

Transient processing and analysis using AMPEL: alert management, photometry, and evaluation of light curves[★]

J. Nordin¹, V. Brinnet¹, J. van Santen², M. Bulla³, U. Feindt³, A. Franckowiak², C. Fremling⁴, A. Gal-Yam⁵, M. Giomi¹, M. Kowalski^{1,2}, A. Mahabal^{4,6}, N. Miranda¹, L. Rauch², S. Reusch¹, M. Rigault⁷, S. Schulze⁵, J. Sollerman^{3,8}, R. Stein², O. Yaron⁵, S. van Velzen⁹, and C. Ward⁹

¹ Institute of Physics, Humboldt-Universität zu Berlin, Newtonstr. 15, 12489 Berlin, Germany
e-mail: Jakob.Nordin@physik.hu-berlin.de

² Deutsches Elektronen-Synchrotron, 15735 Zeuthen, Germany

³ The Oskar Klein Centre, Department of Physics, Stockholm University, AlbaNova, 106 91 Stockholm, Sweden

⁴ Division of Physics, Mathematics, and Astronomy, California Institute of Technology, Pasadena, CA 91125, USA

⁵ Department of Particle Physics and Astrophysics, Weizmann Institute of Science 234 Herzl St., 76100 Rehovot, Israel

⁶ Center for Data Driven Discovery, California Institute of Technology, Pasadena, CA 91125, USA

⁷ Université Clermont Auvergne, CNRS/IN2P3, Laboratoire de Physique de Clermont, 63000 Clermont-Ferrand, France

⁸ Department of Astronomy, Stockholm University, AlbaNova, 106 91 Stockholm, Sweden

⁹ Department of Astronomy, University of Maryland, College Park, MD 20742, USA

Received 5 April 2019 / Accepted 10 June 2019

ABSTRACT

Context. Both multi-messenger astronomy and new high-throughput wide-field surveys require flexible tools for the selection and analysis of astrophysical transients.

Aims. Here we introduce the alert management, photometry, and evaluation of light curves (AMPEL) system, an analysis framework designed for high-throughput surveys and suited for streamed data. AMPEL combines the functionality of an alert broker with a generic framework capable of hosting user-contributed code; it encourages provenance and keeps track of the varying information states that a transient displays. The latter concept includes information gathered over time and data policies such as access or calibration levels.

Methods. We describe a novel ongoing real-time multi-messenger analysis using AMPEL to combine IceCube neutrino data with the alert streams of the Zwicky Transient Facility (ZTF). We also reprocess the first four months of ZTF public alerts, and compare the yields of more than 200 different transient selection functions to quantify efficiencies for selecting Type Ia supernovae that were reported to the Transient Name Server (TNS).

Results. We highlight three channels suitable for (1) the collection of a complete sample of extragalactic transients, (2) immediate follow-up of nearby transients, and (3) follow-up campaigns targeting young, extragalactic transients. We confirm ZTF completeness in that all TNS supernovae positioned on active CCD regions were detected.

Conclusions. AMPEL can assist in filtering transients in real time, running alert reaction simulations, the reprocessing of full datasets as well as in the final scientific analysis of transient data. This is made possible by a novel way of capturing transient information through sequences of evolving states, and interfaces that allow new code to be natively applied to a full stream of alerts. This text also introduces a method by which users can design their own channels for inclusion in the AMPEL live instance that parses the ZTF stream and the real-time submission of high-quality extragalactic supernova candidates to the TNS.

Key words. methods: data analysis – astronomical databases: miscellaneous – virtual observatory tools – supernovae: general – cosmology: observations

1. Introduction

Transient astronomy has traditionally used optical telescopes to detect variable objects, both within and beyond our Galaxy, with a peak sensitivity for events that vary on weekly or monthly timescales. This field has now entered a new phase in which multi-messenger astronomy allows for near real-time detections of transients through correlations between observations of different messengers. The initial report of GW170817 from LIGO/VIRGO and the subsequent search and detection of an X-ray/optical counterpart provides a first, inspiring example of this (Abbott et al. 2017). Shortly afterwards, the observation of a

flaring blazar coincident with a high-energy neutrino detected by IceCube again illustrated the scientific potential of time domain multi-messenger astronomy (Aartsen & Ackermann 2018). Optical surveys now observe the full sky daily, to a depth which encompasses both distant, bright objects and nearby, faint ones. We can thus simultaneously find rare objects, obtain an account of the variable Universe, and probe fundamental physics at scales beyond the reach of terrestrial accelerators. Exploiting these opportunities is currently constrained as much by software and method development as by available instruments (Allen et al. 2018).

The plans for the Large Synoptic Survey Telescope (LSST) provide a sample scale for high-rate transient discovery. The LSST is expected to scan large regions of the sky to great depth, with sufficient cadence for more than 10^6 astrophysical

[★] Table A.1 is only available at the CDS via anonymous ftp to [cdsarc.u-strasbg.fr](ftp://cdsarc.u-strasbg.fr) (130.79.128.5) or via <http://cdsarc.u-strasbg.fr/viz-bin/cat/J/A+A/631/A147>

transients to be discovered each night. Each such detection will be immediately streamed to the community as an alert. The challenge of distributing this information for real-time follow-up observations is to be solved through a set of brokers, which will receive the full data flow and allow end users to select the small subset that merits further study (Juric et al. 2017). Development first started on the Arizona-NOAO temporal analysis and response to events system (ANTARES), which provides a system for real-time characterization and annotation of alerts before they are relayed further downstream (Saha et al. 2014). Other current brokers include MARS¹ and LASAIR (Smith et al. 2019). Earlier systems for transient information distribution include the Central Bureau for Astronomical Telegrams (CBAT), the Gamma-ray Coordinates Network and the Astronomer’s Telegram. The Catalina Real-Time Transient Survey was designed to make transient detections public within minutes of observation (Drake et al. 2009; Mahabal et al. 2011). More recent developments include the Astrophysical Multimessenger Observatory Network (AMON, Smith et al. 2013), which provides a framework for real-time correlation of transient data streams from different high-energy observatories, and the Transient Name Server (TNS), which maintains the current IAU repository for potential and confirmed extragalactic transients².

While LSST will come online only in 2022, the Zwicky Transient Facility (ZTF) has been operating since March 2018 (Graham 2019). The ZTF employs a wide-field camera mounted on the Palomar P48 telescope, and is capable of scanning more than 3750 square degrees to a depth of 20.5 mag each hour (Bellm et al. 2019). This makes ZTF a wider, shallower precursor to LSST, with a depth more suited to spectroscopic follow-up observations. Observations by ZTF are immediately transferred to the Infrared Processing and Analysis Center (IPAC) for processing and image subtraction (Masci et al. 2019). Any significant point source-like residual flux in the subtracted image triggers the creation of an alert. Alerts are serialized and distributed through a Kafka³ server hosted at the DiRAC center at the University of Washington (Patterson et al. 2019). Each alert contains primary properties like position and brightness, but also ancillary detection information and higher-level derived values such as the RealBogus score which aims to distinguish real detections from image artifacts (Mahabal et al. 2019). Full details on the reduction pipeline and alert content can be found in Masci et al. (2019), while an overview of the information distribution can be found in the top row of Fig. 3. The ZTF will conduct two public surveys as part of the US NSF Mid-Scale Innovations Program (MSIP). One of these, the Northern Sky Survey, performs a three-day cadence survey in two bands of the visible northern sky.

Here, we present AMPEL (alert management, photometry, and evaluation of light curves) as a tool to accept, process, and react to streams of transient data. AMPEL contains a broker as the first of four pipeline levels, or “tiers”, but complements this with a framework enabling analysis methods to be easily and consistently applied to large volumes of data. The same set of input data can be repeatedly reprocessed with progressively refined analysis software, while the same algorithms can then also be applied to real-time, archived, and simulated data samples. Analysis and reaction methods can be contributed through the implementation of simple python classes, ensuring that the vast majority of current community tools can be immediately put to

use. AMPEL functions as a public broker for use with the public ZTF alert stream, meaning that community members can provide analysis units for inclusion in the real-time data processing. AMPEL also brokers alerts for the private ZTF partnership. Selected transients together with derived properties are pushed into the GROWTH Marshal (Kasliwal et al. 2019) for visual examination, discussion, and the potential trigger of follow-up observations.

This paper is structured as follows: AMPEL requirements are first described in Sect. 2, after which the design concepts are presented in Sect. 3. Some specific implementation choices are detailed in Sect. 4 and instructions for using AMPEL are provided in Sect. 5. In Sect. 6 we present sample AMPEL uses: systematic reprocessing of archived alerts to investigate transient search completeness and efficiency, photometric typing, and live multi-messenger matching between optical and neutrino data-streams. Our discussion in Sect. 7 introduces the automatic AMPEL submission of high-quality extragalactic astronomical transients to the TNS, from which astronomers can immediately find potential supernovae or AGNs without having to do any broker configuration. The material presented here focuses on the design and concepts of AMPEL, and acts as a complement to the software design tools contained in the AMPEL sample repository⁴. We encourage the interested reader to consult this repository in parallel to this text. We describe the AMPEL system using terms that may not coincide with those used in other fields. This terminology is introduced gradually in this text, but is summarized in Table 1 for reference.

2. Requirements

Guided by an overarching goal of analyzing data streams, here we lay out the design requirements that shaped the AMPEL development.

Provenance and reproducibility. Data provenance encapsulates the philosophy that the origin and manipulation of a dataset should be easily traceable. As data volumes grow, and as astronomers increasingly seek to combine ever more diverse datasets, the concept of data provenance will be of central importance. In this era, individual scientists can be expected neither to master all details of a given workflow, nor to inspect all data by hand. As an alternative, these scientists must instead rely on documentation accompanying the data. While provenance is a minimal requirement for such analyses, a more ambitious goal is replayability. Replaying an archival transient survey offline would involve providing a virtual survey in which the entire analysis chain is simulated, from transient detection to the evaluation of triggered follow-up observations. In essence, this amounts to answering the following question: if I had changed my search or analysis parameters, what candidates would have been selected? Because any given transient will only be observed once, replayability is as close to the standard scientific goal of reproducibility as astronomers can get.

Analysis flexibility. The following decades will see an unprecedented range of complementary surveys looking for transients through gravitational waves, neutrinos, and multiwavelength photons. These will feed a sprawling community of diverse science interests. We would like a transient software framework that is sufficiently flexible to give full freedom in analysis design, while still being compatible with existing tools and software.

