Predicting Resource Requirement in Intermediate Palomar Transient Factory Workflow

Qiao Kang*, Alex Sim‖, Peter Nugent‖, Sunwoo Lee*, Wei-keng Liao*, Ankit Agrawal*,
Alok Choudhary* and Kesheng Wu‖

* *Electrical and Computer Engineering, Northwestern Univeristy*
Evanston, IL, USA
{qiao.kang, slz839, wkliao, ankitag, choudhar}@eecs.northwestern.edu

‖ *Lawrence Berkeley National Laboratory*
Berkeley, CA, USA
{asim, penugent, kwu}@lbl.gov

***Abstract*—Quickly identifying astronomical transients from synoptic surveys is critical to many recent astrophysical discoveries. However, each of the data processing pipelines in these surveys contains dozens of stages with highly varying time and space requirements. Properly predicting the resources required to run these pipelines is critical for the allocation of computing resources and reducing the discovery response time. We propose a machine learning strategy for this prediction task and demonstrate its effectiveness using a set of timing measurements from the intermediate Palomar Transient Factory (iPTF) workflow. The proposed model utilizes the spatiotemporal correlation of astronomical images, where nearby patches of the sky (space) are likely to have a similar number of objects of interest and workflows executed in the recent past (time) are likely to use a similar amount of time because the machines and data storage systems are likely to be in similar states. We capture the relationship among these spatial and temporal features in a Bayesian network and study how they impact the prediction accuracy. This Bayesian network helps us to identify the most influential features for predictions. With proper features, our models achieve errors close to the random variance boundary within batches of images taken at the same time, which can be regarded as the intrinsic limit of prediction accuracy.**

***Keywords*-Workflow Scheduling, iPTF, Spatiotemporal features**

I. Introduction

A transient astronomical event, or simply a transient, is an astronomical object or event with a relatively short lifespan, instead of having a fixed location in the sky, like a star, or a periodic orbit, like a planet [1]. Studying transients such as the thermonuclear or core-collapse of a supernova helps researchers to understand the life-cycle of stars and their interaction in the formation of galaxies and the evolution of our universe. Thus there are several large synoptic surveys searching for these transients, for example, SLOAN [2], CATALINA [3], ATLAS [4], the Palomar Transient Factory (PTF) [5], and the Zwicky Transient Facility (ZTF) [6]. Since the objects of interest in these projects are short-lived, it is critical to be able to process the images taken quickly, so that detailed follow-up observations can be made on the transients just identified [7]–[10]. For this reason, it is important to anticipate the resources required for processing the images from these survey telescopes. Our focus is on the Intermediate Palomar Transient Factory (iPTF) [11], a successor of the PTF and a a predecessor of the ZTF. We show that many spatial and temporal features, common to all similar workflows [12], [13], are most useful for predicting the resource requirements. Therefore, the insight developed in this work is relevant to many astronomical data analysis projects.

Advancements of the optics and the image capturing systems in the synoptic surveys have dramatically increased the volume of data captured and created an unprecedented demand for analyzing these images. For example, the iPTF project uses a large field camera with 11 active 2048x4096 CCDs that capture 7.8 square degrees on the sky every two minutes producing 70GB per night [14]. The ZTF uses the same telescope as iPTF, but has an imager with a wider field-of-view and shorter exposure times yielding an order of magnitude more data in their search for young supernovae and fast-evolving transients. Due to the computing resources required, the data processing tasks are often completed in a computer center. In the case of iPTF, the images are transferred to the National Energy Research Scientific Computing Center (NERSC) and the transient discovery workflow is executed on the Cori supercomputer [13]. The iPTF workflow on Cori consists of three main parts: I/O, image subtraction (computation), and database matching (database queries). The execution time for these tasks within the analysis workflow varies with a number of characteristics of the image and system states.

In this paper, we propose a machine learning prediction strategy for predicting the performance of the iPTF workflow workload. Our conclusions are not limited to the iPTF workflow. Instead, they are readily useable for the ZTF workflow since the transient discovery tasks, and image attributes of iPTF and ZTF are similar [12]. In addition, we have taken care to build the prediction models based on features in image

978-1-7281-6095-5/20/$31.00 ©2020 IEEE
DOI 10.1109/CCGrid49817.2020.00-31

analysis workflows. In particular, we design several features and select two prediction targets that indicate the workload of the workflow pipeline. The features and prediction targets allow us to predict workflow performance in real-time. The critical contribution of this work is a Bayesian network that represents the conditional correlation among the features and prediction targets. The Bayesian network allows us to filter out redundant features, which in turn makes the model more actionable and avoids overfitting problems. We use experimental results to support the correctness of the Bayesian network. The testing data performance of our prediction models trained with the selected features achieves strong correlations between prediction and actual values and NMSEs as low as 0.1.

The rest of this paper is arranged as the following. In Section II, we describe the related work for transient discovery. Later, we present our preprocessing, feature selection, and model design methods in Section III. Finally, we present a case study of iPTF workflow data with our proposed approaches in Section IV.

II. Background

Gravitational wave (GW) detections, neutrino detections, and electromagnetic spectrum observations can reveal the death throes of stars [7], [8], [15]. However, the detailed mechanism of such stellar explosions is still an open question. Thus, many of the recent astronomical observations are focused on gathering data about transients to study the chaotic moments after stars explode [16]–[25].

To study these transients, one set of telescopes perform synoptic surveys by repeatedly looking at a large fraction of sky for any changes, while another set of telescopes, typically, many larger ones, will be directed to make more detailed observations about the newly discovered transients. In this process, the images gathered by the synoptic surveys need to be processed quickly so that the most promising candidates can be identified in near real-time for follow-on observations. Since there are only a limited number of large telescopes that could complete the follow-up observations, the workflow for processing the synoptic survey images must identify the most interesting transients accurately and reliably [12]. Additionally, the astronomers are increasingly focusing on short-lived transients, such as gamma-ray bursts or GW counterparts that might last only hours, which requires image processing workflows to be completed in minutes. Thus, such data processing workflows are often quite complex and require a considerable amount of computing resources, which is why iPTF uses Cori at NERSC.

The iPTF workflow pipeline processes a high volume of images in parallel. It consists of I/O operations, computation for filtering out relevant objects in images with machine learning techniques, and database operations. Scheduling jobs for such a workflow on the supercomputer is challenging since the execution time of different stages depends on the input data properties. A comprehensive study of workloads on supercomputers has shown that large-scale applications on supercomputers have a similar workload within a short period [26]. However, they conclude that the workload of a pipeline is likely to vary across a long-time period. Merzky et al. argue for the same conclusion in their RADICAL-Pilot paper [27]. Existing methods for scheduling jobs on supercomputers need an estimation of job workloads. In the case of iPTF, we use the number of objects to be extracted from an image as an estimation of workflow. Armstrong et al. [28] study the problem of scheduling multiple tasks with different workloads. Tang et al. propose methods for scheduling supercomputer jobs with conflict goals: fairness, wait time, and system utilization [29] [30]. However, predicting workload is a challenging task. Users cannot accurately predict the workload of their programs, so scheduling algorithms may be given wrong workload information. A follow up of Tang's work points out that the workload estimation is usually inaccurate on supercomputers [31]. They propose a method for adjusting the expected execution time of user jobs for better performance. Nevertheless, predicting workload based on the known software task and data can give more accurate estimation because domain knowledge is taken into account. In this paper, we take advantage of the domain-specific knowledge of iPTF.

