The SATCHEL pipeline: A general tool for data classified through citizen science

E. J. Safron, T. S. Boyajian, and N. Eisner

1Department of Physics & Astronomy, Louisiana State University, 202 Nicholson Hall, Baton Rouge, LA 70803, USA
2Department of Physics, University of Oxford, Parks Rd, Oxford OX1 3PJ, United Kingdom

Accepted XXX. Received YYY; in original form ZZZ

ABSTRACT

Citizen science is a powerful analysis tool, capable of processing large amounts of data in a very short time. To bridge the gap between classification data products from web-based citizen science platforms to statistically robust signal significance scores, we present the Search Algorithm for Transits in the Citizen science Hunt for Exoplanets in Lightcurves (SATCHEL) pipeline. This open source, customizable pipeline was constructed to identify and assign significance estimates to one-dimensional features marked by volunteers. We describe the functional capabilities of the SATCHEL pipeline through application to features in photometric time-series data from the Kepler Space Telescope, classified by volunteers as part of the Planet Hunters citizen science project hosted on the Zooniverse platform. We evaluate the SATCHEL pipeline’s overall performance based on recovery of known signals (both simulations and signals corresponding to official Kepler Objects of Interest) and relative contamination by spurious features. We find that, for a range of pipeline hyperparameters and with a reasonable score cutoff, SATCHEL is able to recover volunteer identifications of over 98% of signals from simulations corresponding to exoplanets $\sim 2 R_\oplus$ in radius and about 85% of signals corresponding to the same size range of KOIs. SATCHEL is transparently adaptable to other citizen science classification datasets, and available on GitHub.

Key words: methods: data analysis – software: documentation – software: public release – planets and satellites: detection

1 INTRODUCTION

Large-scale scientific experiments and surveys are producing ever more vast amounts of data. The Kepler Space Telescope, launched in 2009 (Borucki et al. 2010), produced about thirty gigabytes of data per month (Christiansen 2021). By contrast, its currently-operating successor, the Transiting Exoplanet Survey Satellite (TESS, Ricker et al. 2015), generates more than twice that in a single day. The upcoming Rubin Observatory (formerly the Large Synoptic Survey Telescope, LSST, Jurić et al. 2017) will generate a colossal twenty terabytes of raw data per night. For a single scientist or small team of scientists, it is impractical at best and impossible at worst to analyze such copious data. For some datasets, automated or machine learning algorithms have been shown to be successful at various types of tasks (Baron 2019), and the field of machine learning is evolving rapidly in response to the rising need for innovative methodologies.

For analysis tasks involving pattern recognition, however, many automated techniques still fall short of the remarkable capabilities of the human brain. This has led to the development and increasingly widespread use of “citizen science,” in which the general public engages in scientific research by contributing intellectual effort, knowledge, or tools and resources (Gordienko 2013). One popular platform that offers volunteers the opportunity to assist in data analysis is the Zooniverse (Lintott et al. 2008), where volunteers (henceforth “users”) can access curated scientific data (“subjects”) and help analyze them by performing simple tasks (“classifications”). Through the Zooniverse, citizen science has been successful over the last decade for studies involving tasks such as:

- Identification of novel objects, such as Hanny’s Voorwerp (Lintott et al. 2009) and Boyajian’s Star (Boyajian et al. 2016),
- Categorization and cataloging (Willett et al. 2013; Johnson et al. 2015; Simmonset al. 2017; Dickinson et al. 2018; Christiansen et al. 2018; Zink et al. 2019), and

While not the focus of this work, volunteer participation can also come in the form of data collection, as exemplified by the plethora of citizen science projects hosted by the Cornell Lab of Ornithology (Bonney et al. 2009), and by the wealth of astronomical measurements provided by amateurs as part of, for instance, the American Association of Variable Star Observers (AAVSO) or the TESS Follow-up Working Group.

1 http://zooniverse.org

© 2021 The Authors
In addition to its versatility, citizen science accomplishes many of its project goals extremely quickly. For example, users of the Snapshot Serengeti project fully processed 18 months of backlogged data within the first three days of the project's establishment on the Zooniverse platform (Swanson et al. 2015). In 2015, it was estimated that, on average across all Zooniverse projects, volunteers processed an amount of data equivalent to 34 years of full-time work by a single researcher throughout the course of a project (Cox et al. 2015).

Generally, a single subject is classified by many users, as the skill levels of different users may vary significantly. Crowdsourcing, as this method is sometimes called, effectively suppresses bias from any one individual user, and it has been shown that for some kinds of classification tasks, the combined judgment of many volunteers is as good or better than that of a single expert (Willett et al. 2013). Methods for aggregating these classifications and determining the final verdict for a given subject vary according to the needs of the project and the nature of the data. Some projects find it sufficient to rely on simple crowd majority voting (Kuchner et al. 2016; Zink et al. 2019). Others find it more beneficial to "weight" the votes according to some measure of the skill of each user who engaged with the subject. User skill is typically measured based on

- User performance on "gold-standard" data or simulations with known "correct" classifications (e.g., Marshall et al. 2016; Eisner et al. 2020a),
- User classification agreement with consensus (e.g., Willett et al. 2013; Johnson et al. 2015),
- A proxy for engagement level or attentiveness to detail (e.g., Simpson et al. 2012),

or a combination of two or more of the above metrics (e.g., Schwamb et al. 2012; Simmons et al. 2017). If consensus agreement is used to assign user weights, weighted vote fractions (henceforth, "scores") may be recalculated and user weights readjusted, repeating this process several times to achieve convergence of results.

In this paper, we present the Search Algorithm for Transits in the Citizen science Hunt for Exoplanets in Lightcurves (SATCHEL) pipeline, an open source, Python-based pipeline to process classifications comprised of arbitrary, one-dimensional markings made by citizen science volunteers. By one-dimensional, we refer to user markings with boundaries along only one measurement axis, rather markings whose midpoint requires both \((x, y)\) pixel coordinates. SATCHEL is designed to score individual features in classification subjects, rather than subjects themselves. This is especially useful for projects in which multiple types of significant signals may be present in a single classification subject, prompting a need for the ability to filter out some types of features while retaining others. SATCHEL employs a user weighting scheme based on both gold-standard data and consensus agreement, and the latter weighting process is iteratively coupled to the calculation of scores. Unlike the weight-score coupling implemented in other analysis pipelines, SATCHEL does not execute a predetermined number of iterations to achieve convergence. Instead, convergence is determined by stability of the user weight distribution.

In the following sections, SATCHEL is described via application to classification data from the Planet Hunters\(^6\) project (PH, Fischer et al. 2012). An overview of the PH project and interface is given in Section 2. The SATCHEL feature isolation, user weighting, and scoring processes are described in detail in Section 3. A performance study, including measurement of purity and completeness, is presented in Section 4. Section 5 contains a summary and discussion. Appendices A, A2, and A3 provide details of pipeline optimization. Details regarding scientific motivation, sample selection criteria, and astrophysical implications of the study are deferred to an accompanying paper, hereafter “Paper II.”

3 In addition to astronomy and physics, the Zooniverse also hosts projects that apply citizen science data analysis in the areas of climatology, biology, history, and various humanities.

4 The SATCHEL pipeline is publicly available on GitHub, at https://github.com/ejsafron/PH-pipeline.

5 https://planethunters.org

6 At the time of writing, the number of confirmed and candidate exoplanet detections attributed to Kepler is 4780, according to the NASA Exoplanet Archive.
2.1 Data and Interface

Data from the Kepler Space Telescope were transmitted and released in a series of 17 quarters, each consisting of about three months of high-precision photometric time-series, or light curves, for about 150,000 stars. The data products released by the Kepler science team included both “long-cadence” light curves, composed of photometric observations taken every 29.4 minutes, and “short-cadence” light curves, in which observations were taken every 58.89 seconds. The raw long-cadence light curves were scaled and corrected by Pre-Search Data Conditioning (PDC) in the Kepler science processing pipeline (Jenkins et al. 2010) to compensate for systematic instrumental errors, and both long- and short-timescale spurious and non-astrophysical trends.

For display in the classification interface on the PH website, the PDC-corrected light curves from each quarter were obtained for 5625 likely M-dwarf targets7 by the PH team from the Kepler Mission Archive8 hosted by the Mikulski Archive for Space Telescopes9 (MAST). Each quarter-long light curve was divided into three segments (usually about 30 days each), and then normalized by the mean flux of the segment, as shown in Fig. 1. These light curve segments were displayed in the PH user interface as classification subjects, ordered randomly and devoid of any indicator of which star the data represented. The only metadata available to a user were the subject’s effective temperature, magnitude, and estimated radius.

The first time a user accessed the PH site, a brief tutorial was shown in a pop-out interface to guide the user through classification of a typical light curve. An exoplanetary transit example was shown as a localized, dip-like feature, on the order of a couple percent in brightness measurement. The user was instructed to draw a box marking the beginning and end of the transit feature, and do the same for any other transit features on the light curve. After a brief confirmation dialogue, the user was shown real data to classify. Users were given the option to register with a Zooniverse account, or login to their existing account if they already had one. An unregistered user was reminded of this option periodically if they remained on the site to classify more than one light curve. If an already-registered user logged in to the site, they would not be shown the tutorial again, but the tutorial was always accessible via a link in the subject’s right-hand menu bar. About 43.9% of users were registered, performing 84.2% of all classifications (see Fig. 2b).

