

Distinguishing a planetary transit from false positives: a Transformer-based classification for planetary transit signals

Helem Salinas,^{1*} Karim Pichara,^{1†} Rafael Brahm,^{2,3‡} Francisco Pérez-Galarce,^{1§} and Domingo Mery,^{1¶}

¹Departamento de Ciencias Computacionales, Facultad de Ingeniería, Pontificia Universidad Católica de Chile, Santiago, Chile

²Facultad de Ingeniería y Ciencias, Universidad Adolfo Ibáñez, Av. Diagonal las Torres 2640, Peñalolén, Santiago, Chile

³Millennium Institute for Astrophysics, Chile

Accepted 2023 April 12. Received 2023 March 24; in original form 2022 November 30

ABSTRACT

Current space-based missions, such as the Transiting Exoplanet Survey Satellite (TESS), provide a large database of light curves that must be analysed efficiently and systematically. In recent years, deep learning (DL) methods, particularly convolutional neural networks (CNN), have been used to classify transit signals of candidate exoplanets automatically. However, CNNs have some drawbacks; for example, they require many layers to capture dependencies on sequential data, such as light curves, making the network so large that it eventually becomes impractical. The self-attention mechanism is a DL technique that attempts to mimic the action of selectively focusing on some relevant things while ignoring others. Models, such as the Transformer architecture, were recently proposed for sequential data with successful results. Based on these successful models, we present a new architecture for the automatic classification of transit signals. Our proposed architecture is designed to capture the most significant features of a transit signal and stellar parameters through the self-attention mechanism. In addition to model prediction, we take advantage of attention map inspection, obtaining a more interpretable DL approach. Thus, we can identify the relevance of each element to differentiate a transit signal from false positives, simplifying the manual examination of candidates. We show that our architecture achieves competitive results concerning the CNNs applied for recognizing exoplanetary transit signals in data from the TESS telescope. Based on these results, we demonstrate that applying this state-of-the-art DL model to light curves can be a powerful technique for transit signal detection while offering a level of interpretability.

Key words: methods: data analysis, planets and satellites: detection

1 INTRODUCTION

For years, over a million stars have been observed to detect extra-solar planets through a variety of techniques. One of the most successful methods to detect exoplanets is the transit method, which consists of monitoring the flux of stars in the search for periodic dips in brightness. This method considers that those dips could be produced by planets blocking a portion of the host star, as seen by the observer. The CoRoT (Convection, Rotation and planetary Transits) mission (Auvergne et al. 2009), was the first space-based telescope to detect exoplanets using the transit method. Perhaps the most notable contributor is NASA’s Kepler space telescope (Borucki et al. 2010), which was launched in 2009. Kepler identified nearly 4,000 potential planet candidates, out of which \sim 2,600 have been confirmed as exoplanets to date. Because Kepler could no longer observe its original field, an extended version of Kepler, K2, was designed (Howell et al. 2014). K2 identified \sim 500 new planets. From the full sample

of 5,200 discovered planets¹, most have been identified as candidates with the transit method using space-based missions (Borucki et al. 2010; Batalha et al. 2013; Burke et al. 2014; Rowe et al. 2015; Mullally et al. 2015; Coughlin et al. 2016; Morton et al. 2016), and confirmed as exoplanets with radial velocity measurements, the analysis of transit timing variations, and/or through statistical validation (Thompson et al. 2018).

Kepler’s successor, NASA’s Transiting Exoplanet Survey Satellite (TESS; Ricker et al. 2014) that was launched in April 2018, and a future mission, The PLAnetary Transits and Oscillations of stars (PLATO; Rauer et al. 2014) mission, are telescopes whose main objective is the detection of exoplanet candidates. TESS is monitoring millions of stars with high photometric precision to detect signals of transiting extrasolar planets. Unlike Kepler, TESS is surveying a large fraction of the sky (approximately 90% of it), searching for planets around the nearest and brightest stars. The size of each TESS sector is approximately 2300 square degrees, whereas Kepler had a field of view of approximately 115 square degrees. TESS observes a sector for 27 days, and then the spacecraft is reoriented to observe the next sector. The main objective of TESS is to find hundreds of planets

* E-mail: yhsalinas@uc.cl

† E-mail: kpb@ing.puc.cl

‡ E-mail: rafael.brahm@uai.cl

§ E-mail: fjperez10@uc.cl

¶ E-mail: domingo.mery@uc.cl

¹ <https://exoplanetarchive.ipac.caltech.edu/index.html>

in transit smaller than Neptune, including dozens that are comparable in size to Earth and amenable for atmospheric characterization (Ricker et al. 2015).

The identification of transiting exoplanet candidates from photometric time series typically consists of a two-step process. The first step involves the removal of long-term flux variations of stellar and/or instrumental origin, followed by the identification of periodic transit-like signals by phase folding the light curve to different orbital periods, times of transit, and transit durations (e.g. Tenenbaum et al. 2012). The products of this process are known as Threshold Crossing Events (TCEs), which can be exoplanet candidates but also false positives. Some phenomena can mimic a transit signal; among these impostors are eclipsing binaries, instrumental artifacts, noise, and stellar variability. The second step for identifying exoplanet candidates consists of the vetting of the TCEs, by rejecting false positives based on the properties of the phase-folded transiting light curves, stellar properties, and/or follow-up observations. This process is labor intensive and sensitive to human error, and increasingly requires a significant amount of effort since there are large volumes of data to be examined. Thus, identifying exoplanet transit signals has become an increasingly data-science driven task. In response to this challenge, several efforts have been made to classify candidates automatically from space-based and ground-based surveys.

In the era of DL (LeCun et al. 2015; Sun et al. 2017), neural network (NN) architectures are applicable in almost all areas of scientific research, producing great advances in image processing and classification, facial recognition, voice recognition and Natural Language Processing (NLP) (Krizhevsky et al. 2017; Ciregan et al. 2012; He et al. 2016; Voulovodimos et al. 2018; Vaswani et al. 2017). Furthermore, the area of DL and many areas in astronomy largely share the philosophy of data-driven study and analysis. In particular, DL techniques have been applied for the search and examination of exoplanet candidates. One of the most outstanding applications in this area has been the use of CNNs, firstly explored by Pearson et al. (2018), Shallue & Vanderburg (2018) and Schanche et al. (2019). These approaches developed CNNs trained with data from Kepler and from the Wide Angle Search for Planets (WASP; Pollacco et al. 2006) respectively, achieving successful results in the identification of exoplanetary signals. Furthermore, they have demonstrated that the incorporation of additional information to a CNN model, e.g., the information of the flux centroid pixel position, considerably improves their performance (Exonet; Ansdell et al. 2018; Osborn et al. 2020). The centroids represent the pixel position of the centre of the light. This information allows us to know the location of the transits in the pixels. In general, experts perform this analysis to verify whether the transit signals are the product of sources other than an exoplanetary transit signal (Shallue & Vanderburg 2018).

Even though current approaches have reached a high level of accuracy, the results are not interpretable, i.e., it is not possible to understand why the network classifies a given system as a transiting planet or as a false positive. Interpreting the predictions of a model can lead to a better understanding of the model and to a more efficient use of its outputs. Furthermore, interpreting the result of a model prediction generates a greater value by knowing the importance of certain features, in such a way that they can be later justified by experts. This could help future prediction analysis in differentiating the characteristic features of a planetary transit signal from those of a false positive. Unfortunately, the explainability of models such as CNNs is low because convolutional filters are difficult to understand, limiting the interpretability of the hidden representation in the network (Zhang et al. 2019; Vilone & Longo 2020).

The application of one-dimensional CNNs on time series data has

had good results. However, when working with this type of data, CNNs require many layers to capture dependencies Hawkins (2004); Lakew et al. (2018). Typically a layer is a container that receives a weighted input. In fact, building models capable of learning long-range dependencies is challenging for the majority of sequential modeling tasks. In the case of classification of transiting exoplanet signals, the state-of-the-art DL approaches propose a global and a local view of the transit signal as input to the CNN model (Shallue & Vanderburg 2018). The light curve of the “global view” shows the complete phase-folded light curve, i.e., the light curve over an entire orbital period. The “local view” is centered around the transit. Regarding the length of both sequences, the length of the global view has more than twice the number of observations than the local view. In that sense, the NNs for the global view of a planetary transit have many more layers compared to the local view. This is because it is a longer sequence and requires a greater network depth to capture dependencies. Although the layers of a CNN work in parallel, the length and dimensionality of the sequential data could generate some computational limitations in this type of network (Vaswani et al. 2017).

Attention mechanisms were proposed by the NLP community to overcome the computational problems of CNNs and Recurrent neural networks (RNNs; Cho et al. 2014). The Transformer architecture, which is mainly based on self-attention mechanisms (Vaswani et al. 2017), have gained popularity for its performance on sequentially structured data. This type of architecture is capable of working well in complex scenarios for other kinds of models (RNN, CNN), such as sequences (Hawkins 2004; Lakew et al. 2018; Karita et al. 2019). In general, time series data have more long-range correlated information. For example, a planetary transit signal depends on duration and period, which causes a correlation in light curve observations. In this sense, attention mechanisms can capture this information better.

In this paper, we propose a new deep neural network based on the Transformer architecture for the automatic classification of exoplanet transit signals. Unlike previous DL models for recognizing exoplanet signals, we provide a context and interpretability of the model output that would help to better understand the patterns of each signal. Thus, we propose a model that provides that interpretability when classifying a signal as a exoplanet transit or as a false positive signal, in turn, showing the characteristics that differentiate both classes. We include the measurements of the flux centroid pixel positions to improve the representation of an exoplanetary transit signal. Furthermore, we show the relevance of using centroids in terms of interpretability and how it affects the final prediction of the model. We show that the input representation is essential for a model to understand the importance of specific characteristics of a transit signal in a light curve. We also take into account that the transits have temporal information (orbital period, transit duration). Specifically, for the order of the sequence of observations, we injected time information from the light curves into the positional encoding of the input representations. Then, we generate a more representative input of the flux and its centroid pixel position. In addition to the information on the centroids, we include information on the parameters of the transit and the host star like Ansdell et al. (2018), Osborn et al. (2020) and Rao et al. (2021). Finally, we identify the most important parameters to differentiate an exoplanet transit from a false positive according to the self-attention mechanism.

This paper is organized as follows. Section 2 introduces the main concepts that inspired this work, where we briefly describe the self-attention mechanisms related to the original architecture of the Transformer (Vaswani et al. 2017) in order to understand our proposal for planetary transit signal recognition. Section 3, gives an account of

previous work, which is divided into two parts. First, we present a review of machine learning models for exoplanetary transit signal recognition. In the second part of the review, we show how the attention mechanism has been applied in astronomy. Section 4 explains the proposed model in detail. Section 5 describes the data sets used. The training process and performance evaluation are described in Section 6. Section 7 shows our classification result obtained using our model. Finally, in Section 8 we present overall conclusions and future work.

2 BACKGROUND THEORY

In this section, we briefly describe the main components and concepts to understand our method. First, in Section 2.1, we introduce the attention mechanism. Then, in Section 2.2 we give way to the description of the principal core of the Transformer architecture (Vaswani et al. 2017), which is the self-attention mechanism, explaining, in turn, each component of this architecture that inspired this work. Lastly, in Section 2.3, we describe the positional assignment mechanism of each element in a sequence.

