The Astronomical Journal, 155:94 (21pp), 2018 February https://doi.org/10.3847/1538-3881/aa9e09

Identifying Exoplanets with Deep Learning: A Five-planet Resonant Chain around Kepler-80 and an Eighth Planet around Kepler-90

Christopher J. Shallue[1] and Andrew Vanderburg[2,3,4]
[1] Google Brain, 1600 Amphitheatre Parkway, Mountain View, CA 94043, USA; shallue@google.com
[2] Department of Astronomy, The University of Texas at Austin, 2515 Speedway, Stop C1400, Austin, TX 78712, USA
[3] Harvard–Smithsonian Center for Astrophysics, 60 Garden Street, Cambridge, MA 02138, USA

Received 2017 September 19; revised 2017 November 13; accepted 2017 November 20; published 2018 January 30

Abstract

NASA's *Kepler Space Telescope* was designed to determine the frequency of Earth-sized planets orbiting Sun-like stars, but these planets are on the very edge of the mission's detection sensitivity. Accurately determining the occurrence rate of these planets will require automatically and accurately assessing the likelihood that individual candidates are indeed planets, even at low signal-to-noise ratios. We present a method for classifying potential planet signals using deep learning, a class of machine learning algorithms that have recently become state-of-the-art in a wide variety of tasks. We train a deep convolutional neural network to predict whether a given signal is a transiting exoplanet or a false positive caused by astrophysical or instrumental phenomena. Our model is highly effective at ranking individual candidates by the likelihood that they are indeed planets: 98.8% of the time it ranks plausible planet signals higher than false-positive signals in our test set. We apply our model to a new set of candidate signals that we identified in a search of known *Kepler* multi-planet systems. We statistically validate two new planets that are identified with high confidence by our model. One of these planets is part of a five-planet resonant chain around Kepler-80, with an orbital period closely matching the prediction by three-body Laplace relations. The other planet orbits Kepler-90, a star that was previously known to host seven transiting planets. Our discovery of an eighth planet brings Kepler-90 into a tie with our Sun as the star known to host the most planets.

Key words: methods: data analysis – planets and satellites: detection – techniques: photometric

1. Introduction

Over the course of its prime mission from 2009 May 2 to 2013 May 11, NASA's *Kepler Space Telescope* photometrically observed about 200,000 stars with unprecedented precision (Jenkins et al. 2010a; Koch et al. 2010; Christiansen et al. 2012), discovered thousands of transiting exoplanets (and thousands more planet candidates; Borucki et al. 2011a, 2011b; Batalha et al. 2013; Burke et al. 2014; Rowe et al. 2014, 2015; Mullally et al. 2015; Coughlin et al. 2016; Morton et al. 2016; S. E. Thompson et al. 2017, in preparation), and significantly improved our understanding of the population of small planets in the inner parts of planetary systems (Youdin 2011; Howard et al. 2012; Marcy et al. 2014; Dressing et al. 2015; Wolfgang & Lopez 2015).

Kepler was designed as a statistical mission, with the goal of determining the occurrence rate of Earth-sized planets in temperate orbits around Sun-like stars—that is, planets that might (in ideal circumstances) support life as we know it on Earth. But early in the mission, most of the major results coming from *Kepler* were the discoveries of individual particularly interesting planets or planetary systems. *Kepler* discovered the smallest transiting planets (Muirhead et al. 2012; Barclay et al. 2013), the closest transiting planets (Batalha et al. 2011; Jackson et al. 2013; Ofir & Dreizler 2013; Sanchis-Ojeda et al. 2013, 2014), the most-distant transiting planets (Wang et al. 2015a; Foreman-Mackey et al. 2016; Kipping et al. 2016), and some of the most interesting and unusual systems in terms of orbital architectures (Doyle et al. 2011; Lissauer et al. 2011; Cabrera et al. 2014). As such, the first lists of planet candidates (Borucki et al. 2011a, 2011b; Batalha et al. 2013) produced by the *Kepler* team were assembled in a heterogeneous way in an effort to include as many as possible. Most of the first lists of planet candidates were produced from lists of "threshold crossing events" (TCEs; detected periodic signals that may be consistent with transiting planets) from the *Kepler* pipeline (Jenkins et al. 2010b, 2010c), which were manually culled by humans to remove false positives caused by astrophysical variability and instrumental artifacts. A few candidates came from other sources as well; Batalha et al. (2013) listed four planet candidates identified by eye by volunteers for the Planet Hunters project (Fischer et al. 2012).

As the *Kepler* mission matured and the list of planet candidates grew, the focus of the community shifted toward population-level studies (Dressing & Charbonneau 2013, 2015; Fressin et al. 2013; Petigura et al. 2013a, 2013b; Foreman-Mackey et al. 2014; Burke et al. 2015; Mulders et al. 2015a, 2015b). This shift in focus also manifested in changing attitudes toward the way planet candidate lists were produced and vetted. While heterogeneous catalogs that rely on human judgment are an efficient way to spur follow-up observations of *Kepler*'s numerous exciting planet candidates, they are not well suited for population studies, which rely on uniformity. Uniform and automatically produced catalogs make it possible to characterize both precision (the fraction of detections that are true planets) and recall (the fraction of true planets that are detected), which are essential quantities for estimating population-level occurrence rates. Midway through the mission, efforts were made to produce planet candidate catalogs in an automated way, relying less and less on human

[4] National Science Foundation Graduate Research Fellow and NASA Sagan Fellow.

judgment in what became a planet candidate. Projects such as the Robovetter (Coughlin et al. 2016), a decision tree designed to mimic the manual process for rejecting false-positive TCEs, progressed to the point that they could be used in lieu of human vetting to produce fully automated catalogs (Catanzarite 2015; Coughlin et al. 2016; S. E. Thompson et al. 2017, in preparation).

Meanwhile, others began to explore the use of machine learning for automatically vetting *Kepler* TCEs. The Autovetter project (McCauliff et al. 2015) used a random forest model to classify TCEs based on features derived from *Kepler* pipeline statistics. A similar approach was presented by Mislis et al. (2016). Thompson et al. (2015) and Armstrong et al. (2017) used unsupervised machine learning to cluster *Kepler* light curves with similar shapes and defined classification metrics using the distances between new light curves and TCEs with known labels. The *LPP* metric defined by Thompson et al. (2015) is used by the Robovetter to filter out light curves with “not transit like” shapes. In addition to using machine learning on transiting planets in *Kepler*, others have used machine learning to identify different types of candidate signals in *Kepler* and other data sets. Millholland & Laughlin (2017) used supervised learning to identify candidate *nontransiting* planets in *Kepler* data, and Dittmann et al. (2017) used a neural network to identify the most likely real transits among many candidate events in the MEarth data set.

In this paper, we present a deep neural network for automatically vetting *Kepler* TCEs. Our model uses light curves as inputs and is trained on a set of human-classified *Kepler* TCEs. Neural networks have previously been applied to a variety of astronomical problems. Pearson et al. (2017) used a neural network to detect planet transits in simulated light curves. Our best model has similarities with those of Cabrera-Vives et al. (2017) for transient detection and Schaefer et al. (2017) for strong gravitational lens detection. Like those papers, we feed views of our inputs through separate convolutional columns, a technique that has previously been successful for image classification (Cireşan et al. 2012). Our model is able to distinguish with good accuracy the subtle differences between genuine transiting exoplanets and false positives like eclipsing binaries, instrumental artifacts, and stellar variability.

This paper is organized as follows. First, in Section 2, we introduce the concept of neural networks and describe several different neural network architectures that may be useful for problems like vetting transiting planet candidates. We describe the inputs to our neural network models in Section 3 and describe how we designed and trained our models in Section 4. In Section 5, we analyze the performance of our models and visualize some of the features they learned about transiting planets and false-positive signals. In Section 6, we use our best model to rank TCEs from a new search of stars observed by *Kepler* that are already known to host multiple transiting planet candidates. We discover several new planet candidates and apply further scrutiny to statistically validate the best few candidates. In Section 7, we discuss two newly discovered planets—Kepler-80 g and Kepler-90 i—and we also discuss the prospects for using our technique for occurrence-rate calculations in the future.

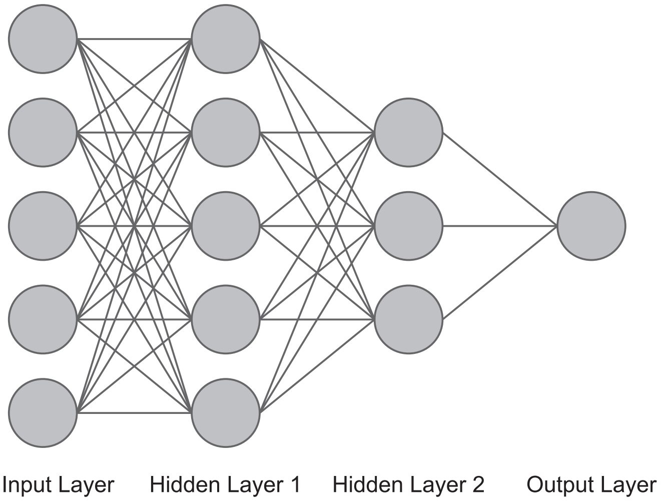

Figure 1. Fully connected neural network. Each connection represents a multiplicative weight parameter learned by the model. Inputs are fed into the first layer of the network, the output layer generates predictions, and the hidden layers represent a hierarchy of learned features.

2. Deep Neural Networks

2.1. Introduction

The goal of machine learning is to combine input features into useful outputs. For example, we might wish to build an image classification model that learns to assign labels (e.g., “dog,” “car,” “tree”) to input images. To build this model, we could manually define the input features that our model needs to make classification decisions (e.g., “number of legs” and “number of wheels” would be useful features for distinguishing between dogs and cars). However, defining those features in terms of image pixels is extremely difficult. Instead, we could build a model that learns features automatically from data: this is called *representation learning*.

Deep learning, which is a type of representation learning, uses computational layers to build increasingly complex features that are useful—in particular—for classification problems (Lecun et al. 2015). For example, a deep image classification model might first detect simple edge features, which can then be used to detect curves and corners, and so on, until the model’s final feature layer can discriminate between complex objects. Deep *neural networks*, which are a type of deep learning model, have recently become the state-of-the-art in a variety of tasks (e.g., image classification; Krizhevsky et al. 2012) and often outperform models built with hand-designed features.

2.2. Fully Connected Neural Networks

Figure 1 shows a *fully connected neural network* (also known as a *multilayer perceptron* or *feed-forward neural network*). Its layers, which are comprised of scalar-valued units called *neurons*, are arranged hierarchically: the outputs from one layer are the inputs to the next. The output value of a neuron for a specific set of inputs is called its *activation*. The activations of the first layer (the *input layer*) are the inputs to the network, and the activations of the final layer (the *output layer*) are the network’s predictions given those inputs. All other layers are *hidden layers* because their activations are not directly observed.

Hidden layer activations are defined by the equation

$$\boldsymbol{a}_i = \phi(\boldsymbol{W}_i \boldsymbol{a}_{i-1} + \boldsymbol{b}_i), \qquad (1)$$

where $\boldsymbol{a}_i$ is the vector of length n_i of activations in layer i, $\boldsymbol{W}_i$ is an $n_i \times n_{i-1}$ matrix of learned *weight* parameters, $\boldsymbol{b}_i$ is a vector of length n_i of learned *bias* parameters, and ϕ is an elementwise *activation function*.

The activation function ϕ is always nonlinear (a stack of linear layers would be equivalent to a single linear layer), so it is also called a *nonlinearity*. The optimal choice of ϕ is application dependent and is often determined experimentally. Popular choices include the sigmoid function $\phi(x) = 1/(1 + e^{-x})$, hyperbolic tangent $\phi(x) = \tanh(x)$, and linear rectification function $\phi(x) = \max\{0, x\}$. The former two choices are *saturating* activation functions because their derivatives approach zero as $x \to \pm\infty$, which can cause gradient-based optimization algorithms to converge slowly. The derivative of the linear rectification function does not diminish as $x \to +\infty$, which helps avoid this "vanishing gradient problem" and often results in faster training and better performance (Nair & Hinton 2010). Neurons with this activation function are called *rectified linear units* (ReLU).

For a classification problem, the output layer is normalized so that each neuron's value lies in (0, 1) and represents the probability of a specific output class. A binary classification problem (i.e., two output classes, {0, 1}) typically has a single output neuron representing the probability of the positive class. In this case, it is standard to use the sigmoid activation function for the output neuron because its range is (0, 1) and it induces a particularly convenient formula for gradient-based optimization.[5]

2.3. Convolutional Neural Networks

Fully connected neural networks, which densely connect every neuron in layer n to every neuron in layer $n + 1$, ignore any spatial structure present in the input. For example, every pixel in a 2D input image would be treated independently without using the fact that some pixels are located near each other. Features that are composed of neighboring pixels, like edges and shapes, need to be learned independently by the model for each location in the input. By contrast, convolutional neural networks (CNNs) exploit spatial structure by learning local features that are detected across the entire input; each feature only needs to be learned once. This significantly reduces the number of parameters that the model needs to learn and the number of computational operations required to compute the output.

A CNN typically consists of *convolutional layers* and *pooling layers* (Figure 2). The input to a (one-dimensional) convolutional layer is a stack of K vectors $\boldsymbol{a}_{i-1}^{(k)}$ $(k = 1, 2, \ldots, K)$ of length n_{i-1}, and the output is a stack of L vectors $\boldsymbol{a}_i^{(l)}$ $(l = 1,2, \ldots, L)$ of length n_i. The operation that takes the stack of K input vectors to the lth output vector is called a *feature map* and is defined by the operation

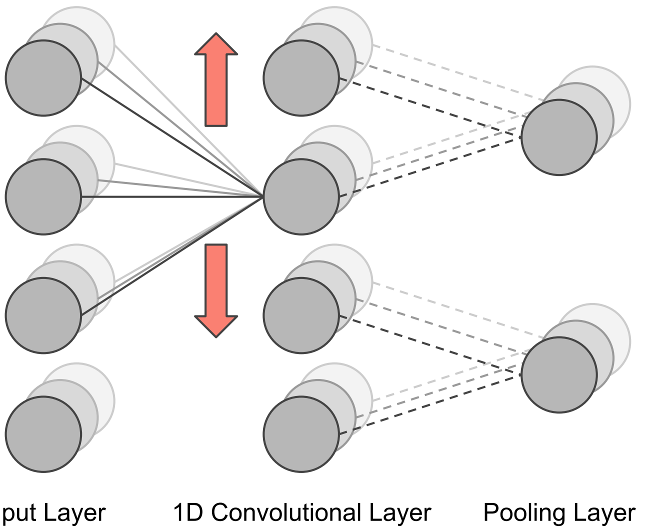

Figure 2. Three layers of a one-dimensional convolutional neural network. The convolutional layer takes the discrete cross-correlation operation of the vectors in its input layer with kernel vectors that are learned by the model. The pooling layer aggregates values in small neighborhoods along its input, typically by taking the mean or maximum value within each neighborhood. The pooling layer aggregates regions spaced s neurons apart, where s is the *stride length*; here, the stride length is 2.

$$\boldsymbol{a}_i^{(l)} = \phi\left(\sum_{k=1}^{K} \boldsymbol{w}_i^{(k,l)} * \boldsymbol{a}_{i-1}^{(k)} + \boldsymbol{b}_i^{(l)}\right), \qquad (2)$$

where $*$ is the discrete cross-correlation operation (colloquially called "convolution"), $\boldsymbol{w}_i^{(k,l)}$ is a vector of length m_i of learned parameters called the convolution *kernel* or *filter*, $\boldsymbol{b}_i^{(l)}$ is a vector of length n_i of learned bias parameters, and ϕ is an elementwise activation function. Typically, the kernel size is small (e.g., $m_i = 3$ or 5), and the feature map detects the presence of a local feature along its input.