Once a light curve subject was independently classified by ten different users, it was retired from the random display rotation. This number was chosen based on calculations from early results of the Galaxy Zoo10 citizen science project (Lintott et al. 2008), which showed that an 80% user agreement out of 10 classifications was equivalent to a 5σ detection.

Users were also shown simulations, constructed by the PH science team using real Kepler light curves injected with synthetic transit signals. The shape of each synthetic signal was generated based on parameters pulled at random from a range of exoplanet radius and orbital period values (further details in Section 4.1. All simulations contained at least one injected signal. Initially, 40 Kepler targets with effective temperatures $T_{\text{eff}} < 4200$ and a representative variety of stellar radii, K-band magnitude, and activity level were chosen as simulation hosts. Combinations of hosts and synthetic transit parameters were used to create 4000 30-day simulation light curves. Of the 40 initially-chosen target stars, 24 remained in our M-dwarf sample after filtering out likely giant stars (Gaidos et al. 2016; Berger et al. 2020) (a process described in detail in Paper II), which left 2268 simulations. And of these simulations, finally, more than half were determined to be too difficult for users to identify through visual inspection, due to the extreme shallowness of signals produced by very small planets. Thus, only the 1428 simulation subjects containing synthetic transits from exoplanets larger than 1 $R_{\oplus}$ were retained for classification and analysis.

The manner in which the simulations were presented was indistinguishable from that of the real, unaltered data, including the display of stellar property metadata. Only after a user had classified the subject were they informed that it was a simulation, and shown where the synthetic transit signal(s) had been injected. The 1428 simulation subjects were classified a total of 149,921 times, with an average of 105 classifications each. About 85.8% of these classifications were done by registered users. The users who classified each simulation were split randomly into groups of ten. For each of these sets of ten users, unique IDs were generated for the subject and its corresponding synthetic exoplanet and signals. This resulted in an effective total of 15,233 simulations. These simulations allow for both an individual user’s performance and the recovery efficiency of PH as a system to be measured (Schwamb et al. 2012, hereafter S12), as discussed in Sec. 4.1 and Appendices A1 and A3.

Figure 2 provides several graphical metrics for gauging user interaction with the PH site, each of which offers its own insight into the user base and classifications used in this study. Fig. 2a, by clearly displaying the colossal number of classifications done by users who saw no simulations, succinctly illustrates the necessity for developing a sophisticated weighting scheme capable of quantifying the performance of all users, including those who interacted with no simulations. This is a significant departure from the methodology of the PH TESS analysis pipeline (Eisner et al. 2020a), and in a more general sense, how the users have interacted with the simulations, divided by color into registered and unregistered users. The excess of registered users with high numbers of simulation classifications is not unexpected; rather, it’s interesting that there are some users, while few, who enter the site and in one session classify enough light curves to have seen hundreds of simulations. Fig. 2b complements this by showing the total number of classifications done by users in these same simulation classification count bins, again divided by color into registered and unregistered users. Over all, though unregistered users outnumber registered users, the registered users classified significantly more data than the unregistered users, about 84.2%. Even among these registered users, however, Fig. 2d shows that many did not classify any simulations.

Fig. 2c breaks down the user population in bins of total classifications done, stacked for visual clarity. This view in particular gives some potential insight as to why the first bin of Fig. 2a is so large: If a user visited the PH site and classified only one light curve, it was most likely not a simulation, as evidenced by the simulated-to-real ratio of 5:4938 in the first bin of Fig. 2c. In this plot, we also see the clear pattern of users who classify more subjects naturally being the ones who classify more simulations, but there are a few notable exceptions visible—namely, one user who somehow classified over ten thousand subjects without encountering a single simulation. Investigation into this case is ongoing.

These simulations are an integral part of the user weighting process (see Sec. 3.2), as well as in determining the recovery effi-

---

7 Paper II discusses in more detail how this sample selection took place, as well as the post-classification pruning of the sample from 5625 stars down to the 3262 whose PH classification data function as the input for this work.
8 http://archive.stsci.edu/kepler
9 http://archive.stsci.edu
10 https://galaxyzoo.org
ciency of the PH system (Sec. 4), which is fundamentally necessary for calculating exoplanet occurrence rates in Paper II.

2.2 User classification outputs

There were 1,252,539 unique classifications of our sample, comprised of 1,856,573 markings and non-markings, made by 31,781 users between 18 September 2014 and 9 January 2016.

The “raw data” output by the PH backend was in the form of a .csv file containing all classifications done on all subjects by all users up to the date of the file export, including simulations. This file contained a row for every single user marking made (multiple markings made by the same user during a single classification of a subject constituted multiple rows), as well as rows indicating when a user saw a subject but chose not to mark anything. A full itemization of fields delivered by this file is described on the SATCHEL GitHub, including which fields are necessary as input for the algorithm. Of particular importance for the following discussion are: the classification ID (unique to each instance of a user viewing a subject); a subject ID (unique to each 30-day light curve segment); the location string of the raw .json file on the PH backend where the subject data was stored; and the “global x” values that denoted the bounds of the user mark relative to the beginning of the Kepler mission time baseline, beginning at Barycentric Julian Date (BJD)–2454833.0 (or NaNs if no marking was made).

The following were also either extracted or derived from the location string for each row: The Kepler Input Catalog (KIC) identifier of the target star to which the light curve subject corresponds; the Kepler “type,” indicating whether the target was observed during the main Kepler mission (“K1”) or the supplementary K2 mission (Osborn et al. 2016; LaCourse & Jacobs 2018), the latter of which were summarily filtered from our sample to leave only data from the Kepler prime mission; a Boolean indicating whether or not the subject is a simulation; and the synthetic exoplanet ID for simulations (or NaN if the subject is not a simulation). Note that a single synthetic exoplanet may generate multiple synthetic transit signals in a simulation subject, provided that it has an orbital period of less than about 15 days. Thus, there are unique identifiers for both the synthetic exoplanet (included in the classification file) and identifiers for the individual synthetic transit signals (not included in the classification file, but recorded elsewhere; see Sec. 3.2.1).

3 METHODS

For past analysis of the PH data, it was sufficient to calculate for each light curve subject a “score,” which measured the likelihood that the subject contained a significant feature (S12). If we were to apply this technique to the Kepler M-dwarf PH classifications, we would expect many of the features responsible for high scoring subjects would be either short-period signals produced by KOIs found in previous studies or signals from eclipsing binary stars, the most common source of false detections in Kepler exoplanet searches. Short-period KOIs provide a useful tool for measuring the recovery efficiency of the SATCHEL pipeline (see Sec. 4.2), as well as for comparing the results of our statistical occurrence rate study in Paper II to other contemporary results. However, to find longer-period signals that are more difficult or impossible to find using standard automated searches, a straightforward way to filter known signals out of a subject while preserving a measure of the significance of remaining features is required. Thus, rather than scoring each subject, the pipeline scores individual features. S12 does not take this approach in their study; the PH TESS pipeline does, notably, score individual features, but using a very different weighting scheme in their algorithm (Eisner et al. 2020a). In particular, Eisner et al. (2020a) do not measure the performance of users who have not classified simulations.

The general structure of the SATCHEL pipeline is as follows. Features of interest, which we call metafeatures, are identified by consolidating user marks that coincide on a given subject, as described in Sec. 3.1. Then, following the strategy of S12, “weights”

\[^{11}\] The backend data exports for PH TESS are the same format, with only very minor differences.
for users who classified simulations are assigned, based on skill at identifying synthetic transits (Sec. 3.2). Using the resulting weights (and dummy weights equivalent to zero upweight and zero down-weight for users who did not classify simulations), preliminary scores are calculated for all metafeatures (Sec. 3.3). Then, using these initial distributions as “seeds,” the user weights and metafeature scores are iteratively coupled, as described in Sec. 3.4. The weights for all users are adjusted based on user agreement with the majority on highly-scoring features, and scores are subsequently adjusted based on the new weight distribution. The coupling is iterated until the user weight distribution converges with a stable mean weight of 1.0, from which the final metafeature scores are calculated. Figure 3 illustrates how the individual pieces of the PH analysis pipeline are designed to work together, from parsing the data exported by the PH website backend up to obtaining a fully-vetted list of Planet Hunters Objects of Interest (PHOIs). The following subsections detail each step of this process.

### 3.1 Metafeatures

The list of metafeatures to be scored is constructed by extracting features directly from the marks that users have made on the light curve subjects. More obvious dip features are marked by multiple users, and these marks are consolidated to represent a single metafeature. Thus, we do not score any feature more than once. This consolidation is done using the pipeline piece named build-metafeatures.py (available in the “aux” folder of the SATCHEL GitHub).

Because SATCHEL is intended to be useful for assessing the significance of signals in data from other projects, which may have unique criteria designating which marks should or should not be ingored or how marks should be consolidated, the process of identifying metafeatures is considered technically auxiliary to the main pipeline. The Kepler data on PH, for instance, contain many data gaps of varying duration, usually on the order of days, caused by such things as scheduled quarterly rolls of the spacecraft, sudden shifts to safe mode due to cosmic ray impacts, and spacecraft rotation toward Earth for data downlinking (Stumpe et al. 2012). Users frequently mark these gaps, or mark anomalous features at the be-
beginning or end of the gaps. To filter out these marks, gaps in the Kepler light curves were identified by NaN entries for the PDCSAP flux. Any user mark for which more than half the mark duration lies inside a gap was discarded.

