The Astronomical Journal, 161:273 (15pp), 2021 June https://doi.org/10.3847/1538-3881/abf4c6

Identifying Planetary Transit Candidates in TESS Full-frame Image Light Curves via Convolutional Neural Networks

Greg Olmschenk[1,2], Stela Ishitani Silva[1,3], Gioia Rau[1,3], Richard K. Barry[1], Ethan Kruse[1,2], Luca Cacciapuoti[4], Veselin Kostov[1], Brian P. Powell[1], Edward Wyrwas[1,5], Jeremy D. Schnittman[1], and Thomas Barclay[1,6]

[1] NASA Goddard Space Flight Center, Greenbelt, MD 20771, USA
[2] Universities Space Research Association, Columbia, MD 21046, USA
[3] Department of Physics, The Catholic University of America, Washington, DC 20064, USA
[4] Department of Physics "Ettore Pancini,"Universita di Napoli Federico II, Compl. Univ. Monte S. Angelo, I-80126 Napoli, Italy
[5] Science Systems and Applications, Inc., Lanham, MD 20706, USA
[6] University of Maryland, Baltimore County, 1000 Hilltop Circle, Baltimore, MD 21250, USA

Received 2021 January 26; revised 2021 March 23; accepted 2021 April 1; published 2021 May 21

Abstract

The Transiting Exoplanet Survey Satellite (TESS) mission measured light from stars in ~75% of the sky throughout its 2 yr primary mission, resulting in millions of TESS 30-minute-cadence light curves to analyze in the search for transiting exoplanets. To search this vast data trove for transit signals, we aim to provide an approach that both is computationally efficient and produces highly performant predictions. This approach minimizes the required human search effort. We present a convolutional neural network, which we train to identify planetary transit signals and dismiss false positives. To make a prediction for a given light curve, our network requires no prior transit parameters identified using other methods. Our network performs inference on a TESS 30-minute-cadence light curve in ~5 ms on a single GPU, enabling large-scale archival searches. We present 181 new planet candidates identified by our network, which pass subsequent human vetting designed to rule out false positives. Our neural network model is additionally provided as open-source code for public use and extension.

Unified Astronomy Thesaurus concepts: Exoplanet detection methods (489); Exoplanets (498); Neural networks (1933); Convolutional neural networks (1938)

1. Introduction

Astronomical photometric data sets are growing at an accelerated pace. Due to their sheer scale, these collections contain data that no human eye has ever seen, nor may ever see. The importance of automated systems, which can filter out data irrelevant to a particular research goal and flag the most promising phenomena, is essential in the era of big data.

The primary goal of the Transiting Exoplanet Survey Satellite (TESS; Ricker et al. 2014) mission is detecting planets orbiting stars via transit signals in flux measurements. Launched in 2018 April, TESS is performing a near-all-sky photometric survey intended to identify planets with bright enough host stars to enable mass estimation from ground-based radial velocity measurements (Ricker et al. 2014).

TESS is positioned in a high-Earth, 13.7-day, elliptical orbit. In 2020 July, TESS completed its 2 yr primary mission and entered into its extended mission. During the 2 yr primary mission, TESS recorded measurements of over 200,000 stars at a cadence of 2 minutes. Of more importance for this work, TESS recorded flux measurements of its entire field of view (24° × 96°) at a 30-minute cadence. These full-frame image (FFI) data cover ~75% of the sky and provide flux measurements of millions of stars (Ricker et al. 2014).

Along with an abundance of potential transit candidates, TESS's FFI data set presents a challenge: searching the vast data set in both an accurate and time-efficient way is not trivial. Machine learning (ML) generally and neural networks (NNs) specifically present a solution to this data filtering matter.

In recent years, deep neural networks (DNNs; e.g., LeCun et al. 2015) have come to dominate the field of ML. A primary reason for this is that NNs have the potential to approximate any transformation function (Cybenko 1989; Leshno et al. 1993; Zhou 2020). In this case, we aim to produce a transformation that converts observed data to the true physical classification. Any algorithm (handcrafted or machine-learned) can only approximate such a transformation. When an NN is trained for such a task, it attempts to learn the optimal data transformations for that specific task (Rumelhart et al. 1985). This has the potential to produce significantly more accurate classifications than handcrafted approaches, which often discard valuable information. Light curves (LCs) pre-processed for exoplanet hunting are often detrended using a general Gaussian process to remove stellar variability and other noise sources (Luger et al. 2016). Then, a box-fitting least-squares algorithm (e.g., Kovács et al. 2002) is often used on the detrended LC to search for transit signals. These processes have to tread a fine line between removing sources of noise and keeping useful signals. Excessive detrending can remove transit signals, while insufficient detrending leads to an abundance of false-positive transit detections. Furthermore, the removal of nontransit signals can often be detrimental to correctly identifying a planetary transit signal. For example, one of the most common sources of false-positive planet transit signals are eclipsing binaries (Armstrong et al. 2017), which have a transit signal similar to a planetary transit candidate. However, eclipsing binary LCs also often exhibit an ellipsoidal signal, which differentiates them from a planetary transit (Kostov et al. 2019). Gaussian process detrending often attempts to remove any form of periodic signal that is a non-planetary-transit signal, including the eclipsing binary ellipsoidal signal (Foreman-Mackey et al. 2017). This may leave only the eclipsing binary transit signal to be falsely detected as a planetary transit signal.

On the contrary, NNs do not explicitly remove sources of noise, but instead learn to use sources of noise to determine the likelihood that a given LC contains the desired transit signal, allowing the NNs to be effective for noisy data (Hinton et al. 2012; Dong et al. 2014; Xu et al. 2014). In the example above, rather than learning to remove the eclipsing binary ellipsoidal signal, the NN has the potential to learn that a transit signal with such an ellipsoidal signal likely originates from an eclipsing binary rather than a planet orbiting its host star.

As generalized function approximators (Cybenko 1989; Leshno et al. 1993; Zhou 2020), a sufficiently large NN can learn any handcrafted transformation, such as the above detrending and box-fitting algorithm. Moreover, if there is a modification to the handcrafted transformation producing better results according to the training process, the NN will instead learn that improved transformation. This property of NNs gives them the potential to outperform their handcrafted counterpart in nearly every situation.

This article is organized as follows: In Section 2 we present the photometric data used in our work. Section 3 presents our NN pipeline, including the NN architecture (Section 3.3) and the pre- and post-processing of the data (Sections 3.5 and 3.7). We also present the motivation of the network and processing choices in these sections. Section 4 shows the new vetted planet candidates identified by our network. Section 4 also provides analysis and discussion of the new planet candidate population. We conclude the work in Section 5.

2. Observational Data

For the present work we have used LC data, i.e., measures of flux over time. Figure 1 illustrates an example of a TESS LC, showing time versus flux of two TESS targets: TIC 258920431 and TIC 394346745. The time is given in Barycentric TESS Julian Date (BTJD) time (Tenenbaum & Jenkins 2018). With the Julian Date in the Barycentric Dynamical Time standard (BJD), BTJD = BJD −2,457,000.0. BJD is usually the most accurate time standard to use, as it accounts for many different timing corrections, including leap seconds (e.g., Eastman et al. 2010). The flux given is median normalized flux for the LC. The LC was produced by the `eleanor` pipeline (Feinstein et al. 2019) from raw flux measurements provided by TESS.

The examples shown in Figure 1 are planetary candidates identified by our NN. We selected these LCs to show cases of relatively simple noise and relatively challenging noise in our data set. These LCs demonstrate some typical sources of noise and data incompleteness in the TESS FFI data. Indeed, the repeating sharp downward spikes and the more gradual spikes near the start and end of an observing session are caused by spacecraft systematics and/or detrending; there is a gap in the middle of the data caused by the spacecraft pausing its observing to downlink data to Earth (Tenenbaum & Jenkins 2018).

Ideally, an LC would contain only the flux from a single TESS target (typically a star system). However, in reality each TESS pixel covers ∼21″ of the sky, and TESS's point-spread function results in blending between pixel measurements. For these reasons, an LC will contain flux from multiple targets. This often makes it challenging to determine which source the signal (or noise) is originating from.

2.1. TESS Data

The TESS data sets include 2-minute-cadence LC data and 30-minute-cadence LC data, both of which are relevant in this work. In the following sections, we describe each of these data sets and their use in our study.

2.1.1. TESS 2-minute-cadence Light-curve Data

TESS takes measurements of a large portion of the sky at regular intervals. During the primary mission, this interval was every 2 minutes. However, due to limitations of the spacecraft's storage and downlinking capabilities, only a small portion of this 2-minute cadence was retained (Ricker et al. 2014). For the present work, the set of known planets and planet candidates we employ for training our NN comes primarily from searches into the 2-minute-cadence data set. For each 2-minute-cadence measurement, the data come from small patches of pixels around targets likely to be of interest in that portion of the sky; a description of the selection criteria for TESS targets can be found in Stassun et al. (2018). The pixels within these patches that are suspected to contain a high signal-to-noise ratio are then summed to form a single flux measurement of an LC. Such summed fluxes are combined for each 2-minute-cadence measurement, forming the content of the 2-minute-cadence LCs we refer to in this work.

TESS collected ∼600,000 2-minute-cadence LCs from ∼200,000 targets during its primary mission (Tenenbaum & Jenkins 2018). Of these targets, ∼326 exhibit the transit signal of a confirmed planet (TESS Follow-up Observing Program Working Group 2020).

2.1.2. TESS 30-minute-cadence Light-curve Data

TESS discards most of the pixels from the 2-minute-cadence measurements (see Section 2.1.1). However, at a 30-minute cadence all pixels' values are retained and downlinked to Earth. These FFIs cover a much larger number of targets at a lower time resolution (Tenenbaum & Jenkins 2018). We used ∼67 million 30-minute-cadence LCs (with TESS magnitudes <15) in this work, and this is the primary data set investigated by our NN.

3. Neural Network Pipeline

A conceptual overview of our pipeline is shown in Figure 2. The new contributions of this work focus on development and use of the NN portion of this pipeline (including data preprocessing). Brief descriptions of the other pipeline components are given below. For details on the other components of the data pipeline, see E. Kruse et al. (2021, in preparation) for the production of LCs from TESS FFIs (via `eleanor` Feinstein et al. 2019), Kruse et al. (2019) for the Quasi-periodic Automated Transit Search pipeline (`QATS`), and Kostov et al. (2019) for the Discovery and Vetting of Exoplanets pipeline (`DAVE`).

3.1. Neural Network Primer

NNs are generalized transformation function learning machines. In our case, the LC is represented as an array of flux values, and the transformation function the network is learning is a transformation from the LC to a prediction of the likelihood that the LC contains a planetary transit signal. The transformation learned by an NN has two notable properties.

