

Suborbital and Orbital Real-Time Localization of Gamma-Ray Bursts via Utility-Guided Coordination

Daisy Wang, Ye Htet, Jeremy Buhler, *Senior Member, IEEE*, Filip Marković, and Marion Sudvarg, *Member, IEEE*

Abstract—Transient astrophysical phenomena such as gamma-ray bursts (GRBs) benefit from multi-instrument, multi-wavelength observation. A wide-field, high-energy telescope first constructs a spatial probability distribution, or likelihood map, from observed gamma-ray photons. A narrow-field optical/UV telescope then uses that map to search for the transient. Coordinating observations between telescopes is a cyber-physical design problem shaped by (a) sensing uncertainty, because observed events include both informative source photons and unrelated background particles; (b) strict timeliness requirements, because the observation window for a transient may last only tens to hundreds of seconds; and (c) on-board embedded computing constraints, because computation executes aboard the telescopes under strict size, weight, and power (SWaP) limits, rather than utilizing costly telescope-to-ground communication that undermines timeliness.

The overall deadline to observe a transient is consumed by two sequential stages: data gathering and likelihood mapping, followed by optical search. Choosing when to stop collecting data, generate the map, and send it to the optical telescope is therefore critical, since the probability of success depends jointly on the three dimensions above: sensing uncertainty, timeliness, and on-board SWaP-constrained computation.

This work builds an integrated on-board pipeline for gamma-ray transient mapping and follow-up search on a SWaP-constrained ARM multicore platform. We introduce a formally grounded, empirically trained, utility-driven method for deciding when to generate the likelihood map and begin the search. We validate both the pipeline and the value of utility maximization using physical simulation of GRB detection with the balloon-borne ADAPT telescope.

Index Terms—embedded systems, gamma-ray bursts, probabilistic scheduling, real-time systems, telescope scheduling

I. INTRODUCTION

Scientific studies of high-energy transient astrophysical phenomena (TAPs), such as gamma-ray bursts, benefit from

coordinated observation using telescopes that are sensitive to distinct electromagnetic wavelength bands. The U.S. National Academies’ Astro2020 decadal survey [1] recognized the scientific importance of such coordinated observation. Typically, a high-energy telescope that can observe much or all of the sky at once detects and localizes a gamma-ray transient, then alerts a cooperating *partner instrument* with a much narrower field of view (FoV) to search near the transient’s location in the sky in hopes of seeing its counterpart in an optical band.

Cooperative observation of TAPs with multiple telescopes requires coordination under time constraints. The gamma-ray signals from a TAP appear and fade within seconds to minutes, while some counterpart signals associated with stages of the TAP’s evolution may appear and fade tens to hundreds of seconds later. The time from when a gamma-ray transient is detected to when a sought counterpart fades – determined by the physics of the TAP – imposes a natural deadline for detection, during which several steps must be completed: (1) the high-energy telescope must gather enough gamma-ray light from the TAP to infer its location in the sky; (2) this telescope must compute a *likelihood map*, which is a probability distribution on the sky describing the TAP’s likely location; (3) the narrow-FoV partner telescope must segment the sky into *tiles*, sized according to its FoV, and plan a sequence of tiles to search for the TAP based on the likelihood map; (4) the partner must actually conduct the search, imaging each tile until the TAP is detected or the deadline passes.

TAP observation raises real-time system design challenges induced by the need to partition the available time before deadline among the steps in the process. The first step, gathering data from the TAP, is elastic: less time spent gathering data allows mapping, planning, and search to begin sooner but reduces the map’s accuracy, which in turn may require that the partner spend more time searching a larger area of the sky. It is therefore essential to quantify the end-to-end *utility*, i.e., probability of successfully imaging the TAP, associated with time spent gathering data.

Further complexity arises because TAP observation is subject to two kinds of uncertainty. First, the high-energy telescope observes signals both from the TAP and from unrelated *background* particles, and these types of signal cannot *a priori* be distinguished. It is therefore uncertain how much signal has been gathered from the TAP. Second, the algorithms for mapping and search planning take varying amounts of time and produce results of varying quality. While compute time

Manuscript received 30 March 2026; accepted 17 July 2026. Date of current version 17 August 2026. This work was supported by NASA award 80NSSC21K1741. This article was presented at the International Conference on Embedded Software (EMSOFT) 2026 and the ESWEEK-TCAD special issue.

Daisy Wang, Ye Htet, and Jeremy Buhler are with the Department of Computer Science and Engineering, Washington University in St. Louis, St. Louis, MO 63130 USA (e-mail: {w.yanwang, htet.ye, jbhuhler}@wustl.edu).

Filip Marković is with the School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, U.K. (e-mail: f.markovic@soton.ac.uk).

Marion Sudvarg is with the Department of Computer Science, Missouri University of Science and Technology, Rolla, MO 65409 USA (e-mail: msudvarg@mst.edu).

This article has supplementary downloadable material available at <https://doi.org/10.5281/zenodo.21572120>, provided by the authors. Digital Object Identifier 10.5281/zenodo.21572120

and result quality depend on the amount of gathered source and background signal, these algorithmic measures are stochastic functions that may be different for every TAP. Both kinds of uncertainty must be factored into decisions about how to partition time among the stages of observation.

As a final complication, the high-energy telescope, and often the partner as well, is deployed on a high-altitude suborbital balloon [2] or an orbiting satellite [3], [4]. While these telescopes could delegate mapping and search planning to a terrestrial computing facility, the low bandwidth and long latency of telemetry to the ground argue for performing these tasks in embedded computing platforms aboard the instruments. However, doing so imposes size, weight, and power (SWaP) constraints on the hardware available for computation. Prior work [5], [6] suggests that mapping and search planning can be done efficiently under such constraints, but we are unaware of prior system designs for TAP observation that both accommodate on-board resource limitations and address the challenges of coordination under deadline and uncertainty.

In this work, we describe and characterize the performance of a real-time computational pipeline for cooperative TAP imaging that runs on SWaP-constrained embedded hardware suitable for on-board use. We use a detailed physical model of a soon-to-be-launched balloon-borne gamma-ray telescope, the Antarctic Demonstrator for the Advanced Particle-physics Telescope (ADAPT) [2], as the basis for our study. Our contributions include integration of mapping and planning methods that run efficiently on a low-power embedded ARM multicore, as well as development of an instrument response model for ADAPT that drives the mapping algorithm. Critically, we develop a probabilistic, empirically trained utility estimation framework to predict how time spent gathering events will impact the probability of successful TAP observation. We validate our work on simulated gamma-ray bursts generated using physical simulations of the ADAPT instrument.

In broader terms, the problem space of this work generalizes beyond TAP observation to utility-aware resource allocation in embedded real-time systems. A SWaP-constrained on-board platform must divide a fixed end-to-end deadline among uncertainty-prone sensing, computation, planning, and downstream tasks, where both the latency and quality of intermediate computations affect the final utility-aware quality of service (QoS). This induces a deadline-budget allocation problem: spending more time in one stage may improve local information quality but necessarily reduces the time available to subsequent tasks. This abstraction connects cooperative TAP observation to utility-accrual real-time scheduling [7], dynamic-deadline adaptation in autonomous driving [8], and real-time perception systems that trade execution time against accuracy [9]. Thus, this paper provides a concrete on-board setting for studying utility-aware scheduling under uncertainty-aware sensing and embedded computation.

II. BACKGROUND

This section further describes the key computational tasks in TAP observation and motivates our approach.

A. Cooperatively Observing Gamma-Ray Transients

TAPs arise from violent events in the distant universe, including supernovae, binary neutron-star and black-hole mergers, and magnetar giant flares. These phenomena, which appear at unpredictable locations throughout the sky, produce gamma-ray signals lasting from a fraction of a second to several minutes. Here, we focus on short-duration transients in the MeV energy band, which are typically visible in gamma rays for at most tens of seconds.

A TAP may produce electromagnetic signals not only in gamma rays but also at longer wavelengths (e.g., optical light). These signals arise from phenomena later in the TAP's evolution, such as supernova shock breakout [10], and so become visible and then fade tens to several hundred seconds after the gamma-ray signal. The longer-wavelength light includes color and spectral information that reveals the TAP's underlying physics, so we wish to capture this light before it fades.

MeV-range gamma-ray detectors are flown high in the atmosphere or in orbit above it to minimize atmospheric blocking of TAP gamma rays. Prior orbital gamma-ray observatories include the Compton Gamma-Ray Observatory (CGRO) [11], the Neil Gehrels Swift observatory [3], and Fermi [12], while planned missions include the suborbital balloon-borne ADAPT [2] and ComPair-2 [13] instruments and the orbital COSI [4]. These detectors' FoVs each cover a large portion of the sky but are restricted to the gamma-ray band. They must therefore cooperate with longer-wavelength instruments, either terrestrial – e.g., 7DT [14] – or co-deployed in flight – e.g., Swift's Ultra-Violet Optical Telescope (UVOT) [15] or ADAPT's optical piggyback, AIRIS [16] – to perform follow-up observations. These partner instruments have narrow FoVs from 1 to 10° wide, so they rely on the gamma-ray detector to localize the TAP to within a small patch of sky in order to image it before its light fades.

B. Computational Tasks in TAP Observation

Figure 1 gives an overview of the pipeline of computations involved in cooperative TAP observation. This pipeline involves two main tasks: *likelihood mapping* (see Figure 1_{II}), which is performed by the gamma-ray instrument, and *follow-up search* (see Figure 1_{IV}), which is performed by the partner.

a) Likelihood Mapping: Mapping determines where in the sky a TAP has occurred by analyzing *detector events* caused by individual gamma rays. As shown in Figure 1_I, the sky is discretized into a number of equally-spaced source directions $\vec{s}$ using a HEALPix grid [17]. After observing a set of events V , mapping assigns to each $\vec{s}$ a probability $\Pr(V \mid \vec{s})$ that the events in V can be explained by a point source (the TAP) originating from $\vec{s}$.

