An Update to the EVEREST K2 Pipeline: Short Cadence, Saturated Stars, and Kepler-like Photometry Down to $K_p = 15$

Rodrigo Luger$^{1,2}$, Ethan Kruse$^1$, Daniel Foreman-Mackey$^3$, Eric Agol$^{1,2,4}$, and Nicholas Saunders$^1$

1 Department of Astronomy, University of Washington, Seattle, WA, USA; rodluger@uw.edu
2 Virtual Planetary Laboratory, University of Washington, Seattle, WA, USA
3 Center for Computational Astrophysics, Flatiron Institute, New York, NY, USA

Received 2017 January 20; revised 2018 July 1; accepted 2018 July 5; published 2018 August 17

Abstract

We present an update to the EVEREST K2 pipeline that addresses various limitations in the previous version and improves the photometric precision of the light curves. We develop a fast regularization scheme for pixel-level decorrelation (PLD) and adapt the algorithm to include the PLD vectors of other stars to enhance the predictive power of the model and minimize overfitting, particularly for faint stars. We also modify PLD to work for saturated stars and improve its performance on variable stars, although some high-frequency variables may still suffer from overfitting. On average, EVEREST 2.0 light curves have 10–20% higher photometric precision than those in version 1, yielding the highest-precision light curves at all $K_p$ magnitudes of any publicly available K2 catalog. For most K2 campaigns, we recover the original Kepler precision to at least $K_p = 14$, and to at least $K_p = 15$ for campaigns 1, 5, 6, and 13. We also define most short-cadence targets observed by K2, obtaining even higher photometric precision for these stars. Like all aggressive, flexible models, EVEREST is prone to overfitting, and may cause a decrease in transit depths by $\sim$10%; we urge users to mask signals of interest using our open-source software, which we show removes this bias. Light curves for campaigns 0–8 and 10–13 are available online in the EVEREST catalog, which will be updated with future campaigns. EVEREST 2.0 is open source and is coded in a framework that can be adapted to other photometric surveys, including Kepler and the upcoming TESS mission.

Key words: catalogs – planets and satellites: detection – techniques: photometric

1. Introduction

The failure of the second of four reaction wheels in 2013 brought the original Kepler mission to a premature conclusion, as the spacecraft could no longer achieve the fine pointing precision required for the groundbreaking transiting exoplanet and stellar variability science that the first four years of the mission allowed. Since 2014, the spacecraft has been operating in a new mode, known as K2, using periodic thruster firings to mitigate drift induced by solar radiation pressure (Howell et al. 2014). Despite these measures, raw K2 photometry is significantly poorer than that of the original Kepler mission. In order to enable continuing precision science with K2, numerous pipelines have been developed (e.g., Vanderburg & Johnson 2014; Armstrong et al. 2015; Crossfield et al. 2015; Foreman-Mackey et al. 2015; Huang et al. 2015; Lund et al. 2015; Aigrain et al. 2016; Libralato et al. 2016a, 2016b), many producing light curves with precision approaching that of Kepler for bright stars.

In Luger et al. (2016) (henceforth Paper I), we developed the first version of our K2 pipeline, EVEREST (EPIC Variability Extraction and Removal for Exoplanet Science Targets). We employed a variant of pixel-level decorrelation (PLD) based on the method of Deming et al. (2015), a data-driven approach that uses a star’s own pixel-level light curve to remove instrumental effects. We showed that EVEREST recovered the original Kepler precision for stars brighter than Kepler-band magnitude $K_p \approx 13$, yielding higher average precision than any publicly available K2 catalog for unsaturated stars.

In this paper, we present an update to our pipeline, which we refer to as EVEREST 2.0. By combining PLD with spacecraft motion information obtained from nearby stars, this update improves the precision of K2 light curves at all magnitudes relative to version 1.0 and addresses certain issues with overfitting. EVEREST 2.0 also reliably detrends saturated stars and stars observed in short-cadence mode, obtaining comparable or even higher detrending power for these targets.

The paper is organized as follows: in Section 2, we derive the mathematical framework of the EVEREST 2.0 model. In Section 3, we describe the implementation of our pipeline in detail. We present our results in Section 4 and additional remarks in Section 5. In Section 6, we discuss how to use the EVEREST catalog and code, and we summarize the work in Section 7.

2. The PLD Model

Here, we describe the mathematical formulation of the EVEREST PLD model. In PLD, products of the fractional fluxes in each pixel of the target aperture are used as regressors in a linear model:

$$m = \sum_i a_i \frac{p_i}{\sum_n p_n} + \sum_i \sum_j b_{ij} \frac{p_ip_j}{\left(\sum_n p_n\right)^2} + \sum_i \sum_j \sum_k c_{ijk} \frac{p_ip_jp_k}{\left(\sum_n p_n\right)^3}. \quad (1)$$

$^2$ Guggenheim Fellow.
In the expression above, \( m \) denotes the component of the target’s lightcurve that is due to instrumental noise and \( p_i \) denotes the flux in the \( i \)th pixel; both are vector quantities defined at an array of times \( t \). Each term corresponds to a different PLD order (first, second, or third) resulting from a Taylor expansion of the instrumental signal. The \( a_i, b_{ij}, \) and \( c_{ijk} \) are the linear weights of the model, which we seek to obtain below. For a detailed discussion of the theory behind PLD, see Deming et al. (2015) and Paper I. Below, we simply discuss its mathematical implementation.

### 2.1. Regularized Regression (rPLD)

Given a timeseries \( y \) with \( N_{\text{dat}} \) data points, we wish to find the linear combination of \( N_{\text{reg}} \) regressors that best fits the instrumental component of \( y \). Expressed in vector form, our linear model is thus

\[
m = X \cdot w,
\]

where \( X \) is the \((N_{\text{dat}} \times N_{\text{reg}})\) design matrix constructed from the set of regressors (the fractional pixel fluxes in Equation (1)) and \( w \) is the \((N_{\text{reg}} \times 1)\) vector of weights (the set \( \{a_i, b_{ij}, c_{ijk}\} \)). If \( w \) is known, the detrended light curve is simply

\[
y' = y - m.
\]

In Paper I, we obtained \( w \) by maximizing the likelihood function

\[
\log \mathcal{L}_0 = -\frac{1}{2} (y - X \cdot w)^\top K^{-1} (y - X \cdot w) - \frac{1}{2} \log |K| - \frac{N_{\text{dat}}}{2} \log 2\pi,
\]

where \( K \) is the \((N_{\text{dat}} \times N_{\text{dat}})\) covariance matrix of the data and \( y \) is the \((N_{\text{dat}} \times 1)\) simple aperture photometry (SAP) flux. Because the number of third order PLD regressors can be quite large (on the order of several thousand for a typical star, which is larger than the number of data points), the problem is ill-posed, meaning that a unique solution does not exist and maximizing \( \log \mathcal{L}_0 \) is likely to lead to overfitting. We thus constructed \( X \) from the (smaller) set of \( N_{\text{pc}} \) principal components of the PLD regressors. We chose \( N_{\text{pc}} \) by performing cross-validation, which aims to maximize the predictive power of the model while minimizing overfitting.

However, while principal component analysis (PCA) yields a set of components that captures the most variance among the PLD vectors, there is no guarantee that the principal components are the ideal regressors in the PLD problem. Dimensionality reduction techniques such as PCA inevitably lead to information loss, so it is worthwhile to consider alternative regression methods to fully exploit the potential of PLD.

A common regression method for ill-posed problems is regularization, in which a prior is imposed on the values of the weights \( w \). Because overfitting occurs when \( w \) becomes very large, regularization recasts the problem by adding a penalty term to the likelihood that increases with increasing \( |w| \). While many forms of regularization exist, we focus on L2 regularization because it has an analytic solution. Recently, Wang et al. (2016) used L2 regularization in a “causal pixel model” to detrend light curves from the original Kepler mission. L2 regularization is mathematically equivalent to placing a Gaussian prior on each of the weights \( w \), such that the posterior likelihood function becomes

\[
\log \mathcal{L} = \log \mathcal{L}_0 - \frac{1}{2} w^\top \cdot A^{-1} \cdot w - \frac{1}{2} \log |A|,
\]

where \( A \) is the \((N_{\text{reg}} \times N_{\text{reg}})\) regularization matrix, which we choose to be diagonal for both simplicity and computational efficiency:

\[
A_{mn} = \lambda_m^2 \delta_{mn}.
\]

Each element \( \lambda_m^2 \) in \( A \) is the variance of the zero-mean Gaussian prior on the weight of the corresponding column of the design matrix, \( X_{m,n} \). In practice, we find that, if we choose the \( \lambda \) correctly, this model has a higher predictive power than the PCA model adopted in Paper I.

Given this formulation, our task is to find the weights \( \hat{w} \) that maximize the posterior probability \( \mathcal{L} \). Differentiating Equation (5) with respect to \( w \), we get

\[
d \log \mathcal{L} = X^\top \cdot K^{-1} \cdot y - (X^\top \cdot K^{-1} \cdot X + A^{-1}) \cdot w.
\]

By setting this expression equal to zero, we obtain the maximum a posteriori prediction for the weights:

\[
\hat{w} = (X^\top \cdot K^{-1} \cdot X + A^{-1})^{-1} \cdot X^\top \cdot K^{-1} \cdot y,
\]

with corresponding model

\[
m = X \cdot (X^\top \cdot K^{-1} \cdot X + A^{-1})^{-1} \cdot X^\top \cdot K^{-1} \cdot y.
\]

In what follows, we refer to the implementation of PLD with regularized regression as rPLD.