⁴ <https://github.com/AmpelProject/Ampel-contrib-sample>

¹ <https://mars.lco.global/>

² <https://wis-tns.weizmann.ac.il/>

³ <https://kafka.apache.org>

Table 1. AMPEL terminology.

Term	AMPEL interpretation
Transient Datapoint Compound State	Object with a unique ID provided from a data source and accepted into AMPEL by at least one channel A single measurement with a specific calibration, processing level, etc. A collection of datapoints (from one or more instruments) A view of a transient object available at some point for some observer. Connects a compound with one (or more) transients
Tier	AMPEL is internally divided into four tiers, where each performs a different kind of operation and is controlled by a separate scheduler
Channel	Configuration of requested behavior at all AMPEL tiers supplied by a user (for one science goal). Typically consists of a list of requested units together with their run parameters
Archive ScienceRecord	All alert data, and also those rejected during live processing, are stored in an archive for reprocessing Records the result of a science computation made based on data available in specific state
TransientView	All information available regarding a specific transient. This can include multiple states, and any Science Records associated with these
Unit	Typically implemented as python modules, a unit allows user-contributed code to be directly called during data processing. Units at different tiers receive different input and are expected to produce different kinds of output
Journal	A time-ordered log included in each transient
Purge	The transfer of a no-longer-active transient from the live database to external storage. This includes all connected datapoints, states, compounds, and ScienceRecords
Live instance	A version of AMPEL processing data in real-time. This includes a number of active channels
Reprocessing	Parsing archived alerts as they would have been received in real-time, using a set of channels defined as for a live instance

Versions of data and software. It is typical that the value of a measurement evolves over time, from a preliminary real-time result to final published data. This is driven by both changes in the quantitative interpretation of the observations and a progressive increase in analysis complexity. The first dimension involves changes such as improved calibration, while the second incorporates, for example, more computationally expensive studies only run on subsets of the data. So far, studying the full impact of incremental changes in these two dimensions has been difficult. To change this requires an end-to-end streaming analysis framework where any combination of data and software can be conveniently explored. A related community challenge is to recognize, reference, and motivate continued development of well-written software.

Alert rate. Current optical transient surveys such as DES, ZTF, ASAS-SN, and ATLAS, as well as future ones (LSST), do or will provide tens of thousands to millions of detections each night. On such a scale, human inspection of all candidates is impossible, even when assuming that artifacts are perfectly removed⁵. A simplistic solution to this problem is to only select a very small subset from the full stream, for example a handful of the brightest objects, for which additional human inspection is feasible. A more complete approach would be based on retaining much larger sets of targets throughout the analysis, from which subsets are complemented with varying levels of follow-up information. As the initial subset selection will, by necessity, be done in an automated streaming context, the accompanying analysis framework must be able to trace and model these real-time decisions.

⁵ For optical surveys, a majority of these “detections” are actually artifacts induced through the subtraction of a reference image. Machine-learning techniques, such as RealBogus for ZTF, are increasingly powerful at separating these from real astronomical transients. However, this separation can never be perfect and any transient program has to decide how strictly they adhere to these classifications.

3. AMPEL in a nutshell

AMPEL is a framework for analyzing and reacting to streamed information, with a focus on astronomical transients. Fulfilling the above design goals requires a flexible framework built using a set of general concepts. These will be introduced in this section, accompanied by examples based on optical data from ZTF. The “life” of a transient in AMPEL is outlined in parallel in Figs. 1 and 2. These figures further illustrate many of the concepts introduced in this section. Figure 1 shows AMPEL used as a straightforward alert broker, while Fig. 2 includes many of the additional features that make AMPEL a full analysis framework.

The core object in AMPEL is a transient, a single object identified by a creation date and typically a region of origin in the sky. Each transient is linked to a set of datapoints that represent individual measurements⁶. Datapoints can be added, updated, or marked as bad. Datapoints are never removed. Each datapoint can be associated with tags indicating, for example, any masking or proprietary restrictions. Transients and datapoints are connected by states, where a state references a compound of datapoints. A state represents a view of a transient available at a particular time and for a particular observer. For an optical photometric survey, a compound can be directly interpreted as a set of flux measurements or a light curve.

Example. A ZTF alert corresponds to a potential transient. Datapoints here are simply the photometric magnitudes reported by ZTF, which in most cases consists of a recent detection and a history of previous detections or non-detections at this position. When first inserted, a transient has a single state with a compound consisting of the datapoints in the initial alert. Should a new alert be received with the same ZTF ID, the new datapoints contained in this alert are added to the collection and a new state is created containing both previous and new data. Should the

⁶ We note that this is a many-to-many connection; multiple transients can be connected to the same datapoint due to e.g., positional uncertainty. Datapoints can also originate from different sources.

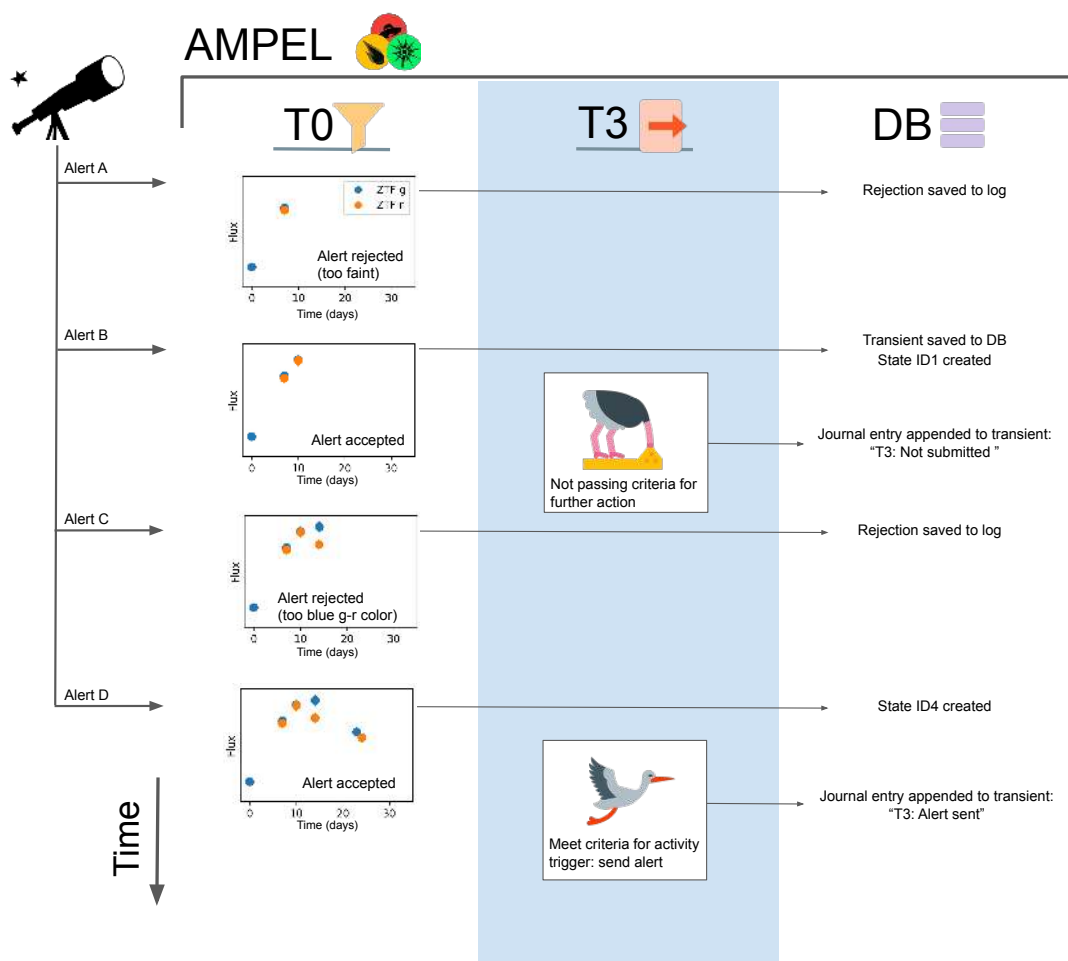


Fig. 1. Outline of AMPEL, acting as broker. Four alerts, A–D, belonging to a unique transient candidate are being read from a stream. In a first step, “Tier 0”, the alert stream is filtered based on alert keywords and catalog matching. Alerts B and D are accepted. In a second step, “Tier 3”, the external resources that AMPEL should notify are chosen. In this example, only Alert D warrants an immediate reaction. The final column shows the corresponding database events.

first datapoint be public but the second datapoint be private, only users with proper access will see the updated state.

Using AMPEL means creating a channel, corresponding to a specific science goal, which prescribes behavior at four different stages, or tiers. The tasks performed at each tier can be determined by answering the following questions: “Tier 0: What are the minimal requirements for an alert to be considered interesting?”, “Tier 1: Can datapoints be changed by events external to the stream?”, “Tier 2: What calculations should be done on each of the candidate states?”, “Tier 3: What operations should be done at timed intervals or on populations of transients?”⁷

– Tier 0 (T0) filters the full alert stream to only include potentially interesting candidates. This tier thus works as a data broker: objects that merit further study are selected from the incoming alert stream. However, unlike most brokers, accepted transients are inserted into a database (DB) of active transients rather than immediately being sent downstream. All alerts, including those that are rejected, are stored in an external archive DB. Users can either provide their own algorithm for filtering, or configure one of the filter classes already available according to their needs.

Example T0. The simple AMPEL channel “BrightAndStable” looks for transients with at least three “well-behaved” detections (few bad pixels and reasonable subtraction FWHM) that are not coincident with a *Gaia* DR2 star-like source. This is implemented through a python class `SampleFilter` that operates on an alert and returns either a list of requests for follow-up (T2) analysis, if selection criteria are fulfilled, or `False` if they are not. AMPEL will test every ZTF alert using this class, and all alerts that pass the cut are added to a DB containing all active transients. The transient is in the DB associated with the channel “BrightAndStable”.

– Tier 1 (T1) is largely autonomous and exists in parallel to the other tiers. T1 carries out duties of assigning datapoints and T2 run requests to transient states. Example activities include completing transient states with datapoints that were present in new alerts but where these were not individually accepted by the channel filter (e.g., in the case of lower significance detections at late phases), as well as querying an external archive for updated calibration or adding photometry from additional sources. A T1 unit could also parse previous alerts at or close to the transient position for old data to include with the new detection.

– Tier 2 (T2) derives or retrieves additional transient information, and is always connected to a state and stored as a ScienceRecord. Tier 2 units either work with the empty state, relevant for catalog matching that only depends on the position

⁷ Timed intervals include very high frequencies or effectively real-time response channels.