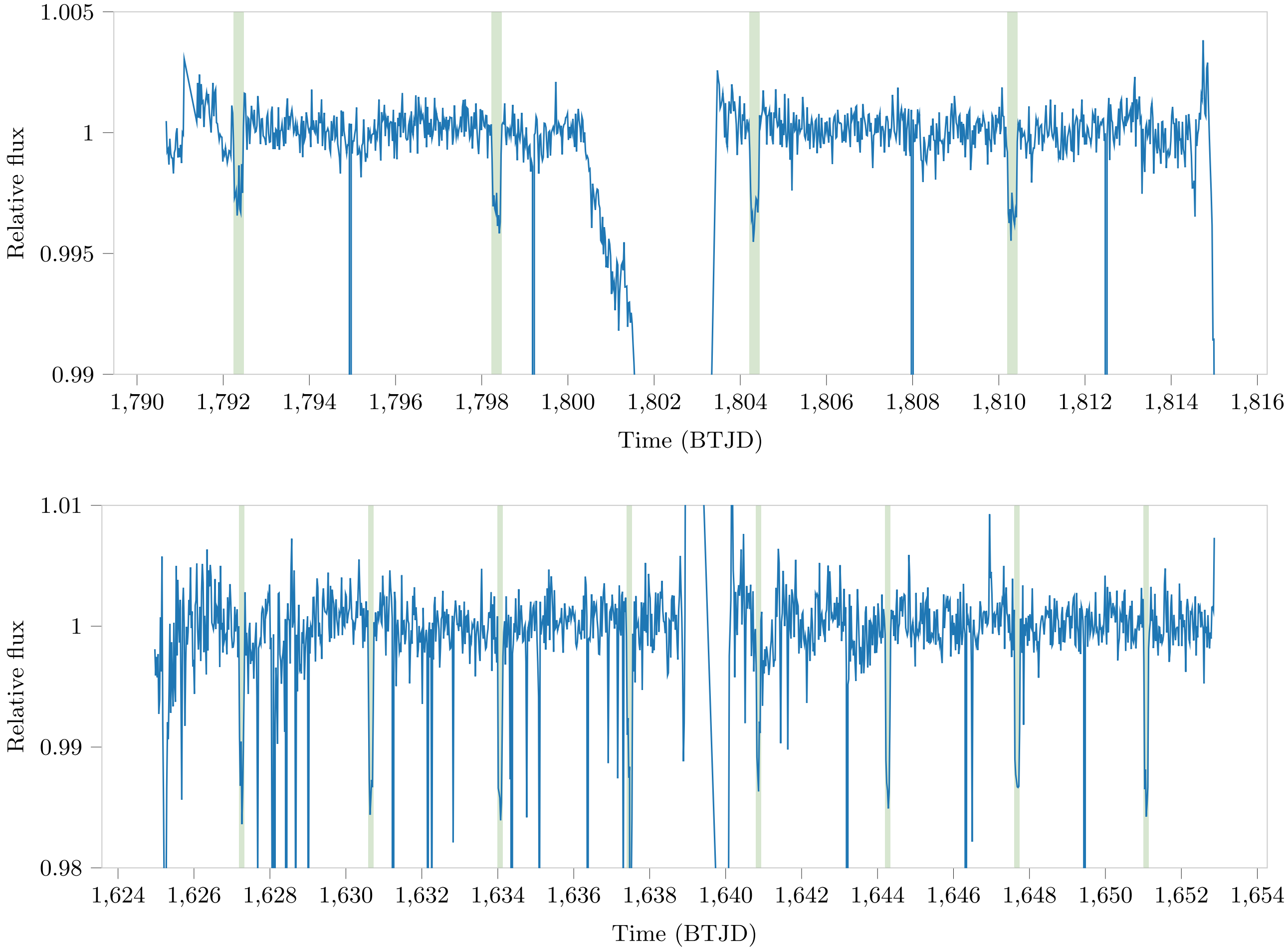


Figure 1. Examples of TESS 30-minute-cadence single-sector LC data from TESS observations for TIC 258920431 (top panel) and TIC 394346745 (bottom panel). Time is in BTJD = BJD −2,457,000.0. The dips in flux (highlighted by green vertical bars) are caused by the transits of planet candidates identified by our network.

First, a sufficiently large NN can approximate any transformation, including any handcrafted transformation (Cybenko 1989; Leshno et al. 1993; Zhou 2020). Second, the transformation function is automatically learned based on training examples. To train a network, we "show" the network examples of LCs known to contain a planetary transit signal and those known to *not* contain such a signal. We then "ask" the network which examples contain planetary transits and which do not. At the start of the training, the network will make predictions similar to random guesses. We use the network's confidence of each prediction to update the parameters of the network's transformation function (Rumelhart et al. 1986). Each parameter in the network is updated with a small change to produce a slightly better prediction for the LC(s) we are currently showing to it. The direction and magnitude of these parameter updates are determined through backpropagation (Rumelhart et al. 1986), which determines the derivative of the NN's parameters with regard to its prediction's correctness. By repeating these prediction and update steps many times, the network approaches a transformation function that can distinguish between LCs with and without planetary transit signals.

3.2. Design Choice Overview

The Exoplanet Follow-up Observing Program for TESS (ExoFOP-TESS; TESS Follow-up Observing Program Working Group 2020) has confirmed only a few hundred planetary targets within TESS LC data. Typically, this number of examples is too small to train an NN of the scale we are using without overfitting to the training data. However, we have layered several techniques to prevent overfitting and promote generalization. These include data augmentation (Section 3.5), data generation through injection (Section 3.6), and various network mechanisms (Section 3.3).

NN overfitting can be intuitively understood by considering the example of a large network with a small amount of training data. In such a case, the NN has the potential to memorize the training examples. For example, such an NN may learn to look exclusively at the first flux value of each LC. So long as this first flux value is unique for each LC, the NN can simply learn which unique values correspond to planetary transit examples and which correspond to non-planetary-transit examples. This produces an NN model that will perfectly distinguish between examples in the training data but will do no better than random guessing when applied to new data. This, however, is only one extreme case of overfitting. The NN may overfit based on several other LC features or statistics. Preventing such overfitting is one of the primary reasons we introduce the various augmentations and mechanisms below (e.g., Srivastava et al. 2014; Ioffe & Szegedy 2015).

When designing and training the network, we use 80% of the ExoFOP-TESS-confirmed planets. We use another 10% as validation data. These are data set aside that the network is not trained with. Instead, these data are used to evaluate the

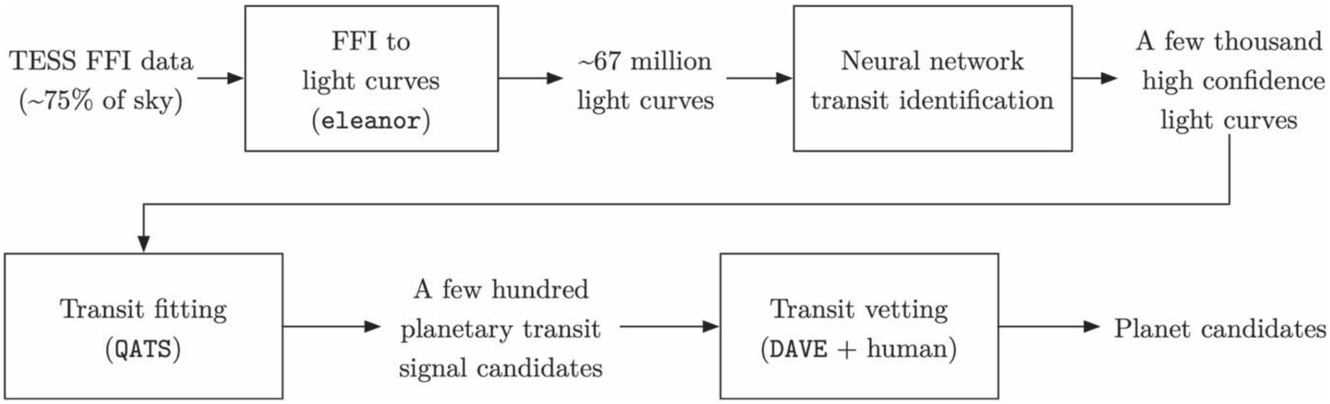


Figure 2. Conceptual overview of our data pipeline. The new contributions in this work are primarily those related to the development and use of the NN. We briefly describe other steps of the full pipeline in this work.

predictive performance of the trained network. This means inspecting how correctly the network makes predictions on data the network was not trained with, but where we know the correct answer. The remaining 10% is set aside as test data to evaluate the trained network after all design decisions are finalized. Several of the specific network and training setup design decisions were guided by preliminary performance results on the validation data. However, this validation evaluation and the evaluation on the test data are beyond the scope of this work. A detailed evaluation of the various network mechanisms and training techniques used in this work (and several excluded from this work) will be provided in G. Olmschenk et al. (2021, in preparation).

Throughout the following sections describing the network and related processing, we provide subsections explaining the rationale behind the design choices.

3.3. Network Architecture

In this work we used the 1D convolutional NN (CNN; Krizhevsky et al. 2012), which is shown in Figure 3.

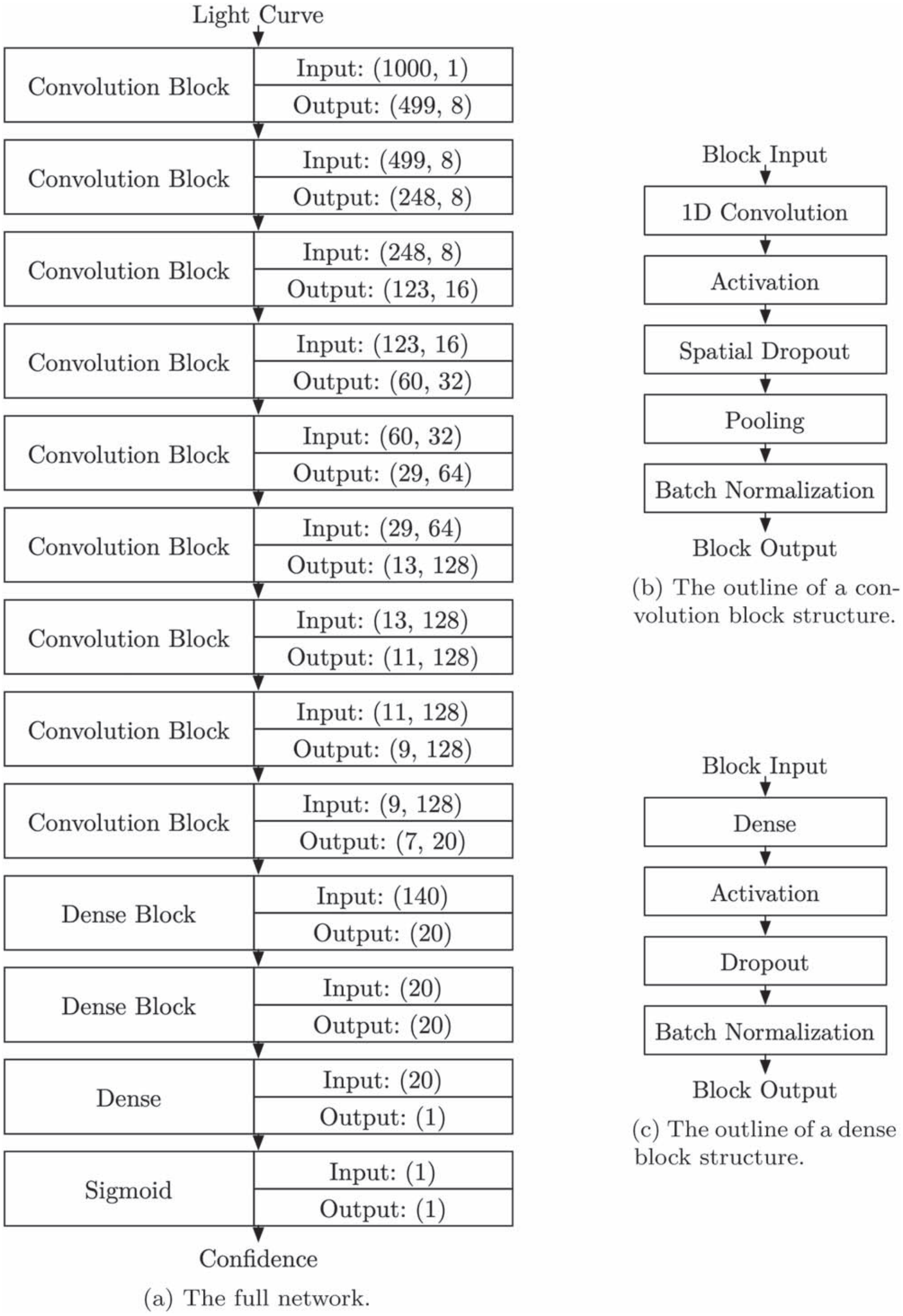
All layers use leaky rectified linear unit (ReLU; Glorot et al. 2011) activations, except the final prediction layer, which uses a sigmoid activation (Wilson & Cowan 1972). Excluding the first and the last two, all layers apply dropout (Srivastava et al. 2014) and batch normalization (Ioffe & Szegedy 2015). The dropout rate is 0.1 and is enabled during training and disabled during inference. We use spatial dropout (Tompson et al. 2015) for any convolutional layers. This is a version of dropout better suited for convolutional layers that drops entire features as opposed to individual neurons (Tompson et al. 2015). Batch normalization moving averages are updated only during training (Srivastava et al. 2014). We used max pooling following Krizhevsky et al. (2012). The ordering of the components in all layers is convolution/dense transformation, activation, dropout, pooling, and batch normalization. We used an Adam optimizer (Kingma & Ba 2014) on a binary cross-entropy loss to train the network. The output of the network for a given input LC is the network's confidence that the LC contains a planetary transit signal. We note that these are uncalibrated confidences, and therefore the distribution of the network's confidences may not directly correspond to the true underlying physical distribution.

