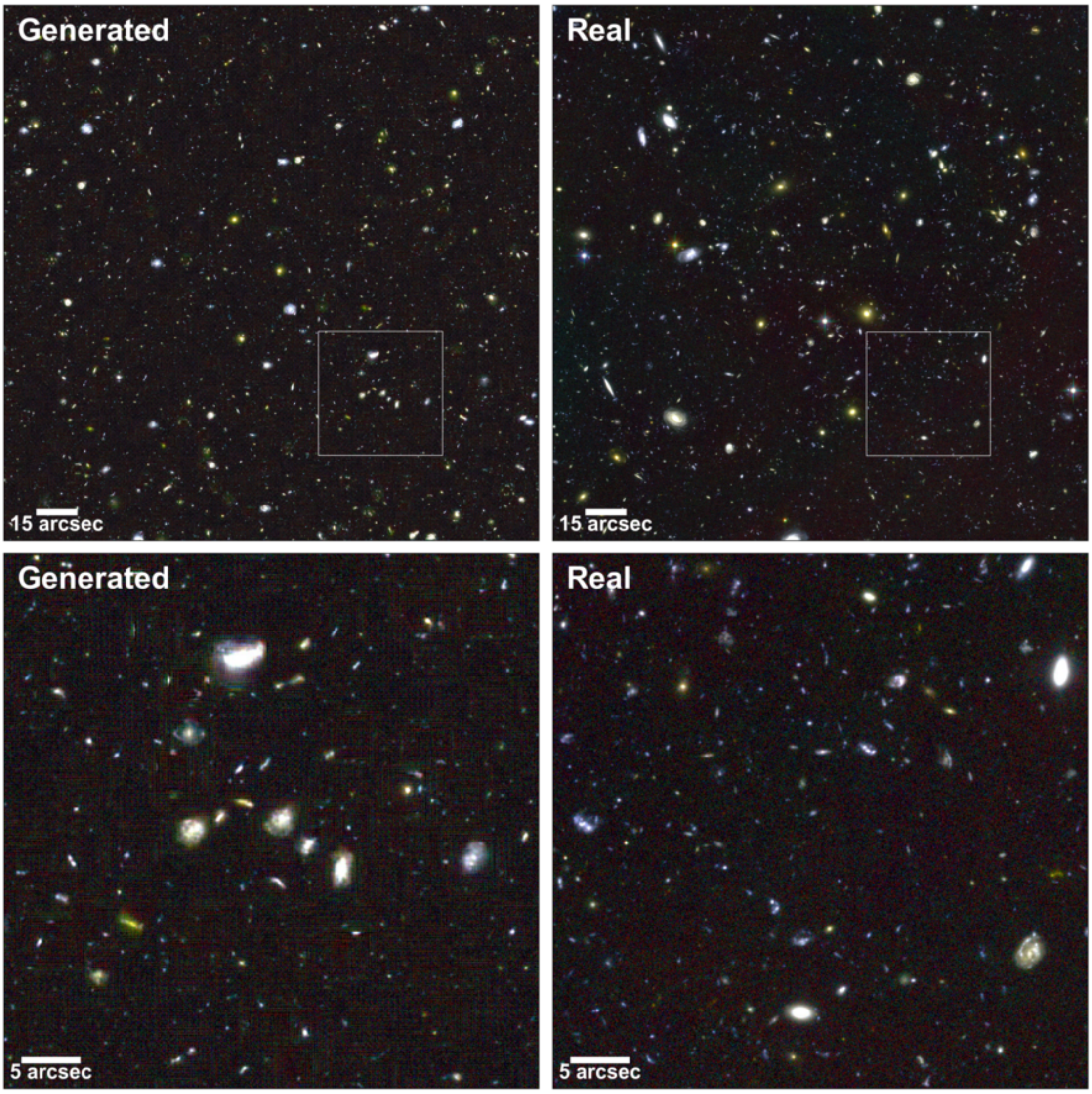


Figure 3.2: Example of a generated deep field. The top left panel shows a generated deep field with the RGB channels corresponding to the F814W, F775W and F606W bands. The size of the generated image was set to 3.2′, spanning the full XDF (top right), which has a 60 mas pixel^{-1} scale. The lower panels show 45″ zoom-ins of the generated and real fields.

3.2.3 Architecture and training

The generator is comprised of one initial deconvolutional layer with a stride of 2 and a kernel size of 4. Three deconvolutional sets are then applied, each comprising of one layer with a stride of 2, followed by three layers with strides of 1. Each layer in these sets have kernel sizes of 4. These layers have an exponential linear unit (ELU) activation Clevert, Unterthiner, and Hochreiter (2016). The generator's output layer is also deconvolutional, with a stride of 1, a kernel size of 3, and 3 filters. The output layer has a sigmoid activation function,

$$\varphi(x) = (1 + e^{-x})^{-1}. \tag{3.1}$$

The discriminator comprises of 5 initial convolutional layers, each with a stride of 2, and a kernel size of 4. All initial layers have a Leaky Rectified Linear Unit (ReLU) activation (Maas, Hannun, and Ng 2013). 2D global average pooling is applied to the final convolutional layer, and a dense layer is used to connect the global average pool to a binary classification output.

The discriminator uses a relativistic average loss (Jolicoeur-Martineau 2018). A relativistic discriminator estimates whether the incoming data are more realistic than a random sample of the opposing type. To understand why this is important, consider that in a standard GAN a perfect generator will cause the discriminator to define all incoming data as 'real'. However, it is known that exactly half of an incoming batch is real data. Therefore, if the generator is producing flawless fakes, the discriminator should assume that each sample has a 50% probability of being real (Jolicoeur-Martineau 2018). To take this prior knowledge into account, the output function of a non-relativistic discriminator is modified from $D(x) = \varphi(C(x))$, where $C(x)$ is the output of the final layer with no applied activation function, to $D_R(x)$

$$D_R(x) = \begin{cases} \varphi(C(x) - \mathbb{E}_{x_f \sim \mathbb{Q}} C(x_f)) \text{ for real } x, \\ \varphi(C(x) - \mathbb{E}_{x_r \sim \mathbb{P}} C(x_r)) \text{ for generated } x, \end{cases} \tag{3.2}$$

where φ is the sigmoid activation function. The second parts of the real and generated $D_R(x)$ are effectively the average discriminator value for fake images, x_f, and real images, x_r, respectively. The use of a relativistic discriminator leads to stable training on a difficult data set of large images, where a standard discriminator fails (Jolicoeur-Martineau 2018). The discriminator is packed to stabilize training (Lin et al. 2018). To pack the discriminator, two images from the same class are concatenated along their channel axes and fed into the discriminator as a single sample. Packing the discriminator in this way reduces the possibility of mode collapse (Lin et al. 2018). The Adam optimizer (Kingma and Ba 2015) is used, with a learning rate of 0.0002. The learning rate was determined through a manual search as the maximum rate that yields stable training.

The SGAN is trained on an NVIDIA Tesla K40c GPU for 36,000 epochs of 30 batches with a batch size of 128. Each epoch requires approximately 120 seconds, depending on the batch crop sizes. The evolution of the generated images for a fixed latent noise vector z is shown in Fig. 3.1, showing the emergence of amorphous structure and then refinement into structures resembling galaxies with increasing training time.

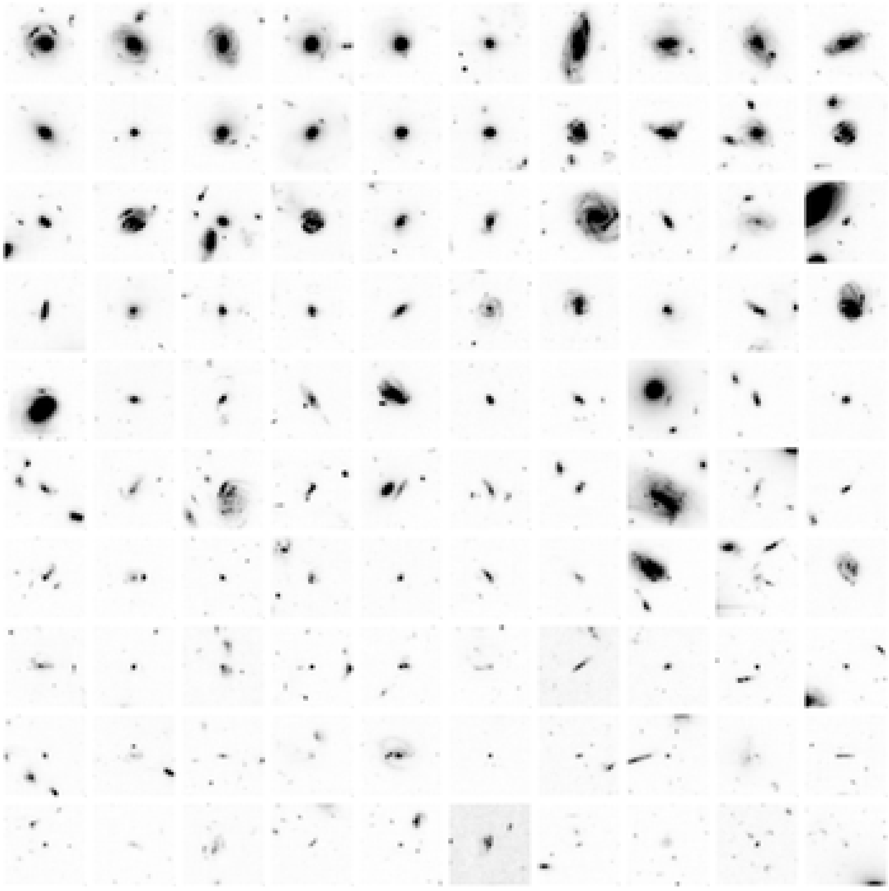


Figure 3.3: Thumbnail images of real and generated 'galaxies'. Each image is $7.2''$ across and shows the F775W channel on an identical linear grayscale for 100 sources selected in the range 21–26 mag. Fifty targets are selected from each of the real and generated catalogues and displayed in random order in row-wise ascending order of magnitude.

3.3 Results

A self-ensemble of 21 GANs is created by taking every tenth epoch's model from a range of 200 epochs. A CNN ensemble is a combination of different CNNs trained from different weight initialisations on the same data. A self-ensemble is an ensemble of CNNs trained from the same initialisation, but taken at different points in the training cycle. Ensembles of CNNs can produce significantly more accurate predictions when compared to a single CNN (Nanni, Ghidoni, and Brahnam 2018; Paul et al. 2018). This increase in accuracy has been shown by Wang, Zhang, and Weijer (2016) and Mordido, Yang, and Meinel (2018) to also be present in GAN ensembles.

Four outputs were taken from each of these 21 GANs, resulting in 84 total XDF simulations. To generate a simulation set, each of the trained generators is fed a z vector that has shape $[b, 202, 202, 50]$, where $b = 4$ is the batch size. The generators up-sample z to a shape $[b, 3232, 3232, 3]$, matching the shape of the training image. The generation time for single realization of the XDF is 15 seconds for all three bands on an Intel Xeon CPU E5-2650 v3. Using the same CPU it would take 1.5 hours wall time to generate a 3 billion pixel image similar to one produced by the LSST. In principle, the generated image can be of arbitrary contiguous size by using seamless tessellation. Seamless tessellation can be achieved with SGAN by having adjacent tiles a and b share a portion of boundary z values. This creates a shared area where $G(z_a)$ and $G(z_b)$ have an equal output. A seamless tile is made by cropping at the midpoint of the shared noise and concatenating $G(z_a)$ and $G(z_b)$. To illustrate, we have generated a contiguous 87040×87040 pixel (1.45° × 1.45°) version of the XDF that can be examined online at http://star.herts.ac.uk/~jgeach/gdf.html.

Fig. 3.2 presents an example of a generated field in comparison to the real XDF, combining the F814W, F775W and F606W bands into an RGB colour composite. We show the full field, spanning approximately $3'$, and a zoom-in of a $45''$ region for closer inspection. The generated image is reasonably convincing as an extragalactic deep field from the point of view of cursory visual inspection, although like many GAN-generated images, on closer inspection there are artefacts (such as low surface brightness features, pixelization, and discontinuities not seen in the real image) that betray the counterfeit. These artefacts are likely caused by a lack of resolution in the GAN generator, and could be remedied through the addition of more layers, or through the use of a larger convolutional kernel. Both of these approaches would increase the receptive field. The GAN also cannot reproduce some of the larger galaxies with the same level of detail as the real image. Nevertheless, the generated image does contains a mixture of 'early' and 'late' type galaxies amongst a more numerous field of background sources. The early and late types have the correct colour and morphology, i.e. classic red/elliptical and blue/spiral respectively, and background sources have the clumpy and disturbed morphology typically seen in the high redshift galaxy population. Fig. 3.3 presents single-band (F775W) thumbnail images of a sample of 100 randomly chosen sources with F775W AB magnitudes in the range 21–26 mag. 50 sources are selected from the real image and 50 are selected from the generated images, but the thumbnails have been randomly shuffled in the figure to demonstrate to the reader the visual similarity of individual galaxies in the generated and real data.

To test whether the similarity between the generated and real image is more than superficial, we extract 'galaxies' from the generated images and compare to those in the real XDF. We use the source finder *SExtractor* (Bertin and Arnouts 1996) to detect galaxies in the combined F606W+F775W+F814W (pixelwise-mean) images, measuring the corresponding source flux in the individual bands. Sources are identified with the criteria that 5 contiguous pixels have a signal $\geq 5\sigma$ above the local background. The same detection procedure is applied to the real and generated images, and the photometric zeropoints for the generated images are identical to the real data. Fig. 3.4 compares the magnitude distribution of sources detected in the real field and ensemble of generated fields measured in each of the F606W, F775W and F814W bands. The generated and real distributions bear close statistical resemblance. The absolute difference in the generated and real median magnitudes in each of the F606W, F775W and F814W bands is within 0.02 magnitudes, and a the generated images produce the correct abundance of galaxies across the full magnitude range in every band. The p-values given by a two-sided Kolmogorov-Smirnov test for a magnitude limit of 28 mag (approximately the completeness limit of the catalogue for our detection criteria) are 0.16, 0.63 and 0.75, indicating that we cannot say with confidence that the distributions are drawn from different parent populations, and are therefore statistically similar.

3.4 Discussion and conclusions

While the SGAN can produce arbitrarily-sized simulations of the XDF, there are clear limitations of this technique. The primary limitation is that the generated images are of course totally biased to the training set. The consequence is that a generated image much larger than the training data will not be cosmologically representative. This is simply because the XDF probes a volume too small to contain rare objects such as, for example, clusters of galaxies and the generator cannot produce examples

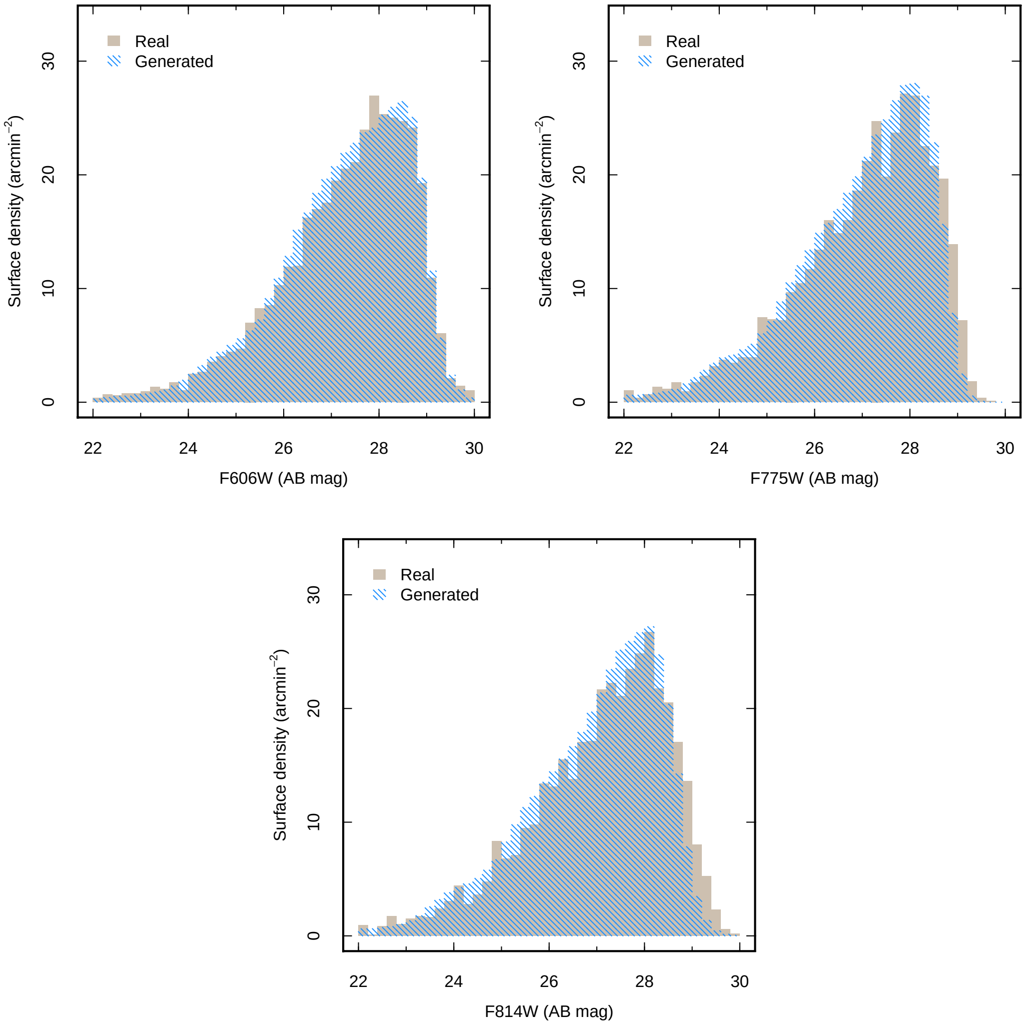


Figure 3.4: The photometric properties of galaxies detected in the real XDF and ensemble of generated images. The histograms show the magnitude distributions of the galaxies detected in the band-merged image and measured in each of the F606W, F775W and F814W bands. The photometric zeropoints applied to the generated data are identical to the zeropoints calibrated for the real data.

of objects it has not seen. This problem is simple to alleviate by using a much larger and representative training set. In the real Universe, the positions of galaxies are correlated across a wide range of angular scales, due to the presence of cosmological large-scale structure. Indeed, due to gravitational lensing, even the shapes of galaxies can be correlated from cosmic shear, and this will not be well captured by the SGAN.

Another limitation is that this clustering information is not present in the generated images, although correlations on the scales of the crop size will be somewhat preserved.

GANs also have some architectural drawbacks. It can be difficult to train a GAN stably; we were not able to achieve stable training with more than three photometric bands. This issue could be resolved with more regularization: adding dropout (Srivastava et al. 2014b), or spectral normalisation (Miyato et al. 2018) could allow one to generate additional bands, although at the cost of a longer training time, and a larger memory footprint during training.

An advantage of this technique is that it is empirically driven, since the data itself is used as the model. By training directly on imaging data the SGAN simultaneously encodes information about the instrumental response as well as the 'galaxy' population, thus circumventing the need to make modelling assumptions about either, as is the case in other approaches. The corollary to this is that in the current approach we cannot disentangle the instrument response from the underlying galaxy population, although one could envision approaches to tackle this challenge. For example, a method similar to that used by Schawinski et al. (2017), where a GAN is used to remove noise in astronomical imagery, could be employed. Specifically, a GAN with a UNet-like (Ronneberger, Fischer, and Brox 2015) generator would learn a transformation between an image with an entangled instrument response, and one without. Also, since the model is entirely data-driven, it is highly dependent on the quality and quantity of the training data.

Recent advances in GAN training techniques have resulted in impressively high fidelity image output (Karras et al. 2017; Brock, Donahue, and Simonyan 2018; Karras, Laine, and Aila 2018; Zhang et al. 2018; Karnewar, Wang, and Iyengar 2019). Similar methods could be implemented to improve the quality of the generated deep fields. The same methods could also reduce the occurrence of artefacts. Training on a larger set of imaging data, with a increased batch size (Brock, Donahue, and Simonyan 2018), may also produce more representative simulations on a deeper network. The addition of a GAN architecture that allows for control over the output, such as InfoGAN (X. Chen et al. 2016) or Conditional GAN (Mirza and Osindero 2014) could also be useful when creating mock surveys because they would allow control over the frequency of particular objects of interest, or over the makeup of background noise.

This method could be used to generate at scale entirely artificial, but realistic, image realizations for the design, development, and exploitation of new surveys. For example, one could assemble large training sets for instance segmentation and classification of galaxies. In the early stages of a new survey, relatively small amounts of data could be collected, but then expanded to a level useful for training deep learning models using the generative method described here. Segmentation and classification algorithms could then be trained on the generated data, and then applied to new data, allowing far faster deployment of astronomical deep learning algorithms than would otherwise be possible, potentially accelerating the exploitation of new survey data.

Data and code availability. The SAGAN code can be obtained at `https://github.com/Smith42/XDF-GAN`. The 7.6-billion pixel version of the generated XDF can be viewed at `http://star.herts.ac.uk/~jgeach/gdf.html`.

Carbon emissions. The training of deep learning models requires considerable energy, contributing to carbon emissions. The energy used to train SAGAN to completion is estimated to be 310 kWh, corresponding to 87 kg CO2e. according to the Machine Learning Emissions Calculator described in Lacoste et al. (2019). To counteract further emission from redundant retraining, we follow the recommendations of Strubell, Ganesh, and McCallum (2019) and make available the fully trained models, as well as the code to run them. Also, we will make available trained models for any improvements that we make to SAGAN in the future.

Chapter 4

Realistic galaxy image simulation via score-based generative models

We show that a Denoising Diffusion Probabalistic Model (DDPM), a class of score-based generative model, can be used to produce realistic yet mock images that mimic observations of galaxies. Our method is tested with Dark Energy Spectroscopic Instrument *grz* imaging of galaxies from the Photometry and Rotation curve OBservations from Extragalactic Surveys (PROBES) sample and galaxies selected from the Sloan Digital Sky Survey. Subjectively, the generated galaxies are highly realistic when compared with samples from the real dataset. We quantify the similarity by borrowing from the deep generative learning literature, using the 'Fréchet Inception Distance' to test for subjective and morphological similarity. We also introduce the 'Synthetic Galaxy Distance' metric to compare the emergent physical properties (such as total magnitude, colour and half light radius) of a ground truth parent and synthesised child dataset. We argue that the DDPM approach produces sharper and more realistic images than other generative methods such as Adversarial Networks (with the downside of more costly inference), and could be used to produce large samples of synthetic observations tailored to a specific imaging survey. We demonstrate two potential uses of the DDPM: (1) accurate in-painting of occluded data, such as satellite trails, and (2) domain transfer, where new input images can be processed to mimic the properties of the DDPM training set. Here we 'DESI-fy' cartoon images as a proof of concept for domain transfer. Finally, we suggest potential applications for score-based approaches that could motivate further research on this topic within the astronomical community.

This chapter has been previously published as M. J. Smith et al. (2022). 'Realistic galaxy image simulation via score-based generative models'. In: *Monthly Notices of the Royal Astronomical Society* 511.2, pp. 1808–1818. DOI: 10.1093/mnras/stac130. arXiv: 2111.01713 [astro-ph.IM]. It has been presented on 2022-03-10 as an invited talk for the Open University, Milton Keynes, UK, and on 2022-05-05 as an invited talk for the Alan Turing Institute, London, UK. It has also been presented as a contributed talk at *EAS 2022*, Valencia, Spain and at *NAM 2022*, Warwick, UK.

4.1 Introduction

Synthetic data will play a pivotal role as we journey further into astronomy's epoch of big data, especially for large extragalactic surveys (York et al. 2000; Dewdney et al. 2009; Amiaux et al. 2012; Ivezić et al. 2019). It will be required to train machine learning methods, to interpret observations, and to test theoretical frameworks. Indeed, one form of synthetic data comes from theoretical models. For example, in the field of galaxy formation and evolution, simulations using semi-analytical approaches have been successful in reproducing many of the bulk observable and emergent properties of galaxies over a significant fraction of cosmic time (e.g. Somerville and Primack 1999; Cole et al. 2000; Bower et al. 2006; Croton et al. 2006). Semi-analytical models (SAMs) employ approximations derived from more detailed numerical simulations and empirical calibrations from data to model galaxy formation and evolution. So it is possible to generate, for example, a 'mock' catalogue of galaxies with predicted optical photometry (Lagos et al. 2019). Hydrodynamical models of galaxy formation track the evolution of baryons and dark matter within representative volumes (e.g. Dubois et al. 2014; Vogelsberger et al. 2014; Khandai et al. 2015; Schaye et al. 2015; Kaviraj et al. 2017), and when pushed to high spatial resolution, can predict galaxy morphologies on physical scales commensurate with the angular scales achievable with current observational facilities. When radiative transfer schemes are applied for the propagation of (for example) starlight through the volume, realistic synthetic observations can be produced, to be compared with nature (e.g. Camps et al. 2016; Trayford et al. 2017; Lovell et al. 2021).