For MeV transients, observed events are *Compton rings* inferred from interactions of particles with the detector [18]. Each event is characterized by its energy and the geometry (scattering angles) of its interactions. In addition to events caused by the TAP's gamma rays, the detector observes unrelated events from, e.g., atmospheric scattering and solar wind particles arriving from all directions. The telescope cannot a

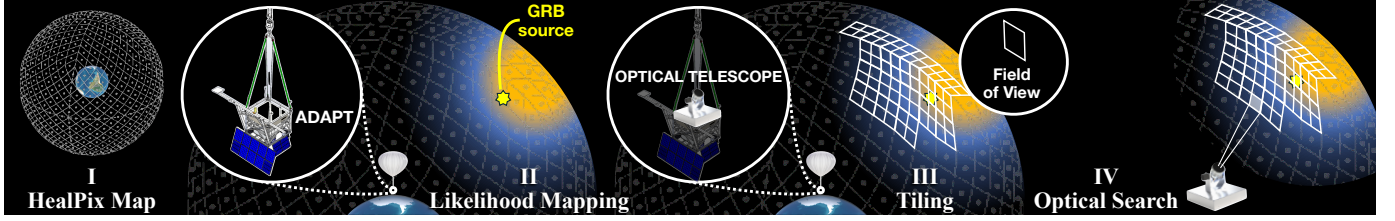


Fig. 1. Computational flow of TAP observation. I: discretization of possible TAP source directions; II: assignment of probability to each source direction (i.e., a likelihood map) using TAP gamma rays; III: tiling of the high-probability region of the sky by the partner instrument and prioritization of tiles for search; IV: search for the transient within each tile.

priori distinguish between *source events* from the TAP and unrelated *background events*; both are present in V .

Likelihood mapping [19], [20] uses forward simulation involving two models: an *instrument response* model R that describes how gamma rays from direction $\vec{s}$ produce source events, and a *background model* B that describes the expected pattern of background events. These models are precomputed through physical simulation. For each candidate source direction $\vec{s}$, we evaluate $\Pr(V | \vec{s}, R, B)$, in the process estimating the fraction of V that is attributable to source vs. background. For moderately bright TAPs observed by current instruments with typical levels of background, the maximum-likelihood estimate of $\vec{s}$ lies within 5-20 degrees of the true source.

b) *Follow-up Search*: Because a likelihood map provides only a probability distribution over the source direction $\vec{s}$, a partner instrument that wishes to image the TAP must search for it. The partner's limited FoV induces a division of the sky into tiles, each the size of one field, as shown in Figure 1III. To image a particular tile, the partner must *slew* (move its aperture) to point to the tile, then *dwell* there gathering light to form an image. If the acquired image does not contain a visible TAP, the partner can then slew to a different location and try again, and so forth until either the TAP is found or enough time has elapsed that it has likely faded.

Follow-up search is characterized by a model of the partner instrument (FoV, slew speed, dwell time required to form an image in a given direction) and a *deadline* D by which the TAP must be imaged before it fades below the limit of detection. Given the likelihood map, the partner must construct and execute a *search plan* specifying a sequence of tiles $\tau_1, \tau_2, \dots$ to image. Its goal is to maximize the probability that the TAP will be imaged in at least one of these tiles prior to D .

Search planning algorithms traditionally rely on general optimization frameworks such as integer linear programming [21]. However, recent work has described much faster yet still effective heuristic approaches [6].

C. The Need for On-Board Computation

Gamma-ray telescopes today communicate raw event observations to the ground, where mapping and search planning happen, after which the plan is communicated to the partner. This is the case even when the cooperating telescopes are physically co-located, as for, e.g., Swift BAT/UVOT. Telescope-to-ground communication suffers from many logistical challenges in practice [22]. Bandwidth is severely limited except within

certain time windows (e.g., when a satellite passes over its ground station); outside these windows, bandwidth may be only 10^4 – 10^5 bits per second. Latency for ground communication and data handling can be tens of seconds. Hence, performing mapping and search planning on the ground either limits these tasks to use only a fraction of observed events in near-real-time or may delay them by several hours – longer than the time-scale of optical phenomena being sought. Moreover, as future instruments become more sensitive to transients, they will need to perform follow-up observation more frequently. Particularly for co-deployed telescopes in orbit, this work should ideally be autonomous without terrestrial involvement. Hence, *we seek to move the computational work of mapping and search planning out of terrestrial data centers and onto the telescopes themselves*.

While on-board, real-time coordination between instruments is a desirable capability, prior work has not demonstrated its feasibility given the SWaP constraints of embedded computing hardware that can be flown. Existing software is designed for a terrestrial server or cluster, and even these tools may not be optimized for prompt response times. For example, we found that the likelihood mapping implementation in the April 2025 release of the `cosipy` library [23] required 2-3 minutes per TAP on eight Intel Xeon Gold 5118 CPUs. In contrast, this work integrates implementations of mapping and search planning software that routinely achieve sub-second latency, even on a low-power ARM multicore.

D. Real-Time Coordination under Uncertainty

TAP discovery and imaging is a cooperative real-time process, in that both instruments must execute their respective computations within an overall deadline D dictated by the TAP's physics. They must therefore partition the time before deadline so as to maximize *utility* – in this case, the chance of successfully imaging the TAP. Mapping and planning, as we will show, occupy little time relative to D , so most time is split between gathering events to construct a likelihood map, and using that map to search a series of tiles for the TAP.

Figure 2 illustrates the core dilemma of coordinating mapping and search. A high-energy telescope may choose to gather events for some number of seconds (1.5–7.5s in the left panel) before mapping. Mapping at 1.5s, well before the TAP's gamma-ray signals end (option A), captures too few of its events and so yields an inaccurate map with a large probable area, which requires more tiles and hence more time

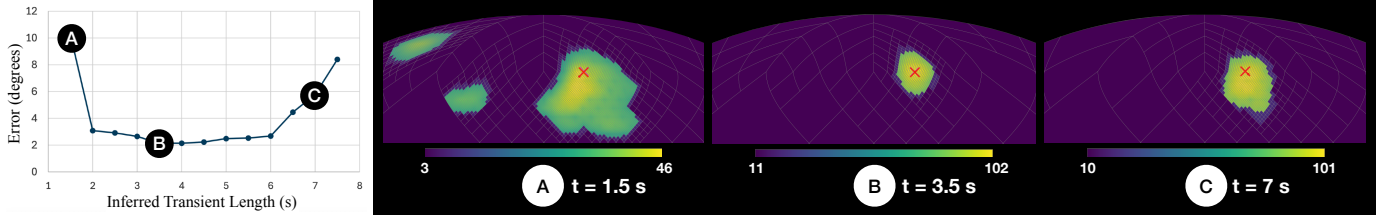


Fig. 2. Maps produced at different times after detecting a TAP. *Left*: error in estimated source direction versus time at which data gathering stops. This TAP’s signals end after ~ 3 seconds, but background events continue to accumulate. *Right*: Maps produced 1.5, 3.5, and 7 seconds after TAP detection. Color scale represents likelihood score, proportional to probability, for each source direction; crosses show true TAP location.

to search. Delaying mapping until 3.5s (option B) yields a much more compact, accurate map that requires less search time but also leaves less time remaining to enact the search. Waiting even longer (option C) not only takes time away from search but also incorporates more background events into the map, which actually makes it larger and less accurate.

This example also highlights uncertainties in cooperative TAP observation. The end of the TAP at ~ 3 s is not known *a priori*, and it cannot easily be detected amid the continuous stream of background events. We have overall knowledge of the background arrival process – specifically, that it is a Poisson process with some average intensity λ per second, which is estimated empirically to a high degree of confidence by the gamma-ray telescope from observations in the minutes leading up to the TAP. However, during the TAP itself, only the *total* number of events seen up to a given time is known, not the numbers of source and background events. Moreover, the mapping algorithm’s accuracy and running time, and the partner’s planning quality and running time, vary stochastically from TAP to TAP even when the numbers of source and background events are held constant. Predicting the probability of successful TAP imaging must account for these uncertainties.

In order to develop a robust cooperative observation framework, then, we first address the following problem. Let E_t be the total number of events observed at time t after the TAP is detected. We have $E_t = S_t + B_t$, where S_t and B_t are the (unknown) numbers of source and background events, respectively. We are given a probabilistic model as above describing the distribution of B_t . We are also given empirical estimates for (1) the times needed to generate a map from s source and b background events and to use this map to construct a search plan, and (2) the probability that, given this map, the partner can successfully image the TAP within d seconds. For an overall deadline D , *how much time t should we spend gathering events before constructing a map so as to maximize the probability of successful TAP imaging by time D ?* We describe our solution to this problem in Section III.

III. MAXIMIZING UTILITY OF COOPERATIVE SEARCH

To support informed runtime decisions about when to generate a likelihood map so as to maximize the probability of successful TAP imaging, we use physical simulations to model the impact of source and background event counts on the

likelihood-mapping and planning algorithms and, ultimately, on the partner’s search performance.

We fix: (i) a map-generation algorithm $\mathcal{M}$ and (ii) a search planning algorithm $\mathcal{A}$ parameterized by the properties of the partner telescope. We then model a stochastic simulation space induced by randomly generated TAPs and latent system execution uncertainty.

Let $(\Omega, \mathcal{F}, \Pr)$ be a probability space, where Ω is the set of all possible simulation outcomes and $\omega \in \Omega$ denotes a single outcome (a single TAP and its mapping and attempted imaging). Further, $\mathcal{F} \subseteq 2^\Omega$ is the σ -algebra of events, and $\Pr : \mathcal{F} \rightarrow [0, 1]$ is the associated probability measure that maps each event to the corresponding probability.

For a single simulation outcome $\omega \in \Omega$, let $d \in \mathbb{R}$ be the residual search time budget remaining after map generation and search planning have completed, as shown in Figure 3a. Then, the end-to-end deterministic success indicator $H : \mathbb{R} \times \Omega \rightarrow \{0, 1\}$ is defined by

$$H(d, \omega) \triangleq \begin{cases} 1, & \text{if } d \geq 0 \text{ and search succeeds within } d, \\ 0, & \text{otherwise.} \end{cases}$$

Successful localization depends strongly on the composition of the events accumulated up to the instant at which map generation is initiated. Accordingly, for each simulation outcome $\omega \in \Omega$, let $S(\omega)$, $B(\omega) \in \mathbb{N}_0$ denote, respectively, the numbers of source and background events observed, and define the total event count by $E(\omega) \triangleq S(\omega) + B(\omega)$. After simulating a single outcome $\omega \in \Omega$, this decomposition is known, so $S(\omega)$, $B(\omega)$, and $E(\omega)$ are all available. At runtime, however, only the total event count can be observed, whereas the source-background split remains unknown.