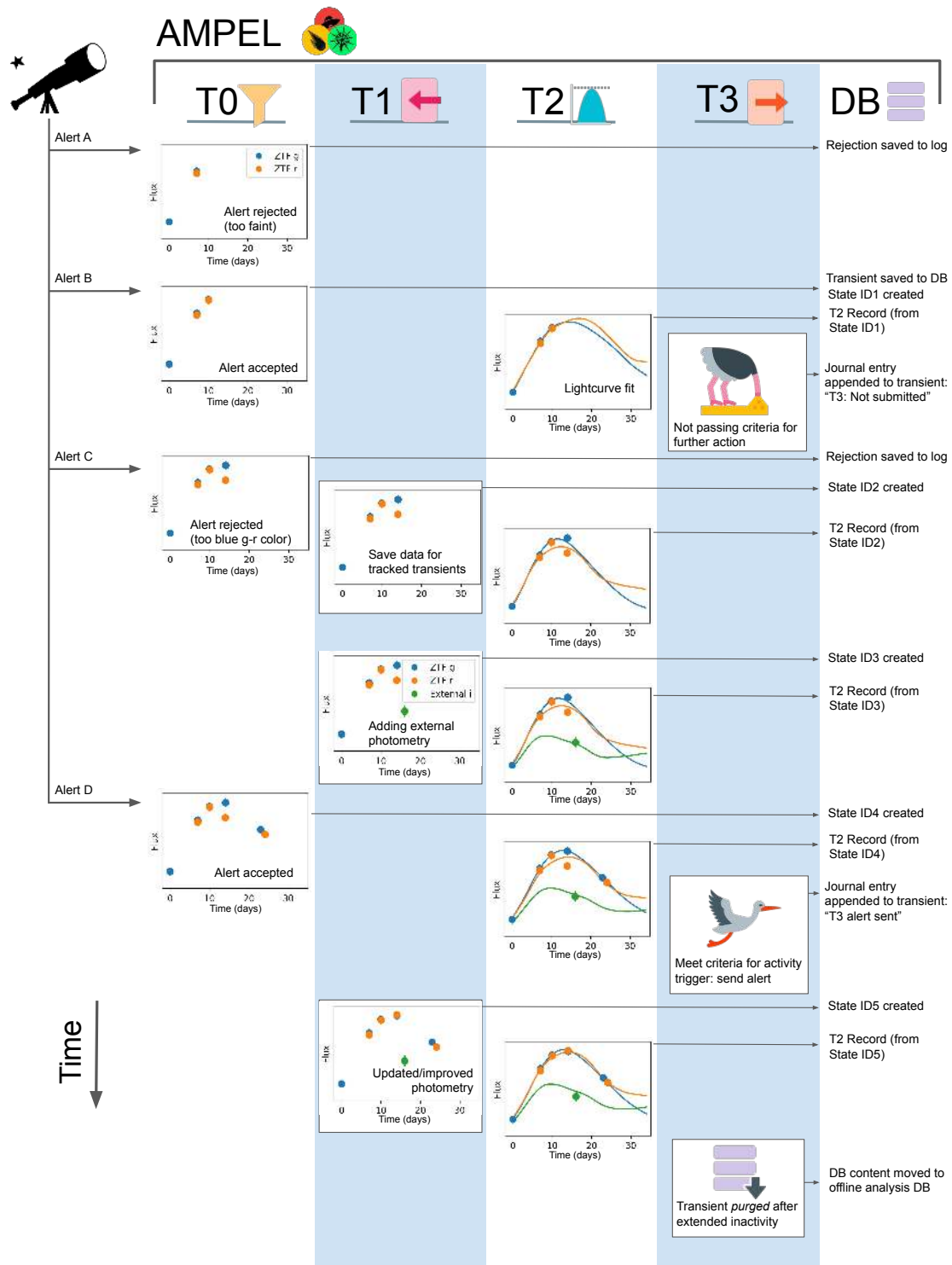


Fig. 2. Life of a transient in AMPEL. Sample behavior at the four tiers of AMPEL as well as the database access are shown as columns, with the left side of the figure indicating when the four alerts belonging to the transient were received. T0: The first and third alerts are rejected, while the second and fourth fulfill the channel acceptance criteria. T1: The first T1 panel shows how the data content of an alert which was rejected at the T0 stage but where the transient ID was already known to AMPEL is still ingested into the live DB. The second panel shows an external datapoint (measurement) being added to this transient. The final T1 panel shows how one of the original datapoints is updated. All T1 operations lead to the creation of a new state. T2: The T2 scheduler reacts every time a new state is created and queues the execution of all T2s requested by this channel. In this case this causes a light-curve fit to be performed and the fit results are stored as ScienceRecords. T3: The T3 scheduler schedules units for execution at pre-configured times. In this example this is a (daily) execution of a unit testing whether any modified transient warrants a Slack posting (requesting potential further follow-up). The submit criteria are fulfilled the second time the unit is run. In both cases, the evaluation is stored in the transient Journal, which is later used to prevent a transient from being posted multiple times. Once the transient has not been updated for an extended time, a T3 unit purges the transient to an external database that can be directly queried by channel owners. Database: A transient entry is created in the DB as the first alert is accepted. After this, each new datapoint causes a new state to be created. T2 ScienceRecords are each associated with one state. The T3 units return information that is stored in the Journal.

for example, or they depend on the datapoints of a state to calculate new, derived transient properties. In the latter case, the T2 task will be called again as soon as a new state is created. This could be due to new observations or, for example, updated calibration of old datapoints. Possible T2 units include light-curve fitting, photometric redshift estimation, machine-learning classification, and catalog matching.

Example T2. For an optical transient, a state corresponds to a light curve and each photometric observation is represented by a datapoint. A new observation of the transient would extend the light curve and thus create a new state. “BrightAndStable” requests a third-order polynomial fit for each state using the `T2PolyFit` class. The outcome – in this case polynomial coefficients – are saved to the database.

– Tier 3 (T3), the final AMPEL level, consists of schedulable actions. While T2s are initiated by events (the addition of new states), T3 units are executed at pre-determined times. These can range from yearly data dumps to daily updates, or to effectively real-time execution every few seconds. Tier 3 processes access data through the `TransientView`, which concatenates all information regarding a transient. This includes both states and `ScienceRecords` that are accessible by the channel. Tier 3 processes iterate through all transients of a channel which have been updated since a previous time-stamp (either the last time the T3 was run or a specified time-range). This allows for an evaluation of multiple `ScienceRecords` and comparisons between different objects (such as any kind of population analysis). One typical case is the ranking of candidates which would be interesting to observe on a given night. Tier 3 units include options to push and pull information to and from, for example, the TNS, web-servers, and collaboration communication tools such as Slack⁸.

Example T3. The science goal of “BrightAndStable” is to observe transients with a steady rise. At the T3 stage, the channel therefore loops through the `TransientViews`, and examines all `T2PolyFit` `ScienceRecords` for fit parameters that indicate a lasting linear rise. Any transients fulfilling the final criteria trigger an immediate notification sent to the user. This test is scheduled to be performed at 13:15 UTC each day.

4. Implementation

Here we expand on a selection of implementation aspects. An overview of the live instance processing of the ZTF alert stream can be found in Fig. 3.

Modularity and units. Modularity is achieved through a system of units. These are Python modules that can be incorporated with AMPEL and directly applied to a stream of data. Units are inherited from abstract classes that regulate the input and output data format, but have great freedom in implementing what is done with the input data. The tiers of AMPEL are designed such that each requires a specific kind of information: at Tier 0 the input is the raw alert content, at Tier 2 a transient state, and at Tier 3 a transient view. The system of base classes allows AMPEL to provide each unit with the required data. In a similar system, each unit is expected to provide output data (results) in a specific format to make sure this is stored appropriately: at Tier 0 the expected output is a list of Tier 2 units to run at each state for accepted transients (None for rejected transients). At Tier 2 output is a `ScienceRecord` (dictionary) which in the DB is automatically linked to the state from which it was derived. The T3 output

is not state-bound, but is rather added to the transient journal, a time-ordered history accompanying each transient. Modules at all tiers can make direct use of well-developed libraries such as `numpy` (Oliphant 2006), `scipy` (Jones et al. 2001), and `astropy` (Astropy Collaboration 2013; Price-Whelan et al. 2018). Developers can choose to make their contributed software available to other users and gain recognition for functional code, or keep them private. The modularity means that users can independently vary the source of alerts, calibration version, selection criteria, and analysis software.

Schemas and AMPEL shapers. Contributed units will be limited as long as they have to be tuned for a specific kind of input, such as ZTF photometry for example. Eventually, we hope that more general code can be written through the development of richer schemas for astronomical information based on which units can be developed and immediately applied to different source streams. The International Virtual Observatory Alliance (IVOA) initiated the development of the `VOEvent` standard with this purpose⁹. Core information of each event is to be mapped to a set of specific tags (such as Who, What, Where, When) stored in an XML document. `VOEvents` form a starting point for this development (see e.g., Williams et al. 2009), but more work is needed before a general T2 unit can be expected to immediately work on data from all sources. As an intermediate solution, AMPEL employs shapers that can translate source-specific parameters to a generalized data format that all units can rely on. While the internal AMPEL structure is designed for performance and flexibility, it is easy to construct T3 units that export transient information according to, for example, `VOEvent` or GCN specifications.

The archive. Full replayability requires that all alerts are available at later times. While most surveys are expected to provide this, we keep local copies of all alerts until other forms of access are guaranteed.

Catalogs, Watch-lists and ToO triggers. Understanding astronomical transients frequently requires matches to known source catalogs. AMPEL currently provides two resources to this end. A set of large, pre-packaged catalogs can be accessed using `catsHTM`, including the *Gaia* DR2 release (Soumagnac & Ofek 2018). As a complement, users can upload their own catalogs using `extcats`¹⁰ for either transient filtering or to annotate transients with additional information. `extcats` is also used to create watch-lists and ToO channels. Watchlists are implemented as a T0 filter that matches the transient stream with a contributed `extcat` catalog. A ToO channel has a similar functionality, but employs a dynamic `extcat` target list where a ToO trigger immediately adds one or more entries to the matchlist. The stream can in this case initially be replayed from some previous time (a delayed T0), which allows preexisting transients to be consistently detected.