The estimation of processing time based on workload is a challenging task since the execution time depends on both hardware conditions and software input data. The hardware condition is often unknown to users of supercomputers. However, we can infer it based on the execution time of recent jobs in the workflow pipeline. The software input data we are dealing with are iPTF images. Based on the available features of the images, we can predict the processing time of the bottleneck stage. We define the features and prediction model in the next section.

III. Design

The iPTF workflow pipeline has 40 checkpoints, labeled from 0 to 39, see an illustration in Fig. 5. We name the task between checkpoint i and $i+1$ to be processing stage i, and, therefore the execution time of stage i is the time difference between the time reported at checkpoint i and that at checkpoint $i+1$. Stage 0 of this workflow is to transfer the image captured by iPTF telescope to NERSC. The remaining stages of this workflow are executed on Cori [12], [13].

A. Removing I/O anomalies

We discover that the processing time of stage 5, 10, 14, and 19 contain anomalies. These are all I/O stages of the pipeline. Their mean processing time is higher than their 90% quantile. Moreover, the pair-wise Pearson correlation coefficients are all greater than 0.9. Figure 1 (a) illustrates the pair-wise correlation coefficients of raw dataset. Spearman correlation coefficients, on the other hand, are not showing such a strong correlation in Figure 1 (c). Thus, we conclude that the strong linear correlation of these I/O stages results from a few large anomalous numerical values. The anomaly values of the I/O stages are up to hours, which is much larger than the typical I/O processing time of the program (less than 2 minutes).

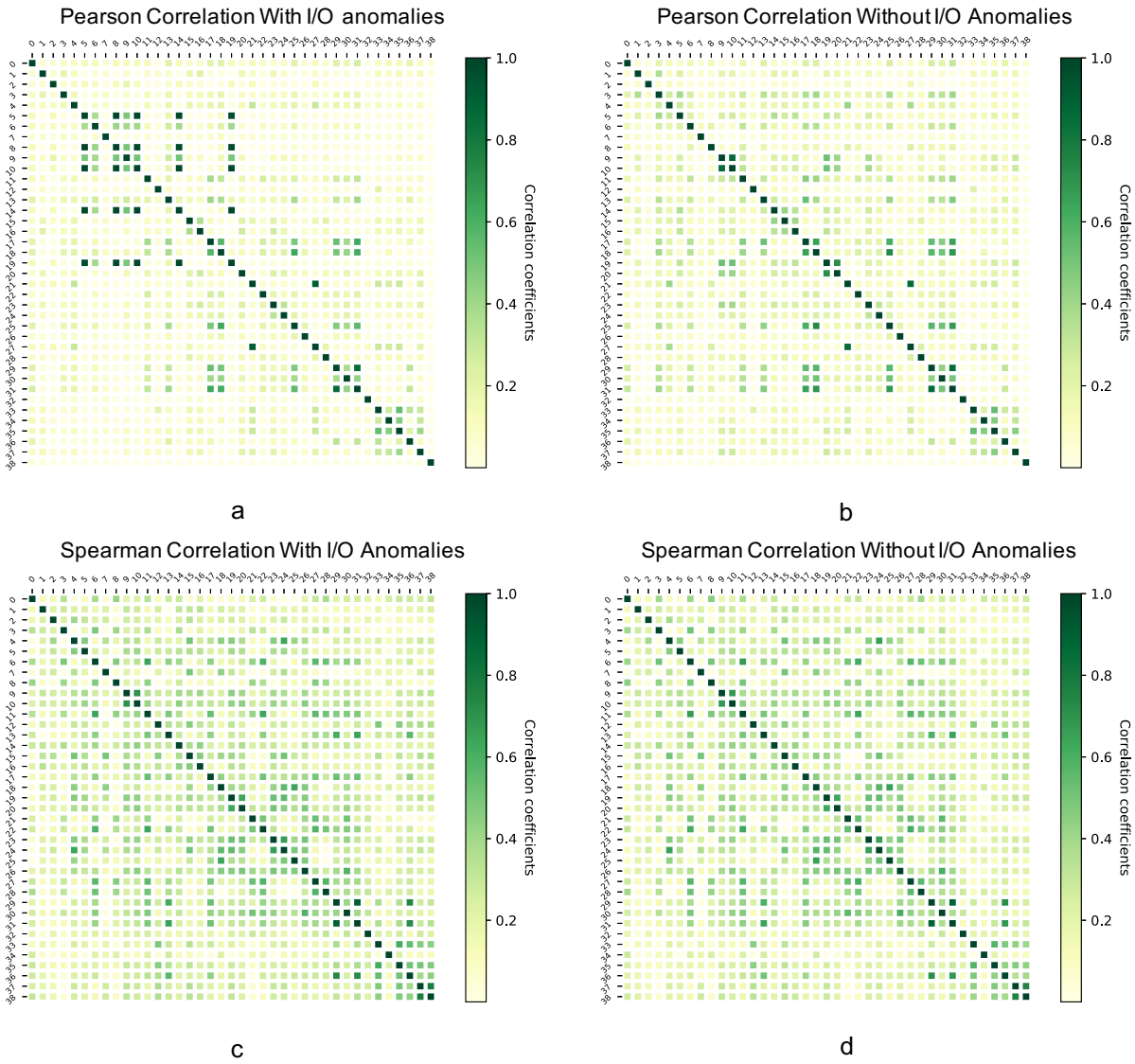

Fig. 1. (a) Pearson correlation coefficients for pair-wise processing time of stages before removing I/O anomalies. (b) Pearson correlation coefficients for pair-wise processing time of stages after removing I/O anomalies. (c) Spearman correlation coefficients for pair-wise processing time of stages before removing I/O anomalies. (d) Spearman correlation coefficients for pair-wise processing time of stages after removing I/O anomalies.

Therefore, we use the means of the I/O stage processing time as breakpoints. After we split the original dataset based on the mean of these four stages, approximately 10% of the dataset are classified as anomalous. The anomalous part shows a pair-wise linear correlation among the I/O stages close to 1, while the rest 90% of the regular dataset does not show a moderate pair-wise correlation. Figures 1 (b) and (d) illustrates the pair-wise correlations between different stages after removing I/O anomalies. This observation implies that the long execution time of these I/O stages occur together, from which it is reasonable to conclude that the I/O system is undergoing a severe degradation when these images passed through the pipeline. Consequently, we can confidently filter out these anomalies. These anomalies have timestamps in 2015, which was the first year that Cori was in service and are likely the product of various system instabilities. Our focus should be on the regular part of the dataset since such anomalies are not expected moving forward.

B. Prediction Target

Figure 2 illustrates the stage processing time distribution of all stages from 1 to 38. Stage 36 (between checkpoints 36 and 37) is the dominant stage. This stage is a database-intensive operation for associating candidates with local galaxies, a complex spatial query in postgres using q3c [32]. It is a critical piece of information in prioritizing follow-up observations. Predicting the processing time of this stage allows the schedule

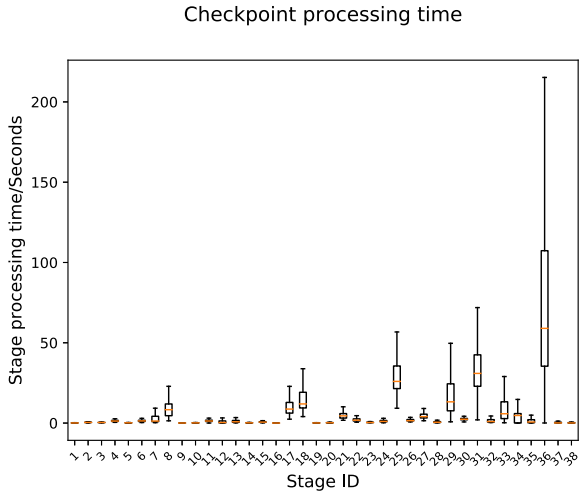

Fig. 2. Stage processing time of all stages after removing all I/O anomalies.