A. Exact Utility Calculation

For each admissible triple (e, b, d) , where $e \in \{0, \dots, e_{\max}\}$ and $e_{\max}$ is the maximum observable event count, $b \in \{0, \dots, e\}$, and $d \in \mathbb{R}$, we define the probability of successful TAP localization within search budget d , conditioned on total event count e and background event count b . For this purpose, first let $\Omega_{e,b} \triangleq \{\omega \in \Omega : E(\omega) = e \wedge B(\omega) = b\}$ be the measurable subset of simulation outcomes with count pair (e, b) . Similarly, let $A_d \triangleq \{\omega \in \Omega : H(d, \omega) = 1\}$ be the measurable subset of simulation outcomes in which localization succeeds within search budget d . Provided that $\Pr[\Omega_{e,b}] > 0$, the corresponding conditional success probability is defined as

$$U(e, b, d) \triangleq \Pr[A_d \mid \Omega_{e,b}].$$

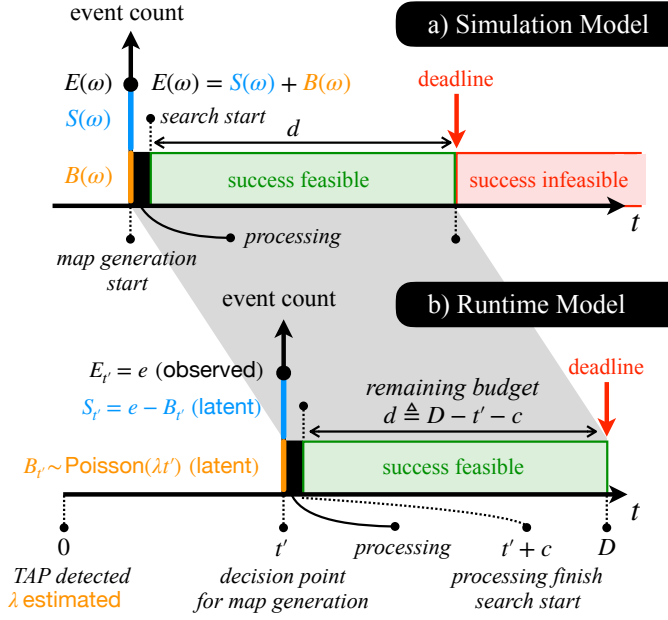


Fig. 3. Visualization of the simulation and the runtime model.

Note that once the total count e and the background count b are fixed, the source count is necessarily $e - b$. Thus, $U(e, b, d)$ can be expressed without introducing any probability model for the source count S , i.e., $\Omega_{e,b} = \{\omega \in \Omega : E(\omega) = e \wedge B(\omega) = b\} = \{\omega \in \Omega : S(\omega) = e - b \wedge B(\omega) = b\}$.

At runtime (see Figure 3b), the system may initiate likelihood-map generation at any decision time $t' \in [0, D)$, where D denotes the deadline for successfully imaging the TAP, measured starting from the instant of TAP detection. At such a decision time, the total event count $E_{t'} = e$ is observable, whereas the corresponding background count $B_{t'}$ is not. As discussed in Section II, we model background arrivals as a Poisson process with known intensity λ , estimated from background observations available at the moment of TAP detection (i.e., at time 0). Hence, the accumulated background count at time t' has mean $\lambda t'$ and satisfies

$$B_{t'} \sim \text{Poisson}(\lambda t'), \Pr[B_{t'} = b] = \exp(-\lambda t') \frac{(\lambda t')^b}{b!}, b \in \mathbb{N}_0.$$

Given the observed value $E_{t'} = e$, we use this information to restrict the possible background counts to $b \in \{0, \dots, e\}$. Since $E_{t'} = S_{t'} + B_{t'}$, the feasible values of the latent background count are $b \in \{0, \dots, e\}$. We use the Poisson model for $B_{t'}$ to assign probabilities to these feasible values.

Before stating the formal definition of the runtime expected utility, it remains to account for the time consumed by computational work prior to searching for the transient. This *processing time* includes both time spent generating a likelihood map from the data present at time t' and time spent planning the order in which to search tiles given this map. For a fixed processing time c , the residual search budget is $d \triangleq D - t' - c$. This value may be negative if map generation finishes after the overall deadline, and by the definition of H , such cases have zero utility.

Finally, we define runtime expected utility as follows.

TABLE I
OVERVIEW OF NOTATION.

Symbol	Explanation
D	Relative deadline for successfully imaging the TAP, measured from TAP detection.
$t' \in [0, D)$	Runtime decision point at which likelihood-map generation may be initiated.
c	Processing time consumed by likelihood-map generation and search planning.
$d \triangleq D - t' - c$	Residual search budget after processing.
b	Latent background event count.
s	Latent source event count.
$e \triangleq s + b$	Total event count (observable).
$U(e, b, d)$	Conditional success probability for count pair (e, b) and residual budget d .
$B_{t'}$	Background event count accumulated by t' .
λ	Estimated background-event-count intensity.
$\tilde{U}(e, t', c; \lambda)$	Runtime expected utility at time t' .
$K(e, b, d)$	Number of successful sampled outcomes for count pair (e, b) and residual budget d .
$\underline{U}_q(e, b, d)$	One-sided Clopper-Pearson q -confidence lower bound on $U(e, b, d)$.
$g_q(e, b), \ell_q(e, b)$	Empirical q -quantile estimates of map-generation and planning time.

Definition 1: Let $t' \in [0, D)$ be the decision time point for map generation and e be the total event count observed at t' . Let c be the fixed processing time and λ be the estimated intensity of background events at t' . Then the runtime expected utility $\tilde{U}(e, t', c; \lambda)$ conditioned on the event of feasible background counts $\mathbb{G}_e \triangleq \{\omega \in \Omega : B_{t'}(\omega) \leq e\}$ is defined as:

$$\tilde{U}(e, t', c; \lambda) \triangleq \sum_{b=0}^e U(e, b, d) \Pr\{B_{t'} = b \mid \mathbb{G}_e\}.$$

In the equation above, we assume that the required conditional utilities are well-defined for all feasible background counts included in the runtime sum, that is, $\Pr[\Omega_{e,b}] > 0$ for all $b \in \{0, \dots, e\}$. The empirical estimation procedure in the next subsection explains how finite-sample and missing-support cases are handled in practice.

Expectation $\tilde{U}(e, t', c; \lambda)$ is the basis of our utility measure. We provide a closed-form solution to its computation with the following lemma.

Lemma 1: Assume $B_{t'} \sim \text{Poisson}(\lambda t')$, with $0 \leq \lambda t' < \infty$, and assume that $U(e, b, d)$ is well defined for all $b \in \{0, \dots, e\}$, i.e., $\Pr[\Omega_{e,b}] > 0$ for all such b . Then,

$$\tilde{U}(e, t', c; \lambda) = \frac{\sum_{b=0}^e U(e, b, d) (\lambda t')^b / b!}{\sum_{j=0}^e (\lambda t')^j / j!}. \quad (1)$$

Proof: We have

$$\begin{aligned} \tilde{U}(e, t', c; \lambda) &\stackrel{(i)}{=} \sum_{b=0}^e U(e, b, d) \Pr\{B_{t'} = b \mid \mathbb{G}_e\} \\ &\stackrel{(ii)}{=} \sum_{b=0}^e U(e, b, d) \frac{\Pr[B_{t'} = b \cap \mathbb{G}_e]}{\Pr[\mathbb{G}_e]} \\ &\stackrel{(iii)}{=} \sum_{b=0}^e U(e, b, d) \frac{\Pr[B_{t'} = b]}{\sum_{j=0}^e \Pr[B_{t'} = j]} \\ &\stackrel{(iv)}{=} \sum_{b=0}^e U(e, b, d) \frac{\exp(-\lambda t') (\lambda t')^b / b!}{\sum_{j=0}^e \exp(-\lambda t') (\lambda t')^j / j!} \\ &\stackrel{(v)}{=} \frac{\sum_{b=0}^e U(e, b, d) (\lambda t')^b / b!}{\sum_{j=0}^e (\lambda t')^j / j!}. \end{aligned}$$

First, we show that the conditioning on $\mathbb{G}_e$ is well defined. Since $B_{t'} \sim \text{Poisson}(\lambda t')$,

$$\Pr[\mathbb{G}_e] = \sum_{j=0}^e \Pr[B_{t'} = j] = \exp(-\lambda t') \sum_{j=0}^e \frac{(\lambda t')^j}{j!} > 0.$$

Then, (i) follows from Definition 1. Step (ii) follows from Kolmogorov's definition of conditional probability. Step (iii) uses $\{B_{t'} = b\} \subseteq \mathbb{G}_e$ for every $b \in \{0, \dots, e\}$, and the disjoint decomposition $\mathbb{G}_e = \bigcup_{j=0}^e \{B_{t'} = j\}$. Step (iv) substitutes the Poisson mass function for $B_{t'}$, and step (v) cancels the common factor $\exp(-\lambda t')$ from numerator and denominator. ■

B. Empirical Estimation of Utility

a) *Estimating U and c :* We follow the principle that it is better to use estimates that are *conservative*, in the sense that they underestimate rather than overestimate the success rate. By Lemma 1, $\tilde{U}(e, t', c; \lambda)$ is a convex combination of the conditional utilities $U(e, b, d)$. We therefore estimate it by lower-bounding each conditional utility. To guide estimation of c , we rely on the following assumption: *for fixed (e, b) , $U(e, b, d)$ increases monotonically with d .* In other words, a longer search budget cannot decrease the chance of success. Hence, overestimating c underestimates d and therefore U .

To estimate $U(e, b, d)$ from a sample of simulation outcomes $\omega_1 \dots \omega_n \in \Omega_{e,b}$, let $K(e, b, d) = \sum_{j=1}^n H(d, \omega_j)$ be the observed number of successful outcomes. Then $K(e, b, d) \sim \text{Binomial}(n, U(e, b, d))$. Define the one-sided Clopper-Pearson q -confidence lower bound $\underline{U}_q(e, b, d) = \text{Beta}(1 - q; K(e, b, d), n - K(e, b, d))$ if $1 \leq K(e, b, d) \leq n$, or 0 otherwise. Then $\underline{U}_q(e, b, d)$ lower-bounds $U(e, b, d)$ with probability at least q given the sample.