### 2.2. Cross-validation

Similarly to Paper I, we solve for \( A \) by cross-validation. For each value of \( A \), the model is trained on one part of the light curve (the training set) and used to detrend the other part of the light curve (the validation set); see Section 3.7 for details. The value of \( A \) that results in the minimum scatter in the validation set is then chosen for the final detrending step.

In principle, each of the \( \lambda \) in \( A \) could take on a different value, but solving for each one requires minimizing an \( N_{\text{reg}} \)-dimensional function and is not computationally tractable. Instead, we simplify the problem by requiring that all regressors of the same order have the same regularization parameter \( \lambda \). Provided we write the third order design matrix in the form

\[
X = (X_1 \ X_2 \ X_3),
\]

where \( X_n \) is the matrix of \( n \)-th order regressors, we may express the regularization matrix as

\[
A = \begin{pmatrix} A_1 & A_2 \\ A_2 & A_3 \end{pmatrix}
\]
where $A_n = \lambda_n^2 I$ is the $n$th-order regularization matrix and $\lambda_n^2$ is the variance of the prior on the $n$th-order regressors.

A typical K2 star with 30 aperture pixels has $N_{\text{reg}} \sim 5000$ regressors and $N_{\text{dat}} \sim 500$ data points in each cross-validation light curve segment (see Section 3.7). Thus, evaluating the matrix inverse in Equation (9) is computationally expensive and becomes prohibitive during cross-validation, as this must be done for every set of $\lambda_n$’s. Fortunately, we can reduce the number of calculations with some linear algebra. First, we apply the Woodbury matrix identity (e.g., Golub & Van Loan 1996) to Equation (9), obtaining

$$m = X \cdot A \cdot X^T \cdot (X \cdot A \cdot X^T + K)^{-1} \cdot y. \quad (12)$$

Next, we note that

$$X \cdot A \cdot X^T = (X_1 \; X_2 \; X_3) \begin{pmatrix} A_1 & A_2 & A_3 \end{pmatrix} \begin{pmatrix} X_1^T \\ X_2^T \\ X_3^T \end{pmatrix}$$

$$= \lambda_1^2 X_1 \cdot X_1^T + \lambda_2^2 X_2 \cdot X_2^T + \lambda_3^2 X_3 \cdot X_3^T$$

$$= \sum_n \lambda_n^2 X_n^2,$$

where we have defined

$$X_n^2 \equiv X_n \cdot X_n^T. \quad (13)$$

We may thus rewrite our maximum a posteriori model as

$$m = \sum_n \lambda_n^2 X_n^2 \cdot \left( \sum_n \lambda_n^2 X_n^2 + K \right)^{-1} \cdot y. \quad (15)$$

The matrix that we must invert in Equation (15) has dimensions $(N_{\text{dat}} \times N_{\text{dat}})$, while that in Equation (9) is $(N_{\text{reg}} \times N_{\text{reg}})$. Given that $N_{\text{reg}} \sim 10N_{\text{dat}}$, casting the model in this form can greatly speed up the computation. In practice, we precompute the three matrices $X_n^2$ at the beginning of the cross-validation step, so the only time-consuming operation in Equation (15) is the inversion.

2.3. Neighboring Stars (nPLD)

One of the downsides of PLD is that the regressors used in the linear model tend to be noisy. This is particularly a problem for faint targets, whose PLD vectors are dominated by photon noise. Their light is also distributed over fewer pixels, resulting in a smaller set of vectors on which to regress. The effect of this is evident in Figure 10 of Paper I, which shows how EVEREST 2.0 light curves for faint (Kp $\geq 15$) stars are significantly noisier than those of the original Kepler mission. This decrease in detrending power at the faint end affects most other K2 pipelines as well, because those usually regress on information derived (either directly or indirectly) from the motion of the stellar image across the detector, which is similarly noisy.

While the spacecraft motion (the dominant source of instrumental noise in K2) is imprinted at relatively low signal-to-noise ratio (S/N) on the light curves of any one star, the collective light curves of all the stars on the detector encode this information at very high S/N. Therefore, a straightforward way to improve the performance of PLD for faint targets is to include this information in the design matrix. To this end, in EVEREST 2.0, we incorporate the PLD vectors of a set of other targets located on the same CCD module as the target of interest when performing the regression. We dub this method nPLD (for “neighboring PLD”) and discuss its implementation in Section 3.4. Our third-order design matrix (Equation (10)) is now

$$X = \begin{pmatrix} X_1 & X_1' & X_2 & X_2' & X_3 & X_3' \end{pmatrix}, \quad (16)$$

where $X_n'$ is the design matrix constructed from the $n$th-order PLD vectors of all the neighboring targets. For computational speed, we still solve for a single prior amplitude $\lambda_n$ for each PLD order, but in principle one could assign different priors to the neighboring vectors. We discuss the implementation of nPLD in Section 3.

3. Implementation

3.1. Light Curves

As in Paper I, we downloaded all stars in the K2 EPIC catalog with long- and/or short-cadence target pixel files (TPFs) and adopted aperture #15 from the K2SFF catalog (Vanderburg 2014; Vanderburg & Johnson 2014). We masked all cadences with QUALITY flags 1–9, 11–14, and 16–17, though we still compute the model prediction on them. For campaigns 0–2, we remove the background signal as described in Paper I; for more recent campaigns, the background is removed by the Kepler team.

Next, we perform iterative sigma clipping to identify and mask outliers at 5$\sigma$. During each iteration, we compute the linear (unregularized) PLD model and smooth it with a Savitsky–Golay filter (Savitzky & Golay 1964), then identify outliers based on a median absolute deviation (MAD) cut. We implement this outlier-clipping step at the beginning of each cross-validation step (Section 3.7), each time computing the model with a higher (regularized) PLD order, to progressively refine the outlier mask.

3.2. GP Optimization

In order to compute the covariance matrix $K$ for each target, we use a GP, as we did in Paper I. GP optimization can be costly, especially when performing model selection over a range of possible kernels and optimizing many hyperparameters simultaneously. For this reason, in Paper I we cut corners and performed kernel selection based on fts to the autocorrelation function of the light curve, which we also used to fix the timescale and/or period of those kernel(s). We then ran a nonlinear minimizer to optimize the overall amplitude of the GP. In practice, this worked reasonably well, but often failed for light curves dominated by high-frequency stellar variability. After much experimentation, we decided to forego the kernel selection step in favor of using a single carefully optimized Matérn-3/2 kernel with an added white noise term:

$$K_{ij} = \alpha \left(1 + \frac{\sqrt{3}(t_i - t_j)^2}{\tau} \right) e^{-\frac{(t_i - t_j)^2}{\sigma^2}} + \sigma^2 \delta_{i,j}, \quad (17)$$

where the hyperparameters $\sigma$, $\alpha$, and $\tau$ are the white noise amplitude, red noise amplitude, and red noise timescale, respectively, and $t_i$ and $t_j$ correspond to the timestamps of cadences $i$ and $j$. We initialize the hyperparameters at random values and run a nonlinear optimizer to solve for the maximum likelihood (Equation (5)), keeping the PLD model parameters fixed; we repeat this process several times and retain the
highest likelihood solution. As with outlier clipping, we progressively optimize the GP at each of the three cross-validation steps, so that each time we train the GP on a light curve that is increasingly dominated by stellar variability (as opposed to instrumental systematics).

In principle, the quasi-periodic kernels used in Paper I should be better suited to handling variable stars, but in practice, we find that a properly optimized Matérn-3/2 kernel is flexible enough to fully capture the variability and prevent PLD overfitting. We discuss this further in Section 5.

3.3. Breakpoints

Because the instrumental noise properties are quite variable over the course of K2 campaigns, we find a significant improvement in the detrending power of our regularized regression model when we subdivide light curves into two or three segments. This is in contrast to the PCA approach in Paper I, where we did not find it necessary to split the timeseries. For all campaigns except 4 and 7, we add a single breakpoint in the light curve near the mid-campaign point, where the spacecraft roll is at a minimum. For campaigns 4 and 7, we find it necessary to insert two breakpoints. We cross-validate and detrend each light curve segment separately and mend them at the end. In order to mitigate flux discontinuities at the breakpoints, we train the model in each segment on an additional 100 cadences past the breakpoint to remove potential edge effects and offset the models in each segment so that they align at the breakpoint. While this method removes flux discontinuities, it can introduce discontinuities in the derivative of the flux, showing up as spurious “kinks” in the light curve. We remove these in a post-processing step (Section 4.7).