2.1 Attention mechanisms

DL, a typical branch of machine learning methods, is a set of computational heuristics to train parametric functions structured in a layered form to perform a certain task (LeCun et al. 2015; Goodfellow et al. 2016), and that has been successful in classification problems from a variety of application domains. The architecture relies on several layers of NNs of simple interconnected units and uses layers to build increasingly complex and useful features by means of linear and non-linear transformation. This family of models is capable of generating increasingly high-level representations (LeCun et al. 2015).

Attention theory is one of the most prominent ideas in the DL community, which emerged from the applications of NLP. Typically, NNs are inspired by different actions and processes of the human brain. Consequently, the attention mechanism also attempts to mimic the action of selectively focusing on some relevant things while ignoring others. Human beings tend to pay attention to ideas, things, language, etc., when carrying out specific tasks or solving problems. In NLP, the attention mechanism emerged as an improvement over the sequence modelling encoder-decoder-based architecture for the translation system Bahdanau et al. (2014).

In astronomy, light curves are sequences as a function of time, where each observation in the sequence is related to the observations that precede or follow it. For example, in the case of light curves of transiting planets these sequences have observations based on an order according to orbital period and transit duration, similar to the structure of sentences in a paragraph. In NLP, each word is an element of the sequence. In light curves, each observation represents an element of the sequence. So, through the attention mechanism, we can selectively focus on some observations that characterize an astronomical event while ignoring others, for example, observations that do not contain any relevant event. This mechanism and its variants have been used not only in modeling sequences but also in computer vision (Dosovitskiy et al. 2020; Wu et al. 2020; Li et al. 2020), speech processing (Karita et al. 2019; Dong et al. 2018) and other areas within DL (Han et al. 2020; Huang et al. 2018).

The traditional attention mechanism was designed to connect the encoder with the decoder, feeding the decoder with information on each hidden state of the encoder. This piece allows comparing the input and output of the phrase, an image, or a previous step. With

sequence modeling, it is possible to build models that take as input a vector of symbols, also named vector embedding (e.g., representation of words, images, numbers, etc.) (x_1, x_2, \dots, x_n) , where n is the sequence length. The encoder transforms this input into a continuous vector z , i.e., vectorizes the input sequence. Then, this is the input to the decoder, which attends to the encoded representation and produces the output sequence (y_1, y_2, \dots, y_n) of symbols. The output type of the prediction depends on the specific task. For example, the output type is a discrete scalar for binary or multi-class classification. In the case of a regression task, this output is a continuous value.

This mechanism contrasts with other architectures, such as RNNs, which process data sequentially. In other words, the RNNs process element by element to access the content of the last input element of the time series. This type of process tends to make the network forget the information from positions that are further away or, in certain instances, merge the information from adjacent positions across the time series (Lakew et al. 2018). In the case of CNN's, they learn by batch with parallelism across layers, being fast and effective in capturing dependencies across the time series. However, in long sequences, capturing the dependencies between different combinations of input elements requires deeper NNs, which can be cumbersome and impractical (Hawkins 2004).

2.2 Self-attention

From the theory of attention mechanism, a new form of attention emerged, called self-attention, and with it, a new class of models, which are mainly focused on sequential data. In contrast to the traditional attention mechanism, which attends to different parts of the input or output sequences while making the predictions, the self-attention mechanism allows each element of the input sequence to interact with each other while encoding each of the input elements.

Self-attention, also referred as intra-attention is designed to find important relationships between each input element (Vaswani et al. 2017). This allows the network to find which elements it should “pay attention” to, similar to the way a human being understands words in a sentence; for example, it complements the subject of a sentence with a verb or adjective. Based on this idea, in light curve time series, the self-attention allows the model to learn the correlation between the current observations and the other observations in the same sequence. For example, a transit signal contains observations, such as noise, some star variation, or brightness dimming, which could be highly correlated with other values in the same sequence. This information could be essential to determine if a signal belongs to an exoplanet. Furthermore, the self-attention quantifies the relationships between the observations without the condition that the operations follow a sequential order. This is beneficial so that operations can be executed in parallel.

The aim of self-attention is to compute the similarities of each input within the sequence with each other. It is computed using three vectors, Q , K and V , referred to as query, key, and value, respectively. Vectors Q and K are compared to calculate similarities and obtain scores/weights for vector V . To compute these vectors as three different subspace representations, the input sequence is projected using three matrices \mathbf{W}^Q , \mathbf{W}^K and \mathbf{W}^V , which are initialized weight matrices with independent random values with mean 0 and variance 1 by definition. Given a light curve, which is the input sequence and is defined as $X \in \mathbb{R}^{n \times d}$, where n is the number of observations of the input sequence, and d is the input dimensionality and the three

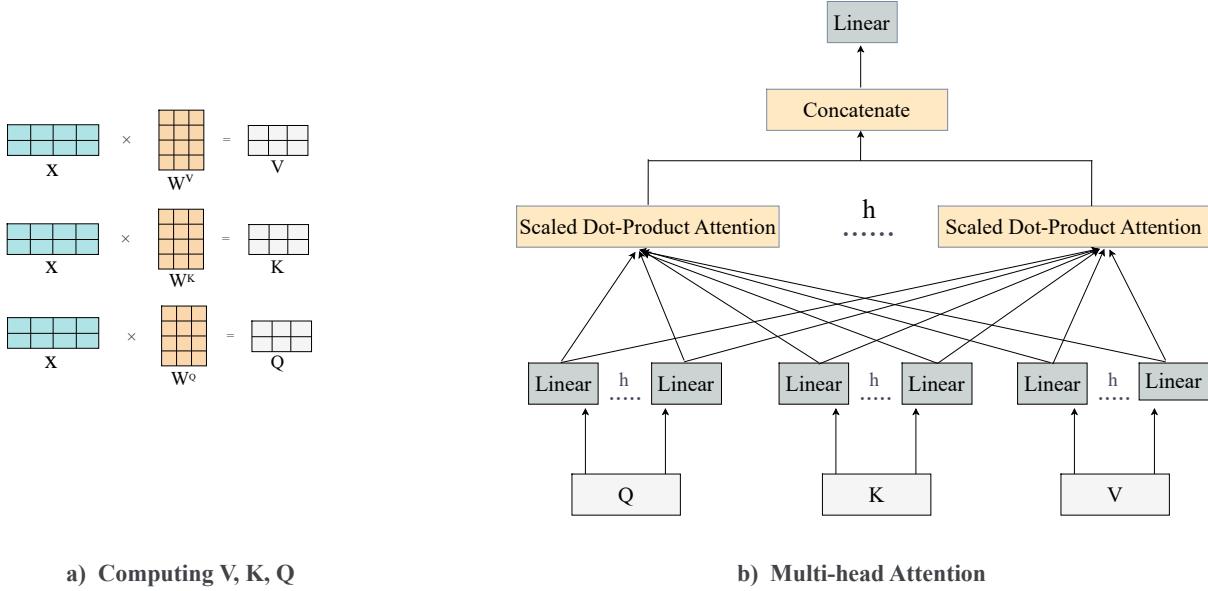


Figure 1. Description of multihead attention mechanism defined in (Vaswani et al. 2017). a) Multiplication of the input sequence X by the weight matrix \mathbf{W}^V , \mathbf{W}^K and \mathbf{W}^Q to produce \mathbf{V} , \mathbf{K} and \mathbf{Q} respectively. b) Multihead attention, where values \mathbf{V} , keys \mathbf{K} , and queries \mathbf{Q} are linearly projected h times with different learned linear projections. Then, multihead attention component generates h dimensional output values in parallel. Finally, these are concatenated and projected, resulting in the final values.

weight matrices \mathbf{W}^Q , \mathbf{W}^K and \mathbf{W}^V , the Q , K and V are computed as:

$$\begin{aligned} \mathbf{Q} &= \mathbf{X}\mathbf{W}^Q \in \mathbb{R}^{n \times d_q}, \\ \mathbf{K} &= \mathbf{X}\mathbf{W}^K \in \mathbb{R}^{n \times d_k}, \\ \mathbf{V} &= \mathbf{X}\mathbf{W}^V \in \mathbb{R}^{n \times d_v}. \end{aligned} \quad (1)$$

Figure 1(a) shows this process, where the first step is multiplying each of the encoder input sequence X by three weights matrices (\mathbf{W}^Q , \mathbf{W}^K and \mathbf{W}^V) that we trained during the training process. We then take the \mathbf{Q} , \mathbf{K} and \mathbf{V} matrices, and supply them as input to the attention function. A self-attention matrix $\mathbf{A} \in \mathbb{R}^{n \times d_v}$ is often referred to as the scaled dot-product attention and can be written as:

$$\mathbf{A} = \text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax} \left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}} \right) \mathbf{V} \quad (2)$$

The attention layer takes the input in the form of those three parameters (after positional encoding). The three parameters have similar structure, where each element in the sequence is represented by a vector.

2.2.1 Multihead attention

In the Transformer architecture, the attention module consists of several self-attention heads running in parallel. This module is also called multihead attention, which allows the neural network to learn from h different representation subspaces of Q , K , and V simultaneously. This process, in turn, generates a richer representation for the model.

In multihead attention (see Figure 1(b)), the key, values, and

queries are linearly projected h times, i.e., we can repeat its computations multiple times in parallel (Vaswani et al. 2017). Then, those results are concatenated, and we obtain the final values:

$$\text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathbf{W}^O \begin{bmatrix} h_1 \\ \vdots \\ h_h \end{bmatrix} \quad (3)$$

with learned output weights $\mathbf{W}^O \in \mathbb{R}^{hd_v \times d}$, and where:

$$h_i = \text{Attention}(\mathbf{X}\mathbf{W}_i^Q, \mathbf{X}\mathbf{W}_i^K, \mathbf{X}\mathbf{W}_i^V) \quad (4)$$

with $h_i \in \mathbb{R}^{n \times d_v}$. Then, the final result of this linear transformation, Equation (3), is the multi-headed attention matrix with dimension $n \times d$. In an attention with $h = 1$, the network learns from a single behavior of the information. On the contrary, the use of multiple heads of attention allows the network to learn from a final representation richer in information since each “head” of attention is potentially focusing on different parts of the input sequence. To determine the appropriate number of heads, we must analyse how the model captures the dependencies along the sequence. For this, we analysed the individual attention of each head through attention entropy. In this way, we can observe how attention weights are concentrated on specific observations (see Section 7.2). Choosing the number of heads is analogous to the number of kernels in a convolutional layer.

2.3 Positional encoding

The Transformer architecture avoids sequential operations in favor of a multihead self-attention mechanism with parallel computation. The positional encoding describes the location or position of a token

in a sequence, such that each position is assigned to a unique representation. The idea of injecting a position to the sequence was first introduced in Gehring et al. (2017), where it was used in the context of sequence modeling with CNNs. In the original architecture of the Transformer, the process of positional embedding is a new way of representing the values of a word. Vaswani et al. (2017) implement a positional encoding from a vector of indices and adds a trigonometric function:

$$\text{PE}_{t,i} := \begin{cases} \sin(p, \omega_i), & \text{if } p = 2i \\ \cos(p, \omega_i), & \text{if } p = 2i + 1 \end{cases} \quad (5)$$

where p represents the position of an object in the input sequence, i that is used for mapping to indices of the positional encoding and ω_i represents the frequency of progression, which is computed as:

$$\omega_i = \frac{1}{10000^{2i/d}}. \quad (6)$$

where d is the dimension of the output embedding space.