A pooling layer aggregates values within small neighborhoods along its input, typically by taking the mean or maximum value within each neighborhood. This makes the network approximately invariant to small translations of the input, which is helpful if we care more about whether a particular feature is present than its precise location. A pooling layer typically aggregates regions spaced s neurons apart, rather than one neuron apart, where s is called the *stride length* (see Figure 2). This reduces the number of neurons in the pooling layer, which allows the feature maps in the next convolutional layer to have a wider view of the input—albeit at a lower resolution—without increasing the number of trainable parameters.

2.4. Neural Network Training

A neural network is trained to minimize a *cost function*, which is a measure of how far its predictions are from the true labels in its training set. The cross-entropy error function is the standard cost function for binary classification and is defined by the equation

$$C(\boldsymbol{w}) = -\frac{1}{M} \sum_{i=1}^{M} (y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)), \qquad (3)$$

[5] The sigmoid is the inverse of the "canonical link function" of the Bernoulli distribution. Assuming the model represents a conditional Bernoulli distribution of the target variable given the input, the inverse of the canonical link function induces a simple formula for the derivative of the log likelihood of the training data with respect to the model's parameters (Bishop 2006, Section 4.3.6).

where $\boldsymbol{w}$ is the vector of all parameters in the model, $y_1, y_2, \ldots, y_M$ are the true labels of all examples in the training set (defined to be either 0 or 1, depending on their classifications), and $\hat{y}_i$ is the model's predicted probability that $y_i = 1$ given $\boldsymbol{w}$.

The gradient of the cost function with respect to the model's parameters indicates how those parameters should be changed to reduce the value of the cost function. The model's parameters, which are initially random, are iteratively updated by descending along the gradient until a suitable minimum value of the cost function is reached. It is unnecessary and computationally expensive to compute the exact gradient of the cost function (e.g., computing the exact gradient of $C(\boldsymbol{w})$ in Equation (3) would involve iterating over the entire training set). Instead, each gradient step approximates the true gradient using a subset of the training set (called a *mini-batch*) of size B, where $1 \leqslant B \ll M$. Also, B is typically called the *(mini-) batch size*.

3. Creating our Training Set

How many are USPs?

3.1. TCEs and Labels

We derived our training set of labeled TCEs from the Autovetter Planet Candidate Catalog for Q1–Q17 DR24 (Catanzarite 2015), which is hosted at the NASA Exoplanet Archive.[6]

We sourced TCE labels from the catalog's av_training_set column, which has three possible values: planet candidate, astrophysical false positive (AFP), and nontransiting phenomenon (NTP). We ignored TCEs with the "unknown" label (UNK). These labels were produced by manual vetting and other diagnostics; see Catanzarite (2015) for full details. In total there are 3600 planet candidates, 9596 AFPs, and 2541 NTPs. We binarized the labels as "planet" (planet candidate) and "not planet" (AFP/NTP), but our model would only require minor changes to perform multi-class classification instead.

For the rest of this paper, we assume that these labels are the ground truth. However, this is not totally accurate—we discovered several mislabeled examples by manually examining a small subset of the full training set. For example, *Kepler* objects of interest (KOIs) 2950.01 and 5308.01 are eclipsing binary false positives and are labeled as such in the latest KOI table (Thompson et al. 2017), but both are labeled planet candidate in the Autovetter Catalog. We assume that the effects of mislabeling on our model and performance metrics are relatively minor.

We randomly partitioned our data into three subsets: training (80%), validation (10%), and test (10%). We used the validation set during development to choose model hyperparameters (Section 4.2), and we used the test set to evaluate final model performance (Section 5.1). This is the gold standard for evaluating model performance because the test set contains no data used to optimize the model in any way.

3.2. Light Curves

We downloaded light curves produced by the *Kepler* mission from the Mikulski Archive for Space Telescopes.[7] These light curves were produced by the *Kepler* pipeline (Jenkins et al. 2010b), which calibrates the pixels, identifies optimal photometric apertures, performs simple (stationary) aperture photometry, and removes common-mode instrumental artifacts using the Presearch Data Conditioning–Maximum A Posteriori (PDCMAP) method (Smith et al. 2012; Stumpe et al. 2012, 2014). Each light curve consists of integrated flux measurements spaced at 29.4 minute intervals for up to four years (approximately 70,000 points).

We then performed a few additional steps to prepare the light curves to be used as inputs to our neural network. For each TCE in the training set, we removed points corresponding to transits of any other confirmed planets in the system. Then, we "flattened" the light curve (that is, we removed low-frequency variability) by fitting a basis spline to the light curve and dividing it by the best-fit spline. To preserve transits, we removed the TCE's in-transit points while fitting the spline and linearly interpolated over those transits. We iteratively fit the spline, removed 3σ outliers, and refit the spline while interpolating over those outliers to prevent the spline from being "pulled" by discrepant points like cosmic-ray hits. This process is illustrated in Figure 3 of Vanderburg & Johnson (2014).

What kind of spline?

One parameter we had to choose when fitting the spline was the spacing of spline "knots" or "break points," between which the spline calculates its piecewise polynomial coefficients. Each *Kepler* light curve has different low-frequency stellar variability characteristics, so using one break-point spacing for all light curves is suboptimal. Instead, we chose the optimal spacing of spline break points for each light curve by fitting splines with different break-point spacings, calculating the Bayesian information criterion (BIC; Schwarz 1978) for each spline, and choosing the break-point spacing that minimized the BIC.

3.3. Input Representations

We generated our neural network inputs by folding each flattened light curve on the TCE period (with the event centered) and binning to produce a 1D vector.

To bin a folded light curve, we define a sequence of uniform intervals on the time axis with width δ and distance λ between midpoints and compute the median flux of the points falling within each interval. If we choose $\delta = \lambda$, then the intervals partition the time axis: each point falls in precisely one bin. If we choose $\delta > \lambda$, then the bins overlap, which reduces scatter and makes some transits more visible.

The simplest choice of λ would be a fixed constant, but this would result in a wide range of input lengths due to the wide range of TCE periods. Instead, we choose two TCE-specific values for λ, each of which generates a different "view" of the light curve (Figure 3).

We generate a *global view* of the light curve by choosing λ as a fraction of the TCE period. All light curves are binned to the same length, and each bin represents the same number of points, on average, across light curves. A disadvantage is that long-period TCEs may end up with very narrow transits that fall entirely within a small number of bins (Figure 3(b)).

We generate a *local view* of the transit by choosing λ as a fraction of the TCE duration. We consider k transit durations on either side of the event so that the transit occupies a fixed fraction of the resulting vector. This technique represents short- and long-period TCEs equally, but it only looks at part of the curve and therefore may miss important information, such as secondary eclipses (Figure 3(c)).

[6] http://exoplanetarchive.ipac.caltech.edu/

[7] http://archive.stsci.edu/

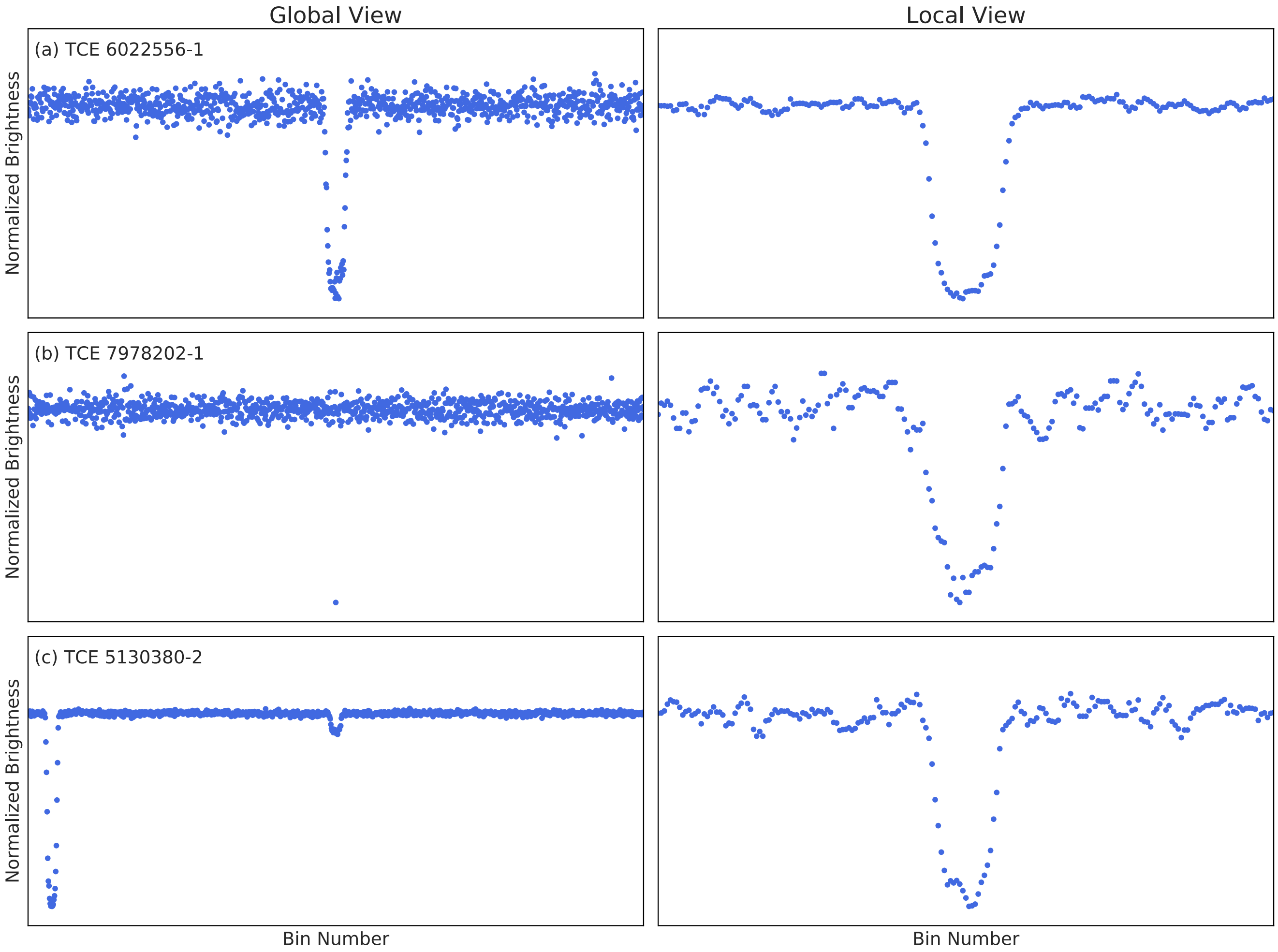

Figure 3. Light-curve representations that we use as inputs to our neural network models. The "global view" is a fixed-length representation of the entire light curve, and the "local view" is a fixed-length representation of a window around the detected transit. (a) Strong planet candidate. (b) Long-period planet where the transit falls into just one bin in the global view. (c) Secondary eclipse that looks like a planet in the local view.

Similar techniques are used by Armstrong et al. (2017; local binning) and Thompson et al. (2015; global binning away from the transit and local binning near the transit). Unlike those papers, we use these two representations as separate inputs to our model.

Finally, we normalize all light curves to have median 0 and minimum value −1 so that all TCEs have a fixed transit depth.

4. Automatic Vetting Models

4.1. Neural Network Architectures

We consider three types of neural network for classifying *Kepler* TCEs as either "planets" or "not planets." For each type, we consider three different input options: just the global view, just the local view, and both the global and local views.

1. *Linear architecture.* Our baseline model is a neural network with zero hidden layers (which is equivalent to a linear logistic regression model). This is a natural choice for its simplicity and popularity, but it makes the strong assumption that the input data is linearly separable—that is, planets and nonplanets are separated by a linear decision surface in the input space. If both global and local views are present, we concatenate them into a single input vector.

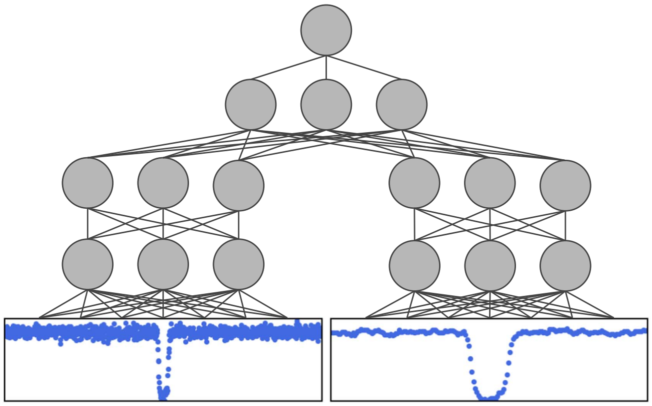

Figure 4. Fully connected neural network architecture for classifying light curves, with both global and local input views.

2. *Fully connected architecture.* A fully connected neural network is the most general type of neural network and makes the fewest assumptions about the input data. If both global and local views are present, we pass the two vectors through disjoint columns of fully connected layers before combining them in shared fully connected layers (Figure 4).

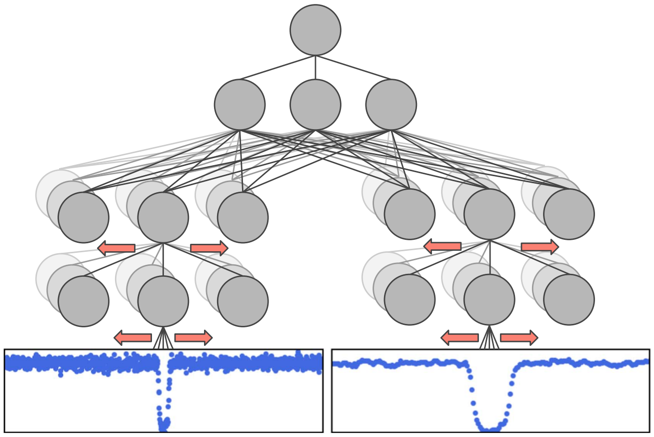

Figure 5. Convolutional neural network architecture for classifying light curves, with both global and local input views.

3. *Convolutional architecture.* CNNs have been tremendously successful in applications with spatially structured input data, including speech synthesis (Oord et al. 2016) and image classification (Krizhevsky et al. 2012). We use a one-dimensional CNN with max pooling. This architecture assumes that input light curves can be described by spatially local features, and that the output of the network should be invariant to small translations of the input. If both global and local views are present, we pass the two vectors through disjoint convolutional columns before combining them in shared fully connected layers (Figure 5).

All hidden layers use the ReLU (linear rectifier) activation function, and the output layer uses the sigmoid activation function. The output of each model is the predicted probability that the input is a transiting planet; values close to 1 indicate high confidence that the input is a transiting planet, and values close to 0 indicate high confidence that the input is a false positive.

4.2. Implementation and Training Procedure

We implemented our models in TensorFlow, an open-source software library for numerical computation and machine learning (Abadi et al. 2016). Our code is available online[8].

We used the Adam optimization algorithm (Kingma & Ba 2015) to minimize the cross-entropy error function over the training set (Section 2.4). We augmented our training data by applying random horizontal reflections to the light curves during training. We also applied dropout regularization to the fully connected layers, which helps prevent overfitting by randomly "dropping" some of the output neurons from each layer during training to prevent the model from becoming overly reliant on any of its features (Srivastava et al. 2014).

We used the Google-Vizier system for black-box optimization (Golovin et al. 2017) to automatically tune our hyperparameters, including those for the input representations (e.g., number of bins, bin width), model architecture (e.g., number of fully connected layers, number of convolutional layers, kernel size), and training (e.g., dropout probability). Each Vizier "study" trained several thousand models to find the hyperparameter configuration that maximized the area under the receiver operating characteristic curve (see Section 5.1) over the validation set. Each model was trained on a single central processing unit (CPU), and training time ranged from 10 minutes to 5 hours depending on the size of the model. Our best model (see Section 5.2) took 90 minutes to train. To speed up the hyperparameter search, we used 100 CPUs per study to train individual models in parallel. We ran many Vizier studies during model development as we iteratively improved our input representations and design decisions. We tuned each combination of architecture and input representation separately.