3.4. Neighboring Stars

In principle, the larger the number of neighboring PLD vectors we include in the nPLD design matrix, the higher the detrending power of our model. However, adding regressors significantly increases computing time, so we would like to instead select a small set of high S/N regressors that capture most of the spacecraft motion information. Moreover, because we employ a single prior for all nth-order regressors, adding many foreign PLD vectors effectively dilutes the contribution of the target’s own PLD vectors, which are the only ones that can correct instrumental signals arising from local pixel sensitivity variations; in practice, this results in poorer-quality light curves. After much experimenting, we obtain the highest average detrending power when the number of neighboring stars is \( \sim 10 \). We therefore detrend each K2 target with the aid of the PLD vectors of 10 randomly selected bright (11 \( \leq Kp \leq 13 \)) stars on the same detector module as the target. To minimize contamination of the target by outliers in its neighbors’ fluxes, we linearly interpolate over all neighbor data with flagged QUALITY bits. Finally, for computational reasons, we neglect all cross terms of the form \( \prod_{i \neq j} p_i p_j \), where \( p_i \) is the flux in the \( i \)th pixel, when computing the neighbors’ PLD vectors. Cross terms typically encode information specific to the sets of pixels from which they are computed and aid in correcting features such as thruster firing discontinuities (Luger et al. 2016). Therefore, cross terms from stars other than the target in question are of little help in the detrending and can be safely neglected.

One potential pitfall of nPLD is that, if the PLD assumptions break down for any of the neighboring targets, the PLD regressors may become contaminated with astrophysical information from that neighbor. This is not an issue in general, because overfitting would only occur if an astrophysical signal in the target star and in its neighbor had the same period and the same phase. However, in the (unlikely) case that PLD fails for the neighboring star and this star happens to be an eclipsing binary or a transiting exoplanet host, it is possible that its transit signals could get imprinted onto the target star’s detrended light curve, resulting in potential false positive planet detections down the line. The two cases relevant to K2 in which PLD could fail in such a way are for saturated stars and stars with bright contaminant sources in their apertures (Luger et al. 2016). As we show in Section 3.5, it is straightforward to adapt PLD to work reliably for saturated stars, thereby circumventing this issue. However, while EVEREST 2.0 is more robust against overfitting of crowded stars (Section 5), highly crowded apertures remain an issue for PLD. When detrending with nPLD, we therefore select neighboring stars with no other known sources in their apertures that are bright enough (\( \Delta Kp < 5 \)) to contaminate the PLD vectors.

3.5. Saturated Stars

As discussed in Paper I, PLD typically fails for stars with saturated pixels, resulting in overfitted light curves with artificially low scatter and suppressed astrophysical information (such as transits with significantly shallower depths). This happens because saturated pixels contain nearly no astrophysical information, as the signal overflows into adjacent pixels in the same column and is ultimately dumped into the pixels at the top and bottom of the bleed trails; these “tail” pixels ultimately contain more astrophysical information than the other pixels in the aperture. Because PLD implicitly assumes that astrophysical information is constant across the aperture, the method breaks down for these stars, and PLD vectors from pixels in the saturated columns become capable of fitting out the astrophysical information in the rest of the aperture.

An obvious workaround is to simply discard pixels in saturated columns from the set of PLD regressors. However, this does not work well in practice, because the remaining regressors often have much lower S/N than the SAP flux and thus have low detrending power. We instead suggested in Paper I that collapsing saturated columns into single pixels—by co-adding the fluxes in each of the pixels and treating the resulting timeseries as a single PLD pixel—could reduce the effect of saturation, because charge is conserved along the bleed trail. While this ensures PLD does not overfit, it leads to the loss of some of the information about the vertical motion of the stellar point-spread function (PSF) across the detector. This leads to significantly poorer detrending, so we did not employ this method in the first version of the pipeline. However, we find that including the PLD vectors of neighboring stars in the design matrix (i.e., nPLD) effectively restores the information lost when saturated columns are collapsed, leading to high-quality detrended light curves of saturated stars.

In practice, we collapse all columns containing one or more pixels whose flux comes within 10% of (or exceeds) the pixel well depth for the corresponding detector channel during more than 2.5% of the timeseries. We obtained the well depths from
Table 13 of the Kepler Instrument Handbook. To avoid aperture losses, we found it necessary to use aperture K2SFF #19 (the largest of the PSF-based apertures) for these stars. We further padded these apertures by two pixels at the top and bottom of saturated columns to ensure that none of the target flux bled out of the aperture; see Section 5.3.

As an example, in Figure 1 we plot the light curve of EPIC 202063160, a saturated campaign 0 eclipsing binary. The raw light curve is shown at the top, and the light curve detrended with EVEREST 1.0 is shown at the center. Because three of the columns in the aperture contain saturated pixels, EVEREST 1.0 almost completely fits out the eclipses. With column-collapsed nPLD (bottom), the eclipse is preserved and the instrumental signal is effectively removed.

3.6. Short Cadence

We treat short-cadence targets in much the same way as long-cadence targets, with the exception that we find it necessary to introduce more breakpoints in the light curves. This is due primarily to computational reasons (short-cadence K2 light curves are over 10^6 cadences in length; computing Equation (15) for the entire light curve is not feasible). Moreover, we find that noise on timescales shorter than 30 minutes is only properly removed when the size of the light curve segments is kept small. In practice, we find that a number of breakpoints on the order of 30 results in the best detrending. This might raise concerns of overfitting, but given that short-cadence light curves contain 30 times more data than long-cadence light curves, and we split the latter into two segments, each of the short-cadence segments has about twice as many cadences as the long-cadence ones do.

The major downside of such a large number of segments is that discontinuities could be introduced at each breakpoint. As before, we overcompute the model into adjacent segments and match the models at the breakpoints, but some detrended light curves display occasional jumps in either the flux or its derivative.

Another issue with short-cadence light curves concerns deep transits and eclipses. As we discussed in Paper I, PLD may attempt to fit out these features if they are not properly masked, as doing so can result in a very large (but spurious) improvement in the photometric precision. With long-cadence light curves, transit masking can be done by the user by simply recomputing the model with the appropriate cadences masked, because the transits are sparse and their presence does not significantly affect the cross-validation step. Moreover, outlier clipping usually masks most deep transits anyways, so this is hardly ever a problem. However, that is not the case with short-cadence light curves, where transits and eclipses span upward of 50 contiguous cadences. Because these features are so smooth, they are not flagged as outliers, and given that the transit signal is no longer sparse—as it makes up a substantial fraction of the light curve segment—it is far more likely to bias the cross-validation step. In practice, we find that this leads to substantial underfitting of short-cadence light curves with deep transits. As \( \lambda_n \) increases, PLD begins to fit out the transit and the scatter in the validation set grows, forcing the algorithm to select very low values of \( \lambda_n \) and resulting in detrended light curves that still contain significant instrumental signals. Therefore, we explicitly mask all deep transits and eclipses in

---

6 archive.stsci.edu/kepler/manuals/KSCI-19033-001.pdf

---

Figure 1. EPIC 202063160, a saturated Kp = 9.2 campaign 0 eclipsing binary. Shown is a portion of the raw light curve (top), the light curve “detrended” with EVEREST 1.0 (center), and the light curve detrended with EVEREST 2.0 (bottom); the y-axis in each of these plots is the normalized flux. The pixel image is shown at the right on a linear scale, with the adopted aperture contour indicated in red. The three columns highlighted in red contain saturated pixels. Despite a great improvement in the precision, EVEREST 1.0 leads to severe overfitting, causing the eclipses to all but disappear. By collapsing saturated columns, EVEREST 2.0 correctly detrends saturated stars without overfitting.
the short-cadence light curves before the cross-validation step. Because only deep transits are likely to bias the cross-validation, and given that the number of short-cadence light curves in each campaign is relatively small, these can easily be identified by inspection.

3.7. Cross-validation

The principal step in the detrending process is determining the prior amplitudes \( \lambda_n \) in Equation (15), which we do by cross-validation. Our method is analogous to that of Paper I, where we performed cross-validation to obtain the optimal number of principal components to regress on. However, here we seek to optimize a three-dimensional function \( \sigma_v(\lambda_1, \lambda_2, \lambda_3) \), where \( \sigma_v \) is the scatter in the validation set; this is a far more expensive calculation to do. While we could employ a nonlinear optimization algorithm (see below), in the interest of computational speed, we perform a simplification. Because we expect the first-order PLD regressors to contain most of the detrending information, with each successive PLD order providing a small correction term to the fit, we break down the minimization problem into three separate one-dimensional problems. First, we perform cross-validation on the first-order PLD model by setting \( \lambda_1 = \lambda_2 = 0 \) to obtain the value of \( \lambda_3 \) that minimizes the validation scatter, \( \hat{\lambda}_3 \). We do this by computing the model for each value of \( \lambda_3 \) in a logarithmically spaced grid with 36 points in the range \([10^{0.1}, 10^{0.9}]\), plus \( \lambda_3 = 0 \), and select the minimum (details below). We then repeat this process on the second-order model by fixing \( \hat{\lambda}_3 \) at this estimate and keeping \( \lambda_3 = 0 \). Finally, we solve for \( \hat{\lambda}_1 \) by fixing the first- and second-order parameters at their optimum values:

\[
\hat{\lambda}_1 = \arg \min \sigma_v(\lambda_1)|_{\lambda_2 = 0, \lambda_3 = 0} \\
\hat{\lambda}_2 = \arg \min \sigma_v(\lambda_2)|_{\lambda_1 = \hat{\lambda}_1, \lambda_3 = 0} \\
\hat{\lambda}_3 = \arg \min \sigma_v(\lambda_3)|_{\lambda_1 = \hat{\lambda}_1, \lambda_2 = \hat{\lambda}_2}.
\]  

(18)

It is important to note that there is no a priori reason that this method should yield the global minimum of \( \sigma_v \); in fact, it very likely does not. However, we explicitly allow for \( \lambda_3 = 0 \) in our grid search, so this approximation cannot lead to overfitting, as it will always prefer a lower-order PLD model to one with higher scatter in the validation set.