TABLE I
THIS TABLE SHOWS THE EARLIEST CHECKPOINT THAT AN IMAGE ATTRIBUTE IS AVAILABLE.

Feature Name	Checkpoint
Right Ascension	2
Declination	2
UTC time of start of observation	2
Julian Date for above	2
Airmass	2
PTF ID	2
Chip ID	2
Filter	2
Seeing	14
Limiting magnitude	14
# of objects extracted	33
# of objects saved	33
Ecliptic Longitude	2
Ecliptic Lattitude	2
Galactic Longitude	2
Galactic Lattitude	2
Extinction on sky	2

of the number of compute nodes ahead of time. For example, if the predicted time is relatively large, a larger number of compute nodes can be reserved. The rest of the stages with relatively long processing time are 25, 29, and 31. Stage 25 is an image subtraction. Stages 29-31 are parts of the real-bogus (RB) machine learning classifier. The RB classifier removes the false-positive detections caused by nonlinearity of the detectors, astrometric misalignment, imperfect convolution kernels, Poisson noise of bright objects, cosmic rays, and many other factors. These three stages are computationally intensive operations.

There are two prediction targets for the workload of an image: One is the dominant stage 36 processing time, and the other is the number of celestial objects extracted. Stage 36 is the most time-consuming step of the entire pipeline. Dominant stage processing time is a direct measurement of workload. However, it is subjected to the variance of the supercomputer's performance. The number of objects extracted determines the workload of specific operations. However, it is not available until the start of database stages, which is the checkpoint 33. Many stages, such as stage 25 and 27, which runs hotpants for image subtraction, have workloads dependent on the number of objects extracted. Moreover, the workloads of database stages are also proportional to the number of objects extracted. Thus, predicting the number of objects is equivalent to predicting the workload of multiple stages, which allows proper resource scheduling. Our goal is to predict the number of the object extracted at checkpoint 1, which is the start of the pipeline on Cori.

The proposed prediction targets are weakly correlated. Figure 3 (a) and (b) depict the boxplots of the processing time of stage 36 and the number of objects extracted in a 24-hour interval. From 9:00 AM to 7:00 PM, no iPTF image is taken because iPTF only operates at night. Different hours have a similar variance of these two prediction targets. Furthermore, we can observe that means of the number of objects extracted and the processing time of checkpoint 36 have correlations over these periods. Nevertheless, these two prediction targets are not interchangeable because they only have 0.62 Pearson and 0.58 Spearman correlations at image granularity.

C. Feature selection

To predict either stage 36 processing time or objects extracted, we define attributes for making such prediction. There are two constraints for the prediction model to make a useful inference. Firstly, the input attributes of the prediction model should be ready before the output of prediction occur. For example, the moment when a model makes a prediction for stage 36, processing time must happen before the timestamp of checkpoint 36. Otherwise, the prediction result is useless for scheduling a computing resource since the program for running stage 36 is already started. Another constraint is that the prediction model should be light-weighted. Since the goal of this research is to schedule computational resources smartly, a heavy machine learning model that consumes larger resources is unacceptable.

Stage 36 is nearly at the end of the pipeline (39 stages in total), so it is possible to apply timings of stages from 0 to 35 before checkpoint 36 as prediction attributes. We denote this class of features as type 1. However, models for predicting the number of objects extracted does not have the access to type 1 features since this prediction is expected at checkpoint 1 in order to have a raw estimate of the entire workload. In addition, some attributes of the image are known in advance of stage 0, so we can directly apply them as model features for both of the prediction targets. This class of features is denoted as type 2. Table I illustrates all the image attributes and the checkpoints that they are available on the pipeline. For the image attributes, we denote the number of objects extracted as type 2a and the rest as type 2b. Type 2b does not include seeing and the number of objects saved for predicting the number of objects extracted by our previous constraints.

