Adaptive Real-Time Computation for Prompt Localization of Transients

Marion Sudvarg (msudvarg@wustl.edu, www.sudvarg.com)
with Ye Htet, Jeremy Buhler, Roger Chamberlain, Chris Gill, Jim Buckley, and Wenlei Chen for the APT collaboration

The Astro2020 decadal survey identified "time-domain and multi-messenger" programs as the highest-priority sustaining activity for space-based missions.

> **"It is essential to maintain and expand space-based time-domain and follow-up facilities in space."**
>
> Engineering National Academies of Sciences and Medicine. Pathways to Discovery in Astronomy and Astrophysics for the 2020s. The National Academies Press, Washington, DC, 2023

Key Motivation

- Need to localize promptly to capture early follow-up observations.
- Can we perform localization on board space-based instrument?
- Limited computational capacity due to radiation hardening, size, weight, and power constraints, etc.
- But if we *can*, we are able to immediately communicate to space-based and ground-based follow-up instruments!

Let's reason about real-time localization of transients in a principled way.

What do we want?

- Ability to localize transients in real-time aboard space-based hardware.
- Make hard guarantees about latency using approaches from real-time, cyber-physical, and safety-critical computing.
- Adjust computation for latency guarantees in the face of ***dynamic*** workloads and deadlines.

Dynamic Workloads:

- Amount of data to process may depend on transient's flux, duration, etc.
- Algorithms may change depending on quality of data, other characteristics.

Dynamic Deadlines:

- How long do we have access to communication network?
- Which follow-up instruments are available?
- How far away are they (communication latency)?
- How exciting is this transient?
- How much time do we have for meaningful observations?

Computational requirements and timing constraints may not be known a priori!

Let's characterize the *shape* of the computation *offline* so that we can *adapt online* to achieve expected Pareto-optimal results within the imposed deadline

Case Study: Real-Time GRB Localization Aboard APT

The Advanced Particle-astrophysics Telescope (APT) is a future space-based observatory that will detect and localize GRBs in real time to enable concurrent, multi-messenger observations from any direction with minimal delay. For these soft transients, Compton-regime gammas should dominate the emission spectrum. We have therefore designed a parallel computational pipeline for real-time multi-Compton reconstruction and GRB localization. To keep latency low, this will execute fully onboard the instrument, which imposes significant size, weight, and power constraints.

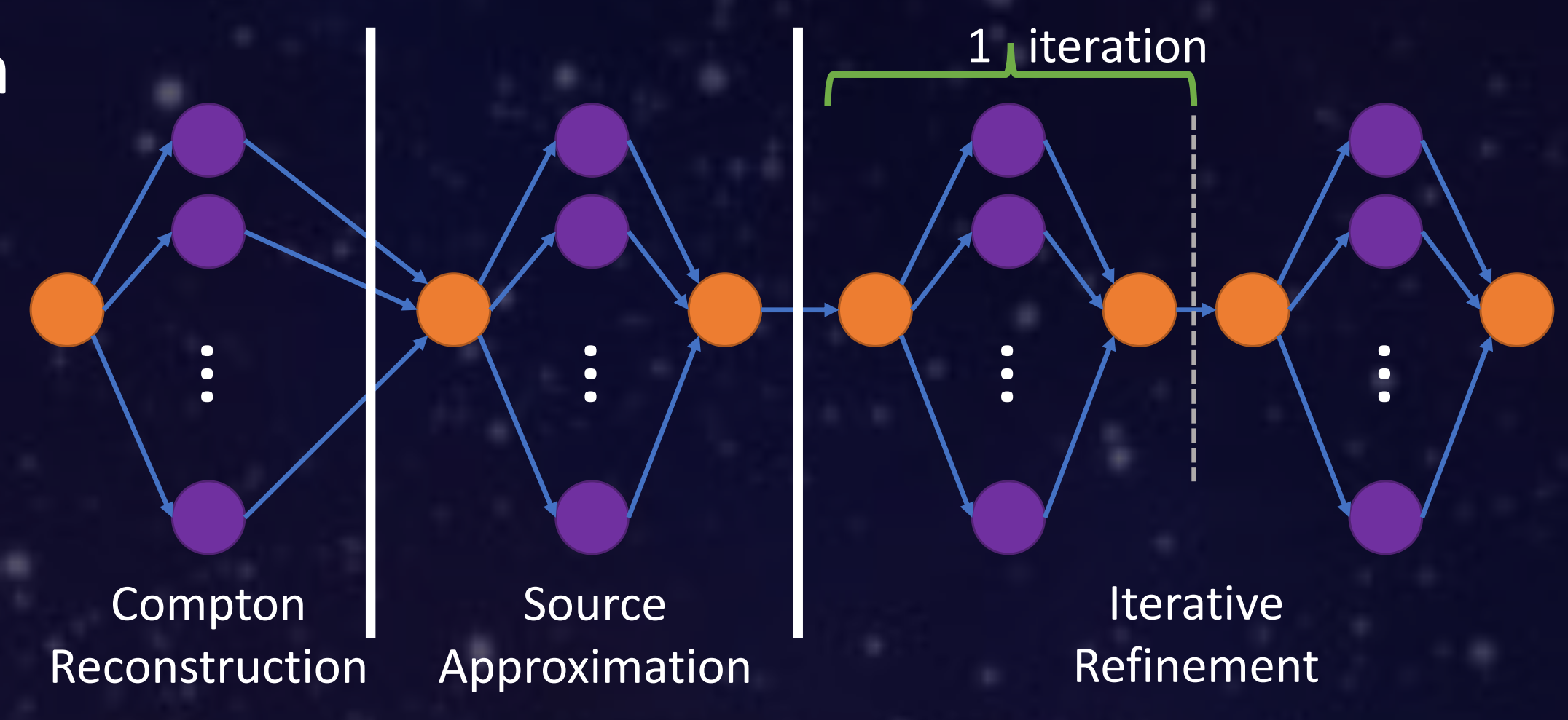
Mission Details

https://adapt.physics.wustl.edu/

Please visit poster #255, "A Computational Pipeline for Prompt Gamma-Ray Burst Localization Aboard APT and ADAPT"

APT simulation model from:

W Chen, et al. "The Advanced Particle-astrophysics Telescope: Simulation of the Instrument Performance for Gamma-Ray Detection." In PoS(ICRC2021), volume 395, pages 590:1–590:9, July 2021.



Challenge: Dynamic Workloads and Deadlines

Every GRB is unique!