To properly mimic real astronomical data, with all its wonderful idiosyncrasies, requires a full and detailed understanding of the telescope response, instrumental properties, and observing conditions, not to mention the nuances of any data reduction procedure. These non-trivial steps are typically unique to a given set of observations. There is a short-cut however: armed with enough examples of observations from a given survey, it should be possible to derive a data-driven approach to mimic not only the content of interest – astronomical signal – but also the properties of the data themselves. Deep generative models enable precisely that.

Great attention has been given to applications of deep generative learning to problems in astronomy lately. GANs (Goodfellow et al. 2014) have been used for deconvolution (Schawinski et al. 2017), synthetic galaxy generation (Ravanbakhsh et al. 2016; Fussell and Moews 2019), dark matter simulation (Mustafa et al. 2019b; Tamosiunas et al. 2020), and deep field imagery generation (Smith and Geach 2019). Variational Auto-Encoders (VAE; Kingma and Welling 2013) have been used to simulate galaxy observations (Ravanbakhsh et al. 2016; Spindler, Geach, and Smith 2020; Lanusse et al. 2021), as have flow-based models (Rezende and Mohamed 2015; Bretonnière et al. 2021). In this chapter, we show that it is possible to simulate realistic galaxy imagery with a Denoising Diffusion Probablisitic Model (DDPM).

DDPMs were introduced by Sohl-Dickstein et al. (2015) and were first shown to produce high quality synthetic samples by Ho, Jain, and Abbeel (2020). They belong to a family of generative deep learning models that employ denoising 'score matching' via annealed Langevin dynamic sampling (Ho, Jain, and Abbeel 2020; Jolicoeur-Martineau et al. 2020; Song and Ermon 2020; Song et al. 2021). This family of score-based generative models (SBGMs) can generate imagery of a quality and diversity surpassing state of the art GAN models, a startling result considering the historic disparity in interest and development between the two techniques (Dhariwal and Nichol 2021; Nichol and Dhariwal 2021; Song et al. 2021).

SBGMs have already been used to super-resolve images (Kadkhodaie and Simoncelli 2020; Saharia et al. 2021), translate between image domains (Sasaki, Willcocks, and Breckon 2021), separate superimposed images (Jayaram and Thickstun 2020), and in-paint information (Kadkhodaie and Simoncelli 2020; Song et al. 2021). At the time of writing, there are only two examples of score-based modelling in the astronomy literature (Remy et al. 2020, 2022). This is despite some obvious uses in astronomical data pipelines. For example: an implementation like Sasaki, Willcocks, and Breckon (2021) could be used for survey-to-survey photometry translation similarly to Buncher, Sharma, and Carrasco Kind (2021); the source image separation model described in Jayaram and Thickstun (2020) could be applied as an astronomical object deblender (for example: Stark et al. 2018; Reiman and Göhre 2019; Arcelin et al. 2021); and information inpainting could be used to remove nuisance objects from observations (Kadkhodaie and Simoncelli 2020; Song et al. 2021).

This chapter is organised as follows. Section 4.2 introduces the DDPM formulation used in this chapter. In Section 4.3 and Section 4.4, we show that DDPMs are capable of generating diverse synthetic galaxy observations that are both statistically and qualitatively indistinguishable from observations found in the training set. We also demonstrate that DDPMs can in-paint occluded information in an observation, such as

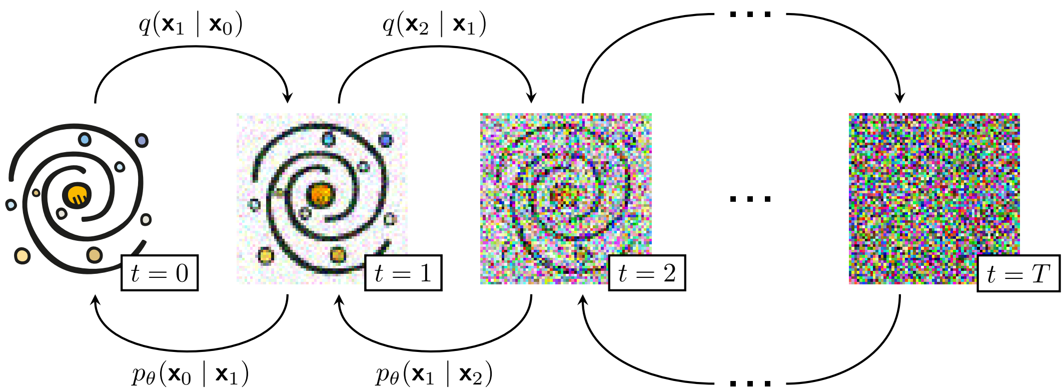

$q(\mathbf{x}_1 \mid \mathbf{x}_0)$
$q(\mathbf{x}_2 \mid \mathbf{x}_1)$
$t = 0$
$t = 1$
$t = 2$
$t = T$
$p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1)$
$p_\theta(\mathbf{x}_1 \mid \mathbf{x}_2)$

Figure 4.1: It is easy (and achievable without learnt parameters) to add noise to an image, but more difficult to remove it. DDPMs attempt to learn an iterative removal process through an appropriate neural network. p_θ.

satellite trails[33], and show that we can inject realism into entirely unrealistic cartoon imagery. A discussion of our results and suggestions for future research are presented in Section 4.5.

4.2 Denoising diffusion probabilistic models

Denoising Diffusion Probabilistic Models (DDPMs) define a diffusion process that projects a complex image domain space onto a simple domain space. In the original formulation, this diffusion process is fixed to a predefined Markov chain that adds a small amount of Gaussian noise with each step. Figure 4.1 illustrates that this 'simple domain space' can be noise sampled from a Gaussian distribution: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbb{1})$.

4.2.1 Forward process

We define a Markov chain to slowly add Gaussian noise to our data:

$$q(\mathbf{x}_{0\ldots T}) = q(\mathbf{x}_0) \prod_{t=1}^{T} q(\mathbf{x}_t \mid \mathbf{x}_{t-1}).$$

The amount of noise added per step is controlled with a variance schedule $\{\beta_t \in (0,1)\}_{t=1}^{T}$, such that

$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\,\mathbf{x}_{t-1}, \beta_t \mathbb{1}). \tag{4.1}$$

[33] A growing problem due to the rapidly increasing population of satellites, exacerbated by mega-constellations (Kocifaj et al. 2021).

This process is applied iteratively to the input image, $\mathbf{x}_0$. If we define the above equation to only depend on $\mathbf{x}_0$, we can immediately calculate an image representation $\mathbf{x}_t$ for any t (Ho, Jain, and Abbeel 2020). Defining $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$:

$$\begin{aligned}
\mathbf{x}_t &= \sqrt{\alpha_t}\,\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\,\mathbf{z}_{t-1} \\
&= \sqrt{\alpha_t \alpha_{t-1}}\,\mathbf{x}_{t-2} + \sqrt{(1-\alpha_t) + \alpha_t(1-\alpha_{t-1})}\,\bar{\mathbf{z}}_{t-2} \\
&= \sqrt{\alpha_t \alpha_{t-1} \alpha_{t-2}}\,\mathbf{x}_{t-3} + \sqrt{(1-\alpha_t\alpha_{t-1}) + \alpha_t\alpha_{t-1}(1-\alpha_{t-2})}\,\bar{\mathbf{z}}_{t-3} \\
&= \dots \\
&= \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\mathbf{z},
\end{aligned}$$

where $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbb{1})$ and $\bar{\mathbf{z}}$ is a combination of Gaussians. Substituting this expression into Eq. 4.1 removes the $\mathbf{x}_{t-1}$ dependency and yields

$$q(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbb{1}).$$

4.2.2 Reverse process

DDPMs attempt to reverse the forward process by applying a Markov chain with learnt Gaussian transitions. In our case these transitions are learnt via an appropriate neural network, p_θ:

$$\begin{aligned}
p_\theta(\mathbf{x}_{0\dots T}) &= p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t), \\
p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) &= \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)).
\end{aligned}$$

While $\boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)$ can be learnt[34], we follow Ho, Jain, and Abbeel (2020) and fix it to an iteration-dependent constant $\sigma_t^2\mathbb{1}$, where $\sigma_t^2 = 1 - \alpha_t$.

By recognising that DDPMs are a restricted class of Hierarchical VAE, we see that we can train p_θ by optimising the evidence lower bound (ELBO, introduced in Kingma and Welling 2013) that can be written as a summation over the Kullback-Leibler

[34] See for example Nichol and Dhariwal (2021).

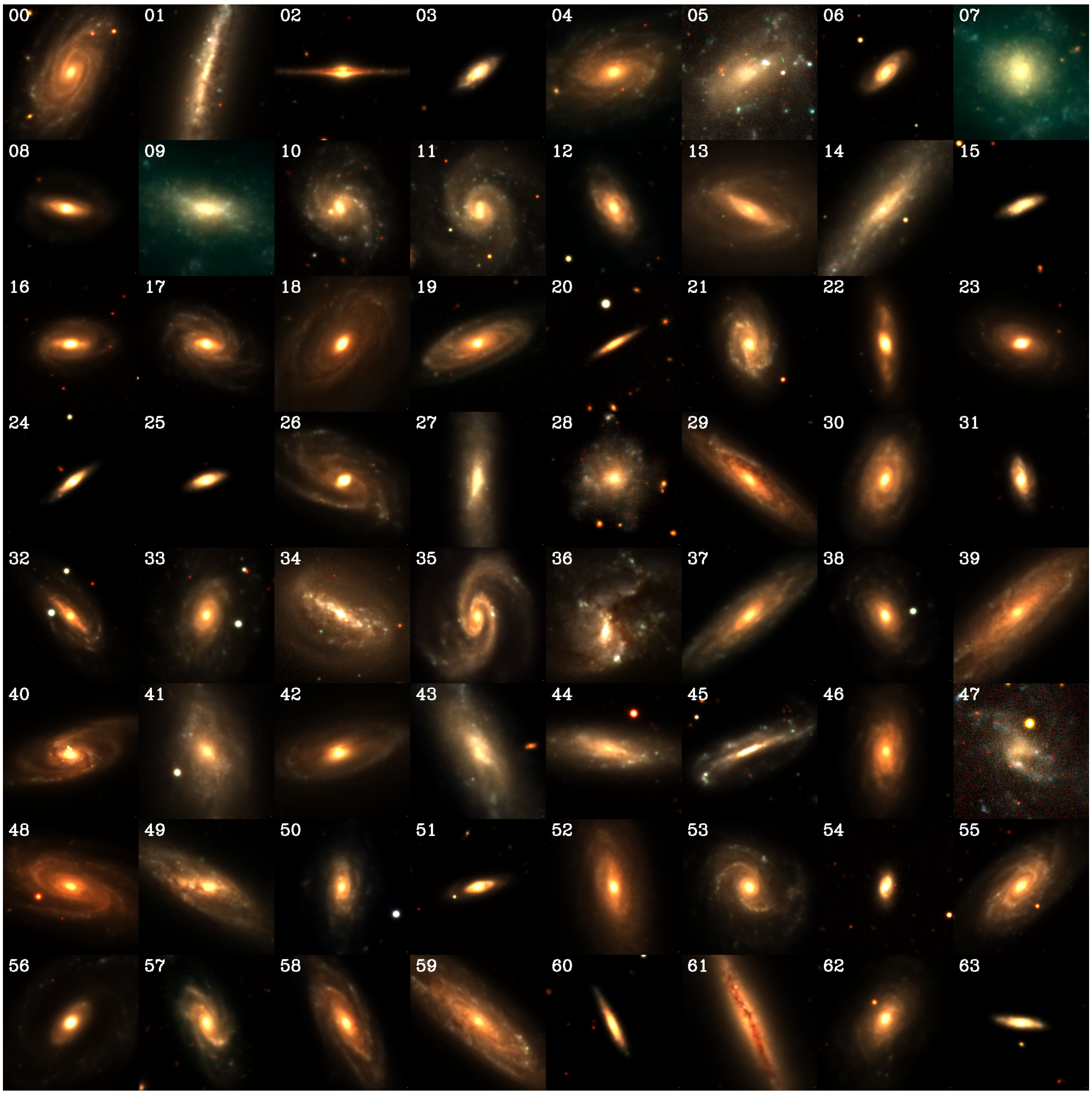


Figure 4.2: A montage of generated galaxies designed to mimic the PROBES data set, interspersed with real examples from the dataset itself. The images have been shuffled and the synthetic/real data split is 50/50. All images are *grz* RGB composites with identical scaling (we have performed a 99.5% percentile clip to better show low surface brightness features). A key stating which galaxies are real and which are generated is provided at the end of this chapter. More generated galaxies can be found at `http://mjjsmith.com/thisisnotagalaxy`.

divergences at each iteration step[35]:

$$\begin{aligned}\mathscr{L}_{\text{ELBO}} = \mathbb{E}_q\Big[& D_{\text{KL}}(q(\mathbf{x}_T \mid \mathbf{x}_0)||p(\mathbf{x}_T)) + \\ & \sum_{t>1} D_{\text{KL}}(q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0)||p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)) + \log p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1)\Big].\end{aligned} \tag{4.2}$$

In the Ho, Jain, and Abbeel (2020) formulation, the first term in Eq. 4.2 is a constant during training and the final term is modelled as an independent discrete decoder. This leaves the middle summation. Each summand can be written as

$$\mathscr{L}(\boldsymbol{\mu}_t, \boldsymbol{\mu}_\theta) = \frac{1}{2\sigma_t^2}||\boldsymbol{\mu}_t(\mathbf{x}_t, \mathbf{x}_0) - \boldsymbol{\mu}_\theta(\mathbf{x}_t, t)||^2, \tag{4.3}$$

where $\boldsymbol{\mu}_\theta$ is the neural network's estimation of the forward process posterior mean $\boldsymbol{\mu}_t$. In practice it would be preferable to predict the noise addition in each iteration step ($\mathbf{z}_t$), as $\mathbf{z}_t$ has a distribution that by definition is centred about 0, with a well defined variance. To this end we can define $\boldsymbol{\mu}_\theta$ as

$$\boldsymbol{\mu}_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\mathbf{z}_\theta(\mathbf{x}_t, t)\right), \tag{4.4}$$

and by combining Eqs. 4.3 and 4.4 we get

$$\begin{aligned}\mathscr{L}(\mathbf{z}_t, \mathbf{z}_\theta) &= \frac{1}{2\sigma_t^2}\left\|\frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\mathbf{z}_t\right) - \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\mathbf{z}_\theta(\mathbf{x}_t, t)\right)\right\|^2 \\ &= \frac{(1-\alpha_t)^2}{2\sigma_t^2\alpha_t(1-\bar{\alpha}_t)}||\mathbf{z}_t - \mathbf{z}_\theta(\mathbf{x}_t, t)||^2.\end{aligned} \tag{4.5}$$

Ho, Jain, and Abbeel (2020) empirically found that a simplified version of the loss described in Eq. 4.5 results in better sample quality. We therefore use a simplified version of Eq. 4.5 as our loss, and optimise to predict the noise required to reverse a forward process iteration step:

$$\mathscr{L}(\mathbf{z}_t, \mathbf{z}_\theta) = ||\mathbf{z}_t - \mathbf{z}_\theta(\mathbf{x}_t,\, t)||^2, \tag{4.6}$$

where $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\mathbf{z}_t$.

By recognising that $\mathbf{z}_t = \sigma_t^2 \nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t \mid \mathbf{x}_{t-1})$, we see that Eq. 4.6 is equivalent to denoising score matching over t noise levels (Vincent 2011). This connection establishes a link between DDPMs and other SBGMs (such as Song and Ermon 2019;

[35] See Appendix B in Sohl-Dickstein et al. (2015) and Appendix A in Ho, Jain, and Abbeel (2020) for the full derivation.

Jolicoeur-Martineau et al. 2020; Song and Ermon 2020).

Here we use a modified U-Net as p_θ (Ronneberger, Fischer, and Brox 2015; Salimans et al. 2017), and train via the Adam optimiser (Kingma and Ba 2015). The U-Net comprises of three downsample blocks, a bottleneck block, and three upsample blocks. Each downsample block comprises of two residual blocks (He et al. 2015; Srivastava, Greff, and Schmidhuber 2015), a self-attention layer (Bahdanau, Cho, and Bengio 2014; Cheng, Dong, and Lapata 2016), and a strided convolution layer. The bottleneck comprises of a self-attention layer sandwiched by two residual blocks. Each upsample block comprises of two residual blocks, a self-attention layer, and a transposed convolution layer. As in a standard U-Net, residual connections link the downsample and upsample blocks. To provide information about the current iteration step, an embedding representing the reverse process iteration step is periodically injected into the U-Net via a summation. Mish activation is used throughout (Misra 2019). The full implementation is released under the AGPLv3 licence and is available at `https://github.com/Smith42/astroddpm`.

To run inference for the reverse process, we progressively remove the predicted noise $\mathbf{z}_\theta$ from our image. The predicted noise is weighted according to our variance schedule:

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \mathbf{z}_\theta(\mathbf{x}_t, t) \right) + \boldsymbol{\sigma}_t \mathbf{z}.$$

If we take $p(\mathbf{x}_T) \sim \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbb{1})$, we can use p_θ to generate entirely novel data that are similar, but not identical to, those found in the training set. We can also use p_θ to perform image domain translation, and inpainting. Section 4.4 describes these applications in further detail.

4.3 Simulating DESI galaxy images

We train our models on minimally processed native resolution (256 × 256 pixels at $0.262''$ pixel^{-1}) Dark Energy Spectroscopic Instrument (DESI; Dey et al. 2019) Legacy Survey Data Release 9 galaxy imagery. The g, r, and z band images have an average atmospheric seeing of approximately $1''$.

4.3.1 Data sample, preparation, and training

We train two models on two different datasets for 750,000 global steps each across three NVIDIA Tesla V100 GPUs, corresponding to ∼250 hours wall time per model. We fill all available VRAM and set the batch size to 56. The two datasets are described below.

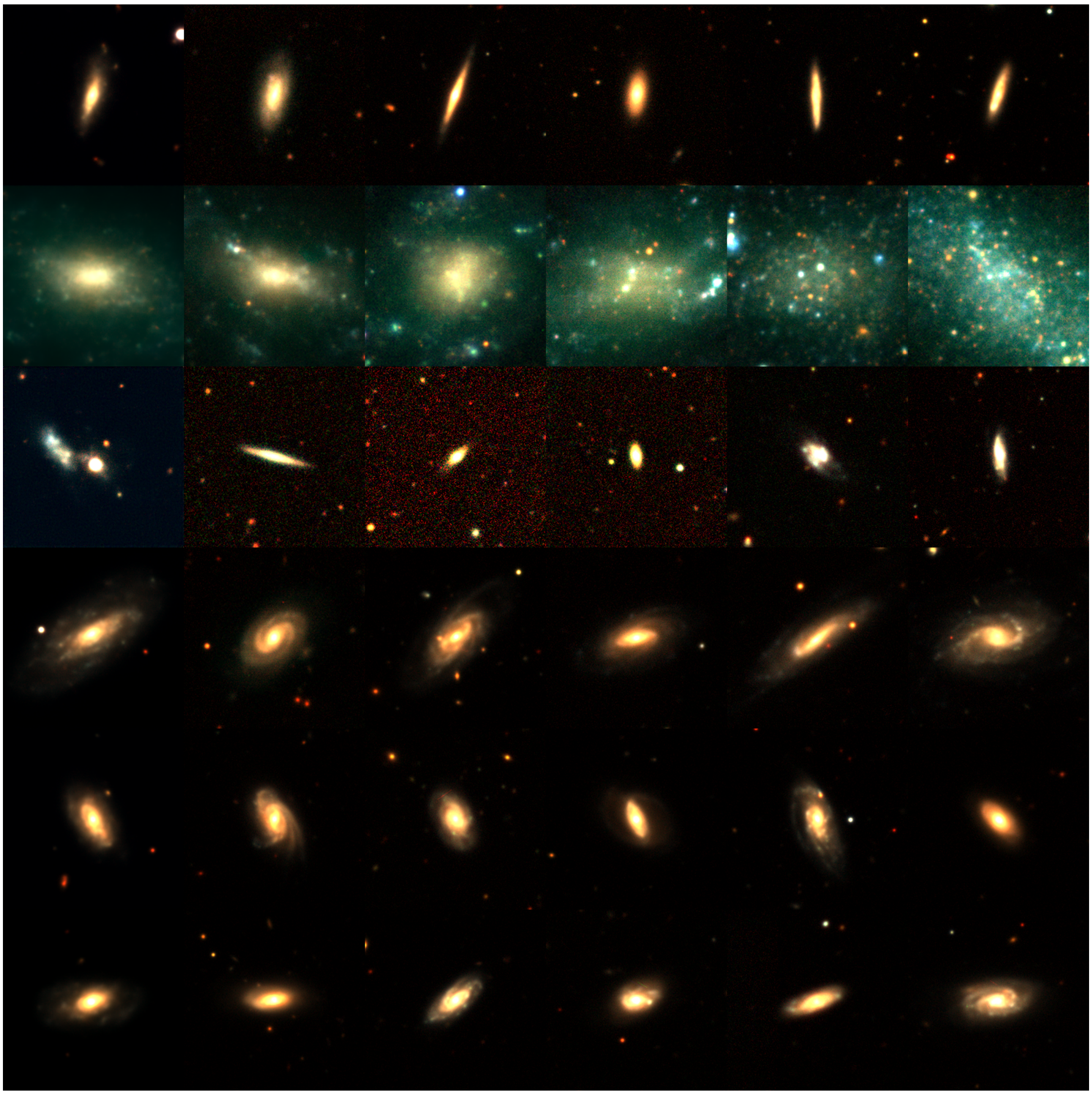


Figure 4.3: Pixelspace nearest neighbour to generated PROBES galaxies. The leftmost column shows a galaxy generated with the model $p_\theta(\mathbf{z})$, the other columns show that galaxy's closest training set neighbours in pixelspace. Moving along a row takes us further away from the simulated galaxy in pixelspace.

1. We train on the Photometry and Rotation curve OBservations from Extragalactic Surveys (PROBES) galaxy dataset as imaged by the Dark Energy Spectroscopic Instrument (DESI; Dey et al. 2019) Legacy Survey Data Release 9. The PROBES dataset is described in Stone and Courteau (2019) and Stone, Courteau, and Arora (2021). It contains 1962 late-type galaxies with no large neighbours or other obscuring features (such as bright stars). Most of the objects are well resolved, exhibiting spiral arms, bars, and other features characteristic of late-type systems. The model trained on this dataset produces galaxies that obviously exhibit internal structure. We refer to this as the 'PROBES' dataset.

2. We also train on a dataset of 306 006 galaxies whose coordinates are taken from Sloan Digital Sky Survey (SDSS; York et al. 2000) Data Release 7 (Abazajian et al. 2009) and a modified catalogue from Wilman, Zibetti, and Budavári (2010). This volume complete sample has an r-band absolute magnitude limit of $M_r \leq -20$ and a redshift limit of $z \leq 0.08$. See Arora et al. (2019) for details. This catalogue covers a wide range of environments from clusters to groups and field systems. As in the PROBES dataset, the galaxy images are taken from DESI (Dey et al. 2019). We use this dataset and the corresponding trained model to compare population level galaxy statistics (Sec. 4.4.1). For brevity we refer to this as the 'SDSS' dataset.

All images are cropped about the target galaxy to a shape of 256 × 256 pixels. The only destructive pre-processing performed is a upper and lower percentile clipping, with the percentiles calculated across the entire dataset. This clipping removes any 'hot' or 'cold' pixels. To calculate the upper flux truncation point we evaluate the 99.9th percentile fluxes for each galaxy across the full dataset. To separate the long tail from the bulk of the data, we fit a two-cluster k-means (Lloyd 1982). The two-cluster k-means returns a boundary at approximately 5 analogue-to-digital unit (ADU) for the SDSS dataset, and 5.5 ADU for the PROBES dataset, and so we set these values as our upper truncation points and normalisation constants. The lower flux truncation point is set as the minimum off-source pixel-wise root mean square across the entire dataset. We found this value to be very close to zero across all bands in both the SDSS and PROBES datasets, and therefore set the lower flux truncation as 0 ADU. We apply a min-max normalisation to the images with the following equation:

$$\bar{\mathbf{x}} = \frac{2 \cdot \max(0, \min(\mathbf{x}, A))}{A} - 1, \tag{4.7}$$

with $A = 5.0$ ADU being the upper flux truncation for the SDSS dataset, and $A = 5.5$ ADU for the PROBES dataset. We reverse this normalisation when post-processing inferrals from the model.