As a proof of concept, we detrended a sample of 2700 randomly selected campaign 6 targets with \( nPLD \), using this approximation in the cross-validation step. We then repeated the detrending by solving for the \( \hat{\lambda}_n \) using Powell’s method, initializing the solver at different points in the vicinity of the \( nPLD \) solution and keeping the solution with the lowest average 6 hr CDPP (combined differential photometric precision) (Christiansen et al. 2012) for each target; we dub this method \( pPLD \). In Figure 2, we plot the star-by-star CDPP difference between the two models, \( (CDPP_{pPLD} - CDPP_{nPLD})/CDPP_{nPLD} \). For some stars, the CDPP improves substantially with \( pPLD \), but cross-validating with Powell’s method leads to an improvement of less than one percent in the CDPP on average. Given that this method is more computationally expensive, we adopt the grid search method outlined above when producing the EVEREST 2.0 catalog.

In addition to it being more computationally tractable, there are two major benefits to minimizing \( \sigma_v \) in this fashion. First, because we perform cross-validation three times (once for each PLD order), we are able to progressively refine the outlier masks (Section 3.1) and the GP hyperparameters (Section 3.2) for each target in between cross-validation steps. Second, it allows for some leeway in how we determine the minimum validation scatter. In Paper I, we sought to minimize the median scatter in groups of random 13 cadence segments of the light curve (the validation set). A potential issue with this method is that the noise properties of \( K2 \) light curves are far from constant over the course of an observing campaign; optimizing the regression based on the median (or mean) validation scatter can still, in principle, lead to overfitting in some segments. While splitting the light curves into segments with similar noise properties (Section 3.3) ameliorates this, we also modify the cross-validation process to prevent localized overfitting. For each PLD order \( n \) and for each value of \( \lambda_n \), we split each light curve segment into three roughly equal sections. For each pair of sections, we train the model on them and compute the model prediction in the third section (the validation set). We then compute the scatter \( \sigma_v \) as the MAD of the detrended validation set after removing the GP prediction.

We now have three \( \sigma_v(\lambda_n) \) curves, one for each section. In general, the minima of these curves will occur at different values of \( \lambda_n \), so determining the optimum value \( \hat{\lambda}_n \) requires a compromise between overfitting and underfitting in the different segments. For each segment, we compute the minimum scatter, find the set of all \( \lambda_n \) for which \( \sigma_v(\lambda_n) \) is
within 5% of the minimum, and keep the largest \( \lambda_n \). We then pick \( \hat{\lambda}_n \) to be the smallest of these values, provided it is smaller than the largest value of \( \lambda_n \) at the minima of the three \( \sigma_i \) curves. This process ensures that \( \hat{\lambda}_n \) falls between the minima of the \( \sigma_i \) curves with the smallest and largest value of \( \lambda_n \), and that it leads to no more than 5% overfitting in one of the segments. We illustrate this procedure in Figure 3, where we show \( \sigma_i(\lambda_1) \) for each of the three light curve sections for EPIC 206103150. The red arrows indicate the minimum of each of the curves, and the dashed vertical line indicates the adopted \( \hat{\lambda}_n \) based on a compromise between slight underfitting in the first two segments and slight overfitting in the third. This results in a more conservative cross-validation process than in Paper I.

### 4. Results

We detrended all stars in campaigns 0–8 and 10–13 with \( \text{nPLD} \) to produce the EVEREST 2.0 catalog. Campaign 9 is currently not available in the catalog because of the extreme crowding toward the galactic center, which can seriously affect the performance of PLD (see Section 5.2). Campaigns 10 and 11 were divided into two subcampaigns, which we refer to as 101 and 102, and 111 and 112, respectively, following the Kepler team’s convention. Because campaign 101 is extremely short, it is not included in the catalog. However, any target not present in the EVEREST 2.0 catalog may easily be detrended by the user via the DetrendFITS function (see Section 6.3.2).

Below, we report our results, starting with injection/recovery tests and a comparison of \( \text{rPLD} \) and \( \text{nPLD} \), followed by comparisons with other pipelines and the original Kepler light curves. We report most of our results in terms of the proxy 6 hr CDPP of the detrended light curves, which we calculate in the same way as we did in Paper I: we smooth the light curves with a Savitsky–Golay filter, clip outliers at 5\( \sigma \), and compute the median standard deviation in 13 cadence segments, normalized by \( \sqrt{13} \).

#### 4.1. Injection Tests

As in Paper I, we perform simple transit injection/recovery tests to ensure our model is not overfitting. For the same sample of 2700 campaign 6 stars as before, we inject synthetic transits of varying depths at the raw pixel level and attempt to recover them after detrending with \( \text{nPLD} \). We follow the exact same procedure as in Section 4.1 of Paper I and plot the results in Figure 4 (compare to Figure 6 in Paper I).

Each panel displays two histograms: a blue one, showing the number of transits recovered with a certain depth after detrending with \( \text{nPLD} \), and a red one, corresponding to a control run in which the transits were injected into the already detrended light curve. Each row corresponds to a different injection depth \( D_0 \) (10\(^{-2} \), 10\(^{-3} \), and 10\(^{-4} \), from top to bottom), and the x-axis in each histogram is the recovered depth \( D \) scaled to this value (\( D/D_0 \)). The left column corresponds to runs in which the transits were not explicitly masked during detrending; the right column shows runs in which they were.

As with the previous version of the pipeline, we find a \( \sim10\% \) bias toward smaller depths for low S/N transits when the transits are not explicitly masked. This is because a small decrease in the transit depth can greatly improve the CDPP of the light curve. Because the PLD regressors are noisy, the method is capable of partially fitting out transits by exploiting linear combinations of white noise in the regressors. This overfitting does not occur for high S/N transits because these are masked during the outlier clipping step.

Conversely, when transits are explicitly masked, there is no bias in the recovered depth; the median \( D/D_0 \) is consistent with unity for all three values of the injected depth. This is the same result we obtained with EVEREST 1.0, so we conclude that our new cross-validation scheme is robust in preventing overfitting when transits are masked. As before, we urge those using EVEREST light curves containing transits or eclipses to recompute the model with those features masked. This process is quick and straightforward—refer to the EVEREST 2.0 documentation for details.

#### 4.2. \( \text{rPLD} \)

In Figure 5, we plot a comparison of the CDPP values obtained with \( \text{rPLD} \) and EVEREST 1.0 for our sample set of 2700 campaign 6 stars. As before, the y-axis corresponds to the normalized relative CDPP of each model, with negative values corresponding to lower CDPP for \( \text{rPLD} \). Each star is plotted as a blue dot and the median relative CDPP is indicated as a black line. \( \text{rPLD} \) outperforms EVEREST 1.0 at all \( \text{Kepler} \) magnitudes by \( \sim1–6\% \) on average. However, the scatter at

---

Footnote 1: [https://rodluger.github.io/everest](https://rodluger.github.io/everest)
any value of $Kp$ is quite large, and the two models are roughly comparable for bright stars. As we argued in Section 3.7, the most important feature of $rPLD$ is its increased robustness to local overfitting (see Section 5).

### 4.3. $nPLD$

The greatest improvement in the CDPP comes when neighboring stars’ PLD vectors are included in the design matrix. In Figure 6, we plot the CDPP comparison between $nPLD$ and EVEREST 1.0. $nPLD$ outperforms regular PLD by $\sim 10\%$ on average, with the largest improvement occurring for fainter stars. Faint stars have the noisiest PLD vectors and benefit the most from the inclusion of higher S/N regressors. As we showed in Paper I, regular PLD already comes close to recovering Kepler photometric precision for bright ($Kp \lesssim 13$) stars, so the improvement for these stars is naturally smaller.

### 4.4. Comparison to Other Pipelines

In this section, we compare the EVEREST 2.0 catalog to those produced by other pipelines, beginning with the previous version of EVEREST.
Figure 7 shows the CDPP comparison between EVEREST 2.0 and EVEREST 1.0 for all stars in campaigns 0–8 (more recent campaigns are not available in the EVEREST 1.0 catalog). As in the example shown in Figure 6, EVEREST 2.0 outperforms EVEREST 1.0 by ~20% for the faintest stars and by ~10% for \( K_p \gtrsim 12 \). For \( 11 \lesssim K_p \lesssim 12 \), the two pipelines yield comparable results, though regularized regression gives EVEREST 2.0 a slight edge. Below \( K_p \approx 11 \), K2 stars become saturated; these are plotted as red dots, and the median relative CDPP for saturated stars is indicated by the dashed red line. For these stars, EVEREST 1.0 yields much lower CDPP: for \( K_p \lesssim 10 \), the CDPP is over a factor of two smaller than that of EVEREST 2.0. As we discussed in Paper I, the increased performance of EVEREST 1.0 for saturated stars is spurious because the astrophysical information content of the pixels is highly variable across the aperture, leading regular PLD to overfit. As we showed in Section 3.5, EVEREST 2.0 does not overfit saturated stars. In Section 4.9, we show that we approximately recover the Kepler photometric precision for these stars.