4.3. Model Averaging

Once we had selected the optimal hyperparameters for a particular architecture and input representation, we trained 10 independent copies with different random parameter initializations. We used the average outputs of these 10 copies for all predictions in Sections 5 and 6. This technique, known as *model averaging* (a type of ensembling), often improves performance because different versions of the same configuration may perform better or worse on different regions of the input space, especially when the training set is small and overfitting is more likely. It also makes different configurations more comparable by reducing the variance that exists between individual models.

5. Model Analysis

5.1. Test Set Performance

We use the following metrics to assess our model's performance.

1. *Precision*: the fraction of signals classified as planets that are true planets (also known as reliability; see, e.g., S. E. Thompson et al. 2017, in preparation).
2. *Recall*: the fraction of true planets that are classified as planets (also known as completeness).
3. *Accuracy*: the fraction of correct classifications.
4. *AUC*: the area under the receiver operating characteristic curve, which is equivalent to the probability that a randomly selected planet is scored higher than a randomly selected false positive.

The values of the first three metrics depend on the classification threshold we choose for our models. A natural choice is to classify a TCE as a planet candidate if its predicted probability of being a planet is above 0.5 and as a false positive if its predicted probability is below 0.5, but this threshold can be adjusted to trade-off precision versus recall. Increasing the threshold typically results in higher precision at the expense of lower recall, and vice versa. The value of the AUC is independent of the choice of classification threshold; it may be considered an aggregate performance metric over all classification thresholds.

Figure 6 shows precision versus recall for our three model architectures on our test set, which consists of 1523 TCEs that were not used to train our models or inform our hyperparameter decisions. All models in Figure 6 use both global and local input views. Our convolutional architecture performs the best, followed by our fully connected architecture. A full description of our best model's configuration is presented in Section 5.2.

Tables 1–2 show the accuracy and AUC, respectively, of each combination of model architecture and input representation on our test set. These tables show that, for all architectures, using both global and local views gives better performance than using either view individually. Interestingly, the local view (on its own) gives

[8] www.github.com/tensorflow/models/tree/master/research/astronet/

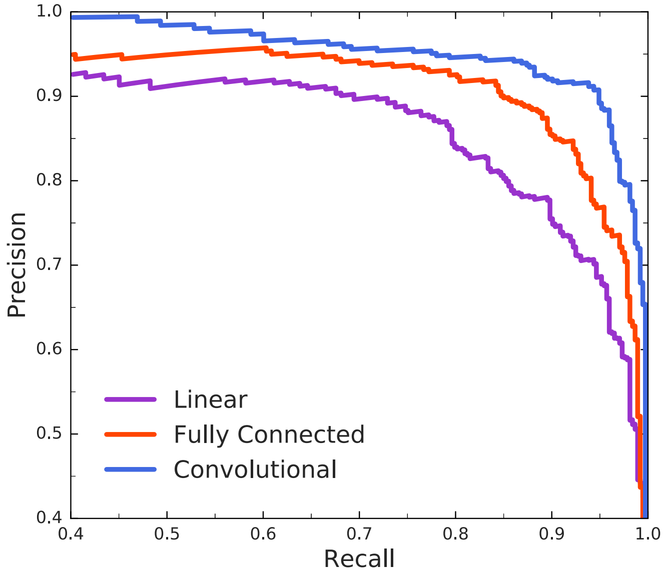

Figure 6. Precision vs. recall on the test set for our three neural network architectures. All models use both global and local input views. For each model, every point on its curve corresponds to a different choice of classification threshold. All TCEs predicted to be planets with a probability greater than the threshold are considered to be classified as planets, and all TCEs with a probability below the threshold are considered to be classified as false positives. Our convolutional architecture has the best overall performance. For example, with an appropriate choice of threshold, it has recall of 0.95 (that is, 95% of real planets are classified as planets) at precision of 0.90 (that is, 90% of its classifications are real planets).

Table 1
Test Set Accuracy

	Global	Local	Global and Local
Linear	0.869	0.879	0.917
Fully Connected	0.902	0.912	0.941
Convolutional	0.954	0.924	0.960

Note. Fraction of correct classifications on the test set for each choice of neural network architecture and input representation. For each model, all TCEs predicted to be planets with a probability greater than 0.5 are considered to be classified as planets, and all TCEs with a probability below 0.5 are considered to be classified as false positives.

Table 2
Test Set AUC

	Global	Local	Global and Local
Linear	0.922	0.933	0.963
Fully Connected	0.961	0.965	0.977
Convolutional	0.985	0.973	0.988

Note. AUC on the test set for each choice of neural network architecture and input representation. AUC is the probability that a randomly selected planet candidate is predicted to be a planet with higher probability than a randomly selected false positive. It measures a model's ability to rank TCEs: the maximum AUC value of 1 would indicate that all planets are ranked higher than all false positives.

better performance than the global view (on its own) in the linear and fully connected architectures, but the opposite is true in the convolutional architecture. We observed that the linear and fully connected architectures have difficulty distinguishing U-shaped transits (like planets) from V-shaped transits (like eclipsing binaries) when the transit width is not normalized in the global view, but the convolutional architecture is able to distinguish U-shaped transits from V-shaped transits at different widths and is also better at identifying secondary eclipses in the global view.

5.2. *Best Model Configuration*

Our best model, based on test set performance, uses the convolutional architecture with global and local input views. Its exact configuration is presented in Figure 7. Its hyperparameters were chosen to maximize AUC on the validation set using the procedure described in Section 4.2. Its performance on the test set is presented in Section 5.1.

We used a global view of length 2001 with $\lambda = \delta = p/2001$ (where p is the TCE period) and a local view of length 201 with $k = 4$ transit durations on either side of the event, $\lambda = 2kd/201$, and $\delta = 0.16d$ (where d is the TCE duration). See Section 3.3 for a description of these parameters.

We trained the model using a batch size of 64 for 50 epochs. We used the Adam optimization algorithm with $\alpha = 10^{-5}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$ (see Kingma & Ba 2015 for a description of these parameters). We did not use dropout regularization in this configuration.

5.3. *Visualizing the Model*

5.3.1. *Prediction Heat Maps*

One way to visualize a CNN is to systematically occlude portions of the input to see how the output prediction changes (Zeiler & Fergus 2014). We can use this technique to create heat maps showing the most important regions of the input contributing to a classification decision.

Figure 8 shows prediction heat maps of a convolutional model using the global view. We occluded points in a sliding window of width 50 points by setting their values to zero (Figure 8(a)). We computed the heat map value for each point by averaging the model's predicted probability of being a planet whenever that point was occluded.

Figure 8(b) shows that the transit is the most important region of the input for a planet candidate. The model's prediction drops to zero when the transit is occluded but is largely unchanged when any other region is blocked.

Figures 8(c) and (d) show that the model learns to identify secondary eclipses. The model's planet prediction increases when a secondary eclipse is occluded because we are hiding the evidence that this TCE is an eclipsing binary. Note that even when the secondary eclipse is blocked, the model's predictions are still low ($\lesssim 0.4$) in Figures 8(c) and (d), likely because the primary transit is V-shaped, which is also evidence of an eclipsing binary.

5.3.2. *t-SNE Embeddings*

We can consider the final hidden layer of a neural network to be a learned representation of the input such that a linear classifier can distinguish between the different classes. Geometrically, the values of the final hidden layer (which are points in d-dimensional space) are such that a linear boundary in $\mathbb{R}^d$ can separate planet candidates from false positives.

To visualize this space in two dimensions, we use t-distributed stochastic neighbor embedding (t-SNE; Maaten & Hinton 2008), a technique for dimensionality reduction that is well suited for visualizing high-dimensional data sets. Figure 9 shows a two-dimensional t-SNE embedding of the final layer activations of our best model using TCEs from the test set.

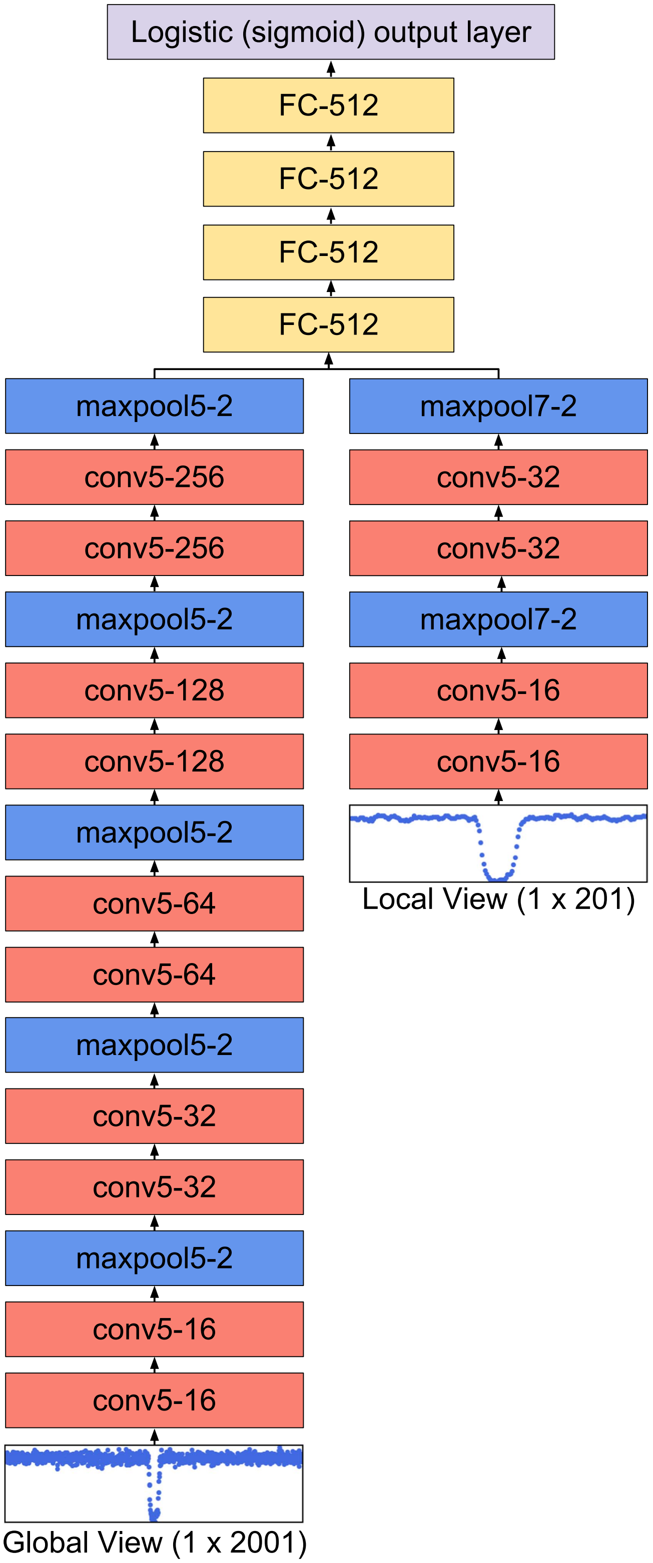

Figure 7. Architecture of our best-performing neural network model. Convolutional layers are denoted conv ⟨kernel size⟩-⟨number of feature maps⟩, max pooling layers are denoted maxpool ⟨window length⟩-⟨stride length⟩, and fully connected layers are denoted FC-⟨number of units⟩.

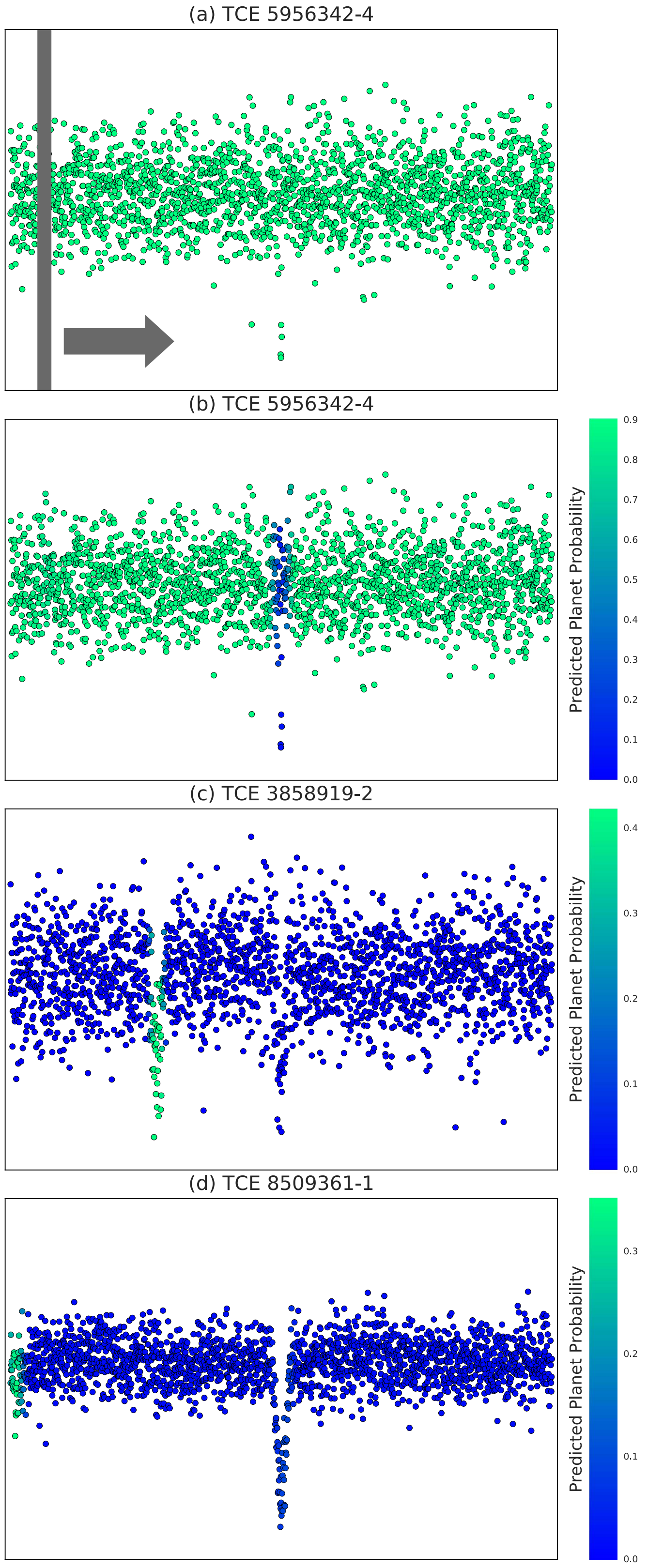

Figure 8. Visualizing the model's focus. Regions that are particularly important to the model's decision change the output prediction when occluded. The color scale for each point shows the mean predicted probability of being a planet over all predictions where that point was occluded. (a) We occluded points in a sliding window of width 50 points (shown to scale). (b) Transiting planet: the model's prediction drops to zero when the transit is occluded. (c) and (d) Eclipsing binaries: the model identifies secondary eclipses, even in subtle cases like (d).