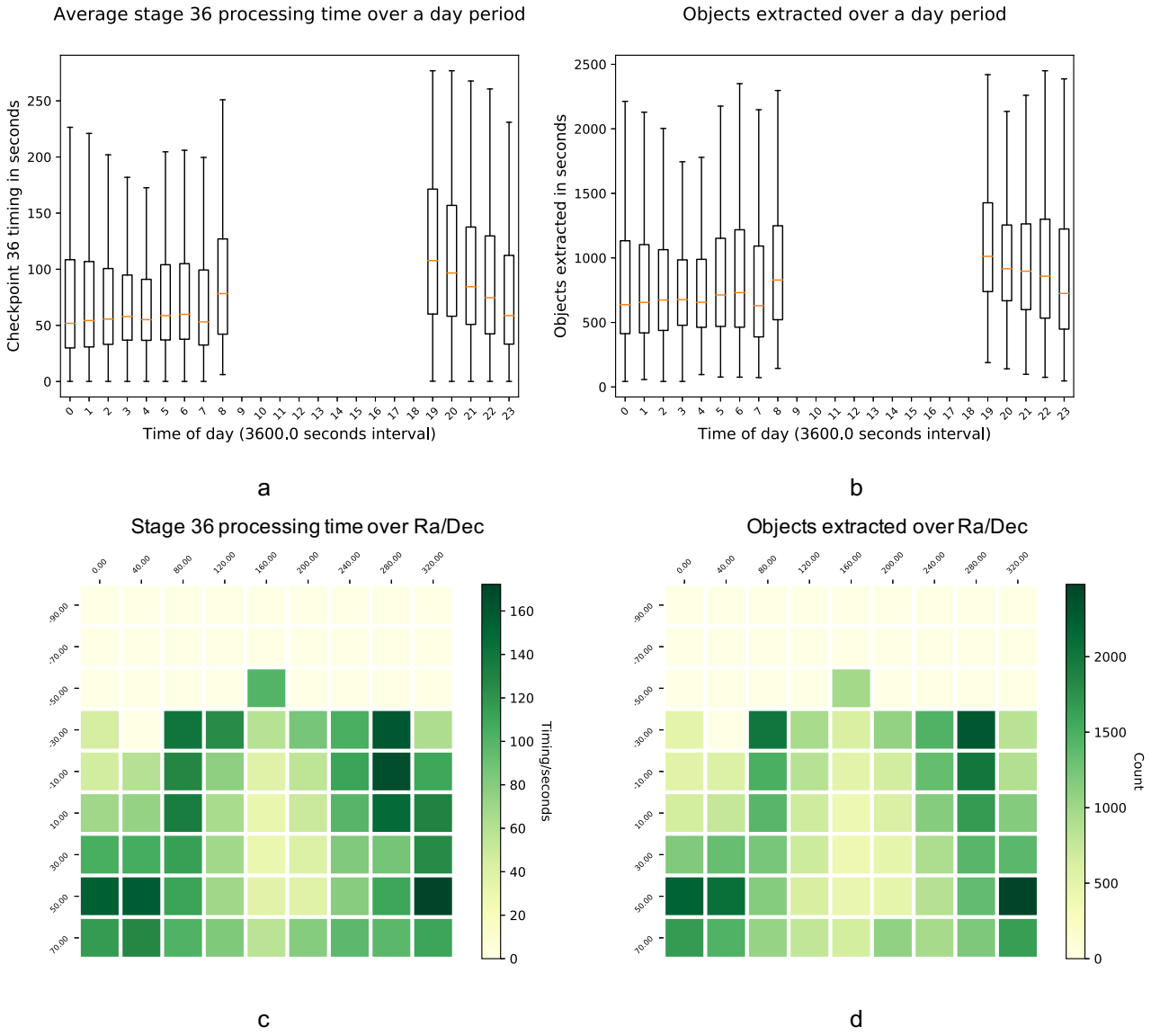

Fig. 3. (a) Average stage 36 processing time in 24 hours period. (b) Average objects extracted in 24 hours period. (c) Average stage 36 processing time in spatial on right ascension (horizontal) and declination (vertical) scale. (d) Average objects extracted in spatial on right ascension (horizontal) and declination (vertical) scale. Note the processing time roughly correlates with the position on the sky following the over density of stars in our own galaxy.

In addition to attributes limited to a single image, we can infer the workload of the current images from the workload of other images that have been processed in the recent past. From a temporal point of view, we expect the nearest k number of images that have been processed by the pipeline should inform the current workload. We notice that the cameras of iPTF rotate in the sky in a continuous fashion. However, on some rare occasions, they jump from one region to another. Thus, the nearest temporal neighbors may contain some spatial gaps. We plot the average processing time of stage 36 and the average objects extracted in Figure 3 (c) and (d) on 9×9 on right ascension and declination grids. The heap maps have continuous colors in most of the regions, which indicate the similarity of the means of these attributes within a spatial neighborhood. Therefore, we decided to construct spatiotemporal features. For example, the processing time of stage 36 of images taken from nearby coordinates within a short period can inform the prediction of stage 36 processing time of the current image. The temporal proximity implies a similar condition of a computing system, and the spatial proximity implies a similar number of objects and workloads. This type of feature is denoted as type 3. Due to the first constraint, all images in the most adjacent neighborhood must have reached checkpoint 37 before the start of stage 36 of the current image for predicting the stage 36 processing time. For the case of predicting the number of objects, the nearest

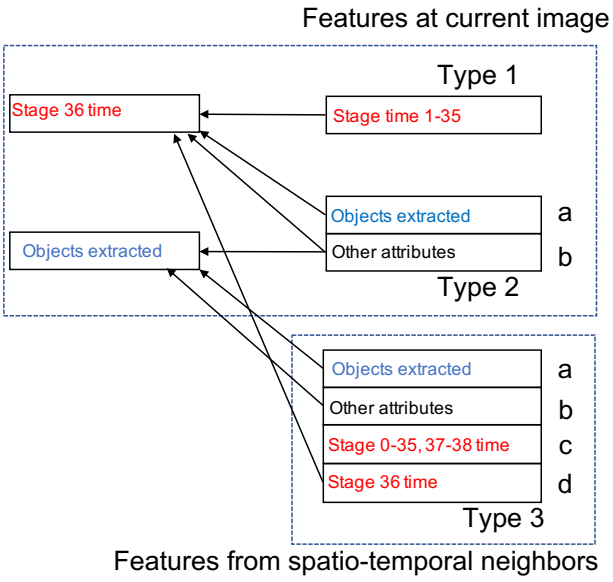

Fig. 4. iPTF prediction model feature summary. Type 1 features are the stage processing time of current images prior to checkpoint 36. Type 2 features are the attributes of current image. Category 2a is the number of objects extracted and category 2b contains other attributes of the images shown in Table I. Type 3 features are the features provided by images in the spatiotemporal nearest neighborhood.

spatiotemporal neighborhood must have passed checkpoint 33, according to Table I.

By the first constraint, all attributes selected from adjacent images must be ready at the checkpoint 36 (for stage 36 processing time) and checkpoint 1 (for the number of objects extracted) of the current image. Figure 5 illustrates these two predictions in the workflow.

Figure 4 illustrates the features selected for predicting the number of objects and stage 36 processing time. Applying all the available features for making predictions is not an optimal strategy since some features can be independent of the prediction target in the presence of other features. In these scenarios, overfitting problems can occur. Although a model yields low error after training, the testing performance is reduced, because random errors of the features and the prediction target that happen in the training data only are correlated. However, the correlations of errors may only exist in the training data. Therefore, the model produces false inference on testing features.

We propose a Bayesian network of features and prediction targets in Figure 6. A Bayesian network can show the conditional independence of variables. Features that are directly linked to a prediction target should be used by prediction models. The network is created by iterating through combinations of features and compare their testing errors of trained models. Although it is possible to construct a Bayesian network from a greedy algorithm, such as [33], we rely on domain knowledge of iPTF for the creation of the Bays network to avoid overfitting. For instance, the part in the dashed rectangle denotes the Spatiotemporal neighborhood of the current image. The rest of the figures are features related to the current image. These two parts should have identical internal topology since they are both image entities. Therefore, the search range of Bays network topology is reduced significantly with the incorporation of domain constraints. In the Section II, we justify the structure of the Bayesian network by analyzing the prediction accuracy on testing data with different combinations of features.

We keep the directions of the arrows that satisfy the constraints of prediction mentioned in Section III-C. In other words, if A points an arrow to B, the value of A is available before B. Thus, this network is a trimmed version designed for feature selections of our objectives. Our prediction targets are labeled in red. Features that are directly connected to the red targets are those that can build a model with the best prediction results. If there is a long path between a feature and the target, the correlation between the feature and the target is expected to be weak, especially when other features are present on its path to the target.

D. Prediction model

We apply a multi-layer perceptron, a typical neural network architecture, also denoted as a fully connected network, for predicting target attributes from input attributes.

A fully connected network consists of layers. The first layer is the input layer, and the last layer is the output layer. There are some hidden layers that in-between the input and the output layer. Let A be a $m \times n$ matrix describing the connections between layer l_1 of size n and layer l_2 of size m for a fully connected network. The j^{th} row and i^{th} column of A, $a_{j,i}$, is the weight from the j^{th} neuron of l_1 to the i^{th} neuron of l_2. Let b be a bias vector of dimension n. The i^{th} element of b is the bias for the i^{th} neuron of l_2. The output feature vector v from l_1 has size n. Then $f(Av + b)$ is the output vector for layer l_2. Our prediction network has one input layer, three hidden layers, and one output layer. There are 1024 neurons in the first hidden layer and 256 neurons in the second and third hidden layers. The final layer has one neuron that outputs the regression result. The activation function is Relu $f(x) = \max(0, x)$.

An advantage of this neural network is that it is capable of simulating any functions [34]. For the problem size we are dealing with, the proposed network size is sufficient. Moreover, this parameter settings of the multi-layer perceptron satisfy the requirement of light-weighted machine learning model, since a prediction for one feature can finish within a few seconds on a regular desktop computer.

IV. Experimental Results

We conduct experiments on a single Linux machine installed with a GTX 1080 and CUDA version 10.1.