Very “wide” user marks were also discarded. In the PH data, there are a significant number of these wide user marks, indicating dip features with very long durations (see Fig. 4). The duration of an exoplanet transit increases with both stellar radius and $\sqrt{a}$, where $a$ is the semi-major axis of the planet’s orbit. However, the stellar sample for this study has been filtered of giant stars (see Paper II), and the geometrical probability of observing a transit is inversely proportional to $a$. In Fig. 4, a peak in the distribution of user mark durations can be seen at about six days. However, the geometrical probability of observing the transit of a planet with the corresponding semi-major axis is extremely small, on the order of $10^{-6}$. Thus, we expect all or nearly all of the markings with such

---

**Figure 3.** Flowchart to illustrate the general process of the PH pipeline, along with the subsequent steps necessary to reach a final PHOI list. Steps in orange bubbles represent steps of the process that require human involvement. Blue and red steps are fully automated, the latter of which comprise the iterative portion of the SATCHEL pipeline. Purple steps are vetting, which is described in Paper II.
MNRAS 000, 1–20 (2021)

The SATCHEL pipeline

Figure 4. Distribution of durations of user markings on non-synthetic M-dwarf lightcurves, from raw classification data (not metafeatures). The maximum width users were allowed to draw in the PH interface was about 6.74 days.

long durations to be identification mistakes. Visual inspection of a subset of these marks supports the assumption that most, if not all, correspond to cases in which users have either brought attention to a long data gap or mistaken stellar activity dips (such as those due to starspots) for exoplanet transit signals. These markings are filtered out using an adjustable “duration cutoff” hyperparameter, for which we found the optimal value was about 2.5 days (see Appendix A). This duration cutoff may not be applicable to datasets from other projects, or a minimum duration cutoff may be more contextually applicable. This can be changed easily, if desired.

Similarly unique to the PH dataset, each 30-day subject is a fragment of a longer light curve with a designated quarter, and the light curve itself from a designated Kepler target. The quarter and Kepler ID corresponding to each metafeature are thus recorded in the metafeature list for post-pipeline reference, but may not have any analogous function for other datasets.

The general methodology for consolidating user marks into metafeatures is as follows. Any user mark that is not filtered out by either the duration cutoff or data gap check is assumed to indicate the presence of a feature that should be scored. If only one user marked the feature, the mark is recorded exactly as is. Where multiple user marks overlap on a subject, they are assumed to indicate the same feature if the midpoints of each mark differ by no more than a specified “midpoint tolerance.” For a given user mark, we find all marks made by other users that satisfy the midpoint criterion, forming a “match list” for the given mark. Then, for each mark in the match list, we obtain new match lists. Marks with identical match lists are consolidated to construct the resulting metafeature, by averaging the bounds of all marks in the list. No user mark contributes to more than one metafeature. A user mark that matches more than one separate feature is included in the consolidation of neither, comprising its own metafeature, to prevent smearing separate features closer together. Finally, each metafeature is assigned a unique feature ID.

Users often mark non-planetary-like features, including data gaps, as mentioned above, as well as transits from EBs and brightness variations associated with intrinsic stellar variability. Though we do attempt to filter some of these marks out of the metafeature list, it is not strictly necessary; a well-constructed scoring process ensures that, if only a small number of users with low weights have marked them, those metafeatures will be dispensed from the pipeline with a very low score.

Two hyperparameters are adjustable in the construction of the list, as defined above, with the following default values:

- Midpoint tolerance: 1.2 days.
- Duration cutoff: 1.0 day.

The default values, which were used to produce the results discussed throughout this work, were chosen based on the optimization study described in Appendix A. The resulting metafeature list, for the PH dataset, contains 486,504 unique features. Figure 5 shows the PH user marks on the Kepler Quarter 17 light curve for KIC 9772531, to illustrate how user marks are culled and consolidated into metafeatures for scoring.

3.2 User weighting

The next step in the process is calculating the user weights that will seed the first calculation of metafeature scores, based on matches to the synthetic transit signals in simulations. It’s important to note, at this stage, that these weights can be calculated only for those users who did see at least one simulation. Due to the random nature of when users were shown simulations, a user who visited the website and classified only one or two light curves before leaving may or may not have seen one. Users who have never classified a simulation comprise about 54% of the total M-dwarf user base, and these users generated about 23% of the total M-dwarf classifications (see Fig. 2a).Weights for such users are calculated iteratively later in the pipeline, by the consensus agreement method described in Section 3.4.

The general strategy for calculating user weights from simulations, adapted from S12, is as follows: All users begin with a weight of 1.0. A user’s weight is increased if they make a mark that matches a synthetic transit, detailed in Section 3.2.1. A decay function, described in Sec. 3.2.1 and Appendix A, adjusts the relative increase in weight for multiple correctly marked synthetic transits in the same light curve. The user’s weight is decreased if they mark spurious features, detailed in Section 3.2.2. The user’s weight is unaffected if they mark nothing at all on a simulation. All user weights are then normalized by the mean of the weight distribution, as described in Section 3.2.3. These processes are all contained in the pipeline piece user-weighting.py

3.2.1 Upweighting

A simulation light curve may contain multiple synthetic transit signals, and each user may make multiple user marks on a simulation. For an individual user, marks that match synthetic signals are identified by two criteria:

- The midpoint of the mark must lie within the specified tolerance of the midpoint (1 day, by default) of the synthetic signal;
- The mark must have non-zero overlap with the duration of the signal.

In the event that a user has made more than one mark that satisfies these conditions, the one with the closest midpoint to that of the synthetic signal is designated as the match, provided that it has not already been designated a match to another signal. No user mark is designated a match for more than one synthetic signal. This process is repeated for all synthetic signals in the simulation.

The user’s weight is increased for marks that match synthetic
signals. For each signal, a quantity called signal ID completeness (SIDC) is calculated, defined as the ratio of the number of users who correctly identified the signal to the number of users who saw the simulation containing it. If a small fraction of users identified the signal correctly, then the SIDC for that signal is small; if the signal was identified correctly by most users, then the SIDC is near unity.

The “raw” upweight for correctly identifying a synthetic signal is given by

$$\text{raw upweight} = \frac{1.0}{\text{SIDC}} - 1.$$  

(1)

After two signals have been found in a light curve, it is reasonable to assume that the user knows where to look for others due to the periodic nature of exoplanet transits. Occasionally, users will even stop marking additional signals after they’ve found a sufficiently large number of them in a subject, seemingly content to have noted the presence of interesting features in general. Thus, allowing large, compounded increases in user weight is not necessarily an accurate representation of a user’s skill. To compensate for this, all of the user’s raw upweights from a given simulation are sorted from largest to smallest, and then each upweight is scaled by a decay function. That is,

$$\text{upweight} = (\text{raw upweight}) \cdot f_{\text{decay}}(k),$$  

(2)

where $k$ is the ordered index of the upweight in the sorted list, starting at $k = 0$. The form of the decay function is

$$f_{\text{decay}}(k) = \begin{cases} g(k) & \text{if } k \leq 3, \\ g(3) \cdot e^{-1.5(k-3)} & \text{if } k > 3, \end{cases}$$  

(3)

where

$$g(k) = 1.03 - 0.03 \cdot e^{1.07 k}. \tag{4}$$

The highest upweight is given to the user in its entirety, the second highest is given at about 94% of its raw value, the third is given at about 78%, and the fourth at about 29%. Any further synthetic signals identified in the light curve decrease with an inverse exponential scaling.

The coefficients in Equation 4 were chosen through author trial and error. Appendix A2 discusses details of this decay function, and compares user weight seed distribution results obtained using a variation with a cutoff, and no decay function whatsoever. We note that, for projects seeking to identify aperiodic or irregular signals, simply setting the decay function to unity will allow all raw upweights to be applied, unaltered, to users’ weights.

### 3.2.2 Downweighting

A user’s weight is not decreased if the user fails to find a synthetic signal. The normalization process following the upweighting and downweighting effectively lowers the weights of users who have made no markings, so to explicitly downweight for failure to identify signals is to punish users redundantly. Instead, downweights are given only for incorrect markings on synthetic lightcurves, which indicate spurious features. For each marking a user made that does not correspond to a synthetic signal, the magnitude of the user’s downweight is increased by 1.

Note that this methodology differs considerably from that of the PH TESS pipeline, in which both incorrect markings and failure to find synthetic signals affect a user’s weight explicitly. We consider the method of Eisner et al. (2020a) a viable alternative strategy, and an interesting opportunity for potential future exploration.
### 3.2.3 Normalization

After upweights and downweights have been calculated for all users who saw synthetics, both sets of numbers are normalized so that the mean of each distribution is 1. That is, if \( n \) is the number of users who have seen synthetics, then the normalized upweight for user \( i \) is

\[
\text{(norm. upweight)}_i = \frac{(\text{upweight})_i}{\sum_{n} (\text{upweight})_n},
\]

and likewise for the user’s downweight. The ratio of the user’s upweight and downweight form their “combined” weight, and finally, the distribution of combined weights is normalized as well.