Brightness (10^3 – 10^6 incident gamma rays)

Spectral energy distributions

Initial burst durations (10 milliseconds – 20 minutes)

Workload depends on

The number of gamma rays entering the detector

Their physical interactions

Deadline may depend on

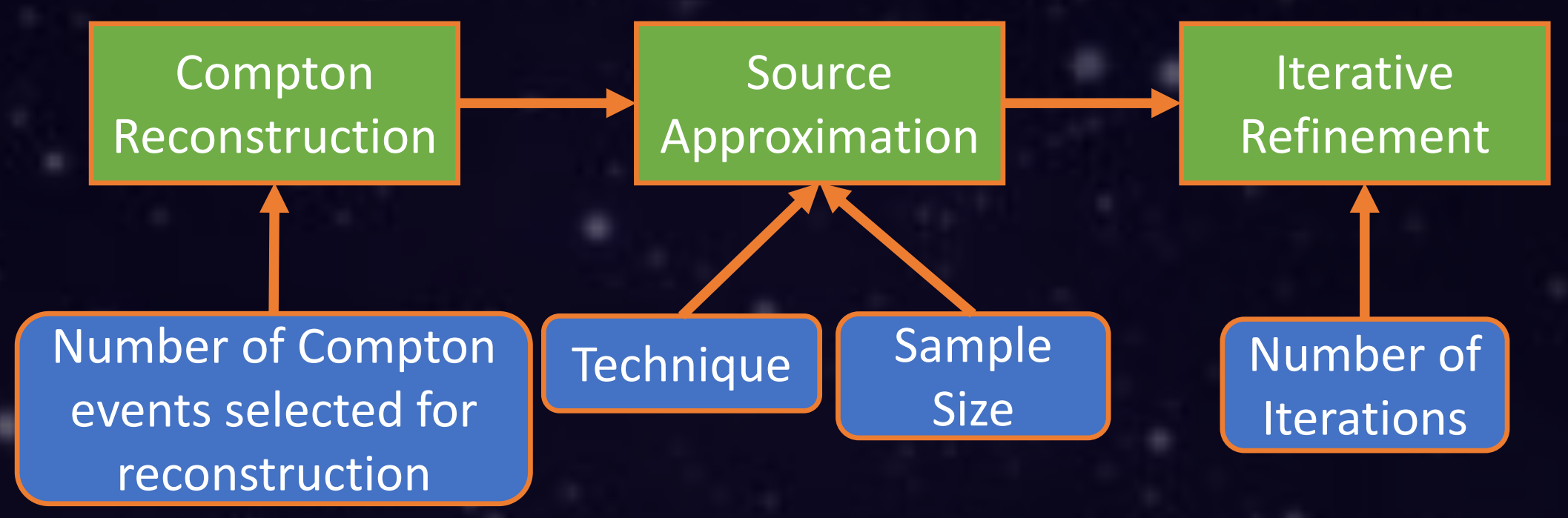
The duration of the burst

Availability of follow-up instruments

How can we *adapt* and *compress* the computational pipeline to maximize localization accuracy even for bright transients while guaranteeing short deadlines?

1 Identify Parameters

Identify the parameterized degrees of freedom over which computational workload may be compressed (i.e., reduced in a way that minimizes loss)



2 Characterize Loss

Identify the impact of reducing the computational workload over its multiple dimensions to construct an objective function for constrained optimization.

Through extensive simulation with synthetic bursts, we characterize loss as two monotonically-decreasing hulls of hyperplanes in the 5 input dimensions.

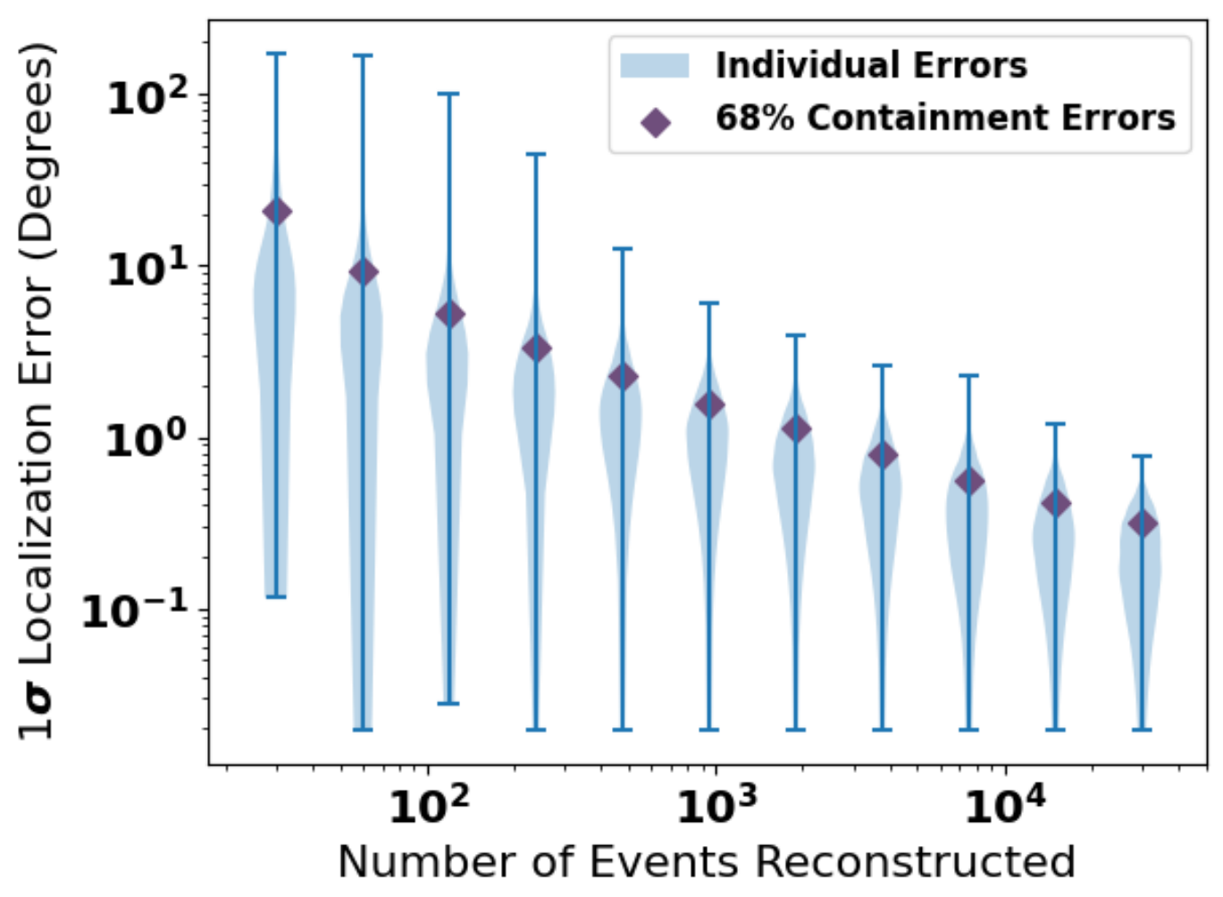
For APT, loss is 68% containment (1σ) localization error (degrees)