We calculate $H(d, \omega)$ for each simulated transient ω as follows. For a given ω , we compute a likelihood map using its true length and hence true source and background events. Using this map, we perform planning and search using various values for the remaining deadline in order to estimate the minimum search budget $d_{\min}(\omega)$ required to successfully image the transient. $H(d, \omega)$ is then taken to be 1 if $d \geq d_{\min}(\omega)$ and 0 otherwise.

The processing time c arises from a random variable $C(e, b)$, the processing time for a simulation outcome in $\Omega_{e,b}$. $C(e, b)$ is the sum of two random variables $G(e, b)$ and $L(e, b)$, respectively representing map generation and search planning costs. The distributions of G and L are unknown, but we obtain their observed values on the same simulation outcomes $\omega_1 \dots \omega_n$ used to estimate U . We use the q th-quantile values $g_q(e, b)$ and $\ell_q(e, b)$ of the map generation and search planning costs observed on the sample to estimate the corresponding quantiles of $G(e, b)$'s and $L(e, b)$'s distributions, then use the value $d_q(e, b) = D - t' - g_q(e, b) - \ell_q(e, b)$ as a soft lower bound on d when computing $\underline{U}_q(e, b, d)$.

b) *Binned Estimates of Utility:* Unfortunately, our transient simulator cannot produce transients with fixed e and b ; rather, it fixes a *fluence* (total brightness) and length and produces a transient for which b is Poisson with mean λ times the length, and for which $e - b$ is Poisson with mean equal

to the fluence. Because of our limited control over e and b for simulated transients, it is practically cost-prohibitive to perform enough simulations to adequately estimate U , g_q and ℓ_q for every observed value of (e, b) – each of these parameters may be as large as several thousand events. We therefore group simulated transients into bins $[e_l, e_h) \times [b_l, b_h)$ and compute a single estimate of U per bin.

Within a single bin, we use the entire sample of simulation outcomes in the bin to compute utility. But we both adopt a conservative approach to estimating $\underline{U}_q$, g_q , and ℓ_q as described above *and* make the following additional assumption: *for fixed e , $U(e, b, d)$ decreases monotonically, and $g_q(e, b)$ and $\ell_q(e, b)$ increase monotonically, with increasing b .* This assumption states that processing time and utility become worse as the proportion of background noise in the observed events increases. We therefore conservatively estimate the expected success rate in a bin with b endpoints $[b_l, b_h)$ by estimating $g_q(e, b_h)$ and $\ell_q(e, b_h)$ and assigning to each $b_l \leq b < b_h$ in the sum for Equation 1 the utility estimate $\underline{U}_q(e, b_h, D - t' - g_q(e, b_h) - \ell_q(e, b_h))$.

In our implementation, we used a resolution of 20×20 for each bin and used $q = 0.95$ for both U and processing-time estimation. We selected this resolution because, relative to the 20×20 setting, varying the numbers of source and background bins between 10 and 40 changed the estimated success probability by less than 1% when $D = 30$ and by at most 1.5% when $D = 10$, whereas using 50 or more bins produced bins that were too sparsely populated for the available training data. Bins with fewer than 10 training examples for g_q and ℓ_q and 20 examples for U were discarded. To reduce sampling noise in our mapping and planning time estimates, we fit a smooth function of (e, b) to the q th-quantile values for all bins not discarded and used that to estimate all g_q and ℓ_q values. For U , if the bin containing (e, b) was discarded, we instead conservatively used the success rate estimated from the retained bin for the same e range with the least $b_\ell > b$, or set $U = 0$ if no such bin existed.

c) *Using Utility to Choose Start Time t' :* At runtime, we compute the expected utility of initiating map generation at the end of each second after the TAP is first detected. If the utility at time $t \in [0, D)$ is larger than any seen so far, mapping is initiated using the events present at time t , and so $t' = t$. For each time step at which a new highest utility is found, any previous mapping, planning, and search results are discarded, and map generation restarts at the new time. In practice, the partner would likely maintain a record of tiles that it has searched using prior, discarded maps; incorporating this information into planning is a topic for future work.

The runtime cost of utility estimation for one transient is ~ 10 ms per time step on our target platform, which is negligible relative to other computational costs in the pipeline. The total memory required to store binned mapping and planning times and search outcomes for over 50,000 simulated transients, which are needed to infer utilities, is under 1.5 Mbytes. Obtaining this many times and outcomes requires about one full day of training time on the Jetson. If faster

training is desired, the necessary mapping and search runs can be parallelized over multiple devices and/or run on faster processors after calibrating for their costs of mapping and planning relative to the target platform.

IV. CONSTRUCTION OF THE TAP DETECTION PIPELINE

In this section, we describe the components of our pipeline for gamma-ray transient mapping and search. This pipeline targets the gamma-ray detector used by the upcoming ADAPT [2] mission. We used ADAPT’s physical detector simulation and its implementation of event reconstruction [24]–[27] both to learn its instrument response model, which drives likelihood mapping, and to simulate its response to TAPs in order to train the utility model of Section III. We note that the ADAPT instrument, once deployed, will likely see at most 1–2 transients per month with sufficient signal to perform meaningful localization; hence, as is typical in astrophysics, detailed physical simulation was the only timely route to obtain enough transient data to design and rigorously test our system. We did not model any particular partner telescope but instead selected values for its parameters within the range of modern instruments used for follow-up search.

We integrated the components of our pipeline on an Nvidia Jetson Orin NX, a low-power multiprocessor with 8 ARM Cortex-A78AE v8.2 CPU cores and 16 GB of DRAM running Linux. The ARM cores were configured to run at their maximum speed of 1.5 GHz, but the on-board GPU was not utilized. While this system can, in principle, draw up to 25 W, we observed power draws of < 10 W during testing.

A. Construction of Response and Background Models

We first describe how we obtained the instrument response model R and background model B required by likelihood mapping. Constructing these models involves simulating physical interactions of incident particles with the ADAPT detector.

The simulated gamma-ray photons used to learn R were emitted isotropically from a hemisphere surrounding the ADAPT instrument, with log-uniform energies in the range 0.03–30 MeV. A total of 4.4×10^9 gamma rays were fired at the detector, of which only a subset interacted with it. Physical interactions were modeled using the Geant4 particle physics simulator [28] against a detector mass model described in [24]. Signals produced by interacting gamma rays in the detector were processed through a simulation of its electronics [25] and its trajectory reconstruction software [26] to generate Compton rings. Although most incident photons were discarded due to insufficient detector interactions or poor-quality measurements, we nevertheless obtained $\sim 10^7$ Compton rings that were used to learn R .

The response is the product of two components: an *effective area model*, which measures the probability that photons with a given source direction and energy will produce an observed event in the detector, and a *conditional probability model*, which estimates the probability that an observed event from a photon with given source direction and energy results in a ring with a given set of *Compton data space* (CDS) parameters,

i.e., a given measured energy and geometry. Each of these components is a continuous-valued function of its parameters, but for reasons of efficiency at mapping time, we discretize (bin) their parameter spaces and learn a binned approximation to each. The resolution of our binned response followed that of a recently published response for the COSI instrument [29]: 400 equally-spaced source directions (spatial bins) around the hemisphere (HEALPix $N_{\text{side}} = 8$, including only directions with non-negative latitude), ten logarithmically spaced energy bins from 0.1 to 10 MeV, and (for the conditional probability model) 230,400 equally-sized voxels in CDS parameter space.

For the effective area model, which has only 4,000 total bins, we simply binned our training data to estimate directly the fraction of photons with each source direction and energy that resulted in an observed event. For the much larger conditional probability model, we used our simulated photons to train a normalizing flow machine-learning model [30], which has been shown [31], [32] to regularize the learned response and thereby improve its accuracy with limited training data. We then computed the conditional probability model by sampling 2×10^6 events in each source direction/energy bin from the ML model and using these to estimate the fraction of events falling into each CDS bin.

For the background model B , which uses the same CDS voxels as the response, we generated 2.4×10^9 particles according to the model of [33] and processed them through the ADAPT detector model and software, obtaining 6.5×10^5 observed Compton rings. We again used these rings to train a normalizing flow model, from which we sampled 8×10^5 events to estimate the fraction of events in each CDS voxel.

The full response model, stored as single-precision floating point values, requires 3.43 GB of DRAM. The background model is of comparatively negligible size at under 1 MB.

B. Likelihood Mapping

We based our mapping procedure on the accelerated Python implementation described in [5]. Given a set V of observed events during the TAP and models R and B , mapping computes the likelihood $L(\vec{s} | V, \rho, R, B)$ for each source direction $\vec{s}$. Here, ρ is the (*a priori* unknown) fluence observed for the TAP; for each $\vec{s}$, ρ is fit to the data V so as to maximize the likelihood. Assuming a uniform prior over $\vec{s}$, normalizing these likelihoods to sum to 1 over all $\vec{s}$ yields $\Pr(V | \vec{s}, \rho, R, B)$.

For each candidate source direction $\vec{s}$, mapping must load a subset $R_{\vec{s}}$ of R describing the arrival rates per CDS voxel of observed events from a TAP of unit fluence at $\vec{s}$. The (unknown) true energies of these photons follow an *energy spectrum*, which we roughly estimate from the measured energies of observed events as described in [5]. $R_{\vec{s}}$ is then convolved with this spectrum to obtain the averaged arrival rates $\bar{R}_{\vec{s}}$ for photons arriving from $\vec{s}$. During fitting, $\bar{R}_{\vec{s}}$ is scaled by the estimated fluence ρ to obtain the expected event rates for the TAP in each CDS voxel.

The background model B similarly gives a rate of observed background events for each CDS voxel. These rates, which sum to 1, are scaled prior to mapping by the same overall

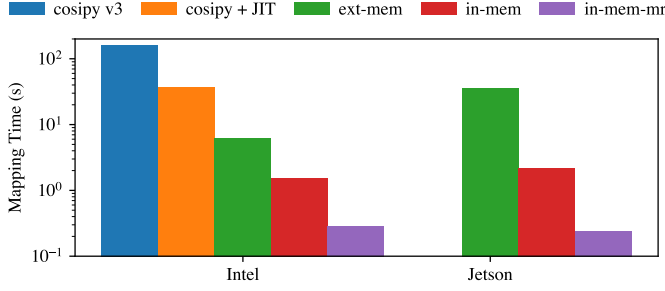


Fig. 4. Comparison of running times for mapping implementations on a simulated GRB from `cosipy`.

background rate λ used in our utility calculation. We estimate λ from the number of events observed in a ten-minute window (with no other transients) just prior to the TAP.