For the PH users in this sample, the weight of a user who saw simulations but received neither any upweights nor downweights from their classifications is \( \approx 0.7874 \) after normalization. This “base weight” is used as the dummy weight for users who have not seen synthetics in the first round of metafeature scoring. At this stage, the user weight table is also refined to include lists of which metafeatures were classified by each individual user. This saves a significant amount of time when the weighting is coupled to the scoring.

### 3.3 Metafeature scoring

As in S12, scores for each feature, including both synthetic and non-synthetic signals, are calculated using the basic formula

\[
s_{k,j} = \frac{1}{W_k} \sum_i w_i \cdot j^*,
\]

where \( i = \) all users who viewed the subject containing \( k \).

Like S12, a vote \( j \) score is calculated for each metafeature \( k \), where \( j_i = \) “yes” or “no” indicates that the metafeature either was or was not marked by user \( i \), respectively. \( w_i \) is the weight of user \( i \), and \( W_k \) is defined as the sum of user weights for all the users who classified the subject containing metafeature \( k \). A user who classified the subject but did not mark the metafeature in question has effectively voted “no” for that metafeature.

All users contribute to this scoring process, which is done by the pipeline piece called score-seeds.py. As the score for each metafeature is calculated, the users who have classified the light curve containing that metafeature are recorded in the scoring dataframe, divided into a usersyes group (those who identified the feature) and a usersno group (those who did not) for easy access later. Much like the recording of the IDs of metafeatures classified by each user, this saves time in the coupled process.

### 3.4 Coupled scoring-weighting process

The initial scores calculated for each metafeature act as seeds, which are used to further adjust the weight of each user based on the user’s agreement with majority votes. This adjustment is mathematically adapted from S12. For user \( i \) with weight \( w_i \), a new weight is recalculated:

\[
w_{i,\text{new}} = \frac{A}{N_i} \sum_k s_k (j \text{ for metafeature } k \text{ chosen by user } i),
\]

where \( k = \) all metafeatures reviewed by user \( i \).

Figure 6. Evolution of convergence quantities over the course of the coupled weighting-scoring process. Top: The mean magnitude of the user weight differences from the first iteration up to convergence, when the mean magnitude difference reaches \( < 10^{-3} \). Bottom: The number of users whose weights are still changing by more than 0.1 units after each iteration.

3.4.1 Iteration history

The process is repeated until the mean absolute difference in weights between iterations is less than \( 10^{-3} \), at which point the weights are considered to have converged. For the PH classifications, convergence was achieved after 24 iterations. Figure 6 shows the mean of the distribution of absolute weight differences as a function of iteration number. The large, early drop corresponds to the most significant change in the distributions of both user weights and metafeature scores. Over these first iterations, the weight distribution is narrowed from a range between roughly 0.01 and 22 to a range of about 0.01 to 1.5. Between iterations 3 and 15, the majority of user weights change negligibly (bottom panel of Fig. 6). During this period, the weights of a small number of users become arbitrarily closer to zero, and the weights of another small subset of users (who have classified only simulations, no real data) increase liberally. The latter continues until near the last few iterations. These two subsets of users are visible in blue in the top pane of Figure 7, which shows how the distributions of weights and scores change from initialization to convergence of the coupled weighting-scoring process.

The consensus agreement weighting method implemented in this coupled process dominates the final weight and score distributions, the latter to a stronger degree than the former. We performed

\[\text{if the user voted “no” for the metafeature, it is the “no” score for that feature that is included in the sum instead. The normalization factor } A \text{ is defined to force the mean of the weight distribution back to unity. Unlike S12, } N_i \text{ is defined as the number of metafeatures classified by user } i, \text{ rather than the number of light curves.}

In practice, the normalization factor \( A \) is applied after all user weights have been adjusted for vote agreement. The entire distribution is then renormalized, and the metafeature scores recalculated based on the new user weights, using the same method as in Eq. 6.

The process is repeated until the mean absolute difference in weights between iterations is less than \( 10^{-3} \), at which point the weights are considered to have converged. For the PH classifications, convergence was achieved after 24 iterations. Figure 6 shows the mean of the distribution of absolute weight differences as a function of iteration number. The large, early drop corresponds to the most significant change in the distributions of both user weights and metafeature scores. Over these first iterations, the weight distribution is narrowed from a range between roughly 0.01 and 22 to a range of about 0.01 to 1.5. Between iterations 3 and 15, the majority of user weights change negligibly (bottom panel of Fig. 6). During this period, the weights of a small number of users become arbitrarily closer to zero, and the weights of another small subset of users (who have classified only simulations, no real data) increase liberally. The latter continues until near the last few iterations. These two subsets of users are visible in blue in the top pane of Figure 7, which shows how the distributions of weights and scores change from initialization to convergence of the coupled weighting-scoring process.

The consensus agreement weighting method implemented in this coupled process dominates the final weight and score distributions, the latter to a stronger degree than the former. We performed

\[\text{13 Animations of the changes in PH user weight distribution, weight difference distribution, and score distribution over the course of the entire process are publicly available on the PH pipeline GitHub.}\]
Figure 7. Comparison of user weight (top) and metafeature score (bottom) distributions at the beginning and end of the coupled iteration. The highest subset of user weights is expanded in the middle panel. Both the user weights and the metafeature scores undergo the most dramatic changes during the first few iterations. Over subsequent iterations, subsets of users break away from the main distribution of weights (one of which approaches zero, and the other of which increases liberally, comprised of users who classified no real data), while the weights of the vast majority change negligibly.

A test run of SATCHEL using only weights from simulations (with uniform dummy weights for users who saw no simulations) and another test run using entirely uniform dummy weights as seeds for the consensus agreement weighting process, and compared the resulting weight and score distributions to each other and to the final distributions when both methods are used. We found that, when users are weighted based only on simulations, SATCHEL recovers slightly more synthetic signals (as discussed below in Sec. 4), while the consensus-only run recovered slightly more KOI signals. Whether one or the other is used, or both, the differences in recovery are well within Poisson errors.

Converging on only the mean stability of the user weight distribution, rather than the stability of all the user weights individually, allows for the possibility that a coincidental pseudo-stability may be reached, in which one or more weights are still increasing by a large amount, but other weights are decreasing by a commensurate amount, balancing out the distribution as a whole. One of the outputs of each iteration is a count of how many user weights are still undergoing “large” changes. In the case of the PH data, changes...
of more than about 0.1 units are considered large; this threshold may be different for other data sets. Whether or not such large changes are still occurring, the iteration is currently built to accept the convergence. For the PH data, convergence was achieved while the number of users with weights still changing by more than 0.1 units was 33. Those weights belonged to users who had classified only simulations (typically only one subject), and no real data, so the scores of all the real metafeatures were already stable, affected negligibly by such changes.

Another circumstance which may potentially arise is that of a resonance loop between the weights and scores, in which the weight adjustment causes a change in metafeature scores that exactly reverts the weights to their prior state after the next adjustment, and so forth. The mean of the weight distribution is compared to that of the iteration before last (as soon as the number of iterations elapsed is high enough to do so), and the same is done for the mean of the score distribution. If both means are the same for three alternating past iterations within a tolerance of $10^{-3}$, a set of small, random perturbations are added to the metafeature scores to attempt to break the resonance loop, and this attempt is noted in the terminal output. For the PH data, a resonance loop did not arise, so random perturbations were not necessary.

Once convergence is reached, the final states of both the user weights and resulting metafeature scores are saved, along with several plots and animations that may be useful. If the loop is exited prior to convergence, plots and animations constructed using weight and score data from iterations up to that point are saved, along with the state of the weights and scores from the last complete iteration. A log file containing potentially significant quantities for each iteration (including, among other things, the mean magnitude weight difference, the means of the weight and score distributions, and the total time elapsed) is also saved as output.

### 4 MEASURING PERFORMANCE

In the following subsections, recovery is assessed against both the simulations (Sec. 4.1) and against the known population of KOIs (Sec. 4.2). Both types of light curve subjects were displayed the same way in the PH interface, to minimize potential user biases in classification, and other than the use of simulations in calculating the user weight seeds, both types of subjects were treated identically throughout the rest of the pipeline. A known signal was considered recovered if a metafeature that overlapped the signal by at least 50% had a final score of 0.5 or greater. This score cutoff was chosen to balance maximum recovery of known signals with minimum false positive contamination (see Appendix A3 for details). Out of 486,504 metafeatures, 88,171 met this requirement, 3522 of which correspond to transits of known KOIs and 13,616 of which correspond to synthetic transit signals in simulations.

Another circumstance which may potentially arise is that of a resonance loop between the weights and scores, in which the weight adjustment causes a change in metafeature scores that exactly reverts the weights to their prior state after the next adjustment, and so forth. The mean of the weight distribution is compared to that of the iteration before last (as soon as the number of iterations elapsed is high enough to do so), and the same is done for the mean of the score distribution. If both means are the same for three alternating past iterations within a tolerance of $10^{-3}$, a set of small, random perturbations are added to the metafeature scores to attempt to break the resonance loop, and this attempt is noted in the terminal output. For the PH data, a resonance loop did not arise, so random perturbations were not necessary.