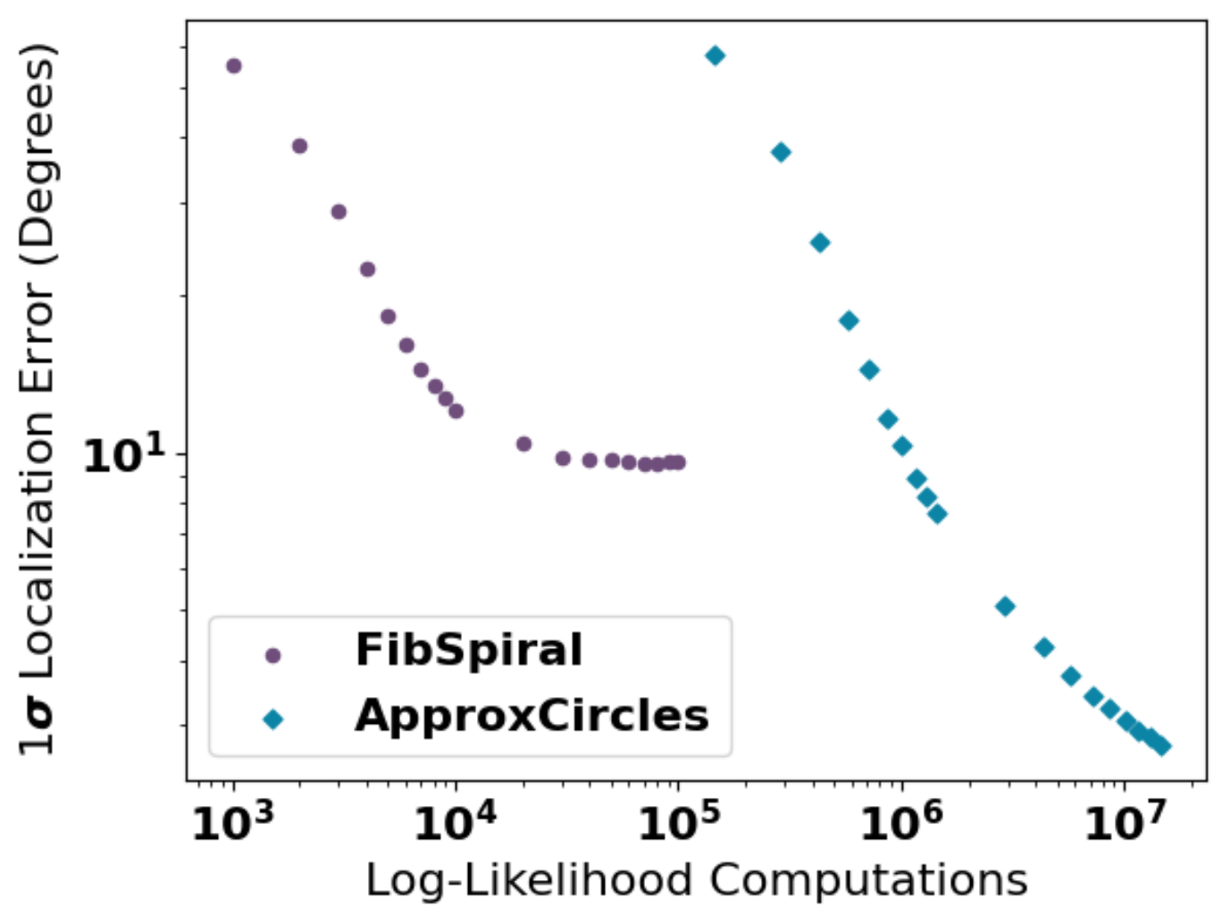
Approximation Techniques

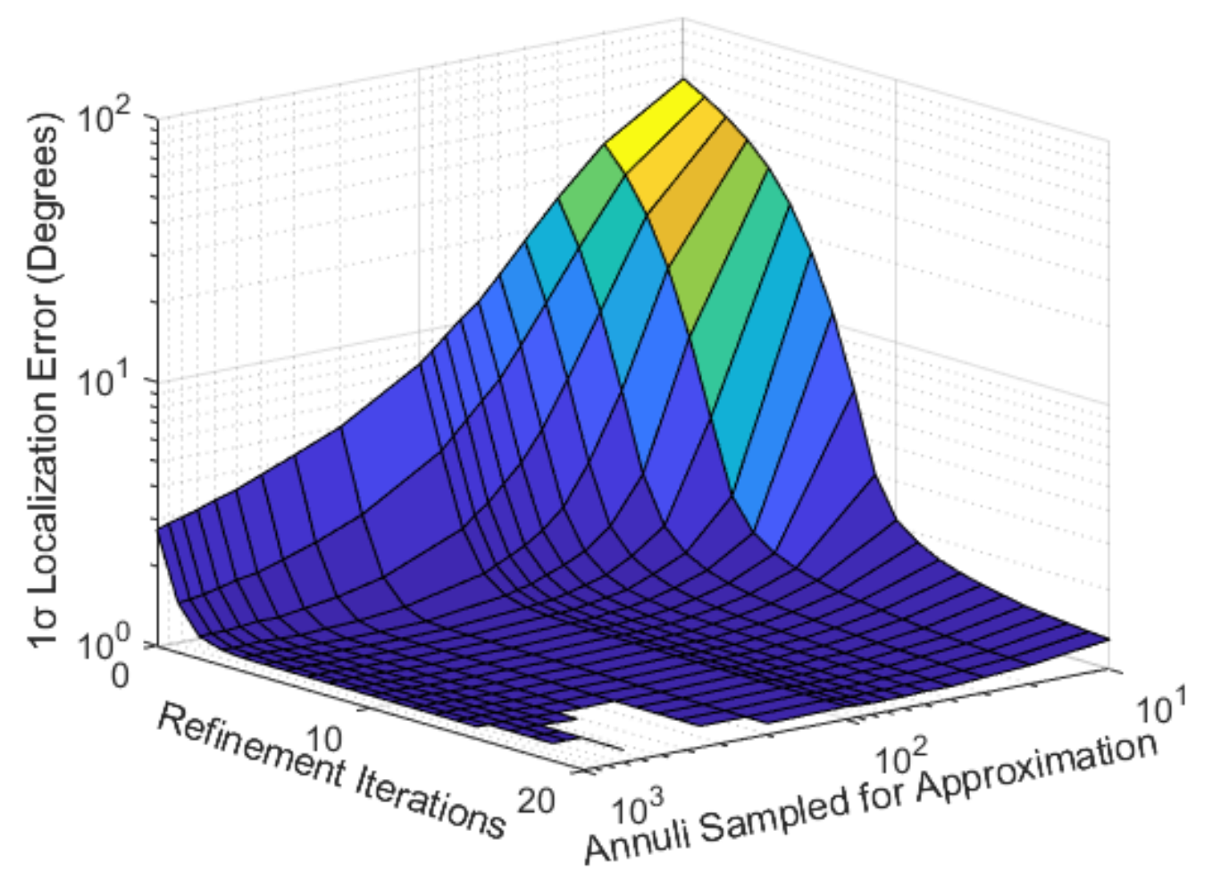
Sample n_s reconstructed Compton rings for approximation.