Implementation: The starting point for our mapping implementation was the code provided by the `cosipy` v3 Python library [23]. Compared to this prior implementation, our mapping software introduced a number of changes to improve mapping performance. We first recoded the core per-direction likelihood computation for greater performance using the Numba JIT library [34]. We then rewrote much of the surrounding code, replacing `cosipy`’s high-level multiprocessing-based parallelism with a low-level multithreading approach using Numba’s `prange` construct and precomputing values such as the spectral flux used for convolution. The resulting mapping implementation, which we denote as **ext-mem**, still loads individual response slices $R_{\vec{s}}$ for each source direction $\vec{s}$ on demand from secondary storage. We further accelerated mapping by preloading the whole of R (which is independent of any particular TAP) into DRAM and then, for each TAP, reducing it to just those CDS voxels corresponding to events that actually appear during the TAP. We denote this implementation as **in-mem**. Finally, we used the multiresolution approach described in [5] to rapidly discard source directions with the lowest likelihoods of containing the TAP, producing the implementation **in-mem-mr** used in this work.

Figure 4 shows the impact of our modifications to mapping. We benchmarked mapping on the simulated GRB provided in `cosipy`’s `ts_map` tutorial. Running times were measured on two platforms: a 2.3 GHz Intel Xeon Gold 5118 CPU with 256 GB DRAM, and the Jetson Orin NX with 16 GB DRAM. We used eight CPU cores on each platform and measured “hot-cache” performance after JIT compilation and any OS-level file caching of R into DRAM. Simply JIT-compiling the likelihood computation achieved nearly $5\times$ speedup on the Intel platform. However, neither this nor the original `cosipy` implementation fit within the Jetson’s limited DRAM, which caused Python’s multiprocessing to become unstable and prevented the computation from completing.

The additional improvements in **ext-mem** achieved a further $6\times$ speedup on the Intel platform and reduced memory usage enough to allow mapping to run on the Jetson. However, the cost to load response slices dynamically during mapping was sensitive both to DRAM file cache size and to overheads

associated with reading, reducing, and convolving one slice of R at a time, which resulted in much slower performance on the Jetson than on the Intel platform. In contrast, once the full response R was loaded into memory offline, **in-mem** needed at most ~ 100 ms to reduce and convolve R for each TAP. By eliminating per-slice overheads, the **in-mem** implementation achieved $17\times$ speedup over **ext-mem** on the Jetson. Finally, multiresolution mapping reduced the number of source directions processed by an order of magnitude and so achieved a correspondingly large speedup.

C. Search Planning

To plan a follow-up search given a likelihood map for a TAP, we implemented a method introduced by Wang et al. [6] that transforms the likelihood map into an actionable search sequence in a way that accounts for the real-time deadline D to image the TAP. Planning consists of two primary stages: sky map discretization (tiling), and deadline-aware path construction.

a) Tiling: A TAP’s likelihood map is presented to search planning as a set of probabilities in each pixel of a hemispherical HEALPix map. To translate this representation into a realizable search plan, we first discretize this map into a set τ of observation tiles, where each tile has the shape of the partner instrument’s FoV. Each tile $\tau_i \in \tau$ is assigned a probability mass p_i equal to the total mass of pixels it covers. This transformation reduces the search task to a discrete prize-collecting s - t path problem over the set of candidate tiles.

b) Modeling the Partner Instrument: In addition to its FoV, the partner is characterized by two models: the *slew time* model, which describes how quickly it can move between two tiles, and the *dwell time* model, which describes how long it needs to image a tile to detect the TAP.

Slew time is constrained by a telescope’s mechanical dynamics. The partner can slew its aperture in an arbitrary direction with a maximum angular velocity $w_{\max}$ and can accelerate and decelerate with constant angular acceleration w_{acc} . For any two tiles $\tau_i, \tau_j \in \tau$, let θ_{ij} be the angular separation of their centers on the unit sphere, and let $\theta_{\min} = w_{\max}^2 / w_{\text{acc}}$. Then slew time $m(\tau_i, \tau_j)$ is given by

$$m(\tau_i, \tau_j) = \begin{cases} \frac{2w_{\max}}{w_{\text{acc}}} + \frac{\theta_{ij} - \theta_{\min}}{w_{\max}}, & \theta_{ij} > \theta_{\min} \\ 2\sqrt{\frac{\theta_{ij}}{w_{\text{acc}}}}, & \theta_{ij} \leq \theta_{\min}. \end{cases}$$

The angular distance threshold $\theta_{\min}$ determines whether slewing can reach its maximum velocity before it must begin to decelerate as it approaches τ_j . We used $w_{\max} = 10^\circ/\text{s}$ and $w_{\text{acc}} = 10^\circ/\text{s}^2$, which are at the low end of rates for modern, fast-slewing instruments [14].

Our **dwell time** model closely follows that of [6], which accounts for directional atmospheric light attenuation at sea level. For a transient appearing in the sky at a polar angle θ_i in degrees relative to the zenith direction, that work computes the atmospheric mass AM of the air column using the Kasten & Young model [35], then uses SMARTS [36] to derive an exponential fit of relative observed flux $s_i \approx 1.1129e^{-0.107AM_i}$

for near-infrared optical light. It assumes a fixed target signal-to-noise ratio (SNR), with signal proportional to exposure time and noise proportional to the square root of exposure time, in which the dwell time scales inversely with the square of the observed flux. Thus, the angle-dependent dwell time assigned to tile τ_i is $C_i = C_0/s_i^2$ assuming a baseline dwell time $C_0 = 1$ sec.

c) *Search Planning*: Search planning must construct a path through some or all of the tiles in τ that uses total time at most d , which is the overall deadline D minus time spent gathering data and creating the likelihood map. The telescope spends time traversing the edges of this path (slewing) and dwelling at each of its tiles to collect light sufficient to detect whether the TAP is present.

To achieve low-latency search planning on our SWaP-constrained platform, we used the *Greedy Christofides Pathfinding* (GCP) planning heuristic of [6]. GCP decomposes the planning process into two stages: subset selection and path sequencing. Subset selection identifies candidate subsets of tiles using a probability-prioritized strategy. For each subset, path sequencing constructs an s - t search path using Hooijgeveen's $\frac{5}{3}$ -approximation for the minimum-weight spanning path problem [37]. To maximize probability mass collected within the available search time, GCP uses binary search to identify a high-probability candidate subset whose corresponding search path remains within the time budget d .

V. RESULTS

In this section, we investigate the performance of our TAP detection system and, in particular, of our utility-based approach to data gathering.

A. Validation Data Sets

To validate our methods, we used synthetic gamma-ray burst transients obtained with the physical simulation framework described in Section IV-A. We generated transients whose source was chosen uniformly from one of 88 directions equally spaced around a hemisphere. Photon energies for each transient were sampled from a Band spectrum [11] with $\alpha = -0.5$, $\beta = -2.35$, and $E_{peak} = 490$ keV per [38]. Each transient's brightness over time followed a Gaussian curve that peaked at half its length, dropping by two standard deviations at its start and end. Total fluence was chosen uniformly at random from $[0.5, 5]$ MeV/cm²; the lower value roughly matches the limits of ADAPT's expected sensitivity to short transients.

We produced two test sets of 10,000 transients, representing two different scenarios. The first set consisted of short transients (length chosen uniformly from $[0.25, 5]$ s) and used the background radiation model of [33]. These transients model short GRBs that will be observed by the balloon-borne ADAPT instrument from within Earth's atmosphere. The second set consisted of longer transients (length chosen uniformly from $[5, 30]$ s) and used one-tenth the background intensity of the first set. This background model approximates what might be seen by an ADAPT-like detector in low Earth orbit, based on the event rates modeled for the COSI mission [29]. We used

a different background model for long transients because they have the same fluence as the short transients but are almost $6\times$ longer on average; hence, their peak brightness is almost $6\times$ less, and they would likely be obscured by the higher atmospheric background.

The actual source photon count for each simulated transient was sampled from a Poisson model using its fluence as the mean. Background events during each transient were sampled from a Poisson process with the rate dictated by our background simulation. We generated at least 10s more background for each transient than its actual length, so that, as in real operation, our pipeline could either over- or under-estimate this length.

B. Determining When to Map

To estimate TAP length and determine when to stop gathering data and begin map generation, we used the utility-maximizing approach of Section III. For each scenario above, we generated a separate training set of 52,800 simulated TAPs with the same distribution of directions, lengths, and fluences and the same background model as the corresponding test set. We used the simulated background for ten minutes prior to each transient to estimate its λ ; these estimates were generally accurate to within 1% of the true value. We generated a likelihood map for each transient using its true length, assuming that mapping started as soon as the transient ended. The resulting training data included the map itself, the time spent in gathering data and mapping, and the actual numbers of source and background events s and b observed for the transient. This data was used to estimate search times and to construct utility estimators as described in Section III for each scenario and each of two partner FoVs: $2.5 \times 2.5^\circ$ and $5.36 \times 4.5^\circ$. All times used in training were measured on the Jetson Orin NX platform.

To evaluate whether deadline-aware utility estimation benefits TAP search, we compared this approach to the following deadline-oblivious baseline method for length determination described in [5]. Time is divided into 1-second steps starting when the TAP is first detected. If the number of events in a time step exceeds the number seen in any previous step, the TAP's length is extended to the end of this maximum step. The TAP is assumed to end just before the first step following the maximum in which the number of observed events does not exceed the 95th percentile value for a Poisson distribution with background mean λ . The total wait time before map generation is therefore one second longer than the length of the time window from which we draw events for mapping. (In contrast, utility incurs no delay between the end of data gathering and the start of mapping.)

During testing, we produced a likelihood map for each transient in the test set using either (1) our utility-maximizing approach under a known deadline D , using the appropriate utility model for the transient's scenario and partner's FoV, or (2) the deadline-oblivious baseline method. We used the resulting map, together with a record of the time until map generation was complete, as input to the GCP search planning

algorithm, which produced a tile sequence to be searched. Finally, we determined whether this sequence imaged a tile that overlapped the true location of the transient within the residual search budget before the overall deadline D ; if so, the transient was considered to be successfully found.