Once convergence is reached, the final states of both the user weights and resulting metafeature scores are saved, along with several plots and animations that may be useful. If the loop is exited prior to convergence, plots and animations constructed using weight and score data from iterations up to that point are saved, along with the state of the weights and scores from the last complete iteration. A log file containing potentially significant quantities for each iteration (including, among other things, the mean magnitude weight difference, the means of the weight and score distributions, and the total time elapsed) is also saved as output.

### Table 1. Performance statistics for the SATCHEL pipeline, with optimal hyperparameters (duration cutoff = 1.0 day, midpoint tolerance = 1.2 days).

| Statistic                   | Value   |
|-----------------------------|---------|
| Area under ROC curve        | 0.288   |
| Area under PR curve         | 0.181   |
| Average precision score     | 0.445   |

Figure 8. The Receiver-Operating Characteristic (ROC) and Precision-Recall (PR) curves for the optimized SATCHEL pipeline. The area under each curve was calculated using the trapezoidal rule.

Using the sklearn Python package. We consider the score threshold as a binary classifier, such that a metafeature in a simulation subject or KOI subject assigned a score higher than the threshold is defined as a “positive.” Such a metafeature that matches a ground-truth signal is then a true positive, while one that does not is a false positive. A false negative may then be defined as a ground-truth signal that was either not identified as a metafeature or a matching metafeature which was assigned by the pipeline a score lower than the threshold value. “True negatives” would, in principle, be features in subjects that were either not identified as metafeatures or non-matching metafeatures that were assigned scores lower than the threshold value. As the number of unmarked features is not quantifiable, neither is the number of true negatives.

The ROC curve is constructed by plotting the fraction of true positives out of the positives vs. the fraction of false positives out of the negatives, at various threshold values. The precision is, effectively, the pipeline’s ability to correctly sort metafeatures into positives and negatives. The recall is the pipeline’s ability to correctly identify all ground-truth signals. For this binary classifier, the ROC and PR curves are shown in Figure 8. The area underneath each curve (AUC) can be used as a single-number metric to gauge pipeline performance. The closer the AUC is to unity, the better the performance. The ROC AUC for SATCHEL is 0.288; the PR AUC is 0.181. We also calculate the average precision score, for ease of performance comparison to future pipelines. A summary of these values and the hyperparameters that optimize them is given in Table 1.

Generally, the ROC curve for a classifier passes through the
points (0, 0) and (1, 1), indicating that, with the strictest threshold, no false positives are recovered but neither are any true positives, and with most lenient threshold, all true positives are recovered, as well as all false positives. For the SATCHEL pipeline, this cannot be achieved, because there is no threshold for which all simulations and KOI transits were found by PH volunteers. If unfound transit signals are removed from the analysis, the ROC curve behaves in the standard way, as shown in Figure 9.

S12 found that, rather than exoplanet radius, the signal depth \( \delta = R_p^3/R_*^2 \) (where \( R_p \) is the radius of the planet and \( R_* \) is the radius of the host star) is the true observable for PH volunteers. Though our sample is comprised of only M-dwarf stars, the range of stellar radii is still significant, spanning 0.13 to 0.78 \( R_\odot \) (Guidos et al. 2016; Berger et al. 2020). Furthermore, M-dwarf light curves can be afflicted with noise from sources such as sunspots, flares, and other magnetic activity to an extent that is generally more extreme than stars of other spectral types (Nielsen et al. 2013). Signal-to-Noise Ratio (SNR) may, therefore, be even more directly related to the identifiability of transit signals by PH volunteers. In the below subsections, we consider SNR for each individual transit only, calculated as the ratio of the signal depth \( \delta \) to the Combined Differential Photometric Precision (CDPP, Koch et al. 2010; Christiansen et al. 2012), omitting the factor of \( \sqrt{n_{tr}} \) that is commonly included in analysis of *Kepler* light curves to account for the strengthening of the phase-folded signal containing a number \( n_{tr} \) of transit signals over the duration of observation (Howard et al. 2010; Christiansen et al. 2012). The PH interface did not offer such algorithmic enhancement of signals, and thus we would not expect such a factor to be applicable in any potential detection trends. It is likely that the presence of multiple signals in a single subject increases the detectability of individual signals, as discussed in Sec. 3.2.1, but quantifying this effect may be either impossible or prohibitively difficult, as behavior varies greatly from one volunteer to another. If this bias toward short-period signals exists, its effect on pipeline completeness is limited only to simulations and KOIs with periods shorter than about 15 days, due to the 30-day time baseline of PH subjects.

### 4.1 Simulation recovery

As mentioned in Sec. 2.1, the synthetic signals that populate the simulations were constructed from a range of exoplanet radius and orbital period values. Radius values were pulled at random from a distribution between 0.5 \( R_\oplus < R_p < 12 R_\oplus \) and orbital periods were pulled from a distribution between 0.7 days \(< P < 1000 \) days.

Neither eccentricity nor orbital inclination were varied; orbits were circular only, and transits passed perfectly along the diameter of the star. For a given synthetic exoplanet, the orbital phase was drawn randomly from the range 0 to 2\( \pi \) until at least one transit occurred within the 30 day window of the simulation subject. The duration and flux loss of the transit signal were calculated based on the orbital period, exoplanet radius, and host star radius, and the flux loss was subtracted directly from the PDCSAP flux values of the stellar light curve. No artificial noise was added. The number of synthetic signals corresponding to exoplanet radii and periods falling in a range of bins is listed in Table 2.

Figure 10 displays recovery rates across the full range of physical properties of the simulations. Note that the orbital period binning

---

**Figure 9.** The Receiver-Operating Characteristic (ROC) and Precision-Recall (PR) curves for the optimized SATCHEL pipeline, not including transit signals that were not found by PH volunteers. The area under each curve was calculated using the trapezoidal rule.

**Figure 10.** Recovery of synthetic signals as a function of both exoplanet radius and orbital period, in which a signal is considered recovered if a matching metafeature was assigned a score of 0.5 or higher by the PH pipeline. Grey bins labeled “N/A” correspond to bins in which there were no synthetic signals to be recovered. Binning in both radius and period are logarithmically even.
Table 2. Recovery frequency of synthetic exoplanet transit signals. Note that the binning here is not identical to that of Fig. 10.

| Planet Radii ($R_{⊕}$) | 0.7 ≤ $P$ < 200 | 200 ≤ $P$ < 400 | 400 ≤ $P$ < 600 | 600 ≤ $P$ < 800 | 800 ≤ $P$ < 1000 |
|-------------------------|-----------------|-----------------|------------------|-----------------|------------------|
| 1 ≤ $R$ < 2             | 848/2408        | 324/883         | 289/594          | 368/666         | 274/523          |
| 2 ≤ $R$ < 3             | 2221/3160       | 1015/1272       | 590/689          | 564/664         | 491/576          |
| 3 ≤ $R$ < 4             | 668/764         | 127/142         | 95/116           | 169/194         | 131/141          |
| 4 ≤ $R$ < 5             | 717/748         | 359/378         | 110/124          | 135/148         | 88/88            |
| 5 ≤ $R$ < 6             | 1078/1133       | 324/326         | 208/228          | 135/135         | 172/172          |
| 6 ≤ $R$ < 7             | 216/231         | 55/57           | 39/39            | 22/22           | 26/26            |
| 7 ≤ $R$ < 10            | 466/488         | 220/235         | 89/108           | 87/94           | 81/81            |
| 10 ≤ $R$ < 15           | 496/516         | 97/97           | 124/124          | 28/28           | 70/71            |

Figure 11. Recovery of synthetic signals as a function of radius (left, in $R_{⊕}$), depth (middle, in parts-per-thousand), and SNR (right), all marginalized over orbital period by sum, with Poisson errors. SNR is calculated per-signal (see discussion in Sec. 4). Binning in radius, depth, and SNR are logarithmically even.

Figure 12. Recovery of synthetic signals as a function of orbital period, marginalized over exoplanet radius by sum. Binning in orbital period is logarithmically even.

Since the signal depth is proportional to the square of the planet radius, larger exoplanets generally produce deeper dips, which are more obvious to the human eye. Conversely, the small dips from synthetic exoplanets with radii < 2 $R_{⊕}$ are easily lost in the noise from typical stellar activity. The smooth, monotonic evolution of the recovery rate as a function of radius and its proxies, particularly SNR which is arguably the true observable, is exactly what we would expect from visual inspection of the Kepler data.

Above 2 $R_{⊕}$, the total recovery of all synthetic signals is about 85.8%. Above 2.5 $R_{⊕}$, the total recovery is about 91.0%.

Figure 12 displays recovery of synthetic signals as a function of orbital period. There appears to be a trend of decreasing recovery rate with increasing orbital period for short-period signals, with a slight increase or possibly stable rates for signals with orbital periods ≥ 30 days. The possible statistical significance of this trend is explored in detail in Paper II.

4.2 KOI recovery

The Kepler M-dwarf dataset has been mined extensively for short-period exoplanets (see, e.g., Dressing & Charbonneau 2015; Gaidos et al. 2016; Meyer et al. 2018; Hardegree-Ullman et al. 2019), and thus, the known population of KOIs is a valuable recovery metric to explore. We here use the NASA Exoplanet Archive\footnote{https://exoplanetarchive.ipac.caltech.edu/} (NEA) as in Fig. 10 is logarithmically even, rather than the linear binning of Table 2. Figure 11 shows the recovery fraction of signals from synthetic exoplanets as a function of radius, transit signal depth, and SNR, all marginalized over orbital period by sum, and with Poisson counting errors shown. The most prominent trend, clear from all binnings, including that of Table 2, is that recovery fraction increases with exoplanet radius. This is an expected trend;
a repository for KOI properties such as planetary radius and orbital period, as well as host star radius and spectral type. We exclude KOIs that have been classified as false positives, and retain both confirmed and candidate exoplanets. Throughout this discussion, we make no distinction between the latter two.