In Figure 8, we show the CDPP comparison between EVEREST 2.0 and K2SFF (Vanderburg 2014; Vanderburg & Johnson 2014) for campaigns 0–8 and 10–13. Our pipeline yields lower average CDPP at nearly all magnitudes, by 40% for faint stars and 10–20% for bright stars. For saturated stars, EVEREST 2.0 outperforms K2SFF by 5–10%.

In Figure 9, we show a comparison to the K2SC PDC light curves (Aigrain et al. 2015, 2016) for campaigns 3–6. EVEREST 2.0 yields lower average CDPP at all magnitudes \( K_p \gtrsim 9 \), with a 20–25% improvement for all unsaturated stars. For \( K_p \approx 9 \), K2SC slightly outperforms EVEREST 2.0, but the scatter in the plot is quite large.

We also computed the relative CDPP for each of the campaigns individually; these are shown in Figures 10–12. The improvement over EVEREST 1.0 is approximately the same for all campaigns despite significant differences in the noise properties and stellar populations across the nine campaigns, showcasing the robustness of the nPLD method. The same is true when compared to K2SFF and K2SC, with the exception of campaigns 0–2, for which EVEREST 2.0 outperforms K2SFF by nearly 50% (i.e., a factor of two) at all magnitudes.

4.5. Outliers

In addition to yielding lower average CDPP than any other publicly available pipeline, EVEREST yields the largest number of usable data points for any of the K2 campaigns to date. These data points generally appear as outliers even in the raw data because they are highly sensitive to inter- and intrapixel sensitivity variations. Therefore, pipelines that regress only on functions of the spacecraft motion have trouble detrending them, resulting in ~5–10% of the data points being discarded as outliers. In contrast, because PLD uses regressors containing both inter- and intrapixel sensitivity information, it naturally detrends data collected during thruster-firing events (see Paper I).

To show this, we calculated the number of non-outlier data points per campaign for each of the four pipelines (K2SFF, K2SC, EVEREST 1.0, and EVEREST 2.0). After removing all data points with flagged QUALITY bits 1–9, 11–14, and 16–17, we smoothed each light curve with a Savitsky–Golay filter and performed iterative sigma clipping to remove all 5σ outliers. In Figure 13, we plot histograms of the number of remaining data points for all stars in each of the first nine campaigns. As expected, both EVEREST 1.0 and EVEREST 2.0 have, on average, 100–300 more usable data points than the other two pipelines. This roughly corresponds to the number of thruster firings per campaign, as these happen every 6–12 hr on average.

4.6. Short Cadence

In Figure 14, we plot the relative CDPP distribution of 671 light curves that were observed at both short and long cadence in campaigns 0–8. In order to compute the 6 hr CDPP of short-cadence light curves, we first mask outliers and then down-bin to long cadence by taking the mean of every 30 cadences; we then compute the CDPP as usual. We achieve higher average precision in the short-cadence light curves by 5–10% for unsaturated stars and by up to 25% for saturated stars. The higher information content in the short-cadence light curves—particularly at timescales under 30 min—allows EVEREST to better detrend those stars. We show an example in Figure 15, where we plot the light curves for EPIC 201601162 (raw short cadence at the top, detrended short cadence in the center). In the bottom panel, we show both the down-binned short-cadence detrended light curve (black) and the long-cadence detrended light curve (red). The CDPP in the short-cadence light curve is ~30% smaller than that of the long-cadence light curve.

As we discussed in Section 3.6, one pitfall of our method for detrending short-cadence targets is the large number of
breakpoints (~30) we introduce in the light curve when computing the model. Overcomputing the model into adjacent

segments and aligning the models at the breakpoints works very well for high S/N light curves, but it can often fail for light curves that are dominated by photon noise. This is the case for very faint stars (Kp ≈ 17) observed in short-cadence mode, such as EPIC 201831393, which displays visible discontinuities at many of the breakpoints. Low S/N short-cadence light curves may also have segments with varying white noise amplitudes due to differences in the value of the A prior on the PLD weights. PLD is known to perform poorly when the white noise dominates (Deming et al. 2015); in these cases, it is often desirable to down-bin the light curve and compute the model on a higher S/N signal, then predict onto the original short-cadence data. Given the relatively small number of light curves for which this is an issue, we do not do this here.

4.7. Cotrending Basis Vectors (CBVs)

One downside of the algorithm employed by EVEREST is that GP regression has trouble distinguishing between low-frequency stellar variability and low-frequency instrumental systematics. Many of the EVEREST light curves display a steady rise over the course of the campaign, with hook-like features at the beginning, end, or both. While this does not affect transits or other high-frequency astrophysical signals, it
could potentially lead to biases in stellar rotation studies that rely on low-frequency modulation in the light curves.

We therefore run a post-processing step on all detrended light curves to remove these residual instrumental signals. After some experimentation, we decided to use the SysRem algorithm \citep{Tamuz2005}, which identifies and removes signals shared by many light curves. SysRem is similar to PCA but allows for weighting of the input signals, and therefore is better suited to dealing with light curves of varying noise properties. Our approach is similar to the presearch data conditioning (PDC) algorithm of the Kepler pipeline, which uses CBVs from many stars on the detector to remove common instrumental signals from light curves \citep{Stumpe2012,Smith2012}.

We apply SysRem to all of the light curves in each campaign, weighting each one by the quadrature sum of the flux measurement errors and the white noise component of its GP. Given known correlations between instrumental signals and spatial position on the detector \citep[e.g.,][]{Petigura2012,Wang2016}, we attempted to apply SysRem individually on each CCD module, but found that the recovered signals were often dominated by astrophysical variability originating from the brightest star(s) in each module. We found that computing the SysRem signals for the entire detector alleviated this issue without compromising the detrending power of the method.

We separately apply SysRem to each light curve segment, obtaining one CBV for each segment of each campaign. We then subtract a linear fit of this CBV from each light curve and repeat the procedure to obtain additional CBVs for each segment of each campaign. In order to prevent the CBVs from fitting out or introducing high-frequency signals in the light curves, we aggressively smooth them with a third-order, 1000 cadence Savitsky–Golay filter. The results for the first nine campaigns are shown in Figure 16. The first CBV for each segment of each campaign is plotted in blue, and the second in red. As expected, the first CBV in each segment is dominated by a linear trend with S-like hooks on either end. The second CBVs are predominantly quadratic or cubic. The remaining CBVs (not shown) are dominated by higher-order trends.

For the purpose of generating the EVEREST 2.0 catalog, we perform ordinary least squares regression to fit all detrended light curves using only the first CBV (blue curves in the figure). We find that fitting with additional CBVs often helps to remove additional systematics—particularly the hook-like features mentioned above—but may lead to overfitting of true astrophysical variability in some light curves. Therefore, we include all five CBVs of each segment of each campaign in the EVEREST 2.0 FITS files so that targets may be further corrected by the user on a case-by-case basis (see Section 6).

4.8. Sample Light Curves

In Figures 17 and 18, we show two sample light curves detrended with both K2SFF and EVEREST 2.0. Figure 17 shows EPIC 201345483, a faint Kp = 15 planet host. The CDPP of the EVEREST light curve (bottom) is a factor of 2.4 lower than that of K2SFF (top), and the light curve has visibly fewer outliers. The folded transit is shown at the right and is similarly less noisy in the EVEREST light curve.

In contrast, Figure 18 shows a very bright (Kp = 10) saturated planet candidate host, EPIC 201862715. With the
column-collapsing scheme and the inclusion of neighboring PLD vectors, EVEREST 2.0 is able to achieve a factor of five lower CDPP than K2SFF, as well as considerably fewer outliers.

Last, in Figure 19, we show the light curve of EPIC 220383386 (HD 3167), for which K2SFF significantly outperforms EVEREST. While the two transiting planets are clearly visible in the K2SFF light curve (an ultra-short period planet at a period of 0.96 days, and a planet at a period of 29.85 days; see Vanderburg et al. (2016)), residual systematics make it difficult to discern the transits in the EVEREST light curve, particularly in the first half of the campaign. The detrending failed in this case, because of the deep, ultra-short period transits of HD 3167b, which drive the GP kernel to a short timescale and a high amplitude. Because this timescale is comparable to that of the spacecraft motion systematics, a significant fraction of the instrumental signal is captured by the GP, rather than the PLD model, leading to poor detrending. This tends to also happen for stars that are variable on very short timescales, which we discuss in more detail in Section 5.1 below. Tests show that the performance of EVEREST can be greatly improved for such light curves by simultaneously optimizing the GP and the PLD, a computationally costly step that we defer to a future version of the code.

4.9. Comparison to Kepler

In Figures 20 and 21, we compare the EVEREST 2.0 photometric precision to that of the original Kepler mission. Figure 20 shows the CDPP as a function of Kepler magnitude for K2 campaigns 0–8 and 10–13 (excluding the first subcampaign of 10).