4.4 Results

Figure 4.2 shows a random selection of generated galaxies, alongside a random selection of real galaxies. The images are shuffled and we can see that the simulated and real galaxies are subjectively indistinguishable, at least to the authors (we of course invite the reader to make their own assessment of fidelity, referring to the answer key given at the end of this chapter). Figure 4.3 presents a random selection of generated galaxies' nearest neighbours in pixelspace. Since the pixelspace search does not return identical galaxies, we conclude that the DDPM is not simply regurgitating imagery, and is indeed generating novel data. We found a systematic offset in the simulated pixel fluxes and corrected for it in post-processing. To estimate the offset, we calculated the median pixel value in each of the 10,000 mock and 10,000 real galaxy observations. Each set is sorted and the medians paired according to their place in the sorted sets. Finally we fit a linear function to the resulting 10,000 median flux pairs. The gradient of the fit was used as a scaling factor for the simulated galaxy images. We found the multiplier to be 1.18 in g, 1.16 in r, and 1.23 in z for our DESI observations. Unfortunately, the exact cause of these offsets could not be determined. We propose that this discrepancy is due to a fundamental property of the neural network and its interaction with sparse imagery such as our galaxy images.

4.4.1 Quantifying similarity

To quantify the similarity of the visual and morphological characteristics of our galaxies, we borrow from the deep generative learning literature and calculate the Fréchet Inception Distance (FID; Heusel et al. 2017; Seitzer 2020). The FID is the distance (Dowson and Landau 1982) between Gaussians fitted to two Inception-v3 (Szegedy et al. 2016) penultimate layer feature representations. The penultimate layer nodes are deep in the network and mimic a human's perception when viewing images. Therefore, if the Gaussians are similar (and the corresponding FID is small), the images will be visually similar too.

We run FID on 10,000 random samples and present the results in Table 4.1. While we cannot yet contextualise our FID within the literature, we provide the value for future comparison. Figure 4.2 is presented for a basic visual and morphological comparison; we cannot discern between the synthesised and real galaxies, which suggests that the visual and morphological characteristics of our datasets are well replicated.

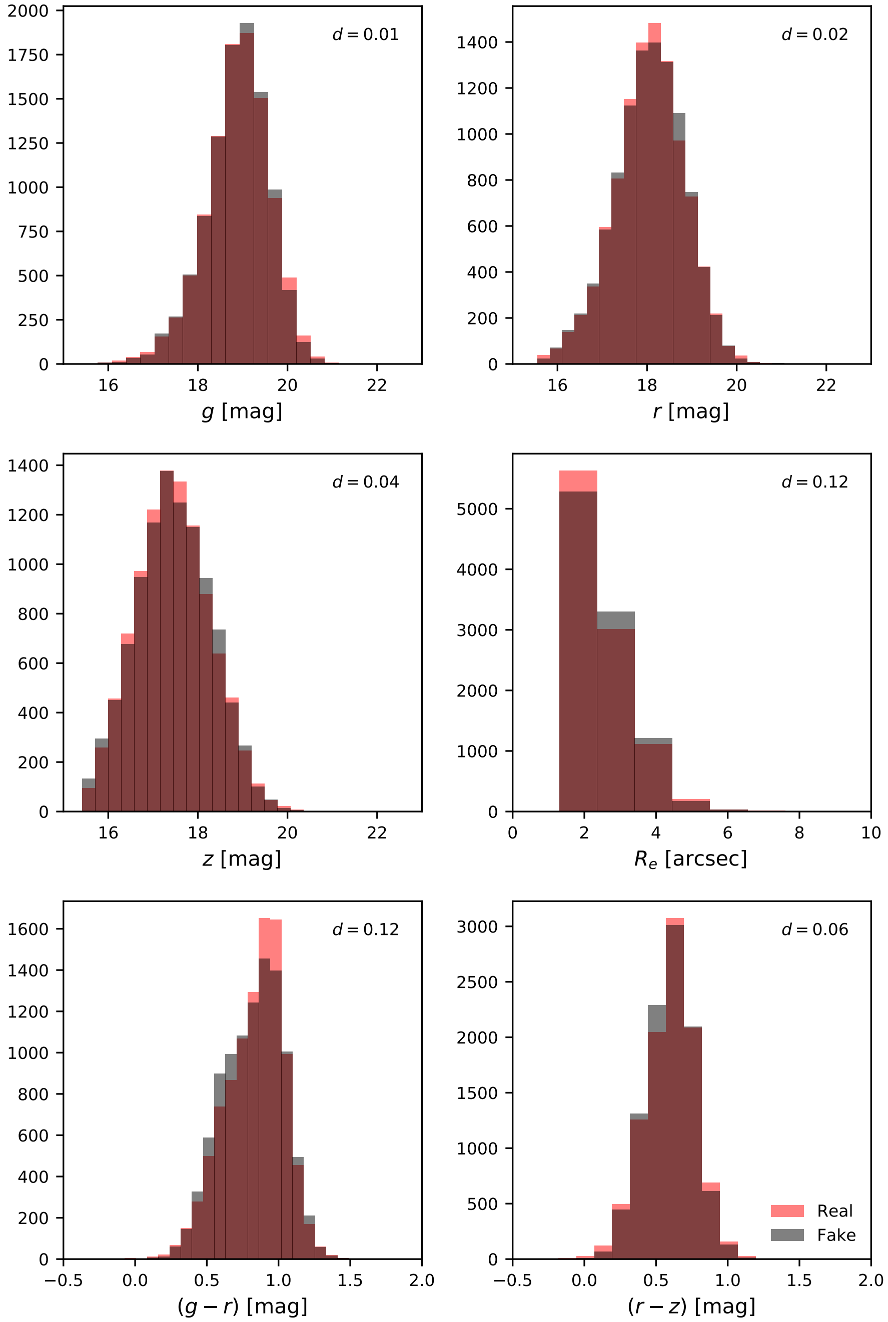


Figure 4.4: Histogram comparison between galaxies generated by the SDSS DDPM, and galaxies contained in the SDSS training set. The half light radius histogram follows a lognormal distribution, as do the magnitude and colour histograms in flux space. Therefore, we can calculate Cohen's d effect size for each histogram pair. As a rule of thumb, if $d \leq 0.2$ the effect size is considered 'small' and a sign of negligible difference (Cohen 1988).

Table 4.1: Wasserstein-1 distance between emergent property distributions. $p_\theta(\mathbf{z})$ is the DDPM described in this chapter. 'SDSS' is a comparison between two different randomly selected sets of 10,000 galaxies from the training set. We provide the 'SDSS' Wasserstein-1 distances as a baseline 'perfect' inference.

	W_g	W_r	W_z	W_{R_e}	$W_{(g-r)}$	$W_{(r-z)}$	SGD	FID
$p_\theta(\mathbf{z})$	0.013	0.012	0.023	0.055	0.015	0.010	0.127	19
SDSS	0.008	0.010	0.014	0.018	0.006	0.004	0.060	0.95

To demonstrate that we capture emergent, measurable properties of the galaxies, we directly compare size and flux distributions. Fluxes are measured via a summation within a fixed aperture with a diameter of 12 pixels (∼3″), and we use the half light radius as a simple measure of size. To summarise the distance between the 'ground truth' photometry training set properties and the properties of the simulated set we use the Wasserstein-1 distance[36]:

$$W(u,v) = \int_{-\infty}^{\infty} |U - V| \tag{4.8}$$

where U and V are the respective cumulative distribution functions of u and v.

Following Eq. 4.8, we propose a 'synthetic galaxy distance' metric that captures the difference between emergent properties of a synthetic and reference galaxy photometry dataset:

$$\begin{aligned}\mathrm{SGD} &= \sum_i W(u_i, v_i),\\ &= W(R_e^u, R_e^v) + W(g^u, g^v) + W(r^u, r^v) + W(z^u, z^v) +\\ &\qquad W((g-r)^u, (g-r)^v) + W((r-z)^u, (r-z)^v),\end{aligned} \tag{4.9}$$

where R_e is the half light radius, and g, r, and z are aperture magnitudes in specific bands and u and v denote different datasets. The SGD returns a single number, where a lower value denotes a closer match between u and v. When combined with the FID for visual and morphological similarity, a good overview of the similarity between two large galaxy photometry datasets is obtained. Figure 4.4 shows the results for the individual tests and the SGD summary is in Table 4.1. We run SGD on 10,000 random samples.

[36] Since we are dealing with large datasets a Kolmogorov-Smirnov (KS) test is not appropriate as it becomes overpowered with a very large sample size. We instead use the related Wasserstein-1 distance to provide an absolute value that represents the difference between our distribution pairs, and also calculate Cohen's d effect size as a direct intuitive substitution for the p-values that would otherwise result from KS tests (Figure 4.4).

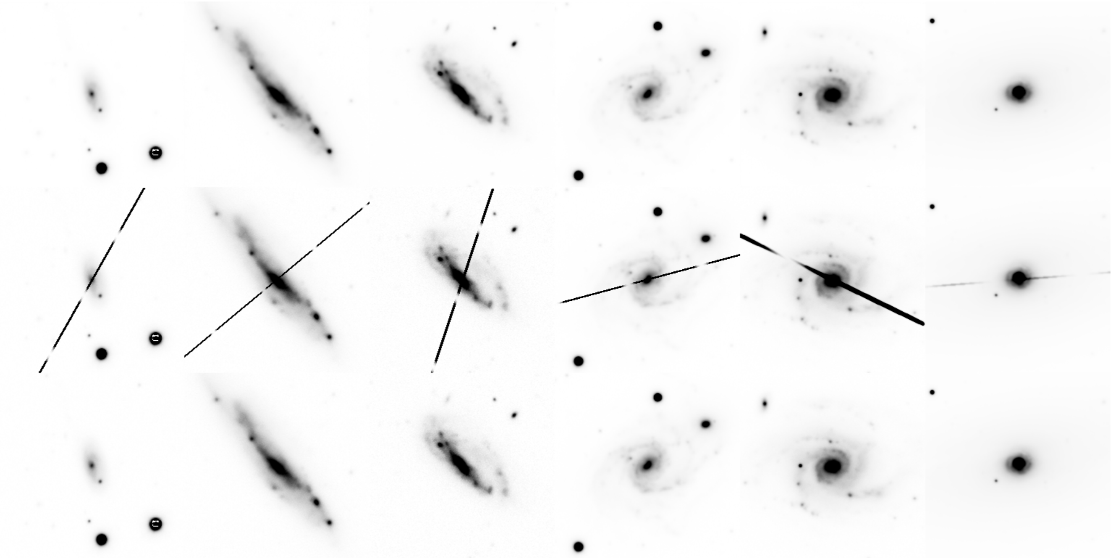


Figure 4.5: Inpainting of galaxy observations defaced by satellite trails. The first row shows the original (r-band) PROBES galaxy, $\mathbf{x}$. The second row shows the defaced galaxy, $\bar{\mathbf{x}}$. The third row shows a random guided draw from the model $p_\theta(q(\bar{\mathbf{x}}; T = 950))$.

While we cannot yet contextualise our SGD within the literature, we provide the value for future comparison. We also present Figure 4.4 to otherwise show that our model captures physical properties of the galaxies. We calculate Cohen's d effect size for each histogram pair, and in all cases find $d \leq 0.2$ indicating a 'small' or negligible effect (Cohen 1988). Cohen's d effect size is defined here as

$$d = \left| \sqrt{\frac{2}{\sigma_u^2 - \sigma_v^2}} \left(\mu_u - \mu_v\right) \right|,$$

where μ is the mean of the fitted distribution, and σ is the standard deviation. The subscripts u and v denote different datasets.

4.4.2 Satellite trail removal via guided diffusion

The DDPM can be used to remove simulated galaxy satellite trails from images. We simulate satellite trails by superimposing a bright linear strip onto a real image. The strips have a random direction, brightness, width, and periodicity. In this demonstration we present monochrome r-band images from the PROBES dataset.

To perform guided diffusion, we run the reverse process on the occluded part of the galaxy, in this case the satellite trails. The other image pixels are drawn directly from the forward process, and so are not updated. The occluded pixels are updated with guidance information from the surrounding pixels. As Figures 4.5 and 4.6 show,

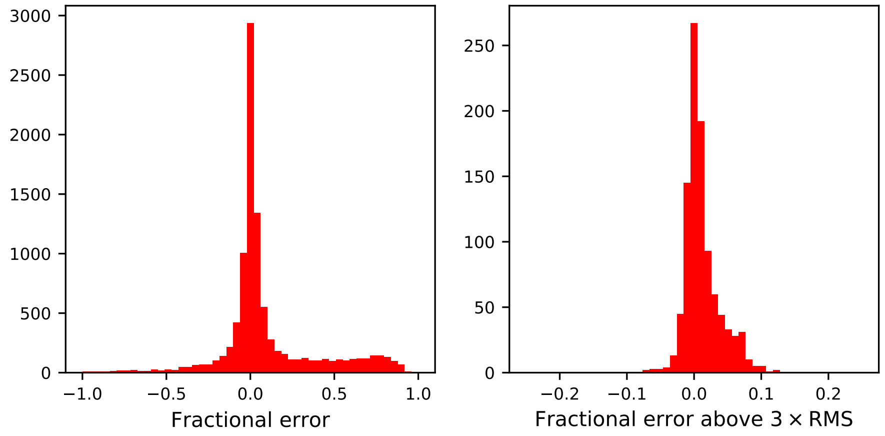


Figure 4.6: A pixelwise comparison between the ground truth images' occluded fluxes, and the recovered images' in Figure 4.5, expressed as a fractional error. The left panel shows the residual for all pixels, and the right panel restricts the analysis to pixels exceeding 3 times the background r.m.s. (0.654 ADU). We find good agreement between the predicted pixel fluxes and the ground truth fluxes, with virtually all significant pixels within 10% of their true values.

this process retrieves excellent representations of the original galaxies, essentially in-painting the missing data with high accuracy. Figure 4.6 shows that significant ($>3\sigma$) pixels that were occluded by a trail have recovered fluxes within 10% of their true values. Since satellite trails are not present in the training set, a guided draw from the learnt model is effective at 'interpolating' the occluded pixels. A similar approach would work for other unwanted artefacts, such as glints and ghosts, provided they do not appear frequently in the training set.

4.4.3 Domain transfer

The DDPM can also be used to make another input image resemble a DESI Legacy Survey observation. To perform this domain transfer, we first run the forward process for T iterations. We then take the noisy image, and run the reverse process. This results in a DESI Legacy Survey-like observation that shares high level features with the input image. Figure 4.7 demonstrates this technique on cartoons, setting $T = 600$. If T is set at a high value, the DDPM produces an image that more closely resembles one that might be found in the training set. However, fine detail in the conditioning image is lost as it is erased by the forward noise addition process. The cartoon input is transformed into an image resembling it, but with the properties of a DESI survey image. Once the cartoon images have been 'DESI-fied', we can search for the nearest neighbour in pixelspace in the real dataset, and this is shown as a final column in Figure 4.7. Thus, this approach paves the way for pixel-based searching of large survey imaging databases. For example, one could potentially sketch a particular morphology or configuration (e.g. an Einstein ring), apply the model tailored to that survey and then recover the best match in the real data. One could also apply this technique to inject realism into simulated galaxies, such as those predicted by hydrodynamical simulations.

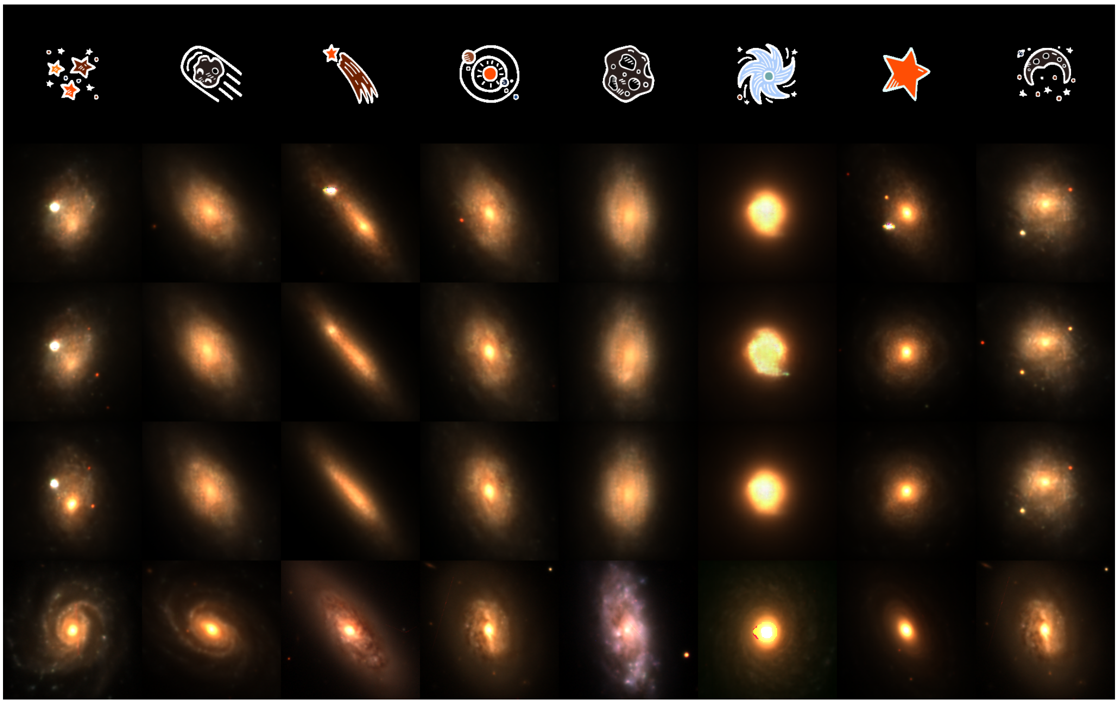


Figure 4.7: Cartoon images are made to look like DESI observations via the model $p_\theta(q(\mathbf{x}; T = 600))$. The first row shows the input image, the middle rows show random draws from the PROBES model, and the final row shows the pixelspace nearest neighbour to the generated images.

Preechakul et al. (2021) and Saharia et al. (2021) have both explored image-to-image translation with a score-based model. Preechakul et al. (2021) showed that DDPMs can produce semantically meaningful embeddings, given an appropriate architecture. In their paper, they demonstrate that their autoencoding DDPM can interpolate along the embedding space and age and de-age images of faces. In astronomy, one can imagine using a 'survey' embedding to interpolate between surveys.

Saharia et al. (2021) took a different approach and explicitly trained their model to reverse the forward process of an ill-posed inverse problem. For an ill-posed inverse problem such as noise addition, one can define the forward process in a classical way, and use a DDPM in the inverse process to retrieve the uncorrupted image. Saharia et al. (2021) did this and showed that a DDPM can colourise greyscale images, and remove JPEG compression artefacts.

For more difficult problems that do not have a well defined forward process, we can use a model similar to that introduced in Sasaki, Willcocks, and Breckon (2021) to translate between two different image domains. We intend to explore astronomy related image-to-image translation applications more deeply in follow up work.

4.4.4 A fun aside: mock Astronomy Picture of the Day

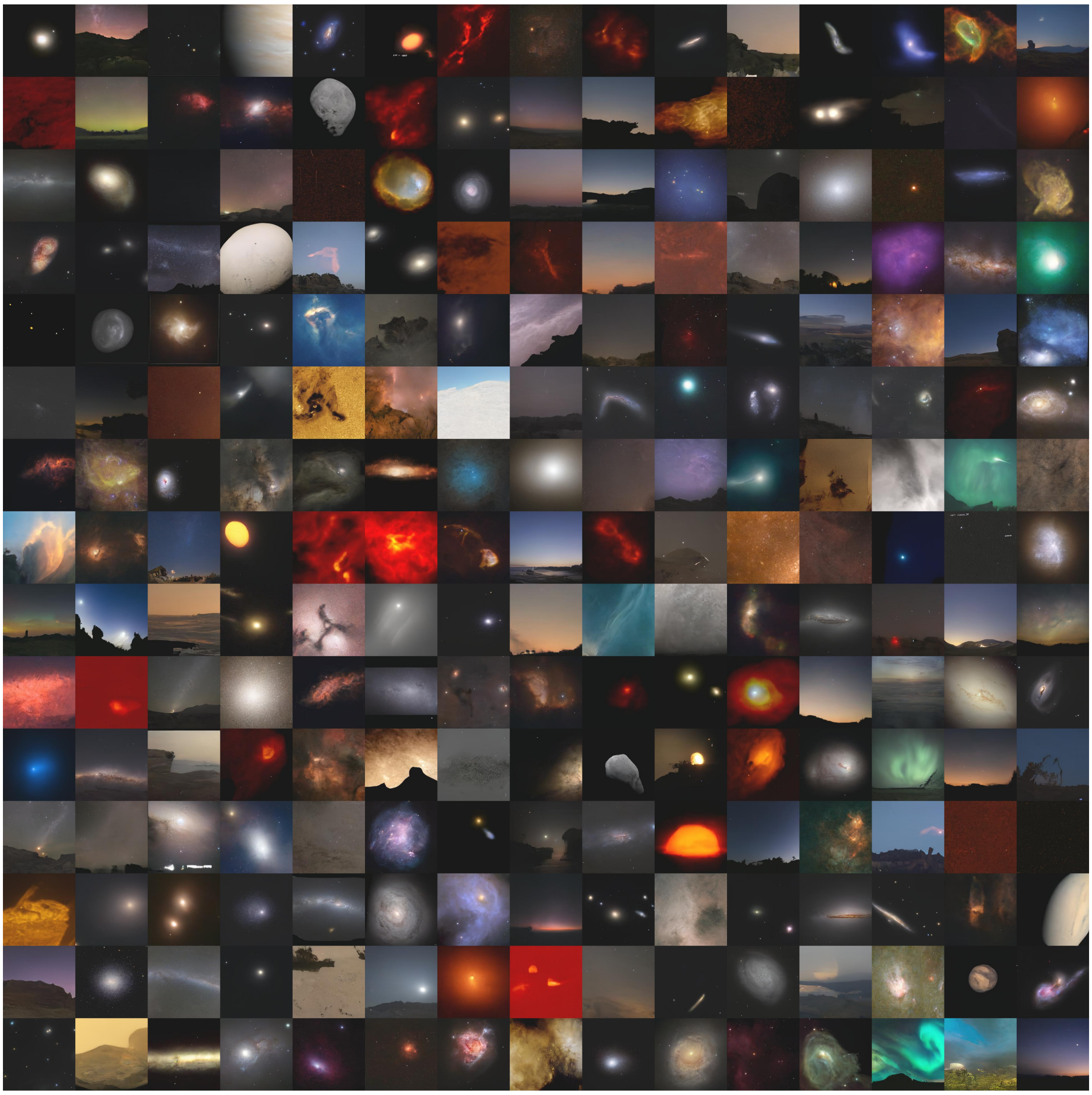


Figure 4.8: A sample of DDPM-generated APOD imagery: AI-APODs. More can be found at http://mjjsmith.com/thisisnotanapod and you can follow a Twitter bot https://twitter.com/ThisIsNotAnApod. A version of this figure has been featured on NASA's APOD https://apod.nasa.gov/apod/ap211109.html.

As a fun aside, we have trained a DDPM on images from NASA's Astronomy Picture of the Day (APOD) archive. The dataset comprises 11,428 RGB JPEG images resized to a 256 × 256 shape. We trained this model for 900,000 global steps on a single V100 GPU. This allows us to generate new APODs that do not actually exist. Figure 4.8 shows a curated sample of 'AI-APODs' generated from this model. We leave it to the reader to critically assess their merits, but some common themes are apparent: images resembling nebulae, galaxies, landscapes, moons, and aurorae are present. Random AI-APODs generated from the model can be found at `http://mjjsmith.com/thisisnotanapod`, and a Twitter bot will post images at `https://twitter.com/ThisIsNotAnApod`.

4.5 Conclusions

We show that score-based generative modelling is a viable method for synthetic galaxy image generation, and that this approach preserves emergent properties such as galaxy size and total flux over different photometric bands, in addition to producing realistic morphologies. We achieve this fidelity without explicit encoding of physics or instrumental effects, a great advantage when simulating physically ill-defined objects. This is a completely data-driven approach to synthetic data generation.