There are 112 KOIs listed in the NEA for our stellar sample, with confirmed and candidate exoplanets spanning a range of radii from $0.08 < R/R_\oplus < 25.0$, and a range of orbital periods from $0.53 < P/\text{day} < 628$. A significant fraction of these KOIs reside in a regime of parameter space unrepresented by our simulations, particularly those with very small radii, which makes direct performance and recovery comparisons difficult. In an effort to retain the maximum number of KOIs for statistical completeness while preserving the comparability of the KOI study to the simulations, the two smallest KOIs with the smallest SNR values (KO6705.01 and KO4777.01), have been excluded from the following study. This limits the parameter space to about $0.4 < R/R_\oplus < 25.0$ in exoplanet radius. Notably, KOI KO6705.01 was recovered by PH volunteers, despite its shallow signals and low SNR (0.045 ppt and about 0.25, respectively). This is likely owing to its short period, only 1.0 day, which guarantees many transit signals even in a single subject.

Figure 13, like Figure 10, displays recovery rates across the truncated range of physical properties of the M-dwarf KOIs. Though the small number of KOIs leaves the plot quite underpopulated, a trend of increasing recovery with planetary radius does seem to be visible. The trend appears more prominent in Figure 14, which shows the recovery rate as a function of exoplanet radius and its proxies, marginalized over the orbital period. A corresponding trend of increasing recovery fraction with signal depth and SNR is also visible. These, again, are trends that we expect, and the similar trend shapes in Figures 11 and 14 reinforce our confidence in both the trends themselves and the effectiveness of the simulations as a tool to measure realistic user and pipeline performance, particularly for exoplanets larger than $2 R_\oplus$. Recovery trends with radius between values of $1 - 2 R_\oplus$ are less robust in comparison, with significantly smaller recovery rates for KOI signals than synthetic signals, the cause of which remains to be explored in future work.

Recovery of KOI signals as a function of orbital period is shown in Figure 15, with Poisson errors. Above $2 R_\oplus$, in the radius range that the simulation study suggests PH volunteers can reliably detect, KOIs are quite sparse, and there are none with periods longer than about 77 days. We are thus unable to make a strong statement about trends in KOI recovery with orbital period, though our results do appear to be loosely consistent with the possibility of a stable recovery rate for signals with periods longer than $\gtrsim 30$ days. For KOIs with short periods, however, a trend seems to suggest increasing recovery with orbital period, the reverse of the weak trend seen for short period signals in Fig. 12. Whether this is due to the large Poisson uncertainties in the KOI sample or a fundamental difference between the two samples in the physical properties that govern detectability, this discrepancy may be worth exploring in future citizen science studies.

For KOIs larger than $2 R_\oplus$ recovery was 72.1%, and 94.8% for KOIs larger than $2.5 R_\oplus$.

### 4.3 Completeness

For any statistical use of the SATCHEL pipeline, a correction must be applied to account for the tendency of some types of signals to be missed more frequently than others. In the case of the Kepler data, as we have seen in Sections 4.1 and 4.2, the signals most frequently missed are shallow signals with low SNR, and the increase in recovery performance of the pipeline appears to be a smooth, monotonic trend as SNR increases. There may also be one or more complex trends in pipeline performance with orbital period, but the trends implied for simulation and KOI recovery seem to contradict each other, as seen in Figs. 12 and 15. Due to the small number of KOIs in the longer period regime, we are currently unable to make strong statements about this contradiction or about pipeline completeness as a function of orbital period. Constructing a completeness function that fully captures these intricacies is beyond the scope of this paper, as our primary goal herein is to offer SATCHEL as a generalized tool. We thus refer the reader to Paper II for a more detailed and specific discussion.

Instead, we include here an empirical treatment of completeness only for SNR, as an illustrative example. Pipeline performance trends with SNR for simulations and KOIs, within the range of parameter space $1 \geq R/R_\oplus < 15$, are in robust agreement, as seen in Figs. 11 and 15. With the same binning as shown in Fig. 12, recovery fractions for each bin were used as data points to fit a generalized logistic function of the form

$$Q(\text{SNR}) = \frac{1}{\left[1 + C e^{-\beta(\text{SNR} - \mu)}\right]^{1/\nu}},$$

(8)

with lower and upper asymptotes of 0 and 1, respectively. A nonlinear least squares fit (using the curve_fit function of Python’s scipy.optimize package) yields constants

$$C = 7.61 \times 10^{-5}, \quad \beta = 0.115, \quad \mu = 1.01, \quad \nu = 3.62 \times 10^{-5}$$

(9)

for the fit, which is shown overlaid with recovery fraction for SNR bins in Figure 16. Marginalization over orbital period in Fig. 16 was done over the mean, rather than the sum; however, choice of marginalization does not noticeably alter either the recovery trend or the fit.

Taken as the pipeline completeness function, Eq. 8 could be
Recovery fraction
KOI signal recovery as a function of planet radius and proxies

Figure 14. Recovery of KOI signals as a function of radius (left, in $R_\odot$), depth (middle, in parts-per-thousand), and SNR (right), all marginalized over orbital period by sum, with Poisson errors. SNR is calculated per-signal (see discussion in Sec. 4). Binning in radius, depth, and SNR are logarithmically even.

KOI signal recovery as a function of orbital period

Figure 15. Recovery of synthetic signals as a function of orbital period, marginalized over exoplanet radius by sum. Binning in orbital period is logarithmically even.

Synthetic signal recovery as a function of single-transit SNR

Figure 16. Simulation recovery interpolated from $1 < R/R_\odot \leq 15$ and $0.7 < P/\text{days} \leq 1000$ as a function of SNR, marginalized over orbital period. The sigmoid (in green) was fit to the recovery fractions in each bin (connected in blue). Fit values for coefficients are shown (see text). Binning in SNR is logarithmically even.

used to interpolate the pipeline completeness corrections for arbitrary SNR bins, or used in conjunction with other correctional factors (such as those addressing orbital geometry constraints and the Kepler observation window) (see, e.g., Foreman-Mackey et al. 2016; Burke et al. 2015) to compute global occurrence rates for single exoplanet signals, given the number found in each bin after fully vetting the pipeline results, rejecting false positives, and correcting for reliability (Bryson et al. 2020a,b). These tasks comprise the majority of Paper II.

5 SUMMARY AND DISCUSSION

We have presented SATCHEL, a pipeline capable of analyzing the 1,253,050 M-dwarf light curve classifications done by users of the Planet Hunters citizen science project. PH users classified both real Kepler data and simulations constructed by injecting real Kepler light curves with synthetic signals corresponding to exoplanets occupying a representative range of parameter space in exoplanet radius and orbital period. The classifications were comprised of either bounds of a marked region in the time-series data within which a user believed they saw a significant feature, or a non-marking, indicating that a user saw the time-series data but did not believe it contained any significant features. The SATCHEL extracts the features marked by users, consolidates them into metafeatures where possible, and calculates, on a signal-by-signal basis, scores for each that represent agreement among users of that feature’s significance, and by proxy the likelihood of that signal to be a legitimate exoplanet transit signal. The score calculation is a form of user-weighted average, done first using weights assigned to users based on their performance at identifying synthetic exoplanet signals, with dummy weights for all users who did not see simulations, and done later using a coupled, iterative scheme which adjusts user weights for all users based on majority agreement with high-scoring signals and subsequent score recalculation. The coupled adjustments continue until the weight and score distributions are both stable with a mean user weight of unity. This process is computationally intensive, but necessary due to the significant fraction of our data that has been classified by users who saw no simulations.

We performed a recovery study, defining a known transit signal as recovered if it met matching criteria with a metafeature assigned a
final score of 0.5 or higher by SATCHEL. For signals corresponding to KOIs and synthetic exoplanets over 2 $R_\oplus$, about 72% of KOI signals and about 86% of synthetic signals were recovered. At least one transit signal was recovered from every KOI, even those with the smallest SNR, corresponding to exoplanets down to 0.22 $R_\oplus$ in radius. At this stage, the list of metafeatures that survive the cutoff is still highly contaminated by spurious features. While choosing a more strict score cutoff decreases the raw number of these “false positives” (FPs) in the list, a preliminary study (see Appendix A3) suggests that the rate of FPs is roughly uniform for any cutoff. Filtering of FPs through multiple rounds of vetting specific to the study of long-period exoplanets and a thorough statistical treatment based on Fressin et al. (2013) are done in Paper II.

The pipeline itself is best suited for offline analysis of a complete dataset. Due to the computationally intensive coupling of the weighting and scoring, typical runtimes make real-time analysis unfeasible. However, the coupled iteration is parallelized in several places, using Python’s multiprocessing package. Thus, an increase in core count would decrease runtime dramatically. Further optimization of the pipeline itself may certainly be possible; both future and past versions will be archived on the SATCHEL GitHub repository.