3 RELATED WORK

In this section, we describe the previous works related to our proposal, which are divided into two parts. In Section 3.1, we present a review of approaches based on machine learning techniques for the detection of planetary transit signals. Section 3.2 provides an account of the approaches based on attention mechanisms applied in Astronomy.

3.1 Exoplanet detection

Machine learning methods have achieved great performance for the automatic selection of exoplanet transit signals. One of the earliest applications of machine learning is a model named AutoVetter (McCauliff et al. 2015), which is a random forest (RF) model based on characteristics derived from Kepler pipeline statistics to classify exoplanet and false positive signals. Then, other studies emerged that also used supervised learning. Mislis et al. (2016) also used a RF, but unlike the work by McCauliff et al. (2015), they used simulated light curves and a box least square (BLS; Kovács et al. 2002)-based periodogram to search for transiting exoplanets. Thompson et al. (2015) proposed a k-nearest neighbors model for Kepler data to determine if a given signal has similarity to known transits. Unsupervised learning techniques were also applied, such as self-organizing maps (SOM), proposed Armstrong et al. (2016); which implements an architecture to segment similar light curves. In the same way, Armstrong et al. (2018) developed a combination of supervised and unsupervised learning, including RF and SOM models. In general, these approaches require a previous phase of feature engineering for each light curve.

The application of DL for exoplanetary signal detection has evolved rapidly in recent years and has become very popular in planetary science. Pearson et al. (2018) and Zucker & Giryes (2018) developed CNN-based algorithms that learn from synthetic data to search for exoplanets. Perhaps one of the most successful applications of the DL models in transit detection was that of Shallue & Vanderburg (2018); who, in collaboration with Google, proposed a CNN named AstroNet that recognizes exoplanet signals in real data from Kepler. AstroNet uses the training set of labelled TCEs from the AutoVetter planet candidate catalog of Q1–Q17 data release 24 (DR24) of the Kepler mission (Catzanzarite 2015). AstroNet analyses the data in two views: a “global view”, and “local view” (Shallue & Vanderburg 2018).

Based on AstroNet, researchers have modified the original AstroNet model to rank candidates from different surveys, specifically for Kepler and TESS missions. Ansdel et al. (2018) developed a CNN trained on Kepler data, and included for the first time the information on the centroids, showing that the model improves performance considerably. Then, Osborn et al. (2020) and Yu et al. (2019) also included the centroids information, but in addition, Osborn et al. (2020) included information of the stellar and transit parameters. Finally, Rao et al. (2021) proposed a pipeline that includes a new “half-phase” view of the transit signal. This half-phase view represents a transit view with a different time and phase. The purpose of this view is to recover any possible secondary eclipse (the object hiding behind the disk of the primary star).

3.2 Attention mechanisms in Astronomy

Despite the remarkable success of attention mechanisms in sequential data, few papers have exploited their advantages in astronomy. In particular, there are no models based on attention mechanisms for detecting planets. Below we present a summary of the main applications of this modeling approach to astronomy, based on two points of view; performance and interpretability of the model.

The application of attention mechanisms has shown improvements in the performance of some regression and classification tasks compared to previous approaches. One of the first implementations of the attention mechanism was to find gravitational lenses proposed by Thuruthipilly et al. (2021). They designed 21 self-attention-based encoder models, where each model was trained separately with 18,000 simulated images, demonstrating that the model based on the Transformer has a better performance and uses fewer trainable parameters compared to CNN. A novel application was proposed by Lin et al. (2021) for the morphological classification of galaxies, who used an architecture derived from the Transformer, named Vision Transformer (ViT) (Dosovitskiy et al. 2020). Lin et al. (2021) demonstrated competitive results compared to CNNs. Another application with successful results was proposed by Zerveas et al. (2021); which first proposed a transformer-based framework for learning unsupervised representations of multivariate time series. Their methodology takes advantage of unlabeled data to train an encoder and extract dense vector representations of time series. Subsequently, they evaluate the model for regression and classification tasks, demonstrating better performance than other state-of-the-art supervised methods, even with data sets with limited samples.

Regarding the interpretability of the model, a recent contribution that analyses the attention maps was presented by Bowles et al. (2021), which explored the use of group-equivariant self-attention for radio astronomy classification. Compared to other approaches, this model analysed the attention maps of the predictions and showed that the mechanism extracts the brightest spots and jets of the radio source more clearly. This indicates that attention maps for prediction interpretation could help experts see patterns that the human eye often misses.

In the field of variable stars, Allam Jr & McEwen (2021) employed the mechanism for classifying multivariate time series in variable stars. And additionally, Allam Jr & McEwen (2021) showed that the activation weights are accommodated according to the variation in brightness of the star, achieving a more interpretable model. And finally, related to the TESS telescope, Morvan et al. (2022) proposed a model that removes the noise from the light curves through the distribution of attention weights. Morvan et al. (2022) showed that the use of the attention mechanism is excellent for removing noise and outliers in time series datasets compared with others approaches.

6 *Helem Salinas*

In addition, the use of attention maps allowed them to show the representations learned from the model.

Recent attention mechanism approaches in astronomy demonstrate comparable results with earlier approaches, such as CNNs. At the same time, they offer interpretability of their results, which allows a post-prediction analysis.

4 METHOD DESCRIPTION

In this section, we describe our methodology, which is able to identify if a transit signal belongs to an exoplanet or a false positive. Our work is mainly motivated by the classification of transiting exoplanet candidates originated from the TESS mission. The proposed architecture considers the exoplanetary transit signal, centroid information, and stellar and transit parameters. Our method consists of three main steps, which we describe in this section. First, in Section 4.1, we explain how the input representation is constructed. Then, Section 4.2 shows the design of our proposed model. Finally, Section 4.3.1 describes the methodology we adopted to analyse the resulting attention matrices.

4.1 Data preprocessing

A light curve is defined by flux observations taken over a given time. We process light curves belonging to TCEs from various TESS sectors². In the case of transit signals, we consider the “local” and “global” views (similar to [Shallue & Vanderburg \(2018\)](#)), both of which depend on the duration of the transit. Then, we include the information of the centroids in the input representation, and we observe the behavior of the attention weights on the main characteristics that differentiate a planetary transit from a false positive. Additionally, we include information on the stellar and transit parameters, and we show the importance of each parameter to distinguish a transit.

4.1.1 Flux and centroid time series

We perform some additional steps to prepare the light curves with a method similar to [Shallue & Vanderburg \(2018\)](#), [Ansdel et al. \(2018\)](#), [Osborn et al. \(2020\)](#), [Yu et al. \(2019\)](#) and [Rao et al. \(2021\)](#). We use the Pre-Search Data Conditioning Simple Photometry flux (PDCSAP_FLUX) that has the instrumental systematic removed. Within the processing of the light curves, we remove outliers if their flux values are at least 5σ above the median of the light curve. Then, we remove the low frequency trend of light curve applying the Savitzky–Golay filter using the `flatten()` function from the Lightkurve toolkit ([Lightkurve Collaboration et al. 2018](#)). Then, we process each light curve to bin the phase-folded local and global views similar to [Shallue & Vanderburg \(2018\)](#). For all processing procedures, we use the Lightkurve³ and Astropy API⁴. For phase-folding the local and global views, we take the transit ephemeris available in the ExoFOP-TESS⁵ catalogue. For the local view, which depends on transit duration t_{dur} , we zoomed in on the transit between $-2.0 \cdot t_{dur}$ to $2.0 \cdot t_{dur}$. In the case of the global view, it corresponds to the full view of the light curve with a phase between -0.5 to 0.5 .

² https://archive.stsci.edu/tess/bulk_downloads.html#catalogs

³ <https://docs.lightkurve.org/index.html>

⁴ `astropy.stats.sigma_clipping`

⁵ <https://exofop.ipac.caltech.edu/tess/>

In addition to the PDCSAP_FLUX, we included the centroid information into our input. Incorporating this information was applied by [Ansdel et al. \(2018\)](#); [Osborn et al. \(2020\)](#). We used the CCD row and column position information of the target centroid using a Point Spread Function (PSF) model that is available in the PDC files from the TESS Input Catalog. And if this information is not available in some files, we used the CCD row and column position of the flux-weighted centroid of the target (MOM centroids), that are less robust against background noise compared to centroids using a PSF model. Having data originated from different distributions or samples is very common in machine learning, and one way to transform the data to comparable scales is by drawing on standardization techniques. Based on this, we standardize the values by removing the mean and scaling to unit variance for both centroids.

4.1.2 Stellar and transit parameters

[Ansdel et al. \(2018\)](#) show that providing stellar and transit parameters can improve the model performance for the recognition of a planetary signal. Unlike [Osborn et al. \(2020\)](#) and [Rao et al. \(2021\)](#), we prefer to take only some stellar and transit parameters since, in many cases, much of the information was not available or had null values. The parameters we process are the metallicity [Fe/H], temperature T_{eff} and magnitude $TESS_{Mag}$ of the host star. Regarding the transit parameters, we consider the period, duration and depth, which were taken from the TCE statistics.

4.2 Model

The core of our architecture is the self-attention mechanism, but it also contains other important components. These components are: i) input embedding, which consists of the representation of the input, ii) positional encoding, iii) multihead self-attention, iv) pooling layers, which are designed to greatly reduce the number of input features. This process is called downsampling, v) linear layer, also known as fully-connected layers, in which each neuron applies a linear transformation to the input vector.

Our proposed architecture is shown in Figure 2, which contains the necessary encoder modules to incorporate the flux, centroid information (local and global view), stellar/transit parameters, and time of folded light curves with a particular period. In contrast to the original architecture by [Vaswani et al. \(2017\)](#), we consider the decoder as the module with the activation function for a binary classification. Additionally, we implement three encoders. The first encoder is for the local view light curve, the second one for the global view, and the third one for the stellar/transit parameters. We transform the information of the light curves (local and global view) into a vector representation through a convolutional layer. A positional encoding is added to this new representation, this representation is the input to the multihead attention block. The output of this block is then passed through a new max pooling layer. In the case of the third encoder, a linear layer transforms the input that corresponds to the stellar and transit parameters, and is the input to the multihead attention block. Finally, we concatenate each output of each block and pass them as input to a linear layer with softmax for class prediction. In the rest of this section, we describe each of the components of our architecture in detail.