Figure 11. Similar to Figure 8, but showing a CDPP comparison between EVEREST 2.0 and K2SFF for K2 campaigns 0–8 and 10–13 (excluding the first subcampaign of 10).
raw photometric precision and in the stellar populations across the campaigns, the results are variable, but for all campaigns except 0, 2, 7, 11, and 12, we recover the original *Kepler* precision down to at least $K_p = 14$. For campaigns 1, 5, 6, and 13, we recover the *Kepler* precision down to at least $K_p = 15$. This also applies to saturated stars, though the EVEREST 2.0 CDPP is slightly higher for these stars in some campaigns. For stars dimmer than $K_p = 15$, the EVEREST 2.0 CDPP is within a few tens of percent—or less—than that of *Kepler*.

5. Additional Remarks

5.1. Variable Stars

In Paper I, we discussed how *EVEREST* often fails to properly detrend stars that display high-amplitude variability on very short timescales, such as RR Lyrae variables and very short-period eclipsing binaries, causing overfitting in many of these light curves. We attributed this to there being too much power in the GP model, which captured both the astrophysical and the instrumental variability, resulting in an improperly optimized PLD model. However, after considerable experimentation, we found that this behavior stemmed in large part from our cross-validation scheme. In Paper I, our cross-validation sets were 13 cadences (6.5 hr) long, and we sought to minimize the median (proxy) CDPP of all such sets. We chose this timescale because it is roughly the duration of a typical transit, and in practice it worked well to minimize overfitting of transits. However, the CDPP as defined in Section 4—the normalized standard deviation in 13 cadence segments after the application of a high-pass filter—is not an adequate metric of the photometric precision for stars that are intrinsically variable on similarly short timescales. In other words, RR Lyrae and other high-amplitude, short-period variables are dominated by astrophysical variability on the timescale at which the CDPP is computed, and therefore minimizing the CDPP is a recipe for overfitting.

In *EVEREST* 2.0, we modified our cross-validation scheme (Section 3.7) in two important ways: we increased the average size of the validation sets to $\sim$500 cadences, and we minimized the validation scatter after the subtraction of a properly optimized GP model. This helps to ensure that we minimize only the instrumental component of the noise. In order to assess the performance of this new scheme, we visually inspected the detrended light curves of 100 RR Lyrae stars in campaigns 0–4, chosen as the targets with the highest probability of being RRab stars according to the K2VARCAT catalog (Armstrong et al. 2016). Among the *EVEREST* 1.0 light curves, 92/100 had visibly damped oscillation amplitudes or clearly overfitted stellar variability features. In contrast, only 44/100 *EVEREST* 2.0 RR Lyrae stars showed any signs of overfitting. For eight of these, the overfitting occurred only with the inclusion of the second- or third-order PLD models. While there are still issues with how the pipeline handles extremely variable stars, the improvement over version 1.0 is substantial.

For comparison, we visually inspected the same stars in the *K2SFF* and *K2SC* catalogs. For the *K2SFF* light curves of RRab stars, 96 out of 100 showed signs of overfitting (dampened oscillation amplitudes, distorted astrophysical signal, or significant detrending artifacts), while only 12/65 (18%) of *K2SC* light curves appeared to be incorrectly detrended (35 of the 100 stars were in campaigns 0–2 and are not present in the *K2SC* catalog). *K2SC* likely outperforms the other pipelines for these stars because of its robust GP optimization scheme. Because we optimize the GP and detrend the light curve in separate steps, the covariance matrix we use is often an improper approximation to the true covariance of the astrophysical signal. Progressive optimization of the GP (Section 3.2) helps to maximize the amount of astrophysical information captured by the GP model, but a better procedure would be to simultaneously fit for the GP hyperparameters and the PLD coefficients. This would ensure that the PLD model captures only instrumental signals and the GP model captures only astrophysical signals. However, such a method is too computationally expensive because the problem would no longer be linear. We therefore settle for our linear method, which works extremely well for stars that do not exhibit high-amplitude, short-timescale variability, and it has a $\sim$50% success rate for those that do. We encourage readers interested...
in RR Lyrae and other highly variable stars to inspect the light curves of the different pipelines on a target-by-target basis. In particular, users should compare the amplitude of the signal in the raw light curve to that in the detrended light curve. The variability amplitude of RR Lyrae is typically on the order of 0.5–1 mag, so the astrophysical signal for stars with $K_p \lesssim 15$ should dominate over instrumental variability, making it relatively easy to tell by eye if a target has suffered from overfitting. We have produced a quick tutorial to help with identifying affected light curves and provide tips on how to improve the performance of EVEREST; it is available at https://rodluger.github.io/everest/issues.html#rr-lyrae.

5.2. Crowded Apertures

In Paper I, we also showed how PLD is not suited to detrend stars in excessively crowded fields, because the algorithm will often use pixels from contaminant sources to fit out the target star’s astrophysical signals. Directly addressing this issue is far less straightforward than mitigating the effects of saturation (Section 3.5) or extreme astrophysical variability (Section 5.1), as it likely requires accurate modeling and subtraction of the PSFs of contaminant sources. However, in practice we find that our new detrending algorithm is far more robust to overfitting when contaminant sources are present. This is partly due to the inclusion of PLD regressors from neighboring stars, which reduce the contribution from potentially contaminated pixels in the target aperture. Our new cross-validation scheme (Section 3.7) also helps guard against the effects of crowding.
because it is better at preventing localized overfitting. Due to the large amplitude of the $K_2$ spacecraft drift, the amount of contamination in the target aperture often varies considerably over the duration of the campaign, as nearby sources move in and out of the aperture. In EVEREST 1.0 light curves, this resulted in a time-variable detrending power—and time-variable overfitting—for crowded targets. As we discussed in Section 3.7, our new cross-validation method strongly disfavors this behavior.

In Figure 22, we show the light curves of five eclipsing binaries whose crowded apertures resulted in severe overfitting in the EVEREST 1.0 catalog. Shown are the light curves (flux versus time) folded on the period of the binary and centered on the primary eclipse for EVEREST 1.0 (left) and EVEREST 2.0 (center). The right panel shows the Palomar Optical Sky Survey8 (POSS II) red filter image of the target postage stamp, where contaminant sources are clearly visible in each of the apertures (indicated by the red contours).

The first three targets (EPIC 202072978, EPIC 202103762, and EPIC 218803648) have relatively bright contaminant sources centered on or near the edge of the aperture. In Paper I, we explained how this was the “worst case” scenario for crowding, as it results in the largest spatial variation of astrophysical information from the target across the aperture; as expected, EVEREST 1.0 almost completely fits out their eclipses. EVEREST 2.0, on the other hand, achieves high detrending power while preserving the eclipse shapes and depths seen in the raw light curves. The fourth target (EPIC 202733088) has a bright nearby source inside its aperture, separated by less than a pixel. In Paper I, we discussed how targets with co-located contaminants are typically unaffected by PLD overfitting because the ratio of the two PSFs is roughly constant everywhere. However, in this case, the contaminant is sufficiently detached to result in severe overfitting in the EVEREST 1.0 light curve. As before, EVEREST 2.0 correctly detrends the light curve with no overfitting. The final target (EPIC 211685048) is an eclipsing binary with deep transits that are completely fit out in both versions of the pipeline. Here, two roughly equal magnitude stars are fully contained in the aperture and separated by $\sim 3$ pixels, causing PLD to fail despite the modifications to the algorithm.

As in Paper I, we urge caution when using the EVEREST light curves of crowded targets, but note that EVEREST 2.0 succeeds in removing the instrumental noise without overfitting astrophysical information in most of the cases known to us in which the previous version overfitted transits and eclipses. Future updates to the pipeline will include a more careful aperture selection, which can mitigate the effects of crowding for targets like EPIC 211685048. It is important to note that crowding is an issue specifically for PLD, which exploits correlations among different pixels on the detector to extract instrumental signals and therefore requires a spatially homogeneous astrophysical signal. Other pipelines, such as K2SFF and K2SC, are not affected in the same way by a crowded field; in fact, the K2SFF and K2SC light curves for EPIC 211685048 show no signs of overfitting. We recommend the light curves from these pipelines when bright contaminant stars are present in the aperture.

---

8 http://stdatu.stsci.edu/cgi-bin/dss_form

---

**Figure 15.** EPIC 201601162, a campaign 1 star observed in both long- and short-cadence modes. A portion of the raw light curve is displayed at the top, and the detrended light curve is shown in the center. In the bottom panel, we plot the down-binned, detrended, short-cadence light curve (black) and the detrended, long-cadence light curve (red). The y-axis in each panel is the flux in $e^{-}$ s$^{-1}$. Short-cadence EVEREST 2.0 light curves have lower CDPP than their long-cadence counterparts.
5.3. Aperture Losses

As we have demonstrated, column-collapsed PLD works extremely well for saturated stars, provided all of the target flux in the saturated columns is used. With this in mind, we used larger apertures for these stars and extended them by two pixels at the top and bottom of all saturated columns (Section 3.5). However, for some extremely bright \( (Kp \approx 8.5) \) targets, bleed trails in the saturated columns can extend past the edge of the target postage stamp. Because most of the astrophysical information in saturated columns is contained in the two pixels at the top and bottom, this can lead to substantial information loss. In this case, the column-collapsing procedure fails to satisfy the PLD constraint that all pixels should have the same fractional astrophysical signal strength, and overfitting can occur. This is the case for EPIC 210703831, a campaign \( 4 \) \( Kp = 8.1 \) star whose detrended light curve displays several discontinuities. These occur because spacecraft drift results in aperture losses only during parts of the campaign. Therefore, we encourage those using the EVEREST catalog to inspect the postage stamps of extremely bright stars to ensure that aperture losses are not present.