The NN pipeline code is available at https://github.com/golmschenk/ramjet (see Olmschenk et al. 2021 for the code version used in this work). This pipeline is also installable as a PyPI package (https://pypi.org/project/astroramjet/). Documentation for the NN pipeline can be found at https://astroramjet.readthedocs.io/en/latest/.

3.3.1. Network Architecture Rationale

We chose a 1D CNN over other alternatives, such as dense networks, for the following reasons. First, we expect the NN to find individual transit events as a primary feature. In early layers, the NN should ignore the position of the transit in the LC and only determine their presence based on the local LC shape. As such, the early layers of our network are convolutional layers, which treat each segment of the LC identically (Krizhevsky et al. 2012), e.g., they search each portion of the LC for a transit occurring in that location. Only after local level features, e.g., individual transits are discovered, do we expect the network to combine these features into global level features—in this case repeating periodic transits. For this reason, in the NN the convolutional layers are followed by global dense layers. Another advantage of convolutional layers is to prevent overfitting; this happens because weights for convolutional layers are shared on every part of the input, preventing the network from applying specific weights to specific positions in the LC (Krizhevsky et al. 2012). Our NN requires no prior transit parameter information; the only input to the network are the flux values of the LCs. The NN performs inference on an LC in ~5 ms on a single GPU, allowing the entire data set of ~67M LCs to be inferred on in a few days.

A common source of overfitting comes from the NN identifying a specific training example, or set of training examples, based on a limited number of features that uniquely identify them; for example, the combination of the first and second flux values, which are floating point values, may be unique for every LC. This allows the network to distinguish training examples, but these unique features do not generalize to data the network did not train with. We added dropout to prevent such overfitting (Srivastava et al. 2014). For each layer where it is applied, dropout randomly sets feature activations to zero during training. This prevents the network from relying on a small number of features to determine the network confidence that specific training LCs contain transits. As the network can no longer rely on specific features to exist that uniquely identify training examples, the network is encouraged to produce features that detect transits, i.e., the shared feature contained in the positive training data. During inference of nontraining data, dropout is disabled to allow the network to use all features.



(a) The full network.

(b) The outline of a convolution block structure.

(c) The outline of a dense block structure.

Figure 3. General architecture of the CNN used in this work. See text for details. All convolution/dense layers within a block use a number of filters/units equivalent to the size of the last dimension of their output tensor. All convolutions use a kernel size of 3. To preserve the clarity of the diagram, three inconsistencies in network blocks are not shown. First, the first convolution block and the last dense block do not apply dropout or batch normalization. Second, the final convolution block applies a standard dropout instead of spatial dropout, as the following layer is a dense layer. Finally, only the first six convolution blocks use pooling (with pooling size 2). The remaining convolution blocks do not pool.

We apply batch normalization both to improve training dynamics and to prevent overfitting. On each layer where it is applied, batch normalization normalizes the activations of the layer for the given batch of training data. This aids in deep network training, as it provides well-distributed training gradients, thereby avoiding the vanishing/exploding gradient problem (Hochreiter 1998). Applying batch normalization also prevents overfitting, as each normalization depends on the batch of data and on the current network weights. As each normalization affects the input of the network layer, even small batch or weight changes have a cascading effect throughout the network. This makes it difficult for the network to overfit to specific LCs and encourages the network to converge toward a generalized solution (Ioffe & Szegedy 2015).

3.4. Full-frame Image Light-curve Production

For details of the FFI LC production, see E. Kruse et al. (2021, in preparation). Briefly, E. Kruse et al. (2021, in preparation) used the 129,000-core *Discover* supercomputer at the NASA Center for Climate Simulation, to build FFI LCs for all stars observed by TESS down to 15th magnitude. All original and calibrated FFIs were produced by the TESS Science Processing Operations Center (Jenkins et al. 2016). Target lists were created through a parallelized implementation of `tess-point` (Burke et al. 2020) on the TESS Input Catalog (TIC; Stassun et al. 2018) provided by Mikulski Archive for Space Telescopes (2020). The LCs for each sector were constructed in 1–4 days of wall clock time (for a total of over 100 CPU-years), depending on the density of targets in the sector, through a parallelized implementation of the `eleanor` Python module (Feinstein et al. 2019). A total of ∼67 million LCs were produced at the time this work was performed. These single-sector LCs are the input to the pre-processing and, subsequently, our NN.

3.5. Pre-processing

We used several forms of data augmentation to prevent network overfitting and to encourage generalization of learned features. In some cases, we did not apply the data augmentation during inference to allow for reproducibility and to allow for the best available input information during the inference phase.

During training, each time an LC is prepared for input to the network, the pipeline removes a random subset of the data points. The ratio of data points removed is randomly selected from $\mathcal{U}(0, 0.01)$. The implementation of this removal shifts the remaining values in the array such that there are no gaps. During inference, no data points are removed in this way.

Next, the LC is randomly rolled, i.e., a random position is chosen in the LC and the data are split at that location. The order of these two pieces is reversed. This rolling is not applied during inference.

Afterward, the pipeline repeats or truncates the LC to have a uniform length of 1000 data points. This is approximately the median length of a single-sector LC generated from TESS FFI data. LCs shorter than 1000 data points are repeated, with the first values of the LC being appended to the end of the LC, until they have 1000 data points. LCs longer than 1000 data points are truncated. This transformation is applied during both training and inference.

Finally, the flux values of LCs are normalized before being input into the network, using a normalization in the following way. A percentile normalization is applied such that the 10th-percentile flux is normalized to −1 and the 90th-percentile flux is normalized to 1. This flux normalization is applied during both training and inference.

3.5.1. Pre-processing Rationale

As described above, we performed several data augmentation steps. We have chosen to do so for the following reasons:

First, randomly removing data points during training helps prevent the network from overfitting. Indeed, a large NN has the potential to memorize exact values or ordering of values within the input data. By removing random data points during training, we encourage the network not to rely on specific data points but to use the overall structure of the LC instead (Zhong et al. 2020). Removing random data points during inference has no benefit and could potentially remove valuable information, so we only removed data points during training.

Second, the random roll of the LCs helps prevent the network from searching for specific positions of features within the LCs. Sector-specific noise can be easily memorized by its position in the LC, and rolling the LC forces the network to generalize feature recognition tasks to the general LC structure as opposed to a single part of it. This process splits the LC into two pieces and swaps the order of these pieces; therefore, the time between two transits where the LCs are recombined will not match the original period. As we expect, the network will take into account the period when inferring for any given LC, so this process may be a slight detriment to the training. However, we determined through preliminary validation experiments that the generalization benefit outweighs the cost. Similar to the random data point removal, there is no benefit to apply this step during inference.

Third, the uniform length of 1000 data points per LC allows for a significantly more efficient and practical training data set for our pipeline. This is because the network only needs to be designed for a single length input and can process in parallel large batches of uniform length inputs. In principle, this uniform LC length could cause two potential detriments to training: (1) transit events may be excluded when the LC is truncated, and (2) a pair of transit events may be artificially given the incorrect period when the LC data are repeated. However, both these factors play only a minor role in altering the LC before its input into the network. Therefore, we determined that for both cases the benefits of having this uniform LC length outweighed the costs.

Fourth and lastly, the data augmentation of percentile normalization of the flux values provides several benefits. Due to the previous data removal, rolling, truncating, and repeating, the LC is normalized differently in one training step than in another. This provides another deterrent to network overfitting, as the NN cannot rely on exact LC input values to define the LC label. Inputs well distributed from −1 to 1 provide several benefits to internal network training dynamics (LeCun et al. 2012) and allow for simplified weight initialization (Glorot & Bengio 2010). We chose a percentile normalization over a standard normalization, as this results in most of the data points being well distributed from −1 to 1. Notably, this provides better distributed values than it does a standard normalization where the minimum and maximum are scaled to −1 and 1, respectively. This is because several astronomical events result in outlier fluxes, which would result in the majority of an LC's data points being normalized to very near −1 or very near 1 when using a standard normalization . For example, a flare may result in a few flux values being relatively high. The standard normalization results in the nonflare flux values being normalized to close to −1 for all values. In contrast, the percentile normalization results in most values being normalized from −1 to 1, which provides better training conditions for the network.

3.6. Ground Truth Training Data Set

We used the dispositions of ExoFOP-TESS for ground truth training labels of LC transits. Positive cases included LCs corresponding to confirmed planets, according to the ExoFOP-TESS catalog. We included as negative cases any targets not listed by ExoFOP-TESS or that ExoFOP-TESS designated as false positives. We excluded from the training process any

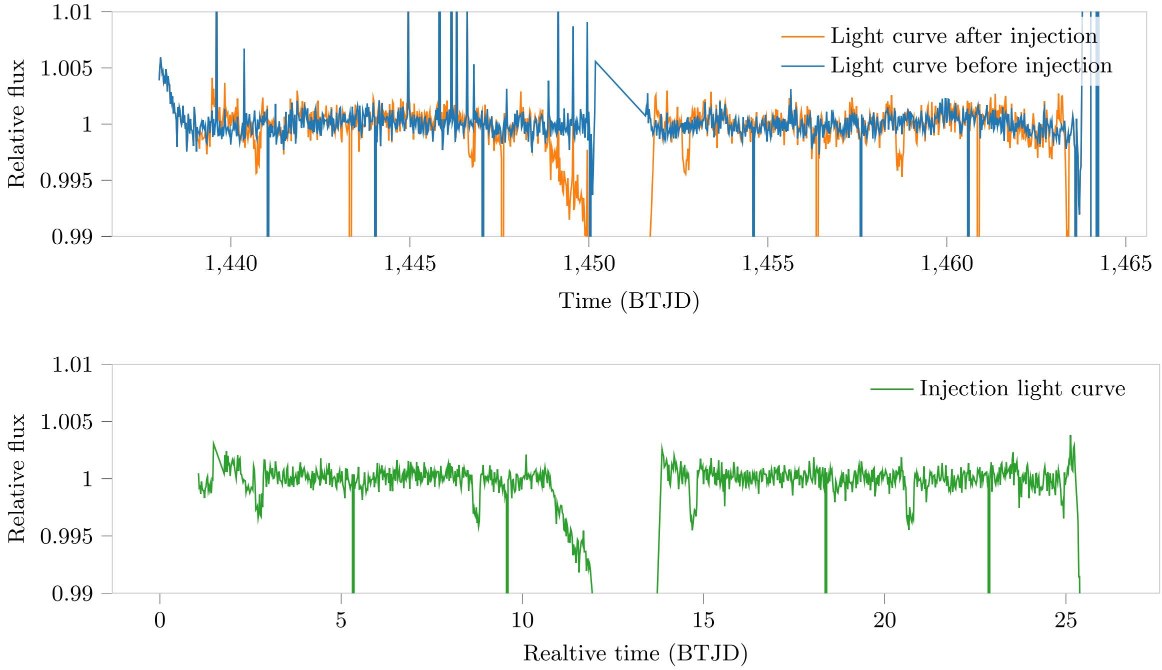


Figure 4. Example of the injection described in the text. The top panel shows an LC before and after it is injected with another LC. The bottom panel shows the LC used in the injection.

target designated by ExoFOP-TESS as a not confirmed candidate. This resulted in ∼377 30-minute-cadence LCs of targets with known planet transit signals. We used this collection of targets for training, validation, and testing of our NN.