The recovery studies for this pipeline, in particular, suggest that future citizen science projects with goals of a statistical nature would benefit from a larger number of simulations. The needs will certainly vary with the project and dataset, but it seems reasonable to assert that some standard minimum number of injections should be constructed for each bin of physical parameter space, and that these bins should be chosen based on the binning most likely to be used for later analysis, led by contemporary understanding of the real distribution of signals across physical parameter space in nature. In general, the complexity of this pipeline has been motivated by the large fraction of users who never interacted with simulations at all (see Figs. 2a and 2c), which could be avoided in future citizen science projects by either requiring users to undergo a short training period before exposure to real data or simply making the first subject displayed in the interface a simulation by default. A bias may be introduced if the user knows to expect a signal to be present in the subject, but including simulations containing no injected signals, “true negatives,” could potentially mitigate such a bias. With more sophisticated weighting scheme may also be possible, in which reliability as a function of, e.g., varying signal depths could be calculated for individual users, rather than the global population (e.g., the Space Warps Analysis Pipeline (SWAP)\textsuperscript{16}, Marshall et al. 2016).

Maximizing user retention must be a high priority for future projects. With the high degree of public attention that the PH project received came the side effect that many users visited the site only very briefly to classify one or two subjects (see Fig. 2c). Studies of similar projects have shown that users who have classified more data generally perform better than brand new users (Eisner et al. 2020a). Projects that are able to retain users for longer periods may have more options for methods of calculating and calibrating user weights (S12, Eisner et al. 2020a; Lintott et al. 2008).

It should be noted: A citizen science project that advertises immediate contribution to real astronomy research should, ideally, be able to utilize participation from all volunteers. This has been one of the cornerstone guiding principles in constructing the SATCHEL pipeline, though, at the inception of PH, it was impossible to foresee some of the challenges that would arise in the effort to remain faithful to the ideal. We present this pipeline as a potential tool for analyzing other past and future PH classification data, or other crowd-sourced time-series data facing similarly complex needs.

\textbf{ACKNOWLEDGEMENTS}

This work has been made possible by the participation of more than 30 000 volunteers in the Planet Hunters project, hosted on the Zooniverse platform. The PH project utilizes data collected by the \textit{Kepler} mission, which is funded by the NASA Science Mission directorate. The authors would like to thank Geoff Clayton and Dan Foreman-Mackey for helpful discussions, as well as the reviewer for providing comments that have greatly improved the clarity and strength of the paper. We would also like to acknowledge Satchel, the red lory, who provided the name of the pipeline and several very loud comments.

\textit{Facilities: Kepler}

\textit{Software:} matplotlib (Hunter 2007), numpy (Van der Walt et al. 2011), pandas (The pandas development team 2020; Wes McKinney 2010), multiprocessing (McKerns et al. 2012)

\textit{Data availability:} The data underlying this article were provided by the Zooniverse with permission. Data will be shared on request to the corresponding author with permission of the Zooniverse. The SATCHEL software is publicly available at https://github.com/ejsafron/PH-pipeline.

\textbf{REFERENCES}

Baron D., 2019, arXiv e-prints, p. arXiv:1904.07248
Berger T. A., Huber D., van Saders J. L., Gaidos E., Taylor J., Kraus A. L., 2020, \textit{AJ}, 159, 280
Bonney R., Cooper C. B., Dickinson J., Kelling S., Phillips T., Rosenberg K. V., Shirk J., 2009, \textit{BioScience}, 59, 977
Borucki W. J., et al., 2010, \textit{Science}, 327, 977
Boyajian T. S., et al., 2016, \textit{MNRAS}, 457, 3988
Bryson S., Coughlin J., Batalha N. M., Berger T., Huber D., Burke C., Dotson J., Mullally S. E., 2020a, \textit{AJ}, 159, 279
Bryson S., Coughlin J. L., Kinnuno M., Mullally S. E., 2020b, \textit{AJ}, 160, 200
Burke C. J., et al., 2015, \textit{ApJ}, 809, 8
Christiansen J., 2021, private communication
Christiansen J. L., et al., 2012, \textit{PASP}, 124, 1279
Christiansen J. L., et al., 2018, \textit{AJ}, 155, 57
Cox J., Oh E., Simmons B., Lintott C., Masters K., Graham G., Greenhill A., Holmes K., 2015, \textit{Computing in Science & Engineering}, 17, 1
Dickinson H., et al., 2018, \textit{Research Notes of the American Astronomical Society}, 2, 120
Dressing C. D., Charbonneau D., 2015, \textit{ApJ}, 807, 45
Eisner N. L., et al., 2020a, \textit{MNRAS}
Eisner N. L., et al., 2020b, \textit{MNRAS}, 494, 750
Feng Q., Jarvis J., VERITAS Collaboration 2017, in 35th International Cosmic Ray Conference (ICRC2017). p. 826 (arXiv:1708.06393)
Fischer D. A., et al., 2012, \textit{MNRAS}, 419, 2900
Foreman-Mackey D., Morton T. D., Hogg D. W., Agol E., Schölkopf B., 2016, \textit{AJ}, 152, 206
Fressin F., et al., 2013, \textit{ApJ}, 766, 81
Gaidos E., Mann A. W., Kraus A. L., Ireland M., 2016, \textit{MNRAS}, 457, 2877
Gordienko Y., 2013, Green Paper on Citizen Science. Societize EU Project Hardegree-Ullman K. K., Cushing M. C., Muirhead P. S., Christiansen J. L., 2019, \textit{AJ}, 158, 75

\textsuperscript{16} On GitHub: https://github.com/zooniverse/swap
The SATCHEL pipeline

APPENDIX A: OPTIMIZATION OF PIPELINE DESIGN AND HYPERPARAMETERS

Construction of the SATCHEL pipeline has involved some decisions that, while seemingly affecting only small details, were non-trivial to make. In an effort to make SATCHEL as useful as possible to other projects, the purpose of this Appendix is to clarify the tests and reasoning that led to those choices with a level of detail that would allow other projects to reproduce them and make the optimal decisions for their own purposes.

Section A1 discusses the optimization of hyperparameters used to construct the metafeature list and match user marks to synthetic signals in the calculation of user weight seeds. Section A2 compares the decay function used in the user weight seed calculation to other possible methods. And finally, Section A3 examines in detail how the post-pipeline choice of score cutoff affects recovery and false positive rates.

Figure A1. Maximum possible signal recovery for KOIs and synthetic exoplanets based on metafeature lists constructed using a range of hyperparameters. For those sets with varying midpoint tolerance (left panel), the duration cutoff was 2.5 days. For sets with varying duration cutoff (right panel), the midpoint tolerance was 0.8 days. Blue points correspond to KOIs, and red points correspond to simulations.

Figure A2. False positive rates for unscored metafeatures in lists constructed using a range of hyperparameters. For those sets with varying midpoint tolerance (left panel), the duration cutoff was 2.5 days. For sets with varying duration cutoff (right panel), the midpoint tolerance was 0.8 days.

A1 Metafeature list hyperparameters

The metafeature list described in Sec. 3.1 is constructed by a process that utilizes two hyperparameters: a “midpoint tolerance” (tolerance), the largest acceptable difference in midpoints between two user marks indicating the same feature in a subject; and a “duration cutoff” (cutoff), the longest acceptable duration of a user mark that might indicate a significant feature. Varying these two parameters does produce slightly different lists, and the metafeatures contained therein may be different in duration and/or midpoint, and more or fewer in number.

While the scoring process provides a useful way to filter out the bulk of metafeatures that likely do not correspond to real signals of interest, it remains true that some signals from both simulations and known KOIs will be lost in the low-scoring regime, simply because many people missed them. In general, the maximum possi-
ble number of known signals recoverable from the PH data — and the maximum number of spurious signals — are obtained when no scoring whatsoever is done and the metafeature list is accepted in its entirety. Since these recovery and “false positive” values depend only on the properties of the metafeatures, we use them to help determine the optimal set of hyperparameters with which to build the metafeature list. That is, we choose the set of parameters that produces the set of metafeatures with the highest maximum possible recovery and the lowest corresponding false positive rate. An equivalent way to describe this methodology is to say that we want the highest possible maximum recall and precision.

Thus far, we have implicitly defined a “positive” as any metafeature that passed a specific post-pipeline score cutoff threshold (recall, in Secs. 4.2 and 4.1, we applied a score cutoff of 0.5). For the purpose of this study, to examine the full metafeature list, we will allow the score floor to be zero, and we will explicitly define a “true positive” as any metafeature that corresponds to a known signal — the transit of either a KOI or a synthetic exoplanet. A “false positive,” then, is a metafeature that does not correspond to a KOI or synthetic exoplanet signal. We limit this discussion to, for KOIs, subjects that correspond to only Kepler targets associated with KOIs, and for synthetic exoplanets, only simulation subjects. That is, we do not use the full set of metafeatures to calculate the false positive rate in either case. We consider this the most correct choice for several reasons:

- No true negatives were included in construction of the simulations; every simulation did contain at least one synthetic transit. To explore the truest possible comparison between the KOI and simulation false positive and recovery rates, we included in the KOI study only subjects displaying data from KOI hosts (though, for some of the longer-period KOIs, this does include subjects that contain no KOI transits).
- To label high-scoring, significant features in targets that are not KOI hosts as false positives, while technically correct as they are not KOI transits, is to imply that those features should be ignored. This practice is incompatible with the spirit of the false positive study.
- KOI hosts (in particular, hosts with systems of multiple KOIs) have been more thoroughly explored for extra long-period planets (Uehara et al. 2016) than targets hosting no known exoplanet candidates, and thus there is a smaller chance that a given false positive feature in a KOI host light curve is actually a signal from a real, yet undiscovered exoplanet.
- Users searching for exoplanet transits in the light curve subjects of KOI hosts are effectively, though unknowingly, specifically searching for KOI transits. The same cannot be said of user classifications of data not corresponding to KOI hosts, therefore the definition of a false positive as “a user mark or set of user marks indicating the transit of a KOI where no such transit truly occurred” is not applicable.