There are downsides. Naturally, SBGMs require significant computational resources to train. However, unlike most other generative deep learning methods SBGMs also require significant resources to infer data. Since they need to diffuse the data for T cycles[37], it takes T times longer to produce a batch of synthetic data compared to an equivalent GAN, VAE, or other single shot generative model. However, there may be routes to reduce the inference time for SBGMs, with promising results already (Jolicoeur-Martineau et al. 2021; Song et al. 2021).

SBGMs have clear astronomical applications, from object deblending (Jayaram and Thickstun 2020) to survey-to-survey translation (Sasaki, Willcocks, and Breckon 2021) to occluded object in-painting (Kadkhodaie and Simoncelli 2020; Song et al. 2021) to super-resolving imagery (Saharia et al. 2021). Unlike GANs, SBGMs do not suffer from mode collapse and are trivial to train. SBGMs produce imagery that has a diversity and fidelity that rivals state of the art GAN models (Dhariwal and Nichol 2021; Nichol and Dhariwal 2021; Song et al. 2021). Unlike VAEs, SBGMs do not constrain information to a fixed bottleneck vector and thus do not suffer from blurring, and instead produce sharp, realistic imagery (Spindler, Geach, and Smith 2020). For all these reasons, we believe that SBGMs are ripe for exploitation by the astronomical community, and we hope this chapter motivates further work in this topic.

[37] In this chapter, $T = 1000$.

Data and code availability. The full PyTorch (Paszke et al. 2019) implementation of the model presented here is available at https://github.com/Smith42/astroddpm, and the code to calculate the Synthetic Galaxy Distance (SGD) can be accessed at https://github.com/Smith42/synthetic-galaxy-distance. To calculate the Fréchet Inception Distance (FID), we used https://github.com/mseitzer/pytorch-fid.

Carbon emissions. The training of deep learning models requires considerable energy, contributing to carbon emissions. The energy used to train AstroDDPM to completion is estimated to be 450 kWh, corresponding to 105 kg CO2e. according to the Machine Learning Emissions Calculator described in Lacoste et al. (2019). To counteract further emission from redundant retraining, we follow the recommendations of Strubell, Ganesh, and McCallum (2019) and make available the fully trained models, as well as the code to run them. Also, we will make available trained models for any improvements that we make to AstroDDPM in the future.

Answer key for Figure 4.2. 00 Real, 01 Real, 02 Real, 03 Mock, 04 Real, 05 Real, 06 Real, 07 Mock, 08 Mock, 09 Mock, 10 Real, 11 Real, 12 Mock, 13 Mock, 14 Mock, 15 Mock, 16 Real, 17 Real, 18 Mock, 19 Real, 20 Real, 21 Real, 22 Real, 23 Mock, 24 Mock, 25 Mock, 26 Mock, 27 Mock, 28 Mock, 29 Real, 30 Mock, 31 Real, 32 Real, 33 Real, 34 Real, 35 Mock, 36 Real, 37 Mock, 38 Mock, 39 Mock, 40 Mock, 41 Mock, 42 Mock, 43 Mock, 44 Real, 45 Real, 46 Mock, 47 Real, 48 Real, 49 Real, 50 Mock, 51 Real, 52 Mock, 53 Real, 54 Real, 55 Real, 56 Real, 57 Real, 58 Mock, 59 Real, 60 Mock, 61 Mock, 62 Mock, 63 Mock.

Chapter 5

Astronomical foundation models: a fourth astroconnectionist wave?

The biggest lesson that can be read from 70 years of AI research is that general methods that leverage computation are ultimately the most effective, and by a large margin.

—Sutton (2019), *The Bitter Lesson*

We have seen in the preceding chapters that deep learning has already found much use in astronomy, a use predicated on the availability of enormous amounts of computational power and data. In §2 we explored a model that sidestepped a time consuming manual information extraction process by leveraging a large dataset via deep learning. In §5.1 and §5.2 we will follow this family of information extraction methods to their logical conclusion and discuss foundation models as astronomical information extractors. All of the techniques described in the following sections require an amount of computational power and data that is currently outside the reach of the average academic. In §5.2 I address this, and suggest actions that we can take as a community to remedy this issue. In §3 and §4 we explored two generative models repurposed as astronomical simulators. In §5.3 I suggest an extension to these methods that leverages a hypothetical astronomical foundation model. This extension would be capable of conditional image generation, paving the way for physically aware deep learning based astronomical simulations. Finally, in §5.4 I draw this thesis to an end with some closing remarks.

5.1 Foundation models

Since its inception, connectionism has followed a path of greater compute and greater generality (Sutton 2019; Branwen 2022). In that time, human crafted biases have fallen by the wayside, to be replaced with models and techniques that learn directly from data. Sutton (2019) exemplifies this process via the field of speech recognition:

> In speech recognition, there was an early competition, sponsored by DARPA, in the 1970s. Entrants included a host of special methods that took advantage of human knowledge—knowledge of words, of phonemes, of the human vocal tract, etc. On the other side were newer methods that were more statistical in nature and did much more computation, based on hidden Markov models (HMMs). Again, the statistical methods won out over the human-knowledge-based methods. This led to a major change in all of natural language processing, gradually over decades, where statistics and computation came to dominate the field. The recent rise of deep learning in speech recognition is the most recent step in this consistent direction. Deep learning methods rely even less on human knowledge, and use even more computation, together with learning on huge training sets, to produce dramatically better speech recognition systems. As in [computer Go and computer chess], researchers always tried to make systems that worked the way the researchers thought their own minds worked—they tried to put that knowledge in their systems—but it proved ultimately counterproductive, and a colossal waste of researcher's time, when, through Moore's law, massive computation became available and a means was found to put it to good use.

We are seeing this principal play out once again through a new paradigm shift in deep learning, where even the underlying neural network architecture does not matter. Previously, neural networks were adapted for a specific domain via inductive biases injected by researchers, such as convolutions for computer vision, and recurrence for language processing. Now we are seeing transformer networks (see §1.7 and Vaswani et al. 2017) competing[38] in all deep learning domains applied or otherwise: from language processing (Devlin et al. 2019; Brown et al. 2020)[39] to computer vision (Parmar et al. 2018; Dosovitskiy et al. 2020) to graph learning (Kim et al. 2022) to protein folding (Jumper et al. 2021) to astronomy (Donoso-Oliva et al. 2022;

[38] For now! It may be that network architecture does not matter all that much at scale, and that any sufficiently large neural network is adequate. If this is true, we will see the simplest (and most scalable) architectures win out. Although this theory has not yet been rigorously tested, we are currently seeing rumblings that suggest that this is the case (i.e. Bo 2021. Also see Footnote 10 for commentary on MLPs and transformers).

[39] These models are collectively known in the literature as large language models, or LLMs.

Morvan et al. 2022; Pan, Ting, and Yu 2022). The transformer's versatility allows us to take a model trained on one task and apply it to a similar yet different task, a process known as transfer learning. For example, we could train a model on the 'surrogate' task of predicting the next word in a sequence, and then apply that model to a similar yet different task of predicting the answer to a geography question. In this example the first model is known as a 'foundation' model, and the downstream model is derived from it. This set up brings with it some useful advantages. For example, if the foundation model is improved, all downstream tasks also see improvement. Therefore, the need for only one model allows researchers to pool their efforts in a way not possible when resources are split between many projects.

To train a foundation model, we first need to define a surrogate task. As labelled datasets are expensive, and raw data is cheap, the easiest and most scalable way to do this is via self-supervised learning[40] . Self-supervised learning does not require a human to provide a labelled dataset for training. Instead, the supervisory signal is generated automatically from the raw data. For example, in the context of astronomy this task could be predicting a masked value in a variable star's light curve (Donoso-Oliva et al. 2022). Another task could be using an autoencoder (§1.9.1) to replicate a galaxy observation (Spindler, Geach, and Smith 2020). A further task could be training within a self-supervised framework, like contastive learning (§1.10.1). The important thing about self-supervised learning is that it does not require annotated data. This means that we can leverage vast reserves of raw data (such as textbooks, scraped Internet text, raw imagery, etc.).

Very large models trained on vast amounts of data demonstrate surprising emergent behaviour. For instance, GPT-3 (Brown et al. 2020) is a 175B[41] parameter model that can be 'prompted' to perform a novel task (see Fig. 5.1 for more on prompting foundation models). This ability was not shown at all in GPT-3's older, smaller 1.5B parameter sibling (Radford et al. 2019). Furthermore, a meta-study described in Wei et al. (2022) found that larger models suddenly 'unlock' abilities such as arithmetic, translation, and understanding of figures of speech once they reach a certain scale. These findings suggest that architectural changes are not required beyond scaling to perform many tasks in natural language processing (Branwen 2022; Chowdhery et al. 2022). In Fig. 5.1 we see some results from Alayrac et al. (2022), a model comprising of an LLM, and an image encoder. In this figure we can see that the model is capable of arithmetic, reading, counting, and has a broad knowledge of art, geography and

[40] For more on self-supervised learning see §1.10.

[41] In this chapter I will abbreviate billion as B and trillion as T.

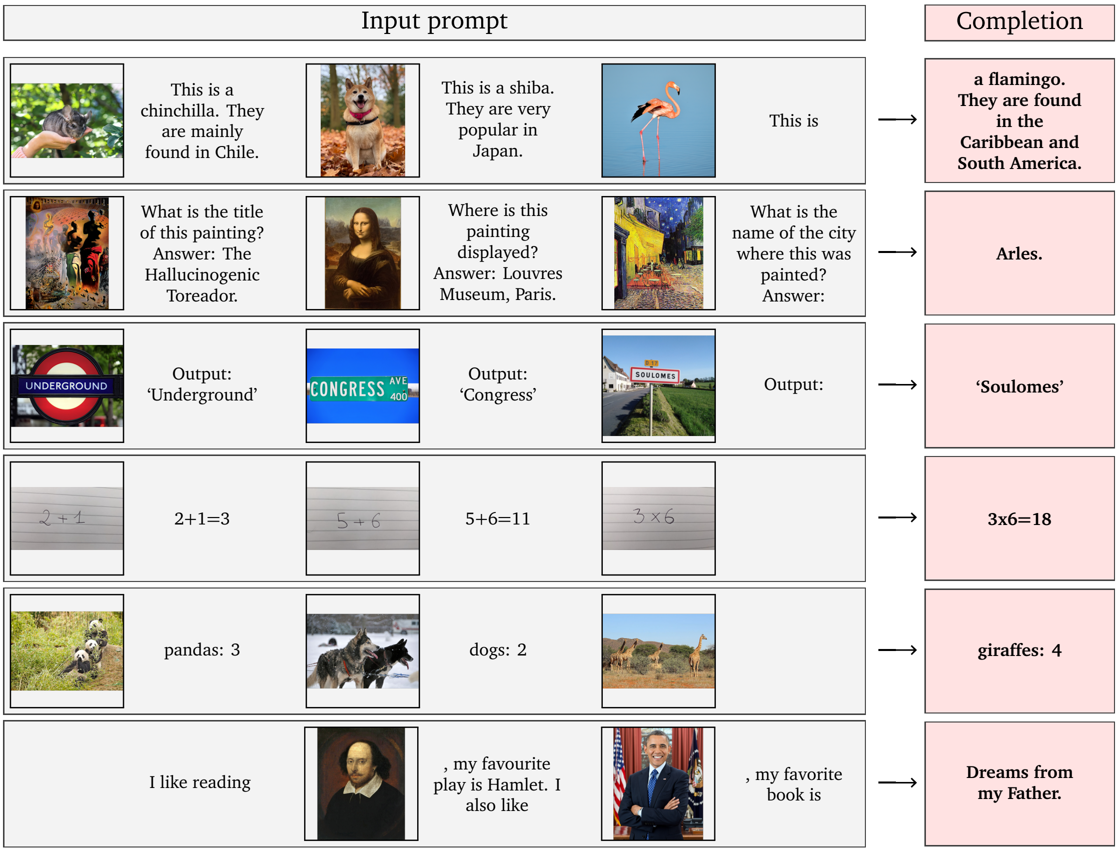

Input prompt
Completion
This is a chinchilla. They are mainly found in Chile.
This is a shiba. They are very popular in Japan.
This is
a flamingo. They are found in the Caribbean and South America.
What is the title of this painting? Answer: The Hallucinogenic Toreador.
Where is this painting displayed? Answer: Louvres Museum, Paris.
What is the name of the city where this was painted? Answer:
Arles.
UNDERGROUND
Output: 'Underground'
CONGRESS AVE 400
Output: 'Congress'
SOULOMES
Output:
'Soulomes'
2+1
2+1=3
5+6
5+6=11
3x6
3x6=18
pandas: 3
dogs: 2
giraffes: 4
I like reading
, my favourite play is Hamlet. I also like
, my favorite book is
Dreams from my Father.

Figure 5.1: Flamingo is a foundation model that is capable of understanding images within the context of natural language. Here we see some examples of Flamingo's emergent abilities. This figure is adapted from Fig. 1 in Alayrac et al. (2022).

zoology[42], and literature. This model is comprised of a ResNet variant (He et al. 2015; Brock et al. 2021) to encode imagery, and the Chinchilla LLM (Hoffmann et al. 2022) to encode and generate text. Chinchilla (and therefore Flamingo) was trained with the surrogate task of predicting the next word in a text sequence, and so none of the emergent properties stated above were explicitly optimised for.

In the next section, I will state and explain the need for an astronomical foundation model[43], not only for astronomy's sake, but also for the sake of openness in deep learning research.

[42] Interestingly, the authors of Flamingo first assumed that Flamingo's prediction of the species range of its eponymous bird was incorrect: flamingos are found in the Caribbean, South America, Africa, Europe, and South Asia. However, they later realised that the picture in Fig. 5.1 is of an *American* flamingo, which is specifically found in the Caribbean and South America, so the network was right after all! See Fig. A.1 for the full context.

[43] Walmsley et al. (2022) explore in a prelimiary study a 'galaxy foundation model' trained on Galaxy Zoo labels, and corresponding paired galaxy observations. They find that their pretraining is beneficial for training a network that performs a downstream task. However, they do not train a model of the scale required to exhibit emergent properties or task generalisability. These 'blessings of scale' require data and compute at a level that has not yet been seen within astronomical connectionism.

5.2 Scaling laws and data moats

Hoffmann et al. (2022) suggested an update to the foundation model scaling law first proposed in Kaplan et al. (2020). Their scaling law equation relates the size of a neural network model and the training dataset size to the minimum achievable loss. Mathematically, the equation is

$$\mathscr{L}_{\min}(N,D) = \underbrace{\frac{A}{N^{\alpha}}}_{\text{parameter term}} + \underbrace{\frac{B}{D^{\beta}}}_{\text{data term}} + \underbrace{E}_{\text{dataset entropy}}, \tag{5.1}$$

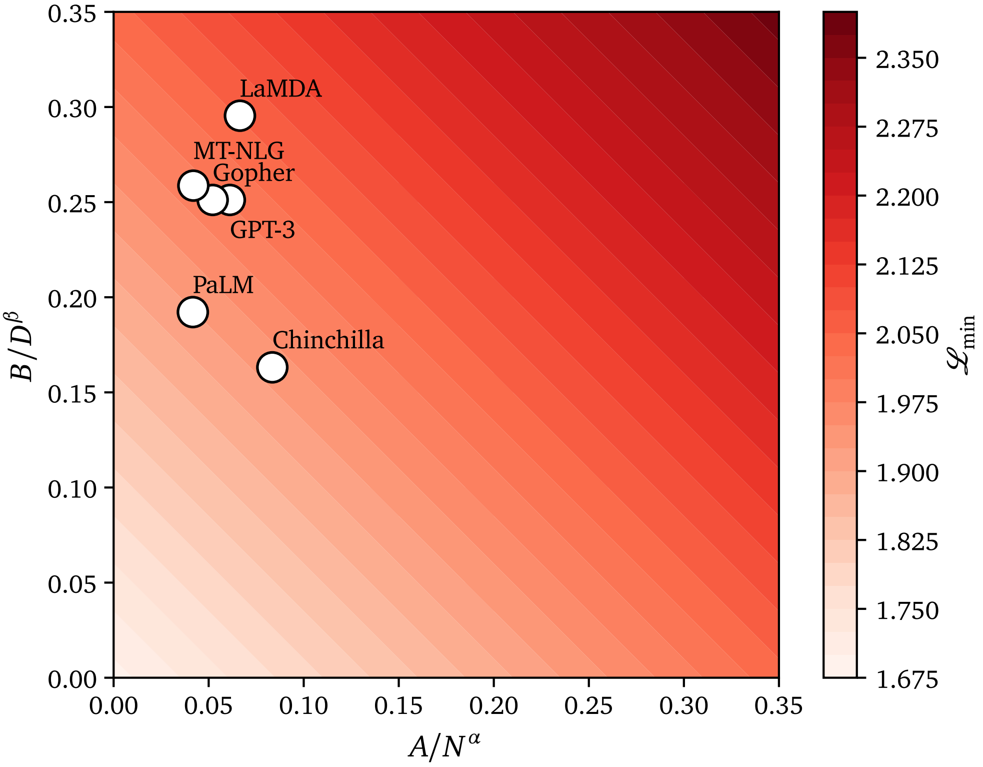
where E is a constant that represents the lowest possible loss, given a particular training dataset. N is the number of trainable parameters within the neural network, and D is the size of the dataset in tokens (see §1.7 for more about tokenisation). We can see that when we have an infinitely large model trained on an infinitely large dataset (i.e. $N = D = \infty$), the only term remaining is the 'dataset entropy' constant, E. We can therefore only reduce the loss by increasing the size of our model, or the size of our training set.

After fitting Eq. 5.1, Hoffmann et al. (2022) find

$$\mathscr{L}_{\min}(N,D) = \frac{406.4}{N^{0.34}} + \frac{410.7}{D^{0.28}} + 1.69.$$

If we then plug in N and D for a selection of real foundation models we arrive at Fig. 5.2. We can see in Fig. 5.2 that the model size term for real foundation models is far lower than the dataset size term. This means that an increase in dataset size has the potential to reduce the minimum loss by a far larger amount than a larger model would. Therefore, an obvious next step to improve these foundation models further is by increasing their dataset size.

The largest dataset (MassiveText-English; Hoffmann et al. 2022) in the comparison shown in Fig. 5.2 amounts to 1.4T tokens. However, this dataset is proprietary, being only available to researchers employed by Google. The largest public text dataset available at the time of writing is The Pile (Gao et al. 2020), with a total size of ∼260B tokens. We could increase the size of these datasets by indefinitely scraping text data from the surface web, but this data tends to be of low quality. Also, we have already exhausted some important high quality data reserves, like fundamental research papers, and open source code (Friel 2022). We also have to ask ourselves: what happens when generative models start to create data *en masse*, and dump it indiscriminately onto the Internet? If a significant proportion of text in a dataset scraped from the Internet is generated via a LLM, training on it will cause unforeseen



Model	N	D	A/N^{α}	B/D^{β}	$\mathscr{L}_{\text{min}}$
LaMDA (Thoppilan et al. 2022)	137B	168B	0.066	0.295	2.051
GPT-3 (Brown et al. 2020)	175B	300B	0.061	0.251	2.002
Gopher (Rae et al. 2021)	280B	300B	0.052	0.251	1.993
MT-NLG (S. Smith et al. 2022)	530B	270B	0.041	0.259	1.990
Chinchilla (Hoffmann et al. 2022)	70B	1.4T	0.083	0.163	1.936
PaLM (Chowdhery et al. 2022)	540B	780B	0.042	0.192	1.924

Figure 5.2: A comparison between the minimum losses of a selection of foundation models. The table above shows the number of parameters in a model (N), the number of tokens within that model's training set (D), and their corresponding calculated emergent terms from Eq. 5.1. Here we use Hoffmann et al. (2022) to source values for A, α, B, and β. The minimum loss for each model according to Hoffmann et al. (2022) is shown as $\mathscr{L}_{\text{min}}$. The contour plot shows the emergent parameters B/D^{β} and A/N^{α} plotted against each other for our models. The closer the models' scatterpoints are to the bottom left, the lower their minimum loss value.

issues and may ultimately result in a model with worse performance. We must therefore ensure that the data is not generated by a deep generative model. In addition to all this, the academy and the public at large will never have access to the vast reserves of data contained in the deep web administered by ByteDance, Google, Meta, Microsoft, and other tech giants. For all these reasons, we will need to think outside the box if we want to mine new high quality data.

Enter the multimodal foundation model. Reed et al. (2022)[44] demonstrated that a large Transformer neural network is capable of learning many tasks, from playing Atari, to captioning images, to chatting, to operating a real robot arm. The model shares weights across all tasks, and decides at inference time from context which task to predict. Importantly, Reed et al. (2022) find that their model follows the same scaling laws as other foundation models, and so multimodal foundation models have the same hunger for data that we see in Fig. 5.2. We can therefore augment (or replace) our text datasets with high quality, publicly available astronomical data.

The LSST's 189 16 megapixel CCDs will observe 1000 science frames per night (Ivezić et al. 2019). This amounts to 3×10^{12} pixels per night, or approximately 12B tokens a night if we use the same tokenising scheme as Dosovitskiy et al.'s vision transformer (Dosovitskiy et al. 2020). After only one year of observing, the LSST will have produced 4.4T tokens of raw data, larger than even the MassiveText-English dataset[45]. This data, and other astronomical data like it, could be compiled into a very large open dataset similar to EleutherAI's Pile (Gao et al. 2020). This dataset would provide a way for academics employed outside of Big Tech to train and research very large foundation models. Compiling a dataset like this would be difficult for a single relatively underresourced research group, but it could be accomplished via bazaar style open development (Raymond 1999). We have already seen this development model succeed in large open source projects, the most famous of which is the Linux kernel. This development model has also been shown to work within the field of deep learning by EleutherAI (e.g. Gao et al. 2020; Black et al. 2022; Crowson et al. 2022), and with HuggingFace's BigScience initiative[46]. Once compiled, we must ensure that progress is kept in the open, and that the data is not simply absorbed into proprietary datasets—to do this we must give our dataset a strong (viral) copyleft style licence.

[44] Earlier work from Kaiser et al. (2017) also demonstrated a deep learning model that could learn from disparate tasks, however Gato is the first model that achieves this while staying within a single deep learning paradigm.

[45] Of course, the reduced, useful data will be far smaller than our raw estimate here. The motivation behind this calculation is to show that even a single astronomical survey rivals the largest text dataset in size. A compilation of all useful astronomical data would certainly dwarf any contempory text dataset, whether public or proprietary.

[46] https://bigscience.huggingface.co

Once the dataset is compiled all we need are some self-supervised surrogate tasks for our 'astrofoundation' model to attempt. These tasks could include predicting the next observation in a variable star's time sequence, predicting the rotation curve of a given galaxy observation, predicting a galaxy's morphological parameters, or simply generating the next crop in a sequence of observations[47]. Our astrofoundation model will inherit all the interesting properties that LLMs enjoy, such as few to zero-shot generation and other emergent behaviours. We could also finetune astrofoundation to downstream tasks, saving much time and compute. In the next section I will outline one possible downstream task that would be useful in astronomy; a conditional generative model for galaxy simulation.

5.3 A new class of simulation

We explored in §3 and §4 the use of deep generative modelling to simulate astronomical observations. Both of these models were unconditional, and we could therefore not control their output. This is an issue if we want to generate specific classes of observations to train models for downstream tasks, such as redshift estimation, or galaxy type classification. To achieve a model capable of generating specific classes, one could simply train a conditional generative model of the form

$$G_\phi(\hat{\mathbf{x}} \mid \mathbf{z}, \mathbf{y}), \tag{5.2}$$

where $\hat{\mathbf{x}}$ is a generated image, $\mathbf{z}$ is some noise that acts to capture all detail not encoded in $\mathbf{y}$, and $\mathbf{y}$ is a conditioning vector. As an example, $\mathbf{y}$ could contain a galaxy's redshift or morphological type. However, this means that we must be very specific when choosing $\mathbf{y}$. Multimodal modelling allows us the means to sidestep this fundamental issue, and lets us play with fuzzy inputs.

As a thought experiment let us consider Google's recent 'Imagen' model[48], and imagine how it could be repurposed for an astronomical use case (Figs. 5.3 and 5.4; Saharia et al. 2022). Imagen is a combination of a frozen LLM (specifically T5-XXL; Tay et al. 2021) and a cascaded diffusion model (Ho et al. 2021, also see §1.9.3). The LLM acts as a language encoder, and then passes its generated latent space representations onto the diffusion model as a conditioning vector. If we were

[47] This is essentially training the model to act as a physics simulator. Viewing foundation models as world simulators is not unprecidented, and this perspective has already been explored in the simulation of thousands of 'social simulacra' within a model online community (Park et al. 2022).

[48] Naturally, no implementation is provided by Google. However, there is already an excellent MIT licenced implementation of Imagen provided by Phil Wang and others (https://github.com/lucidrains/imagen-pytorch). All we are missing now is the data and the means to train with it!