In addition, we used a catalog of eclipsing binaries (E. Kruse et al. 2021, in preparation) as negative cases, as eclipsing binaries are the most likely targets to result in false positives (see Section 3.6.1).

The NN was trained using these initial training data sets, and human researchers analyzed the top candidates output by the network.

During training, we showed the three network base sets of LCs at equal rates: (1) LCs of known transiting planets, (2) LCs of targets from an eclipsing binary catalog, and (3) all available nontransit ExoFOP-TESS candidate LCs. In each case, only TESS 30-minute-cadence LCs were used as described in Section 2.1.2.

We additionally trained the network with LCs artificially injected with signals from another LC. Thus, in addition to the three base LC sets above, we trained the network with three injected LC sets, one for each of the three base sets. The corresponding injected set for each of the base sets is produced as follows. During training, we randomly sampled an LC from the base set. This LC is median normalized to produce a relative magnification signal. We then randomly sample an LC from the nontransit candidate set and then multiply each value in this LC by the relative magnification signal generated from the previous LC. An example of this injection is shown in Figure 4. Please note that the nontransit LC set is always used as the source of LCs to have a signal injected into, but it is also used for the source of signals to inject in one of the three injected sets.

During this injection process, we interpolated linearly between generated signal times to determine signal magnifications to be injected. LCs injected with signals from the known transit base LC set are labeled with a positive ground truth label. Those injected with signals from the eclipsing binary and nontransit candidate sets are labeled with a negative ground truth label. The resulting LCs produced by this injection process are treated identically to base set LC (e.g., are prepared for input to the network using the same pre-processing steps). We train the network sampling evenly from each of the six LC sets, i.e., using three base sets and three injected sets.

3.6.1. Ground Truth Training Data Set Rationale

The known planetary transit signals we use to train our network were identified primarily by non-NN search methods working on the 2-minute-cadence LC data (Huang et al. 2020). However, we train the network using the corresponding 30-minute-cadence LC data. In this way, the NN is trained to predict the same labels on lower-quality data that traditional methods obtained using higher-quality data. That is, the network is not simply learning to duplicate the traditional methods; it must learn to identify the same targets with 1/15th the cadence. This has the potential to have the NN to learn patterns that would be missed by traditional algorithms.

We used two types of negative data sets to train our NN: (1) the data set of all 30-minute-cadence LCs, which are from targets that are not known planetary candidates or confirmed planets; and (2) the data set of all 30-minute-cadence LCs labeled as eclipsing binary candidates according to E. Kruse et al. (2021, in preparation). The reasoning for having the two separate sets of negative LCs arises from the ratio of occurrences of each type of phenomenon. The vast majority of TESS 30-minute-cadence LCs (>99%) are expected to contain neither planetary transit signals nor eclipsing binary signals. Most of the remaining LC signals are not due to transiting planets, but instead to eclipsing binaries (>95%; Sullivan et al. 2015).

We excluded from the training process any targets designated by ExoFOP-TESS as a planet candidate that were not confirmed. The reason behind this choice is that candidates added to ExoFOP-TESS are frequently subsequently confirmed or designated as false positives, and we want to train our network only with signals of planets (TESS Follow-up Observing Program Working Group 2020).

To understand our training setup, we first considered a simpler setup that would sample from all available LCs equally, assigning the appropriate label to each LC. In this training setup planetary transit LCs would be very rare, and the CNN would have little incentive to learn to predict them, as predicting negative in every case would provide the correct answer for nearly every LC. This would be compounded by frequent mini-batches, which have no positive cases and would then result in training noise.

Next, we considered a training setup using equal cases of negative and positive LCs. This forces the CNN to learn to distinguish between non-planetary-transit LCs and planetary transit LCs. However, in this case there is an issue in training data ratios. The CNN can obtain the correct prediction on nearly every LC it is shown simply by labeling any periodic event as positive and any nonperiodic event as negative. While this will help to filter out quiescent or otherwise nonperiodic LCs, the vast majority of LCs labeled as positive will have signals not caused by transit events. Instead, they will be most often due to eclipsing binaries, or other periodic signals somewhat resembling short dips in flux.

The solution we employed to handle these labeling imbalances was to train the CNN with LCs sampled equally from three sets of LCs: (1) all negatives, (2) eclipsing binaries, and (3) planet transits. In this way, the CNN must be able to distinguish nonperiodic events from periodic ones, to correctly make predictions about the general negative LCs, but it must also be able to distinguish eclipsing binaries from planetary transit events. As eclipsing binaries often look very similar to planetary transits (Kostov et al. 2019), this forces the CNN to learn explicitly how a planetary transit appears relative to other types of periodic signals.

We produced the artificially injected LCs to provide the network with examples of signals in a variety of real noise. With only a few hundred confirmed training examples, NNs may be prone to only learn the specific positive examples provided. By injecting the known signals into other LCs, we force the network to learn to recognize transit signals within any other LC in our training data set. This encourages the network to learn that the transit signal is the important feature to identify and encourages it to learn how to ignore any other signals.

These artificially created LCs retained noise from both the injected and the injectee LCs. This results in statistically more noisy artificial LCs than the average real LC. This could lead the network to wrongly learn to give higher confidence to noisier LCs. To counteract this, we also used injected eclipsing binary and nontransit signals. In order to provide realistic noise cases, we trained with both the injected LCs and the original real-data LCs; this choice has been made because the real data are expected to have similar amounts of noise to the data the network will perform inference on.

The evaluation of the impact this injection technique has on predictive performance goes beyond the scope of this work and will be presented in G. Olmschenk et al. (2021, in preparation).

3.7. Post-processing

After the network produces a confidence value for each LC, an arbitrary number of the highest-confidence candidates are passed through the post-processing portion of the pipeline.

First, the candidate target LCs are passed through QATS, which provides the fitting of a transit model for each LC. As with the NN, the input LCs used by QATS (Kruse et al. 2019) are produced via the eleanor pipeline (Feinstein et al. 2019). As eleanor cleans and detrends the LC data, occasionally this process results in an LC with spurious relative flux scales. In particular, in the processed version of the LC, the depths of the transits may be artificially reduced. As QATS uses these relative flux scales to estimate the depth of the transits, this also affects our radius estimates. Notably, our candidates with the smallest radii likely have underestimated radii. Most of the estimates are expected to be accurate. The transit model determined by QATS supplies transit parameters such as transit depth, duration, period, and epoch. These transit parameters are used by the remaining parts of the pipeline.

Based on the transit parameters determined by QATS, we filter candidates on the predicted radius r_p, calculated as follows:

$$r_p = r_t\sqrt{d(1 + c)}, \tag{1}$$

where r_t is the target star radius obtained from the Gaia mission's (Brown et al. 2018) data release 2, c is the target's background contamination obtained from the TIC, and d is the transit depth as modeled by QATS. The pipeline discards planet candidates with predicted radius greater than $1.8\,R_{\text{Jupiter}}$. We chose this threshold to be over the 95% exoplanet radii expected to be discovered in TESS FFI data (Barclay et al. 2018), while still being below the radii of largest known exoplanets (Crouzet et al. 2017; Zhou et al. 2017).

The pipeline then passes any candidates that are not removed from the above filtering to DAVE (Kostov et al. 2019), which provides an automated vetting of transit candidates. This includes checking for secondary signals and for in- and out-of-transit difference image photometric centroid shifts.

Finally, a group of exoplanet researchers visually examine QATS and the DAVE analysis results to accept or reject the candidates.

3.7.1. Post-processing Rationale

The pipeline discarded planet candidates with predicted radius greater than $1.8R_{\text{Jupiter}}$. This threshold allows more than 95% of planet radii expected to be found in TESS FFI signals (Barclay et al. 2018), while being below the radii of the largest known exoplanets (Crouzet et al. 2017; Zhou et al. 2017). Objects with a radius greater than this threshold might be brown dwarfs (Carmichael et al. 2020).

The final human analysis, done with the results of QATS and DAVE, consisted of removing any candidates that were likely caused by a non-planetary-transit signal. The most common source of such false positives were eclipsing binaries. This occurred most often when a nearby eclipsing binary's signal appeared in the target's LC. Often, this can be seen owing to an in/out-of-transit photodifference centroid offset (Kostov et al. 2019). Often targets have another neighboring target a subpixel distance away, where the brightness of the neighboring target was such that an eclipsing binary transit signal from that source would appear as a planetary transit signal, from the primary target. In such ambiguous cases, the candidate was discarded.

3.8. Active Learning

The positive labels confirmed by the researchers were fed back to the network's training process in order to supplement its list of positive training candidates. Some newly identified eclipsing binary cases were also labeled as negatives and fed back to the training process. We performed this active learning in a subjective fashion, and no formal process was used to guide when or how it should occur. However, we provide here an approximate description of the process. Once the network training had converged, we passed ∼1000 candidates with the highest confidence to the post-processing portion of the pipeline. The output of QATS typically showed ∼90% of these candidates to have unrealistic physical parameters for transiting planets, leaving ∼100 to be analyzed by DAVE and a human researcher. A total of ∼30% of these candidates passed the human vetting reviewing the output of DAVE, leaving ∼30 candidates. These candidates were then added back into the training data set. This process was repeated ∼6 times.

4. Results and Analysis

The primary output of this work are the human-vetted planet candidates shown in Table 1. These 181 candidates have passed the entire automated vetting process and were verified by humans. The radii given in Table 1 are estimated using the method described in Section 3.7. The distribution of the candidates' radii is shown in Figure 5.

In this section, we examine the planet candidates we identified and compare them to the confirmed ExoFOP-TESS planets. We also compare our findings with the estimations from Barclay et al. (2018) on the expected exoplanet yield of the TESS mission and expected physical properties of the population of exoplanets and their host stars. Follow-up analyses, especially radial velocity measurements, are necessary to confirm our candidates as planets; however, we compare various properties of our candidates with the previously confirmed planets from ExoFOP-TESS and expected planetary detections for TESS. The following sections detail this property comparison. Generally, our candidates have properties consistent with the confirmed and expected planet distributions.