5.4. Performance on Simulated Data

The *Kepler* team generated multiple kinds of simulated data to test the performance of the *Kepler* pipeline (which detects

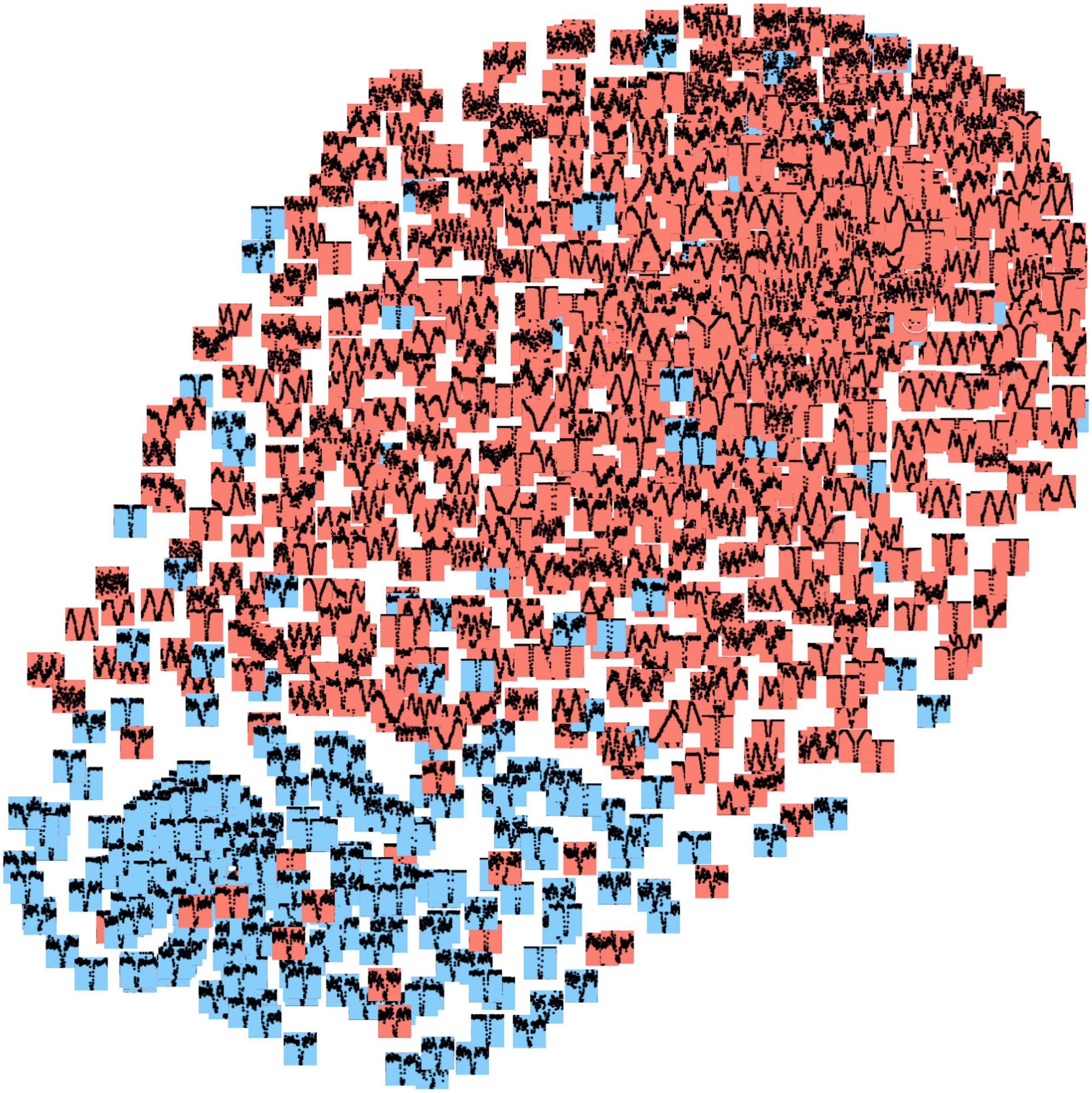

Figure 9. Visualizing the geometric space in which our convolutional neural network embeds its input light curves. We used the t-SNE algorithm for dimensionality reduction to create a two-dimensional embedding such that light curves that are close in the original space are also close in the 2D space. Light curves that are far apart in 2D space are not necessarily far apart in the original space. Notice that true planet candidates (blue) are close to other planet candidates, and false positives (red) are close to similarly shaped false positives.

TCEs in *Kepler* light curves) and the Robovetter (which classifies TCEs by automating the manual review process developed by the TCE Review Team; Christiansen 2017; Coughlin 2017; S. E. Thompson et al. 2017, in preparation). The Robovetter differs from machine learning vetting systems (like ours) in that its heuristics are explicitly defined by humans instead of learned automatically from data.[9]

The *Kepler* team recently released their simulated data products to the public, so we decided to test the effectiveness of our model at distinguishing simulated planets from simulated false positives. We considered the following simulated data sets, which are available at the NASA Exoplanet Archive.

1. *Injected Group 1* contains a single on-target injected periodic signal for 146,294 *Kepler* targets. Each injection simulates a transiting planet.
2. *Injected Group 2* contains a single off-target injected periodic signal for 22,978 *Kepler* targets. Each injection simulates an off-target transiting planet or eclipsing binary false positive.
3. *Injected Group 3* contains two on-target injected periodic signals, with a common period, for 9856 *Kepler* targets. Each injection simulates an eclipsing binary false positive.

[9] The Robovetter has two heuristics based on the *LPP* metric, which is computed by a machine learning model (Thompson et al. 2015), but the heuristics applied to this metric are explicitly defined.

Table 3
Simulated Data Classifications

Simulated Group	TCEs	Convolutional Model	Robovetter
Injected 1	45377	36826 (81.2%)	38668 (85.2%)
Injected 2	18897	17044 (90.2%)	8805 (46.6%)
Injected 3	10502	6852 (65.2%)	3932 (37.4%)
Inverted	19536	621 (3.18%)	95 (0.49%)
Inverted (cleaned)	14953	539 (3.60%)	75 (0.50%)

Notes. Number of simulated TCEs predicted to be planets by our convolutional neural network (exceeding the probability threshold of 0.5) and the Robovetter (classified as planet candidates). Injected Group 1 contains simulated planets, Injected Group 2 contains simulated background planets or eclipsing binaries, Injected Group 3 contains simulated eclipsing binaries, and the Inverted Group contains false positives from inverted light curves. We also include a "cleaned" subset of the Inverted Group, from which certain types of signals known to look like inverted transits have been removed (S. E. Thompson et al. 2017, in preparation).

4. *Inverted Group* contains 198,640 *Kepler* targets that have had their light curves inverted. Any detected transits are likely to be false positives.

The *Kepler* pipeline uses the multiple event statistic (MES; Jenkins et al. 2002) to measure the strength of a transit signal relative to the noise. A detected signal must have MES greater than 7.1 to be considered significant enough to become a TCE. Therefore, the *Kepler* pipeline only recovers a subset of the simulated planets and false positives as TCEs. Table 3 shows the number of these simulated TCEs classified as planets by our model with probability greater than 0.5, compared to the Robovetter. We discuss these results in the following sections.

5.4.1. Injected Group 1

Among the 45,377 simulated planets recovered by the *Kepler* pipeline, our model is similarly permissive to the latest version of the Robovetter, correctly classifying 36,826 of these TCEs as planets versus the Robovetter's 38,668. Among the full set of 146,294 TCEs (many of which were not recovered by the *Kepler* pipeline), we find that our model's predictions depend strongly on the MES of the simulated signal (see Figure 10). At an MES greater than 10, our model typically has high confidence that injected planet signals are real, but its confidence declines steeply from TCEs with MES = 10 to TCEs with MES = 5. The region between MES = 5 and MES = 10 is of particular interest because (a) it is the likely region of Earth-sized planets transiting Sun-like stars in temperate orbits, and (b) it is at the detection limit of the *Kepler* pipeline, where the recovery rate is low and there are likely to be as-yet-undiscovered planets (Christiansen et al. 2016; Christiansen 2017).

We suspect that we may be able to improve our model's performance on low signal-to-noise TCEs by including simulated transits like these in our training set. In our current implementation, only 8% of planets and planet candidates in our training set have an MES less than 10; our model rarely saw any low-MES positive examples during training and may have learned that low-MES events are a priori unlikely to be true planets. Including simulated low-MES planet candidates (as well as simulated low-MES false positives) in our training set may make our model more discriminating at low MES.

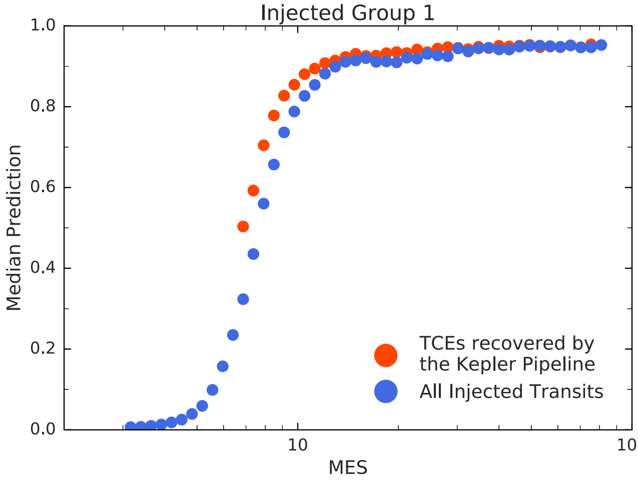

Figure 10. Median predicted probability of being a planet by our model vs. MES for all simulated planet transits in Injected Group 1. The red points show TCEs recovered by the *Kepler* pipeline at the detected period and epoch vs. MES measured by the *Kepler* pipeline. The blue points show injected TCEs at the injected period and epoch vs. "expected MES," which was the intended value of the MES when injecting the signal. Note that the measured MES is systematically slightly lower than expected MES (Christiansen 2017).

5.4.2. Injected Group 2

We find that our model classifies many simulated false-positive TCEs from Injected Group 2 as planets. The signals in Injected Group 2 were injected into the *Kepler* pixels off-target, simulating an astrophysical signal from a nearby star, which can mimic the transit of an on-target planet.[10] The fact that our model often classifies these signals as planets makes sense, because it does not take in any information about the location of the transits on the pixels, like centroid shifts during transit or in-transit difference imaging. These types of diagnostics can help identify when transit signals are coming from off-target sources. We have plans to incorporate this type of centroid or positional information into our model in the future to help identify this kind of false positive.

5.4.3. Injected Group 3

We find that our model predicts about 65% of the simulated eclipsing binary TCEs (with secondary events injected at the same orbital period) to be planets with a probability greater than 0.5. Our model is therefore not able to identify these false positives as well as the Robovetter, which only classifies about 37% of these simulated eclipsing binaries as planets. Figure 11 shows that our model rejects most simulated TCEs when the secondary signal is stronger than the primary signal but passes most simulated TCEs when the secondary signal is weaker than the primary signal. We know from Section 5.3.1 that our model can, in some cases at least, recognize subtle secondary eclipses (see Figure 8(d)), but it is evidently not sensitive enough to weak secondary eclipses to classify many of these simulated signals as false positives. It is possible that some of the simulated eclipsing binaries do not look like real examples from the training set (e.g., not V-shaped) or have a secondary eclipse that is too faint to detect in our representation. One way to increase our model's sensitivity to eclipsing binaries would

[10] We note that in many cases, the off-target signal could be caused by a transiting planet around a background star, but these signals are still typically classified as false positives.

So maybe pixel analysis?

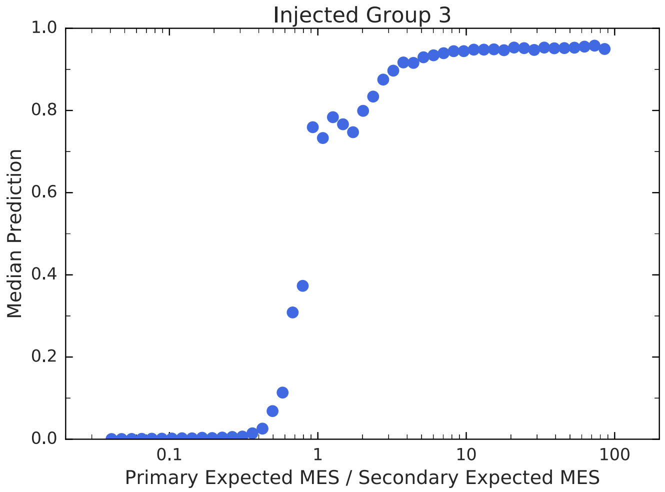

Figure 11. Median predicted probability of being a planet by our model vs. primary-to-secondary expected MES ratio for simulated eclipsing binaries in Injected Group 3 recovered by the *Kepler* pipeline.

be to include secondary and tertiary local views to give the model a "close-up" view of any putative secondary eclipses and avoid information about the shape of secondary events from being lost in coarsely binned global views. Including these simulated eclipsing binaries in our training set would likely help as well.

5.4.4. Inverted Group

We find that our model predicts about 3.2% of the false-positive TCEs in the Inverted Group to be planets with probabilities greater than 0.5, compared to the Robovetter, which only passes about 0.4%. One thing we discovered when examining these false positives is that our light-curve "flattening" technique (Section 3.2) occasionally makes non-transit-like phenomena look like transits. Specifically, the issue arises when we remove "in-transit" points before fitting the basis spline and linearly interpolate over those points. If the light curve sits below the linear interpolation, then the flattened light curve gets an artificial dip that sometimes looks like a transit. We mainly observed this in light curves with high-frequency stellar variability. We originally decided to remove in-transit points to prevent the basis splines from being distorted by real transits. We could mitigate this issue in the future by improving our flattening routine (for example, by interpolating nonlinearly over in-transit points) or including views of the data with alternative detrending methods, which might give different or better representations of the putative transits. We also noticed that some of the model's confident predictions were spurious signals that had some features in common with real planet candidates but were easily distinguishable by eye. Signals like these were not present in our training set, which limited our model's ability to properly classify them as false positives. Our model would likely benefit from including this set of false positives in our training set to further improve its ability to reject spurious signals.

5.4.5. Summary

There are several caveats to keep in mind when using these simulated data to assess our model's performance. First, inverted light-curve tests do not reproduce all known types of false positives (e.g., those caused by sudden dropouts in pixel sensitivity caused by cosmic rays), so they do not assess our model's ability to identify these signals. On the other hand, inverted light-curve tests may introduce new types of false positives not present in normal transit searches or our training set, which our model is not well equipped to identify. Second, some of the simulated false positives may not be physically realistic. For example, in many cases, eclipsing binaries with nearly equal primary and secondary eclipse depths will have V-shaped eclipses, which are not necessarily reproduced in these tests. Finally, we have not yet used these simulated data to train our model, as was done for the Robovetter (whose metrics were tuned for optimal performance on the simulated data; S. E. Thompson et al. 2017, in preparation). The performance of our model would likely substantially improve if we were to include these simulated data in our training set.

All in all, while our model in general performs well on these tests, it does not yet match the performance of the more mature Robovetter in identifying simulated false positives. The failure modes we have observed in these tests highlight limitations in our current model implementation, which will have to be addressed before our model is ready to blindly classify TCEs to determine planet occurrence rates. We are planning to work on improving our model's performance on these types of false positives, and we discuss some avenues for future work in Section 7. In the meantime, our model is still able to quite accurately sort true planets from planet candidates (even if it does require some human supervision to assure reliable classifications). We take advantage of this in Section 6 when we use our model to sort TCEs from a new transit search and identify the most likely new planets.

5.5. Comparisons to Other Automatic Vetting Systems

5.5.1. Robovetter

We computed the performance of the Robovetter on our test set as an additional way of comparing it with our model. To do this, we matched the latest Robovetter classifications from the DR25 KOI table (S. E. Thompson et al. 2017, in preparation) with our training labels from the DR24 Autovetter Planet Candidate Catalog (Catanzarite 2015). Note that the Robovetter only promotes the most plausible subset of TCEs to KOIs (S. E. Thompson et al. 2017, in preparation), so the DR25 KOI table contains only a subset of our test set (661 out of 1523). We assumed that all missing TCEs were classified by the Robovetter as false positives.[11]

To compute the Robovetter's metrics, we assumed our training labels were the ground truth.[12] We used the Robovetter's "disposition score" to rank the TCEs and assumed that any TCEs missing from the DR25 KOI table were rejected by the Robovetter with disposition score zero. Under these assumptions, the Robovetter has an AUC of 0.974 (compared to our model's 0.988) and accuracy of 0.974 (compared to our model's 0.960).

[11] It is possible that some of the missing TCEs were not detected by the DR25 *Kepler* pipeline due to changes in its Transiting Planet Search module between DR24 and DR25. This likely had negligible impact on our calculations because we computed nearly identical accuracy values for the DR24 (Coughlin et al. 2016) and DR25 Robovetters.

[12] We made this assumption because our labels were assigned by human reviewers (Catanzarite 2015). Some TCEs in our test set appear to be misclassified, but usually our model and the Robovetter agree on these TCEs.