For a real-world case study, we split the iPTF pipeline data[1] into training data and testing data. The training data

[1]https://portal.nersc.gov/project/astro250/xswap/iptf/

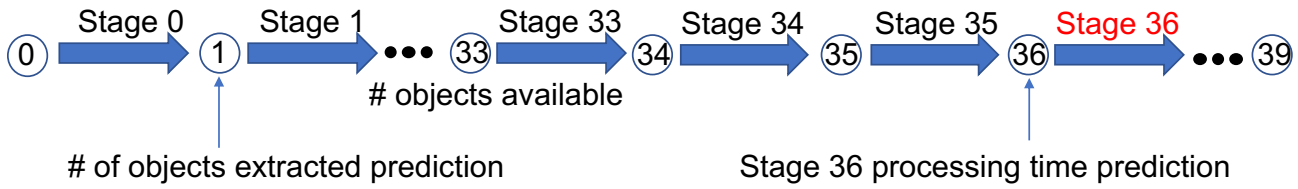

Fig. 5. iPTF online prediction illustration. The circles indicate checkpoints of the workflow pipeline. The stages are denoted as arrows. The prediction of the number of objects is at checkpoint 1. The prediction of the stage 36 processing time is at checkpoint 36.

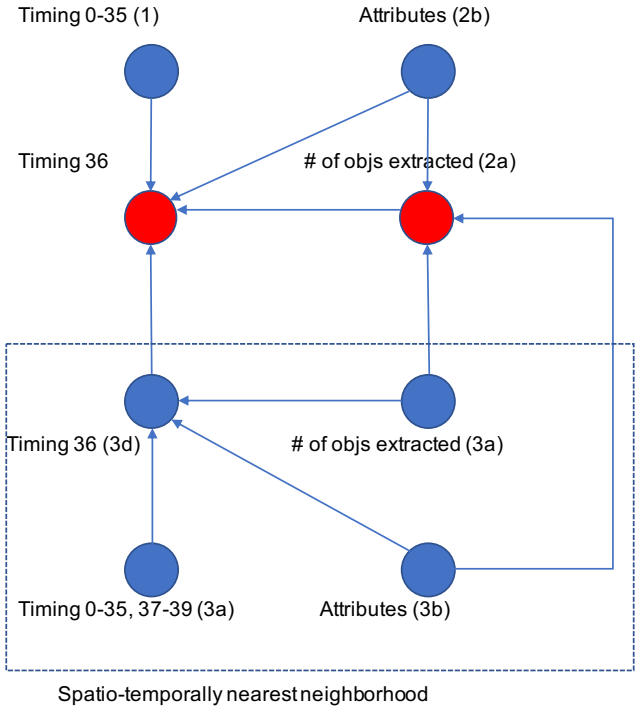

Fig. 6. iPTF prediction model feature Bayesian network. Features that are directly connected to the red prediction targets are the one that should be used by the prediction models.

contains 90% of the data with lower timestamps, and the testing data includes the other 10%. With such a splitting strategy, predictions on the testing data are equivalent to real-time simulations. This splitting method is a simulation of a production run since the goal is to train historical data for predicting future results. In addition, we also present 10-fold cross-validation results for evaluation.

We train the neural network discussed in Section III-D with Stochastic Gradient Descent (SGD) for 200 epochs with mini-batch size 256. The learning rate is from 0.88 to 1, with a 0.1 momentum factor. This training strategy guarantees the convergence of all our experiments. We set the drop-out ratio per layer to be 0.33 to avoid the overfitting of the model [35]. Training with an adaptive learning rate such as ADAM [36] does not converge to a reasonable regression accuracy the same way as SGD. Besides, the final losses of the models are unstable with different initial parameters. The models trained with SGD, on the other hand, are stable with better accuracy results.

More complicated model architectures may yield better training results after convergence. However, the main goal of this paper is to evaluate the proposed iPTF workflow features. Thus, we control the neural network model and training methods, treating the model as a black-box for regression, for comparing different choices of input features.

Root mean square error (RMSE), Normalized mean square error (NMSE), and correlation coefficients (Pearson and Spearman) are the evaluation metrics in this section. MSE evaluates the regression error with formula $\frac{1}{N}\sum_{i=1}^{N}\left(\hat{f}\left(x_i\right)-y_i\right)^2$ for prediction model $\hat{f}$, input feature x, and prediction output y. RMSE is the square root of MSE. NMSE scales down the units of RMSE with formula $\frac{1}{xy}$MSE. A good model should have small RMSE and NMSE. Correlation coefficients measure if the predictions have the same trend as the actual data in the parity graph. The metrics are between 0 and 1. A good model has large correlation coefficients. Pearson correlation takes the magnitude of difference into account. The Spearman correlation examines if the trend of two variables is the same or not. Although our training objective is to reduce regression error, the problem can be reduced to a classification problem. For example, we can classify the workload to be discrete results, such as small, medium, and large, for different ranges of the outputs. In this scenario, the absolute errors are less important due to the change of the error function. If the correlations between prediction and real measurements are large, the classification result should be mostly correct, since the prediction is likely to lie in the right region.

A. Baselines

We can set up the baselines that evaluate the performance of our model via RMSE. Testing RMSE should fall between a lower bound and an upper bound.

An upper bound of the prediction targets for RMSE is the standard deviation. If a prediction model always predicts the mean of the prediction target in the dataset, its RMSE is equivalent to the standard deviation. The estimated standard deviations from the populations are 60.4 for the stage 36 processing time and 653 for the number of objects extracted.

A lower bound of the prediction targets for RMSE is the estimation of the prediction targets' standard deviation within

a batch. iPTF cameras take images in batches. Images within a batch are processed in parallel with the same checkpoint 0 start time. These images have very similar spatiotemporal attributes. A regression model based on our proposed features without overfitting should give the same results. However, there are still some non-negligible variances of workload within a single batch. These variances are unpredictable by the information given in the features for regression since the attributes are identical. Thus, the variance within batches can serve as a lower bound of prediction error. We estimate the average standard deviation within a batch for the stage 36 processing time as 20.4 and the number of objects extracted as 210 from all data.

B. Predicting stage 36 processing time

Discussed in Section III-C, several different types of features can be applied to predict stage 36 processing time, as summarized in Figure 4. The input features have three categories. The first type is historical processing time from stage 0 to 35. The second category is the image attributes, including spatial coordinates. The final type (type 3a) is the stage 36 processing time of spatiotemporally nearby images in the pipeline.

The number of the nearest neighbors is a tunable parameter. We use 1 for simplicity. In this section, we demonstrate the individual effectiveness of these three types of features.

Rows 1 to 10 of Table II illustrate the results of predicting stage 36 processing time with a different combination of features for ten-fold cross-validation. CV refers to the average of 10-fold cross-validation results, and LF refers to the last fold (as testing data) of the cross-validation results. The values in the brackets are prediction evaluations on testing data. The values that are not in the brackets are prediction evaluations on training data.

From the first three rows, we can observe relatively large mean square errors and low correlation coefficients. Thus, using any of the three features alone, the regression models do not have reasonably good results on testing data.