4.1. Conditional Candidates

Often, a potential transit signal will have an ambiguous source owing to the proximity of two or more potential sources. Most signals where the source target is ambiguous are not included as candidates in our list (see Section 3.7). The exception to this is when the signal would result in a planet candidate regardless of which of the ambiguous sources the signal originates from. Of our planet candidates, there are four for which there are two potential host star targets, where the signal coming from either source would suggest a radius consistent with a planet. These candidate source pairs are (TIC 372596795, TIC 372596796), (TIC 120232318, TIC 120232321), (TIC 360816293, TIC 360816296), and (TIC 452810326, TIC 452810327). Our total count of candidates includes these four candidates. Table 1 includes all eight ambiguous source targets and the parameters assuming that the

Table 1
List of Human-vetted Planet Candidates by TIC ID with Transit Parameters

TIC ID	Candidate Radius (Jupiter radii)	Transit Epoch (BTJD)	Orbital Period (days)	Transit Duration (hr)	Transit Depth (ppm)
7548817	1.45	1687.086	4.5207	4.597	8864
9443323	1.41	1818.027	4.0135	3.923	7659
11755687	1.24	1468.686	3.0525	2.308	15591
12090836	1.61	1470.350	3.1972	3.740	10744
14173089	0.69†	1441.471	4.0756	3.113	3709
21279791	0.90	1685.697	3.0183	3.232	7249
27414976	1.39†	1741.338	3.9765	2.339	32848
32677675	1.61	1469.366	3.6366	3.900	17487
32949762	1.21	1468.981	3.7682	2.931	12042
34371411	1.41	1468.603	3.8828	4.489	8918
35022727	0.44	1793.140	3.6521	2.851	2234
35636165	1.45†	1470.308	3.4592	2.854	6298
37862966	0.77	1439.935	3.5837	3.190	5148
38399060	1.19†	1411.722	3.6523	3.557	20164
38965512	1.16†	1327.556	2.2689	3.040	11746
39724477	1.20	1469.452	4.0353	2.374	24447
43475220	0.35†	1440.064	18.5202	3.518	1348
45896295	1.25	1520.296	3.0045	3.353	7642
49045066	1.30	1544.069	3.0072	3.828	7119
52745699	1.45†	1470.621	3.1659	3.351	13464
53461742	1.21†	1470.723	4.4118	2.861	5186
55849732	1.05†	1327.282	5.3708	3.186	11362
56096837	1.19	1441.052	4.6640	2.857	10173
63889661	1.45†	1493.416	2.9711	3.031	6167
66296167	1.24†	1819.491	3.4124	2.986	15164
74274839	1.20	1519.195	4.6955	2.832	16234
74712191	1.47	1523.877	7.2310	4.822	9744
77287067	1.14†	1411.391	4.0762	4.079	7916
78441371	N/A‡	1469.528	3.9985	4.441	5597
85429139	1.55	1818.389	4.7427	3.406	7050
88385463	1.26	1845.469	3.1235	2.840	13961
95589845	1.33	1493.886	3.5657	4.208	5791
102713734	1.76	1819.066	3.3976	4.665	8672
103751498	1.41	1686.314	3.7146	4.000	11286
104195270	0.39	1792.626	3.6859	4.906	312
104986789	1.18	1818.762	3.4581	2.708	8112
105379013	0.91†	1871.426	3.2950	2.552	11121
107340585	1.34	1493.547	2.6893	2.554	10833
109929845	1.27†	1494.648	4.7634	4.712	38070
112316665	1.42	1495.136	3.7479	3.461	5018
115453244	1.35	1817.882	3.8231	3.798	11376
119685627	1.13	1472.760	5.0324	4.797	2307
120232318	1.31$^{\dagger *}$	1742.194	3.6204	5.366	2590
120232321	1.36$^{\dagger *}$	1742.194	3.6203	5.281	2638
137389526	0.82	1820.004	3.9116	4.029	2775
142628514	1.42	1493.628	2.6262	3.366	4868
142784687	1.16†	1684.026	3.1712	3.585	6412
147034452	0.34†	1470.893	4.5304	3.106	1771
147797743	1.40	1843.989	3.6915	2.634	16691
151483286	1.37	1494.244	4.0398	3.776	11916
152245179	0.88	1492.059	3.7897	3.700	937
153937417	0.92†	1843.119	4.1595	2.614	16780
155873992	1.31	1685.204	4.9513	3.924	8447
155993822	1.48	1415.689	12.2838	4.697	8621
156999527	1.33	1471.370	4.3932	2.697	9954
159332859	1.70	1684.774	4.4818	2.352	18702
160004025	1.24†	1383.517	2.7983	2.691	13063
160037058	1.51†	1329.504	4.5821	3.418	20104
160432093	1.67	1715.073	3.9916	4.356	8640
161003569	1.23	1740.431	3.5115	3.639	7845
165464482	1.35	1713.025	4.3346	3.630	9015

Table 1
(Continued)

TIC ID	Candidate Radius (Jupiter radii)	Transit Epoch (BTJD)	Orbital Period (days)	Transit Duration (hr)	Transit Depth (ppm)
167671392	1.52[†]	1441.306	3.9633	3.225	19450
167714124	1.08[†]	1439.728	4.1628	3.346	12590
168903062	1.22	1413.056	2.6739	2.141	18947
175604949	1.14	1492.618	2.6488	2.215	15943
178162579	1.08	1493.862	3.5509	2.393	6217
178242751	0.85[†]	1412.718	5.2962	4.242	4635
182405015	1.36	1494.364	3.1123	2.589	13989
186936449	0.41	1518.648	2.8448	2.480	947
192976435	0.86	1546.779	4.2274	4.324	2301
193754373	1.57	1687.190	3.9540	4.196	5762
200606159	1.43	1742.810	11.9466	3.755	15414
201660996	1.43	1656.373	3.5028	3.232	13760
202059363	1.73	1472.225	4.2465	4.468	6318
202425357	1.39	1712.269	3.6216	2.311	8654
207339000	1.25	1684.298	4.6183	2.775	20041
220460087	1.24	1355.995	3.2707	4.036	4047
220524097	0.07[†]	1325.971	3.9614	2.222	79
229581160	1.26[†]	1685.895	3.4149	2.723	12536
229671380	1.35	1683.487	3.9394	5.828	3996
230980206	1.27	1357.499	3.3763	3.610	10903
233188747	1.27	1683.605	3.4506	3.190	14658
233823679	0.83	1713.577	2.7585	3.420	5066
234489133	1.42[†]	1491.724	3.5996	4.794	9666
235072851	0.10[†]	1440.941	3.1191	1.906	188
237205154	1.26[†]	1683.523	5.8930	3.844	11719
237406657	0.31	1470.639	3.7474	4.793	235
237637903	1.22	1471.716	4.1053	3.326	8858
239638934	0.84	1712.806	6.0013	4.200	5285
239816546	1.42	1816.314	2.9886	3.018	12587
243333538	0.52[†]	1684.275	2.9986	2.970	2500
248655096	0.27[†]	1439.638	2.7623	2.163	1053
248655556	1.14	1440.246	13.1180	4.576	4384
250707118	1.25	1797.579	11.4461	4.011	7806
255780160	0.33[†]	1470.528	3.8767	3.225	606
256158543	1.37	1627.062	5.0652	4.432	9880
257060897	1.67	1683.378	3.6600	4.662	7877
257067559	1.33	1492.005	3.4834	3.674	5619
258920431	1.09	1684.647	5.9840	5.577	3729
262570313	0.09[†]	1468.479	2.9285	3.026	31
267545252	1.26[†]	1683.530	3.3792	3.093	17745
268187322	1.45	1496.527	5.4792	4.638	5659
269859655	1.50	1544.685	2.9920	2.843	18520
272785423	1.31[†]	1685.872	3.2400	3.066	18371
273270473	1.27[†]	1656.567	2.7088	2.215	22467
278825345	1.06[†]	1332.214	7.9892	3.156	16391
279383896	1.17	1771.671	12.0321	4.875	4440
279947414	1.20	1791.690	4.0664	2.155	18326
280613644	1.50[†]	1494.736	3.7211	3.220	17813
284173938	1.32	1817.484	2.9160	3.651	11308
284206913	0.88	1768.011	10.5113	3.203	7683
284679064	0.73[†]	1712.601	4.2103	2.913	7978
284706890	0.58[†]	1440.122	3.8359	3.519	2742
284859630	0.45	1440.048	7.5429	4.274	1735
285272237	0.56	1795.809	13.9175	2.797	3064
285524410	N/A[‡]	1491.736	3.0251	3.658	3108
287145649	0.32	1546.403	3.6021	3.136	948
292477426	1.53	1654.934	4.1801	4.112	5903
296354161	1.72	1496.001	6.0067	6.912	6636
300604770	0.93	1325.625	3.4563	2.354	10081
302381397	1.22	1818.452	2.7283	2.593	9941
306648160	1.47	1626.701	3.6525	4.302	9622

Table 1
(Continued)

TIC ID	Candidate Radius (Jupiter radii)	Transit Epoch (BTJD)	Orbital Period (days)	Transit Duration (hr)	Transit Depth (ppm)
310002617	1.26	1685.330	3.1775	3.004	12839
317252733	1.12	1819.143	3.3527	3.963	11350
318756174	1.26	1493.110	3.3744	3.115	12672
321086563	1.18	1494.057	3.9704	4.596	3389
323194443	0.72[†]	1767.802	3.9077	3.155	12706
323295908	1.13	1417.653	7.7061	3.386	14958
327369524	0.79	1767.257	4.0237	3.743	1970
328814605	1.18	1823.368	11.3840	5.593	4791
331146317	1.76	1740.850	2.3875	2.708	11388
332579116	1.10	1441.196	3.9025	1.993	21885
332860211	1.49	1469.638	4.4440	2.876	16600
334482103	1.06	1492.273	3.7260	2.132	8551
336399144	0.73	1738.924	5.7033	3.372	3091
340228388	1.30	1599.301	3.1727	2.261	17303
343933335	1.24[†]	1471.997	4.3703	5.777	12260
349582831	1.55	1816.954	3.4541	2.222	13220
350335633	0.82[†]	1326.926	2.3888	3.614	3026
353960022	1.97[†]	1819.195	3.9214	4.430	18015
359648342	1.55[†]	1545.214	3.9498	3.755	13007
360816293	1.41[*]	1597.648	3.5506	2.784	7872
360816296	1.63[*]	1597.648	3.5506	2.784	7872
370010846	1.24[†]	1327.004	3.4854	2.927	14683
371864043	1.66	1571.788	2.9599	3.789	6916
372443687	1.08	1657.771	4.2952	2.857	11120
372596795	1.31[*]	1571.058	3.5367	2.678	6135
372596796	1.32[*]	1571.058	3.5367	2.737	6274
382521776	1.05	1358.257	4.3943	2.968	11609
387409975	0.45[†]	1739.847	5.0182	3.038	1986
394346745	1.42	1327.840	3.4025	3.265	13811
396246765	1.34	1656.165	3.4414	3.513	8726
396896334	1.43[†]	1441.838	5.7603	3.150	16507
399346877	0.17	1792.026	3.9607	3.581	146
400103802	1.10	1817.927	4.2512	2.892	13752
400432230	1.33	1684.205	3.1834	2.410	5427
403507814	1.79	1627.340	2.7800	3.463	11664
403693870	1.40[†]	1469.703	3.5245	3.112	14204
407653728	1.33[†]	1714.814	6.5037	3.708	11017
410119093	1.13	1657.624	4.1293	2.847	9708
415443327	0.33[†]	1440.584	4.8899	3.924	532
417646390	1.31	1819.394	3.8016	3.329	7155
417942201	1.47	1817.719	4.2307	4.735	4763
420016507	1.65	1411.458	3.4073	4.213	10833
420177051	1.35	1685.774	4.0484	3.251	11887
422385684	1.59	1739.111	2.7754	3.480	8596
428892437	1.24[†]	1819.483	3.7519	3.619	16277
438073782	1.70	1468.984	3.6413	3.871	9906
441122551	1.06[†]	1326.564	3.3856	3.138	15577
441155146	1.30[†]	1327.617	3.5510	2.948	17684
441453629	1.42[†]	1329.237	4.2006	2.676	24451
441763252	1.14[†]	1684.793	2.7709	2.379	21126
445582110	1.22	1492.217	3.0581	3.002	5854
446158352	1.56[†]	1818.668	4.0433	4.517	8015
452810326	0.76[†*]	1439.365	2.3539	2.196	2696
452810327	1.08[*]	1439.365	2.3539	2.196	2696
454248975	1.67	1571.801	3.9016	4.946	8872
455006978	0.20[†]	1438.604	2.7510	2.687	198
455036659	0.53	1440.708	4.0070	4.530	917
457104362	1.56[†]	1843.355	4.8160	4.583	17739
458641144	1.48	1493.843	4.6974	3.253	10591
462206806	1.51	1545.385	3.3736	3.434	11061
468178519	1.70	1626.064	2.9851	3.962	13101