The live database. The live transient DB is built using the `NoSQL MongoDB`¹¹ engine. The flexibility of not having an enforced schema allows AMPEL to integrate varying alert content and give full freedom to algorithms to provide output of any shape. The live AMPEL instance is a closed system that users cannot directly interact with, and contributed units do not directly interact with the DB. Instead, the AMPEL core system manages

⁹ <http://www.ivoa.net/documents/VOEvent/20110711/REC-VOEvent-2.0.pdf>

¹⁰ <https://github.com/MatteoGiomi/extcats>

¹¹ <https://docs.mongodb.com/manual/>

⁸ <https://slack.com>

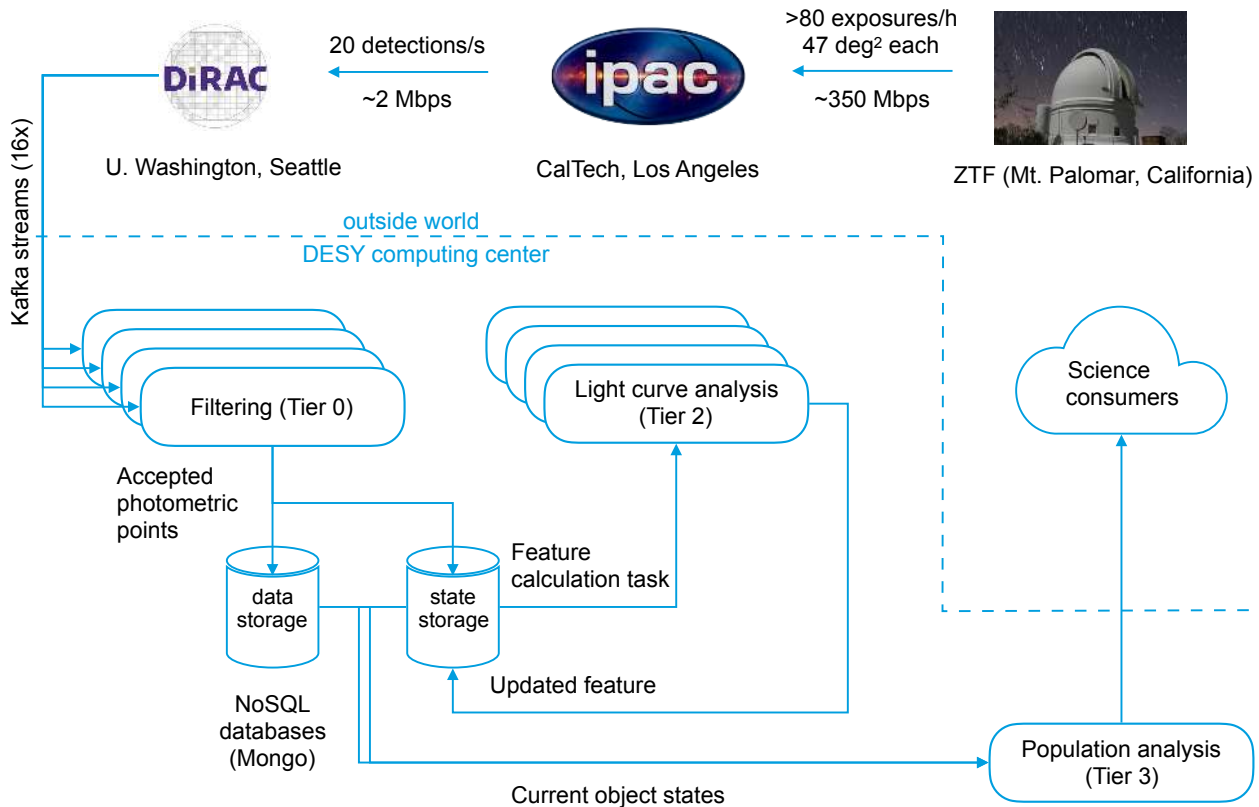


Fig. 3. AMPEL schematic for the live processing of ZTF alerts. External events, above dashed lines: This includes ZTF observations, processing, and the eventual alert distribution through the DiRAC center. Finally, science consumers external to AMPEL receive output information. This can include both tools for transient visualization (“front-ends”) as well as alerts through e.g., TNS or GCN. A set of parallel alert processors examine the incoming Kafka Stream (Tier 0). Accepted alert data are saved into a collection, while states are recorded in another. A light curve analysis (Tier 2) is performed on all states. The available data, including the Tier 2 output, are examined in a Tier 3 unit that selects which transients should be passed out. This particular use case does not contain a Tier 1 stage.

interactions through the alert, state, and transient view objects introduced above¹². Each channel also specifies conditions for when a transient is no longer considered “live”; at this point it is purged, that is, extracted from the live DB together with all states, computations, and logs, and then inserted into a channel-specific offline DB which is provided to the channel owner.

Horizontal scaling. AMPEL is designed to be fully parallelizable. The DB, the alert processors, and tier controllers all scale horizontally such that additional workers can be added at any stage to compensate for changes to the workload. Alerts can be processed in any order, that is, not necessarily in time-order.

AMPEL instances and containers. An AMPEL instance is created through combining tagged versions of core and contributed units into a Docker (Merkel 2014) image, which is then converted to the Singularity (Kurtzer et al. 2017) format for execution by an unprivileged user. The final product is a unique “container” that is immutable and encapsulates the AMPEL software, contributed units, and their dependencies. These can be reused and referenced for later work, even if the host environment changes significantly. The containers are coordinated with a simple orchestration tool¹³ that exposes an interface similar to Docker’s “swarm mode.” Previously deployed AMPEL versions are stored, and can be run off-line on any sequence of archived

or simulated alerts. Several instances of AMPEL might be active simultaneously, with each processing either a fraction of a full live-stream, or some set of archived or simulated alerts; each works with a distinct database. The current ZTF alert flow can easily be parsed by a single instance, called the live instance. A full AMPEL analysis combines this active parsing and reacts to the live streams with subsequent or parallel runs in which the effects of the channel parameters can be systematically explored.

Logs and provenance. AMPEL contains extensive, built-in logging functions. All AMPEL units are provided a logger, and we recommend this to be consistently used. Log entries are automatically tagged with the appropriate channel and transient ID, and are then inserted into the DB. These tools, together with the DB content, alert archive, and AMPEL container, make provenance straightforward. The IVOA has initiated the development of a provenance data model (DM) for astronomy, following the definitions proposed by the W3C (Sanguillon et al. 2017)¹⁴. Scientific information is described here as flowing between agents, entities, and activities. These are related through causal relations. The AMPEL internal components can be directly mapped to the categories of the IVOA provenance DM: Transients, datapoints, states, and ScienceRecords are entities; Tier units are activities and users; AMPEL maintainers, software developers, and alert stream producers are agents. A streaming analysis carried out

¹² Eventually, daily snapshot copies of the DB will be made available for users to interactively examine the latest transient information without being limited with what was reconfigured to be exported.

¹³ <https://github.com/AmpelProject/singularity-stack>

¹⁴ <http://www.ivoa.net/documents/ProvenanceDM/20181015/PR-ProvenanceDM-1.0-20181015.pdf>

in AMPEL will thus automatically fulfill the IVOA provenance requirements.

Hardware requirements. The current live instance installed at the DESY computer center in Berlin-Zeuthen consists of two machines, “Burst” and “Transit”. Real-time alert processing is done at Burst (32 cores, 96 GB memory, 1 TB SSD) while alert reception and archiving is done at Transit (20 cores, 48 GB memory, 1 TB SSD + medium-time storage). This system has been designed for extragalactic programs based on the ZTF survey, with a few tens of thousands of alerts processed each night, of which between 0.1 and 1% are accepted. Reprocessing large alert volumes from the archive on Transit is done at a mean rate of 100 alerts per second. As the ZTF live alert production rate is lower than this, and Burst is a more powerful machine, this setup is never running at full capacity. It would be straightforward to distribute processing of T2 and T3 tasks among multiple machines, but as the expected practical limitation is access to a common database, this is of limited use until extremely demanding units are requested.

5. Using AMPEL

5.1. Creating a channel for the ZTF alert stream

The process for creating AMPEL units and channels is fully described in the `Ampel-contrib-sample` repository¹⁵, which also contains a set of sample channel configurations. The steps to implementing a channel can be summarized as follows:

- (1) Fork the sample repository and rename it `Ampel-contrib-groupID` where `groupID` is a string identifying the contributing science team.
- (2) Create units through populating the `t0/t2/t3` sub-directories with Python modules. Each is designed through inheritance from the appropriate base class.
- (3) Construct the repository channels by defining their parameters in two configuration files: `channels.json` which defines the channel name and regulates the T0, T1, and T2 tiers, and `t3_jobs.json` which determines the schedule for T3 tasks. These can be constructed to make use of AMPEL units present either in this repository, or from other public AMPEL repositories.
- (4) Notify AMPEL administrators. The last step will trigger channel testing and potential edits. After the channel is verified, it will be added to the list of AMPEL contribution units and included in the next image build. The same channel can also (or exclusively) be applied to archived ZTF alerts.

5.2. Using AMPEL for other streams

Nothing in the core AMPEL design is directly tied to the ZTF stream, or even to the optical data. The only source-specific software class is the Kafka client reading the alert stream, and the alert shapers, which make sure key variables such as coordinates are stored in a uniform matter. Using a schema-free DB means that any stream content can be stored by AMPEL for further processing. A more complex question concerns the design of units that are usable with different stream sources. As an example, different optical surveys use different conventions when encoding filters, magnitude reference systems, and photometric uncertainties, and they often provide unique alert metrics (such as the Real-Bogus value of ZTF). Until common standards are developed, classes will have to be tuned directly to every new alert stream.

¹⁵ <https://github.com/AmpelProject/Ampel-contrib-sample>

6. Initial AMPEL applications

6.1. Exploring the ZTF alert parameter space

It has been notoriously challenging to quantify transient detection efficiencies, search old surveys for new kinds of transients, and predict the likely yield from a planned follow-up campaign. Here we demonstrate how AMPEL can assist with such tasks. For this case study we reprocess 4 months of public ZTF alerts using a set of AMPEL filters spanning the parameter space of the main properties of ZTF alerts. The accepted samples of each channel are, in a second step, compared with confirmed Type Ia supernovae (SNe Ia) reported to the TNS during the same period. We can thus examine how different channel permutations differ in detection efficiency, and at what phase each SNIa was “discovered”. The base comparison sample consists of 134 normal SNe Ia. The creation of this sample is described in detail in Appendix A.