5.5.2. Autovetter

The Autovetter (Catanzarite 2015; McCauliff et al. 2015) is a machine learning system for classifying *Kepler* TCEs into three classes (planet candidates, AFPs, and NTPs). Most of its input features are derived from outputs of the *Kepler* pipeline, including fitted transit parameters (e.g., period), stellar parameters (e.g., effective temperature), and signal-to-noise statistics (e.g., MES). An advantage of this approach is that features can be chosen that are known a priori to correlate with one of the output classes. For example, the Autovetter has a feature that indicates whether a TCE's period is close to the period of another TCE on the same star, which would indicate that the TCE is an astrophysical false positive caused by starspots or an eclipsing binary. A disadvantage is that the Autovetter depends on the *Kepler* pipeline. We could not have used the Autovetter to rank our new TCEs in Section 6 because we discovered those TCEs using an alternative detection pipeline.

Our approach differs from the Autovetter in that we directly model the shapes of light curves, rather than statistics derived from light curves and other sources. However, the two approaches are not mutually exclusive: the Autovetter includes the *LPP* metric (Thompson et al. 2015), which encodes information about light-curve shape, and we intend to experiment with auxiliary input features in our model in the future (see Section 7.2).

The Autovetter's accuracy on its training set is 0.973, or 0.986 if its classification objective is simplified to "planet" versus "not planet" (Catanzarite 2015). This is not directly comparable to our model's test set accuracy because the Autovetter reweights its random forest votes to minimize the total number of misclassifications across its training set (Catanzarite 2015; McCauliff et al. 2015).

Empirically, we found that the main difference between our model's classifications and the Autovetter's is on the set of false-positive TCEs whose signals originate on stars other than the intended target. We found that our model misclassifies many of these TCEs as planets (see Section 5.4.2), whereas the Autovetter correctly classifies them as false positives. The Autovetter has multiple features related to centroid shifts during transits, but our model does not currently take any information about the location of the transits on the pixels. We plan to incorporate this information into our model in the future (see Section 7.2).

5.5.3. Armstrong et al. (2017)

Armstrong et al. (2017) presented a method for classifying *Kepler* TCEs using an unsupervised machine learning algorithm to cluster light curves with similar shapes. They reported accuracy of 0.863 on *Kepler* planet candidates (compared to our model's 0.949) and 0.875 on *Kepler* false positives (compared to our model's 0.963).

6. Testing on New Candidates

6.1. Transit Search

We tested the effectiveness of our new vetting tools by conducting a search of known *Kepler* multi-planet systems for new TCEs and planet candidates. Known multi-transiting-planet systems are fertile grounds for detecting new transiting planets. We know a priori both that these systems have edge-on inclinations and that the planets in these systems very likely have small mutual inclinations, as is required for the known planet candidates to transit. It is likely that in many of these systems there are undiscovered planets that have not yet been detected.

We chose to search a subset of the *Kepler* multi-planet candidate systems. We searched 455 two-candidate systems, 141 three-candidate systems, 50 four-candidate systems, 21 five-candidate systems, 2 six-candidate systems, and 1 seven-candidate system. We used light curves produced by the *Kepler* pipeline and processed by its Pre-search Data Conditioning (PDC) module. We performed the search using the transit search pipeline described by Vanderburg et al. (2016) for transit searches in K2 data.[13] In brief, our pipeline flattens the light curves by fitting a basis spline to the light curve with outlier rejection and uses a box least-squares (BLS; Kovács et al. 2002) periodogram search to detect transiting planets. Our pipeline searches for points in the BLS periodogram with a signal-to-noise ratio (S/N) as measured by the BLS (S/N_{BLS}) greater than a predetermined cutoff. If any part of the BLS periodogram has high enough S/N_{BLS} to warrant further scrutiny, the pipeline automatically tests for and rejects various types of false alarms before promoting detections to the status of TCEs. Upon detecting a TCE, the pipeline fits a transit model to the TCE, saves the best-fit parameters, and removes the region around the transit from the light curve before recalculating the BLS periodogram on the clipped light curve to attempt to detect additional TCEs around that same star.

We removed transits of previously known planets from the *Kepler* multi-planet system light curves and searched for new candidate signals. For this search, we set a S/N_{BLS} threshold for detecting and recording TCEs of $S/N_{BLS} = 5$. This S/N_{BLS} threshold is considerably more liberal than both thresholds typically used for transit searches in data from the *K2* mission (which we normally set at $S/N_{BLS} = 9$; Vanderburg et al. 2016) and the threshold[14] used by the *Kepler* pipeline of $\text{MES} = 7.1$ (Jenkins et al. 2002). Traditionally, higher S/N_{BLS} thresholds are used for two reasons: (a) to prevent the number of spurious TCEs (caused by systematics or otherwise) from overwhelming the number of real detected planets, and (b) to ensure that no TCEs produced by purely Gaussian fluctuations are detected (e.g., Jenkins et al. 2002). The main benefit to using a cutoff as low as $S/N_{BLS} = 5$ is that we greatly increase our sensitivity to TCEs with S/N_{BLS} *above* more conservative cutoffs, because a variety of factors can cause S/N_{BLS} to be underestimated and TCEs to be erroneously rejected (see Section 7.2). Therefore, our search had the potential to discover new planets with S/N_{BLS} slightly above the traditional *Kepler* cutoff of 7.1.

The trade-off of choosing such a low S/N_{BLS} cutoff is that we greatly increase the number of spurious TCEs returned by our transit search. We added post-processing logic to automatically remove two common sources of spurious detections.

1. We discarded any detected TCEs with a transit duration less than 29.4 minutes (the spacing between consecutive *Kepler* data points). With a low S/N_{BLS} threshold, our pipeline tends to find a large number of "transits" consisting of only a couple of points, which would

[13] Because our pipeline was originally developed to search for planets in data from the *K2* mission, it is optimized for detecting the shorter-period ($\lesssim$40 days) planets commonly found by *K2*, so most of our new detections were at short orbital periods.

[14] Our calculation of S/N_{BLS} is roughly equivalent to the MES calculated by the *Kepler* pipeline.

Table 4
Summary of New Highly Ranked TCEs

KIC ID[a]	KOI ID[b]	*Kepler* Name	Period (days)	S/N	Prediction
11442793	351	Kepler-90	14.4	8.7	0.942
8480285	691	Kepler-647	10.9	8.4	0.941
11568987	354	Kepler-534	27.1	9.8	0.920
4852528	500	Kepler-80	14.6	8.6	0.916
5972334	191	Kepler-487	6.02	10.2	0.896
10337517	1165	Kepler-783	11.1	9.6	0.860
11030475	2248	⋯	4.75	8.7	0.858
4548011	4288	Kepler-1581	14.0	9.7	0.852
3832474	806	Kepler-30	29.5	14.1	0.847

Notes. Summary of new *Kepler* TCEs predicted to be planets by our model with probability greater than 0.8. See Table 6 for full details.
[a] *Kepler* Input Catalog identification number.
[b] *Kepler* Object of Interest identification number.

Unless they're USPs

translate to extremely short transit durations that are not physically realistic.

2. We discarded any TCEs with fewer than three complete transits. Often, one or two spurious events (like instrumental noise or cosmic-ray hits) lined up with one or two gaps in the light curve, causing the BLS algorithm to detect a long-period TCE. We required TCEs to have at least three transits containing at least half the number of expected in-transit points. Even if some of these discarded TCEs were caused by real planets, we would not have been able to confidently validate them with fewer than three complete transits.

After discarding these types of false positives, our search of 670 stars returned 513 new TCEs[15]—nearly one TCE per star. While the number of TCEs produced by this search is still manageable because we only searched a relatively small number of stars, such a high TCE rate would make it impractical for manual vetting of a similar search of the entire *Kepler* data set. Therefore, we used our neural network model to sort the TCEs by the probability of being planets. Out of the 513 TCEs produced by our search, our model only predicted that 30 were planets with a score greater than 0.5. These 30 highly ranked TCEs are shown in Table 6. We also inspected each TCE by eye to confirm that the ranking of TCEs by our model appeared reasonable. Empirically, our model did a good job of ranking the TCEs, with very few obvious errors.

6.2. *Summary of New Highly Ranked TCEs*

Of the 30 TCEs our model classified as “more-likely-than-not” planets, we give more serious consideration to the nine with planet predictions above 0.8. On our test set, the signals our model classified as planets with probability greater than 0.8 turned out to be planets or candidates 94% of the time. We calculated the S/Ns of these TCEs by measuring the mean depth of transit, dividing by the scatter of each individual point, and multiplying by the square root of the number of points in transit. All of the TCEs with scores above 0.8 have S/Ns above the traditional *Kepler* cutoff of 7.1. We summarize these these TCEs with predictions greater than 0.8 in Table 4 and give a brief description of each below.

1. *KIC 11442793*. This system hosts seven transiting planets, ranging from slightly larger than Earth to Jupiter-sized, all of which have been either validated or confirmed via transit-timing variations (TTVs; Cabrera et al. 2014; Lissauer et al. 2014; Schmitt et al. 2014). We detect an additional TCE with a period of 14.4 days with an S/N of 8.7.
2. *KIC 8480285*. This system hosts two planets: one previously validated Earth-sized planet (Morton et al. 2016) and one candidate super-Earth in periods of 16.2 and 29.7 days, respectively. We detect an additional TCE at S/N = 8.4 with a period of 10.9 days.
3. *KIC 11568987*. This system hosts one validated super-Earth (Morton et al. 2016) and one candidate Earth-sized planet in 16 and 7.3 day periods, respectively. We detect a new TCE with a period of 27.1 days, S/N = 9.8, and a depth corresponding to a roughly Earth-sized planet.
4. *KIC 4852528*. This system hosts five confirmed small planets, the outer four of which are in a rare dynamical configuration with the middle three planets and the outer three planets each locked in a three-body resonance (Lissauer et al. 2014; MacDonald et al. 2016). We find a new TCE with a period of 14.64 days at S/N = 8.6 exterior to all five previously known planets.
5. *KIC 5972334*. This system has two validated planets (Morton et al. 2016) and two candidates, including a warm Jupiter, an ultra-short-period planet (Sanchis-Ojeda et al. 2014), and a super-Earth in between, giving it a distinct architecture reminiscent of the 55 Cnc (Nelson et al. 2014) and WASP-47 (Becker et al. 2015) planetary systems. We detect a strong (S/N = 10.2) TCE with a period of 6.02 days that, if real, would have a radius a bit larger than that of the Earth. We see extra scatter in the light curve during the transit of this new TCE, however, which needs to be better understood before this TCE can be considered a good candidate.
6. *KIC 10337517*. This system hosts a sub-Earth-sized validated planet in a 4.3 day orbit (Morton et al. 2016) and a super-Earth-sized planet candidate in a 7 day orbit. We detect an additional TCE in an 11.1 day orbit, which also would likely be sub-Earth-sized.
7. *KIC 11030475*. This system has four planet candidates, none of which have been validated. We detect a new TCE at an orbital period of 4.75 days and S/N = 8.7, with the same time of transit, depth, and duration and exactly half the period of a previously detected candidate, KOI 2248.02. This “new” TCE, therefore, appears to be the true period of the candidate KOI 2248.02.
8. *KIC 4548011*. This system hosts one validated planet (Morton et al. 2016) in a 6.3 day period and one additional candidate in a 9.1 day period. We detect a new TCE with a period of 13.95 days with a relatively strong S/N = 9.7 that, if real, would likely be a sub-Earth-sized planet. Subsequently, this signal was also detected by the *Kepler* pipeline and included in the DR25 release of the *Kepler* planet candidate catalog as KOI 4288.04.
9. *KIC 3832474*. This system hosts two confirmed giant planets and a smaller confirmed super-Earth-sized planet

[15] On occasion, our pipeline detected residuals from planet candidates we attempted to remove, producing a TCE with the same orbital period and time of transit as the previously known candidates. We identified these cases by matching transit times and periods with known candidates and ignore them in the following paragraphs.

called Kepler-30 b, which orbits with a period of 29.16 days (Fabrycky et al. 2012). All of these planets show TTVs, and Kepler-30 d shows extremely large TTVs with an amplitude of more than a day (Fabrycky et al. 2012; Panichi et al. 2017). We detect a new TCE with a period of 29.45 days—very close to the mean period of Kepler-30 d. We suspect that this TCE is due to residual transits of Kepler-30 d that we were unable to completely remove from the light curve due to the large TTVs.

Four of these new TCEs were predicted to be planets by our model with a probability of 0.9 or greater. We consider this to be a very confident prediction; among those objects in our test set to which our model assigned a planet probability greater than 0.9, 96% turned out to be correctly classified as planets. Given our model's high confidence in these four new candidates, and the fact that they all have S/Ns greater than the traditional *Kepler* threshold of $\mathrm{S/N} = 7.1$, we subject these four new TCEs to additional scrutiny in order to validate them as bona fide exoplanets. We describe our additional tests and calculations to this end in the following subsections.

6.3. *Vetting New TCEs*

6.3.1. *Nearby Stars to the New TCE Hosts*

First, we attempted to identify whether any stars have been detected close on the sky to the host stars of our four new TCEs with planet predictions greater than 0.9. It is important to identify nearby stars and rule out the possibility that these companion stars are the sources of the new transit signals we see. To do this, we used information uploaded to the *Kepler* Community Follow-up Program (CFOP) website[16] for these four targets.

CFOP reports that there is a companion star about 7″ south of Kepler-90, detected in imaging from the United Kingdom Infra-Red Telescope (UKIRT). This star is bright enough to potentially be the source of the transit, and its nearby position overlaps Kepler-90's point-spread function, so it could, in principle, be the source of the new shallow transit signal we see. Similarly, UKIRT revealed a faint companion about 7″ southeast of Kepler-647 that could contribute the transit signal. Near Kepler-534, there are six stars within three *Kepler* pixels that could, in principle, contribute the transit signals we see, and Howard Isaacson reported on CFOP the detection of a faint secondary set of spectral lines in a high-resolution spectrum taken at Keck Observatory (these lines were identified using the method of Kolbl et al. 2015). Finally, Kraus et al. (2016) reported that Kepler-80 hosts two faint companions to the south —one at a distance of 1.″7 and one at a distance of 3.″9, both of which could, in principle, be the sources of the new transit signal we see.

Most of these detected companions—in particular, the ones detected visually in UKIRT or adaptive optics imaging—are at projected distances far enough from the *Kepler* target stars that we can either show that they are the sources of the transits we see or rule them out. The spectroscopically detected companion to Kepler-534, however, is apparently close enough to the target star that it (a) contributed light to the Keck HIRES spectrograph's slit and (b) was not detected with Robo-AO and Palomar Observatory adaptive optics imaging (Wang et al. 2015b; Furlan et al. 2017; Ziegler et al. 2017). Since we do not know this secondary star's angular separation, we cannot rule it out as the source of the new transiting signal we see, and we do not proceed to statistically validate this TCE.

6.3.2. *Image Centroid Analysis*

Each of the three remaining high-quality TCE hosts have nearby visually detected companions within their target apertures that might cause the transits. In this subsection, we use the lack of apparent motion of the positions of the target stars in the *Kepler* images to show that the transits must be cospatial with the target stars and cannot be contamination from these nearby visually detected companions.

We started by downloading the long-cadence target pixel time-series files for these three objects. For each of the roughly 60,000 images for each of the three stars, we measured the position of the target star by fitting a two-dimensional Gaussian to the image and recorded the best-fit center position in both the X and Y dimensions. This yielded a time series of X and Y positions for the entire four years of the *Kepler* mission. About every 90 days, *Kepler* rotated to keep its solar panels pointed toward the Sun, which meant that the stars fell on different cameras and the pixels were in different orientations.[17] We therefore transformed the X- and Y-pixel time series into right ascension (R.A.) and declination (decl.) time series. We high-pass-filtered the R.A. and decl. time series for each star, identified the points that occurred during the transits of the three TCE signals, and calculated the average change in R.A. and decl. for the target star during transit of the new TCEs.