Approx Circles: Randomly select 20 rings from n_s and uniformly distribute 720 points around each. Find the point on each ring with the greatest joint log-likelihood with respect to all n_s rings. Weighted mean over those points approximates the source.

Fibonacci Spiral: A fast but less accurate technique. Distribute 100 points uniformly over the surface of the unit sphere. Find the joint log-likelihood of each point with respect to all n_s rings. Weighted mean over the top 10 approximates the source.







3 Quantify Response Time

For a highly-parallel fork-join task like GRB localization, worst-case response time can be quantified by decomposing it into constituent subtasks, then profiling execution times as functions of input parameters

Response Time Expression

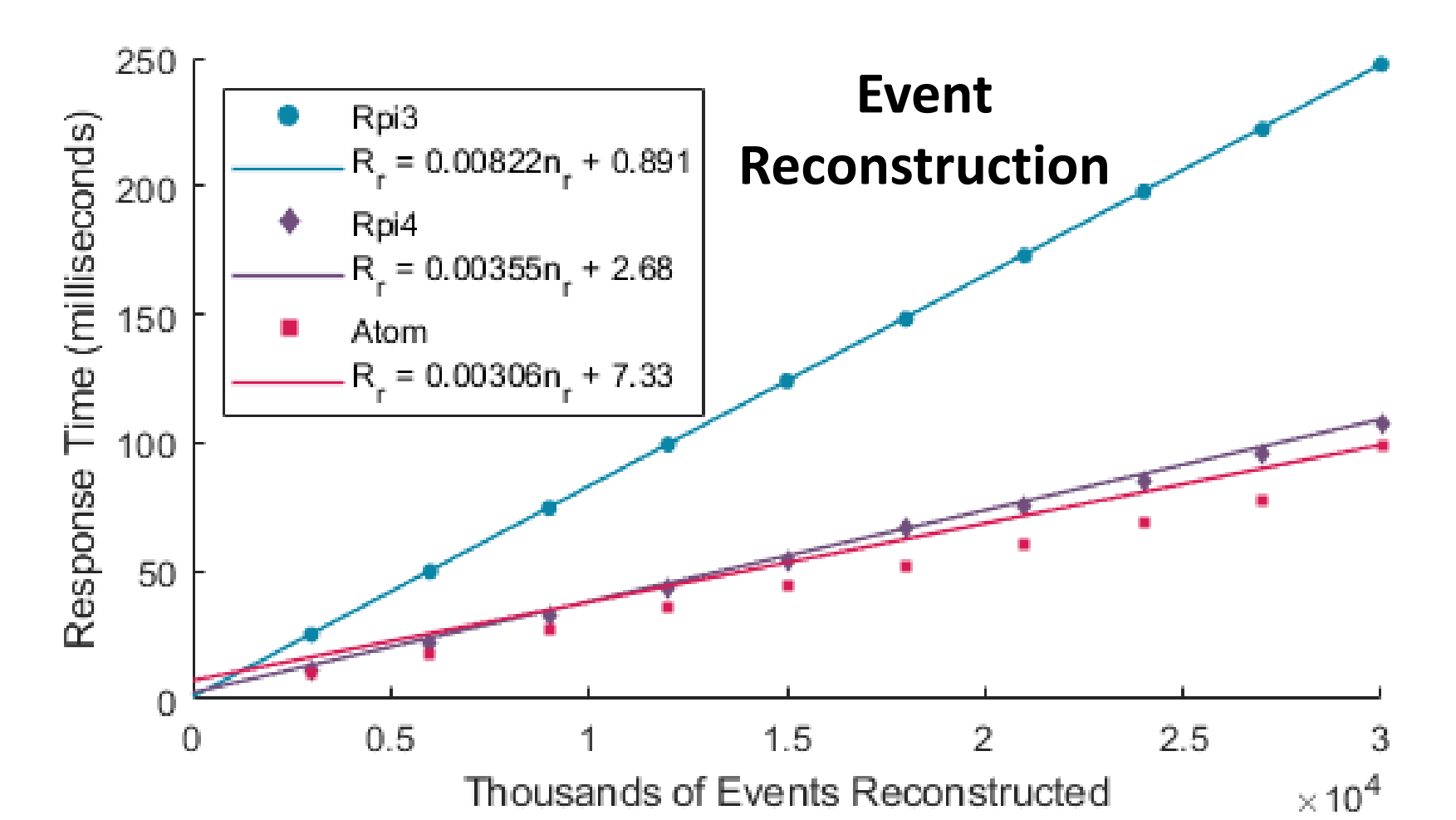
$$R = \sum_{\tau_i \in \boldsymbol{S}} C_i + \sum_{\tau_i \in \boldsymbol{S}} \frac{C_i}{n}$$

- τ_i: a subtask
- $\boldsymbol{S}$: the set of sequential subtasks
- $\boldsymbol{P}$: the set of highly-parallel subtasks
- $\{a_j\}$: the set of adjustable workload parameters
- $C_i(\{a_j\})$: the execution time of τ_i on a single core given a set of assigned parameter values
- $R_i(\{a_j\})$: the response time of the task for the given set of assigned parameter values
- $\boldsymbol{n}$: the number of CPU cores