We processed the ZTF alert archive from June 2 2018 (start of the MSIP Northern Sky Survey) to October 1 2018 using 90 potential filter configurations based on the `DecentFilter` class. In total 28 667 252 alerts were included. Each channel exists on a grid constructed by varying the properties described in Table 2. We also include 24 OR combinations where the accept criteria of two filters are combined. We further consider two additional versions of each filter or filter-combination:

- (1) Transients in galaxies with known active Sloan Digital Sky Survey (SDSS) or MILLIQUAS active nuclei (Flesch 2015; Pâris et al. 2017) are rejected;
- (2) Transients are required to be associated with a galaxy for which there is a known NASA/IPAC Extragalactic Database (NED) or SDSS spectroscopic redshift $z < 0.1$.

In total, this amounts to 342 combinations. All of these variants include some version of alert rejection based on coincidence with a star-like object in either PanSTARRS (using the algorithm of Tachibana & Miller 2018) or *Gaia* DR2 (Gaia Collaboration 2018). We also tested channels not including any such rejection, which lead to transient counts of around 10 000 (an order of magnitude greater than with the star-rejection veto). Reprocessing the alert stream in this way took 5 days even in a nonoptimized configuration, demonstrating that AMPEL can process data at the expected LSST alert rate.

This study is neither complete nor unbiased: A large fraction of the SNe were classified by ZTF, and we know that the real number of SNe Ia observed is much larger than the classified subset. Nonetheless it serves both as a benchmark test for channel creation, as well as a starting point for a more thorough analysis. An estimate of the total number of supernovae we expect to be hidden in the ZTF detections can be obtained through the `simsurvey` code (Feindt et al. 2019), in which known transient rates are combined with a realistic survey cadence and a set of detection thresholds¹⁶. The predicted number of SNe Ia fulfilling the criteria of one or more of these channels over the same time-span as the comparison sample and with weather conditions matching those observed was found to be 1033 (average over ten simulations). `Simsurvey` also conveniently returns estimates for other supernova types and we find that an additional 276 Type Ibc, 92 Type IIIn, and 377 Type IIP supernovae are likely to have been observed by ZTF under the same conditions. The total number of supernovae present in the alert sample is therefore estimated to be 1778.

The results for channel efficiencies compared to the total number of accreted transients can be found in Fig. 4. Though

¹⁶ <https://github.com/ufeindt/simsurvey>

Table 2. Dominant channel selection variables and potential settings.

Channel property	Options
RealBogus Detections	Nominal: Require ML score above 0.3 or Strong: above 0.5
Alert history	More than [2, 4, 6, 8] (any filter)
Image quality	New: Not older than 5 days, Multi-night: 4–15 days, Persistent: Older than 8 days
<i>Gaia</i> DR2	All: No requirements, Good: Limited cuts on e.g., FWHM and bad pixels, Excellent: Strong cuts on e.g., FWHM and bad pixels
Star-Galaxy separation	Nominal: Reject likely stars from <i>Gaia</i> DR2, Moderate: only search in small aperture or Disabled
Match confusion	Using PS1 star-galaxy separation (Tachibana & Miller 2018) to reject potential stars (Hard), likely stars (Nominal) or no rejection (Disabled)
	Nominal: Allow candidates close to nearby (confused) sources, or Disabled: reject anything close to stars even if other sources exist

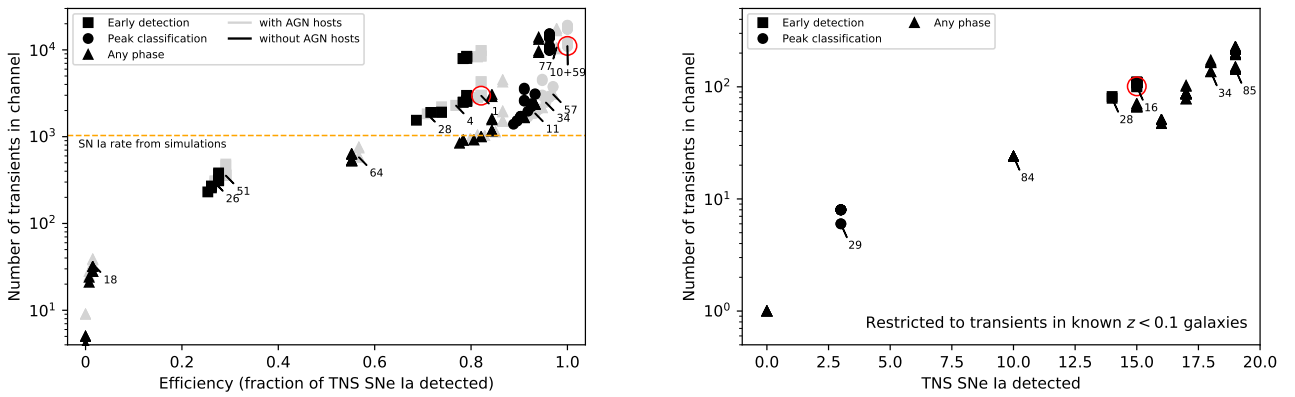


Fig. 4. Comparison of the total number of accepted candidates (y -axis) with the fraction of the comparison sample SNe Ia detected (x -axis). Symbol shapes indicate the typical phase at which objects in the comparison sample were detected: channels where more than 25% were detected prior to phase -10 are marked as early (squares). If instead more than 95% were detected prior to peak light, the channel is defined as suitable for peak classification (circles). Channels not fulfilling either criteria are marked with triangles. *Left panel:* full channel content. Channels are divided here according to those where transients in galaxies known to host AGNs are cut (black) and channels where these are accepted (gray); cf. Table 3. *Right panel:* comparison of the total number of accepted candidates (y -axis) with the number of comparison sample SNe Ia found, with only candidates linked to a galaxy with known spectroscopic redshift $z < 0.1$. All channels reject transients in host galaxies with known AGNs; cf. Table 4. Three channels further discussed in the main text are highlighted (red circles).

we observe the obvious trend that channels with larger coverage of the comparison sample also accept a larger total number of transients, there is also a variation in the total transient counts between configurations that find the same fraction of the comparison sample. Figure 4 highlights a subset of the channels as particularly interesting. Selection statistics for these channels can be found in Table 3. For comparison objects with a well-defined time of peak light, we also determine the phase relative to peak light at which the transient was accepted into each channel. As an estimate for this we use the time of B -band peak light as determined by a SALT light curve fit, which is carried out for each candidate at the T2 tier (Betoule et al. 2014). This information can be used to study the performance of channels in finding early SNe Ia, which constitute a prime target for many supernova science studies. In Fig. 4 we therefore mark all channels where more than 25% of all SNe Ia were accepted prior to -10 days relative to peak light (“Early detection”). Alternatively, SN Ia cosmology programs often look for a combination of completeness and discovery around light-curve peak to facilitate spectroscopic classification. Channels not fulfilling the Early detection criteria but where more than 95% of all SNe Ia were accepted prior to peak light are therefore marked as “Peak classification”. These two simple examples highlight how reprocessing alert streams (reruns) can be used to optimize transient programs, and to esti-

mate yields that are useful for follow-up proposals, for example. We also find that a 4% fraction of the comparison sample (5 out of 134) were found in galaxies with documented AGNs, suggesting that programs which prioritize supernova completeness cannot reject nuclear transients with active hosts.

With AMPEL we are getting closer to one main goal of future transient astronomy – the immediate robotic follow-up of the most interesting detections. Facilities such as the Las Cumbres Observatory, the Liverpool Telescope, and the Palomar P60 now have the instrumental capabilities for robotic triggers and execution of observations. As the next step towards this we also explored how to select candidates for such automatic programs. Figure 4 (right panel) and Table 4 show channels where only transients in confirmed nearby galaxies are accepted. While total transient and matched SN Ia counts are much reduced here, all remaining transient candidates can be said to be both extragalactic and nearby with high probability, and are thus good candidates for follow-up. Channels such as “16” and “28” can here be expected to automatically detect multiple early SNe Ia each year and still have small total counts (160 and 117 transients accepted, respectively).

Based on this exploration we highlight three channels:
 – Channel 10+59, the union of Channels 10 and 59 and including AGN galaxies, is the channel which accepts the least

Table 3. AMPEL sample channel parameter settings and rerun statistics.

Channel	RealBogus	Detections	History	Image	<i>Gaia</i>	SNe Ia (all)	Detections (all)	Phase(<0) (all)	Phase(<-10) (all)	SNe Ia (no AGN)	Detections (no AGN)	Phase(<0) (no AGN)	Phase(<-10) (no AGN)
11	0.5	4	Multinight	Good	Nominal	125	1857	0.956	0.067	120	1514	0.953	0.047
26	0.3	4	New	Excellent	Nominal	36	293	1.000	0.296	35	257	1.000	0.308
34	0.5	2	Multinight	Excellent	Nominal	128	2479	0.944	0.079	123	1964	0.941	0.059
18	0.5	6	New	All	Nominal	2	34	0.000	0.000	2	28	0.000	0.000
77	0.3	4	Persistent	All	Moderate	131	10 672	0.832	0.000	126	9705	0.824	0.000
51	0.3	4	New	Good	Nominal	39	357	1.000	0.321	37	311	1.000	0.308
57	0.3	4	Persistent	Good	Nominal	130	3078	0.830	0.000	125	2351	0.822	0.000
64	0.3	8	Multinight	Good	Nominal	76	580	0.554	0.000	74	524	0.556	0.000
1	0.3	2	New	Good	Nominal	110	2968	0.987	0.557	106	2562	0.987	0.533
28	0.5	2	New	Excellent	Nominal	95	1833	0.985	0.485	92	1547	0.985	0.462
4	0.5	2	New	Good	Nominal	103	2286	0.986	0.514	99	1909	0.985	0.485
10	0.5	2	Multinight	Good	Nominal								
59	0.3	4	Persistent	Good	Moderate								
10+59						134	11 112	0.926	0.084	129	9973	0.923	0.055

Notes. Columns 2–6 show settings used for parameters in Table 2, Cols. 7–10 show statistics including all targets, and Cols. 11–14 show statistics when excluding AGN-associated candidates. The phase estimates describe the fraction of the matched SNe Ia with a good peak phase estimate that were accepted by the channel either prior to light curve peak or prior to -10 days with respect to peak light.

Table 4. AMPEL sample channel parameter settings and rerun statistics for cases when only transients close to host galaxies of $z < 0.1$ are included.