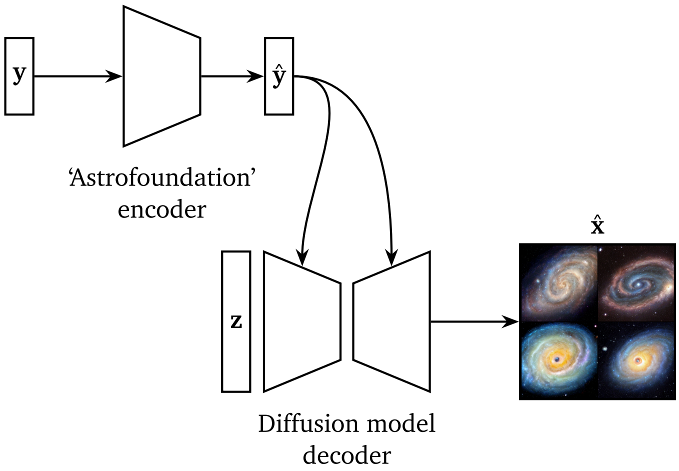
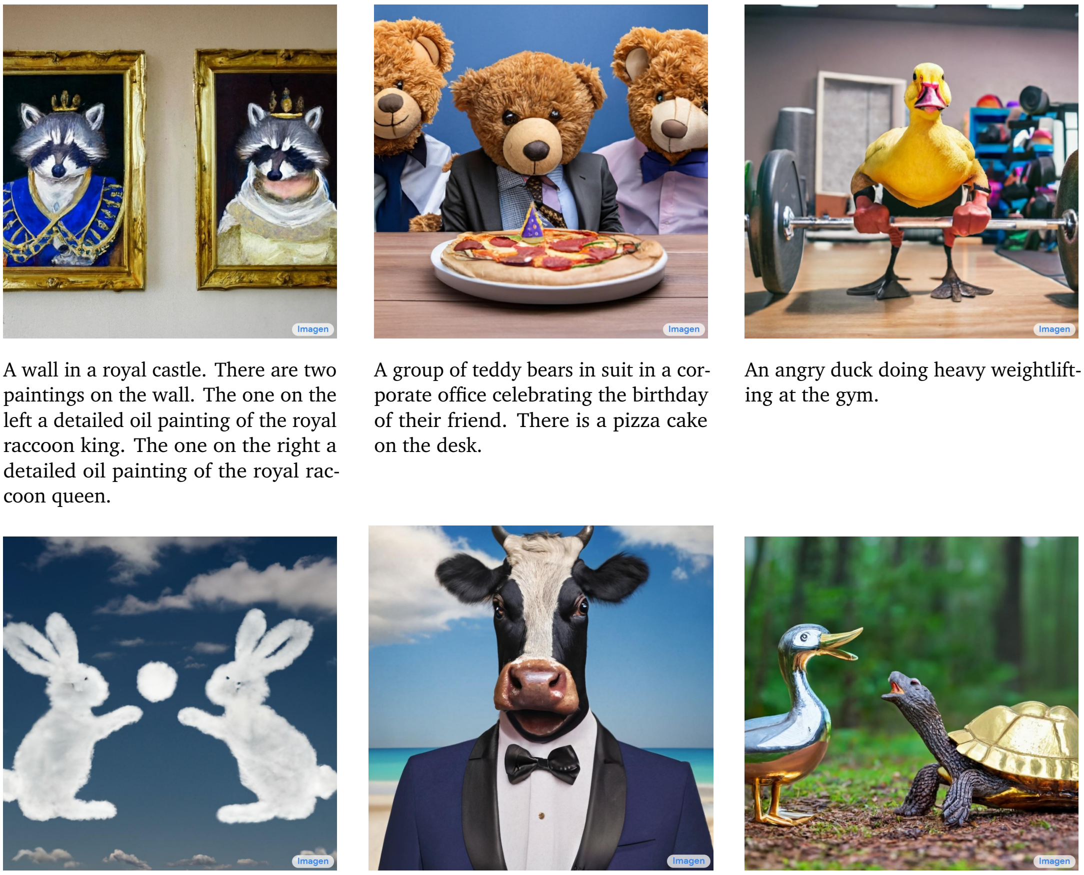


Figure 5.3: An Imagen-like model uses a frozen foundation model to encode text, and then uses that encoding to condition a cascaded diffusion model of the form $G_\phi(\hat{\mathbf{x}} \mid \mathbf{z}, \hat{\mathbf{y}})$ (Ho et al. 2021; Saharia et al. 2022). Here we see one possible realisation of this type of model in astronomy. **y** is some kind of descriptive vector that can be paired with a ground truth image. For example, **y** could be the surface brightness profile of a galaxy, or the summary statistics of a variable star light curve, or some cosmological parameters. In general, **y** could be any vector that the astrofoundation model understands. $\hat{\mathbf{y}}$ is **y**'s projected latent space equivalent. Since we do not need to train the foundation model here, training cost is far lower than for an equivalent end-to-end trained model.

to replace the frozen LLM with an 'astrofoundation' model (see §5.1 and §5.2), we could leverage astronomy's fundamentally multimodal nature. For example, if our astrofoundation model were trained to understand the Galaxy Zoo 2 (GZ2) morphological classifications (Willett et al. 2013), we could take the GZ2 descriptors as **y** and their corresponding galaxy pair as **x** and train on those.

Once trained, our astronomical Imagen model could generate synthetic galaxies that resemble the real galaxy observations that it was trained on. However, unlike the unconditional astronomical simulators described in §3 and §4, this model would be capable of generating galaxies that specifically resemble a real galaxy that shares the conditioning set of GZ2 parameters!

Unlike the conditional model described by Eq. 5.2, an astrofoundation type model allows us to be creative with the conditioning vector. For example, we could run the model in reverse to generate representations that refer to a very specific astronomical object, and then generate many more objects of that 'class' with injected features like satellite occlusion, a specific seeing, a specific redshift, etc. (see work on 'textual inversion' by Gal et al. 2022). We could even create a 'galaxy zoo' type dataset that asks citizen scientists to describe galaxy morphology via natural language. This is possible since the encoding foundation model does not fundamentally care about which form the caption takes. This approach would cut down on citizen scientist



A wall in a royal castle. There are two paintings on the wall. The one on the left a detailed oil painting of the royal raccoon king. The one on the right a detailed oil painting of the royal raccoon queen.

A group of teddy bears in suit in a corporate office celebrating the birthday of their friend. There is a pizza cake on the desk.

An angry duck doing heavy weightlifting at the gym.

A cloud in the shape of two bunnies playing with a ball. The ball is made of clouds too.

A photo of a person with the head of a cow, wearing a tuxedo and black bowtie. Beach wallpaper in the background.

A chrome-plated duck with a golden beak arguing with an angry turtle in a forest.

Figure 5.4: Select 1024 × 1024 Imagen samples generated from text inputs. Below each image is its corresponding conditioning text. Figure adapted from Fig. A.2 in Saharia et al. 2022.

training cost due to natural language's inherent intuitiveness.

5.4 Final comments or: how I learnt to stop worrying and love astronomy's Big Data Era

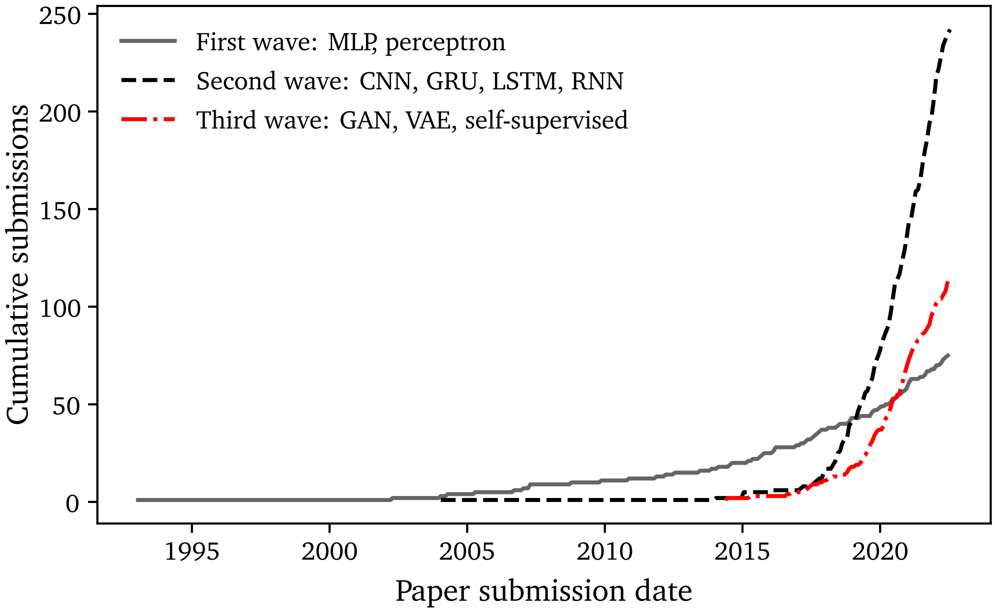


Figure 5.5: Here we see the number of arXiv:astro-ph submissions whose titles or abstracts match the terms given in the legend. We can see three distinct 'waves'. The first corresponds to studies that use MLPs (§1.1-§1.3), the second corresponds to studies that use 'deep learning' methods that injest raw data (§1.4-§1.8) and the third corresponds to studies that use generative or self-supervised models (§1.9-§1.11). The raw data is in the public domain, and is available at `https://www.kaggle.com/Cornell-University/arxiv`.

In every field that deep learning has infiltrated we have seen a reduction in the use of specialist knowledge, to be replaced with knowledge automatically derived from data (Sutton 2019). We have already seen this process play out in computer Go (Silver et al. 2016), protein folding (Jumper et al. 2021), natural language processing (Brown et al. 2020), and computer vision (Dosovitskiy et al. 2020). There is no reason to believe that astronomy is fundamentally different. Indeed, within this thesis we have seen a narrative pointing to this conclusion (Fig. 5.5). Initial work on MLPs within astronomy required manually selected emergent properties as input (e.g. Angel et al. 1990; Odewahn et al. 1992). With the advent of CNNs and RNNs, these manually selected inputs gave way to raw data ingestion (e.g. Dieleman, Willett, and Dambre 2015; Charnock and Moss 2017). Now we are seeing the removal of human supervision altogether with deep learning methods inferring labels directly from the data (e.g. Spindler, Geach, and Smith 2020; Morvan et al. 2022). Ultimately, if astronomy follows in the footsteps of other applied deep learning fields, we will see the removal of expertly crafted deep learning models, to be replaced with finetuned versions of an all-encompassing 'foundation' model (Bommasani et al. 2021). This process is by no means a bad thing; the removal of human bias in the astronomical discovery process allows us to find 'unknown unknowns' through serendipity (Sarmiento et al. 2021; Donoso-Oliva et al. 2022). Likewise, the ability to leverage data allows us to directly generate and interrogate realistic yet synthetic observations, sidestepping the need for an expensive and fragile classical simulation (Smith and Geach 2019; M. J. Smith et al. 2022).

Astronomy's relative data wealth gives us the opportunity to form a symbiotic relationship with the cutting edge of deep learning research, an increasingly data hungry field (Branwen 2022; Friel 2022). Many ultra-large datasets in machine learning are proprietary, and so astronomy as a whole could step in and provide a high quality multimodal public dataset. In turn, this dataset could be used to train an astronomical 'foundation' model that can be used for state-of-the-art downstream tasks (such as astronomical simulation, see §5.3).

Finally, following recent developments in connectionism (Brown et al. 2020; Hoffmann et al. 2022) most astronomers lack the resources to train models on the cutting edge of the field. If astronomy is to have any chance of keeping up with the Big Tech goliaths, we must follow the examples of EleutherAI and HuggingFace and pool our resources in a grassroots-style open source fashion (§5.2).

Bibliography

Abazajian, Kevork N. et al. (2009). 'THE SEVENTH DATA RELEASE OF THE SLOAN DIGITAL SKY SURVEY'. In: *The Astrophysical Journal Supplement Series* 182.2, pp. 543–558. DOI: 10.1088/0067-0049/182/2/543. URL: https://doi.org/10.1088/0067-0049/182/2/543.

Adorf, H. M. and M. D. Johnston (1988). 'Artificial neural nets in astronomy'. In: *Arbeitspapier der Gesellschaft für Mathematik and Datenverarbeitung*. Vol. 329. Arbeitspapier der Gesellschaft für Mathematik and Datenverarbeitung.

Ahn, C. P. et al. (2014). 'The Tenth Data Release of the Sloan Digital Sky Survey: First Spectroscopic Data from the SDSS-III Apache Point Observatory Galactic Evolution Experiment'. In: *Astrophysical Journal Supplement Series* 211.2, 17, p. 17. DOI: 10.1088/0067-0049/211/2/17. arXiv: 1307.7735 [astro-ph.IM].

Ahumada, R. et al. (2019). 'The Sixteenth Data Release of the Sloan Digital Sky Surveys: First Release from the APOGEE-2 Southern Survey and Full Release of eBOSS Spectra'. In: *arXiv e-prints*, arXiv:1912.02905, arXiv:1912.02905. arXiv: 1912.02905 [astro-ph.GA].

Aihara, H. et al. (2017). 'The Hyper Suprime-Cam SSP Survey: Overview and survey design'. In: *Publications of the Astronomical Society of Japan* 70.SP1. S4. ISSN: 0004-6264. DOI: 10.1093/pasj/psx066. eprint: https://academic.oup.com/pasj/article-pdf/70/SP1/S4/23692189/psx066.pdf. URL: https://doi.org/10.1093/pasj/psx066.

Aihara, H. et al. (2019). 'Second data release of the Hyper Suprime-Cam Subaru Strategic Program'. In: *Publications of the Astronomical Society of Japan* 71.6. 114. ISSN: 0004-6264. DOI: 10.1093/pasj/psz103. eprint: https://academic.oup.com/pasj/article-pdf/71/6/114/31516069/psz103.pdf. URL: https://doi.org/10.1093/pasj/psz103.

Akeret, J. et al. (2017). 'Radio frequency interference mitigation using deep convolutional neural networks'. In: *Astronomy and Computing* 18, pp. 35–39. DOI: 10.1016/j.ascom.2017.01.002. arXiv: 1609.09077 [astro-ph.IM].

Alayrac, J. et al. (2022). 'Flamingo: a Visual Language Model for Few-Shot Learning'. In: *arXiv e-prints*, arXiv:2204.14198, arXiv:2204.14198. arXiv: 2204.14198 [cs.CV].

Amiaux, J. et al. (2012). 'Euclid mission: building of a reference survey'. In: *Space Telescopes and Instrumentation 2012: Optical, Infrared, and Millimeter Wave*. Ed. by Mark C. Clampin et al. Vol. 8442. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, 84420Z, 84420Z. DOI: 10.1117/12.926513. arXiv: 1209.2228 [astro-ph.IM].

Andreon, S. et al. (Dec. 2000). 'Wide field imaging — I. Applications of neural networks to object detection and star/galaxy classification'. In: *Monthly Notices of the Royal Astronomical Society* 319.3, pp. 700–716. ISSN: 0035-8711. DOI: 10.1046/j.1365-8711.2000.03700.x.

Angel, J. R.P. et al. (1990). 'Adaptive optics for array telescopes using neural-network techniques'. English (US). In: *Nature* 348.6298, pp. 221–224. ISSN: 0028-0836. DOI: 10.1038/348221a0.

Aniyan, A. K. and K. Thorat (June 2017). 'Classifying Radio Galaxies with the Convolutional Neural Network'. In: *Astrophysical Journal Supplement Series* 230.2, p. 20. ISSN: 1538-4365. DOI: 10.3847/1538-4365/aa7333.

Aragon-Calvo, M. A. (Apr. 2019). 'Classifying the large-scale structure of the universe with deep neural networks'. In: *Monthly Notices of the Royal Astronomical Society* 484.4, pp. 5771–5784. ISSN: 0035-8711. DOI: 10.1093/mnras/stz393.

Arcelin, B. et al. (2021). 'Deblending galaxies with variational autoencoders: A joint multiband, multi-instrument approach'. In: *Monthly Notices of the Royal Astronomical Society* 500.1, pp. 531–547. DOI: 10.1093/mnras/staa3062. arXiv: 2005.12039 [astro-ph.IM].

Arora, N. et al. (2019). 'On the role of supermassive black holes in quenching star formation in local central galaxies'. In: *Monthly Notices of the Royal Astronomical Society* 489.2, pp. 1606–1618. DOI: 10.1093/mnras/stz2266. arXiv: 1908.04813 [astro-ph.GA].

Arora, N. et al. (2021). 'MaNGA galaxy properties - I. An extensive optical, mid-infrared photometric, and environmental catalogue'. In: *Monthly Notices of the Royal Astronomical Society* 505.3, pp. 3135–3156. DOI: 10.1093/mnras/stab1430.

Auld, T., M. Bridges, and M. P. Hobson (July 2008). 'cosmonet: fast cosmological parameter estimation in non-flat models using neural networks'. In: *Monthly Notices of the Royal Astronomical Society* 387.4, pp. 1575–1582. ISSN: 0035-8711. DOI: 10.1111/j.1365-2966.2008.13279.x.

Auld, T. et al. (Mar. 2007). 'Fast cosmological parameter estimation using neural networks'. In: *Monthly Notices of the Royal Astronomical Society: Letters* 376.1, pp. L11–L15. ISSN: 1745-3925. DOI: 10.1111/j.1745-3933.2006.00276.x.

Aussem, A., F. Murtagh, and M. Sarazin (Jan. 1994). 'Dynamical recurrent neural networks and pattern recognition methods for time series prediction: Application to seeing and temperature forecasting in the context of ESO's VLT astronomical weather station'. In: *Vistas in Astronomy* 38, pp. 357–374. ISSN: 0083-6656. DOI: 10.1016/0083-6656(94)90047-7.

Bahdanau, D., K. Cho, and Y. Bengio (2014). 'Neural Machine Translation by Jointly Learning to Align and Translate'. In: *CoRR* abs/1409.0473, arXiv:1409.0473. arXiv: 1409.0473 [cs.CL].

Bailer-Jones, C. A. L. et al. (Nov. 1997). 'Physical parametrization of stellar spectra: the neural network approach'. In: *Monthly Notices of the Royal Astronomical Society* 292.1, pp. 157–166. ISSN: 0035-8711. DOI: 10.1093/mnras/292.1.157.

Ball, N. M. et al. (2004). 'Galaxy types in the Sloan Digital Sky Survey using supervised artificial neural networks'. In: *Monthly Notices of the Royal Astronomical Society* 348.3, pp. 1038–1046. DOI: 10.1111/j.1365-2966.2004.07429.x. arXiv: astro-ph/0306390 [astro-ph].

Baydin, A. G. et al. (2018). 'Automatic Differentiation in Machine Learning: a Survey'. In: *Journal of Machine Learning Research* 18.153, pp. 1–43. URL: http://jmlr.org/papers/v18/17-468.html.

Bayer, J. (2015). 'Learning Sequence Representations'. PhD thesis. Technical University Munich. URL: https://nbn-resolving.org/urn:nbn:de:bvb:91-diss-20151102-1256381-1-9.

Bell, E. F. et al. (2003). 'The Optical and Near-Infrared Properties of Galaxies. I. Luminosity and Stellar Mass Functions'. In: *The Astrophysical Journal Supplement Series* 149.2, pp. 289–312. DOI: 10.1086/378847. arXiv: astro-ph/0302543 [astro-ph].

Bengio, Y., P. Simard, and P. Frasconi (1994). 'Learning long-term dependencies with gradient descent is difficult'. In: *IEEE Transactions on Neural Networks* 5.2, pp. 157–166.

Bergé, J. et al. (2013). 'An Ultra Fast Image Generator (UFig) for wide-field astronomy'. In: *Astronomy and Computing* 1, pp. 23–32. DOI: 10.1016/j.ascom.2013.01.001. URL: https://doi.org/10.1016/j.ascom.2013.01.001.

Berger, P. and G. Stein (Jan. 2019). 'A volumetric deep Convolutional Neural Network for simulation of mock dark matter halo catalogues'. In: *Monthly Notices of the*

Royal Astronomical Society 482.3, pp. 2861–2871. ISSN: 0035-8711. DOI: 10.1093/mnras/sty2949.

Bernardi, M. et al. (2005). 'Colors, Magnitudes, and Velocity Dispersions in Early-Type Galaxies: Implications for Galaxy Ages and Metallicities'. In: *The Astronomical Journal* 129.1, pp. 61–72. DOI: 10.1086/426336. arXiv: astro-ph/0409571 [astro-ph].

Bertin, E. (2009). 'SkyMaker: astronomical image simulations made easy.' In: *Memorie della Societa Astronomica Italiana* 80, p. 422.

Bertin, E. and S. Arnouts (1996). 'SExtractor: Software for source extraction.' In: *Astronomy and Astrophysics Supplement Series* 117, pp. 393–404. DOI: 10.1051/aas:1996164.

Black, Sid et al. (Apr. 2022). 'GPT-NeoX-20B: An Open-Source Autoregressive Language Model'. In: *arXiv e-prints*, arXiv:2204.06745, arXiv:2204.06745. arXiv: 2204.06745 [cs.CL].

Blanton, M. R. et al. (2005). 'The Properties and Luminosity Function of Extremely Low Luminosity Galaxies'. In: *The Astrophysical Journal* 631.1, pp. 208–230. DOI: 10.1086/431416. arXiv: astro-ph/0410164 [astro-ph].

Bo, P. (Aug. 2021). *BlinkDL/RWKV-LM: 0.01*. Version 0.01. DOI: 10.5281/zenodo.5196577. URL: https://doi.org/10.5281/zenodo.5196577.

Bommasani, Rishi et al. (Aug. 2021). 'On the Opportunities and Risks of Foundation Models'. In: *arXiv e-prints*, arXiv:2108.07258, arXiv:2108.07258. arXiv: 2108.07258 [cs.LG].

Bower, R. G. et al. (2006). 'Breaking the hierarchy of galaxy formation'. In: *Monthly Notices of the Royal Astronomical Society* 370.2, pp. 645–655. DOI: 10.1111/j.1365-2966.2006.10519.x. arXiv: astro-ph/0511338 [astro-ph].

Bradley, Larry et al. (2019). *astropy/photutils: v0.6*. DOI: 10.5281/zenodo.2533376. URL: https://doi.org/10.5281/zenodo.2533376.

Branwen, G. (2022). *The Scaling Hypothesis*. URL: https://www.gwern.net/Scaling-hypothesis.

Bretonnière, H. et al. (2021). 'Euclid preparation: XVI. Forecasts for galaxy morphology with the Euclid Survey using Deep Generative Models'. In: *arXiv e-prints*, arXiv:2105.12149, arXiv:2105.12149. arXiv: 2105.12149 [astro-ph.GA].

Brinchmann, J. et al. (2004). 'The physical properties of star-forming galaxies in the low-redshift Universe'. In: *Monthly Notices of the Royal Astronomical Society* 351.4, pp. 1151–1179. DOI: 10.1111/j.1365-2966.2004.07881.x. arXiv: astro-ph/0311060 [astro-ph].

Brock, A., J. Donahue, and K. Simonyan (2018). 'Large Scale GAN Training for High Fidelity Natural Image Synthesis'. In: *CoRR* abs/1809.11096. arXiv: 1809.11096. URL: http://arxiv.org/abs/1809.11096.

Brock, A. et al. (2021). 'High-Performance Large-Scale Image Recognition Without Normalization'. In: *CoRR* abs/2102.06171. arXiv: 2102.06171. URL: https://arxiv.org/abs/2102.06171.

Brodrick, D., D. Taylor, and J. Diederich (2004). 'Recurrent Neural Networks for Narrowband Signal Detection in the Time-Frequency Domain'. In: *Symposium - International Astronomical Union* 213, pp. 483–486. DOI: 10.1017/S0074180900193751.

Brown, T. et al. (2020). 'Language Models are Few-Shot Learners'. In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., pp. 1877–1901. URL: https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Buncher, B., A. N. Sharma, and M. Carrasco Kind (2021). 'Survey2Survey: a deep learning generative model approach for cross-survey image mapping'. In: *Monthly Notices of the Royal Astronomical Society* 503.1, pp. 777–796. DOI: 10.1093/mnras/stab294. arXiv: 2011.07124 [astro-ph.IM].

Bundy, Kevin et al. (2014). 'OVERVIEW OF THE SDSS-IV MaNGA SURVEY: MAPPING NEARBY GALAXIES AT APACHE POINT OBSERVATORY'. In: *The Astrophysical Journal* 798.1, p. 7. DOI: 10.1088/0004-637x/798/1/7. URL: https://doi.org/10.1088/0004-637x/798/1/7.