The fourth to the sixth rows contain results with pair-wise combinations of three features. The historical processing time from stage 0 to 35 of the current image (type 1), combined with the image attributes (type 2), gives better testing results than replacing type 2 features with spatiotemporally nearby image processing time (type 3d). The combination of type 2 and 3d features yields similar results as the combination of type 1 and 2 features. Thus, we observe that image attributes can improve the overall performance significantly when added to either type 1 and 3d features. Type 1 and 3d images combined (row 6), on the other hand, demonstrate improvement compared with models trained with either of them. This observation is expected because both features are timing features. They are likely correlated with each other. Nevertheless, joining them together can moderately improve testing accuracy compared with the results predicted by each of them separately. Finally, if all these three features are combined, the overall regression performance on testing data is the best (row 7). Therefore, the direct connections to stage 36 processing time in Figure 6 is verified.

Rows 9 and 10 of figure Table II illustrate the reasons why the attributes (types 3a, 3b) and stage processing time 0-35 of the spatiotemporally nearest neighborhood (type 3c) are not useful for predicting the stage processing time of current image given the stage 36 processing of the neighborhood (type 3d). We can observe that the testing metrics in rows 9 and 10 are worse than those of row 8. Moreover, if we compare the results of rows 8 and 9, it is clear that the correlation between the prediction feature is stronger to the nearest neighbors' stage 36 processing time than to the rest of type 3 attributes. Consequently, type 3d should be selected as a feature for prediction, instead of the rest of the features.

C. Predicting the number of objects extracted

Details of selecting the features of the number of objects extracted are presented in Figure 4. Prediction of the number of objects extracted takes place at the beginning of the pipeline. Thus, the type 1 features adopted for predicting stage 36 processing time cannot be used in this case, since the checkpoints for computing stage processing time have not reached. All image attributes excluding the seeing, limiting magnitude, objects extracted and saved, on the other hand, are available at the start of the pipeline. Furthermore, attributes from spatiotemporally adjacent images are available at checkpoint 2 if we carefully filter the neighborhood N with constraint $\{n.\text{checkpoint}_{39} < x.\text{checkpoint}_1\} \forall n \in N$ for current image x. Similar to predicting stage 36 processing time, the attribute to be predicted (the number of objects extracted) is selected from the neighborhood. Furthermore, we also employ the stage processing time and other image attributes of its spatiotemporally nearest neighborhood. These features are type 3 features. From the Bayesian network in Figure 6, we expect types 2b, 3a, and 3b to have a strong correlation with the prediction objective.

Rows 11-20 of Table II illustrate the results of predicting the number of objects extracted by our real-time prediction algorithm on testing data. Rows 11-13 illustrate the prediction results of using the attributes of the current image (type 2b), the number of objects extracted (type 3a), and other attributes (type 3b) of the spatiotemporally nearest neighborhood of current image independently. Row 14 demonstrates the prediction accuracy using the stage processing time of spatiotemporally nearest neighborhood (type 3c,3d) that are not directly connected to the prediction target in Figure 6. Results in row 14 are worse than rows 11, 12, and 13, which is expected from the Bayesian network since the correlations of features in row 14 to the prediction target is weaker. Rows 15-17 demonstrate the results of using different combinations of pair-wise features in rows 11-13. The results of pair-wise combinations of features are better than results using each independent feature. Row 18 illustrates the prediction results of features that are directly connected to the prediction targets in Figure 6. We can observe further accuracy improvements in both testing and training data compared with rows 15-17.

TABLE II

ACCURACY RESULTS OF PREDICTION MODELS. CV REFERS TO THE AVERAGE OF 10-FOLD CROSS-VALIDATION RESULTS AND LF REFERS TO THE LAST FOLD (AS TESTING DATA) OF THE CROSS-VALIDATION RESULTS. THE VALUES IN THE BRACKETS ARE PREDICTION EVALUATIONS ON TESTING DATA. THE VALUES THAT ARE NOT IN THE BRACKETS ARE PREDICTION EVALUATIONS ON TRAINING DATA. ROWS 1 TO 10 ARE RESULTS FOR PREDICTING STAGE 36 PROCESSING TIME. THE RANGE OF RMSES SHOULD BE BETWEEN 20.4 AND 60.4. ROWS 11 TO 20 ARE RESULTS FOR PREDICTING THE NUMBER OF OBJECTS EXTRACTED. THE RANGE OF RMSES SHOULD BE BETWEEN 210 AND 653.

Row ID	Features	CV RMSE	CV NMSE	CV Pearson	CV Spearman	LF RMSE	LF NMSE	LF Pearson	LF Spearman
1	1	38.7 (45.1)	0.212 (0.303)	0.79 (0.66)	0.81 (0.68)	36.9 (51.0)	0.207 (0.230)	0.80 (0.65)	0.81 (0.69)
2	2	29.2 (31.8)	0.135 (0.160)	0.88 (0.82)	0.93 (0.87)	28.1 (44.0)	0.129 (0.216)	0.88 (0.73)	0.93 (0.80)
3	3d	43.4 (42.4)	0.287 (0.280)	0.70 (0.63)	0.75 (0.67)	41.8 (55.1)	0.281 (0.317)	0.72 (0.50)	0.76 (0.56)
4	3a,3b,3c	37.4 (41.7)	0.221 (0.278)	0.79 (0.65)	0.82 (0.67)	36.2 (54.3)	0.223 (0.331)	0.80 (0.53)	0.83 (0.59)
5	1,2	24.0 (28.2)	0.088 (0.125)	0.92 (0.87)	0.95 (0.91)	22.6 (38.6)	0.084 (0.158)	0.93 (0.79)	0.96 (0.86)
6	1,3d	34.1 (36.8)	0.179 (0.213)	0.83 (0.75)	0.86 (0.77)	32.6 (47.5)	0.170 (0.201)	0.84 (0.71)	0.86 (0.75)
7	2,3d	28.0 (30.2)	0.124 (0.145)	0.89 (0.84)	0.94 (0.89)	26.7 (42.1)	0.118 (0.200)	0.90 (0.76)	0.94 (0.83)
8	**1,2,3d**	**23.6 (27.0)**	**0.085 (0.114)**	**0.93 (0.88)**	**0.96 (0.92)**	**22.5 (36.0)**	**0.084 (0.137)**	**0.93 (0.82)**	**0.96 (0.88)**
9	1,2,3a,3b,3c	23.9 (28.5)	0.087 (0.127)	0.92 (0.87)	0.95 (0.90)	22.6 (38.9)	0.084 (0.160)	0.93 (0.79)	0.96 (0.86)
10	1,2,3d,3c,3b,3a	23.7 (27.8)	0.086 (0.122)	0.93 (0.88)	0.96 (0.92)	22.1 (36.3)	0.080 (0.139)	0.93 (0.82)	0.96 (0.88)
11	2b	309.3 (352.5)	0.122 (0.168)	0.83 (0.72)	0.83 (0.68)	311.9 (319.7)	0.125 (0.113)	0.83 (0.75)	0.83 (0.75)
12	3a	321.9 (320.0)	0.135 (0.138)	0.81 (0.76)	0.82 (0.76)	319.2 (335.8)	0.134 (0.128)	0.82 (0.71)	0.83 (0.72)
13	3b	270.7 (303.2)	0.098 (0.127)	0.87 (0.78)	0.87 (0.77)	267.3 (318.5)	0.098 (0.124)	0.87 (0.75)	0.87 (0.75)
14	3c,3d	351.2 (370.5)	0.176 (0.200)	0.77 (0.67)	0.78 (0.65)	344.8 (391.2)	0.166 (0.164)	0.78 (0.63)	0.78 (0.63)
15	2b,3a	262.9 (288.9)	0.090 (0.113)	0.88 (0.81)	0.88 (0.79)	260.3 (287.6)	0.091 (0.095)	0.88 (0.80)	0.88 (0.80)
16	2b,3b	225.6 (265.1)	0.067 (0.097)	0.91 (0.84)	0.91 (0.82)	224.6 (263.4)	0.070 (0.084)	0.91 (0.83)	0.91 (0.83)
17	3a,3b	270.9 (301.8)	0.098 (0.126)	0.87 (0.78)	0.87 (0.77)	267.9 (316.8)	0.098 (0.123)	0.87 (0.75)	0.87 (0.75)
18	**2b,3a,3b**	**226.3 (266.9)**	**0.067 (0.098)**	**0.91 (0.84)**	**0.91 (0.82)**	**224.5 (265.0)**	**0.068 (0.083)**	**0.91 (0.83)**	**0.91 (0.83)**
19	2b,3a,3c,3d	250.5 (284.4)	0.084 (0.114)	0.89 (0.81)	0.89 (0.79)	251.2 (284.7)	0.087 (0.098)	0.89 (0.79)	0.89 (0.79)
20	2b,3a,3b,3c,3d	229.6 (269.2)	0.071 (0.102)	0.91 (0.83)	0.90 (0.82)	226.7 (269.8)	0.070 (0.089)	0.91 (0.82)	0.90 (0.82)