Channel	RealBogus	Detections	History	Image	<i>Gaia</i>	SNe Ia (all)	Detections (all)	Phase(<0) (all)	Phase(<-10) (all)	SNe Ia (no AGN)	Detections (no AGN)	Phase(<0) (no AGN)	Phase(<-10) (no AGN)
85	0.3	4	Persistent	Good	Nominal	21	261	0.750	0.000	19	143	0.714	0.000
34	0.5	2	Multinight	Excellent	Nominal	20	231	0.867	0.067	18	136	0.846	0.077
16	0.5	2	New	All	Nominal	17	160	0.923	0.385	15	101	0.909	0.273
29	0.5	4	New	Excellent	Nominal	4	9	1.000	0.000	3	6	1.000	0.000
84	0.3	8	Multinight	All	Moderate	11	32	0.333	0.000	10	24	0.375	0.000
28	0.5	2	New	Excellent	Nominal	15	117	0.917	0.333	14	79	0.909	0.273

Notes. Columns 2–6 show settings used for parameters in Table 2, Cols. 7–10 show statistics including all targets, and Cols. 11–14 show the same statistics when excluding AGN-associated candidates. The phase estimates describe the fraction of the matched SNe Ia with a good peak phase estimate that were accepted by the channel either prior to light-curve peak or to -10 days with respect to the peak.

amount of transients while recovering the full comparison sample prior to peak light. We refer to this as the “complete” channel.

- Channel 1 (including AGN galaxies) strikes a balance between a relatively high completeness ($>80\%$) and the early detection of transients and with a limited number of total accepted transients. As is discussed in Sect. 7.1, this channel performs the initial selection for the current automatic candidate submission to TNS and is thus referred to as the TNS channel.
- Channel 16, coupled with only accepting transients in nearby nonAGN host galaxies, provides a very pure selection suitable for automatic follow-up. Consequently, this is referenced as the robotic channel. We add “N” to the channel number (16N) to remind the user that only transients in nearby ($z < 0.1$) galaxies are admitted.

The complete and TNS channels differ mainly in that the former accepts transients closer to *Gaia* sources.

6.2. Channel content and photometric transient classification

The previous section examines channels mainly based on the fraction of a known comparison SN Ia sample which was rediscovered. However, as mentioned, the real number of unclassified supernovae (of all types) will be much larger. Every channel will also contain subsets of all other known astronomical variables (e.g., AGNs, variable stars, and solar-system objects), still-unknown astronomical objects, and noise. This gap between photometric detections and the number of spectroscopically clas-

sified objects will only increase as the number and depth of survey telescopes increase. Developing photometric classification methods is therefore one of the key requisites for the LSST transient program.

The ZTF is different in that most transients are nearby and could be classified and the ZTF stream therefore provides a way to develop classification methods where the predictions can be verified. As a more immediate application we would like to gain a more general understanding of what transients the AMPEL channels produce. As a first step in this process we can use the SN Ia template fits introduced in Sect. 6.1 as a primitive photometric classifier. The fits were carried out using a T2 wrapper to the SMCOSMO package¹⁷. In this case the run configuration only requested the SALT2 SN Ia model to be included, but any transient template could have been requested. During the stream processing a fit will be done to each state, but here we only analyze the final state fit as we are investigating sample content rather than the evolution of classification accuracy with time (the latter question is more interesting but also more complex).

Out of the 11 112 transients accepted by the complete (10 + 59) channel, 6995 have the minimal number of detections (5) required to fit the SALT2 parameters: x_1 (light-curve width), c (light-curve color), t_0 (time of peak light), x_0 (peak magnitude), and z_{phot} (redshift from template fit). Further requiring the central values of the fit parameters to match parameter ranges observed among nearby SNe Ia ($-3 < x_1 < 3$, $-1 < c < 2$, $0.001 < z_{\text{phot}} < 0.2$ and $z_{\text{err}} < 0.1$) leaves 634 transients. In

¹⁷ <https://sncosmo.readthedocs.io>

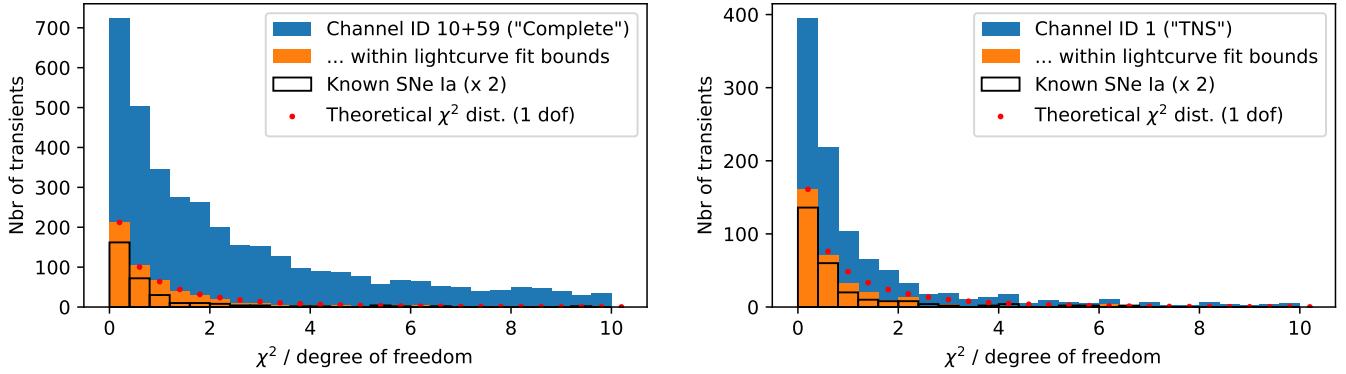


Fig. 5. Histogram of SALT2 SN Ia fit quality (χ^2 per degree of freedom) for the complete 10 + 59 channel. Blue bars show the full sample (with enough detections for fit) while orange shows the subset which also fulfill the expected fit parameter requirements. These are compared with the fit quality for the subset of known SN Ia in the comparison sample (outlined bars, scaled with a factor 2) as well as a standard χ^2 distribution for one degree of freedom (scaled to match the first bin of the restricted sample).

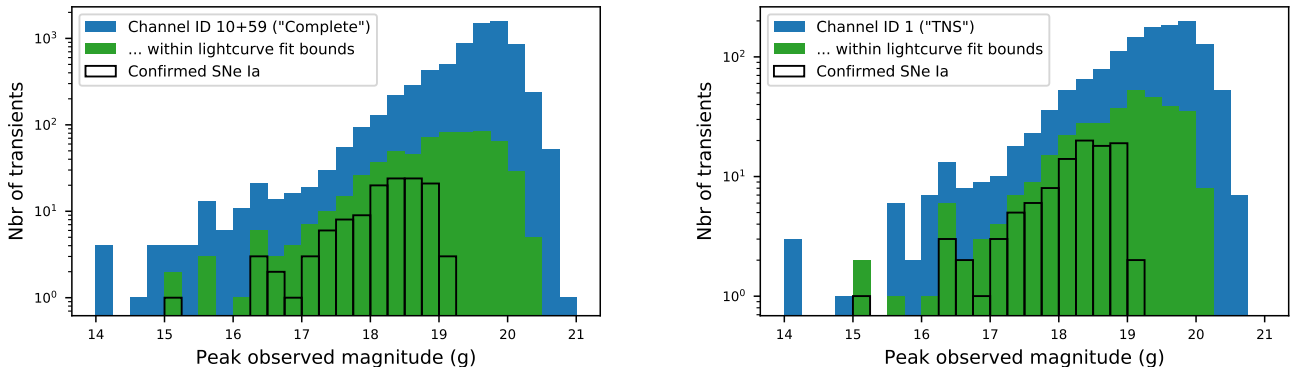


Fig. 6. Peak magnitude distributions (ZTF g band) for the same subsets. The comparison sample is not scaled. *Left panel:* data for the complete 10 + 59 channel. *Right panel:* data for the efficient 1 channel.

Fig. 5 we compare the distributions of χ^2 per degree of freedom for these samples. We find that the subset following typical SN Ia parameters matches both the expected theoretical fit quality distribution and has a distribution similar to the values obtained for the comparison sample of spectroscopically confirmed SNe Ia. This “SN Ia-compatible” subset can therefore be used as an approximate photometric SN Ia sample¹⁸. Repeating this study for the “TNS” channel 1, which accepted 2968 transients, we find that 1342 objects can be fit, and that out of these 349 are compatible with the standard SN Ia parameter expectations.

We next examine the observed peak magnitudes for both the complete and efficient channels (Fig. 6). For both channels, the subsets restricted to standard SN Ia parameter ranges agree well with the comparison objects for bright magnitudes (<18.5 mag). Fainter than this limit, both channels contain a large sample of likely SNe Ia with a detection efficiency that rapidly drops beyond 19.5 mag. Both limits are expected as the ZTF RCF program attempts to classify all extragalactic transients brighter than 18.5 mag, and supernovae peaking fainter than ~ 19.5 mag often do not yield the five significant measurements that are required to trigger the production of an alert and will therefore not be included in the light-curve fit. Most of these fainter SNe will have several late-time observations below the 5σ threshold that

did not trigger alerts but which will be recoverable once the ZTF image data is released. We find no significant differences between the complete and TNS channels in terms of magnitude coverage, consistent with the fact that they differ mainly in that the complete channel accepts transients closer to *Gaia* sources.

We can thus define two (overlapping) subsets for each channel: The comparison sample of known SN Ia (“Reference SN Ia”) and the photometric SNe Ia (“Photo SN Ia”) with light-curve fit parameters compatible with a SN Ia. We complement these with five subsets based on external properties:

- Transients that coincide with an AGN in the Million Quasar Catalog or SDSS QSO catalogs are marked as “Known AGN”.
- Transients that coincide with the core of a photometric SDSS galaxy are marked “SDSS core” (distance less than $1''$).
- Transients that coincide with a SDSS galaxy outside the core are marked “SDSS off-core” (distance larger than $1''$).
- Transients that were reported to the TNS as a likely extragalactic transient but do not have a confirmed classification are marked “TNS AT”.
- Transients that do have a TNS classification but are not part of the reference sample of SNe Ia are marked “TNS SN (other)”

The count and overlap between these groups are shown in Fig. 7. Here we only include transients with a peak brighter than 19.5 mag as the fraction with a light-curve fit falls quickly below this limit (Fig. 6). We can already make several observations based on this crude accounting: For the complete channel these

¹⁸ Any algorithm for evaluating photometric data can similarly be implemented as a T2 unit and applied to the same rerun dataset. Transient models that can be incorporated into SINCOSMO can even use the same T2 unit and only vary run configuration.