For all three of our targets, we found no significant motion of the centroid during transit. For Kepler-80, we calculated a mean shift during transit in R.A. of 0.1 mas and a shift in decl. of 0.05 mas, each with uncertainties of 0.13 mas. If the transits, which have a depth of about 250 parts per million (ppm), were in fact coming from the closest visually detected companion at a distance of 1.″7, the expected centroid shift would be $1700\,\mathrm{mas}\ \times 2.5\times 10^{-4} = 0.42\,\mathrm{mas}$, smaller than what we measure. We conclude that the new TCE around Kepler-80 is cospatial with the target star (with an uncertainty of about 0.″6). Similarly, we find no shift in the centroid position of Kepler-90 during the transit of the new TCE with uncertainties of about 0.1 mas. Therefore, a roughly 90 ppm transit like our new TCE must come from a cospatial source with uncertainties of about 1″, which excludes the closest detected companion star (7″ distant) as the potential source of the transits. Finally, we confirm that the source of the new TCE around Kepler-647 must be cospatial with the target star with an uncertainty of roughly 1.″5—much closer than the closest detected companion at a distance of 7″.

We therefore conclude that none of the visually identified companions to the three stars hosting new high-quality TCEs are the sources of the new transit signals we see.

6.3.3. *Photometric Tests*

In this subsection, we describe the tests and checks we performed on the *Kepler* light curves of the TCE host stars to make sure that the TCE signals are not due to instrumental artifacts or other false-positive scenarios. In particular, we performed the following checks and tests.

[16] https://exofop.ipac.caltech.edu/cfop.php

[17] These 90 day periods are called *quarters*.

1. *Secondary eclipse search.* We inspected the phase-folded *Kepler* light curves for hints of a secondary eclipse. Detecting a secondary event would indicate that the TCE is likely due to the decrease in light caused by a faint, cospatial star that happens to be an eclipsing binary. We see no evidence for a significant secondary eclipse for any of our targets.
2. *Different apertures.* We produced light curves from the pixels outside of the optimal *Kepler* aperture for each of our targets and inspected these light curves folded on the period of the TCE signals. If the transits are present in the background pixels with a depth considerably greater than what we see in the on-target pixels, the signals are likely being introduced to the targets through scattered background light. In all three cases, we found that the background pixels showed no significant signals, so the TCEs are consistent with having their source on or near the target star.
3. *Even-odd transits.* We checked to make sure that the depths, timings, and shapes of even- and odd-numbered transits were consistent. Detecting a significant difference between even- and odd-numbered transits would indicate that the signal is caused by a fainter eclipsing binary star with twice the period that we detect and slightly different primary and secondary eclipses. We see no evidence for even/odd differences in the three targets we inspected.
4. *Consistency between quarters.* We checked that the transit signals did not change significantly depending on which CCD module the target stars were observed on. Every quarter during its original mission, *Kepler* rolled to keep its solar panels facing the Sun, so every roughly 90 days, the stars fell on different positions on the CCD array. Sometimes, contaminating signals could be introduced into the light curve more strongly in some positions on the CCD than others, causing differences in the strengths and depths of transit signals on each CCD. We saw no evidence for these quarterly transit depth differences.
5. *Consistency over mission lifetime.* We checked that the transit signals did not change significantly between the first and second halves of the *Kepler* mission. Sometimes, contamination can be caused by a process called *charge transfer inefficiency*, or *column anomaly* (Coughlin et al. 2014). In this scenario, a variable star (likely an eclipsing binary) that happens to be on the same CCD column as the target star can bleed signal during readout of the images. This effect worsens over time (Coughlin et al. 2016), so often, the transit depths at the beginning of the mission are considerably shallower than the transit depths at the end of the mission (see, for example, Figures 5 and 6 of Gaidos et al. 2016). We see no significant differences in transit depth in the light curves of the new TCEs around Kepler-80 and Kepler-90, making it unlikely that these two TCE signals are caused by column anomaly. However, we do see a possible increase in transit depth for the new TCE around Kepler-647. We fit the increase in transit depth as a linear function of time and determined a slope of about 13 ± 5 ppm per year—inconsistent with a constant transit depth with about 2.6σ confidence. While the detected slope is weak and could plausibly be due to red noise in the light curve or underestimated uncertainties in the transit depths, out of an abundance of caution, we chose not to validate this TCE.
6. *Ephemeris matching.* We searched for ephemeris matches, or other targets with nearly identical periods and times of transit either observed by *Kepler* or falling near the *Kepler* field in the sky. Finding an ephemeris match is an indication that a signal is likely caused by contamination from either the matched source or another unknown astrophysical source. We searched through the list of periods and transit times collected by Coughlin et al. (2014). We found no other plausible sources for the TCE signals in question (including the TCE around Kepler-647, which we suspect might be due to column anomaly).
7. *Simple aperture photometry.* We confirmed that the transit signals were still present in the raw simple aperture photometry (SAP) time series and that their characteristics did not substantially change between the SAP photometry and the photometry corrected for systematics via PDCMAP. This shows that the transit signals are likely not residuals of larger systematic errors in the *Kepler* light curves that were partially corrected by PDCMAP.

In the end, we find that the two new TCE signals around Kepler-80 and Kepler-90 appear to be astrophysical and not caused by *Kepler* systematics, and we find no evidence that these TCEs are associated with other known astrophysical signals from elsewhere in the *Kepler* field of view. We therefore consider these two signals to be planet candidates. While the TCE around Kepler-647 passes most tests, we find that the transit depth appears to increase over the course of the mission—a possible sign that this signal is an astrophysical false positive caused by column anomaly. While this candidate could still be a planet, we are not confident enough in this interpretation to validate it with high confidence.

6.4. False-positive Probability Calculations

After we vetted the two candidate signals around Kepler-80 and Kepler-90 to rule out various instrumental and astrophysical false positive scenarios, we calculated the probability that these two signals are caused by other astrophysical false positive scenarios that are indistinguishable from the signals we see, given the data at our disposal. We performed this calculation using `vespa` (Morton 2015), an open-source code library that implements the methodology described by Morton (2012) and Morton et al. (2016). In brief, `vespa` considers the possibility that the signal is caused by a transiting planet, as well as several different astrophysical false positive scenarios, including the possibility that the signal is caused by a binary star eclipsing the target or by some binary star around another, fainter star nearby on the sky to the target. After measuring basic transit parameters and fitting the host star's spectral energy distribution, `vespa` calculates how well these parameters match the various scenarios it considers and calculates the relative likelihood of each scenario using priors based on observational constraints and the characteristics of the host star.

We ran `vespa` on the new candidates around Kepler-80 and Kepler-90. We included as constraints the contrast curves from speckle-imaging observations uploaded to CFOP by Mark Everett (two observations taken at 692 and 880 nm at the Gemini-N Observatory of Kepler-90 and one observation taken at 880 nm at the WIYN telescope of Kepler-80). We imposed constraints on the depths of putative secondary eclipses by calculating the 3σ limit based on the scatter in the phase-folded light curve on each candidate's period binned to the transit duration. We used our constraints from the image centroid analysis to specify the 3σ maximum radius at which a contaminating star could contribute the transit signal. We used the spectroscopic parameters from MacDonald et al. (2016) and Petigura et al. (2017) as inputs for Kepler-80 and Kepler-90, respectively, and we included near-infrared photometry from the 2MASS survey (Skrutskie et al. 2006). Recently, it has been shown that systematic photometric uncertainties and/or offsets between photometric data sets can cause `vespa` to misclassify planet candidates (Shporer et al. 2017; Mayo et al. 2017, in preparation), so we avoided this issue by only including the high-quality 2MASS photometry.

Based on these inputs, `vespa` calculated that the new candidate around Kepler-80 has a false-positive probability of about 0.002, and the new candidate around Kepler-90 has a false-positive probability of about 0.005. Both of these probabilities are quite low, below the 1% threshold often used to validate planets in *Kepler* (e.g., Morton et al. 2016), thanks to the well-characterized stars, high-resolution imaging, and tight constraints on the source of the transits from our centroid analysis. The `vespa` analysis assumes that the candidates being validated are the only candidates in their systems, but these two candidates are found in multi-planet systems. Candidates in multi-planet systems are a priori significantly less likely to be false positives than candidates in single-planet systems (Latham et al. 2011; Lissauer et al. 2012; Rowe et al. 2014). Lissauer et al. (2012) estimated that candidates in systems with at least three planets are about 50 times more likely to be real planets than the average planet. When we factor in this multiplicity "boost," the false-positive probabilities of our two candidates decrease to less than 10^{-4}. We therefore consider these two new candidates to be validated as bona fide exoplanets and designate them Kepler-80 g and Kepler-90 i.

6.5. Transit Fits

After statistically validating the two newly discovered planets around Kepler-80 and Kepler-90, we fit the full *Kepler* long-cadence light curves with Mandel & Agol (2002) models to determine planet and orbital parameters and uncertainties. We explored parameter space using a Markov chain Monte Carlo (MCMC) algorithm with affine invariant ensemble sampling (Goodman & Weare 2010). We fit the new candidates' orbital periods, times of transit, orbital inclinations, scaled semimajor axes ($a/R_\star$), radius ratios ($R_p/R_\star$), and quadratic limb-darkening coefficients (following the parameterization of Kipping 2013). We imposed Gaussian priors on the two limb-darkening coefficients centered at values predicted by Claret & Bloemen (2011) for the host stars (using spectroscopic parameters from MacDonald et al. 2016 for Kepler-80 and Petigura et al. 2017 for Kepler-90). We set the widths of these Gaussian priors to be 0.07 in both the linear and quadratic coefficients, following Müller et al. (2013).

For each planet fit, we initialized 100 walkers and evolved them for 20,000 links, discarding the first 10,000 for burn-in. We tested convergence of the MCMC chains by calculating the Gelman–Rubin potential reduction factor (Gelman & Rubin 1992), which we found to be below 1.2 for all of our parameters (Brooks & Gelman 1998). We show the results of our transit fits in Table 5, and we plot the transit light curves of these two newly discovered planets in Figure 12.

7. Discussion

7.1. Newly Validated Planets

We identified two new exoplanets by conducting a transit search and using our new machine learning planet candidates vetting pipeline. Both of these planets are in high-multiplicity systems (there are five and seven other planets in the Kepler-80 and Kepler-90 systems, respectively), which are interesting to study because of our fine understanding of their architectures. In this subsection, we comment on each of these two planets individually.

7.1.1. Kepler-80 g

Kepler-80 g is the outermost planet in its system. Previously, five planets were known, with orbital periods of 0.98, 3.07, 4.65, 7.05, and 9.52 days. The outer four[18] of these planets are in a rare dynamical configuration called a *resonant chain* (Lissauer et al. 2014; MacDonald et al. 2016). The orbital periods of each pair of adjacent planets are close to an integer ratio (in this system, either 2:3 or 3:4), and the orbital periods of consecutive trios of planets satisfy the Laplace relationship,

$$\frac{p}{P_1} - \frac{p+q}{P_2} + \frac{q}{P_3} \approx 0, \tag{4}$$

where p and q are integers and P_i are the orbital periods of the three planets. Kepler-80 d, e, and b satisfy this relationship with $(p,\, q) = (2,\, 3)$, and Kepler-80 e,b, and c satisfy this relationship with $(p,\, q) = (1,\, 2)$. Satisfying the Laplace relationship is equivalent to the fact the TTV "super-period" for adjacent planets in the chain is nearly the same for each pair of planets. In the Kepler-80 system, the super-period of each adjacent pair of planets is close to 192 days (MacDonald et al. 2016). The super-period is the time it takes for the orbital angle at which two adjacent planets line up in conjunction circulates about the orbit—that is, for planets in the Kepler-80 system, two adjacent planets which happen to transit simultaneously will repeat this every 192 days. The super-period, $P_{\rm super}$, for two planets close to a first order mean motion resonance is given by:

$$P_{\rm super} = \frac{1}{\left|\frac{j+1}{P_2} - \frac{j}{P_1}\right|} \tag{5}$$

where j is an integer and $_{Pi}$ are the orbital periods of the two planets. Resonant chains like the one around Kepler-80 are extremely finely tuned, specifying orbital periods to high precision. Recently, this fact was used by Luger et al. (2017) to predict and recover the orbital period of the planet TRAPPIST-1 h using data from the *K2* mission. Following Luger et al. (2017), we calculated the orbital periods of putative additional outer planets in the Kepler-80 system under the assumption that they happened to be in a three-body resonance with Kepler-80 b and c. After rearranging Equation (4) and plugging in values of p and q between 1 and 3 (inclusive), we identified possible orbital periods of 14.64671, 11.54078, 10.77887, 31.72031, 12.41860, and 20.04003 days. For $p = q = 1$, the period predicted by the Laplace relation is within 100 s of the measured orbital period (14.64558 ± 0.00012 days) of the newly discovered planet—agreement to within one part in

[18] The shortest-period planet in the system, Kepler-80 f, is dynamically decoupled from its resonant counterparts.

Table 5
Parameters for Kepler-80 g and Kepler-90 i

Parameter	Kepler-80 g	Kepler-90 i
Stellar Parameters		
R.A.	19:44:27.02	18:57:44.04
Decl.	39:58:43.6	49:18:18.58
$M_\star$ [$M_\odot$]	0.83 ± 0.03	1.12 ± 0.046
$R_\star$ [$R_\odot$]	0.678 ± 0.023	1.29 ± 0.167
Limb darkening q_1	0.55 ± 0.14	0.42 ± 0.13
Limb darkening q_2	0.447 ± 0.045	0.278 ± 0.040
$\log g_\star$ [cgs]	4.639 ± .012	4.27 ± 0.1
[M/H]	0.04 ± 0.08	0.098 ± 0.04
$T_{\rm eff}$ [K]	4540 ± 100	6015 ± 60
New Planet Parameters		
Orbital period, P [days]	14.64558 ± 0.00012	14.44912 ± 0.00020
Radius ratio, $(R_P/R_\star)$	0.0153 ± 0.0019	0.00935 ± 0.00089
Scaled semimajor axis, $a/R_\star$	$45\ ^{+9.9}_{-15}$	$33.8\ ^{+6.6}_{-12}$
Orbital inclination, i [deg]	$89.35\ ^{+0.47}_{-0.98}$	$89.2\ ^{+0.59}_{-1.3}$
Transit impact parameter, b	0.52 ± 0.28	0.50 ± 0.28
Time of transit, t_t [BJD$_{\rm TDB}$]	2455658.6073 ± 0.0037	2455644.3488 ± 0.0048
Transit duration, t_{14} [hr]	2.12 ± 0.28	2.80 ± 0.31
Equilibrium temperature ,T_{eq} [K]	418 ± 36	709 ± 75
R_P [$R_\oplus$]	1.13 ± 0.14	1.32 ± 0.21

Note. Spectroscopic and stellar parameters for Kepler-80 come from MacDonald et al. (2016), while the parameters for Kepler-90 come from Petigura et al. (2017) and Johnson et al. (2017). Equilibrium temperatures are calculated assuming perfect heat redistribution and albedos uniformly distributed between 0 and 0.7.

10,000. Kepler-80 g orbits just outside a 2:3 period ratio with the 9.52 day planet, and these two planets have a super-period of 192 days, matching the super-periods of other planet pairs in the system. Kepler-80 g therefore almost certainly is part of the chain of three-body resonances around Kepler-80, bringing the total number of planets in the chain to five and the total number of planets in the system to six. The fact that we can so accurately predict the orbital period of Kepler-80 g through resonance arguments is strong independent confirmation that the signal is indeed caused by an exoplanet and a validation of our methods from end-to-end.