C. Results: Mapping and Search Planning Times

We first quantify the costs of likelihood mapping and search planning, which are important inputs to our utility method. Times were measured on our training sets assuming that the pipeline correctly inferred the ground-truth length for each TAP; however, the results were not sensitive to small variations in estimated TAP length.

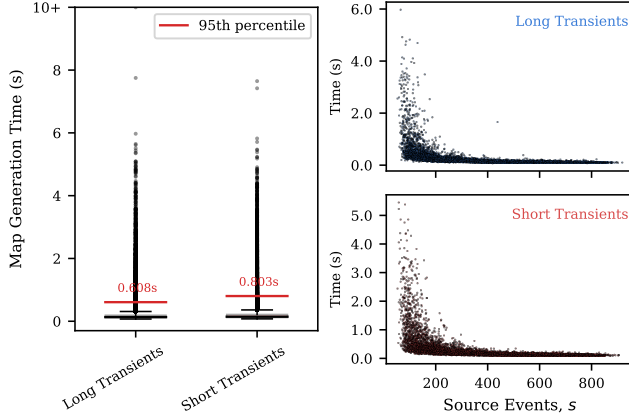


Fig. 5. *Left*: Box plots of map-generation times for 52,800 simulated TAPs on the Jetson Orin NX for short and long transients. *Right*: Map-generation time vs. number of source events $s = e - b$. Most outliers occur when $s < 350$.

Figure 5 shows distributions of time spent in mapping for each of our two training sets. The 95th percentile time to compute a likelihood map for a TAP is < 1 sec, while the 75th percentile time is below 0.25s. Behavior is qualitatively similar for the short and long scenarios. Higher mapping times correspond to TAPs with the fewest observed source events (fewer than 350), for which likelihood maximization for each $\vec{s}$ takes the longest to converge.

Figure 6 shows the distributions of time spent in search planning for maps generated from short and long TAPs and for different partner telescope FoVs. For the vast majority of maps, which are compact and localize the TAP to within a few degrees, planning with GCP requires only tens of milliseconds. The long tail of planning times again corresponds to maps from TAPs with few (< 200) observed source events. Search planning time is inversely proportional to the telescope’s FoV area, which determines the number of distinct tiles that must be considered, so for low-quality maps requiring many tiles to cover, the tail of search times is longer for smaller-FoV instruments. Even so, less than 2% of transients for the smaller FoV, and less than 0.1% for the larger, required more than 0.1s.

D. Results: Likelihood Mapping Behavior

We next investigate the accuracy of likelihood mapping in isolation. To quantify the accuracy of mapping, we define

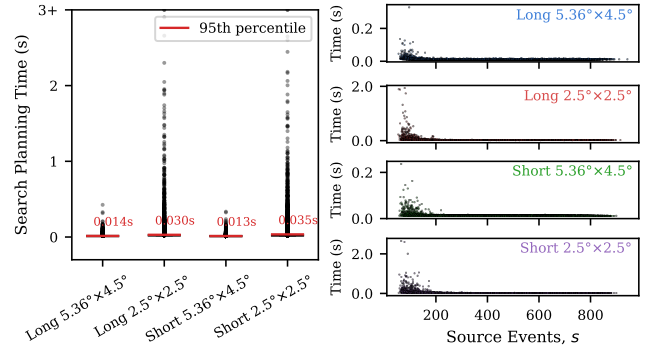


Fig. 6. *Left*: Box plots of search-planning time for short and long transients using different partner FoVs. *Right*: Search-planning time vs. number of source events. Most outliers have $s < 200$.

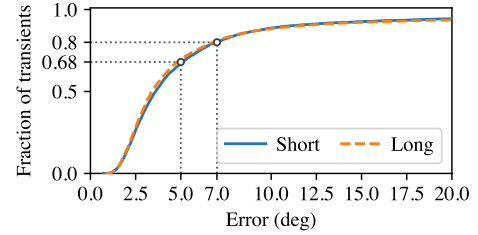


Fig. 7. Cumulative distribution of mapping errors for short and long transients. CDFs reach 1 at $\sim 160^\circ$ error.

the *average angular error* of a map. Let $\vec{s}^*$ be the true source direction of a transient, and let M be the set of source directions $\vec{s}$ to which the map assigns a non-zero probability $p(\vec{s})$. Then the average angular error is given by $\sum_{\vec{s} \in M} p(\vec{s}) d(\vec{s}, \vec{s}^*)$, where $d(x, y)$ is the angular distance between x and y on the surface of the unit sphere.

Figure 7 shows the cumulative distribution of mapping error for our two test sets when mapped with the deadline-oblivious method. 68% of maps have error within $\sim 5^\circ$, while 80% have error within 7° . There is a long tail of transients that produce less accurate maps, either because the transient has low signal-to-noise (low s relative to b) or because its endpoint has been poorly determined and so includes too little signal or too much background. These error rates are comparable to those reported for a different ADAPT transient localization method in [26], though that method produced only a point-estimate of the source direction rather than a full likelihood map.

Figure 8 shows how mapping error varies with the numbers of source events s and background events b . Error (indicated by point color) decreases rapidly as more source events accumulate. Error grows with increased background, but only slowly; when the number of source events is large, even a signal-to-noise ratio of 1:3 can produce accurate maps. The long tails of the CDFs in Figure 7 are composed almost entirely of transients inferred from very few source events.

Impact of Utility Estimation: We hypothesized that utility estimation would alter estimated transient lengths – i.e., the duration of time the pipeline waits while collecting more data before generating a map – compared to our baseline method,

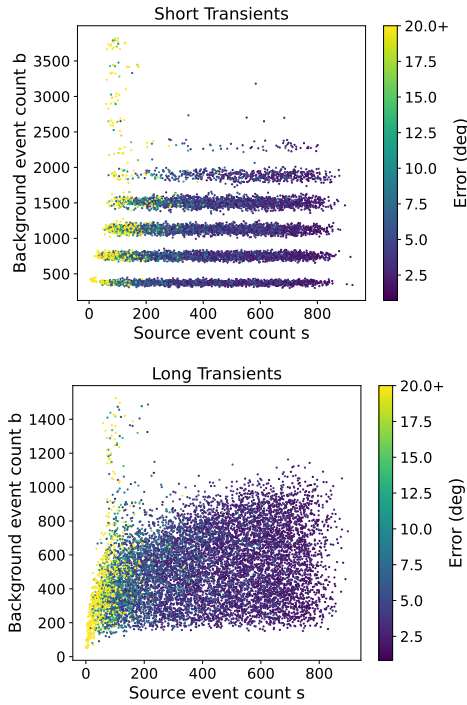


Fig. 8. Dependence of mapping error on numbers of source and background events. Horizontal stripes reflect integer-valued transient lengths inferred by mapping; these stripes are more numerous and overlap for long transients.

which (as shown in Figure 2) could impact mapping accuracy. To assess its impact, we compared the distributions of transient lengths and the resulting mapping errors obtained by utility vs. the baseline. Here, we show results only for long transients using an FoV of $5.36 \times 4.5^\circ$; these results are qualitatively similar to those seen for the other test set and FoV.

When a deadline D is imposed, mapping must trade off the advantage of accumulating more source events, and hence producing a more accurate map, against the disadvantage of increased waiting time before the partner instrument can begin to search for the transient. Figure 9 shows the change in estimated transient lengths under two deadlines $D \in \{30, 60\}$ s relative to the deadline-oblivious method. Each plot is a histogram of *difference* in estimated lengths for each transient in the test set under the utility model vs. the base deadline-oblivious model. Values greater than (resp. less than) 0 indicate that the utility method waited more (resp. less) time than the baseline before initiating map generation. When deadline pressure is stronger ($D = 30$), the utility method is biased toward mapping earlier than the baseline, even at the cost of dropping some signal from the transient. In contrast, for a longer deadline ($D = 60$), the utility method often chooses to gather more data than the baseline in hopes of increasing map accuracy and hence the chance of successfully imaging the transient. We find that these deadline extensions mostly occur for the brightest transients, where there is clear evidence that the event count at later times remains well in excess of the background rate.

Figure 10 illustrates how the utility method’s decisions

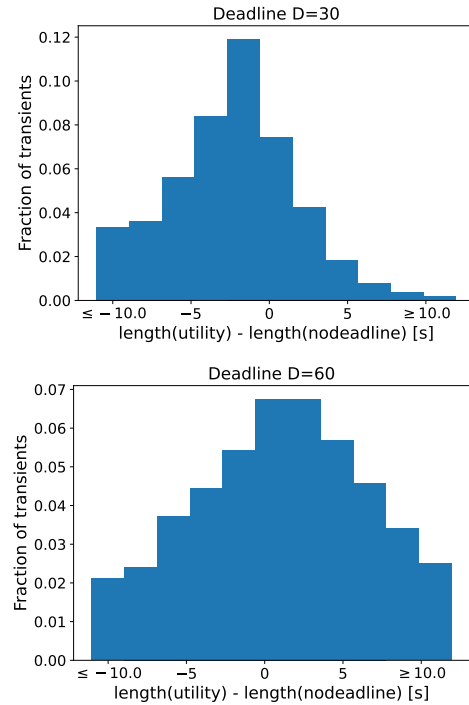


Fig. 9. Impact of deadline on utility-based transient length determination relative to deadline-oblivious baseline, measured on long transient set for FoV $5.36 \times 4.5^\circ$.

about the time at which mapping begins impact accuracy of the produced map for the same transients considered in Figure 9. For each deadline, we group transients according to how much their estimated length changed under the utility method vs. the deadline-oblivious baseline. Each group represents a range of length changes, with a typical value for each group shown on the x -axis. For each group, we show the distribution of the *difference* in mapping error for the utility method vs. the baseline. Differences less than (resp. greater than) 0 indicate that the utility method achieved better (resp. worse) mapping accuracy than the baseline. For most transients, the length change imposed by the utility vs. the deadline-oblivious approach incurs only a small change in error. A small number of outliers exhibit a large change in mapping error. These are biased toward higher error than the baseline (a failure of the utility method) when length is reduced under deadline pressure and toward lower error (recovery of a baseline method failure) when length is extended.