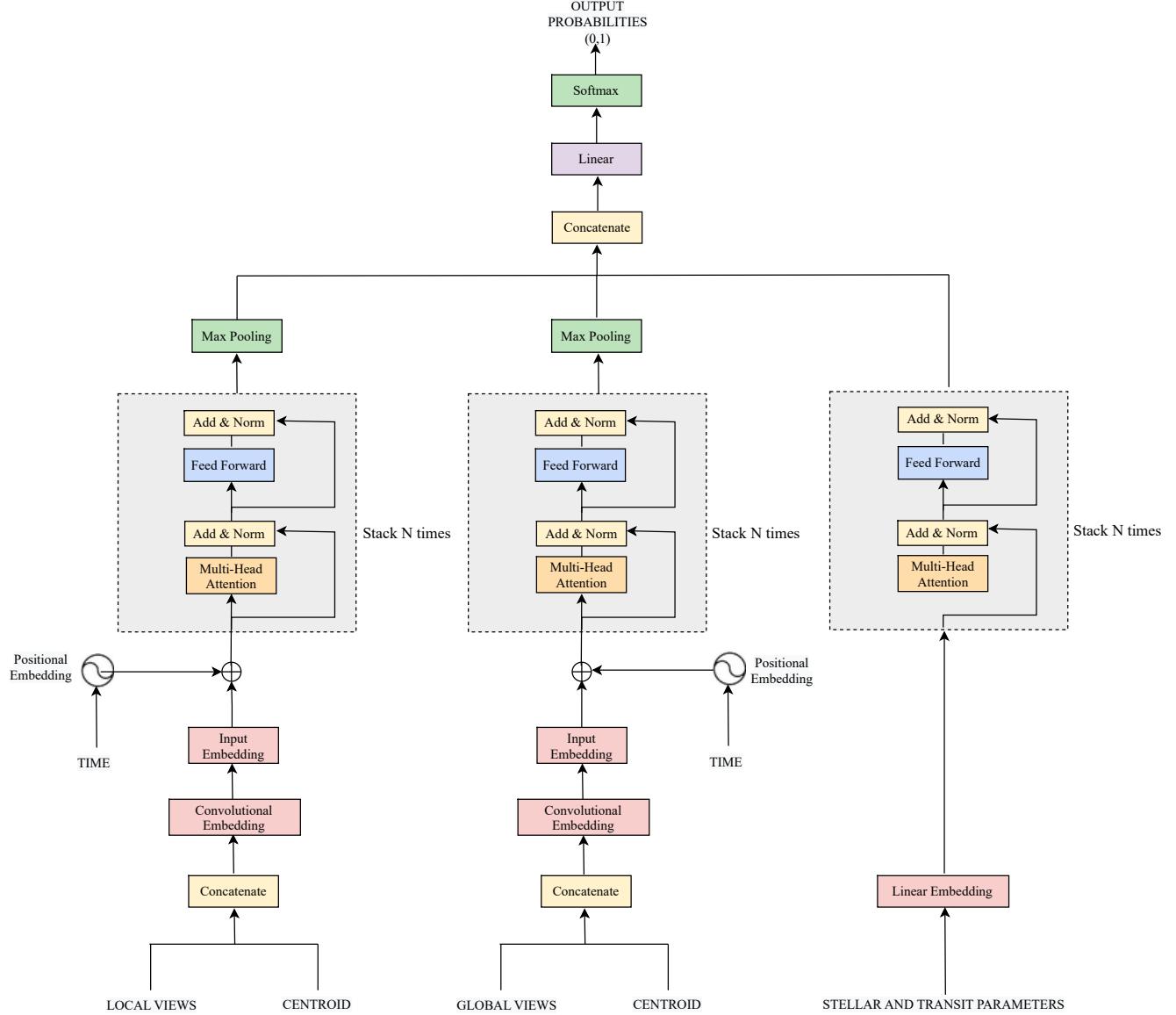


Figure 2. Scheme of the proposed architecture for the analysis of transit signals. The architecture consists of three encoders, where each encoder is designed for i) local view, ii) global view, and iii) stellar and transit parameters. The first encoder block corresponds to the representation of the local view. The input is the concatenated time series of flux time and centroid. A convolutional embedding layer follows, which transforms this input into a vector representation with a positional encoding. This vector is the input of the multihead attention block, which is comprised of N identical layers and a feed-forward sub-layer. The output of the multihead attention block is passed as input to a max pooling layer. The second encoder is implemented for the global view. Then, both time series are concatenated and follow the same steps as the first encoder. The third encoder contains the stellar/transit parameter information and is followed by a linear layer to transform this information into a vector representation. This representation is the input of the multihead attention block. Finally, the output of each encoder is passed as input to a linear layer with softmax for class prediction.

4.2.1 Input embedding

We transform the information of the light curves (local and global view) into a vector embedding through a convolutional layer.

The first step for each piece of information is to generate an input sequence before building the input embedding of each encoder in our model. In the case of the ‘‘local view’’, after the phase folding process, we generate a sequence of observations in terms of the flux and centroid $x_1, x_2, x_3, \dots, x_n$ and $c_1, c_2, c_3, \dots, c_n$ respectively. Then, we concatenate both time series to create a sequence

as input of the first vector embedding. In this way, we obtain a single sequence for the ‘‘local view’’ $\{x'_1, x'_2, \dots, x'_n\}$ which is equivalent to $\{(x_1, c_1), (x_2, c_2), \dots, (x_n, c_n)\}$. For the input sequence of the ‘‘global view’’, we perform the same procedure as for the local view.

As a second step, we build our input data representation. In the context of DL, the data representation is also called vector embeddings and consists of a representation of symbols from data (e.g., words, observations, images, etc.) in a transformed space. Embeddings can be vectors of 1s and 0s, numeric, or neural networks (NNs). In general, an embedding captures dependencies of the input across different

variables in the embedding space; this generates a more expressive input vector for the network and facilitates sequential modeling. To obtain our input embedding, we apply a 1D convolutional layer to our input sequence. We chose to use this type of representation since the CNN of 1D can identify local patterns or features within the convolution window. Each feature in our input sequence only needs to be learned once and a pattern learned at one position can also be recognized at a different position. Thus, given a sequence of observations in terms of the flux and centroid $\{x'_1, x'_2, \dots, x'_n\}$, $x' \in \mathbb{R}^m$, where m is the number of variables of our time series, we apply 1D convolution to the sequence. We denote the convolution filter coefficients as $w \in \mathbb{R}^{n \times k}$, and our input embedding as $w_i^{(k)} \cdot x_{i-1}^{(k)}$ for each x'_i at position i . For 1D, the kernel $k = 1$ slides along one dimension transforming an m -dimensional space to a d -dimensional space for every t .

4.2.2 Positional embedding

Once the embedding vector is built, we add an encoding that assigns the position of each element of the embedding vector. This encoding is added to the local and global view (see Figure 2). Vaswani et al. (2017) implement a positional encoding from a vector of indices and add a trigonometric function as described in Section 2.3. That implementation is applied to sentences of words, where the space of a word with its predecessor is the same. In the case of light curves, the time series are irregular, i.e., a sequence of observation time and value (t_n, x_n) . In the observations we analysed of the TESS telescope, the cadence is ~ 2 min, which means that it is mostly constant but with some interruptions. To account for these slight variations, we inject the observation time t of the light curve instead of the position p with the same trigonometric function defined by Vaswani et al. (2017) (see Equation 5). Thus, our final positional encoding is $P \in \mathbb{R}^{n \times d}$ (one for each element of the sequence), which is a d -dimensional vector that contains information about the specific position in a time series of a light curve of our input embedding. Additionally, we also experimented with injecting p as done initially by Vaswani et al. (2017) instead of t , and as a result, we found that both encodings produce almost identical results (see Section 7.1). This is due to the cadence (mostly constant) of the TESS observations. These results would probably not be the same for light curves with larger or random cadences, which usually occur with ground-based telescopes.

4.2.3 Attention module

In the attention module (gray blocks in Figure 2), we consider the self-attention mechanism described in Section 2.2. This mechanism is the core of every encoder block. Our input embedding is divided into multiple heads, so the split sections of the embedding learn different aspects of the meanings of each observation as it relates to other observations of the same sequence. With this process, the model captures the most important features of the sequence.

In the original self-attention encoder by Vaswani et al. (2017), the mask is used to identify which tokens or values are padding from being part of the attention score; i.e., with the mask, it is possible to control the interaction between each input observation with other input observations. In our model, we do not mask the sequence, i.e., the mechanism observes past and future values. In this way, our model observes when the brightness of the star begins to decrease, and in parallel is able to observe when its brightness increases. With this, we get the mechanism to compare the beginning and ending of a transit signal.

4.2.4 Max pooling layer

The output of each encoder enters a layer of pooling. Pooling layers provide a down sampling approach that reduces features into patches. We introduce a max pooling layer that is added after each implemented encoder for the global and local views. As mentioned before, this reduces the sample of the entities of the next layer, allowing the features in the next layer to have a broader view of the input.

4.2.5 Final Linear and Softmax layer

The Linear layer is a simple fully connected neural network that projects the vector produced by the encoder, into vector called a logits vector. We consider the decoder as our activation function for binary classification, and we use a softmax layer for this task, which returns a vector of probability distribution z . The label of the predicted class is the one with the highest probability score. The loss function is defined by the equation:

$$\mathcal{L} = -\frac{1}{n} \sum_{i=1}^n (y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i)) \quad (7)$$

y_1, y_2, \dots, y_n are the true labels of all samples in the training set (defined to be either 0 or 1), and \hat{y}_i is the prediction probability of the model.

4.3 Analysis of the attention heads

Part of our proposed methodology consists of analysing the attention distributions that the model captured. For this, we measure the entropy in each attention head and average the attention weights for each time step. In this section, we describe both criteria.

4.3.1 Attention entropy

We analysed individual attention heads to observe the patterns of transit signals that have been learned. First, we measure whether the attention is focused on all observations of the phase-folded light curve or only when the transit occurs. In other words, we measure whether attention is focused when the star's brightness decreases. One way to measure this is by computing the average entropy of attention probabilities for each layer as a “sparsity measure” of attention. We compute the attention entropy (8) (Ghader & Monz 2017) to measure how the attention distribution is focused at the timestep t :

$$E_{At_t} = - \sum_i^{|x|} At(x_i, y) \log At(x_i, y) \quad (8)$$

where x_i represents the i -th source token (observation), y denotes the prediction label, and $At_t(x_i, y)$ is the attention distribution at timestep t . Then, we average the entropy of attention over all timesteps to get the final entropy.

4.3.2 Attention weights

To interpret the attention distribution, we averaged the attention weights that the model attended along the observation sequence of dimension n for an element at time t :

$$At(x_t) = \frac{1}{n} \sum_i^n At(x_i), \quad \text{for a time } t \quad (9)$$

4.4 Complexity

The complexity of the proposed approach for each layer is given by $O(n^2 \cdot d)$, where n is the sequence length ($n = 100$ for the local view and $n = 200$ for the global view of the transit signal), $d = 512$ and as a result $n \ll d$. In the case of CNN's, layers are generally more computationally expensive, by a factor of k , which is a filter to extract the features from the light curve. The complexity per each CNN layer is given by $O(k \cdot n \cdot d^2)$ (Vaswani et al. 2017). So, the increase of layers in the network generates a higher complexity that could lead to an overfitting of the model by acting as a filter (Hawkins 2004). As a result, our approach is faster than a CNN.

5 DATASETS

We used the pre-processed TESS light curves from the Mikulski Archive for Space Telescopes (MAST)⁶, which have 2-minute cadence observations of 200,000 to 400,000 selected stars. We use two datasets to collect the light curves. The first dataset corresponds to the TESS TOI catalog (ExoFOP-TESS)⁷, which contains the TIC id of the confirmed and known planets to date, and all the parameters of the planet. In this database, there are roughly 147 systems labeled as Confirmed Planets (CP), which were confirmed as such after the TESS observations were performed; and 420 systems labeled as Known Planets (KP), which were discovered previous to the TESS observations. As second dataset, we use the light curves previously labeled as planet candidate (PC) by DL models, which correspond to the data labeled by Yu et al. (2019) and Rao et al. (2021)(Nigraha), that were disclosed in a public repository⁸. We filter the light curves with greater confidence with a threshold > 0.7 , of which we discarded 23% of them since they corresponded to light curves previously labeled as FP. For this second dataset we collected 153 TCEs.