Kepler-80 g warrants further study for several reasons. First, although the orbital period we found makes it extremely likely that this newly discovered planet is the outer link in a chain of three-body resonances, the system requires further dynamical analysis to confirm the resonant state. Even though its low S/N will make it difficult, it could be worthwhile to search for Kepler-80 g's TTVs, perhaps by measuring the timing of many transits averaged together (e.g., Mills et al. 2016). Finally, it will be important to assess the impact of a new planet in the resonant chain on the Kepler-80 TTV mass measurements made by MacDonald et al. (2016). In principle, if some of the TTV signal measured in the 9.52 day Kepler-80 c was caused by Kepler-80 g, that would impact the measured mass of Kepler-80 b, and so on.

7.1.2. Kepler-90 i

Kepler-90 is a star that was previously known to host seven transiting planets. Planetary systems with this many known planets are very rare. According to the NASA Exoplanet Archive (accessed 2017 August 19), only Kepler-90 and TRAPPIST-1 (Gillon et al. 2017) are known to host this many planets.[19] Kepler-90 was initially detected by the *Kepler* pipeline and designated KOI 351 by Borucki et al. (2011b), who detected three large, long-period transiting planet candidates, which eventually would be known as Kepler-90 e, g, and h. As more data became available, additional searches (Burke et al. 2014; Cabrera et al. 2014; Lissauer et al. 2014; Schmitt et al. 2014) identified additional shorter-period and smaller transiting planet candidates, bringing the total number of planets to seven. Kepler-90 hosts two planets about 1.5 times larger than Earth in 7 and 8.7 day orbits; three planets about three times the size of Earth in 59.7, 91.9, and 124.9 day orbits; and two giant planets in 210.6 and 331.6 day orbits.

The newly discovered planet, Kepler-90 i, falls in the largest gap in orbital-period space in the Kepler-90 system between Kepler-90 c (8.7 days) and Kepler-90 d (59.7 days). With a period of 14.4 days and a radius of 1.3 $R_\oplus$, Kepler-90 i fits in well with Kepler-90 b and c, the two shortest-period planets in the system, which have radii comparable to Kepler-90 i of 1.4 and 1.6 $R_\oplus$, respectively. Kepler-90 b, c, and i all have radii small enough that they are either likely to be or might be rocky (Rogers 2015). It is striking that the planets around Kepler-90 are so well sorted by radius and orbital period, with small planets orbiting close-in and giant planets orbiting farther from the star, although it is possible that this is a detection bias. Kepler-90 i is just barely above *Kepler*'s detection limits, so we would be unlikely to detect any similarly small planets orbiting farther from Kepler-90 than this.

Another particularly striking feature of the Kepler-90 system is how orderly it appears to be. The transit durations $t_{1,4}$ of the seven previously known Kepler-90 planets scale quite well with period P (where $t_{1,4} \propto P^{1/3}$). Kepler-90 i appears to somewhat buck that trend. While the mean stellar density derived from Kepler-90 i's transit is still consistent with that of the other planets and Kepler-90's spectroscopic density, Kepler-90 i's transit duration of about 2.8 hr is shorter than we expect (about 5 hr) assuming circular and coplanar orbits. Therefore, Kepler-90 i is likely slightly inclined away from our line of sight. If the orbit of Kepler-90 i is perfectly

[19] HD 10180 (Lovis et al. 2011) shows six confidently detected radial velocity signals and up to three more speculative signals (Lovis et al. 2011; Tuomi 2012) that have not been found necessary in subsequent analyses (Kane & Gelino 2014).

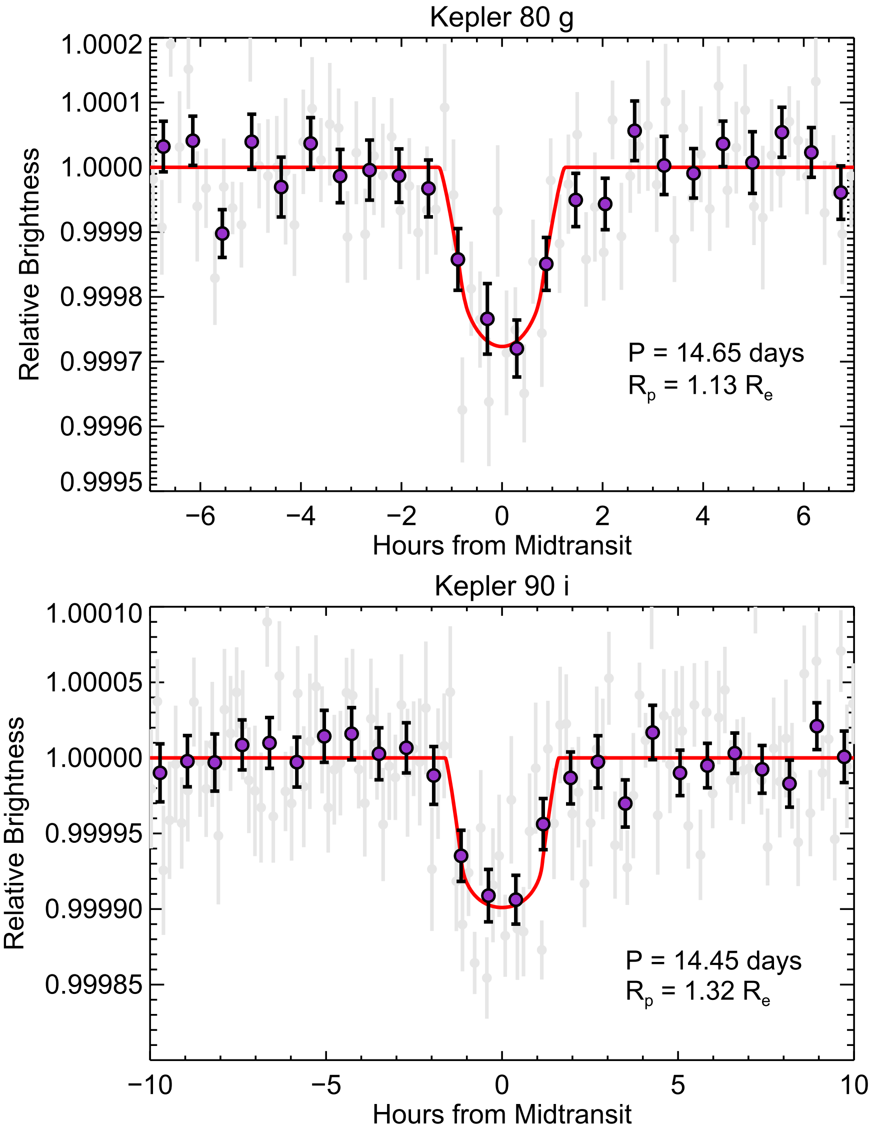

Figure 12. Transit light curves and best-fit models for the newly discovered planets around Kepler-80 and Kepler-90. In these plots, the gray points are robust averages of bins with a width of approximately 10 minutes. The purple points are robust averages of bins with a size of about one-fourth the transit duration of the planet (bins of about 30 minutes for Kepler-80 g and about 45 minutes for Kepler-90 i).

circular, its impact parameter would have to be about 0.85; this gives it an orbital inclination of about 88°, compared to the other seven planets in the system, which all likely have orbital inclinations of 89° or higher. There are reasons to expect that the Kepler-90 system should host some planets with slightly higher inclinations than are determined for the seven previously known planets—Becker & Adams (2016) showed that the Kepler-90 system was among the *Kepler* multi-planet systems where self-excitation of mutual inclinations was most likely to occur. The addition of an eighth planet would make it even more likely that planets might be excited to impact parameters of ≈0.85, as we see for Kepler-90 i.

The addition of an eighth planet makes Kepler-90 a record-setter as the first exoplanetary system known to have eight planets and, along with the Sun, one of only two stars known to host this many planets. Finding systems with this many planets and learning about their sizes and orbits is a unique window into their formation. As noted by Cabrera et al. (2014) and Schmitt et al. (2014), Kepler-90 has an architecture reminiscent of that of the solar system, with smaller planets orbiting close to the host star and larger planets orbiting at larger distances. Unlike the solar system, however, the entire Kepler-90 system is packed within about the orbital radius of Earth. This compact architecture suggests that the Kepler-90 planets may have formed like the solar system planets, with the more distant planets forming beyond the snow line, where heavy volatile elements like water are in solid phase, and with the smaller planets forming closer to the star, where heavy volatile elements are in gaseous phase, limiting their growth. Then, unlike the solar system, Kepler-90 may have undergone disk migration to bring it into its current flat, compact, and orderly configuration. If this is true, then there might be more undetected transiting planets orbiting Kepler-90 that have radii too small or orbital periods too long to detect without more *Kepler*-like observations.

7.2. Looking Ahead

The goal of the *Kepler* mission was to study Earth-sized planets orbiting Sun-like stars. However, even with four years of continuous data, these planets are difficult to detect because (a) their smaller radii and the relative dimness of their stars means that the dips caused by their transits are likely to be relatively shallow compared to the scatter of their light curves, and (b) they are likely to have long orbital periods, so we can only detect at most a handful of transits in a four years window. These effects imply that Earth-like planets are more likely to have signal-to-noise ratios and multiple event statistics near or below the detection limit of the *Kepler* pipeline. The probability of the *Kepler* pipeline missing these planets is high—it recovers only 26% of injected transits[20] with expected MESs between 7.1 and 8, despite these signals being strong enough (in principle) to rule out being fluctuations of pure random noise. There are many reasons that real transiting planets above the MES threshold of 7.1 can be rejected: residual non-Gaussian noise (like systematics and cosmic-ray hits) can artificially reduce the apparent MES, light-curve detrending can make transits appear shallower, and quantization in the grid search can cause slight mismatches in the detected period, epoch, and transit duration (Christiansen 2017).

We can recover some of these lost planets by reducing the signal to noise threshold of the detection pipeline (e.g., *Kepler* pipeline or BLS algorithm) to allow less-significant signals to become TCEs. As we showed in Section 6, this increases the recovery rate of planets at signal to noise above more conservative thresholds, but at the cost of detecting a large number of false-positive TCEs that cannot practically be examined by eye. Therefore, a thorough study of the low signal to noise regime of *Kepler* data would require an automatic vetting system that can accurately separate plausible planet candidates from false positives.

Our neural network model for automatic vetting can quite accurately distinguish real planet signals from false positives caused by astrophysical and instrumental phenomena. It ranks planet candidates higher than false positives in our test set 98.8% of the time. When testing our model on simulated data, we found that while our model recovers simulated true planets at a good rate, our current implementation does not yet match the performance of the more mature Robovetter system at identifying certain classes of simulated false positives, such as weak secondary eclipses and spurious detections. Nonetheless, when we applied our model to new *Kepler* planet candidates, we found that it does a good job of ranking these TCEs by the probability of being a planet with very few obvious errors. We discovered several real new planets at the top of our model's rankings, which validates both the effectiveness of our model and our motivation for exploring the lower S/N region of *Kepler* data. The immediate impact of this work is the discovery of several new *Kepler* planets and planet candidates, including the exciting discovery of an eighth planet in the Kepler-90 system.

At a higher level, a technique like ours could be used in the future to make more accurate estimates of planetary occurrence rates. In particular, the occurrence rate of Earth-like planets, colloquially called "η-Earth," is one of the most important and exciting open questions in exoplanet research—it is directly proportional to the estimated fraction of planets that might harbor life as we know it on Earth (Drake 1965). A good measurement of η-Earth is crucial for designing the next generation of large space telescopes like *LUVOIR* and *HabEx*, which hope to directly image temperate, terrestrial worlds. To this end, we have identified the following ways that we can improve our model in the future to increase its accuracy and reduce known types of false positives.

1. *Improved training set.* Our current training set contains only about 15,000 labeled examples, all of which are TCEs detected by the *Kepler* pipeline with MESs greater than 7.1. Only 8% of planets and planet candidates in our training set have MESs less than 10, and as a result, we see a significant drop-off in our model's performance on real planets and planet candidates at MES $\lesssim$ 10 (see Figure 10). Incorporating simulated or unlabeled data into our training set would significantly increase its size and would likely improve the performance of our model on low-MES planet candidates, eclipsing binaries, and spurious false positives (see Section 5.4).
2. *Improved input representations.* Our current method for flattening light curves occasionally creates bogus transits in stars with high-frequency stellar variability (see Section 5.4). Improving our flattening routine could cut down on the number of signals due to stellar variability that are classified as likely planets.
3. *Centroid information.* Incorporating some form of centroid information into our input representation may improve our model's ability to classify transits that occur on a background star instead of the target star. This could either be a phase-folded time series of R.A./decl. (and some information about nearby stars and the expected centroid motion) or an in-transit difference image.
4. *Segmented local views.* Splitting the local views into different segments would enable our model to examine the consistency of transits between different slices of the data set. For example, "even-numbered transits" and "odd-numbered transits," "first half of the data set" and "second half of the data set," or "CCD module 1," "CCD module 2," "CCD module 3," and "CCD module 4."
5. *Secondary and tertiary local views.* This would give our model a "close-up" view of any putative secondary eclipses.
6. *Auxiliary scalar features.* We experimented with TCE features, such as period, duration, depth, and impact, but we did not see any improvement on our validation set. We would like to try using stellar host features in the future, such as radius and density (to compare against the stellar density inferred from the transits).
7. *Robust (outlier-resistant) means.* The uncertainty on a sample median is $\sqrt{\pi/2} \approx 1.253$ times larger than the uncertainty on a sample mean, so in principle, by using robust means instead of medians in our input representation, we could boost the S/N the model sees by up to 25%.

Huber est - gist.github.com/ramhiser/9d0ea69e3b64ed4eabb1

8. Conclusion

We presented a method for automatically classifying *Kepler* transiting planet candidates using deep learning. Our neural network model is able to accurately distinguish the subtle differences between transiting exoplanets and false positives like eclipsing binaries, instrumental artifacts, and stellar variability. On our test set, our model ranks true planet candidates above false positives 98.8% of the time. On simulated data, while it recovers simulated planets at a good rate, our model is not yet as effective as the more mature Robovetter system at rejecting certain types of simulated false positives, such as weak secondary eclipses. We are planning on improving our model in the future to address the failure modes we observed in the simulated data, with the ultimate goal of using our model to calculate the occurrence rate of Earth-like planets in the galaxy.

We tested the ability of our model to identify new planet candidates by conducting a search of known *Kepler* multi-planet

[20] Calculated as described in Section 4 of Christiansen (2017). Also see Figure 7 in Christiansen et al. (2016).

systems. We conducted our search at a lower signal to noise threshold than traditionally used, allowing a high rate of spurious false positives in order to discover new candidates that were not recovered by previous searches. Our detection rate of almost one event per star would have produced too many candidates to examine manually in a full search of the ∼200,000 *Kepler* target stars. We used our model to rank the new candidates by likelihood of being a real planet and discovered several plausible planet candidates at signal to noise above the traditional threshold of 7.1. We statistically validated two of these new candidates: Kepler-80 g, which is part of an interesting five-planet resonant chain, and Kepler-90 i, which is now the eighth confirmed planet in the Kepler-90 system. This paper brings Kepler-90 into a tie with our Sun as the star known to host the most planets.

We thank Juliette Becker, Jessie Christiansen, Steve Bryson, Susan Thompson, Geert Barentsen, George Dahl, Samy Bengio, and Martti Kristiansen for helpful conversations. We thank Susan Thompson for sharing her manuscript in advance of its submission, and we thank Jeff Coughlin for his expertise on column anomaly false positives in *Kepler* data. We thank the anonymous referee for a thoughtful report and helpful suggestions. AV acknowledges partial support from the NSF Graduate Research Fellowship, grant No. DGE 1144152, and from NASA's TESS mission under a subaward from the Massachusetts Institute of Technology to the Smithsonian Astrophysical Observatory, Sponsor Contract Number 5710003554. This research has made use of NASA's Astrophysics Data System and the NASA Exoplanet Archive, operated by the California Institute of Technology, under contract with NASA under the Exoplanet Exploration Program. Some of the data presented in this paper were obtained from the Mikulski Archive for Space Telescopes (MAST). STScI is operated by the Association of Universities for Research in Astronomy, Inc., under NASA contract NAS5–26555. Support for MAST for non-*HST* data is provided by the NASA Office of Space Science via grant NNX13AC07G and by other grants and contracts. This paper includes data collected by the *Kepler* mission. Funding for the *Kepler* mission is provided by the NASA Science Mission directorate. This work was performed in part under contract with the California Institute of Technology (Caltech)/Jet Propulsion Laboratory (JPL), funded by NASA through the Sagan Fellowship Program executed by the NASA Exoplanet Science Institute.