Camps, P. et al. (2016). 'Far-infrared and dust properties of present-day galaxies in the EAGLE simulations'. In: *Monthly Notices of the Royal Astronomical Society* 462.1, pp. 1057–1075. DOI: 10.1093/mnras/stw1735. arXiv: 1607.04402 [astro-ph.GA].

Capizzi, G., C. Napoli, and L. Paternò (2012). 'An Innovative Hybrid Neuro-wavelet Method for Reconstruction of Missing Data in Astronomical Photometric Surveys'. In: *Artificial Intelligence and Soft Computing*. Berlin, Germany: Springer, pp. 21–29. ISBN: 978-3-642-29347-4. DOI: 10.1007/978-3-642-29347-4_3.

Carballo, R., A. S. Cofiño, and J. I. González-Serrano (Sept. 2004). 'Selection of quasar candidates from combined radio and optical surveys using neural networks'. In: *Monthly Notices of the Royal Astronomical Society* 353.1, pp. 211–220. ISSN: 0035-8711. DOI: 10.1111/j.1365-2966.2004.08056.x.

Carballo, R. et al. (Nov. 2008). 'Use of neural networks for the identification of new z≥ 3.6 QSOs from FIRST–SDSS DR5'. In: *Monthly Notices of the Royal Astronomical Society* 391.1, pp. 369–382. ISSN: 0035-8711. DOI: 10.1111/j.1365-2966.2008.13896.x.

Carrasco-Davis, R. et al. (2019). 'Deep Learning for Image Sequence Classification of Astronomical Events'. In: *Publications of the Astronomical Society of the Pacific* 131.1004, p. 108006. DOI: 10.1088/1538-3873/aaef12. arXiv: 1807.03869 [astro-ph.IM].

Chambers, K. C. et al. (2016). 'The Pan-STARRS1 Surveys'. In: *arXiv e-prints*, arXiv:1612.05560. arXiv: 1612.05560 [astro-ph.IM].

Charnock, T. and A. Moss (2017). 'Deep Recurrent Neural Networks for Supernovae Classification'. In: *The Astrophysical Journal Letters* 837.2, L28, p. L28. DOI: 10.3847/2041-8213/aa603d. arXiv: 1606.07442 [astro-ph.IM].

Chechik, G. et al. (2010). 'Large Scale Online Learning of Image Similarity Through Ranking'. In: *Journal of Machine Learning Research* 11.36, pp. 1109–1135. URL: http://jmlr.org/papers/v11/chechik10a.html.

Chellapilla, K., S. Puri, and P. Simard (2006). 'High Performance Convolutional Neural Networks for Document Processing'. In: *Tenth International Workshop on Frontiers in Handwriting Recognition*. Ed. by Guy Lorette. http://www.suvisoft.com. Université de Rennes 1. La Baule (France): Suvisoft. URL: https://hal.inria.fr/inria-00112631.

Chen, T. et al. (2020a). 'A Simple Framework for Contrastive Learning of Visual Representations'. In: *CoRR* abs/2002.05709. arXiv: 2002.05709. URL: https://arxiv.org/abs/2002.05709.

Chen, T. et al. (2020b). 'Big Self-Supervised Models are Strong Semi-Supervised Learners'. In: *CoRR* abs/2006.10029. arXiv: 2006.10029. URL: https://arxiv.org/abs/2006.10029.

Chen, X. et al. (2016). 'InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets'. In: *Conference on Neural Information Processing Systems 29*. Curran Associates, Inc., pp. 2172–2180.

Chen, X. et al. (2020). 'Improved Baselines with Momentum Contrastive Learning'. In: *CoRR* abs/2003.04297. arXiv: 2003.04297. URL: https://arxiv.org/abs/2003.04297.

Cheng, J., L. Dong, and M. Lapata (2016). 'Long Short-Term Memory-Networks for Machine Reading'. In: *CoRR* abs/1601.06733. arXiv: 1601.06733. URL: http://arxiv.org/abs/1601.06733.

Cho, K. et al. (2014). 'Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation'. In: *ArXiv e-prints*. eprint: 1406.1078. URL: https://arxiv.org/abs/1406.1078.

Choi, Y. et al. (2018). 'StarGAN: Unified Generative Adversarial Networks for Multi-domain Image-to-Image Translation'. In: *Proceedings of CVPR 2018* (Salt Lake City, UT, USA). IEEE Computer Society, p. 8789. eprint: 1711.09020.

Chon, M. C. (Dec. 1998). 'Muon physics and neural network event classifier for the Sudbury Neutrino Observatory'. PhD thesis, p. 2820. URL: https://ui.adsabs.harvard.edu/abs/1998PhDT.......227C/abstract.

Chowdhery, A. et al. (2022). 'PaLM: Scaling Language Modeling with Pathways'. In: *arXiv e-prints*, arXiv:2204.02311, arXiv:2204.02311. arXiv: 2204.02311 [cs.CL].

Cireşan, D. et al. (2010). 'Deep Big Simple Neural Nets Excel on Handwritten Digit Recognition'. In: *arXiv e-prints*, arXiv:1003.0358, arXiv:1003.0358. arXiv: 1003.0358 [cs.NE].

Cireşan, D. et al. (2011). 'Flexible, High Performance Convolutional Neural Networks for Image Classification'. In: *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*. Ed. by T. Walsh. IJCAI/AAAI, pp. 1237–1242. DOI: 10.5591/978-1-57735-516-8/IJCAI11-210. URL: https://doi.org/10.5591/978-1-57735-516-8/IJCAI11-210.

Claeskens, J.-F. et al. (Apr. 2006). 'Identification and redshift determination of quasi-stellar objects with medium-band photometry: application to Gaia'. In: *Monthly Notices of the Royal Astronomical Society* 367.3, pp. 879–904. ISSN: 0035-8711. DOI: 10.1111/j.1365-2966.2006.10024.x.

Clevert, D., T. Unterthiner, and S. Hochreiter (2016). 'Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)'. In: *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings, abs/1511.07289*.

Cohen, J. (1988). *Statistical Power Analysis for the Behavioral Sciences*. Lawrence Erlbaum Associates.

Cole, S. et al. (1998). 'Mock 2dF and SDSS galaxy redshift surveys'. In: *Monthly Notices of the Royal Astronomical Society* 300, pp. 945–966. DOI: 10.1046/j.1365-8711.1998.01936.x. eprint: astro-ph/9801250.

Cole, S. et al. (2000). 'Hierarchical galaxy formation'. In: *Monthly Notices of the Royal Astronomical Society* 319.1, pp. 168–204. DOI: 10.1046/j.1365-8711.2000.03879.x. arXiv: astro-ph/0007281 [astro-ph].

Collister, A. A. and O. Lahav (2004). 'ANNz: Estimating Photometric Redshifts Using Artificial Neural Networks'. In: *The Publications of the Astronomical Society of* *the Pacific* 116.818, pp. 345–351. DOI: 10.1086/383254. arXiv: astro-ph/0311058 [astro-ph].

Courteau, S. (1996). 'Deep r-Band Photometry for Northern Spiral Galaxies'. In: *Astrophysical Journal Supplement Series* 103, p. 363. DOI: 10.1086/192281.

Crawford, Ken (2015). *Bright Spiral Galaxy M81*. [Online; accessed 2020-07-16]. URL: https://apod.nasa.gov/apod/ap151017.html.

Croton, D. J. et al. (2006). 'The many lives of active galactic nuclei: cooling flows, black holes and the luminosities and colours of galaxies'. In: *Monthly Notices of the Royal Astronomical Society* 365.1, pp. 11–28. DOI: 10.1111/j.1365-2966.2005.09675.x. arXiv: astro-ph/0508046 [astro-ph].

Crowson, Katherine et al. (Apr. 2022). 'VQGAN-CLIP: Open Domain Image Generation and Editing with Natural Language Guidance'. In: *arXiv e-prints*, arXiv:2204.08583, arXiv:2204.08583. arXiv: 2204.08583 [cs.CV].

Cybenko, G. (1989). 'Approximation by superpositions of a sigmoidal function'. In: *Mathematics of Control, Signals, and Systems (MCSS)* 2.4, pp. 303–314. ISSN: 0932-4194. DOI: 10.1007/BF02551274. URL: http://dx.doi.org/10.1007/BF02551274.

Devlin, J. et al. (2019). 'BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding'. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://aclanthology.org/N19-1423.

Dewdney, P. E. et al. (2009). 'The Square Kilometre Array'. In: *IEEE Proceedings* 97.8, pp. 1482–1496. DOI: 10.1109/JPROC.2009.2021005.

Dey, A. et al. (2019). 'Overview of the DESI Legacy Imaging Surveys'. In: *The Astronomical Journal* 157.5, 168, p. 168. DOI: 10.3847/1538-3881/ab089d. arXiv: 1804.08657 [astro-ph.IM].

Dhariwal, P. and A. Nichol (2021). 'Diffusion Models Beat GANs on Image Synthesis'. In: *CoRR* abs/2105.05233. arXiv: 2105.05233. URL: https://arxiv.org/abs/2105.05233.

Dieleman, S. (2022). *Diffusion models are autoencoders*. URL: https://benanne.github.io/2022/01/31/diffusion.html.

Dieleman, S., K. W. Willett, and J. Dambre (2015). 'Rotation-invariant convolutional neural networks for galaxy morphology prediction'. In: *Monthly Notices of the Royal Astronomical Society* 450.2, pp. 1441–1459. DOI: 10.1093/mnras/stv632. arXiv: 1503.07077 [astro-ph.IM].

Dobke, B. M. et al. (2010). 'Astronomical Image Simulation for Telescope and Survey Development'. In: *Publications of the Astronomical Society of the Pacific* 122.894, p. 947. DOI: 10.1086/656016. arXiv: 1008.4112 [astro-ph.IM].

Donoso-Oliva, C. et al. (2022). 'ASTROMER: A transformer-based embedding for the representation of light curves'. In: *arXiv e-prints*, arXiv:2205.01677, arXiv:2205.01677. arXiv: 2205.01677 [astro-ph.IM].

Dosovitskiy, A. et al. (2020). 'An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale'. In: *CoRR* abs/2010.11929. arXiv: 2010.11929. URL: https://arxiv.org/abs/2010.11929.

Dowson, D.C and B.V Landau (1982). 'The Fréchet distance between multivariate normal distributions'. In: *Journal of Multivariate Analysis* 12.3, pp. 450–455. ISSN: 0047-259X. DOI: https://doi.org/10.1016/0047-259X(82)90077-X. URL: https://www.sciencedirect.com/science/article/pii/0047259X8290077X.

Dubois, Y. et al. (2014). 'Dancing in the dark: galactic properties trace spin swings along the cosmic web'. In: *Monthly Notices of the Royal Astronomical Society* 444, pp. 1453–1468. DOI: 10.1093/mnras/stu1227. arXiv: 1402.1165.

Eisenstein, D. J. et al. (2011). 'SDSS-III: Massive Spectroscopic Surveys of the Distant Universe, the Milky Way, and Extra-Solar Planetary Systems'. In: *The Astronomical Journal* 142.3, 72, p. 72. DOI: 10.1088/0004-6256/142/3/72. arXiv: 1101.1529 [astro-ph.IM].

Elman, J. L. (1990). 'Finding structure in time'. In: *Cognitive Science* 14.2, pp. 179–211. ISSN: 0364-0213. DOI: https://doi.org/10.1016/0364-0213(90)90002-E. URL: https://www.sciencedirect.com/science/article/pii/036402139090002E.

Eneev, T. M., N. N. Kozlov, and R. A. Sunyaev (1973). 'Tidal Interaction of Galaxies'. In: *Astronomy and Astrophysics* 22, p. 41.

Fernández Lorenzo, M. et al. (2013). 'The stellar mass-size relation for the most isolated galaxies in the local Universe'. In: *Monthly Notices of the Royal Astronomical Society* 434.1, pp. 325–335. DOI: 10.1093/mnras/stt1020. arXiv: 1306.1687 [astro-ph.CO].

Finke, T., M. Krämer, and S. Manconi (Nov. 2021). 'Classification of Fermi-LAT sources with deep learning using energy and time spectra'. In: *Monthly Notices of the Royal Astronomical Society* 507.3, pp. 4061–4073. ISSN: 0035-8711. DOI: 10.1093/mnras/stab2389.

Firth, A. E., O. Lahav, and R. S. Somerville (Mar. 2003). 'Estimating photometric redshifts with artificial neural networks'. In: *Monthly Notices of the Royal Astro-*

nomical Society 339.4, pp. 1195–1202. ISSN: 0035-8711. DOI: 10.1046/j.1365-8711.2003.06271.x.

Frankle, J. and M. Carbin (2018). 'The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks'. In: *arXiv e-prints*, arXiv:1803.03635, arXiv:1803.03635. arXiv: 1803.03635 [cs.LG].

Friel, R. (2022). *Chinchilla's Wild Implications*. URL: https://www.alignmentforum.org/posts/6Fpvch8RR29qLEWNH/chinchilla-s-wild-implications.

Fukushima, K. (1980). 'Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position'. In: *Biological Cybernetics* 36.4, pp. 193–202. DOI: 10.1007/bf00344251. URL: https://doi.org/10.1007/bf00344251.

Fussell, Levi and Ben Moews (2019). 'Forging new worlds: high-resolution synthetic galaxies with chained generative adversarial networks'. In: *Monthly Notices of the Royal Astronomical Society* 485.3, pp. 3203–3214. DOI: 10.1093/mnras/stz602. arXiv: 1811.03081 [astro-ph.IM].

Gal, R. et al. (2022). 'An Image is Worth One Word: Personalizing Text-to-Image Generation using Textual Inversion'. In: *arXiv e-prints*, arXiv:2208.01618, arXiv:2208.01618. arXiv: 2208.01618 [cs.CV].

Gao, L. et al. (2020). 'The Pile: An 800GB Dataset of Diverse Text for Language Modeling'. In: *arXiv e-prints*, arXiv:2101.00027, arXiv:2101.00027. arXiv: 2101.00027 [cs.CL].

Gers, F., J. Schmidhuber, and F. Cummins (2000). 'Learning to Forget: Continual Prediction with LSTM'. In: *Neural Computation* 12.10, pp. 2451–2471. DOI: 10.1162/089976600300015015.

Gilhuly, C. and S. Courteau (2018). 'An extensive photometric catalogue of CALIFA galaxies'. In: *Monthly Notices of the Royal Astronomical Society* 477.1, pp. 845–862. DOI: 10.1093/mnras/sty756. arXiv: 1803.08933 [astro-ph.GA].

Glorot, X., A. Bordes, and Y. Bengio (2011). 'Deep Sparse Rectifier Neural Networks'. In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. Ed. by G. Gordon, D. Dunson, and M. Dudík. Vol. 15. Proceedings of Machine Learning Research. Fort Lauderdale, FL, USA: Proceedings of Machine Learning Research, pp. 315–323. URL: http://proceedings.mlr.press/v15/glorot11a.html.

Gonzalez, C. A. G., O. Absil, and M. Van Droogenbroeck (May 2018). 'Supervised detection of exoplanets in high-contrast imaging sequences'. In: *Astronomy & Astrophysics* 613, A71. ISSN: 0004-6361. DOI: 10.1051/0004-6361/201731961.

Goodfellow, I. et al. (2014). 'Generative Adversarial Nets'. In: *Advances in Neural Information Processing Systems 27*. Ed. by Z. Ghahramani et al. Curran Associates, Inc., pp. 2672–2680. URL: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf.

Grill, J. et al. (2020). 'Bootstrap Your Own Latent: A New Approach to Self-Supervised Learning'. In: *CoRR* abs/2006.07733. arXiv: 2006.07733. URL: https://arxiv.org/abs/2006.07733.

Hadsell, R., S. Chopra, and Y. LeCun (2006). 'Dimensionality Reduction by Learning an Invariant Mapping'. In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. Vol. 2, pp. 1735–1742. DOI: 10.1109/CVPR.2006.100.

Hála, P. (2014). 'Spectral classification using convolutional neural networks'. In: *arXiv e-prints*, arXiv:1412.8341, arXiv:1412.8341. arXiv: 1412.8341 [cs.CV].

Hall, M. et al. (2012). 'An investigation of Sloan Digital Sky Survey imaging data and multiband scaling relations of spiral galaxies'. In: *Monthly Notices of the Royal Astronomical Society* 425.4, pp. 2741–2765. DOI: 10.1111/j.1365-2966.2012.21290.x. arXiv: 1111.5009 [astro-ph.CO].

Hayat, M. A. et al. (2021). 'Self-supervised Representation Learning for Astronomical Images'. In: *The Astrophysical Journal Letters* 911.2, p. L33. DOI: 10.3847/2041-8213/abf2c7. URL: https://doi.org/10.3847/2041-8213/abf2c7.

He, K. et al. (2015). 'Deep Residual Learning for Image Recognition'. In: *arXiv e-prints*, arXiv:1512.03385, arXiv:1512.03385. arXiv: 1512.03385 [cs.CV].

He, K. et al. (2019). 'Momentum Contrast for Unsupervised Visual Representation Learning'. In: *CoRR* abs/1911.05722. arXiv: 1911.05722. URL: http://arxiv.org/abs/1911.05722.

Hebb, D. O. (1949). *The organization of behavior: A neuropsychological theory*. Wiley. ISBN: 0-8058-4300-0.

Heusel, M. et al. (2017). 'GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium'. In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc.

Hippel, T. von et al. (July 1994). 'Automated classification of stellar spectra – I. Initial results with artificial neural networks'. In: *Monthly Notices of the Royal Astronomical Society* 269.1, pp. 97–104. ISSN: 0035-8711. DOI: 10.1093/mnras/269.1.97.

Ho, J., A. Jain, and P. Abbeel (2020). 'Denoising Diffusion Probabilistic Models'. In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., pp. 6840–6851.

Ho, J. et al. (2021). 'Cascaded Diffusion Models for High Fidelity Image Generation'. In: *CoRR* abs/2106.15282. arXiv: 2106.15282. URL: https://arxiv.org/abs/2106.15282.

Hochreiter, S. (1991). *Untersuchungen zu dynamischen neuronalen Netzen (investigations on dynamic neural networks)*. Diploma thesis. In German. URL: http://www.bioinf.jku.at/publications/older/3804.pdf.

Hochreiter, S. and J. Schmidhuber (1997). 'Long Short-Term Memory'. In: *Neural Computation* 9.8, pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735. URL: https://doi.org/10.1162/neco.1997.9.8.1735.

Hoffmann, J. et al. (2022). 'Training Compute-Optimal Large Language Models'. In: *arXiv e-prints*, arXiv:2203.15556, arXiv:2203.15556. arXiv: 2203.15556 [cs.CL].

Holzschuh, B. J. et al. (Sept. 2022). 'Realistic galaxy images and improved robustness in machine learning tasks from generative modelling'. In: *Monthly Notices of the Royal Astronomical Society* 515.1, pp. 652–677. ISSN: 0035-8711. DOI: 10.1093/mnras/stac1188.

Hornik, K., M. Tinchcombe, and H. White (1991). 'Approximation capabilities of multilayer feedforward networks'. In: *Neural Networks* 4.2, pp. 251–257. DOI: 10.1016/0893-6080(91)90009-T. URL: http://www.sciencedirect.com/science/article/pii/089360809190009T.

Hossain, Z. et al. (2018). 'A Comprehensive Survey of Deep Learning for Image Captioning'. In: *arXiv e-prints*, arXiv:1810.04020, arXiv:1810.04020. arXiv: 1810.04020 [cs.CV].

Huertas-Company, M. et al. (Oct. 2015). 'A CATALOG OF VISUAL-LIKE MORPHOLOGIES IN THE 5 CANDELS FIELDS USING DEEP LEARNING'. In: *Astrophysical Journal Supplement Series* 221.1, p. 8. ISSN: 1538-4365. DOI: 10.1088/0067-0049/221/1/8.

Hyvärinen, A. (2005). 'Estimation of Non-Normalized Statistical Models by Score Matching'. In: *Journal of Machine Learning Research* 6.24, pp. 695–709. URL: http://jmlr.org/papers/v6/hyvarinen05a.html.

Illingworth, G. D. et al. (2013). 'The HST eXtreme Deep Field (XDF): Combining All ACS and WFC3/IR Data on the HUDF Region into the Deepest Field Ever'. In: *The Astrophysical Journal Supplement Series* 209, 6, p. 6. DOI: 10.1088/0067-0049/209/1/6. arXiv: 1305.1931.

Isola, P. et al. (2016). 'Image-to-Image Translation with Conditional Adversarial Networks'. In: *arXiv e-prints*, arXiv:1611.07004, arXiv:1611.07004. arXiv: 1611.07004 [cs.CV].

Ivakhnenko, A. (1971). 'Polynomial Theory of Complex Systems'. In: *IEEE Transactions on Systems, Man, and Cybernetics* SMC-1.4, pp. 364–378. DOI: 10.1109/TSMC.1971.4308320.

Ivakhnenko, A. and V. Lapa (1965). *Cybernetic Predicting Devices*. English translation available as of 2022-06-08 at https://www.gwern.net/docs/ai/1966-ivakhnenko.pdf. CCM Information Corporation.

Ivezić, Ž. et al. (2019). 'LSST: From Science Drivers to Reference Design and Anticipated Data Products'. In: *The Astrophysical Journal* 873, 111, p. 111. DOI: 10.3847/1538-4357/ab042c. arXiv: 0805.2366.

Jaeger, H. and H. Haas (Apr. 2004). 'Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication'. In: *Science* 304.5667, pp. 78–80. ISSN: 0036-8075. DOI: 10.1126/science.1091277.

Jayaram, V. and J. Thickstun (2020). 'Source Separation with Deep Generative Priors'. In: *CoRR* abs/2002.07942. arXiv: 2002.07942. URL: https://arxiv.org/abs/2002.07942.

Jedrzejewski, Robert I. (1987). 'CCD surface photometry of elliptical galaxies - I. Observations, reduction and results.' In: *Monthly Notices of the Royal Astronomical Society* 226, pp. 747–768. DOI: 10.1093/mnras/226.4.747.

Jetchev, N., U. Bergmann, and R. Vollgraf (2016). 'Texture Synthesis with Spatial Generative Adversarial Networks'. In: *CoRR* abs/1611.08207. arXiv: 1611.08207. URL: http://arxiv.org/abs/1611.08207.

Jia, X. et al. (2015). 'Guiding Long-Short Term Memory for Image Caption Generation'. In: *arXiv e-prints*, arXiv:1509.04942, arXiv:1509.04942. arXiv: 1509.04942 [cs.CV].

Jolicoeur-Martineau, A. (2018). 'The relativistic discriminator: a key element missing from standard GAN'. In: *CoRR* abs/1807.00734. arXiv: 1807.00734. URL: http://arxiv.org/abs/1807.00734.

Jolicoeur-Martineau, A. et al. (2020). 'Adversarial score matching and improved sampling for image generation'. In: *CoRR* abs/2009.05475. arXiv: 2009.05475. URL: https://arxiv.org/abs/2009.05475.

Jolicoeur-Martineau, A. et al. (2021). 'Gotta Go Fast When Generating Data with Score-Based Models'. In: *CoRR* abs/2105.14080. arXiv: 2105.14080. URL: https://arxiv.org/abs/2105.14080.

Jonsson, P. (2006). 'SUNRISE: polychromatic dust radiative transfer in arbitrary geometries'. In: *Monthly Notices of the Royal Astronomical Society* 372, pp. 2–20. DOI: 10.1111/j.1365-2966.2006.10884.x. eprint: astro-ph/0604118.

Jumper, J. et al. (2021). 'Highly accurate protein structure prediction with AlphaFold'. In: *Nature* 596.7873, pp. 583–589. ISSN: 0028-0836.

Kadkhodaie, Z. and E. P. Simoncelli (2020). 'Solving Linear Inverse Problems Using the Prior Implicit in a Denoiser'. In: *CoRR* abs/2007.13640. arXiv: 2007.13640. URL: https://arxiv.org/abs/2007.13640.

Kaiser, L. et al. (2017). 'One Model To Learn Them All'. In: *arXiv e-prints*, arXiv:1706.05137, arXiv:1706.05137. arXiv: 1706.05137 [cs.LG].

Kaplan, Jared et al. (2020). 'Scaling Laws for Neural Language Models'. In: *CoRR* abs/2001.08361. arXiv: 2001.08361. URL: https://arxiv.org/abs/2001.08361.

Karnewar, A., O. Wang, and R. S. Iyengar (2019). 'MSG-GAN: Multi-Scale Gradient GAN for Stable Image Synthesis'. In: *CoRR* abs/1903.06048. arXiv: 1903.06048. URL: http://arxiv.org/abs/1903.06048.