The objective of our model is to detect the differences between the characteristics of planetary transit signals from those of typical false positives, such as eclipsing binaries (EBs), background EBs, stellar variability (V), or Instrumental systematic (IS). In that sense, for the case of false positive signal samples, we include information available in the ExoFOP-TESS catalog, where we collected a total of 224 EBs and 6 variable stars, and 7 candidates labeled as instrument systematics. We also included 1,886 light curves from the Yu et al. (2019) work, which correspond to 711 EBs, 520 IS, and 655 V.

Additionally, we include to our dataset some non-transit samples that were not previously classified as TOI's, and not belonging to the Nigraha or Yu et al. (2019) PCs. For this, we use a method for detecting time-series planetary transits, the Transit Least Squares (TLS) algorithm proposed by Hippke & Heller (2019). This method has already been successfully applied by Rao et al. (2021) to identify candidates and compute the transit parameters. In addition to the transit parameters, TLS tool⁹ returns the signal to noise ratio (SNR), and the signal detection efficiency (SDE) related to the value of the TLS power spectrum. SDE has become a standard metric with successful results in the exoplanet hunting community (Hippke & Heller 2019). To find signals that do not correspond to planetary systems, firstly, we compute the transit parameters on random samples from

⁶ https://archive.stsci.edu/teess/bulk_downloads/bulk_downloads_ffi-tp-lc-dv.html

⁷ <https://exofop.ipac.caltech.edu/teess/>

⁸ <https://github.com/yuliang419/Astronet-Triage/blob/master/astronet/tces.csv>

⁹ https://transitleastsquares.readthedocs.io/_downloads/en/latest/pdf/

sectors 1-6. Secondly, we compute the peak in the TLS power spectrum to select the light curves with a threshold of $SNR < 6$. Finally, we filter to include those that have already been previously classified by Yu et al. (2019) as J (Junk). This class includes a mix of V and IS, of which we collected a total of 3,000 samples. This way, we obtain samples with greater confidence of belonging to non-transit signals.

The final dataset contains 720 true transit signals and 5,123 signals associated with false positive scenarios. We then split our dataset into three subsets: i) 80% for the training set, ii) 10% for the validation set and iii) 10% for the testing set, which is dedicated to measure the performance of our model of which 15.4% are positive instances. As we can see, our dataset is highly imbalanced, so we work with specific metrics to evaluate the performance of our model (see Section 6.4).

5.1 Data augmentation

Data augmentation is a technique used to augment data from the same data set. This process consists of modifying random samples from the training set. We take this technique to augment our training data. For this process we consider two techniques. First, we vary the t_{min} and t_{max} of the transit view following the same procedure described in Section 4.1.1. We perform this process on a subset of training data corresponding to 50% of the samples. As second step, similar to Ansdell et al. (2018) and Osborn et al. (2020), we add white noise with a standard deviation that is randomly chosen from a uniform distribution between 0 and the mean of the flux to random samples during training. We follow the same steps for each time series of the centroids. This means that the model is trained with different samples in each epoch, preventing overfitting and improving generalization. Adding white noise to signals has been shown to achieve signal variability without changing transit patterns and a better performance of the model (Ansdell et al. (2018); Pearson et al. (2018); Zucker & Giryes (2018); Osborn et al. (2020)). Regarding the stellar and transit parameters, these do not have any change during the data augmentation process since the techniques we adopt do not affect the characteristics of the star, such as T_{eff} , [Fe/H], depth or duration of the transit.

6 TRAINING PROCESS AND PERFORMANCE EVALUATION

In this section, we describe the implementation, and training process, including the regularization and optimization techniques we adopted to implement our model. Finally, we describe the metrics we adopted to evaluate the performance of our model.

6.1 Implementation

Our model described in this work has been implemented in Pytorch (Paszke et al. (2019)) an open source machine learning framework. All experiments are conducted using NVidia GeForce 1080 Ti with 11GB of memory. The code is available at https://github.com/helemysm/attention_exoplanets

6.2 Training

We train our model using the labeled datasets described in Section 5. Our model has exoplanet transit and false positive signals labels (1, 0) as output options.

We perform the data augmentation process described in Section 5.1 for the training set only, i.e. to 80% of all samples. This training set

does not have augmented data from the testing set, and furthermore, the samples for model evaluation do not have any augmentation process. Finally, to train our model, we use all samples in batches of 100 over 60 epochs to learn features of a transit signal.

6.3 Regularization & Optimizers

Dropout is a regularization technique, where some neurons are randomly deactivated during the training phase (Srivastava et al. 2014). We used this technique the same way as in Vaswani et al. (2017). We apply a dropout of 0.1 on the output of each layer, to the positional encoding of the local and global view encoders.

The ADAM algorithm is a variant of the stochastic gradient descent (SGD) (Kingma & Ba 2014). Using Adam instead of SGD allows the learning rate to change as the network learns. To initialize each model, we set a high learning rate of $\alpha = 0.001$, which decreases as the network learns and the value of the loss function decreases. The idea of starting the training with a high learning rate allows the model to start learning faster, but with the risk of skipping the optimal solution (Goodfellow et al. 2016). For this reason, we reduce the learning rate as the model learns, allowing it to learn a more optimal set of weights. To achieve this, we introduce a learning schedule, which is beneficial for reducing the learning rate (Goodfellow et al. 2016). The initial learning rate of $\alpha = 0.001$, decreases by 20% if after 5 or 10 epochs the performance has not been improved.

6.4 Performance Evaluation

In machine learning, classification is the task of predicting the class to which the input data belongs. In this sense, to evaluate how well the classifier manages to predict, evaluation metrics are typically used to reflect the true performance of the classifier.

We evaluate our model using two classes: planet (positive instances) and not planet (negative instances which correspond to false positive signals). As we mentioned before, most of our dataset belongs to false positive signal. Therefore, we evaluate the performance of our model with evaluation metrics for unbalanced datasets. Confusion matrix, recall, precision and F1-score are metrics that consider an unbalanced dataset. Thus, the higher the recall and precision value, the better the performance of our model (Ivezic et al. 2014). Below we describe the metrics for measuring performance based on the number of TP (true positive value, number of positive samples classified correctly), FN (false negative value, number of actual positive samples classified as negative), and FP (false positive value, number of actual negative samples classified as positive instance), which we use to evaluate our model:

- **Confusion Matrix:** displays a matrix with the performance summary of a classification model (Stehman 1997). This summary shows the percentage of correct and incorrect predictions for each class. Each row of the matrix represents the prediction of the classes (x-axis), while the columns represent the true label (y-axis). Thus, we obtain information about the prediction errors of our model with respect to each class.

- **Recall:** from all the true transit signals that exist in our testing dataset, how many are predicted as transit signal. As our classification is binary, the recall is defined as:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (10)$$

In most cases of binary classification with unbalanced datasets, the recall is the best metric to avoid overfitting.

Table 1. Summary of the results in terms of precision, recall and F1-score of our best model on the test set. The standard deviation σ for all three metrics is less than 0.005.

Class	Precision	Recall	F1-score
not planet	0.96	0.96	0.96
planet	0.80	0.80	0.80

- **Precision:** from the cases that our model predicts true transit signal, how many actually are transit signals. In the same way, from the cases that our model predicts no transit signal, how many actually are no transit signal.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (11)$$

In general, the precision is a quantitative measure of the samples that are correctly classified.

- **F1-score:** corresponds to the weighted average of recall and precision:

$$\text{F1score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (12)$$

Considers recall and precision, where a high value indicates good performance in binary classification.

- **Precision-Recall curve:** One of the most important metrics to evaluate the performance of an unbalanced classification model is the precision-recall curve, which summarizes the trade-off between precision and recall for different thresholds.

7 RESULTS

Our model was evaluated with samples, which the network has not seen up to this point. We evaluate our model with metrics described in Section 6.4. As mentioned before, our dataset is highly imbalanced. In view of this, we focused on evaluating our model based on the recall precision and the F1-score.

For each evaluation in all of the experiments, we perform 10-fold stratified cross-validation taking the mean as the final result. Table 1 shows the result of the evaluation. The table shows that our model correctly classified 88.0% of the samples in total. We computed confusion matrices to display the percentage of correct and incorrect predictions. Figure 3 shows the confusion matrix for the test set, where 80% of the instances with planet labels were correctly classified.

We compared our model with Rao et al. (2021). We recreated the Nigraha architecture with the parameters described in Rao et al. (2021). Then, we train and evaluate it with the same dataset described in this work. As a result, Nigraha correctly classified 86.6% of the samples in total, and with a recall of 77.7% for the samples with planet labels. In this way, we compared two strategies that use different DL models. Additionally, we compared our architecture against traditional models, such as support vector machine (SVM) and Multilayer perceptron (MLP). In all the experiments performed, both models do not exceed a precision with a threshold > 0.6 . Therefore, like Shallue & Vanderburg (2018) and Pearson et al. (2018), architectures based on convolutional layers and, in our case, self-attention mechanisms achieve superior results for this type of time series.

	True label	
Not Planet	478	17
Planet	18	72
	Predicted label	
Not Planet		
Planet		

Figure 3. Confusion matrix resulting from the application of our architecture on the evaluation dataset that contains confirmed planets CP and known planets KP described in Section 5. Planet corresponds to positive instances and no planet to negative instances.

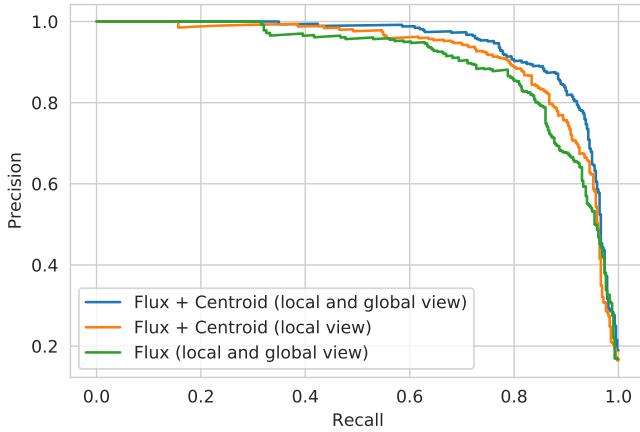


Figure 4. The precision-recall curve for test sets. The figure shows better performance if we include the centroid information in the model training. Furthermore, we can see a slight difference between training the model with only the local view of the transit + centroid compared to training the model without the centroid but with both views.

Nonetheless, beyond getting better performance, we focused on detecting the characteristics of an exoplanet transit and of a false positive signal. In the ablation study (see Section 7.1), we show the contribution of each component of our model, and we also observe how the centroid information improves the performance.

7.1 Ablation study

To understand the contribution of the most important components of our architecture, we evaluate the performance with ablation experiments. It is important to mention that all our experiments consider

Table 2. Ablation study of different components of our model. Default model is: $PE = \text{time}$, $d_{emb} = 512$, $h = 4$, $l = 4$ with local and global view and centroid information.