Facility: Kepler.

Appendix

Our search of 670 known *Kepler* multi-planet systems returned 513 new TCEs. Table 6 shows the subset of these

Table 6
Ranked List of New *Kepler* TCEs

Kepler ID	Period (days)	T_0 (BJD$_{\rm TDB}$[a]-2454833)	Duration (hr)	Impact	$R_p/R^{\star}$	Predicted Planet Probability	Notes
11442793	14.44912434	2.200	2.704	0.777	0.0085	0.942	[b]
8480285	10.91423130	1.507	4.809	0.746	0.0056	0.941	…
11568987	27.13809776	17.397	3.007	0.550	0.0091	0.920	…
4852528	14.64553642	5.453	1.636	0.592	0.0145	0.916	[c]
5972334	6.02215719	4.013	1.859	0.877	0.0145	0.896	…
10337517	11.14567280	8.359	3.193	0.612	0.0077	0.860	…
11030475	4.74503899	3.396	2.470	0.606	0.0122	0.858	…
4548011	13.95128345	4.758	4.092	0.661	0.0051	0.852	…
3832474	29.45780563	0.826	11.117	0.661	0.0153	0.847	…
11669239	7.55533743	6.545	2.214	0.527	0.0081	0.788	…
10130039	96.10472107	11.038	5.548	0.484	0.0075	0.764	…
5351250	261.91778564	247.108	7.775	0.009	0.0202	0.752	…
5301750	5.18081331	1.166	1.423	0.901	0.0088	0.751	…
8282651	22.22732544	6.448	2.596	0.515	0.0138	0.731	…
6786037	21.14108849	20.541	5.479	0.964	0.0155	0.712	…
6690082	9.82555485	8.214	2.032	0.632	0.0084	0.711	…
9896018	5.55300617	0.074	3.037	0.667	0.0087	0.707	…
11450414	7.87495708	2.770	1.533	0.709	0.0085	0.655	…
6021275	5.92093372	5.056	1.791	0.505	0.0043	0.618	…
4851530	206.21182251	62.663	5.313	0.375	0.0467	0.613	…
8804283	7.70983267	2.427	3.593	0.645	0.0068	0.604	…
3323887	39.06724930	15.271	5.383	0.923	0.0583	0.595	…
6508221	4.11963272	3.461	1.542	0.441	0.0063	0.595	…
9006186	51.94103241	34.969	3.557	0.854	0.0112	0.585	…
12061969	14.10193443	13.709	2.306	0.047	0.0192	0.579	…
11074178	4.88644457	1.973	2.691	0.752	0.0095	0.560	…
8397947	5.14999104	0.281	4.094	0.629	0.0107	0.516	…
11968463	3.37368155	1.139	2.512	0.850	0.0090	0.509	…
10600261	7.49254370	6.303	1.909	0.771	0.0089	0.507	…
3323887	38.83024216	20.515	9.247	1.022	0.1017	0.507	…

Notes.
[a] T_0 is measured in Barycentric Julian Date (BJD) minus a constant offset of 2,454,833.0 days. The offset corresponds to 12:00 on 2009 January 1 UTC. The time given is the center of the first transit occurring after the reference date.
[b] Kepler-90 i.
[c] Kepler-80 g.

TCEs that our model predicted were planets with a probability greater than 0.5.

References

Abadi, M., Agarwal, A., Barham, P., et al. 2016, arXiv:1603.04467

Armstrong, D. J., Pollacco, D., & Santerne, A. 2017, MNRAS, 465, 2634

Barclay, T., Rowe, J. F., Lissauer, J. J., et al. 2013, Natur, 494, 452

Batalha, N. M., Borucki, W. J., Bryson, S. T., et al. 2011, ApJ, 729, 27

Batalha, N. M., Rowe, J. F., Bryson, S. T., et al. 2013, ApJS, 204, 24

Becker, J. C., & Adams, F. C. 2016, MNRAS, 455, 2980

Becker, J. C., Vanderburg, A., Adams, F. C., Rappaport, S. A., & Schwengeler, H. M. 2015, ApJL, 812, L18

Bishop, C. M. 2006, Pattern Recognition and Machine Learning (Berlin: Springer)

Borucki, W. J., Koch, D. G., Basri, G., et al. 2011a, ApJ, 728, 117

Borucki, W. J., Koch, D. G., Basri, G., et al. 2011b, ApJ, 736, 19

Brooks, S. P., & Gelman, A. 1998, Journal of Computational and Graphical Statistics, 7, 434

Burke, C. J., Bryson, S. T., Mullally, F., et al. 2014, ApJS, 210, 19

Burke, C. J., Christiansen, J. L., Mullally, F., et al. 2015, ApJ, 809, 8

Cabrera, J., Csizmadia, S., Lehmann, H., et al. 2014, ApJ, 781, 18

Cabrera-Vives, G., Reyes, I., Förster, F., Estévez, P. A., & Maureira, J.-C. 2017, ApJ, 836, 97

Catanzarite, J. H. 2015, Autovetter Planet Candidate Catalog for Q1-Q17 Data Release 24 (KSCI-19090-001), Tech. Rep.

Christiansen, J. L. 2017, Planet Detection Metrics: Pixel-Level Transit Injection Tests of Pipeline Detection Efficiency for Data Release 25 (KSCI-19110-001), Tech. Rep.

Christiansen, J. L., Clarke, B. D., Burke, C. J., et al. 2016, ApJ, 828, 99

Christiansen, J. L., Jenkins, J. M., Caldwell, D. A., et al. 2012, PASP, 124, 1279

Cireşan, D., Meier, U., & Schmidhuber, J. 2012, arXiv:1202.2745

Claret, A., & Bloemen, S. 2011, A&A, 529, A75

Coughlin, J. L. 2017, Planet Detection Metrics: Robovetter Completeness and Effectiveness for Data Release 25 (KSCI-19114-001), Tech. Rep.

Coughlin, J. L., Mullally, F., Thompson, S. E., et al. 2016, ApJS, 224, 12

Coughlin, J. L., Thompson, S. E., Bryson, S. T., et al. 2014, AJ, 147, 119

Dittmann, J. A., Irwin, J. M., Charbonneau, D., et al. 2017, Natur, 544, 333

Doyle, L. R., Carter, J. A., Fabrycky, D. C., et al. 2011, Sci, 333, 1602

Drake, F. 1965, Current Aspects of Exobiology, 323

Dressing, C. D., & Charbonneau, D. 2013, ApJ, 767, 95

Dressing, C. D., & Charbonneau, D. 2015, ApJ, 807, 45

Dressing, C. D., Charbonneau, D., Dumusque, X., et al. 2015, ApJ, 800, 135

Fabrycky, D. C., Ford, E. B., Steffen, J. H., et al. 2012, ApJ, 750, 114

Fischer, D. A., Schwamb, M. E., Schawinski, K., et al. 2012, MNRAS, 419, 2900

Foreman-Mackey, D., Hogg, D. W., & Morton, T. D. 2014, ApJ, 795, 64

Foreman-Mackey, D., Morton, T. D., Hogg, D. W., Agol, E., & Schölkopf, B. 2016, AJ, 152, 206

Fressin, F., Torres, G., Charbonneau, D., et al. 2013, ApJ, 766, 81

Furlan, E., Ciardi, D. R., Everett, M. E., et al. 2017, AJ, 153, 71

Gaidos, E., Mann, A. W., & Ansdell, M. 2016, ApJ, 817, 50

Gelman, A., & Rubin, D. B. 1992, StaSc, 457

Gillon, M., Triaud, A. H. M. J., Demory, B.-O., et al. 2017, Natur, 542, 456

Golovin, D., Solnik, B., Moitra, S., et al. 2017, in Proc. 23rd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (New York: ACM), 1487, doi:http://doi.acm.org/10.1145/3097983.3098043

Goodman, J., & Weare, J. 2010, Communications in Applied Mathematics and Computational Science, 5, 65

Howard, A. W., Marcy, G. W., Bryson, S. T., et al. 2012, ApJS, 201, 15

Jackson, B., Stark, C. C., Adams, E. R., Chambers, J., & Deming, D. 2013, ApJ, 779, 165

Jenkins, J. M., Caldwell, D. A., & Borucki, W. J. 2002, ApJ, 564, 495

Jenkins, J. M., Caldwell, D. A., Chandrasekaran, H., et al. 2010a, ApJL, 713, L120

Jenkins, J. M., Caldwell, D. A., Chandrasekaran, H., et al. 2010b, ApJL, 713, L87

Jenkins, J. M., Chandrasekaran, H., McCauliff, S. D., et al. 2010c, Proc. SPIE, 7740, 77400D

Johnson, J. A., Petigura, E. A., Fulton, B. J., et al. 2017, arXiv:1703.10402

Kane, S. R., & Gelino, D. M. 2014, ApJ, 792, 111

Kingma, D., & Ba, J. 2015, in Int. Conf. Learning Representations (ICLR 2015), arXiv:1412.6980

Kipping, D. M. 2013, MNRAS, 435, 2152

Kipping, D. M., Torres, G., Henze, C., et al. 2016, ApJ, 820, 112

Koch, D. G., Borucki, W. J., Basri, G., et al. 2010, ApJL, 713, L79

Kolbl, R., Marcy, G. W., Isaacson, H., & Howard, A. W. 2015, AJ, 149, 18

Kovács, G., Zucker, S., & Mazeh, T. 2002, A&A, 391, 369

Kraus, A. L., Ireland, M. J., Huber, D., Mann, A. W., & Dupuy, T. J. 2016, AJ, 152, 8

Krizhevsky, A., Sutskever, I., & Hinton, G. E. 2012, in Advances in Neural Information Processing Systems 25, ed. F. Pereira et al. (Newry: Curran Associates, Inc.), 1097, http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

Latham, D. W., Rowe, J. F., Quinn, S. N., et al. 2011, ApJL, 732, L24

Lecun, Y., Bengio, Y., & Hinton, G. 2015, Natur, 521, 436

Lissauer, J. J., Fabrycky, D. C., Ford, E. B., et al. 2011, Natur, 470, 53

Lissauer, J. J., Marcy, G. W., Bryson, S. T., et al. 2014, ApJ, 784, 44

Lissauer, J. J., Marcy, G. W., Rowe, J. F., et al. 2012, ApJ, 750, 112

Lovis, C., Ségransan, D., Mayor, M., et al. 2011, A&A, 528, A112

Luger, R., Sestovic, M., Kruse, E., et al. 2017, NatAs, 1, 0129

Maaten, L. v. d., & Hinton, G. 2008, Journal of Machine Learning Research, 9, 2579, http://www.jmlr.org/papers/v9/vandermaaten08a.html

MacDonald, M. G., Ragozzine, D., Fabrycky, D. C., et al. 2016, AJ, 152, 105

Mandel, K., & Agol, E. 2002, ApJL, 580, L171

Marcy, G. W., Isaacson, H., Howard, A. W., et al. 2014, ApJS, 210, 20

McCauliff, S. D., Jenkins, J. M., Catanzarite, J., et al. 2015, ApJ, 806, 6

Millholland, S., & Laughlin, G. 2017, AJ, 154, 83

Mills, S. M., Fabrycky, D. C., Migaszewski, C., et al. 2016, Natur, 533, 509

Mislis, D., Bachelet, E., Alsubai, K. A., Bramich, D. M., & Parley, N. 2016, MNRAS, 455, 626

Morton, T. D. 2012, ApJ, 761, 6

Morton, T. D. 2015, VESPA: False Positive Probabilities Calculator, Astrophysics Source Code Library, ascl:1503.011

Morton, T. D., Bryson, S. T., Coughlin, J. L., et al. 2016, ApJ, 822, 86

Muirhead, P. S., Johnson, J. A., Apps, K., et al. 2012, ApJ, 747, 144

Mulders, G. D., Pascucci, I., & Apai, D. 2015a, ApJ, 798, 112

Mulders, G. D., Pascucci, I., & Apai, D. 2015b, ApJ, 814, 130

Mullally, F., Coughlin, J. L., Thompson, S. E., et al. 2015, ApJS, 217, 31

Müller, H. M., Huber, K. F., Czesla, S., Wolter, U., & Schmitt, J. H. M. M. 2013, A&A, 560, A112

Nair, V., & Hinton, G. E. 2010, in Proc. 27th Int. Conf. Machine Learning (ICML-10), 807, http://dl.acm.org/citation.cfm?id=3104322.3104425

Nelson, B. E., Ford, E. B., Wright, J. T., et al. 2014, MNRAS, 441, 442

Ofir, A., & Dreizler, S. 2013, A&A, 555, A58

Oord, A., v. d. Dieleman, S., Zen, H., et al. 2016, arXiv:1609.03499

Panichi, F., Goździewski, K., Migaszewski, C., & Szuszkiewicz, E. 2017, MNRAS, submitted, arXiv:1707.04962

Pearson, K. A., Palafox, L., & Griffith, C. A. 2017, arXiv:1706.04319

Petigura, E. A., Howard, A. W., & Marcy, G. W. 2013a, PNAS, 110, 19273

Petigura, E. A., Howard, A. W., Marcy, G. W., et al. 2017, AJ, 154, 107

Petigura, E. A., Marcy, G. W., & Howard, A. W. 2013b, ApJ, 770, 69

Rogers, L. A. 2015, ApJ, 801, 41

Rowe, J. F., Bryson, S. T., Marcy, G. W., et al. 2014, ApJ, 784, 45

Rowe, J. F., Coughlin, J. L., Antoci, V., et al. 2015, ApJS, 217, 16

Sanchis-Ojeda, R., Rappaport, S., Winn, J. N., et al. 2013, ApJ, 774, 54

Sanchis-Ojeda, R., Rappaport, S., Winn, J. N., et al. 2014, ApJ, 787, 47

Schaefer, C., Geiger, M., Kuntzer, T., & Kneib, J. 2017, A&A, in press, arXiv:1705.07132

Schmitt, J. R., Wang, J., Fischer, D. A., et al. 2014, AJ, 148, 28

Schwarz, G. 1978, AnSta, 6, 461

Shporer, A., Zhou, G., Vanderburg, A., et al. 2017, arXiv:1708.08455

Skrutskie, M. F., Cutri, R. M., Stiening, R., et al. 2006, AJ, 131, 1163

Smith, J. C., Stumpe, M. C., Van Cleve, J. E., et al. 2012, PASP, 124, 1000

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

Stumpe, M. C., Smith, J. C., Catanzarite, J. H., et al. 2014, PASP, 126, 100

Stumpe, M. C., Smith, J. C., Van Cleve, J. E., et al. 2012, PASP, 124, 985

Thompson, S. E., Coughlin, J. L., Hoffman, K., et al. 2017, ApJS, submitted, arXiv:1710.06758

Thompson, S. E., Mullally, F., Coughlin, J., et al. 2015, ApJ, 812, 46

Tuomi, M. 2012, A&A, 543, A52

Vanderburg, A., & Johnson, J. A. 2014, PASP, 126, 948

Vanderburg, A., Latham, D. W., Buchhave, L. A., et al. 2016, ApJS, 222, 14

Wang, J., Fischer, D. A., Barclay, T., et al. 2015a, ApJ, 815, 127

Wang, J., Fischer, D. A., Xie, J.-W., & Ciardi, D. R. 2015b, ApJ, 813, 130

Wolfgang, A., & Lopez, E. 2015, ApJ, 806, 183

Youdin, A. N. 2011, ApJ, 742, 38

Zeiler, M. D., & Fergus, R. 2014, European Conference on Computer Vision (Berlin: Springer), 818

Ziegler, C., Law, N. M., Morton, T., et al. 2017, AJ, 153, 66