Rows 19 and 20 of Table II prove the point that the stage processing time of spatiotemporally nearest neighborhood of the current image is independent of the number of objects extracted of current image given types 3b, 3a, and 3b. Furthermore, type 3b has a stronger correlation with the prediction target compared with types 3c and 3d. Therefore, type 3b should be directly connected to the prediction target, as shown in the Bays network in Figure 6.

V. CONCLUSION

We study the features for predicting the workload of the iPTF workflow pipeline. I/O anomalies of the iPTF workflow pipeline are filtered out. The proposed features and prediction targets of workloads are summarized into a Bayesian network that shows the conditional dependency of features. The selected features from the Bayesian network achieve prediction accuracy close to the lower bound defined by the randomness within data. Our spatiotemporal feature construction approach is not limited to the iPTF survey.

In the future, we plan to apply our predictions of workloads to schedule resources for the ZTF survey. A customized scheduling approach for the ZTF workflow pipeline can further improve the throughput of the system.

ACKNOWLEDGMENT

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Scientific Discovery through Advanced Computing (SciDAC) program. This work was supported in part by the Office of Advanced Scientific Computing Research, Office of Science, of the U.S. Department of Energy, and used resources of the National Energy Research Scientific Computing Center, under Contract No. DE-AC02-05CH11231. This work is also supported in part by the DOE awards DE-SC0014330 and DE-SC0019358.

REFERENCES

[1] D. Chatterjee, P. E. Nugent, P. R. Brady, C. Cannella, D. L. Kaplan, and M. M. Kasliwal, "Toward rate estimation for transient surveys. i. assessing transient detectability and volume sensitivity for iPTF," *The Astrophysical Journal*, vol. 881, p. 128, aug 2019.

[2] M. Sako, B. Bassett, A. Becker, D. Cinabro, F. DeJongh, D. L. Depoy, B. Dilday, M. Doi, J. A. Frieman, P. M. Garnavich, *et al.*, "The sloan digital sky survey-ii supernova survey: search algorithm and follow-up observations," *The Astronomical Journal*, vol. 135, no. 1, p. 348, 2007.

[3] A. Drake, S. Djorgovski, A. Mahabal, E. Beshore, S. Larson, M. Graham, R. Williams, E. Christensen, M. Catelan, A. Boattini, *et al.*, "First results from the catalina real-time transient survey," *The Astrophysical Journal*, vol. 696, no. 1, p. 870, 2009.

[4] T. Shanks, N. Metcalfe, B. Chehade, J. Findlay, M. Irwin, E. Gonzalez-Solares, J. Lewis, A. K. Yoldas, R. Mann, M. Read, *et al.*, "The vlt survey telescope atlas," *Monthly Notices of the Royal Astronomical Society*, vol. 451, no. 4, pp. 4238–4252, 2015.

[5] N. M. Law, S. R. Kulkarni, R. G. Dekany, E. O. Ofek, R. M. Quimby, P. E. Nugent, J. Surace, C. C. Grillmair, J. S. Bloom, M. M. Kasliwal, *et al.*, "The palomar transient factory: system overview, performance, and first results," *Publications of the Astronomical Society of the Pacific*, vol. 121, no. 886, p. 1395, 2009.

[6] E. C. Bellm, S. R. Kulkarni, M. J. Graham, R. Dekany, R. M. Smith, R. Riddle, F. J. Masci, G. Helou, T. A. Prince, S. M. Adams, *et al.*, "The zwicky transient facility: System overview, performance, and first results," *Publications of the Astronomical Society of the Pacific*, vol. 131, no. 995, p. 018002, 2018.

[7] A. Goobar, R. Amanullah, S. R. Kulkarni, P. E. Nugent, J. Johansson, C. Steidel, D. Law, E. Mörtsell, R. Quimby, N. Blagorodnova, A. Brandeker, Y. Cao, A. Cooray, R. Ferretti, C. Fremling, L. Hangard, M. Kasliwal, T. Kupfer, R. Lunnan, F. Masci, A. A. Miller, H. Nayyeri, J. D. Neill, E. O. Ofek, S. Papadogiannakis, T. Petrushevska, V. Ravi, J. Sollerman, M. Sullivan, F. Taddia, R. Walters, D. Wilson, L. Yan, and

O. Yaron, "iptf16geu: A multiply imaged, gravitationally lensed type ia supernova," *Science*, vol. 356, no. 6335, pp. 291–295, 2017.

[8] M. Kasliwal, S. Cenko, L. Singer, A. Corsi, Y. Cao, T. Barlow, V. Bhalerao, E. Bellm, D. Cook, G. Duggan, *et al.*, "iptf search for an optical counterpart to gravitational-wave transient gw150914," *The Astrophysical Journal Letters*, vol. 824, no. 2, p. L24, 2016.

[9] P. E. Nugent, M. Sullivan, S. B. Cenko, R. C. Thomas, D. Kasen, D. A. Howell, D. Bersier, J. S. Bloom, S. R. Kulkarni, M. T. Kand rashoff, A. V. Filippenko, J. M. Silverman, G. W. Marcy, A. W. Howard, H. T. Isaacson, K. Maguire, N. Suzuki, J. E. Tarlton, Y.-C. Pan, L. Bildsten, B. J. Fulton, J. T. Parrent, D. Sand, P. Podsiadlowski, F. B. Bianco, B. Dilday, M. L. Graham, J. Lyman, P. James, M. M. Kasliwal, N. M. Law, R. M. Quimby, I. M. Hook, E. S. Walker, P. Mazzali, E. Pian, E. O. Ofek, A. Gal-Yam, and D. Poznanski, "Supernova SN 2011fe from an exploding carbon-oxygen white dwarf star," *Nature*, vol. 480, pp. 344–347, Dec 2011.

[10] A. Gal-Yam et al., "Real-Time Detection and Rapid Multiwavelength Follow-Up Observations of a Highly Subluminous Type II-P Supernova from the Palomar Transient Factory Survey," *Astrophysical Journal*, vol. 736, no. 2, 2011.

[11] S. Kulkarni, "The intermediate palomar transient factory (iptf) begins," *The Astronomer's Telegram*, vol. 4807, 2013.