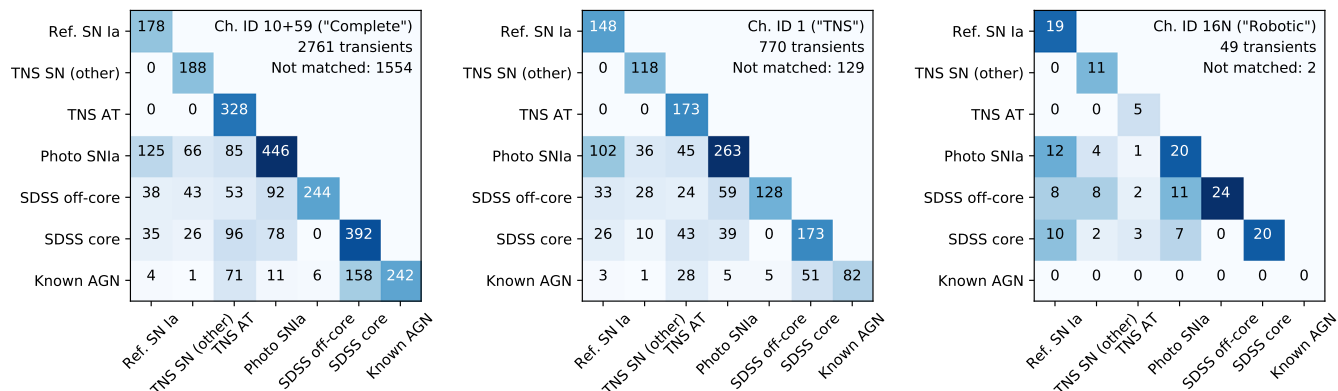


Fig. 7. Estimated transient types for objects with a peak magnitude brighter than 19.5 for the channels 10 + 59 (“complete”), 1 (“efficient”), and 16 (“robotic”). The channel 16 selection also requires transients to be close to host galaxies with a spectroscopic $z < 0.1$ and not in any registered AGN galaxy.

categorizations account for 40% of all accepted transients. The remaining fraction consists of a combination of real extragalactic transients that were not reported to the TNS, stellar variables not listed in *Gaia* DR2, and “noise”. For the efficient channel, only 20% of all detections (152 of 771) are in this sense unaccounted for. We observe that large fractions of SNe are found both aligned with the core of SDSS galaxies as well as without association to a photometric SDSS galaxy. This directly demonstrates why care must be taken when selecting targets for surveys looking for complete samples.

A main goal for transient astronomy, and AMPEL, during the coming decade is to decrease the fraction of unknown transients as much as possible. Machine-learning-based photometric classification will be essential to this endeavor, but other developments are equally critical. These include the possibility to better distinguish image and subtraction noise (“bogus”) and the ability to compare with calibrated catalogs containing previous variability history. We plan to revisit this question once the ZTF data can be investigated for previous or later detections.

6.3. Real-time matching with *IceCube* neutrino detections

The capabilities and flexibility of AMPEL can also be highlighted through the example of the *IceCube* realtime neutrino multi-messenger program. Several years ago, the *IceCube* Neutrino Observatory discovered a diffuse flux of high-energy astrophysical neutrinos (*IceCube Collaboration 2013*). Despite recent evidence identifying a flaring blazar as the first neutrino source (*Aartsen & Ackermann 2018*), the origin of the bulk of the observed diffuse neutrino flux remains undiscovered. One promising approach to identifying these neutrino sources is through multi-messenger programs which explore the possibility of detecting multi-wavelength counterparts to detected neutrinos. Likely high-energy neutrino source classes with an optical counterpart are typically variables or transients emitting on timescales of hours to months; for example core-collapse supernovae, AGNs, or tidal disruption events (*Waxman 1995; Atoyan & Dermer 2001, 2003; Farrar & Gruzinov 2009; Murase & Ioka 2013; Petropoulou et al. 2015; Senno et al. 2016, 2017; Lunardini & Winter 2017; Dai & Fang 2017*). To detect counterparts on these timescales, telescopes that feature a high cadence and a large field-of-view are required in order to cover a significant fraction of the sky. In addition to an optimized volumetric survey speed capable of discovering large numbers of objects, neutrino correlation studies require robustly classified samples of optical transient populations. In order to provide a prompt

response to selected events within large data volumes, a software framework is required that can analyze and combine optical data streams with real-time multi-messenger data streams.

Two complementary strategies to search for optical transients in the vicinity of the neutrino sources are currently active in AMPEL. First, a target-of-opportunity T0 filter selects ZTF alerts which pass image-quality cuts while being spatially and temporally coincident with public *IceCube* high-energy neutrino alerts distributed via GCN notifications. This enables rapid follow-up of potentially interesting counterparts, but is only feasible for the handful of neutrinos that have sufficiently large energy to identify them as having a likely astrophysical origin.

A second program therefore seeks to exploit the more numerous collection of lower-energy astrophysical neutrinos detected by *IceCube* that are hidden among a much larger sample of atmospheric background neutrinos. We therefore created a T2 module which in real-time performs a maximum likelihood calculation of the correlation between incoming alerts and an external database of recent neutrino detections. This calculation is based on both spatial and temporal coincidence as well as the estimated neutrino energy. In particular, the consistency of the light curve with a given transient class, and the consistency of the neutrino arrival times with the emission models expected for that class, enable us to greatly reduce the number of chance coincidences between neutrinos and optical transients. The *IceCube* collaboration is currently using this setup to search for individual neutrinos or neutrino clusters likely to have an astrophysical origin but with insufficient energy to warrant an individual GCN notice.

The neutrino DB is populated by the *IceCube* collaboration in real-time with $O(100)$ neutrinos per day with directional, temporal, and energy information (*Aartsen et al. 2017*). Output is provided as a daily summary of potential matches sent to the *IceCube* Slack.

This program allows a systematic selection of transients to be subsequently followed-up spectroscopically. The final sample will provide a magnitude-limited, complete, typed catalog of all optical transients which are coincident with neutrinos and can be used to probe neutrino emission from a source population.

7. Discussion

7.1. The AMPEL TNS stream for new extragalactic transients

Most astronomers looking for extragalactic transients have similar requests: a candidate feed which is made available as fast

as possible with a large fraction of young supernovae and/or AGNs. By definition, young candidates will not have a lot of detections and the potential gain from photometric classifiers is limited. The efficient TNS channel defined above fulfills these criteria as a large fraction of the comparison sample is recovered while the overall channel count is manageable. Most confirmed SNe Ia were detected more than 10 days before peak, confirming the potential for early detections.

To allow the community fast access to these transients, we use channel ID1 (“TNS”) to automatically submit all ZTF detections from the MSIP program as public astronomical transients to the TNS using senders starting with the identifier ZTF_AMPEL. An example of this process is provided by AT 2019abn (ZTF19aadyppr) in the Messier 51 (Whirlpool Galaxy). This latter object was observed by ZTF at JD 2 458 509.0076 and reported to the TNS by AMPEL slightly more than one hour later.

To make the published candidate stream even more pure, the following additional cuts are made prior to submission. First, we restrict the sample to transients brighter than 19.5 mag (the limit to which the channel content study was carried out). The magnitude depth will be increased once a sufficiently low stellar contamination rate has been confirmed for fainter transients. Figure 8 shows the expected cumulative distributions of peak magnitudes for SNe Ia below different redshift limits as determined by *simsurvey*. A 19.5 mag peak limit implies a $\sim 90\%$ completeness for SNe Ia at $z < 0.08$ based on the expected magnitude distribution. For the volumetric completeness this should be combined with the 80% coverage completeness determined above (which is mainly driven by sky position). We currently only submit candidates found above a Galactic latitude of 14° to reduce contamination by stellar variables. Inspection of the candidates reported so far finds less than 5% to be of likely stellar origin. Candidates compatible with known AGN/QSOs are marked as such in the TNS comment field. Users of the TNS looking for the purest SN stream can therefore disregard any transients with this comment.

Two TNS bots are currently active: ZTF_AMPEL_NEW specifically aims to submit only young candidates with a significant nondetection available within 5 days prior to detection and no history of previous variability. This will create a bias against AGNs with repeated, isolated variability as well as transients with a long, slow rise-time, but further rejects variable stars and provides a quick way to find follow-up targets. A second sender, ZTF_AMPEL_COMPLETE, only requires a nondetection within the previous 30 days¹⁹.

In summary, the live submission of AMPEL detections to the TNS provides a high-quality feed for anyone looking for new, extragalactic transients brighter than 19.5 mag. The contamination by variable stars is estimated to be $< 5\%$, the fraction of SNe to be $> 50\%$, and for SNe Ia with a peak brighter than ~ 18.5 mag the SN Ia completeness is 80%, out of which $\sim 60\%$ will be detected prior to ten days before light-curve peak. Extrapolating rates from the four-(summer)month ZTF rerun would predict this program to submit approximately 9000 astronomical transients to the TNS each year. The breaks due to typical Palomar winter weather makes this an upper limit.

7.2. Work towards an AMPEL testing and rerun environment

The next AMPEL version is already being developed. We plan for this to contain an interface where users can directly upload channel and unit configurations and have them process a stream of

¹⁹ These bots replace the initial ZTF_AMPEL_MSIP sender, which is no longer in use.

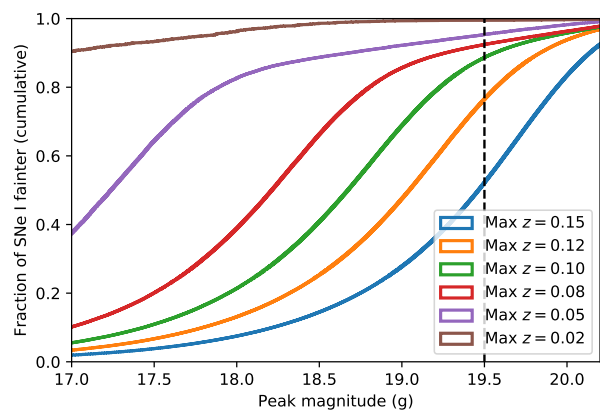


Fig. 8. Cumulative *simsurvey* peak magnitude for simulated data, divided according to max redshift. Dashed lines show the current 19.5 depth of AMPEL TNS submissions.

archived alerts. The container generation means that such a configuration could be automatically spun up in an automatic and secure mode at a computer center. This run environment would allow both increasingly complete tests and more flexibility in carrying out large-scale reruns.