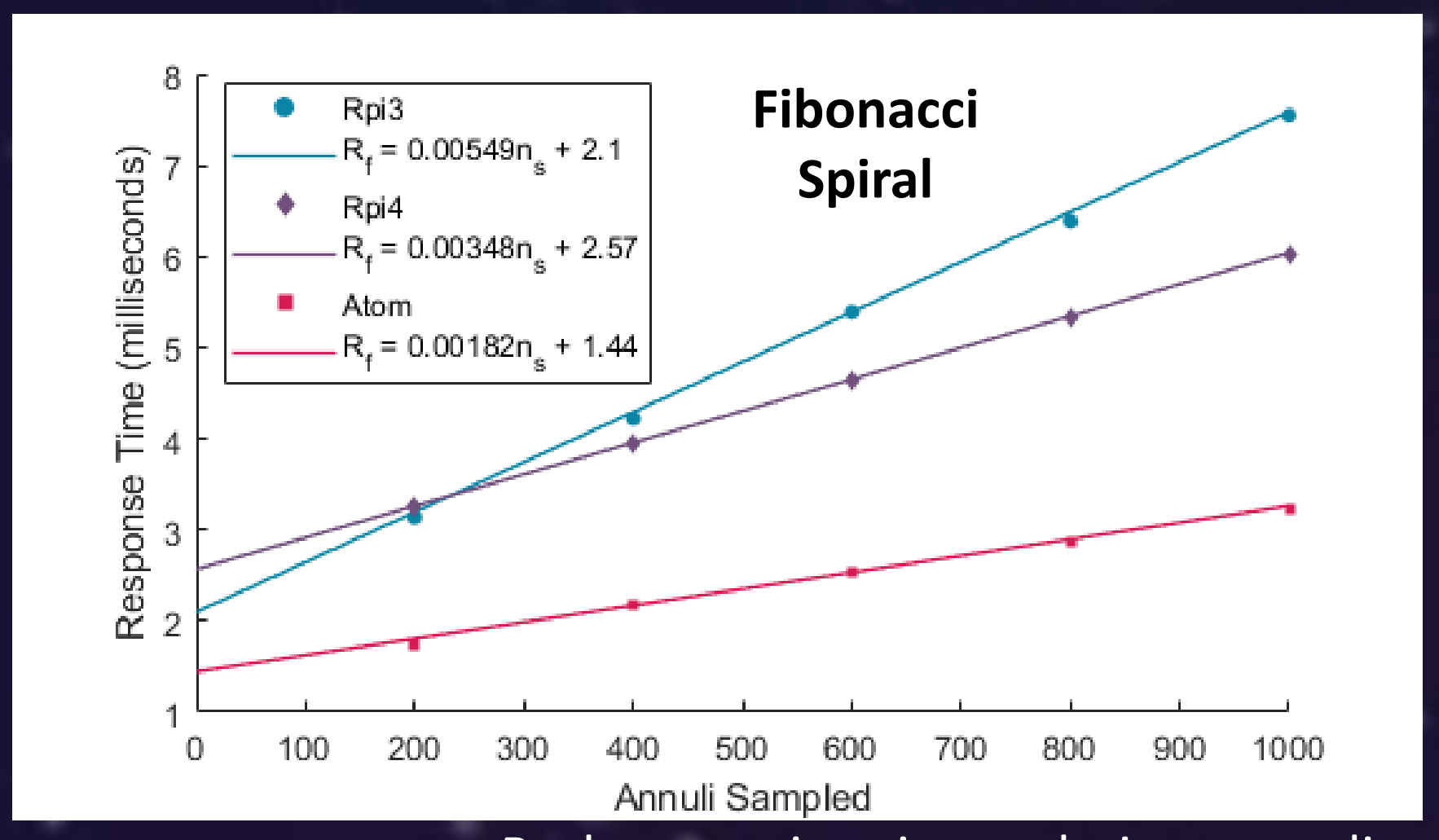
Platform	Abbr.	CPU	Freq
Raspberry Pi 3B+	RPi3	4-Core Cortex-A53	700MHz**
Raspberry Pi 4B	RPi4	4-Core Cortex-A72	600MHz**
Winsystems EBC-C413*	Atom	4-Core Intel Atom E3845	1.92GHz

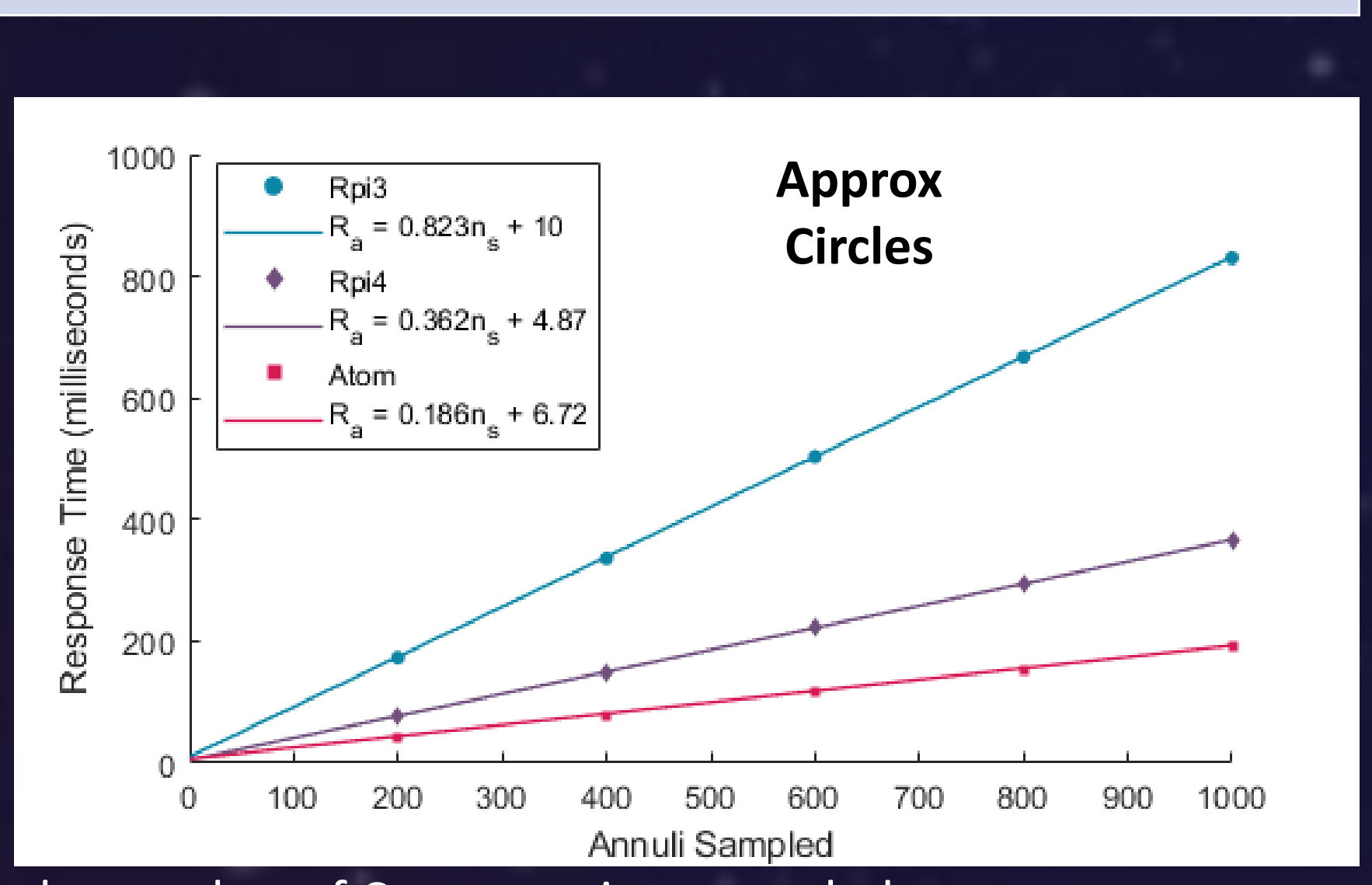
* Will fly on APT's high-altitude Antarctic demonstrator (ADAPT)