[12] J. Bloom, J. Richards, P. Nugent, R. Quimby, M. Kasliwal, D. Starr, D. Poznanski, E. Ofek, S. Cenko, N. Butler, *et al.*, "Automating discovery and classification of transients and variable stars in the synoptic survey era," *Publications of the Astronomical Society of the Pacific*, vol. 124, no. 921, p. 1175, 2012.

[13] Y. Cao, P. E. Nugent, and M. M. Kasliwal, "Intermediate Palomar Transient Factory: Realtime Image Subtraction Pipeline," *PASP*, vol. 128, p. 114502, Nov 2016.

[14] G. Rahmer, R. Smith, V. Velur, D. Hale, N. Law, K. Bui, H. Petrie, and R. Dekany, "The 12Kx8K CCD mosaic camera for the Palomar Transient Factory," in *Ground-based and Airborne Instrumentation for Astronomy II* (I. S. McLean and M. M. Casali, eds.), vol. 7014, pp. 1714 – 1725, International Society for Optics and Photonics, SPIE, 2008.

[15] D. Radice, V. Morozova, A. Burrows, D. Vartanyan, and H. Nagakura, "Characterizing the gravitational wave signal from core-collapse supernovae," *The Astrophysical Journal Letters*, vol. 876, no. 1, p. L9, 2019.

[16] A. Rubin, A. Gal-Yam, A. De Cia, A. Horesh, D. Khazov, E. O. Ofek, S. R. Kulkarni, I. Arcavi, I. Manulis, O. Yaron, *et al.*, "Type ii supernova energetics and comparison of light curves to shock-cooling models," *The Astrophysical Journal*, vol. 820, no. 1, p. 33, 2016.

[17] D. Khazov, O. Yaron, A. Gal-Yam, I. Manulis, A. Rubin, S. Kulkarni, I. Arcavi, M. Kasliwal, E. Ofek, Y. Cao, *et al.*, "Flash spectroscopy: emission lines from the ionized circumstellar material around¡ 10-day-old type ii supernovae," *The Astrophysical Journal*, vol. 818, no. 1, p. 3, 2016.

[18] Y. Cao, S. Kulkarni, D. A. Howell, A. Gal-Yam, M. M. Kasliwal, S. Valenti, J. Johansson, R. Amanullah, A. Goobar, J. Sollerman, *et al.*, "A strong ultraviolet pulse from a newborn type ia supernova," *Nature*, vol. 521, no. 7552, p. 328, 2015.

[19] E. Hsiao, C. Burns, C. Contreras, P. Höflich, D. Sand, G. H. Marion, M. Phillips, M. Stritzinger, S. González-Gaitán, R. Mason, *et al.*, "Strong near-infrared carbon in the type ia supernova iptf13ebh," *Astronomy & Astrophysics*, vol. 578, p. A9, 2015.

[20] S. B. Cenko, A. L. Urban, D. A. Perley, A. Horesh, A. Corsi, D. B. Fox, Y. Cao, M. M. Kasliwal, A. Lien, I. Arcavi, *et al.*, "iptf14yb: the first discovery of a gamma-ray burst afterglow independent of a high-energy trigger," *The Astrophysical Journal Letters*, vol. 803, no. 2, p. L24, 2015.

[21] E. Gorbikov, A. Gal-Yam, E. O. Ofek, P. M. Vreeswijk, P. E. Nugent, N. Chotard, S. R. Kulkarni, Y. Cao, A. De Cia, O. Yaron, *et al.*, "iptf13beo: the double-peaked light curve of a type ibn supernova discovered shortly after explosion," *Monthly Notices of the Royal Astronomical Society*, vol. 443, no. 1, pp. 671–677, 2014.

[22] A. Gal-Yam, I. Arcavi, E. Ofek, S. Ben-Ami, S. Cenko, M. Kasliwal, Y. Cao, O. Yaron, D. Tal, J. Silverman, *et al.*, "A wolf–rayet-like progenitor of sn 2013cu from spectral observations of a stellar wind," *Nature*, vol. 509, no. 7501, p. 471, 2014.

[23] C. Fremling, J. Sollerman, F. Taddia, M. Ergon, S. Valenti, I. Arcavi, S. Ben-Ami, Y. Cao, S. Cenko, A. Filippenko, *et al.*, "The rise and fall of the type ib supernova iptf13bvn-not a massive wolf-rayet star," *Astronomy & Astrophysics*, vol. 565, p. A114, 2014.

[24] A. Goobar, J. Johansson, R. Amanullah, Y. Cao, D. Perley, M. Kasliwal, R. Ferretti, P. Nugent, C. Harris, A. Gal-Yam, *et al.*, "The rise of sn 2014j in the nearby galaxy m82," *The Astrophysical Journal Letters*, vol. 784, no. 1, p. L12, 2014.

[25] Y. Cao, M. M. Kasliwal, I. Arcavi, A. Horesh, P. Hancock, S. Valenti, S. B. Cenko, S. Kulkarni, A. Gal-Yam, E. Gorbikov, *et al.*, "Discovery, progenitor and early evolution of a stripped envelope supernova iptf13bvn," *The Astrophysical Journal Letters*, vol. 775, no. 1, p. L7, 2013.

[26] H. You and H. Zhang, "Comprehensive workload analysis and modeling of a petascale supercomputer," in *Workshop on Job Scheduling Strategies for Parallel Processing*, pp. 253–271, Springer, 2012.

[27] A. Merzky, M. Santcroos, M. Turilli, and S. Jha, "Radical-pilot: Scalable execution of heterogeneous and dynamic workloads on supercomputers," *Computer Research Repository (CoRR), abs/1512.08194*, 2015.

[28] T. G. Armstrong, Z. Zhang, D. S. Katz, M. Wilde, and I. T. Foster, "Scheduling many-task workloads on supercomputers: Dealing with trailing tasks," in *2010 3rd Workshop on Many-Task Computing on Grids and Supercomputers*, pp. 1–10, IEEE, 2010.

[29] W. Tang, D. Ren, Z. Lan, and N. Desai, "Adaptive metric-aware job scheduling for production supercomputers," in *2012 41st International Conference on Parallel Processing Workshops*, pp. 107–115, IEEE, 2012.

[30] W. Tang, D. Ren, Z. Lan, and N. Desai, "Toward balanced and sustainable job scheduling for production supercomputers," *Journal of Parallel Computing*, vol. 39, no. 12, pp. 753–768, 2013.

[31] W. Tang, N. Desai, D. Buettner, and Z. Lan, "Job scheduling with adjusted runtime estimates on production supercomputers," *Journal of Parallel and Distributed Computing*, vol. 73, no. 7, pp. 926–938, 2013.

[32] S. Koposov and O. Bartunov, *Q3C, Quad Tree Cube – The new Sky-indexing Concept for Huge Astronomical Catalogues and its Realization for Main Astronomical Queries (Cone Search and Xmatch) in Open Source Database PostgreSQL*, vol. 351 of *Astronomical Society of the Pacific Conference Series*, p. 735. Astronomical Society of the Pacific, 2006.

[33] M. Singh and M. Valtorta, "Construction of bayesian network structures from data: a brief survey and an efficient algorithm," *International journal of approximate reasoning*, vol. 12, no. 2, pp. 111–131, 1995.

[34] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural networks*, vol. 4, no. 2, pp. 251–257, 1991.

[35] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.