8. Conclusions

Here we introduce AMPEL as a comprehensive tool for working with streams of astronomical data. More and more facilities provide real-time data shaped into streams, which creates opportunities to make new discoveries while emphasizing the challenge in that actions not taken are scientific choices. AMPEL includes tools for brokering (distributing), analyzing, selecting, and reacting to transients. Users contribute channels, which regulate how transients are processed at four internal tiers. The implementation was guided by our suggestions for how to embrace these new opportunities and face the related challenges for transient analysis:

- Provenance and reproducibility are guaranteed by the combination of information stored in a permanent database, containerized software, and an alert archive in a system designed to allow autonomous analysis chains.
- A modular system provides analysis flexibility, and introduces a method for developers to allow software distribution and referencing.
- The combination of these two capabilities allows users to track the impact of versions of both data and software.
- Finally, the database has been designed to manage the alert rates expected from surveys such as LSST.

The fundamental system design allowing us to achieve these goals was the division of the alert processing into four tiers and the recognition that each transient is connected to a growing set of states, each of which consists of a specified set of datapoints. A transient view collects the information of a transient available at a given time.

We presented three sample uses of AMPEL. We first used a reprocessing of alerts from the first four months of ZTF operations to create a “recipe book” of filter definitions with defined acceptance and completeness rates. As part of this study, we show that ZTF detected and issued alerts for all SNe Ia reported to the TNS, and that AMPEL can operate at the high data rates expected for LSST. Three channels were highlighted: A “complete” channel recovering all known SN Ia with a comparably small total count, a “TNS” channel which allows SNe to be detected early and efficiently, and a small “robotic” channel

which can serve as a starting point for automatic follow-up observations. Channel/program distinctions along these lines will become natural as astronomers tap into future large transient flows. We subsequently took a first step in identifying the content of these three channels. For the complete channel, the fraction of real extragalactic transients is estimated to be larger than 40%; for the TNS channel this is above above 80%. The robotic channel is designed to retain only target transients in known nearby galaxies. We plan to continue reprocessing alerts with refined analysis units, improved photometry, and larger alert sets. As a third example, we introduce the live correlation analysis between optical ZTF alerts and candidate extragalactic neutrinos from IceCube, where a T2 unit calculates test statistics between all potential matches and selects targets for spectroscopic follow-up. This methodology can be directly applied to other kinds of multi-messenger studies.

The AMPEL live instance processes the ZTF alert stream and anyone can become a user through creating a channel following the guidelines available at the `AmpelProject/Ampel-contrib-sample` github repository. However, as many astronomers are interested in similar objects, AMPEL also provides a more immediate avenue to likely young extragalactic transients through a real-time propagation of high-quality candidates to the TNS. The chosen channel configuration (“TNS”, ID1) was shown to detect ~80% of the SNe in the comparison sample, with more than 50% detected prior to the phase -10 days (10 days before peak light-curve). This setup is expected to provide O(1000) astronomical transients each year.

Acknowledgements. Based on observations obtained with the Samuel Oschin Telescope 48-inch and the 60-inch Telescope at the Palomar Observatory as part of the Zwicky Transient Facility project. ZTF is supported by the National Science Foundation under Grant No. AST-1440341 and a collaboration including Caltech, IPAC, the Weizmann Institute for Science, the Oskar Klein Center at Stockholm University, the University of Maryland, the University of Washington, Deutsches Elektronen-Synchrotron and Humboldt University, Los Alamos National Laboratories, the TANGO Consortium of Taiwan, the University of Wisconsin at Milwaukee, and Lawrence Berkeley National Laboratories. Operations are conducted by COO, IPAC, and UW. The authors are grateful to the IceCube Collaboration for providing the neutrino dataset and supporting its use with AMPEL. N.M. acknowledges the support of the Helmholtz Einstein International Berlin Research School in Data Science (HEIBRiDS), Deutsches Elektronensynchrotron (DESY), and Humboldt-Universität zu Berlin. M.R. acknowledges the support from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation program (grant agreement no. 759194 – USNAC).

References

- Aartsen, M. G., Ackermann, M., Adams, J., et al. 2017, *Astropart. Phys.*, **92**, 30
 Abbott, B. P., Abbott, R., Abbott, T. D., et al. 2017, *ApJ*, **848**, L12
 Allen, G., Anderson, W., Blaufuss, E., et al. 2018, ArXiv e-prints [arXiv:1807.04780]
 Astropy Collaboration (Robitaille, T. P., et al.) 2013, *A&A*, **558**, A33
 Atoyan, A., & Dermer, C. D. 2001, *Phys. Rev. Lett.*, **87**, 221102
 Atoyan, A. M., & Dermer, C. D. 2003, *ApJ*, **586**, 79
 Bellm, E. C., Kulkarni, S. R., Graham, M. J., et al. 2019, *PASP*, **131**, 018002
 Betoule, M., Kessler, R., Guy, J., et al. 2014, *A&A*, **568**, A22
 Dai, L., & Fang, K. 2017, *MNRAS*, **469**, 1354
 Drake, A. J., Djorgovski, S. G., Mahabal, A., et al. 2009, *ApJ*, **696**, 870
 Farrar, G. R., & Gruzinov, A. 2009, *ApJ*, **693**, 329
 Feindt, U., Nordin, J., Rigault, M., et al. 2019, *J. Cosmol. Astropart. Phys.*, **2019**, 005
 Flesch, E. W. 2015, *Publ. Astron. Soc. Aust.*, **32**, e010
 Gaia Collaboration (Brown, A. G. A., et al.) 2018, *A&A*, **616**, A1
 Graham, M. J., Kulkarni, S. R., Bellm, E. C., et al. 2019, *PASP*, **131**, 078001
 IceCube Collaboration. 2013, *Science*, **342**, 1242856
 IceCube Collaboration (Aartsen, M. G., et al.) 2018, *Science*, **361**, eaat1378
 Jones, E., Oliphant, T., Peterson, P., et al. 2001, *SciPy: Open Source Scientific Tools for Python*, Online: accessed 2019-01-22
 Jurić, M., Kantor, J., Lim, K. T., et al. 2017, in *Astronomical Data Analysis Software and Systems XXV*, eds. N. P. F. Lorente, K. Shorridge, & R. Wayth, *ASP Conf. Ser.*, **512**, 279
 Kasliwal, M. M., Cannella, C., Bagdasaryan, A., et al. 2019, *PASP*, **131**, 038003
 Kurtzer, G. M., Sochat, V., & Bauer, M. W. 2017, *PLoS One*, **12**, 1
 Lunardini, C., & Winter, W. 2017, *Phys. Rev. D*, **95**, 123001
 Mahabal, A. A., Djorgovski, S. G., Drake, A. J., et al. 2011, *Bull. Astron. Soc. India*, **39**, 387
 Mahabal, A., Rebbapragada, U., Walters, R., et al. 2019, *PASP*, **131**, 038002
 Masci, F. J., Laher, R. R., Rusholme, B., et al. 2019, *PASP*, **131**, 018003
 Merkel, D. 2014, *Linux J.*, **2014**
 Murase, K., & Ioka, K. 2013, *Phys. Rev. Lett.*, **111**, 121102
 Oliphant, T. 2006, *NumPy: A guide to NumPy* (USA: Trelgol Publishing), Online: accessed 2019-01-22
 Pâris, I., Petitjean, P., Ross, N. P., et al. 2017, *A&A*, **597**, A79
 Patterson, M. T., Bellm, E. C., Rusholme, B., et al. 2019, *PASP*, **131**, 018001
 Petropoulou, M., Dimitrakoudis, S., Padovani, P., Mastichiadis, A., & Resconi, E. 2015, *MNRAS*, **448**, 2412
 Price-Whelan, A. M., Sipőcz, B. M., Günther, H. M., et al. 2018, *AJ*, **156**, 123
 Saha, A., Matheson, T., Snodgrass, R., et al. 2014, in *Observatory Operations: Strategies, Processes, and Systems V*, *SPIE Conf. Ser.*, **9149**, 914908
 Sanguillon, M., Servillat, M., Louys, M., et al. 2017, in *Astronomical Data Analysis Software and Systems XXV*, eds. N. P. F. Lorente, K. Shorridge, R. Wayth, et al., *ASP Conf. Ser.*, **512**, 581
 Senno, N., Murase, K., & Mészáros, P. 2016, *Phys. Rev. D*, **93**, 083003
 Senno, N., Murase, K., & Mészáros, P. 2017, *ApJ*, **838**, 3
 Smith, K. W., Williams, R. D., Young, D. R., et al. 2019, *Res. Notes AAS*, **3**, 26
 Smith, M. W. E., Fox, D. B., Cowen, D. F., et al. 2013, *Astropart. Phys.*, **45**, 56
 Soumagnac, M. T., & Ofek, E. O. 2018, *PASP*, **130**, 075002
 Tachibana, Y., & Miller, A. A. 2018, *PASP*, **130**, 128001
 Waxman, E. 1995, *Phys. Rev. Lett.*, **75**, 386
 Williams, R. D., Djorgovski, S. G., Drake, A. J., Graham, M. J., & Mahabal, A. 2009, in *Astronomical Data Analysis Software and Systems XVIII*, eds. D. A. Bohlender, D. Durand, & P. Dowler, *ASP Conf. Ser.*, **411**, 115

Appendix A: Creating the TNS comparison sample

The comparison sample that is used to estimate channel efficiencies was constructed through retrieving all TNS SNe classified as Type Ia supernovae (not including peculiar subtypes) and with a detection date between June 5 and September 15, 2018. This was further restricted to SNe above an absolute Galactic latitude of 14°. This leaves 310 objects shown in Table A.1. Out of these, 20 have positions outside the ZTF MSIP primary field grid, and 8 were projected to land in gaps between ZTF CCDs or within the 1% of chip pixels closest to a readout edge.

As ZTF field references have been continuously produced during the first season of operations we also verify that subtractions were made at least 3 days prior to the SN detection. For 89 SNe no references were available while for 58 SNe a reference was only available in either *g* or *r* band. One TNS object included in this list, SN 2018ekt, was, as part of this study, found to have been erroneously classified (and has therefore now been removed). Excluding this leaves a main comparison sample of 134 SNe Ia that were observed by ZTF in the nominal ZTF MSIP cadence. Among SNe only found in one band, SN 2018fvh is located on bad pixels, SN 2018cmu is only detected in a single alert (one detection), and SN 2018cmk was detected by ZTF but more than 3 arcsec from the reported TNS position. Table A.1 is only available at the CDS.