** Lower frequencies prevent thermal throttling and instability in power-constrained environments

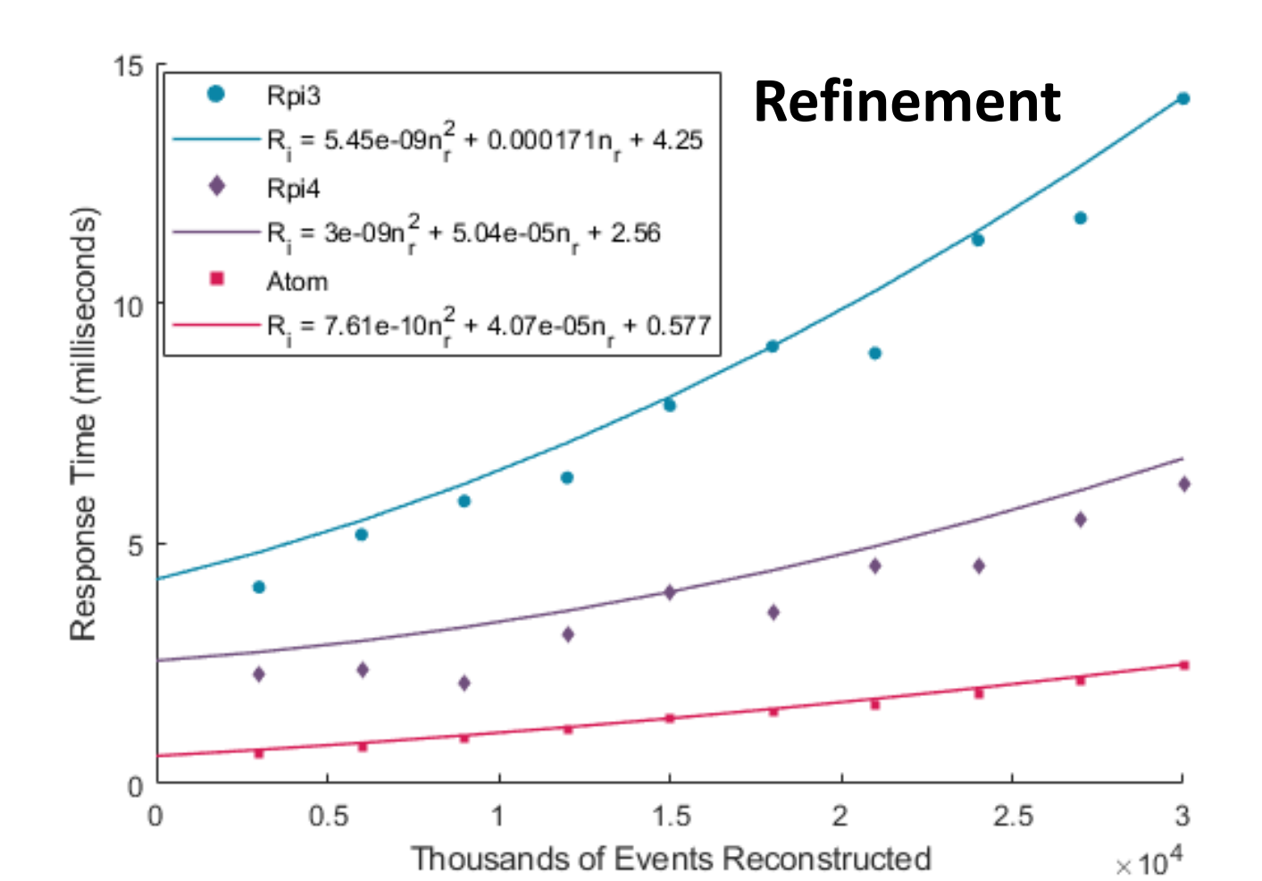


Reconstruction is linear in the number of Compton events selected





Both approximation techniques are linear in the number of Compton rings sampled



Each iteration of refinement is quadratic in the number of Compton events selected

4 Generate Pareto-Optimal Surface

Sort candidate states by response time, discarding any with a higher loss than a previous state. We are left with just those state for which *more* execution results in *better* expected outcome.

2657 initial candidate parameter sets → **~80** parameter sets in Pareto-optimal subset

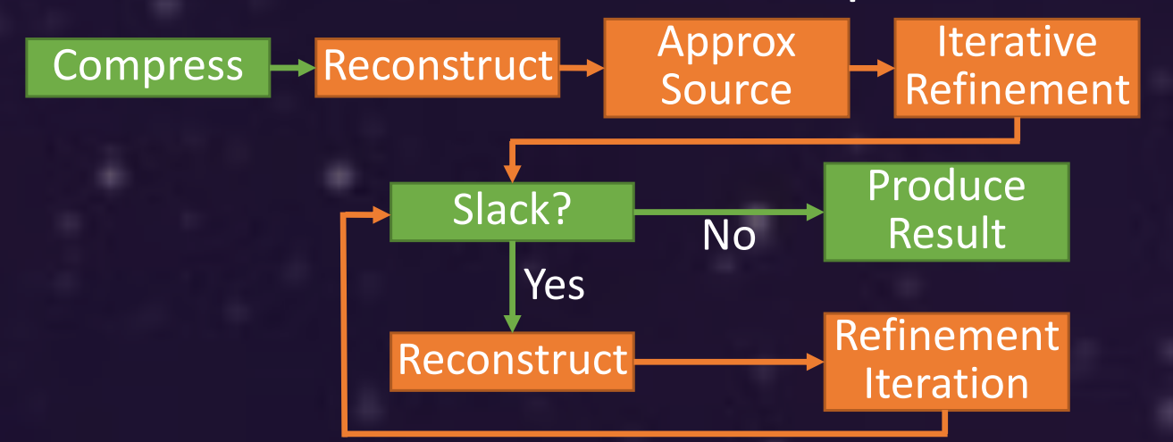
5 Online Solution Search

When a transient appears, determine workload (based on quantity of data) and deadline, then adapt computation parameters to Pareto-optimal selection

1. Binary search over Pareto-optimal subset for best set of parameters not exceeding deadline
2. Data structure includes gradients for linear interpolation/extrapolation to exactly meet deadline (we use log-linear interpolation)