Overall, we find that the utility-based approach incurs a relatively small impact on the distribution of mapping errors. Strict deadline pressure tends to increase mapping error, while less pressure can actually decrease error because utility is willing to gather more data before mapping.

E. Results: Successful Detection of Transients

Finally, we turn to end-to-end evaluation of how often our cooperative pipeline successfully images the transient. Figures 11a and 11b show the rates of successful detection as a function of deadline for our test transients with a FoV

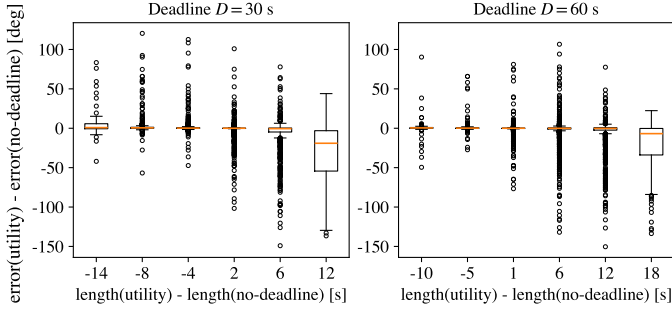


Fig. 10. Impact of deadline on mapping error of utility-based method, measured on long transient set for FoV $5.36 \times 4.5^\circ$. Transients are grouped by change in length vs. the deadline-oblivious method; x -axis labels show a representative value for each group. Within each group, a box plot shows the distribution of changes in error for the utility method vs. the baseline.

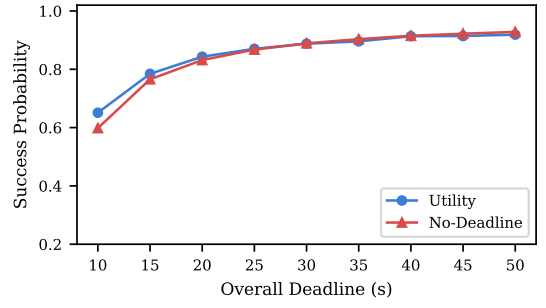
of $5.36 \times 4.5^\circ$. For short transients (0.25–5s), the utility-based approach resulted in more successful detections for deadlines between 10 and 20s. For longer deadlines, the success rate of the utility-based method matched that of the deadline-oblivious method to within 1%.

For longer transients (5–30s) with lower background, the utility-based approach consistently outperformed the baseline at all deadlines tested. As noted above, the utility approach tends to favor gathering more data for bright transients when deadline pressure is low, which improves map quality. This preference was empirically stronger for the long-transient scenario, which plausibly explains the difference in success rate.

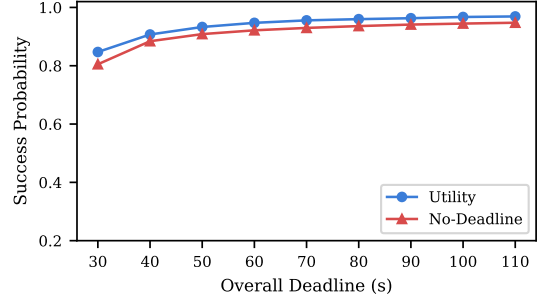
When the partner has a smaller FoV, searching a given map requires imaging a larger number of tiles. Hence, for a fixed deadline, we expect that overall success rates will decrease compared to a larger FoV. This is consistent with the observed success rates for the $2.5 \times 2.5^\circ$ FoV in Figures 11c and 11d. However, the qualitative patterns match those observed for the larger FoV: for short transients, the utility method outperforms the baseline under strong deadline pressure and matches it for longer deadlines, while for longer transients, it consistently outperforms the baseline across all deadlines tested.

VI. CONCLUSION AND FUTURE WORK

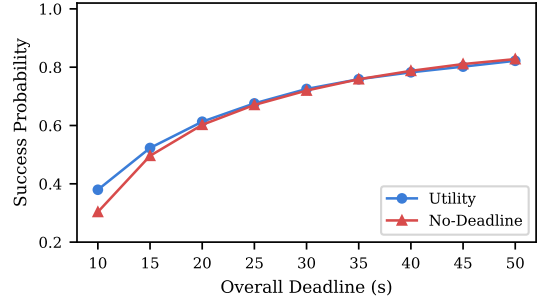
Cooperative transient search is both a scientifically valuable application and a design problem involving coordination of computational actors under uncertainty that must meet both time and resource constraints. To address the challenge of uncertainty, we model how a key parameter – data gathering time – impacts the overall chance of successful transient imaging through changes in both available time and data quality, then use this model to make runtime decisions that maximize expected success. We have demonstrated that the components of cooperative search can be efficiently implemented and integrated on SWaP-constrained hardware and have validated, using a model of a soon-to-fly gamma-ray telescope, that the resulting system’s rate of success does indeed benefit from applying our utility model at runtime. Our design offers guidance not only for ADAPT but also



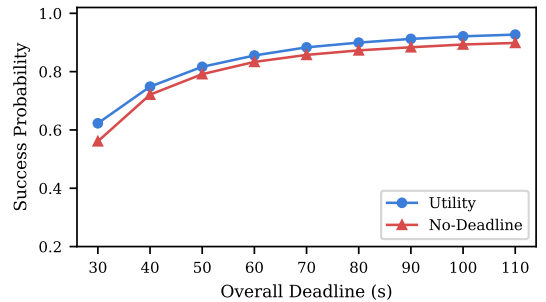
(a) Short transients, FoV $5.36 \times 4.5^\circ$



(b) Long transients, FoV $5.36 \times 4.5^\circ$



(c) Short transients, FoV $2.5 \times 2.5^\circ$



(d) Long transients, FoV $2.5 \times 2.5^\circ$

Fig. 11. Probability of successful transient detection by cooperating telescopes as a function of overall deadline D .

for future observing missions that will do on-board, real-time coordination of telescopes in flight.

Our work offers opportunities for extension. First, our utility model assumed that likelihood mapping and search operate either sequentially on the same processor or on separate processors with negligible communication delay. For

cooperation between flying and terrestrial instruments, or among a constellation of nearby instruments in space, we can add the cost and uncertainty of communication as an additional step in our utility model. For multiple instruments sharing a computational resource, we may instead consider the impact of concurrently executing mapping and search computations, since utility estimation does not stop even after search has begun. Satellite-to-ground and instrument-to-instrument communication is a complex, dynamic field [22] that will likely require substantial work to develop realistic latency and throughput models.

Another assumption of our model is that the overall deadline D is fixed based on astrophysicists' desire to image a particular kind of TAP whose optical light curve can be anticipated. In practice, discovering the properties of the TAP, and hence the deadline D , might require analysis of the TAP's gamma-ray light curve, energy spectrum, and polarization signature. We have developed preliminary implementations of such analyses that run with speed comparable to mapping under the resource constraints of our SWaP-constrained platform, but integrating them into on-board decision making requires a more complex notion of utility. On the one hand, we recently described [6] a strategy for search planning under *multiple* deadlines (say, for imaging different types of TAP) where the chances of success at each deadline must be balanced. On the other hand, our utility model could potentially extend to include uncertainty about the deadline that is resolved only after observing some events. Ultimately, our goal is to enable effective on-board decision-making among instruments so as to support greater telescope autonomy in cooperative astrophysical observation.

ACKNOWLEDGMENTS

We thank the ADAPT and COSI Collaborations for their guidance on the simulations and mapping methods used in this work. Special thanks are due to Pascal Janowski for guidance on machine-learning response models and to Israel Martinez and the `cosipy` team for assistance in adapting COSI's analysis software. Thanks also to Sophie Fendler, Aavik Wadivkar, and Nate Hayman of WashU Satellite for information about AIRIS's optomechanical properties and software pipeline.

The authors attest that generative AI tools were used in this work only (1) to produce standard training loop and inference code over the ML response model of Section IV-A, (2) to produce plotting code for some visualizations of our results, and (3) to perform minor stylistic revisions and figure positioning for the final manuscript.