5.4. Overfitting Metrics

In the previous section, we focused on the CDPP of the detrended light curves as a means of comparing EVEREST to other publicly available pipelines. However, as we argued above, a pipeline’s robustness against overfitting is a similarly important metric to consider. In Section 4.1, we computed the tendency of EVEREST to overfit based on transit injection/recovery tests by measuring the recovered transit depth divided by the injected transit depth, \( D/D_0 \). We did this twice, once for light curves in which the transits were not masked, and once for light curves in which the transits were masked (thus preventing the PLD model from being trained on data collected during transits). We showed that, while EVEREST tends to overfit shallow transits by about 10% on average when they are not masked, the overfitting is negligible when they are properly masked.

The results in Figure 4 show considerable spread in \( D/D_0 \), meaning individual \( K2 \) light curves suffer from different amounts of overfitting. Moreover, we find that the overfitting is also variable within light curves, and tends to be significantly higher close to discontinuities due to thruster firings when the transits are not masked. In order to quantify this overfitting for individual light curves, we have written code that enables the user to easily compute two different overfitting statistics: the unmasked overfitting metric \( \mathcal{O}_u \) and the masked overfitting metric \( \mathcal{O}_m \), both defined as

\[
\mathcal{O} = 1 - \left( \frac{D}{D_0} \right),
\]

the average fractional decrease in depth for injected transits when the transits are unmasked (\( \mathcal{O}_u \)) and masked (\( \mathcal{O}_m \)). For any particular star, we compute these by sliding an injected transit across every cadence in the raw light curve and computing its maximum likelihood depth after detrending. Because EVEREST is a linear model, both of these metrics are relatively fast to evaluate. Users can compute them by executing

```
$ everest XXXXXXXXXX -o
```

in a Unix shell, where XXXXXXXXXX is the nine-digit EPIC ID of the target. For more information, refer to the online documentation.

As expected from the results in Section 4.1, the average values of \( \mathcal{O}_u \) and \( \mathcal{O}_m \) across all light curves are about 0.10 and

![Figure 16. Cotrending Basis Vectors (CBVs) for each of the first nine campaigns. We apply SysRem to all detrended light curves in each campaign to obtain the first (blue) and second (red) CBVs; we do this independently for each of the segments in each campaign. The first set of CBVs contains primarily linear trends with hook-like features at the beginning or end of the segments; the second set of CBVs is dominated by quadratic or cubic trends. We correct all light curves by simple linear regression with the first two CBVs.](image-url)
The folded transit of K2-45b is shown at right. The CDPP of each light curve is indicated in the top left. The folded transit of K2-45b is shown at right. The EVEREST 2.0 light curve has 2.4× higher photometric precision.

When using the EVEREST light curves for science purposes, it is important to properly reject these outliers. Data flagged with bit 23 is typically affected by cosmic rays or detector anomalies, and can usually be ignored. In some cases, however, deep transits or eclipses can be mistaken for detector anomalies and are incorrectly flagged in the original TPF, so a visual inspection of the light curves is recommended. Data flagged with bit 24, on the other hand, is missing in the original TPF and can thus be safely ignored. Finally, bit 25 usually corresponds to instrumental outliers that were not properly detrended with PLD. However, transits, flares, and other short-timescale astrophysical features will also be flagged with this bit, so this data should not be blindly excluded when performing transit searches.

The four remaining FITS extensions include the PLD regressors, the target aperture, and additional information used internally by the Python code.

### 6.1. FITS Files

Each FITS file contains six extensions. The primary (index 0) extension consists of a header with miscellaneous target information copied from the K2 TPF. The second (index 1) extension contains a header and a binary table. The header stores miscellaneous information about the target and the settings used in the detrending, such as the GP hyperparameters and information on the neighboring targets used in the regression. The binary table stores arrays corresponding to the cadence number (CADN), the timestamp (TIME), the raw SAP flux (FRAW), the raw SAP flux errors (FRAW_ERR), the PLD-detrended flux (FLUX), the five CBV regressors (CBV01—CBV05), and the detrended flux with the CBV correction (FCOR). This extension also includes the original K2 QUALITY bit array, with four additional bits that indicate which cadences were masked when computing the model:

| Bit | Description               |
|-----|---------------------------|
| 0   | NaN in K2 TPF             |
| 1   | NaN in K2 TPF             |
| 4   | NaN in the K2 TPF         |
| 8   | NaN in the K2 TPF         |
| 16  | NaN in the K2 TPF         |

### 6.2. Data Validation Summaries (DVSs)

Each target in the catalog also has an associated DVS, a PDF document showing the raw, detrended, and CBV-corrected light curves, as well as cross-validation diagnostic plots such as those in Figure 3 and a high-resolution POSS image of the target aperture like those shown in Figure 22.

### 6.3. Python Code

The EVEREST code can be installed using the package managing system pip:

```
pip install everest-pipeline
```

or directly from source by following the instructions on the github page. The primary way of interfacing with the catalog is to instantiate an EVEREST object:

```python
import everest
cat = everest.Everest(EPIC)
```

where EPIC is the EPIC number of the target. These lines download the target’s FITS file and populate the `cat` object with the detrending information and light curve arrays.
flux, fcor, etc.). The QUALITY flags 23, 24, 25, and 27 are used to generate the badmask, nanmask, outmask, and transitmask arrays, respectively; these arrays contain the indices of all data points whose corresponding bit is flagged. The detrended light curve can be viewed by executing

```python
star.plot()
```

or

```python
star.dvs()
```

the latter of which displays the DVS report for the target. The code implements various other visualization routines, which are described in detail in the documentation at [https://rodluger.github.io/everest](https://rodluger.github.io/everest).

### 6.3.1. Dealing with Transits

As we discussed in Section 4.1, transits and other such high-frequency astrophysical signals can impact the performance of PLD and lead to slight overfitting. One way to prevent this is to mask known transits when computing the PLD model. Users can easily do this as follows:

```python
star.mask_planet(time, period, duration)
star.compute()
```

where `time` is the time of first transit (in units of BJD − 2454833), `period` is the transit period (in days), and `duration` is the full transit duration (also in days). Recomputing the model takes no more than a few seconds for long-cadence light curves.

Alternatively, users may simultaneously optimize a transit model along with the instrumental model. Because the PLD model is linear, it is straightforward to add a column corresponding to the transit model to the design matrix (Equation (10)). This is done by specifying

```python
model = everest.TransitModel
(name,**kwars)
```

```python
star.transit_model = model
```

prior to running `compute()`. Above, `name` is the planet identifier (a string, such as “b”), and `kwars` are keyword arguments describing the period (per), time of first transit (t0), impact parameter (b), etc.; refer to the documentation for details. The corresponding term in Equation (8) is then the maximum likelihood transit depth, given fixed values of the transit time, period, and shape. This can be accessed as

```python
depth = star.transit_depth
```

Users may also assign a list of transit models to `star.transit_model` in the case of a multiplanet system, for which `star.transit_depth` stores a list of the corresponding maximum likelihood depths.

It is important to note that a transit model is linear only in the depth, so if properties such as the period and time of transit are not known, the previous method is of little use. Fortunately, the linear nature of the PLD model can once more be of aid, allowing us to compute the value of these properties marginalized over the parameters of the PLD model. As in Luger et al. (2017a), the marginal likelihood of a zero-baseline, unit depth transit model $m_0$ with parameters (period, time of transit, etc.) given by the vector $\theta$ is

$$
\log \mathcal{L} = -\frac{1}{2}(y - \delta m_0^T \Sigma^{-1} \cdot (y - \delta m_0) + C
$$

(20)

where $C$ is a constant, $y$ is the (raw) SAP flux, and $\delta$ is the transit depth. The data covariance matrix, $\Sigma$, is the sum of the astrophysical covariance and the PLD covariance, and is given by

$$
\Sigma = X \cdot \Lambda \cdot X^T + K
$$

(21)

where $K$ is the astrophysical covariance given by the GP model, $X$ is the design matrix, and $\Lambda$ is the regularization matrix. In principle, the transit depth $\delta$ could be included in the parameter vector $\theta$, but because the model is linear with respect to it, the maximum likelihood depth can be obtained from (e.g., Luger et al. 2017b)

$$
\delta = \sigma_\delta^2 m_0^T \cdot \Sigma^{-1} \cdot y,
$$

(22)

where

$$
\sigma_\delta^2 = (m_0^T \cdot \Sigma^{-1} \cdot m_0)^{-1}
$$

(23)

is the variance on the depth estimate.
The EVEREST code makes it easy to compute this likelihood. Given the transit model object (model) from above, the log-likelihood (Equation (20)) may be computed by calling

\[ m = \text{model}(\text{star.time}) \]
\[ \ln\text{like} = \text{star.lnlike}(m) \]

To return the maximum likelihood depth and variance in addition to the likelihood, run

\[ \ln\text{like}, \text{depth, vardepth} = \text{star.lnlike}(m, \text{full_output}=\text{True}) \]

Executing the above statements is fast because the covariance (Equation (21)) need only be inverted once. Therefore, this method is ideal for use in MCMC chains and other optimization algorithms in which a likelihood must be specified to obtain estimates of the posterior distributions or maximum likelihood values of transit parameters. Refer to the tutorial section of the documentation for examples on how to do this.