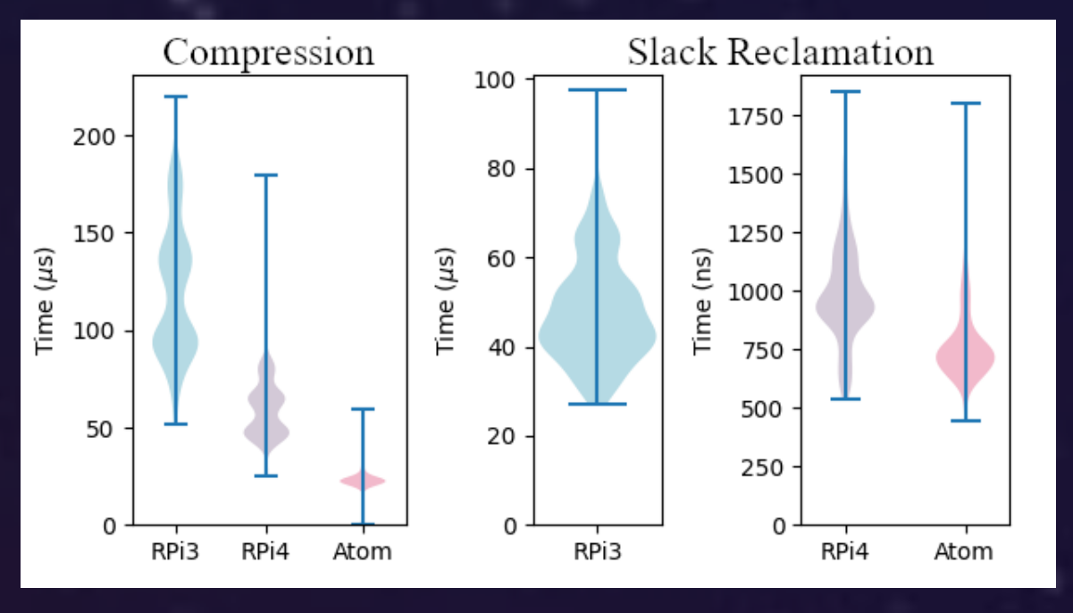
6 Reclaim Slack

We use worst-case response times to guarantee we meet dynamic deadlines. But if we complete early, slack time remains. We can reclaim slack via further computation



Low Overheads

By constructing a Pareto-optimal surface *offline*, we can adjust *online* with low overhead. Keeps CPU free for the actual science!



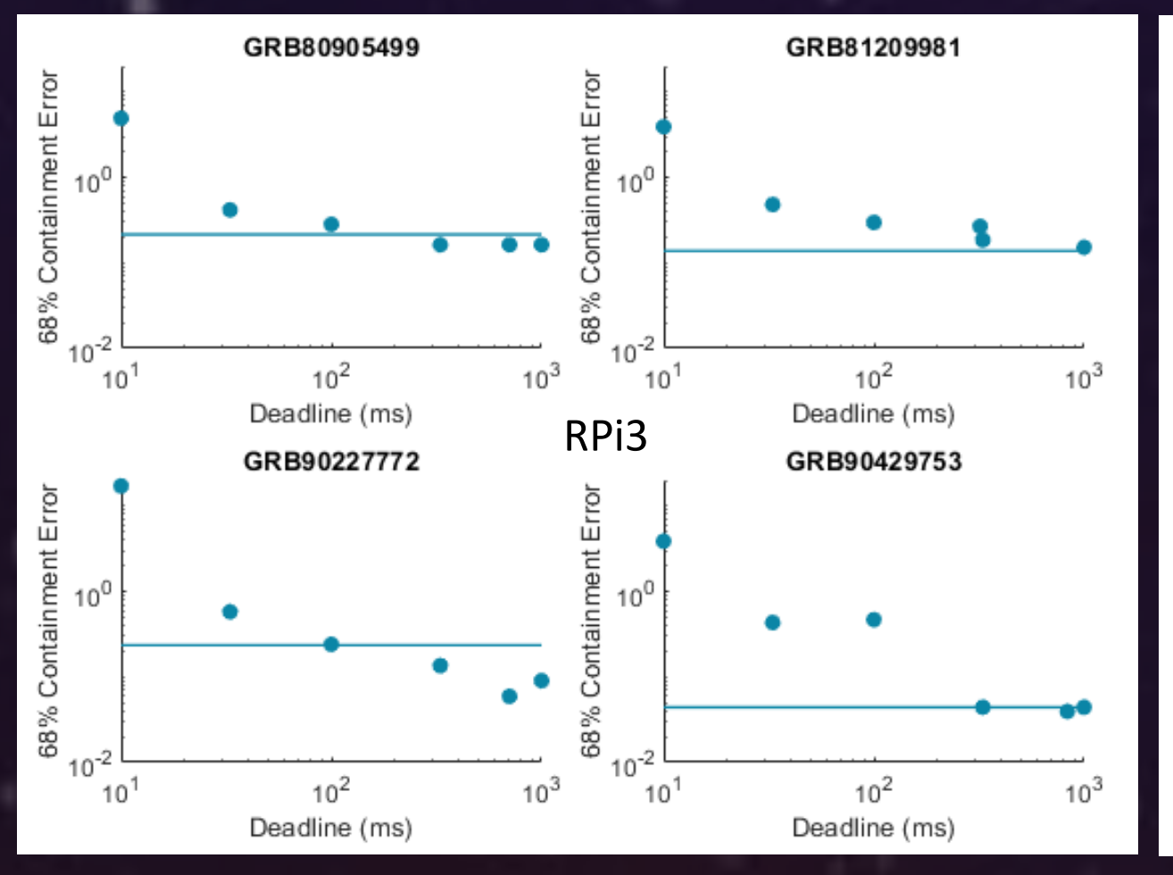
Localization Results

GEANT-based simulation of 4 short GRBs observed by Fermi GBM with fluence and Band function spectral parameters taken from