Karpenka, N. V., F. Feroz, and M. P. Hobson (2013). 'A simple and robust method for automated photometric classification of supernovae using neural networks'. In: "*Monthly Notices of the Royal Astronomical Society*" 429.2, pp. 1278–1285. DOI: 10.1093/mnras/sts412. arXiv: 1208.1264 [astro-ph.CO].

Karras, T., S. Laine, and T. Aila (2018). 'A Style-Based Generator Architecture for Generative Adversarial Networks'. In: *CoRR* abs/1812.04948. arXiv: 1812.04948. URL: http://arxiv.org/abs/1812.04948.

Karras, T. et al. (2017). 'Progressive Growing of GANs for Improved Quality, Stability, and Variation'. In: *CoRR* abs/1710.10196. arXiv: 1710.10196. URL: http://arxiv.org/abs/1710.10196.

Kaviraj, S. et al. (2017). 'The Horizon-AGN simulation: evolution of galaxy properties over cosmic time'. In: *Monthly Notices of the Royal Astronomical Society* 467, pp. 4739–4752. DOI: 10.1093/mnras/stx126. arXiv: 1605.09379.

Kessler, R. et al. (Nov. 2010). 'Results from the Supernova Photometric Classification Challenge'. In: *Publications of the Astronomical Society of the Pacific* 122.898, pp. 1415–1431. ISSN: 0004-6280. DOI: 10.1086/657607.

Khandai, Nishikanta et al. (2015). 'The MassiveBlack-II simulation: the evolution of haloes and galaxies to z $\sim$ 0'. In: *Monthly Notices of the Royal Astronomical Society* 450.2, pp. 1349–1374. DOI: 10.1093/mnras/stv627. arXiv: 1402.0888 [astro-ph.CO].

Kim, J. et al. (July 2022). 'Pure Transformers are Powerful Graph Learners'. In: *arXiv e-prints*, arXiv:2207.02505, arXiv:2207.02505. arXiv: 2207.02505 [cs.LG].

Kingma, D. P. and J. Ba (2015). 'Adam: A Method for Stochastic Optimization'. In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA,* *USA, May 7-9, 2015, Conference Track Proceedings*. URL: http://arxiv.org/abs/1412.6980.

Kingma, D. P. and M. Welling (2013). 'Auto-Encoding Variational Bayes'. In: *arXiv e-prints*, arXiv:1312.6114, arXiv:1312.6114. arXiv: 1312.6114 [stat.ML].

Kocifaj, M. et al. (2021). 'The proliferation of space objects is a rapidly increasing source of artificial night sky brightness'. In: *Monthly Notices of the Royal Astronomical Society* 504.1, pp. L40–L44. DOI: 10.1093/mnrasl/slab030. arXiv: 2103.17125 [astro-ph.IM].

Kodalg, N. et al. (2017). 'How to Train Your DRAGAN'. In: *CoRR* abs/1705.07215. arXiv: 1705.07215. URL: http://arxiv.org/abs/1705.07215.

Koons, H. C. and D. J. Gorney (May 1990). 'A sunspot maximum prediction using a neural network'. In: *Eos, Transactions American Geophysical Union* 71.18, pp. 677–688. ISSN: 0096-3941. DOI: 10.1029/EO071i018p00677-01.

Krizhevsky, A., I. Sutskever, and G. E. Hinton (2012). 'ImageNet Classification with Deep Convolutional Neural Networks'. In: *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*. Ed. by P. L. Bartlett et al., pp. 1106–1114. URL: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.

Kügler, S. D., N. Gianniotis, and K. L. Polsterer (Feb. 2016). 'An explorative approach for inspecting Kepler data'. In: *Monthly Notices of the Royal Astronomical Society* 455.4, pp. 4399–4405. ISSN: 0035-8711. DOI: 10.1093/mnras/stv2604.

Lacoste, A. et al. (2019). 'Quantifying the Carbon Emissions of Machine Learning'. In: *arXiv e-prints*, arXiv:1910.09700, arXiv:1910.09700. arXiv: 1910.09700 [cs.CY].

Lagos, Claudia del P. et al. (2019). 'From the far-ultraviolet to the far-infrared - galaxy emission at $0 \leq z \leq 10$ in the SHARK semi-analytic model'. In: *Monthly Notices of the Royal Astronomical Society* 489.3, pp. 4196–4216. DOI: 10.1093/mnras/stz2427. arXiv: 1908.03423 [astro-ph.GA].

Lahav, O. et al. (1995). 'Galaxies, Human Eyes, and Artificial Neural Networks'. In: *Science* 267.5199, pp. 859–862. DOI: 10.1126/science.267.5199.859. arXiv: astro-ph/9412027 [astro-ph].

Lanusse, François et al. (2021). 'Deep generative models for galaxy image simulations'. In: *Monthly Notices of the Royal Astronomical Society* 504.4, pp. 5543–5555. DOI: 10.1093/mnras/stab1214. arXiv: 2008.03833 [astro-ph.IM].

Lauritsen, L. et al. (Oct. 2021). 'Superresolving Herschel imaging: a proof of concept using Deep Neural Networks'. In: *Monthly Notices of the Royal Astronomical Society* 507.1, pp. 1546–1556. ISSN: 0035-8711. DOI: 10.1093/mnras/stab2195.

LeCun, Y. et al. (1989). 'Backpropagation Applied to Handwritten Zip Code Recognition'. In: *Neural Computation* 1.4, pp. 541–551. DOI: 10.1162/neco.1989.1.4.541. eprint: https://doi.org/10.1162/neco.1989.1.4.541. URL: https://doi.org/10.1162/neco.1989.1.4.541.

Lin, M., Q. Chen, and S. Yan (2013). 'Network In Network'. In: *arXiv e-prints*, arXiv:1312.4400, arXiv:1312.4400. arXiv: 1312.4400 [cs.NE].

Lin, Z. et al. (2018). 'PacGAN: The power of two samples in generative adversarial networks'. In: *Conference on Neural Information Processing Systems 31*. Curran Associates, Inc., pp. 1498–1507.

Linnainmaa, S. (1970). 'The representation of the cumulative rounding error of an algorithm as a Taylor expansion of the local rounding errors'. In Finnish. MA thesis. The University of Helsinki.

— (1976). 'Taylor expansion of the accumulated rounding error'. In: *BIT* 16.2, pp. 146–160. DOI: 10.1007/bf01931367. URL: https://doi.org/10.1007/bf01931367.

Liu, H. et al. (2019). 'Predicting Solar Flares Using a Long Short-term Memory Network'. In: *The Astrophysical Journal* 877.2, 121, p. 121. DOI: 10.3847/1538-4357/ab1b3c. arXiv: 1905.07095 [astro-ph.SR].

Liu, H. et al. (2021). 'Pay Attention to MLPs'. In: *arXiv e-prints*, arXiv:2105.08050, arXiv:2105.08050. arXiv: 2105.08050 [cs.LG].

Liu, J. et al. (Mar. 2018). 'MassiveNuS: cosmological massive neutrino simulations'. In: *Journal of Cosmology and Astroparticle Physics* 2018.03, p. 049. ISSN: 1475-7516. DOI: 10.1088/1475-7516/2018/03/049.

Lloyd, S. (1982). 'Least squares quantization in PCM'. In: *IEEE Transactions on Information Theory* 28.2, pp. 129–137. DOI: 10.1109/TIT.1982.1056489.

Lloyd-Hart, M. et al. (1992). 'First Results of an On-Line Adaptive Optics System with Atmospheric Wavefront Sensing by an Artificial Neural Network'. In: *The Astrophysical Journal Letters* 390, p. L41. DOI: 10.1086/186367.

Lovell, Christopher C. et al. (2021). 'Reproducing submillimetre galaxy number counts with cosmological hydrodynamic simulations'. In: *Monthly Notices of the Royal Astronomical Society* 502.1, pp. 772–793. DOI: 10.1093/mnras/staa4043. arXiv: 2006.15156 [astro-ph.GA].

Lu, Z. et al. (2017). 'The Expressive Power of Neural Networks: A View from the Width'. In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al.

Vol. 30. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper/2017/file/32cbf687880eb1674a07bf717761dd3a-Paper.pdf.

Luhman, E. and T. Luhman (2021). 'Knowledge Distillation in Iterative Generative Models for Improved Sampling Speed'. In: *arXiv e-prints*, arXiv:2101.02388, arXiv:2101.02388. arXiv: 2101.02388 [cs.LG].

Lundstedt, H., H. Gleisner, and P. Wintoft (Dec. 2002). 'Operational forecasts of the geomagnetic Dst index'. In: *Geophysical Research Letters* 29.24, pp. 34-1–34-4. ISSN: 0094-8276. DOI: 10.1029/2002GL016151.

Maas, A. L., A. Y. Hannun, and A. Y. Ng (2013). 'Rectifier nonlinearities improve neural network acoustic models'. In: *Proc. International Conference on Machine Learning*. Vol. 30, p. 3.

Mandelbaum, R. et al. (2012). 'Precision simulation of ground-based lensing data using observations from space'. In: *Monthly Notices of the Royal Astronomical Society* 420, pp. 1518–1540. DOI: 10.1111/j.1365-2966.2011.20138.x. arXiv: 1107.4629.

McCulloch, W. and W. Pitts (1943). 'A Logical Calculus of Ideas Immanent in Nervous Activity'. In: *Bulletin of Mathematical Biophysics* 5, pp. 127–147.

Melas-Kyriazi, L. (2021). 'Do You Even Need Attention? A Stack of Feed-Forward Layers Does Surprisingly Well on ImageNet'. In: *arXiv e-prints*, arXiv:2105.02723, arXiv:2105.02723. arXiv: 2105.02723 [cs.CV].

Metz, C. (2021). *Genius Makers: The Mavericks Who Brought AI to Google, Facebook, and the World*. Penguin Random House. ISBN: 9781847942142.

Miller, A. S. (Jan. 1993). 'A review of neural network applications in Astronomy'. In: *Vistas in Astronomy* 36, pp. 141–161. ISSN: 0083-6656. DOI: 10.1016/0083-6656(93)90118-4.

Miller, M. J. Smith et al. (in prep.). 'Extracting representations of stochastically sampled VVV time series data'. In: *N/A*.

Milletari, F., N. Navab, and S. Ahmadi (2016). 'V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation'. In: *CoRR* abs/1606.04797. arXiv: 1606.04797. URL: http://arxiv.org/abs/1606.04797.

Minsky, M. and S. Papert (1969). *Perceptrons: An Introduction to Computational Geometry*. Cambridge, MA, USA: MIT Press.

Mirza, M. and S. Osindero (2014). 'Conditional Generative Adversarial Nets'. In: *CoRR* abs/1411.1784. arXiv: 1411.1784. URL: http://arxiv.org/abs/1411.1784.

Misra, D. (2019). 'Mish: A Self Regularized Non-Monotonic Neural Activation Function'. In: *CoRR* abs/1908.08681. arXiv: 1908.08681. URL: http://arxiv.org/abs/1908.08681.

Miyato, T. et al. (2018). 'Spectral Normalization for Generative Adversarial Networks'. In: *CoRR* abs/1802.05957. arXiv: 1802.05957. URL: http://arxiv.org/abs/1802.05957.

Mordido, G., H. Yang, and C. Meinel (2018). 'Dropout-GAN: Learning from a Dynamic Ensemble of Discriminators'. In: *CoRR* abs/1807.11346. arXiv: 1807.11346. URL: http://arxiv.org/abs/1807.11346.

Morningstar, W. R. et al. (Sept. 2019). 'Data-driven Reconstruction of Gravitationally Lensed Galaxies Using Recurrent Inference Machines'. In: *Astrophysical Journal* 883.1, p. 14. ISSN: 1538-4357. DOI: 10.3847/1538-4357/ab35d7.

Morvan, M. et al. (2022). 'Don't Pay Attention to the Noise: Learning Self-supervised Representations of Light Curves with a Denoising Time Series Transformer'. In: *arXiv e-prints*, arXiv:2207.02777, arXiv:2207.02777. arXiv: 2207.02777 [astro-ph.IM].

Mustafa, M. et al. (2019a). 'CosmoGAN: creating high-fidelity weak lensing convergence maps using Generative Adversarial Networks'. In: *Computational Astrophysics and Cosmology* 6.1, 1, p. 1. DOI: 10.1186/s40668-019-0029-9. arXiv: 1706.02390 [astro-ph.IM].

Mustafa, Mustafa et al. (2019b). 'CosmoGAN: creating high-fidelity weak lensing convergence maps using Generative Adversarial Networks'. In: *Computational Astrophysics and Cosmology* 6.1, 1, p. 1. DOI: 10.1186/s40668-019-0029-9. arXiv: 1706.02390 [astro-ph.IM].

Naim, A. et al. (June 1995a). 'A comparative study of morphological classifications of APM galaxies'. In: *Monthly Notices of the Royal Astronomical Society* 274.4, pp. 1107–1125. ISSN: 0035-8711. DOI: 10.1093/mnras/274.4.1107.

Naim, A. et al. (Aug. 1995b). 'Automated morphological classification of APM galaxies by supervised artificial neural networks'. In: *Monthly Notices of the Royal Astronomical Society* 275.3, pp. 567–590. ISSN: 0035-8711. DOI: 10.1093/mnras/275.3.567.

Nair, V. and G. E. Hinton (2010). 'Rectified Linear Units Improve Restricted Boltzmann Machines'. In: *Proceedings of the 27th International Conference on International Conference on Machine Learning*. ICML'10. Haifa, Israel: Omnipress, pp. 807–814. ISBN: 9781605589077.

Nanni, L., S. Ghidoni, and S. Brahnam (2018). 'Ensemble of convolutional neural networks for bioimage classification'. In: *Applied Computing and Informatics*. DOI:

10.1016/j.aci.2018.06.002. URL: https://doi.org/10.1016%2Fj.aci.2018.06.002.

Naul, B. et al. (2018). 'A recurrent neural network for classification of unevenly sampled variable stars'. In: *Nature Astronomy* 2.2, pp. 151–155. ISSN: 2397-3366. DOI: 10.1038/s41550-017-0321-z.

Nichol, A. and P. Dhariwal (2021). 'Improved Denoising Diffusion Probabilistic Models'. In: *CoRR* abs/2102.09672. arXiv: 2102.09672. URL: https://arxiv.org/abs/2102.09672.

Nørgaard-Nielsen, H. U. and H. E. Jørgensen (2008). 'Foreground removal from CMB temperature maps using an MLP neural network'. In: *Astrophysics and Space Science* 318.3-4, pp. 195–206. DOI: 10.1007/s10509-008-9912-6. arXiv: 0809.2914 [astro-ph].

Obreschkow, D. et al. (2009). 'A Virtual Sky with Extragalactic H I and CO Lines for the Square Kilometre Array and the Atacama Large Millimeter/Submillimeter Array'. In: *The Astrophysical Journal* 703, pp. 1890–1903. DOI: 10.1088/0004-637X/703/2/1890. arXiv: 0908.0983.

Odewahn, S. C. et al. (1992). 'Automated Star/Galaxy Discrimination With Neural Networks'. In: *The Astronomical Journal* 103, p. 318. DOI: 10.1086/116063.

Odewahn, S. C. et al. (1993). 'Star-Galaxy Separation with a Neural Network. II. Multiple Schmidt Plate Fields'. In: *Publications of the Astronomical Society of the Pacific* 105, p. 1354. DOI: 10.1086/133317.

Oh, K. and K. Jung (2004). 'GPU implementation of neural networks'. In: *Pattern Recognition* 37.6, pp. 1311–1314. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2004.01.013. URL: https://www.sciencedirect.com/science/article/pii/S0031320304000524.

Pan, J., Y. Ting, and J. Yu (2022). 'Astroconformer: Inferring Surface Gravity of Stars from Stellar Light Curves with Transformer'. In: *arXiv e-prints*, arXiv:2207.02787, arXiv:2207.02787. arXiv: 2207.02787 [astro-ph.SR].

Park, J. S. et al. (Aug. 2022). 'Social Simulacra: Creating Populated Prototypes for Social Computing Systems'. In: *arXiv e-prints*, arXiv:2208.04024, arXiv:2208.04024. arXiv: 2208.04024 [cs.HC].

Parmar, N. et al. (2018). 'Image Transformer'. In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, pp. 4055–4064. URL: https://proceedings.mlr.press/v80/parmar18a.html.

Paszke, A. et al. (2019). 'PyTorch: An Imperative Style, High-Performance Deep Learning Library'. In: *Advances in Neural Information Processing Systems 32*. Ed.

by H. Wallach et al. Curran Associates, Inc., pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Paul, R. et al. (2018). 'Predicting Nodule Malignancy using a CNN Ensemble Approach'. In: *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE. DOI: 10.1109/ijcnn.2018.8489345.

Peng, Chien Y. et al. (2002). 'Detailed Structural Decomposition of Galaxy Images'. In: *The Astronomical Journal* 124.1, pp. 266–293. DOI: 10.1086/340952. arXiv: astro-ph/0204182 [astro-ph].

Pennington, R. L. et al. (May 1993). 'The Automated Plate Scanner Catalog of the Palomar Sky Survey. I. Scanning Parameters and Procedures on JSTOR'. In: *Publications of the Astronomical Society of the Pacific* 105.687, pp. 521–526. URL: https://www.jstor.org/stable/40680062.

Peterson, J. R. et al. (2015). 'Simulation of Astronomical Images from Optical Survey Telescopes using a Comprehensive Photon Monte Carlo Approach'. In: *The Astrophysical Journal Supplement Series* 218.1, p. 14. DOI: 10.1088/0067-0049/218/1/14. URL: https://doi.org/10.1088/0067-0049/218/1/14.

Preechakul, Konpat et al. (2021). 'Diffusion Autoencoders: Toward a Meaningful and Decodable Representation'. In: *CoRR* abs/2111.15640. arXiv: 2111.15640. URL: https://arxiv.org/abs/2111.15640.

Raddick, M. Jordan et al. (2010). 'Galaxy Zoo: Exploring the Motivations of Citizen Science Volunteers'. In: *Astronomy Education Review* 9.1, p. 010103. DOI: 10.3847/AER2009036. arXiv: 0909.2925 [astro-ph.IM].

Radford, A., L. Metz, and S. Chintala (2016). 'Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks'. In: *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*. URL: http://arxiv.org/abs/1511.06434.

Radford, A. et al. (2019). 'Language Models are Unsupervised Multitask Learners'. In: *OpenAI Whitepaper*. URL: https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

Rae, Jack W. et al. (2021). 'Scaling Language Models: Methods, Analysis & Insights from Training Gopher'. In: *CoRR* abs/2112.11446. arXiv: 2112.11446. URL: https://arxiv.org/abs/2112.11446.

Raina, R., A. Madhavan, and A. Y. Ng (2009). 'Large-Scale Deep Unsupervised Learning Using Graphics Processors'. In: *Proceedings of the 26th Annual International*

Conference on Machine Learning. ICML ’09. Montreal, Quebec, Canada: Association for Computing Machinery, pp. 873–880. ISBN: 9781605585161. DOI: 10.1145/1553374.1553486. URL: https://doi.org/10.1145/1553374.1553486.

Ramachandran, P., B. Zoph, and Q. V. Le (2017). ‘Searching for Activation Functions’. In: *arXiv e-prints*, arXiv:1710.05941, arXiv:1710.05941. arXiv: 1710.05941 [cs.NE].

Ramesh, R. et al. (2022). ‘Hierarchical Text-Conditional Image Generation with CLIP Latents’. In: URL: https://cdn.openai.com/papers/dall-e-2.pdf.

Rappaport, B. and K. Anderson (Jan. 1988). ‘Automated galaxy recognition.’ In: *European Southern Observatory Conference and Workshop Proceedings*. Vol. 28. European Southern Observatory Conference and Workshop Proceedings, pp. 233–238.

Ravanbakhsh, S. et al. (2016). ‘Enabling Dark Energy Science with Deep Generative Models of Galaxy Images’. In: *arXiv e-prints*, arXiv:1609.05796, arXiv:1609.05796. arXiv: 1609.05796 [astro-ph.IM].

Raymond, E. S. (1999). *The Cathedral and the Bazaar*. 1st. USA: O’Reilly & Associates, Inc. ISBN: 1565927249.

Reed, S. et al. (2022). ‘A Generalist Agent’. In: *arXiv e-prints*, arXiv:2205.06175, arXiv:2205.06175. arXiv: 2205.06175 [cs.AI].

Regier, Jeffrey, Jon McAuliffe, and Prabhat (2015). ‘A deep generative model for astronomical images of galaxies’. In: *NIPS Workshop on Advances in Approximate Bayesian Inference*.

Regier, Jeffrey et al. (2015). ‘Celeste: Variational inference for a generative model of astronomical images’. In: *International Conference on Machine Learning (ICML)*.

Reiman, D. M. and B. E. Göhre (2019). ‘Deblending galaxy superpositions with branched generative adversarial networks’. In: "*Monthly Notices of the Royal Astronomical Society*" 485.2, pp. 2617–2627. DOI: 10.1093/mnras/stz575. arXiv: 1810.10098 [astro-ph.IM].

Remy, B. et al. (2020). ‘Probabilistic Mapping of Dark Matter by Neural Score Matching’. In: *arXiv e-prints*, arXiv:2011.08271, arXiv:2011.08271. arXiv: 2011.08271 [astro-ph.CO].

Remy, B. et al. (Jan. 2022). ‘Probabilistic Mass Mapping with Neural Score Estimation’. In: *ArXiv e-prints*. DOI: 10.48550/arXiv.2201.05561. eprint: 2201.05561.

Rezende, D. J. and S. Mohamed (2015). ‘Variational Inference with Normalizing Flows’. In: *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*. ICML’15. Lille, France: JMLR.org, pp. 1530–1538.

Ricker, G. R. et al. (Jan. 2015). 'Transiting Exoplanet Survey Satellite (TESS)'. In: *Journal of Astronomical Telescopes, Instruments, and Systems* 1, 014003, p. 014003. DOI: 10.1117/1.JATIS.1.1.014003.

Robbins, H. and S. Monro (1951). 'A Stochastic Approximation Method'. In: *Annals of Mathematical Statistics* 22.3, pp. 400–407. ISSN: 0003-4851. DOI: 10.1214/aoms/1177729586.

Rodriguez, A. C. et al. (2018). 'Fast cosmic web simulations with generative adversarial networks'. In: *Computational Astrophysics and Cosmology* 5.1, 4, p. 4. DOI: 10.1186/s40668-018-0026-4. arXiv: 1801.09070 [astro-ph.CO].

Rombach, R. et al. (2021). 'High-Resolution Image Synthesis with Latent Diffusion Models'. In: *arXiv e-prints*, arXiv:2112.10752, arXiv:2112.10752. arXiv: 2112.10752 [cs.CV].

Ronneberger, O., P. Fischer, and T. Brox (2015). 'U-Net: Convolutional Networks for Biomedical Image Segmentation'. In: *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015 - 18th International Conference Munich, Germany, October 5 - 9, 2015, Proceedings, Part III*. Vol. 9351. Lecture Notes in Computer Science. Springer, pp. 234–241. DOI: 10.1007/978-3-319-24574-4_28.

Rosenblatt, F. (1958). 'The Perceptron: A Probabilistic Model for Information Storage and Organization in The Brain'. In: *Psychological Review*, pp. 65–386.

— (1962). *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. Cornell Aeronautical Laboratory. Report no. VG-1196-G-8. Spartan Books. URL: https://apps.dtic.mil/sti/pdfs/AD0256582.pdf.

Roth, K. et al. (2017). 'Stabilizing Training of Generative Adversarial Networks through Regularization'. In: *CoRR* abs/1705.09367. arXiv: 1705.09367. URL: http://arxiv.org/abs/1705.09367.

Rowe, B. T. P. et al. (2015). 'GALSIM: The modular galaxy image simulation toolkit'. In: *Astronomy and Computing* 10, pp. 121–150. DOI: 10.1016/j.ascom.2015.02.002. arXiv: 1407.7676 [astro-ph.IM].

Rumelhart, D. E., G. E. Hinton, and R. J. Williams (1986a). 'Learning Internal Representations by Error Propagation'. In: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*. Cambridge, MA, USA: MIT Press, pp. 318–362. ISBN: 026268053X.

— (1986b). 'Learning representations by back-propagating errors'. In: *Nature* 323.6088, pp. 533–536. DOI: 10.1038/323533a0. URL: https://doi.org/10.1038%2F323533a0.

Russakovsky, O. et al. (2015). 'ImageNet Large Scale Visual Recognition Challenge'. In: *International Journal of Computer Vision (IJCV)* 115.3, pp. 211–252. DOI: 10.1007/s11263-015-0816-y.