Metafeature lists were constructed and run through the rest of the SATCHEL pipeline using tolerance values from 0.4 to 1.4 days, in increments of 0.2 days (with cutoff = 2.5 days), and cutoff values from 0.15 to 4.0 days (with tolerance = 0.8 days). Figure A1 shows the percentage of signals corresponding to KOIs and synthetic exoplanets with radii > 2 \( R_\oplus \) recoverable in metafeature lists with hyperparameters in these ranges. Figure A2 shows the false positive rates for the same metafeature sets. Both the maximum possible recoveries and the false positive rates, for signals corresponding to objects > 2 \( R_\oplus \) in radius, are roughly the same for KOIs and simulations, separated by only about two per cent.

This is a reassuring result, indicating that users were classifying the simulations the same way, qualitatively, that they were classifying the real data. The significant difference in false positive rate for the whole set of KOIs and simulations may be attributable to the much larger population of KOIs with radii < 2 \( R_\oplus \), and even occasionally < 1 \( R_\oplus \), which was the cutoff for radii of the synthetic exoplanets.

The simulations and real data also display the same trends across varying hyperparameters: A slight decrease in possible recovery and decrease in false positive rate with increasing midpoint tolerance, and stable recovery with increasing false positive rate for increasing duration cutoff. A trend of higher possible recovery with increasing duration cutoff does appear for signals corresponding to objects with radii < 2 \( R_\oplus \), but, as we do not expect the pipeline to be even approximately complete in this regime, these objects are not the focus of our study and we instead prioritize a lower false positive rate in our choice of cutoff.

Figures A3 and A4 show the precision as a function of recall for varying midpoint tolerance and duration cutoff values, respect-
For varying midpoint tolerance, maximum recall and average precision both increase with midpoint tolerance up to about 1.0 day, after which average precision continues to rise but maximum recall falls. Beyond a midpoint tolerance of 1.4 days, average precision also falls. The maximum AUC corresponds to a midpoint tolerance of 1.2 days. For varying duration cutoff, we find the highest maximum recall for the largest, most lenient values. As the duration cutoff decreases, the maximum recall decreases and average precision increases. Average precision rises steeply with duration cutoffs less than 0.5 days, but maximum recall decreases sharply as well. There is a significant change in the behavior of the PR curve between duration cutoffs of 0.45 and 0.5 days. While we are uncertain what the reason for this is, we offer that it may be due to a drastic change in behavior of PH volunteers at marking widths of 0.5 days; for instance, the default marking width in the PH interface may have been 0.5 days. The maximum AUC corresponds to a duration cutoff value of 1.0 days.

Thus, we choose a midpoint tolerance of 1.2 days and a duration cutoff of 1.0 day.

### A2 User weight decay function

The need for a decay function in calculation of the user weight seeds was not immediately apparent in the initial design of the PH pipeline. No decay function was implemented in previous PH studies, including S12, nor is any decay function used in the single-calculation user weighting process of the PH TESS pipeline.

In the case of the Kepler PH data, the decay function was conceptually motivated by examination of the distribution of user weight seeds. Prior to inclusion of the decay function, the weights of some users were inflated to many times the mean weight, as shown by the red distribution in Figure A6. With a mean weight of 1.0 after normalization, the highest weight was over 50 units in magnitude, and belonged to a user who had classified only three subjects, all of which were simulations and one of which contained seven shallow transits of a 1.5 $R_E$ synthetic exoplanet. While, individually, these seven transits would all be equally difficult to find, the strictly periodic placement of them in a light curve subject decreases that difficulty after the first two or three signals found. That is, after finding two transits, a user knows exactly where to look to find the next. Thus, it seemed unfair to award the full raw upweight for each of a large number of signals identified in a single subject.

Rather, it was determined that a more reasonable approach would be to award full or very high upweights only to the few most accurately identified synthetic signals in a subject, with any others identified being given decreased amounts of upweight, until eventually the user is being awarded very little or no upweight for further signals identified. A function consistent with this behavior was constructed, and a few variants devised. Particularly, we tested a function with an inverse exponential tail as given in Equation 3, which awards a low but non-zero fraction of the raw upweight for any synthetic signal identified past the fourth, and a similar function with a strict cutoff,

$$f_{\text{decay}}(k) = \begin{cases} g(k) & \text{if } k \leq 3, \\ 0 & \text{if } k > 3, \end{cases} \quad (A1)$$

(where $g(k)$ is again identical to Eq. 4), which awards no upweight at all past the fourth synthetic signal identified. These two functions are shown in blue and green in Figure A5, and their effect on the user weight seed distribution is shown in the same respective colors in Fig. A6. Both functions significantly narrow the spread of the distribution toward large user weights, and both completely remedy the issue of the inflated outliers. We chose to implement Eq. 3, with the inverse exponential tail, because we deemed it more correct in principle to reward users for marking all significant features in a subject, even if that reward must be small.

The inner function $g(k)$, of the general form $a + b e^{-c k}$, contains three coefficients $a$, $b$, and $c$, which can be adjusted or optimized. For the PH data, the user weight distributions resulting from implementation of both Eqs. 3 and A1 were robust under slight variations of all coefficients, and, further, the final score distribution was robust under slight changes in the user weight seed distribution, so we did not feel that optimizing the coefficients was a high priority. It is possible that data from other projects could be more sensitive to the choice of coefficients, in which case optimization may be necessary. On the other hand, however, for projects in which the signals of interest are not periodic or patterned in nature, the decay function may be done away with entirely, or set simply to unity.
A3 Score cutoff

The purpose of the score cutoff, in practice, is to limit any further vetting of metafeatures to only the most promising subset. Much like the hyperparameters discussed in Section A1, choosing the optimal score cutoff with which to filter results from the pipeline is a balancing act that may vary from project to project. Generally, we seek to maximize the recovery of truly interesting signals while minimizing contamination by spurious features.

For this study, our definitions of true and false positives will be fundamentally identical to those of Appendix A1, in that a “positive” shall be any metafeature that passes a specific score cutoff threshold. We investigated the effect of applying different score thresholds on the sample, by recalculating recovery and false positive rates with score floors varying from 0.0 (no floor) to 0.9 (the most restrictive), the results of which are summarized in Figure A7. The bottom panel, displaying recovery fraction as a function of score cutoff for both KOIs and simulations (and subsets of both with radii > 2 \( R_{\oplus} \)) shows a steep rise as the score cutoff is made less strict, dropping from values of 0.9 to 0.8 and 0.7. At the point where all metafeatures with scores > 0.6 are included, over 65% of signals produced by KOIs larger than 2 \( R_{\oplus} \) are recovered and over 80% of signals produced by the same size subset of synthetic exoplanets. Dropping the score cutoff to 0.5 increases signal recovery for these objects to over 70% for KOIs, and over 85% for simulations. Below about 0.5, however, the increase in recovery rate is slower, leveling out significantly toward the lowest score values. The outlier case is that of the full set of KOIs. The presence there of very small exoplanets, much smaller than the smallest of the synthetic exoplanets, keeps the signal recovery rate extremely low even at the most lenient of score cutoffs. While recovery does increase as the score cutoff decreases, the increase is almost perfectly linear, unlike that of the other subsets examined.

Of course, as the score cutoff decreases, the false positive rate also increases. As shown in the top panel of Fig. A7, around the same score cutoff value of 0.5, the ratio of false positives to true positives begins to rise at a higher rate, particularly for the full sets of KOIs and simulations. This is interesting, in itself — it may indicated that, in the absence of obvious signals, users are more likely to mark ambiguous features. Or, conversely, that in the presence of obvious signals, users may ignore ambiguous features, potentially missing some transit signals that are real, but much shallower. For the subsets of KOIs and synthetic exoplanets larger than 2 \( R_{\oplus} \), the false positive to true positive ratio does increase more steeply below a score cutoff of 0.5, but the difference is more subtle.

The combination of the upper and lower panels of Fig. A7 suggests that a score cutoff near the middle is most practical. In the more linear low-scoring regime of the bottom panel, where the signal recovery fraction is increasing shallowly but steadily, the false positive to true positive ratios are also increasing in a fairly linear fashion. The proportionality between these increases begins to deviate in a significant way at a score cutoff of 0.5. Thus, we choose 0.5 as the working score cutoff for our analyses in both Section 4 and Paper II.

A last oddity to note is that the false positive to true positive ratios for simulations appear to be twice as high as their KOI counterparts, in Fig. A7. As far as the authors are aware, there was no difference in user methodology when classifying simulations or KOIs, and no significant qualitative differences in either the transit signals or the background data. The source of this offset may be worth exploring in the future.