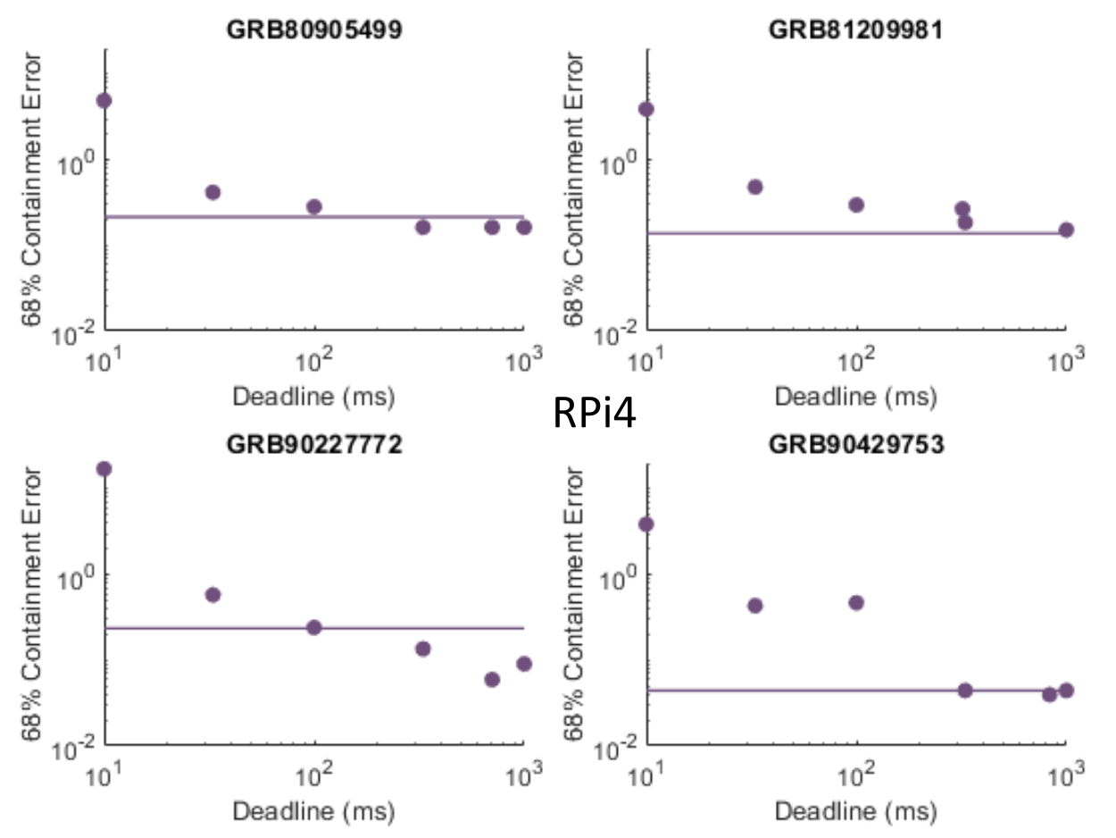
L. Nava, G. Ghirlanda, G. Ghisellini, and A. Celotti, "Spectral properties of 438 GRBs detected by Fermi GBM," Astronomy & Astrophysics, vol. 530, p. A21, Apr. 2011.

Tested localization accuracy when adapting to short imposed deadlines

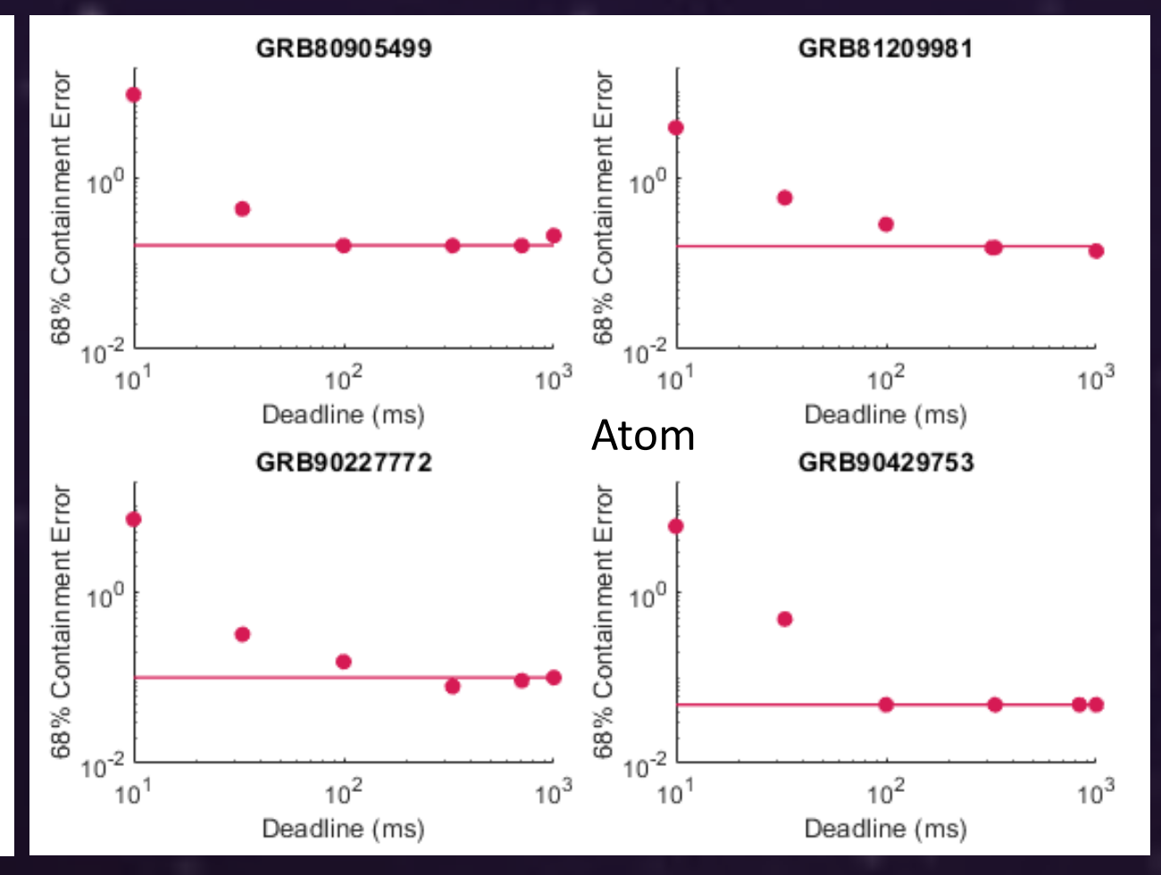
Our approach enables sub-degree localization even for 33ms deadlines!



RPi3



RPi4



Atom

Final Thoughts

More details can be found in our paper:

Marion Sudvarg et al. "Parameterized Workload Adaptation for Fork-Join Tasks with Dynamic Workloads and Deadlines." RTCSA 2023.

These techniques are not just for GRBs!

Can we apply to search for optical counterparts of FRBs?

Let's talk about how these ideas can extend to your application!

Let's also talk about accelerating your application on heterogenous multicore, GPU and FPGA architectures

Please also visit poster #226, "Accelerating Compton Imaging of Astrophysical Sources in Python"

This work funded by NASA grant 80NSSC21K1741 and NSF grants CSR-1814739, CNS-17653503, and CNS-2229290 (CPS)

Background courtesy Pikbest.