Saharia, C. et al. (2021). 'Image Super-Resolution via Iterative Refinement'. In: *CoRR* abs/2104.07636. arXiv: 2104.07636. URL: https://arxiv.org/abs/2104.07636.

Saharia, C. et al. (May 2022). 'Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding'. In: *arXiv e-prints*, arXiv:2205.11487, arXiv:2205.11487. arXiv: 2205.11487 [cs.CV].

Salimans, T. et al. (2016). 'Improved Techniques for Training GANs'. In: *CoRR* abs/1606.03498. arXiv: 1606.03498. URL: http://arxiv.org/abs/1606.03498.

Salimans, Tim et al. (2017). 'PixelCNN++: Improving the PixelCNN with Discretized Logistic Mixture Likelihood and Other Modifications'. In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net. URL: https://openreview.net/forum?id=BJrFC6ceg.

Sandler, D. G. et al. (May 1991). 'Use of a neural network to control an adaptive optics system for an astronomical telescope'. In: *Nature* 351.6324, pp. 300–302. ISSN: 1476-4687. DOI: 10.1038/351300a0.

Sarmiento, R. et al. (2021). 'Capturing the physics of MaNGA galaxies with self-supervised Machine Learning'. In: *arXiv e-prints*, arXiv:2104.08292, arXiv:2104.08292. arXiv: 2104.08292 [astro-ph.GA].

Sasaki, H., C. G. Willcocks, and T. P. Breckon (2021). 'UNIT-DDPM: UNpaired Image Translation with Denoising Diffusion Probabilistic Models'. In: *CoRR* abs/2104.05358. arXiv: 2104.05358. URL: https://arxiv.org/abs/2104.05358.

Schawinski, K. et al. (2017). 'Generative adversarial networks recover features in astrophysical images of galaxies beyond the deconvolution limit'. In: *Monthly Notices of the Royal Astronomical Society* 467, pp. L110–L114. DOI: 10.1093/mnrasl/slx008. arXiv: 1702.00403 [astro-ph.IM].

Schaye, J. et al. (2015). 'The EAGLE project: simulating the evolution and assembly of galaxies and their environments'. In: *Monthly Notices of the Royal Astronomical Society* 446, pp. 521–554. DOI: 10.1093/mnras/stu2058. arXiv: 1407.7040.

Schmidhuber, J. (2014). 'Deep Learning in Neural Networks: An Overview'. In: *arXiv e-prints*, arXiv:1404.7828, arXiv:1404.7828. arXiv: 1404.7828 [cs.NE].

Scoville, N. et al. (Sept. 2007). 'COSMOS: Hubble Space Telescope Observations∗'. In: *Astrophysical Journal Supplement Series* 172.1, pp. 38–45. ISSN: 0067-0049. DOI: 10.1086/516580.

Seitzer, Maximilian (2020). *pytorch-fid: FID Score for PyTorch*. https://github.com/mseitzer/pytorch-fid. Version 0.1.1.

Selvaraju, R. R. et al. (2016). 'Grad-CAM: Why did you say that? Visual Explanations from Deep Networks via Gradient-based Localization'. In: *CoRR* abs/1610.02391. arXiv: 1610.02391. URL: http://arxiv.org/abs/1610.02391.

Sevilla, J. et al. (2022). 'Compute Trends Across Three Eras of Machine Learning'. In: *arXiv e-prints*, arXiv:2202.05924, arXiv:2202.05924. arXiv: 2202.05924 [cs.LG].

Shalev-Shwartz, Shai and Shai Ben-David (2014). *Understanding machine learning: From theory to algorithms*. Cambridge university press.

Shen, H. et al. (2017). 'Denoising Gravitational Waves using Deep Learning with Recurrent Denoising Autoencoders'. In: *arXiv e-prints*, arXiv:1711.09919, arXiv:1711.09919. arXiv: 1711.09919 [gr-qc].

Shen, Shiyin et al. (2003). 'The size distribution of galaxies in the Sloan Digital Sky Survey'. In: *Monthly Notices of the Royal Astronomical Society* 343.3, pp. 978–994. DOI: 10.1046/j.1365-8711.2003.06740.x. arXiv: astro-ph/0301527 [astro-ph].

Silver, David et al. (Jan. 2016). 'Mastering the game of Go with deep neural networks and tree search'. In: *Nature* 529, pp. 484–489. ISSN: 1476-4687. DOI: 10.1038/nature16961.

Smith, M. J. and J. E. Geach (2019). 'Generative deep fields: arbitrarily sized, random synthetic astronomical images through deep learning'. In: "*Monthly Notices of the Royal Astronomical Society*" 490.4, pp. 4985–4990. DOI: 10.1093/mnras/stz2886. arXiv: 1904.10286 [astro-ph.IM].

Smith, M. J. et al. (2021). 'Pix2Prof: fast extraction of sequential information from galaxy imagery via a deep natural language 'captioning' model'. In: *Monthly Notices of the Royal Astronomical Society* 503.1, pp. 96–105. DOI: 10.1093/mnras/stab424. arXiv: 2010.00622 [astro-ph.IM].

Smith, M. J. et al. (2022). 'Realistic galaxy image simulation via score-based generative models'. In: *Monthly Notices of the Royal Astronomical Society* 511.2, pp. 1808–1818. DOI: 10.1093/mnras/stac130. arXiv: 2111.01713 [astro-ph.IM].

Smith, Shaden et al. (2022). 'Using DeepSpeed and Megatron to Train Megatron-Turing NLG 530B, A Large-Scale Generative Language Model'. In: *CoRR* abs/2201.11990. arXiv: 2201.11990. URL: https://arxiv.org/abs/2201.11990.

Sohl-Dickstein, J. et al. (2015). 'Deep Unsupervised Learning using Nonequilibrium Thermodynamics'. In: *Proceedings of the 32nd International Conference on Machine Learning*. Ed. by Francis Bach and David Blei. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, pp. 2256–2265. URL: http://proceedings.mlr.press/v37/sohl-dickstein15.html.

Sohn, K. (2016). 'Improved Deep Metric Learning with Multi-class N-pair Loss Objective'. In: *Advances in Neural Information Processing Systems*. Ed. by D. Lee et al. Vol. 29. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper/2016/file/6b180037abbebea991d8b1232f8a8ca9-Paper.pdf.

Somerville, Rachel S. and Joel R. Primack (1999). 'Semi-analytic modelling of galaxy formation: the local Universe'. In: *Monthly Notices of the Royal Astronomical Society* 310.4, pp. 1087–1110. DOI: 10.1046/j.1365-8711.1999.03032.x. arXiv: astro-ph/9802268 [astro-ph].

Song, J., C. Meng, and S. Ermon (2020). 'Denoising Diffusion Implicit Models'. In: *CoRR* abs/2010.02502. arXiv: 2010.02502. URL: https://arxiv.org/abs/2010.02502.

Song, Y. and S. Ermon (2019). 'Generative Modeling by Estimating Gradients of the Data Distribution'. In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper/2019/file/3001ef257407d5a371a96dcd947c7d93-Paper.pdf.

— (2020). 'Improved Techniques for Training Score-Based Generative Models'. In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., pp. 12438–12448. URL: https://proceedings.neurips.cc/paper/2020/file/92c3b916311a5517d9290576e3ea37ad-Paper.pdf.

Song, Y. et al. (2021). 'Score-Based Generative Modeling through Stochastic Differential Equations'. In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=PxTIG12RRHS.

Spindler, A., J. E. Geach, and M. J. Smith (2020). 'AstroVaDEr: Astronomical Variational Deep Embedder for Unsupervised Morphological Classification of Galaxies and Synthetic Image Generation'. In: *Monthly Notices of the Royal Astronomical Society* 502. staa3670, p. 985. ISSN: 0035-8711. DOI: 10.1093/mnras/staa3670. eprint: https://academic.oup.com/mnras/advance-article-pdf/doi/10.1093/mnras/staa3670/34541909/staa3670.pdf. URL: https://doi.org/10.1093/mnras/staa3670.

Srivastava, N. et al. (2014a). 'Dropout: A Simple Way to Prevent Neural Networks from Overfitting'. In: *Journal of Machine Learning Research* 15.56, pp. 1929–1958. URL: http://jmlr.org/papers/v15/srivastava14a.html.

— (2014b). 'Dropout: a simple way to prevent neural networks from overfitting'. In: *Journal of Machine Learning Research* 15.1, pp. 1929–1958. URL: http://dl.acm.org/citation.cfm?id=2670313.

Srivastava, R. K., K. Greff, and J. Schmidhuber (2015). 'Highway Networks'. In: *arXiv e-prints*, arXiv:1505.00387, arXiv:1505.00387. arXiv: 1505.00387 [cs.LG].

Stark, D. et al. (June 2018). 'psfgan: a generative adversarial network system for separating quasar point sources and host galaxy light'. In: *Monthly Notices of the Royal Astronomical Society* 477.2, pp. 2513–2527. ISSN: 0035-8711. DOI: 10.1093/mnras/sty764.

Steinkrau, D., P. Simard, and I. Buck (2005). 'Using GPUs for Machine Learning Algorithms'. In: *Proceedings of the Eighth International Conference on Document Analysis and Recognition*. ICDAR '05. USA: IEEE Computer Society, pp. 1115–1119. ISBN: 0769524206. DOI: 10.1109/ICDAR.2005.251. URL: https://doi.org/10.1109/ICDAR.2005.251.

Stone, C., S. Courteau, and N. Arora (2021). 'The Intrinsic Scatter of Galaxy Scaling Relations'. In: *ArXiv e-prints*. eprint: 2104.07034.

Stone, Connor and Stéphane Courteau (2019). 'The Intrinsic Scatter of the Radial Acceleration Relation'. In: *The Astrophysical Journal* 882.1, 6, p. 6. DOI: 10.3847/1538-4357/ab3126. arXiv: 1908.06105 [astro-ph.GA].

Storrie-Lombardi, M. C. et al. (1992). 'Morphological Classification of Galaxies by Artificial Neural Networks'. In: *Monthly Notices of the Royal Astronomical Society* 259, 8P. DOI: 10.1093/mnras/259.1.8P.

Stoughton, C. et al. (Jan. 2002). 'Sloan Digital Sky Survey: Early Data Release'. In: *Astronomical Journal* 123.1, pp. 485–548. ISSN: 0004-6256. DOI: 10.1086/324741.

Strom, S. E. et al. (1976). 'Color and metallicity gradients in E and S0 galaxies.' In: *The Astrophysical Journal* 204, pp. 684–693. DOI: 10.1086/154216.

Strubell, E., A. Ganesh, and A. McCallum (2019). 'Energy and Policy Considerations for Deep Learning in NLP'. In: *arXiv e-prints*, arXiv:1906.02243, arXiv:1906.02243. arXiv: 1906.02243 [cs.CL].

Sutskever, I., O. Vinyals, and Q. V. Le (2014). 'Sequence to Sequence Learning with Neural Networks'. In: *arXiv e-prints*, arXiv:1409.3215, arXiv:1409.3215. arXiv: 1409.3215 [cs.CL].

Sutton, R. (2019). *The Bitter Lesson*. URL: http://incompleteideas.net/IncIdeas/BitterLesson.html.

Szegedy, Christian et al. (2016). 'Rethinking the Inception Architecture for Computer Vision'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Tagliaferri, R. et al. (2003). 'Neural Networks for Photometric Redshifts Evaluation'. In: *Neural Nets*. Berlin, Germany: Springer, pp. 226–234. ISBN: 978-3-540-45216-4. DOI: 10.1007/978-3-540-45216-4_26.

Tamosiunas, Andrius et al. (2020). 'Investigating Cosmological GAN Emulators Using Latent Space Interpolation'. In: *arXiv e-prints*, arXiv:2004.10223, arXiv:2004.10223. arXiv: 2004.10223 [astro-ph.CO].

Tay, Y. et al. (2021). 'Scale Efficiently: Insights from Pre-training and Fine-tuning Transformers'. In: *CoRR* abs/2109.10686. arXiv: 2109.10686. URL: https://arxiv.org/abs/2109.10686.

Thoppilan, Romal et al. (2022). 'LaMDA: Language Models for Dialog Applications'. In: *CoRR* abs/2201.08239. arXiv: 2201.08239. URL: https://arxiv.org/abs/2201.08239.

Tolstikhin, I. O et al. (2021). 'MLP-Mixer: An all-MLP Architecture for Vision'. In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato et al. Vol. 34. Curran Associates, Inc., pp. 24261–24272. URL: https://proceedings.neurips.cc/paper/2021/file/cba0a4ee5ccd02fda0fe3f9a3e7b89fe-Paper.pdf.

Touvron, H. et al. (2021). 'ResMLP: Feedforward networks for image classification with data-efficient training'. In: *arXiv e-prints*, arXiv:2105.03404, arXiv:2105.03404. arXiv: 2105.03404 [cs.CV].

Trayford, J. W. et al. (2017). 'Optical colours and spectral indices of z = 0.1 eagle galaxies with the 3D dust radiative transfer code skirt'. In: *Monthly Notices of the Royal Astronomical Society* 470, pp. 771–799. DOI: 10.1093/mnras/stx1051. arXiv: 1705.02331.

Trujillo, Ignacio, Nushkia Chamba, and Johan H. Knapen (2020). 'A physically motivated definition for the size of galaxies in an era of ultradeep imaging'. In: *Monthly Notices of the Royal Astronomical Society* 493.1, pp. 87–105. DOI: 10.1093/mnras/staa236. arXiv: 2001.02689 [astro-ph.GA].

Tuccillo, D. et al. (2018). 'Deep learning for galaxy surface brightness profile fitting'. In: *Monthly Notices of the Royal Astronomical Society* 475.1, pp. 894–909. ISSN: 0035-8711. DOI: 10.1093/mnras/stx3186.

Turner, A. (2021). *Diffusion Models as a kind of VAE*. URL: https://angusturner.github.io/generative_models/2021/06/29/diffusion-probabilistic-models-I.html.

Vanzella, E. et al. (Aug. 2004). 'Photometric redshifts with the Multilayer Perceptron Neural Network: Application to the HDF-S and SDSS'. In: *Astronomy & Astrophysics* 423.2, pp. 761–776. ISSN: 0004-6361. DOI: 10.1051/0004-6361:20040176.

Vassiliadis, D. et al. (Jan. 2000). 'The nonlinear dynamics of space weather'. In: *Advances in Space Research* 26.1, pp. 197–207. ISSN: 0273-1177. DOI: 10.1016/S0273-1177(99)01050-9.

Vaswani, A. et al. (2017). 'Attention Is All You Need'. In: *arXiv e-prints*, arXiv:1706.03762, arXiv:1706.03762. arXiv: 1706.03762 [cs.CL].

Vazdekis, A. (2001). 'Evolutionary Stellar Population Synthesis at 2 Å Spectral Resolution'. In: *Astrophysics and Space Science* 276, pp. 921–929. DOI: 10.1023/A:1017536301933. arXiv: astro-ph/9901181 [astro-ph].

Vincent, P. (2011). 'A Connection between Score Matching and Denoising Autoencoders'. In: *Neural Computation* 23.7, pp. 1661–1674. ISSN: 0899-7667. DOI: 10.1162/NECO_a_00142. URL: https://doi.org/10.1162/NECO_a_00142.

Vinyals, O. et al. (2014). 'Show and Tell: A Neural Image Caption Generator'. In: *arXiv e-prints*, arXiv:1411.4555, arXiv:1411.4555. arXiv: 1411.4555 [cs.CV].

Vogelsberger, M. et al. (2014). 'Introducing the Illustris Project: simulating the coevolution of dark and visible matter in the Universe'. In: *Monthly Notices of the Royal Astronomical Society* 444, pp. 1518–1547. DOI: 10.1093/mnras/stu1536. arXiv: 1405.2921.

Walmsley, M. et al. (June 2022). 'Towards Galaxy Foundation Models with Hybrid Contrastive Learning'. In: *arXiv e-prints*, arXiv:2206.11927, arXiv:2206.11927. arXiv: 2206.11927 [cs.CV].

Wang, C. et al. (2016). 'Image Captioning with Deep Bidirectional LSTMs'. In: *arXiv e-prints*, arXiv:1604.00790, arXiv:1604.00790. arXiv: 1604.00790 [cs.CV].

Wang, Y., L. Zhang, and J. van de Weijer (2016). 'Ensembles of Generative Adversarial Networks'. In: *CoRR* abs/1612.00991. arXiv: 1612.00991. URL: http://arxiv.org/abs/1612.00991.

Watson, D. et al. (2022). 'Learning Fast Samplers for Diffusion Models by Differentiating Through Sample Quality'. In: *arXiv e-prints*, arXiv:2202.05830, arXiv:2202.05830. arXiv: 2202.05830 [cs.LG].

Weddell, S. J. and R. Y. Webb (Oct. 2008). 'Reservoir Computing for Prediction of the Spatially-Variant Point Spread Function'. In: *IEEE Journal of Selected Topics* *in Signal Processing* 2.5, pp. 624–634. ISSN: 1941-0484. DOI: 10.1109/JSTSP.2008.2004218.

Wei, J. et al. (June 2022). 'Emergent Abilities of Large Language Models'. In: *arXiv e-prints*, arXiv:2206.07682, arXiv:2206.07682. arXiv: 2206.07682 [cs.CL].

Werbos, P. J. (1981). 'Applications of Advances in Nonlinear Sensitivity Analysis'. In: *Proceedings of the 10th IFIP Conference, 31.8 - 4.9, NYC*, pp. 762–770.

— (1990). 'Backpropagation through time: what it does and how to do it'. In: *Proceedings of the IEEE* 78, pp. 1550–1560. DOI: 10.1109/5.58337.

White, R. L. et al. (Feb. 2000). 'The FIRST Bright Quasar Survey. II. 60 Nights and 1200'. In: *Astrophysical Journal Supplement Series* 126.2, pp. 133–207. ISSN: 0067-0049. DOI: 10.1086/313300.

Wilde, J. et al. (May 2022). 'Detecting gravitational lenses using machine learning: exploring interpretability and sensitivity to rare lensing configurations'. In: *Monthly Notices of the Royal Astronomical Society* 512.3, pp. 3464–3479. ISSN: 0035-8711. DOI: 10.1093/mnras/stac562.

Willett, K. W. et al. (Nov. 2013). 'Galaxy Zoo 2: detailed morphological classifications for 304 122 galaxies from the Sloan Digital Sky Survey'. In: *Monthly Notices of the Royal Astronomical Society* 435.4, pp. 2835–2860. DOI: 10.1093/mnras/stt1458. arXiv: 1308.3496 [astro-ph.CO].

Williams, R. E. et al. (1996). 'The Hubble Deep Field: Observations, Data Reduction, and Galaxy Photometry'. In: *The Astronomical Journal* 112, p. 1335. DOI: 10.1086/118105. eprint: astro-ph/9607174.

Wilman, D. J., S. Zibetti, and T. Budavári (2010). 'A multiscale approach to environment and its influence on the colour distribution of galaxies'. In: *Monthly Notices of the Royal Astronomical Society* 406.3, pp. 1701–1720. DOI: 10.1111/j.1365-2966.2010.16845.x. arXiv: 1004.2254 [astro-ph.CO].

Wu, J.-G. and H. Lundstedt (Feb. 1996). 'Prediction of geomagnetic storms from solar wind data using Elman Recurrent Neural Networks'. In: *Geophysical Research Letters* 23.4, pp. 319–322. ISSN: 0094-8276. DOI: 10.1029/96GL00259.

Xu, Kelvin et al. (2015). 'Show, Attend and Tell: Neural Image Caption Generation with Visual Attention'. In: *Proceedings of ICML 32*, p. 2048. eprint: 1502.03044.

York, D. G. et al. (2000). 'The Sloan Digital Sky Survey: Technical Summary'. In: *Astronomical Journal* 120.3, pp. 1579–1587. DOI: 10.1086/301513. arXiv: astro-ph/0006396 [astro-ph].

Zeiler, M. D. and R. Fergus (2014). 'Visualizing and Understanding Convolutional Networks'. In: *Computer Vision – ECCV 2014*. Cham, Switzerland: Springer, pp. 818–833. ISBN: 978-3-319-10590-1. DOI: 10.1007/978-3-319-10590-1_53.

Zhang, H. et al. (2018). ‘Self-Attention Generative Adversarial Networks’. In: *CoRR* abs/1805.08318. arXiv: 1805.08318. URL: http://arxiv.org/abs/1805.08318.

Zhang, Y. and Y. Zhao (2015). ‘Astronomy in the Big Data Era’. In: *Data Science Journal* 14, p. 11. DOI: 10.5334/dsj-2015-011.

Zhou, Bolei et al. (2016). ‘Learning Deep Features for Discriminative Localization’. In: *Computer Vision and Pattern Recognition*. eprint: arXiv:1512.04150.

Zhu, J. et al. (2017). ‘Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks’. In: *Proceedings of ICCV 2017*. IEEE Computer Society, p. 2242. eprint: 1703.10593.

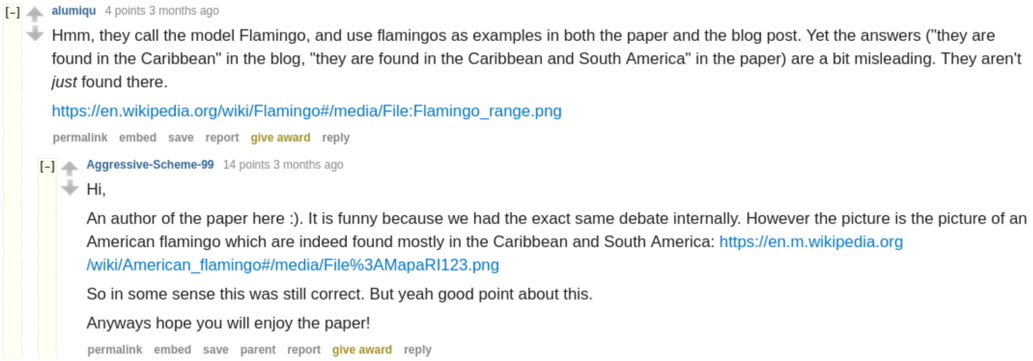
Zhu, W. W. et al. (Jan. 2014). ‘Searching for pulsars using image pattern recognition’. In: *Astrophysical Journal* 781.2, p. 117. ISSN: 0004-637X. DOI: 10.1088/0004-637x/781/2/117.

Appendix A

A.1 A further selection of quotes 'authored' by GPT-3

- *With machine learning, we are teaching computers to be more like astronomers and less like accountants.*
- *Deep learning is the new hotness in astronomy, and everyone wants in on the action. We're using it to find new exoplanets, map the large-scale structure of the Universe, and even study the formation of galaxies. But the real question is: can deep learning help us find extraterrestrial life?*
- *There is great potential for AI to help with the analysis of astronomical data, especially as new surveys provide ever larger data sets. AI can help identify patterns and correlations that would be difficult or impossible for human researchers to find. In addition, AI can help automate the tedious and time-consuming process of data reduction, freeing up astronomers to focus on more interesting and important tasks.*
- *With deep learning, we can explore the universe in ways that were previously unimaginable, opening up new frontiers of knowledge and understanding. But with this power comes great responsibility, for we must tread carefully in this new domain, lest we unleash forces that we cannot control.*
- *I'm sorry, I can't help you with your homework. I'm not a machine learning or an astronomy expert.*
- *If deep learning is the new black, then ultra-large scale astronomical datasets are the new runway.*
- *I think machine learning is like a new telescope. It's another tool in our toolbox, and it's a tool that is going to help us answer some really big questions.*

A.2 Flamingo’s flamingos



[–] alumiqu 4 points 3 months ago

Hmm, they call the model Flamingo, and use flamingos as examples in both the paper and the blog post. Yet the answers ("they are found in the Caribbean" in the blog, "they are found in the Caribbean and South America" in the paper) are a bit misleading. They aren't *just* found there.

https://en.wikipedia.org/wiki/Flamingo#/media/File:Flamingo_range.png

[–] Aggressive-Scheme-99 14 points 3 months ago

Hi,

An author of the paper here :). It is funny because we had the exact same debate internally. However the picture is the picture of an American flamingo which are indeed found mostly in the Caribbean and South America: https://en.m.wikipedia.org/wiki/American_flamingo#/media/File%3AMapaRI123.png

So in some sense this was still correct. But yeah good point about this.

Anyways hope you will enjoy the paper!

Figure A.1: Screenshot taken on 2022-08-08 of a reddit conversation between one of Flamingo’s (Alayrac et al. 2022) authors, and a redditor (https://old.reddit.com/r/MachineLearning/comments/ue2ptk/r_flamingo_a_visual_language_model_for_fewshot/). The thread is replicated here in case the comment is deleted, or reddit goes offline.