REFERENCES

- [1] National Academies of Sciences Engineering and Medicine, *Pathways to Discovery in Astronomy and Astrophysics for the 2020s*. Washington, DC, USA: The National Academies Press, 2023. [Online]. Available: <https://nap.nationalacademies.org/catalog/26141/pathways-to-discovery-in-astronomy-and-astrophysics-for-the-2020s>
- [2] J. Buckley *et al.*, "The Advanced Particle-Astrophysics Telescope (APT) project status," in *Proc. 37th Int'l Cosmic Ray Conf.*, vol. 395, Jul. 2021, pp. 655:1–655:9.
- [3] N. Gehrels, E. Ramirez-Ruiz, and D. B. Fox, "The Swift gamma-ray burst mission," *Annual Review of Astronomy and Astrophysics*, vol. 47, pp. 567–617, 2009.
- [4] J. A. Tomsick, S. E. Boggs, A. Zoglauer, D. Hartmann *et al.*, "The Compton Spectrometer and Imager," 2023. [Online]. Available: <https://arxiv.org/abs/2308.12362>
- [5] J. Buhler and M. Sudvar, "Real-time likelihood map generation to localize short-duration gamma-ray transients," in *Proc. 39th Int'l Cosmic Ray Conf.*, vol. 501, Jul. 2025, pp. 587:1–587:9.
- [6] D. Wang, M. Sudvar, F. Marković, J. Buhler, S. Baruah, and G. Kehne, "Probabilistic response-time-aware search for transient astrophysical phenomena," in *Proc. 46th IEEE Real-Time Systems Symposium (RTSS)*, 2025, pp. 1–15.
- [7] H. Wu, B. Ravindran, E. D. Jensen, and P. Li, "Energy-efficient, utility accrual scheduling under resource constraints for mobile embedded systems," in *Proc. 4th ACM Int'l Conf. Embedded Software (EMSOFT)*. ACM, 2004, pp. 64–73.
- [8] J. Chen, D. V. Le, Y. Li, Y. Liu, and R. Tan, "Timelynet: Adaptive neural architecture for autonomous driving with dynamic deadline," *ACM Transactions on Embedded Computing Systems*, vol. 24, no. 5s, pp. 1–23, 2025, presented at EMSOFT 2025.
- [9] D. Kang, S. Lee, C.-H. Hong, J. Lee, and H. Baek, "Batch-MOT: Batch-enabled real-time scheduling for multiobject tracking tasks," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 43, no. 11, pp. 3539–3550, 2024, presented at EMSOFT 2024.
- [10] E. Waxman and B. Katz, "Shock breakout theory," in *Handbook of Supernovae*, A. W. Alsabti and P. Murdin, Eds. Cham: Springer International Publishing, 2017, pp. 967–1015.
- [11] D. Band *et al.*, "BATSE observations of gamma-ray burst spectra. I. spectral diversity," *Astrophysical J.*, vol. 413, p. 281, Aug. 1993.
- [12] C. Meegan, G. Lichti, P. N. Bhat *et al.*, "The Fermi gamma-ray burst monitor," *Astrophysical J.*, vol. 702, no. 1, pp. 791–804, Aug. 2009.
- [13] R. Caputo *et al.*, "ComPair-2: a next-generation medium-energy gamma-ray telescope prototype," *Proc. SPIE Int. Soc. Opt. Eng.*, vol. 13093, p. 130932L, 2024.
- [14] J. H. Kim, M. Im, H. M. Lee, S.-W. Chang, H. Choi, and G. S. H. Paek, "Introduction to the 7-Dimensional Telescope: Commissioning procedures and data characteristics," 2024. [Online]. Available: <https://arxiv.org/abs/2406.16462>
- [15] P. W. Roming, T. E. Kennedy, K. O. Mason, J. A. Nousek, L. Ahr, R. E. Bingham, P. S. Broos, M. J. Carter, B. K. Hancock, H. E. Huckle *et al.*, "The Swift ultra-violet/optical telescope," *Space Science Reviews*, vol. 120, no. 3, pp. 95–142, 2005.
- [16] A. Wadivkar, B. Cook, S. Fendler, I. Shultz, N. Hayman, J. Billips, O. Cromly, J. Galloway, O. Nieuwenhuizen, G. Goffman, E. Teixeira, O. Yeaman, W. Gao, and WashU Satellite, "AIRIS: Pushing the boundaries of GRB optical afterglow observation," in *AAS/High Energy Astrophysics Division*, ser. AAS/High Energy Astrophysics Division, vol. 22, Nov. 2025, p. 111.07.
- [17] K. M. Gorski *et al.*, "HEALPix: A framework for high-resolution discretization and fast analysis of data distributed on the sphere," *Astrophysical J.*, vol. 622, no. 2, p. 759, 2005.
- [18] S. Boggs and P. Jean, "Event reconstruction in high resolution Compton telescopes," *Astronomy and Astrophysics Suppl.*, vol. 145, no. 2, pp. 311–321, 2000.
- [19] A. Pollock, G. Bignami, W. Hermsen, G. Kanbach, G. Lichti, J. Masnou, B. Swanenburg, and R. Wills, "Search for gamma-radiation from extragalactic objects using a likelihood method," *Astronomy and Astrophysics*, vol. 94, pp. 116–120, 1981.
- [20] T. Herbert, "Estimating the sky map in gamma-ray astronomy with a Compton telescope," *IEEE Transactions on Nuclear Science*, vol. 38, no. 2, pp. 536–542, 1991.
- [21] L. P. Singer, A. W. Criswell, S. C. Leggio, R. W. Kiendrebeogo, M. W. Coughlin, H. P. Earnshaw, S. Gezari, B. W. Grefenstette, F. A. Harrison, M. M. Kasliwal, B. M. Morris, E. Tollerud, and S. B. Cenko, "Optimal follow-up of gravitational-wave events with the UltraViolet EXplorer (UVEX)," *Publications of the Astronomical Society of the Pacific*, vol. 137, no. 7, p. 074501, Jul. 2025.
- [22] J. A. Kennea, J. L. Racusin, E. Burns, B. W. Grefenstette, R. A. Hounsell, C. M. Hui, D. Kocevski, T. J. W. Lazio, S. Lesage, T. A. Pritchard, A. Tohuavohu, J. A. Tomsick, D. Traore, and C. A. Wilson-Hodge, "Time-domain and multimessenger astrophysics communications science analysis group report," 2024. [Online]. Available: <https://arxiv.org/abs/2410.03980>

- [23] I. Martinez *et al.*, “The cosipy library: COSI’s high-level analysis software,” in *Proc. 38th Int’l Cosmic Ray Conf.*, vol. 444, 2023, pp. 858:1–858:8.
- [24] W. Chen *et al.*, “The Advanced Particle-astrophysics Telescope: Simulation of the instrument performance for gamma-ray detection,” in *Proc. 37th Int’l Cosmic Ray Conf.*, vol. 395, 2021, pp. 590:1–590:9.
- [25] M. Sudvarg *et al.*, “Front-end computational modeling and design for the Antarctic Demonstrator for the Advanced Particle-astrophysics Telescope,” in *Proc. 38th Int’l Cosmic Ray Conf.*, vol. 444, Jul. 2023, pp. 764:1–764:9.
- [26] Y. Htet *et al.*, “Prompt and accurate GRB source localization aboard the Advanced Particle Astrophysics Telescope (APT) and its Antarctic Demonstrator (ADAPT),” in *Proc. 38th Int’l Cosmic Ray Conf.*, vol. 444, Jul. 2023, pp. 956:1–956:9.
- [27] Y. Htet, M. Sudvarg, H. Yang, J. Buhler, R. Chamberlain, W. Chen, J. Buckley, and the APT Collaboration, “Performance modeling and improvements on the GRB source localization streaming pipeline aboard the Antarctic Demonstrator for the Advanced Particle-Astrophysics Telescope (ADAPT),” in *Proc. 39th Int’l Cosmic Ray Conf.*, vol. 501, Jul. 2025, pp. 679:1–679:9.
- [28] S. Agostinelli, J. Allison, K. Amako *et al.*, “Geant4 — a simulation toolkit,” *Nucl. Instrum. Methods Phys. Res. A*, vol. 506, no. 3, pp. 250–303, 2003.
- [29] Compton Spectrometer and Imager (COSI) Collaboration, “COSI Data Challenges,” Apr. 2025. [Online]. Available: <https://doi.org/10.5281/zenodo.15126188>
- [30] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan, “Normalizing flows for probabilistic modeling and inference,” *J. Mach. Learn. Res.*, vol. 22, no. 1, Jan. 2021.
- [31] P. Janowski, S. Gallego, U. Oberlack, J. Lommler, I. Martinez-Castellanos, J. Tomsick, A. Zoglauer, and S. Boggs, “Approximating the COSI telescope response with neural networks,” in *Proc. 39th Int’l Cosmic Ray Conf.*, vol. 501, Jul. 2025, pp. 698:1–698:8.
- [32] P. Janowski, “Development of an efficient response description for the COSI MeV gamma-ray telescope,” Master’s thesis, Johannes Gutenberg-Universität Mainz, Institut für Physik, 2025.
- [33] W. Chen *et al.*, “Simulation of the instrument performance of the Antarctic Demonstrator for the Advanced Particle-astrophysics Telescope in the presence of the MeV background,” in *Proc. 38th Int’l Cosmic Ray Conf.*, vol. 444, Jul. 2023, pp. 841:1–841:9.
- [34] S. K. Lam, A. Pitrou, and S. Seibert, “Numba: a LLVM-based Python JIT compiler,” in *Proc. 2nd Workshop on the LLVM Compiler Infrastructure in HPC*, ser. LLVM ’15. New York, NY, USA: Association for Computing Machinery, 2015.
- [35] F. Kasten and A. T. Young, “Revised optical air mass tables and approximation formula,” *Appl. Opt.*, vol. 28, no. 22, pp. 4735–4738, Nov 1989.
- [36] C. Gueymard *et al.*, *SMARTS2: a simple model of the atmospheric radiative transfer of sunshine: algorithms and performance assessment*. Florida Solar Energy Center Cocoa, FL, USA, 1995, vol. 1.
- [37] J. Hoogeveen, “Analysis of Christofides’ heuristic: Some paths are more difficult than cycles,” *Operations Research Letters*, vol. 10, no. 5, pp. 291–295, 1991.
- [38] Y. Htet *et al.*, “Machine learning aboard the ADAPT gamma-ray telescope,” in *Proc. Workshops of the Int’l Conf. on High Performance Computing, Network, Storage, and Analysis*, Nov. 2024, pp. 4–10.



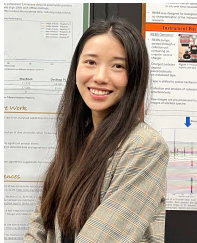
Ye Htet received his B.S. in mathematics and computer science from Illinois College in 2021. He is currently a Ph.D. student at Washington University in Saint Louis. His research interests include modeling and inference optimization.



Jeremy Buhler (Senior Member, IEEE) received his B.S. in computer science from Rice University in 1996 and his Ph.D. from the University of Washington in 2001. He is currently Professor of Computer Science and Engineering at Washington University. His research interests include parallel computing and acceleration of scientific computations. Dr. Buhler leads analytical algorithm design for the ADAPT gamma-ray observatory mission and is a member of the COSI Science Team.



Filip Marković is an Assistant Professor in Computer Science at the University of Southampton. He previously held a postdoctoral position at the Max Planck Institute for Software Systems and holds a PhD in Computer Science from Mälardalen University. His research focuses on uncertainty-aware analysis and scheduling for cyber-physical systems. His work received Outstanding Paper Awards at the IEEE Symposium on Real-Time Systems (RTSS) in 2024 and the Euromicro Conference on Real-Time Systems in 2021; Best Presentation Award at RTSS 2022; and the Outstanding Reviewer Award at the IEEE Real-Time and Embedded Technology and Applications Symposium in 2025.



Daisy Wang is a Ph.D. student in computer science at Washington University in St. Louis. Her research interests lie at the intersection of real-time systems, parallel computing, online algorithms, and combinatorial optimization.



Marion Sudvarg (Member, IEEE) is an Assistant Professor in Computer Science at the Missouri University of Science & Technology. He previously held a postdoctoral position in the Department of Physics at Washington University in St. Louis after completing a Ph.D. in Computer Science at the same institution. He is the firmware and flight software development lead for the ADAPT gamma-ray observatory mission and is a member of the COSI science team. He received the Best Presentation and Best Reviewer awards at the IEEE Real-Time and Embedded Technology and Applications Symposium in 2026.