Description	Avg. Precision	Avg. Recall	Avg. F1-score
Default	0.880	0.880	0.880
w/o PE	0.495	0.430	0.425
PE_{abs_time}	0.875	0.870	0.870
$l = 1$	0.860	0.865	0.860
$l = 8$	0.885	0.880	0.880
Without centroid	0.850	0.860	0.850
Without global view	0.865	0.870	0.870
$d_{emb} = 128$	0.845	0.855	0.850
$d_{emb} = 256$	0.865	0.870	0.870

the information on the stellar/transit parameters since the objective is to observe the relative importance of each parameter. Table 2 shows the performance of our model based on the number of heads and layers. In addition, we can observe the impact of incorporating the centroid into the encoder input with the flux time series. Specifically, we train our model by ablating specific components:

- **Importance of positional information:** We used three types of positional encoding (PE): no positional encoding, a sinusoidal encoding that depends on observation time, and sinusoidal encoding that relies on the absolute position. Since the attention is calculated at each time step independently, the sequential order of the observations is lost. Without the sequential order, the signal of a transit would have observations with the value of the brightness decrease in any time step of the sequence, making it challenging to identify the transit patterns in relation to time. We evaluated the performance if we did not inject a sequential order into the input, and the average precision performance of the model was 50%, similar to a random result for binary classification. Furthermore, we note that the type of positional encoding slightly influences the results. A PE that depends on the observation time performs ~1.0% better than a PE that considers the same space between tokens. With these results, we found that both encoding produce very similar results due to the fact that the observation time variation is minimal without achieving significant changes in the positional encoding vector.

- **Number of layers and attention heads:** In all our experiments we use the same number of layers and heads for each encoder. To choose the appropriate number of heads, we study the entropy of each head in each layer (see Section 7.2), where we considered 8 heads. We noticed that there were no significant changes in the model’s performance, either with $h = 8$ or $h = 4$. However, the first 4 heads have the lowest entropy, meaning that each head’s attention is less dispersed and more focused on specific observations of the sequence. As we mentioned before, in multihead attention theory, the heads are specialized in different parts of a sequence, but also, some layers can be reduced to only one head without changing the performance of the model (Michel et al. 2019).

Our default model is trained with 4 heads and a varying number of layers, since there were no significant differences in the results with 8 heads. By reducing $l = 1$, the model performance evaluation metrics decreases 1.5%. In the case of working with $l = 8$, the model

has a similar performance to $l = 4$, where basically there is a greater difference in the model training time.

- **Without centroid Information:** We perform experiments without considering the centroids. As a result, the model reduces its performance by 2%. So, we achieve better results if we include this information in the encoder. In addition, the attention weights are more clearly focused when the planetary transit begins and ends, which benefits the interpretability of the model decision (see Section 7.3).

- **Without global view:** We perform experiments without considering the global view of the transit signal. In the experiments, we observed that without including the global view of the transit signal, the performance of the model decreases slightly by 1.0%, but not significantly.

Figure 4 shows the precision versus recall for three different models: i) considering the information of the centroids and both views (local and global), ii) local view and centroids, iii) local and global view, but without centroids. In the figure, we observe that the architecture that considers the information of the centroids and both views (local and global) performs better. In the case of the architecture that considers the centroids but not the global view, it behaves with an improvement compared to the architecture without centroids but with both views.

7.2 Individual Attention Heads

To analyse each attention head, we use the attention entropy described in Section 4.3.1. The entropy is measured in *nats* units, which means “The natural unit of information”, and we use it to measure the concentration of the attention distribution at a time t . An entropy above the average indicates that the attentions are more dispersed. Otherwise, and if the entropy values of each head vary, it could indicate that the attention of those heads is focused on specific observations.

Figure 5 shows the average entropy, where we observe that some attention heads, especially in the last layers (6, 7 and 8), have wide attention, but also the heads have similar entropy values. This means that in those layers, there is more attention. These attention heads generally have at most 10% of their attention weights on a single token (observation). Furthermore, in Figure 5 we show the attention entropy of the heads in each layer, where the first heads (1-4 blue dots) have below average entropy compared to the last heads (5-8 red dots). This means that the attention weights are concentrated on specific observations. This analysis helps us define the optimal number of heads for our model, in our case, $h = 4$. With $h = 8$ there were no significant differences in model performance. Another option for head analysis is to perform a process called “head pruning” on each layer (Voita et al. 2019). This means that heads can be removed and will not significantly affect the result. In future work, we will consider a combination of techniques, including head pruning, in such a way that we can improve the analysis of the model results.

7.3 Analysis of Attention Maps

We interpret the importance of the signal characteristics for the final prediction by analysing the weights of attention. To do this, we analyse the attention head with an average entropy value over all the layers. Then, we interpret the distribution of attention based on the average attention described in Equation 9.

We take two random samples from the test set in order to observe

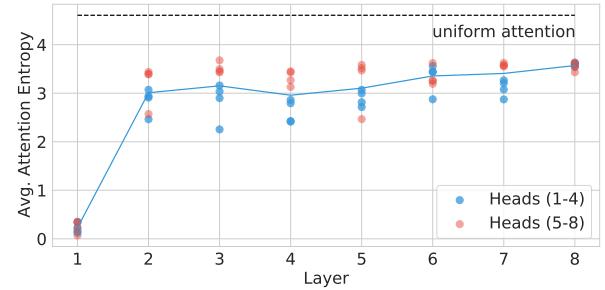


Figure 5. Average entropy of attention distributions. The blue line indicates the average attention entropy calculated for each layer, where each layer has 8 attention heads. The points represent the average value of the attention entropy of each head. The first four heads (1-4) of each layer are the blue dots, and the red dots correspond to the last four heads (5-8).

the attention map for the model trained without the information of the centroids, and for the model trained with the concatenated representation of the flux and centroid. Figure 6(a) on the left shows the two exoplanetary transit signal samples of the trained model without the centroids. We observe that the distribution of attention weights is focused at various transit times. In other words, the attention weights are more dispersed. In the case of the model trained with the information of the centroids concatenated to the flux, the Figure 6(b) on the right shows that the attention weights are focused when the planetary transit begins and ends.

Beyond slightly improving the performance of the model, including the centroid information makes the attention weights focus when the brightness decreases and increases, i.e. we obtain a better input embedding for the model. This means that in many cases, the analysis of the centroid information could be essential to identify a planetary transit signal. In others words, the probability that a signal belongs to a planetary transit signal increases if it does not have false positive characteristics. This interpretation is similar to the analysis that experts perform to confirm a candidate.

7.3.1 Transit signals of planets

Figure 7 shows four true positive samples, i.e. the signal of a planetary transit of TOI’s labelled as KP or CP and their stellar/transit parameters. Our first observation is that there is more attention when the transit starts, and this attention continues until the end of the sequence of observations. In addition to the beginning of the transit, there is also attention when the transit ends. Another characteristic to highlight are the attention patterns towards the observations before and after the transit, which have very low attention and are sometimes ignored. This is because these observations are not relevant to the final decision of the model. This analysis is very similar to the interpretation of a text in NLP, since there are irrelevant words that do not affect the meaning of a paragraph. Regarding the stellar/transit parameters, the four samples in the figure show the attention map for the six parameters that we consider. In these figures, we observe that the decision of whether the signal belongs to a planetary transit is based mostly on the values of the transit depth and the temperature of the host star. In contrast, the attention is low for the [Fe/H] parameter in all examples.

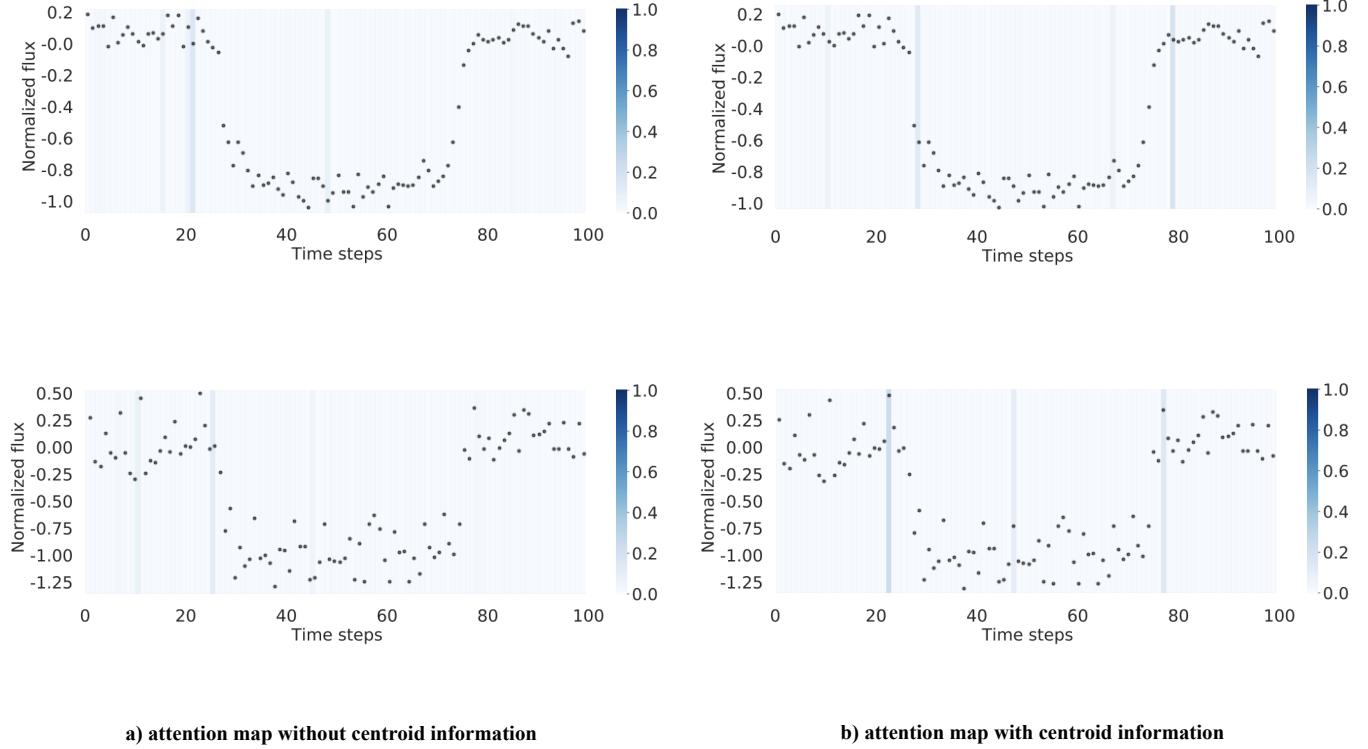


Figure 6. Interpretation of predictions for two different planetary transit signals from the test set. Each plot represents the transit signal (gray dots), with the attention weights that the model calculated and the bar on the left represents the proportion of the associated weight (Heatmap of the self-attention weight matrix). On the left in a), the two transits are shown with the attention map for the trained model without the information of the centroids. On the right in b), the two transits are shown with the attention map for the model trained with the information of the centroids. Notice that in a), the weights of attention are more dispersed, whereas in b) the weight of attention is stronger when the transit begins and ends. Uniform weights are observed for the other timesteps.

7.3.2 Transit signals of false positives

Figure 8 shows four samples of false positive signals. Specifically, two of these samples belong to an EB; (a) and (c), and two belong to variable stars; (b) and (d). In the case of EB samples, we observe that the attention is focused close to the beginning and at the minimum point of depth of the transit. Also, in the attention map of its stellar/transit parameters, the depth value shows more attention than average. The distribution of these attentions is similar to the characteristics of EBs since their transits usually have greater depth than a planetary transit. It is the relevant characteristic that the model detected to distinguish this type of false positive. In the case of, Figure 8(b) belongs to another sample of a false positive signal, a variable star sample. In this sub-figure, we observe that the attention weights are more significant at the beginning of the light curve, which could seem like the beginning of some planetary transit but with a longer duration. However, that attention ends before the maximum point of depth. This is related to the attention to the stellar/transit parameters since the attention is concentrated on the duration of the transit. Finally, the last sample (d) belongs to another variable star, this sub-figure shows a more dispersed distribution of attention weights and this dispersion is concentrated until the maximum point of transit depth. This means that the model has found the most important features up to that point and thus classifies it as a non-planet.