Finally, it is important to note that, while this section deals explicitly with transit models, in principle any astrophysical model can be passed to the \text{lnlike()} method. For instance, the user may provide a flare model as a function of \( \theta = \{\text{time of flare, flare timescale, etc.}\} \) and solve for the posterior distribution of the flare properties, marginalized over the instrumental model. It is also important to bear in mind that any model passed to \text{lnlike()} must be zero-baseline and unit amplitude. This is because the scalar amplitude of the model is a linear parameter whose posterior is normal and can be obtained from some quick linear algebra. Its value for any instance of the model can be obtained by setting the \text{full_output} flag in calls to \text{lnlike} (see above).

6.3.2. Customized Detrending

The final point we wish to highlight regarding version 2 of the software is the option to perform detrending on custom K2 FITS files. This is particularly useful for detrending uncalibrated FITS files, which are typically released by the Kepler team ahead of the pipeline-processed TPFs:

\[
\text{everest.DetrendFITS(fitsfile)}
\]

where \text{fitsfile} is the path to the raw FITS file for the target. Refer to the documentation for more information.

7. Conclusions

We have presented EVEREST 2.0, an update to the EVEREST pipeline (Luger et al. 2016) for removing instrumental noise from K2 photometry. In version 1.0, we constructed a linear model from the principal components of products of the fractional pixel fluxes in the aperture of each star, a variant of a method known as PLD (Deming et al. 2015). Here, we regress on all PLD vectors, imposing Gaussian priors on the model weights to prevent overfitting. We additionally include the PLD vectors of bright neighboring stars to increase the signal-to-noise ratio of the regressors and enhance the predictive power of the model. We developed a fast GP regression scheme to detrend all stars in the K2 catalog, achieving lower CDPP than in version 1.0, by \( \sim 10\% \) for bright stars and \( \sim 20\% \) for faint stars. We also adapted PLD to work for saturated stars, yielding comparable detrending power, and stars observed in short-cadence mode, yielding higher photometric precision on 6 hr timescales than for their long-cadence counterparts. We further find that the inclusion of neighboring PLD vectors and a more conservative cross-validation scheme enhance the pipeline’s robustness to overfitting, particularly for highly variable stars.

EVEREST 2.0 light curves have higher photometric precision than the two other publicly available catalogs, K2SFF (Vanderburg & Johnson 2014) and K2SC (Aigrain et al. 2016), at all \( K_p \) magnitudes. For faint stars, EVEREST 2.0 has \( \sim 40\% \) lower CDPP than K2SFF and \( \sim 25\% \) lower CDPP than K2SC; for bright unsaturated stars, the CDPP improvement is \( \sim 20\% \) compared to both pipelines. For saturated stars, EVEREST outperforms both pipelines, but by a smaller margin. We also find that EVEREST light curves have, on average, 100–300 fewer outliers than those of other pipelines, owing primarily to the ability of PLD to correct data collected during thruster-firing events. It is important to
remember that there exists a trade-off between a pipeline’s detrending power and its proneness to overfitting. We find that the higher photometric precision in EVEREST light curves comes at the cost of more overfitting than in K2SFF and K2SC. However, we also find that our algorithm is robust to overfitting if transits and other astrophysical signals of interest are properly masked or modeled, which can be done easily using our Python code (see Section 6.3.1 and below).

When compared to the original Kepler mission, EVEREST 2.0 recovers Kepler photometry on average to $K_p \approx 14.5$, and past $K_p = 15$ for some campaigns. For dimmer stars, the CDPP is within a few tens of percent of that of Kepler. EVEREST light curves should thus enable continued high-precision transiting exoplanet and stellar variability science for the vast majority of K2 stars as if they had been observed by the original four-wheeled mission.

The EVEREST 2.0 catalog of detrended light curves is publicly available from the MAST archive (EVEREST, doi:10.17909/T9501J; https://archive.stsci.edu/prepds/everest). As with the previous version of the code, EVEREST 2.0 is open source under the MIT license and available at https://github.com/rodluger/everest, with a static release of
the code used to generate the catalog archived on Zenodo (Everest 2.0, doi: 10.5281/zenodo.1302521; 10.5281/zenodo.1302521). The reader is encouraged to use this code to interface with the EVEREST catalog and to customize the detrending of targets of interest, particularly for masking transits to remove biases in the depth due to overfitting. We have implemented each of the PLD models discussed above (rPLD, nPLD, pPLD) in a general framework that can be adapted to different missions, including Kepler and the upcoming TESS. For more information, refer to the documentation at https://rodluger.github.io/everest.

We would like to thank Benjamin Pope and Tsevi Mazeh for their useful comments and suggestions. R.L., R.B., and E.A. acknowledge support from NASA grant NNX14AK26G and from the NASA Astrobiology Institute’s Virtual Planetary Laboratory Lead Team, funded through the NASA Astrobiology Institute under solicitation NNH12ZDA002C and Cooperative Agreement Number NNA13AA93A. E.A. acknowledges additional support from NASA grants NNX13AF20G and NNX13AF62G and National Science Foundation grant AST-1615315. E.K. acknowledges support from an NSF Graduate Student Research Fellowship. This research used the advanced computational, storage, and networking infrastructure provided by the Hyak supercomputer system at the University of Washington.

ORCID iDs

Rodrigo Luger @ https://orcid.org/0000-0002-0296-3826
Ethan Kruse @ https://orcid.org/0000-0002-0493-1342
Daniel Foreman-Mackey @ https://orcid.org/0000-0002-9328-5652
Eric Agol @ https://orcid.org/0000-0002-0802-9145

References

Aigrain, S., Hodgkin, S. T., Irwin, M. J., Lewis, J. R., & Roberts, S. J. 2015, MNRAS, 447, 2880
Aigrain, S., Parvainen, H., & Pope, B. J. S. 2016, MNRAS, 459, 2408
Armstrong, D. J., Kirk, J., Lam, K. W. F., et al. 2015, A&A, 579, A19
Armstrong, D. J., Kirk, J., Lam, K. W. F., et al. 2016, MNRAS, 456, 2260
Christiansen, J. L., Jenkins, J. M., Caldwell, D. A., et al. 2012, PASP, 124, 1279
Crossfield, I. J. M., Petigura, E., Schlieder, J. E., et al. 2015, ApJ, 804, 10
Deming, D., Knutson, H., Hammer, J., et al. 2015, ApJ, 805, 132
Foreman-Mackey, D., Montet, B. T., Hogg, D. W., et al. 2015, ApJ, 806, 215
Golub, G. H., & Van Loan, C. F. 1996, Matrix Computations (3rd ed.; Baltimore: Johns Hopkins Univ. Press)
Howell, S. B., Sobeck, C., Haas, M., et al. 2014, PASP, 126, 398
Huang, C. X., Penev, K., Hartman, J. D., et al. 2015, MNRAS, 454, 4159
Libralato, M., Bedin, L. R., Nardiello, D., & Pietto, G. 2016a, MNRAS, 456, 1137
Libralato, M., Nardiello, D., Bedin, L. R., et al. 2016b, MNRAS, 463, 1780
Luger, R., Agol, E., Kruse, E., et al. 2016, AJ, 152, 100
Luger, R., Foreman-Mackey, D., & Hegg, D. W. 2017a, RNASS, 1, 7
Luger, R., Sestovic, M., Kruse, E., et al. 2017b, NatAs, 1, 0129
Lund, M. N., Handberg, R., Davies, G. R., Chaplin, W. J., & Jones, C. D. 2015, ApJ, 806, 30
Petigura, E. A., & Marcy, G. W. 2012, PASP, 124, 1073
Savitzky, A., & Golay, M. J. E. 1964, AnalCh, 36, 1627
Smith, J. C., Stumpe, M. C., Van Cleve, J. E., et al. 2012, PASP, 124, 1000
Stumpe, M. C., Smith, J. C., Van Cleve, J. E., et al. 2012, PASP, 124, 985
Tamuz, O., Mazeh, T., & Zucker, S. 2005, MNRAS, 356, 1466
Vanderburg, A. 2014, arXiv:1412.1827
Vanderburg, A., Bieryla, A., Duev, D. A., et al. 2016, ApJL, 829, L9
Vanderburg, A., & Johnson, J. A. 2014, PASP, 126, 948
Wang, D., Hogg, D. W., Foreman-Mackey, D., & Schölkopf, B. 2016, PASP, 128, 094503

Figure 22. Five eclipsing binaries with significant contamination by bright near-by stars. The first two panels in each row show the folded EVEREST 1.0 and EVEREST 2.0 light curves, respectively, and the third shows the POSS high-resolution image of the target postage stamp with our adopted aperture indicated in red. The magnitudes of the target and its bright neighbors are also indicated. While EVEREST 1.0 severely overfits the eclipses of all five targets, EVEREST 2.0 preserves the eclipse depths in all but the last one.