Table 1
(Continued)

TIC ID	Candidate Radius (Jupiter radii)	Transit Epoch (BTJD)	Orbital Period (days)	Transit Duration (hr)	Transit Depth (ppm)
468832904	1.55[†]	1817.395	1.4005	2.346	21551
470171739	1.64	1795.184	4.6174	4.653	9619

Note. Candidate radius is estimated based on several assumptions (see text for details). A "†" mark signifies cases where the contamination ratio of the target is unknown and, in the radius estimation, zero is used. A "‡" mark signifies cases where a radius could not be estimated owing to a lack of stellar parameters. A "*" mark signifies that there is ambiguity about whether this target or another marked with "*" is the candidate host (see Section 4.1).

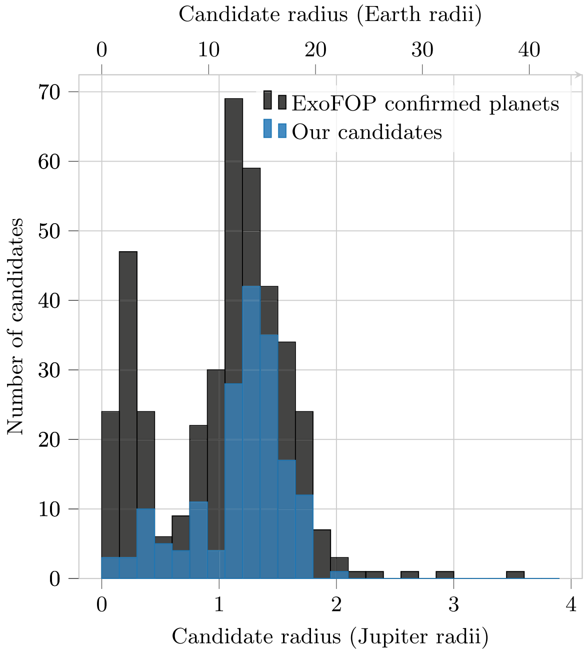


Figure 5. Radius distribution of our planet candidates and the confirmed planets from ExoFOP-TESS.

candidate is from that source target. All other figures exclude these ambiguous targets.

4.2. Comparison with the ExoFOP-confirmed Planets

ExoFOP-TESS (TESS Follow-up Observing Program Working Group 2020) provides follow-up studies of targets observed by TESS. ExoFOP-TESS uses the stellar parameters from the TIC (Stassun et al. 2018) and planet parameters from the NASA Exoplanet Archive (Akeson et al. 2013). As our network is trained using the LCs of targets with ExoFOP-TESS-confirmed planets, we expect our candidates' LCs to exhibit similar features; the network learns that these features correspond to planetary transit signals. When these LC features correspond to planet and/or star properties, we expect the characteristics of our candidates to be similar.

For example, the majority of ExoFOP-TESS-confirmed planets have a period of less than 5 days. As such, we expect our network to be inclined to search for planets with similar periods. Indeed, this tendency is observed in Figure 6. This trend might likely be due not only to the training data distribution but also to the shorter period resulting in more transiting events in a single LC, which likely makes the candidate easier to detect. The LCs used by the NN are single TESS sector LCs, which have observing periods of ~27 days. This aspect of the data results in the majority of the transits identified having a relatively short period, as these are the only cases where multiple transits can be observed within a single sector. While there is nothing that explicitly restricts the network from labeling an LC as a candidate even if it only contains a single transit event, nonperiodic events are likely discouraged by the training process owing to the possibility that they are caused more frequently by nonplanetary sources. We expect the network to likely be more confident about signals with many periods being exhibited. This, combined with the majority of training samples being of short period, probably plays a factor in the network's decisions. At the same time, Figure 7 shows that the network does not seem to overemphasize predicting candidates with longer-duration transits.

In addition to orbital period, Figure 6 also shows the radii of our candidates and ExoFOP-TESS-confirmed planets. Similar to the case of the periods, our candidates show a similar distribution of radii when compared to the ExoFOP-TESS-confirmed planets. The majority of candidates have a radius larger than Jupiter, with a smaller number of candidates having a radius between Jupiter and Earth. When observing the smallest-radius candidates in Figure 6, we again note the potential for spurious small-radius estimates (see Section 3.7).

Figure 8 shows a color–magnitude diagram of the host stars of our candidates and the ExoFOP-TESS-confirmed planets. The range in TESS magnitudes of the host stars of the planet candidates from our work is comparable to that of the host stars of planets confirmed by ExoFOP-TESS. The host stars of our candidates have a higher magnitude (are less bright) than the host stars of the ExoFOP-TESS-confirmed planets. This is expected, as TESS selects relatively bright targets for 2-minute-cadence observing (Ricker et al. 2014) and the TESS team's FFI search (the Quick Look Pipeline, `QLP`) only searched for planets down to a magnitude of 13.5 (Huang et al. 2020). In comparison, our network uses the TESS FFI LCs, which include dimmer stars. Indeed, the number of potential targets increases exponentially as the magnitude increases. However, we do not expect candidates to increase exponentially with the number of targets, as transit events become more difficult to detect around dimmer targets, whose signals are relatively more contaminated with sources of noise.

By comparing the position in the sky of our candidates for host stars with those confirmed by ExoFOP-TESS, we note a homogeneous distribution of candidates across the sky (no region with strong preference). This comparison is shown in Figure 9. This is expected, as the network design does not take into account sky position and has no mechanisms designed to specifically prefer any region of the sky. One bias we expect from the network with regard to sky position would be a preference to candidate positions similar to those in confirmed ExoFOP-TESS distribution, as these were the positions of the training examples. This may lead the network to prefer LCs with sector-specific noise from the sectors containing the most confirmed planets. However, to find candidates in similar regions to the confirmed planets may also simply be due to these regions having clearer signals and less noise, in which case the reason for candidates being in similar regions may be

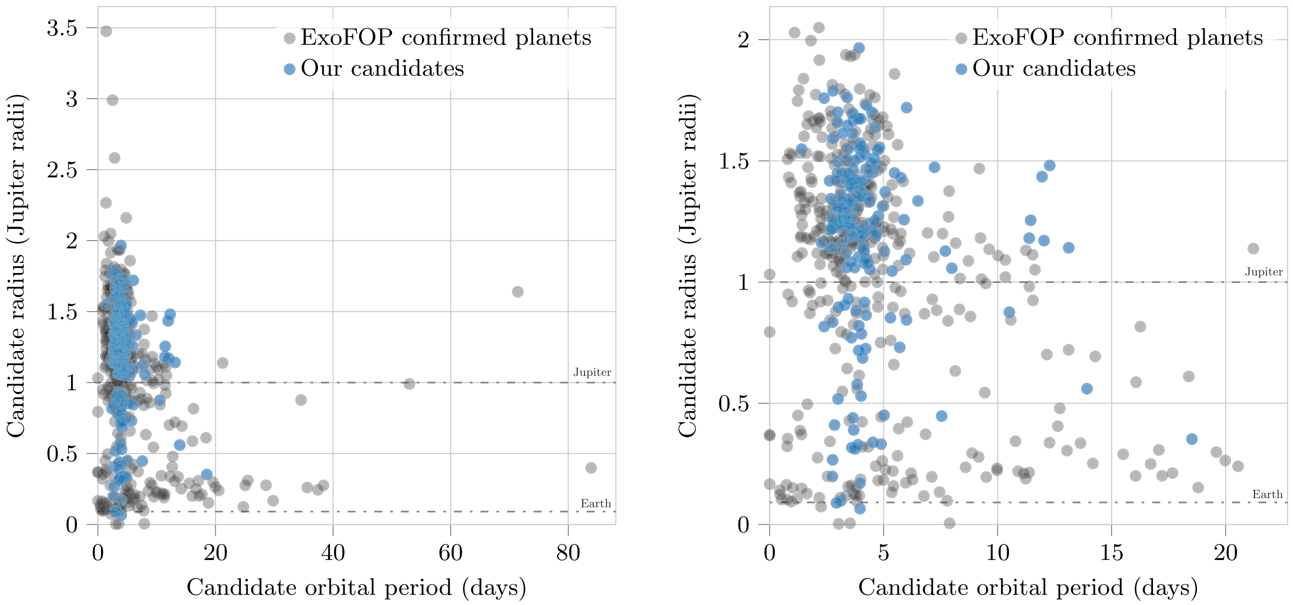


Figure 6. Left: comparison between the distribution of the orbital periods and planetary radii of our planet candidates and ExoFOP-confirmed planets; right: zoomed-in version. The candidates in this graph have their properties displayed in Table 1.

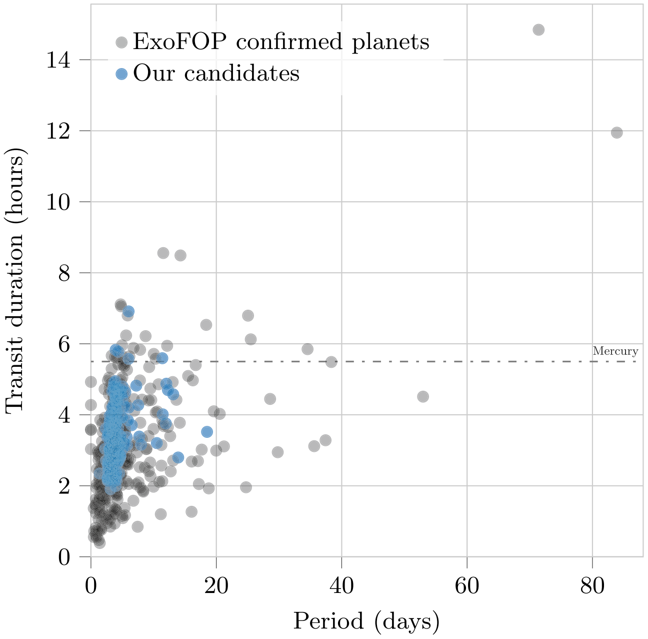


Figure 7. Distribution of the period and transit duration of the planet candidates.