In general, the attention maps help us to indentify the patterns

that the model detected to characterize planetary and false positive signals. All the attention of the model is distributed in each datapoint, and most of the datapoints have low attention. This is because the model determines they are irrelevant to characterize a transit signal. The datapoints with the most attention are those that represent the decrease or increase in brightness of the star in a significant way.

Figure 9 shows the attention maps of all the test samples set for planetary and false positive signals, and below the heat maps that show the general trend of all the samples for each time step. Figure 9(a) clearly shows the patterns that characterize planetary transit signals, where the transit begins and ends. Then, Figure 9(b) shows slightly more dispersed attention. However, the pattern of attention is concentrated close to the time in which there is a greater depth of transit.

Analysis of the attention maps showed that it is possible to distinguish observations containing the main features of a planetary transit and a false positive signal. Mainly, we remember that the depth of the transit is an essential characteristic that exoplanet hunters typically analyze to examine false positive signals. In this sense, our results regarding the analysis of the attention maps showed that the model also observes this depth characteristic to make a decision. With this in mind, working a light curve with a sequential model instead of a CNN has advantages for a better understanding of the predictions. Moreover, it is possible to analyze other parameters in addition to the light curve, such as stellar or transit parameters. This analysis would help us better understand the correlation of the stellar/transit parame-

ters with the prediction and, thus, facilitate the candidate examination process.

8 CONCLUSIONS

In this work, we present a DL architecture for classifying the light curves of transiting exoplanet candidates. Furthermore, through our architecture, it is possible to interpret a prediction to distinguish signals of planetary and false positives. The interpretation of the prediction consists of showing the parameters and observations that the model pays attention to classify a transit-like signal. In addition to the interpretation, the implemented architecture is able to achieve comparable results to state-of-the-art architectures for the identification of exoplanetary signals.

While this work is motivated by the problem of the classification of signals of planetary transits, the architecture is adequate to interpret and observe the importance of each element of the input. Previous works have shown that model performance improves with the inclusion of centroids. In our work, with the help of the attention interpretation, we also show how it affects the model to improve performance, which we think is also important. In addition, to obtain a better interpretation of the model's attention, we analysed the attention entropy of the heads. This helps us determine the number of heads for implementing a model based on attention mechanisms.

The attention mechanism is a theory that has been proven to be successful in NLP applications but has not been previously explored in exoplanet science, where much of the analysis is from sequential data such as light curves. Models like the Transformers are designed to work in complex scenarios for CNNs and RNNs, like time series, since the greater the number of observations of the light curve, the greater the complexity of the model. In this sense, working with architectures such as the Transformer is more beneficial if we consider light curves with more observations, overcoming the limitations of other models. For example, the light curves from telescopes designed to detect transiting planets are characterized by long light curves. We hope that applying this approach can improve other astronomical event classification tasks.

Our future work aims to scale up to longer light curves without building phase-folded light curves, which are typically used to train a DL model. For example, to classify entire light curves and predict whether or not they belong to a star that hosts exoplanets since a light curve could contain one or more exoplanet candidates. Furthermore, we can take advantage of the interpretability of attention mechanisms since analysing the distribution of attention weights could indicate where the possible transit signals are located.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the support from the National Agency for Research and Development (ANID), through the FONDECYT Regular project number 1180054. HS acknowledges support from ANID, through Scholarship Program/Doctorado Nacional/2020-21212185. FP acknowledges support from ANID, through Scholarship Program/Doctorado Nacional/2017-21171036. DM acknowledges the support from ANID by Proyecto Basal FB210017, through the National Center for Artificial Intelligence CENIA. This paper utilizes public information of the TESS and Kepler mission. Funding for the TESS mission is provided by the NASA Explorer Program. Finally, thanks to the availability of data

from TESS, we were able to examine and display our analysis of transit signals.

DATA AVAILABILITY

All the data referenced here and that was used is public information about the TESS mission from Mikulski Archive for Space Telescopes (MAST)¹⁰. For the original light curve dataset, the reader is advised to explore the available documentation on the TESS dataset (Tenenbaum & Jenkins 2018).

REFERENCES

- Allam Jr T., McEwen J. D., 2021, arXiv preprint arXiv:2105.06178
- Ansdell M., et al., 2018, *The Astrophysical Journal Letters*, 869, L7
- Armstrong D. J., Pollacco D., Santerne A., 2016, *Monthly Notices of the Royal Astronomical Society*, p. stw2881
- Armstrong D. J., et al., 2018, *Monthly Notices of the Royal Astronomical Society*, 478, 4225
- Auvergne M., et al., 2009, *Astronomy & Astrophysics*, 506, 411
- Bahdanau D., Cho K., Bengio Y., 2014, arXiv preprint arXiv:1409.0473
- Batalha N. M., et al., 2013, *The Astrophysical Journal Supplement Series*, 204, 24
- Borucki W. J., et al., 2010, *Science*, 327, 977
- Bowles M., Bromley M., Allen M., Scaife A., 2021, arXiv preprint arXiv:2111.04742
- Burke C. J., et al., 2014, *The Astrophysical Journal Supplement Series*, 210, 19
- Catanzarite J. H., 2015, Technical report, Autovetter Planet Candidate Catalog for Q1-Q17 Data Release 24. KSCI-19091-001 <https://exoplanetarchive.ipac.caltech.edu/docs/KSCI-19091...>
- Cho K., Van Merriënboer B., Gulcehre C., Bahdanau D., Bougares F., Schwenk H., Bengio Y., 2014, arXiv preprint arXiv:1406.1078
- Ciregan D., Meier U., Schmidhuber J., 2012, in 2012 IEEE conference on computer vision and pattern recognition. pp 3642–3649
- Coughlin J. L., et al., 2016, *The Astrophysical Journal Supplement Series*, 224, 12
- Dong L., Xu S., Xu B., 2018, in 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp 5884–5888
- Dosovitskiy A., et al., 2020, arXiv preprint arXiv:2010.11929
- Gehring J., Auli M., Grangier D., Yarats D., Dauphin Y. N., 2017, in International Conference on Machine Learning. pp 1243–1252
- Ghader H., Monz C., 2017, arXiv preprint arXiv:1710.03348
- Goodfellow I., Bengio Y., Courville A., 2016, Deep learning. MIT press
- Han K., et al., 2020, arXiv preprint arXiv:2012.12556, 2
- Hawkins D. M., 2004, *Journal of chemical information and computer sciences*, 44, 1
- He K., Zhang X., Ren S., Sun J., 2016, in Proceedings of the IEEE conference on computer vision and pattern recognition. pp 770–778
- Hippke M., Heller R., 2019, *Astronomy & Astrophysics*, 623, A39
- Howell S. B., et al., 2014, *Publications of the Astronomical Society of the Pacific*, 126, 398
- Huang C.-Z. A., et al., 2018, arXiv preprint arXiv:1809.04281
- Ivezic Ž., Connolly A. J., VanderPlas J. T., Gray A., 2014, in, Statistics, Data Mining, and Machine Learning in Astronomy. Princeton University Press
- Karita S., et al., 2019, in 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU). pp 449–456
- Kingma D. P., Ba J., 2014, arXiv preprint arXiv:1412.6980
- Kovács G., Zucker S., Mazeh T., 2002, *Astronomy & Astrophysics*, 391, 369
- Krizhevsky A., Sutskever I., Hinton G. E., 2017, *Communications of the ACM*, 60, 84

¹⁰ https://archive.stsci.edu/tess/bulk_downloads/bulk_downloads_ffi-tp-lc-dv.html

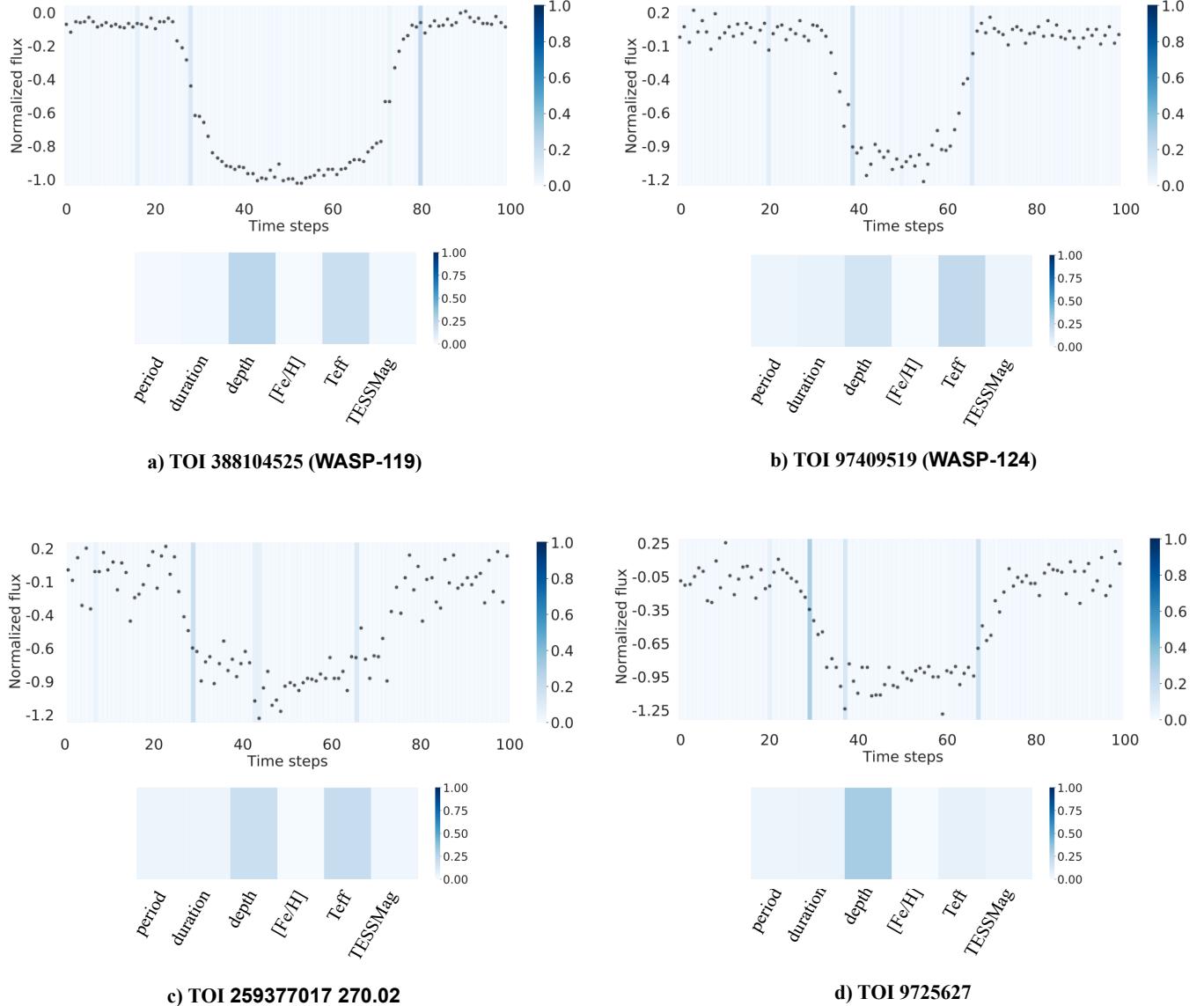


Figure 7. Interpretation of predictions for four signals of a planetary transit of TOI's labelled as KP or CP. Each sub-figure shows the transit signal (gray dots) with the attention weights that the model calculated in the figure above and the stellar/transit parameters in the heat map below.