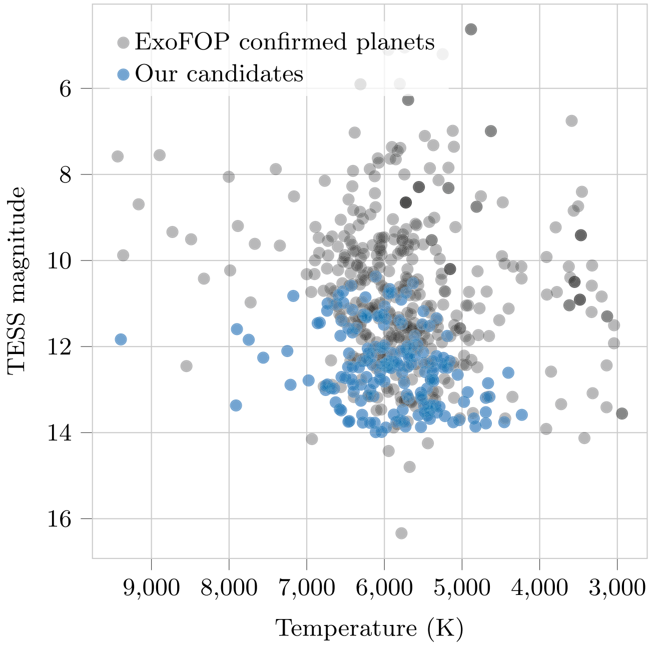


Figure 8. Color–magnitude diagram, showing the effective temperature of the host star vs. its TESS magnitude. Planet candidates from the present article are shown in blue, while those from ExoFOP-TESS are in gray.

the result of data quality rather than network bias. As the FFI LCs include dimmer magnitudes than those of the ExoFOP-TESS-confirmed planets, we might expect the NN to prefer less crowded areas of the sky, where low brightness targets will have a high signal-to-noise ratio. Regardless, the network does not show any significant region omissions compared to the confirmed ExoFOP-TESS planets.

4.3. Comparison with the Expected Candidates

The majority of ExoFOP-TESS-confirmed planets were found using TESS 2-minute-cadence data (Huang et al. 2020). While Barclay et al. (2018) expected the FFI LCs to lead to proportionally higher radius planet discoveries compared to the 2-minute-cadence LCs, we do not see a significant difference in the distributions of our candidates compared to the ExoFOP-TESS-confirmed planets, as shown in Figure 10. This may be in part because the network is trained to find candidates similar to those in the training data set. However, more likely is that larger planets are easier to find and confirm, and most of the existing ExoFOP-TESS-confirmed planets come from the higher end of the expectations of Barclay et al. (2018). A comparison of the distribution of our candidates to the expectations presented by Barclay et al. (2018) is shown in Figure 11.

Barclay et al. (2018) expected that the majority of planets found in TESS FFI data would orbit G- and F-type stars. This is

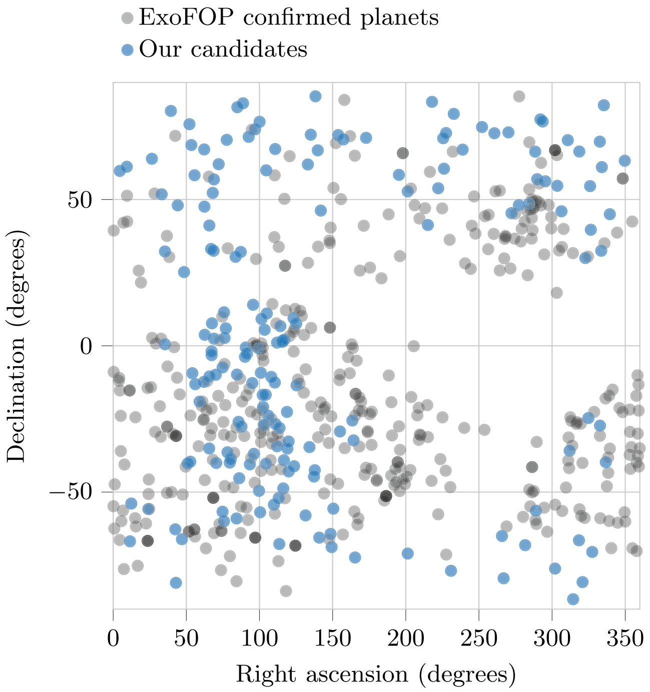


Figure 9. Position on the sky of our candidates compared with the ExoFOP-TESS-confirmed planets.

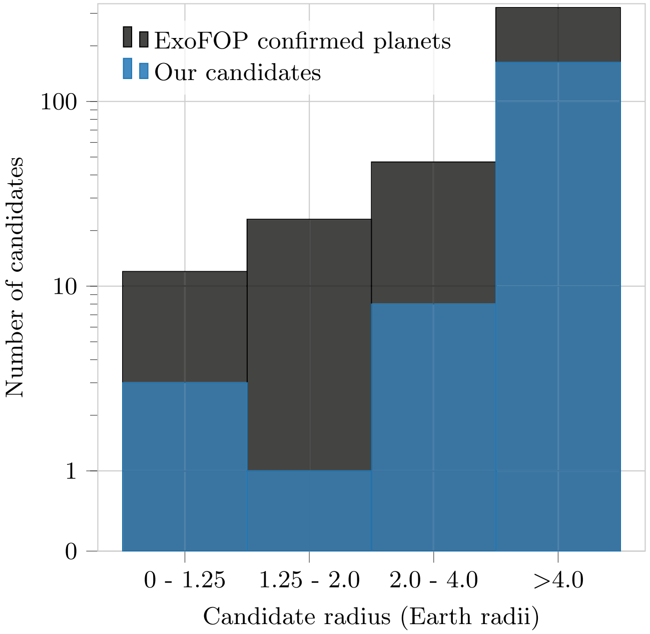


Figure 10. Planet radius distribution of our candidates compared with the confirmed ExoFOP-TESS planets. The data are binned as follows: < 1.25, $1.25 - 2.0$, $2.0 - 4.0$, > 4.0 Earth radii.

consistent with our finding, as shown in Figure 12. As shown in Figure 13, this trend is true for the existing ExoFOP-TESS-confirmed planets as well. Notably, our results contain no M-type star hosts and relatively few A-type star hosts. While planet candidates around these hosts are expected to be relatively rare in TESS data and are relatively rare in the training data, our candidates are disproportionally low in these categories. This disparity is likely primarily caused by these categories having relatively few training examples. The ExoFOP-TESS-confirmed planets with M-type hosts have relatively smaller transit depths and shorter periods, which may explain the dearth of such candidates identified by our network.

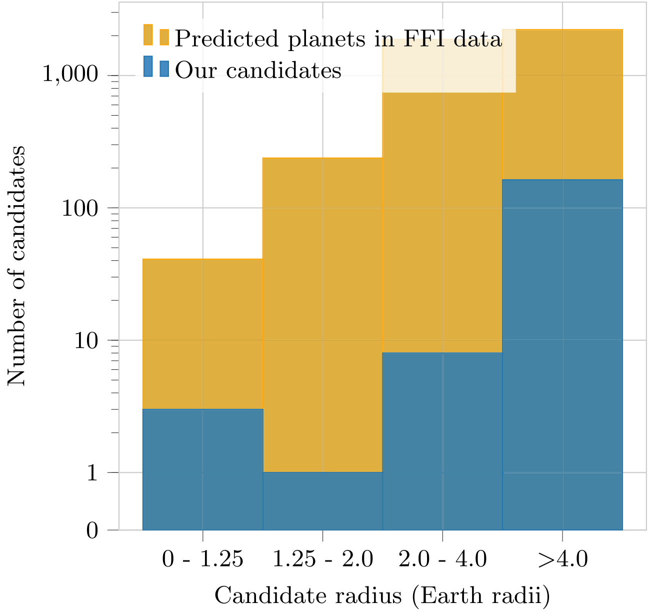


Figure 11. Planet radius distribution of our candidates compared with the expected findings predicted for the FFI data by Barclay et al. (2018). The data are binned as follows: < 1.25, $1.25 - 2.0$, $2.0 - 4.0$, > 4.0 Earth radii.

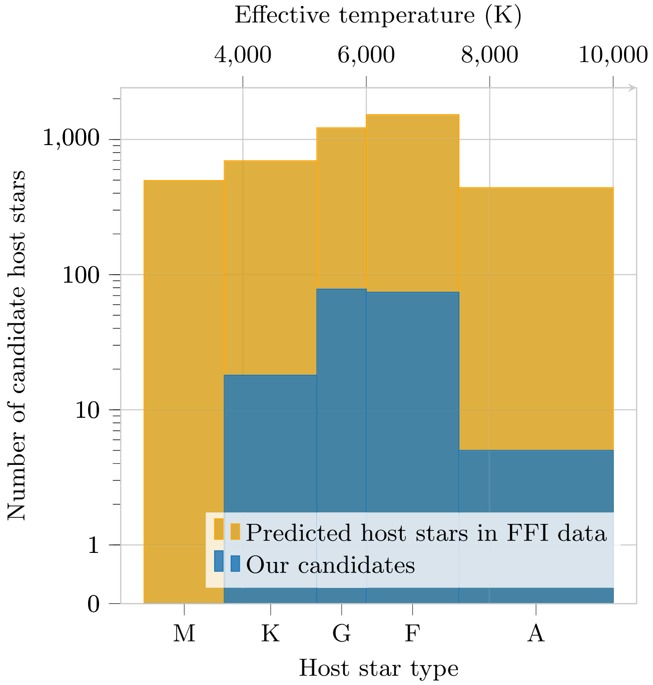


Figure 12. Stellar types of our candidates' hosts compared to the predictions for the FFI data by Barclay et al. (2018). The effective temperature of the stellar host is shown on the x-axis, shown also as spectral type.

Barclay et al. (2018) predict that 80% of planets found in FFI data are expected to orbit stars with radii larger than the Sun. Figure 14 shows the distribution of our candidate host stars, where approximately 74% are larger than the Sun.

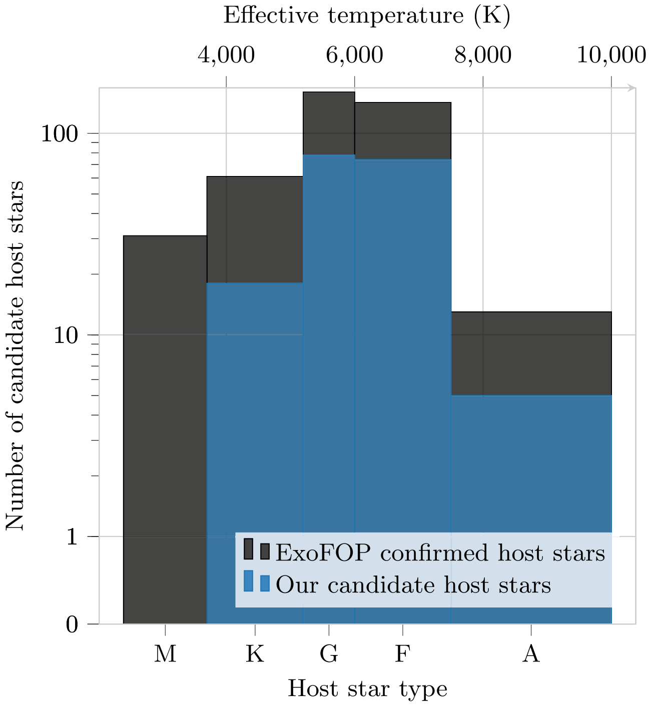


Figure 13. Stellar types of our candidates' hosts compared to the confirmed ExoFOP-TESS planet hosts. The effective temperature of the stellar host is shown on the *x*-axis, shown also as spectral type.

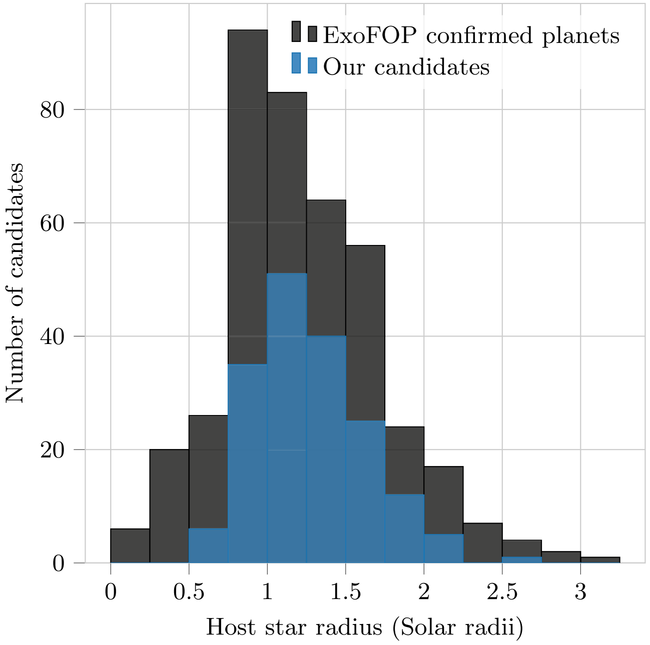


Figure 14. Stellar radii of our candidates' host stars compared to the ExoFOP-confirmed host stars.

5. Conclusion

We present our convolutional NN, which we train to identify planetary transit signals and dismiss false positives. To make a prediction for a given LC, our network requires no prior transit parameters identified using other methods. We train our network using a data set of confirmed exoplanets. Additionally, the network is trained to dismiss eclipsing binaries using a data set of eclipsing binary candidates. We explain several network mechanisms and training techniques used to promote generalization of inference, including a method of injecting LCs into other LCs to create more varied training examples. Our network performs inference on a TESS 30-minute-cadence LC in ∼5 ms on a single GPU, enabling large-scale archival searches. We describe our post-identification analysis used to estimate transiter physical parameters. We present 181 new planet candidates identified by our network, which have passed subsequent human vetting designed to rule out false positives. We provide population analysis of our planet candidates and their host stars compared to a set of confirmed planets and the expected yield from TESS. We provide to the public our NN model as an open-source code for further use and extension.

This paper includes data collected by the TESS mission, which are publicly available from the Mikulski Archive for Space Telescopes (MAST). Funding for the TESS mission is provided by NASA's Science Mission directorate.

This research was supported by an appointment to the NASA Postdoctoral Program at the NASA Goddard Space Flight Center, administered by Universities Space Research Association under contract with NASA.

Resources supporting this work were provided by the NASA High-End Computing (HEC) Program through the NASA Center for Climate Simulation (NCCS) at Goddard Space Flight Center.

This research has made use of the Exoplanet Follow-up Observation Program website, which is operated by the California Institute of Technology, under contract with the National Aeronautics and Space Administration under the Exoplanet Exploration Program.

This work has made use of data from the European Space Agency (ESA) mission Gaia (https://www.cosmos.esa.int/gaia), processed by the Gaia Data Processing and Analysis Consortium (DPAC, https://www.cosmos.esa.int/web/gaia/dpac/consortium).

The material is based on work supported by NASA under award No. 80GSFC17M0002.

Facilities: Gaia, MAST, NCCS, TESS.

Software: Astropy (Astropy Collaboration et al. 2013, 2018), Bokeh (Bokeh Development Team 2020), Eleanor (Feinstein et al. 2019), Keras (Chollet & Others 2015), Lightkurve (Lightkurve Collaboration et al. 2018), Matplotlib (Hunter 2007), NumPy (Harris et al. 2020), Pandas (McKinney 2010), pytest (Krekel et al. 2004), Python (Python Core Team 2020), Tensorflow (Abadi et al. 2015).

ORCID iDs

Greg Olmschenk https://orcid.org/0000-0001-8472-2219
Stela Ishitani Silva https://orcid.org/0000-0003-2267-1246
Gioia Rau https://orcid.org/0000-0002-3042-4539
Richard K. Barry https://orcid.org/0000-0003-4916-0892
Ethan Kruse https://orcid.org/0000-0002-0493-1342
Veselin Kostov https://orcid.org/0000-0001-9786-1031
Brian P. Powell https://orcid.org/0000-0003-0501-2636
Jeremy D. Schnittman https://orcid.org/0000-0002-2942-8399
Thomas Barclay https://orcid.org/0000-0001-7139-2724

References

Abadi, M., Agarwal, A., Barham, P., et al. 2015, TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, https://www.tensorflow.org/

Akeson, R., Chen, X., Ciardi, D., et al. 2013, PASP, 125, 989

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

Astropy Collaboration, Price-Whelan, A. M., Sipőcz, B. M., et al. 2018, AJ, 156, 123
Astropy Collaboration, Robitaille, T. P., Tollerud, E. J., et al. 2013, A&A, 558, A33
Barclay, T., Pepper, J., & Quintana, E. V. 2018, ApJS, 239, 2
Bokeh Development Team 2020, Bokeh: Python Library for Interactive Visualization, https://bokeh.org/
Brown, A., Vallenari, A., Prusti, T., et al. 2018, A&A, 616, A1
Burke, C. J., Levine, A., Fausnaugh, M., et al. 2020, TESS-Point: High precision TESS pointing tool, ascl:2003.001
Carmichael, T. W., Quinn, S. N., Mustill, A. J., et al. 2020, AJ, 160, 53
Chollet, F. & Others 2015, Keras, https://keras.io
Crouzet, N., McCullough, P., Long, D., et al. 2017, AJ, 153, 94
Cybenko, G. 1989, Mathematics of Control, Signals and Systems, 2, 303
Dong, C., Loy, C. C., He, K., & Tang, X. 2014, in European Conf. on Computer Vision, ed. D. Fleet et al. (Berlin: Springer), 184
Eastman, J., Siverd, R., & Gaudi, B. S. 2010, PASP, 122, 935
Feinstein, A. D., Montet, B. T., Foreman-Mackey, D., et al. 2019, PASP, 131, 094502
Foreman-Mackey, D., Agol, E., Ambikasaran, S., & Angus, R. 2017, AJ, 154, 220
Glorot, X., & Bengio, Y. 2010, in Proc. Thirteenth Int. Conf. on Artificial Intelligence and Statistics, ed. Y. W. Teh & M. Titterington (PMLR), 249, http://proceedings.mlr.press/v9/glorot10a/glorot10a.pdf
Glorot, X., Bordes, A., & Bengio, Y. 2011, in Proc. Fourteenth Int. Conf. on Artificial Intelligence and Statistics, ed. G. Gordon, D. Dunson, & M. Dudík (PMLR), 315, http://proceedings.mlr.press/v15/glorot11a/glorot11a.pdf
Harris, C. R., Millman, K. J., van der Walt, S. J., et al. 2020, Natur, 585, 357
Hinton, G., Deng, L., Yu, D., et al. 2012, ISPM, 29, 82
Hochreiter, S. 1998, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 6, 107
Huang, C. X., Vanderburg, A., Pál, A., et al. 2020, RNAAS, 4, 206
Hunter, J. D. 2007, CSE, 9, 90
Ioffe, S., & Szegedy, C. 2015, in Proc. 32nd Int. Conf. on Machine Learning, ed. F. Bach & D. Blei (PMLR), 448, http://proceedings.mlr.press/v37/ioffe15.pdf
Jenkins, J. M., Twicken, J. D., McCauliff, S., et al. 2016, Proc. SPIE, 9913, 99133E
Kingma, D. P., & Ba, J. 2014, arXiv:1412.6980
Kostov, V. B., Mullally, S. E., Quintana, E. V., et al. 2019, AJ, 157, 124
Kovács, G., Zucker, S., & Mazeh, T. 2002, A&A, 391, 369
Krekel, H., Oliveira, B., Pfannschmidt, R., et al. 2004, pytest x.y, https://github.com/pytest-dev/pytest
Krizhevsky, A., Sutskever, I., & Hinton, G. E. 2012, in Advances in Neural Information Processing Systems, ed. F. Pereira et al. (Red Hook, NY: Curran Associates, Inc.), 1097, https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf
Kruse, E., Agol, E., Luger, R., & Foreman-Mackey, D. 2019, ApJS, 244, 11
LeCun, Y., Bengio, Y., & Hinton, G. 2015, Natur, 521, 436
LeCun, Y. A., Bottou, L., Orr, G. B., & Müller, K.-R. 2012, in Neural Networks: Tricks of the Trade, ed. G. Montavon, G. Orr, & K.-R. Müller (Berlin: Springer), 9
Leshno, M., Lin, V. Y., Pinkus, A., & Schocken, S. 1993, NN, 6, 861
Lightkurve Collaboration, Cardoso, J. V. d. M, Hedges, C., et al. 2018, Lightkurve: Kepler and TESS Time Series Analysis in Python, Astrophysics Source Code Library, ascl:1812.013
Luger, R., Agol, E., Kruse, E., et al. 2016, AJ, 152, 100
McKinney, W. 2010, in Proc. 9th Python in Science Conf., ed. S. van der Walt & J. Millman (Austin, TX: SciPy), 56
Mikulski Archive for Space Telescopes 2020, Mikulski Archive for Space Telescopes, https://archive.stsci.edu/missions-and-data/tess
Olmschenk, G., Silva, S. I., Barry, R., & Wyrwas, E. 2021, golmschenk/ramjet: Identifying Planetary Transit Candidates in TESS Full-Frame Image Light Curves via Convolutional Neural Networks, 2021-tess-ffi-planet-candidates-paper, Zenodo, doi:10.5281/zenodo.4460662
Python Core Team 2020, Python: A Dynamic, Open Source Programming Language, Python Software Foundation, https://www.python.org/
Ricker, G. R., Winn, J. N., Vanderspek, R., et al. 2014, JATIS, 1, 014003
Rumelhart, D. E., Hinton, G. E., & Williams, R. J. 1985, in Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations, ed. D. E. Rumelhart & J. L. McClelland (Cambridge, MA: MIT Press), 318
Rumelhart, D. E., Hinton, G. E., & Williams, R. J. 1986, Natur, 323, 533
Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. 2014, JMLR, 15, 1929, https://jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf
Stassun, K. G., Oelkers, R. J., Pepper, J., et al. 2018, AJ, 156, 102
Sullivan, P. W., Winn, J. N., Berta-Thompson, Z. K., et al. 2015, ApJ, 809, 77
Tenenbaum, P., & Jenkins, J. M. 2018, TESS Science Data Products Description Document, Tech. Rep., EXP-TESS-ARC-ICD-0014 Rev D, https://archive.stsci.edu/missions/tess/doc
TESS Follow-up Observing Program Working Group 2020, The Exoplanet Follow-up Observing Program for TESS, https://exofop.ipac.caltech.edu/tess/
Tompson, J., Goroshin, R., Jain, A., LeCun, Y., & Bregler, C. 2015, in Proc. IEEE Conf. on Computer Vision and Pattern Recognition (Piscataway, NJ: IEEE), 648
Wilson, H. R., & Cowan, J. D. 1972, BpJ, 12, 1
Xu, L., Ren, J. S., Liu, C., & Jia, J. 2014, in Advances in Neural Information Processing Systems, ed. Z. Ghahramani et al. (Red Hook, NY: Curran Associates, Inc.), 1790, https://papers.nips.cc/paper/2014/file/1c1d4df596d01da60385f0bb17a4a9e0-Paper.pdf
Zhong, Z., Zheng, L., Kang, G., Li, S., & Yang, Y. 2020, in Proc. AAAI Conf. on Artificial Intelligence 34 (Palo Alto, CA: AAAI), 13001
Zhou, D.-X. 2020, Applied and Computational Harmonic Analysis, 48, 787
Zhou, G., Bakos, G., Hartman, J. D., et al. 2017, AJ, 153, 211