Lakew S. M., Cettolo M., Federico M., 2018, arXiv preprint arXiv:1806.06957

LeCun Y., Bengio Y., Hinton G., 2015, nature, 521, 436

Li Y., et al., 2020, Scientific reports, 10, 1

Lightkurve Collaboration et al., 2018, Lightkurve: Kepler and TESS time series analysis in Python, Astrophysics Source Code Library (ascl:1812.013)

Lin J. Y.-Y., Liao S.-M., Huang H.-J., Kuo W.-T., Ou O. H.-M., 2021, arXiv preprint arXiv:2110.01024

McCaullif S. D., et al., 2015, ApJ, 806, 6

Michel P., Levy O., Neubig G., 2019, Advances in neural information processing systems, 32

Mislis D., Bachelet E., Alsubai K., Bramich D., Parley N., 2016, Monthly Notices of the Royal Astronomical Society, 455, 626

Morton T. D., Bryson S. T., Coughlin J. L., Rowe J. F., Ravichandran G., Petigura E. A., Haas M. R., Batalha N. M., 2016, The Astrophysical Journal, 822, 86

Morvan M., Nikolaou N., Yip K. H., Waldmann I., 2022, arXiv preprint arXiv:2207.02777

Mullally F., et al., 2015, The Astrophysical Journal Supplement Series, 217, 31

Osborn H. P., et al., 2020, Astronomy & Astrophysics, 633, A53

Paszke A., et al., 2019, Advances in neural information processing systems, 32

Pearson K. A., Palafox L., Griffith C. A., 2018, Monthly Notices of the Royal Astronomical Society, 474, 478

Pollacco D., et al., 2006, in, Close Binaries in the 21st Century: New Opportunities and Challenges. Springer, pp 251–253

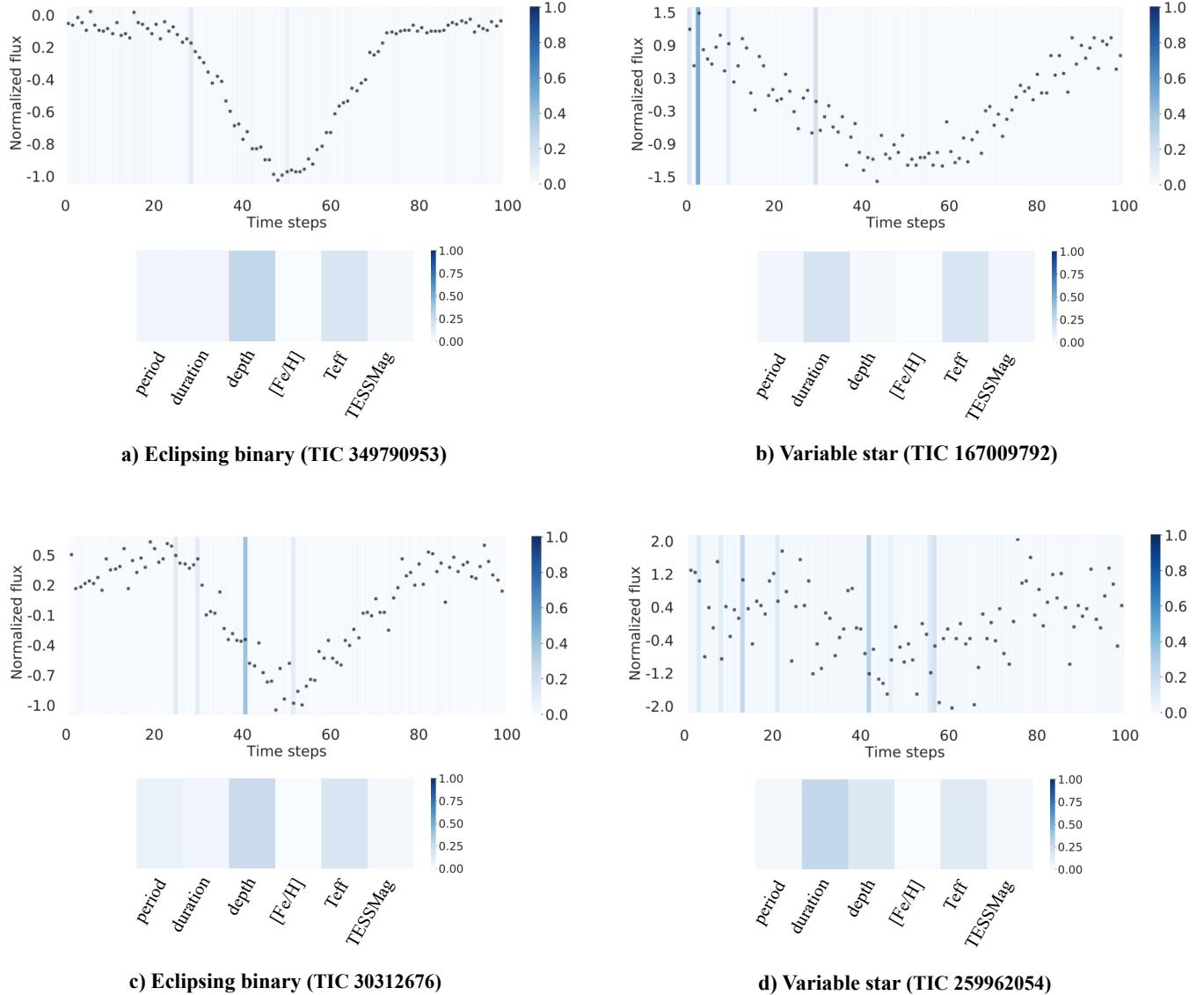


Figure 8. Interpretation of predictions for four signals, which was correctly classified as not planet, which correspond to TOI's labelled as false positives. Each sub-figure shows the signal (gray dots) with its background attention heat map (blue), and the attention map of its stellar/transit parameters below.

- Rao S., Mahabal A., Rao N., Raghavendra C., 2021, Monthly Notices of the Royal Astronomical Society, 502, 2845
- Rauer H., et al., 2014, Experimental Astronomy, 38, 249
- Ricker G. R., et al., 2014, Journal of Astronomical Telescopes, Instruments, and Systems, 1, 014003
- Ricker G. R., et al., 2015, Journal of Astronomical Telescopes, Instruments, and Systems, 1, 014003
- Rowe J. F., et al., 2015, The Astrophysical Journal Supplement Series, 217, 16
- Schanche N., et al., 2019, Monthly Notices of the Royal Astronomical Society, 483, 5534
- Shallue C. J., Vanderburg A., 2018, *The Astronomical Journal*, 155, 94
- Srivastava N., Hinton G., Krizhevsky A., Sutskever I., Salakhutdinov R., 2014, The journal of machine learning research, 15, 1929
- Stehman S. V., 1997, Remote sensing of Environment, 62, 77

- Sun C., Shrivastava A., Singh S., Gupta A., 2017, in Proceedings of the IEEE international conference on computer vision. pp 843–852
- Tenenbaum P., Jenkins J. M., 2018, Technical report, TESS Science Data Products Description Document: EXP-TESS-ARC-ICD-0014 Rev D
- Tenenbaum P., et al., 2012, *ApJS*, 199, 24
- Thompson S. E., Mullally F., Coughlin J., Christiansen J. L., Henze C. E., Haas M. R., Burke C. J., 2015, *The Astrophysical Journal*, 812, 46
- Thompson S. E., et al., 2018, *ApJS*, 235, 38
- Thuruthipilly H., Zadrozny A., Pollo A., 2021, arXiv preprint arXiv:2110.09202
- Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A. N., Kaiser L., Polosukhin I., 2017, in Advances in neural information processing systems. pp 5998–6008
- Vilone G., Longo L., 2020, arXiv preprint arXiv:2006.00093

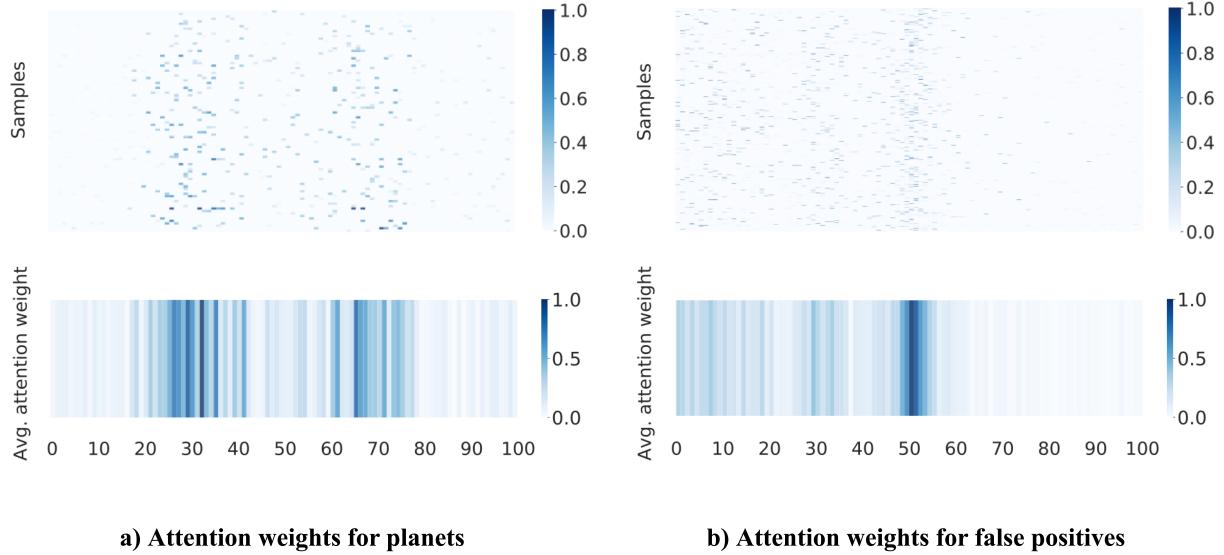


Figure 9. Attention maps of transit signal samples. The graphs overhead are the attention maps of all the test samples set. The x-axis in a heat map representing time step. The charts down show the general attention trend for each time step, with warmer colors indicating a higher concentration of weights at specific times.

- Voita E., Talbot D., Moiseev F., Sennrich R., Titov I., 2019, arXiv preprint arXiv:1905.09418
- Voulodimos A., Doulamis N., Doulamis A., Protopapadakis E., 2018, Computational intelligence and neuroscience, 2018
- Wu B., et al., 2020, arXiv preprint arXiv:2006.03677
- Yu L., et al., 2019, The Astronomical Journal, 158, 25
- Zerveas G., Jayaraman S., Patel D., Bhamidipaty A., Eickhoff C., 2021, in Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pp 2114–2124
- Zhang Q., Yang Y., Ma H., Wu Y. N., 2019, in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 6261–6270
- Zucker S., Giryes R., 2018, The Astronomical Journal, 155, 147

This paper has been typeset from a \TeX / \LaTeX file prepared by the author.