Applications of IaaS in Gravitational Wave Astronomy

Ethan Marx[1*], Will Benoit[2], Deep Chatterjee[1], Alec Gunny[1], Katya Govorkova[1], Eric Moreno[1], Rafia Omer[2], Ryan Raikman[1], Muhammed Saleem[2], Michael Coughlin[2], Philip Harris[1], Erik Katsavounidis[1],

* - Presenter
1 - Massachusetts Institute of Technology
2 - University of Minnesota

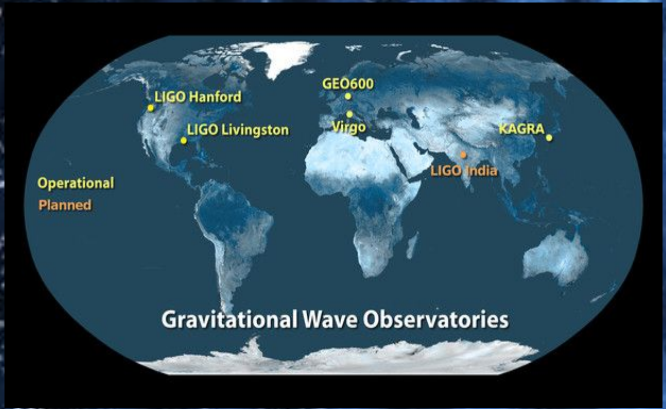
Gravitational Wave Astronomy

Large scale astrophysical events ripple the fabric of spacetime

Detect with (for now) ground-based interferometers

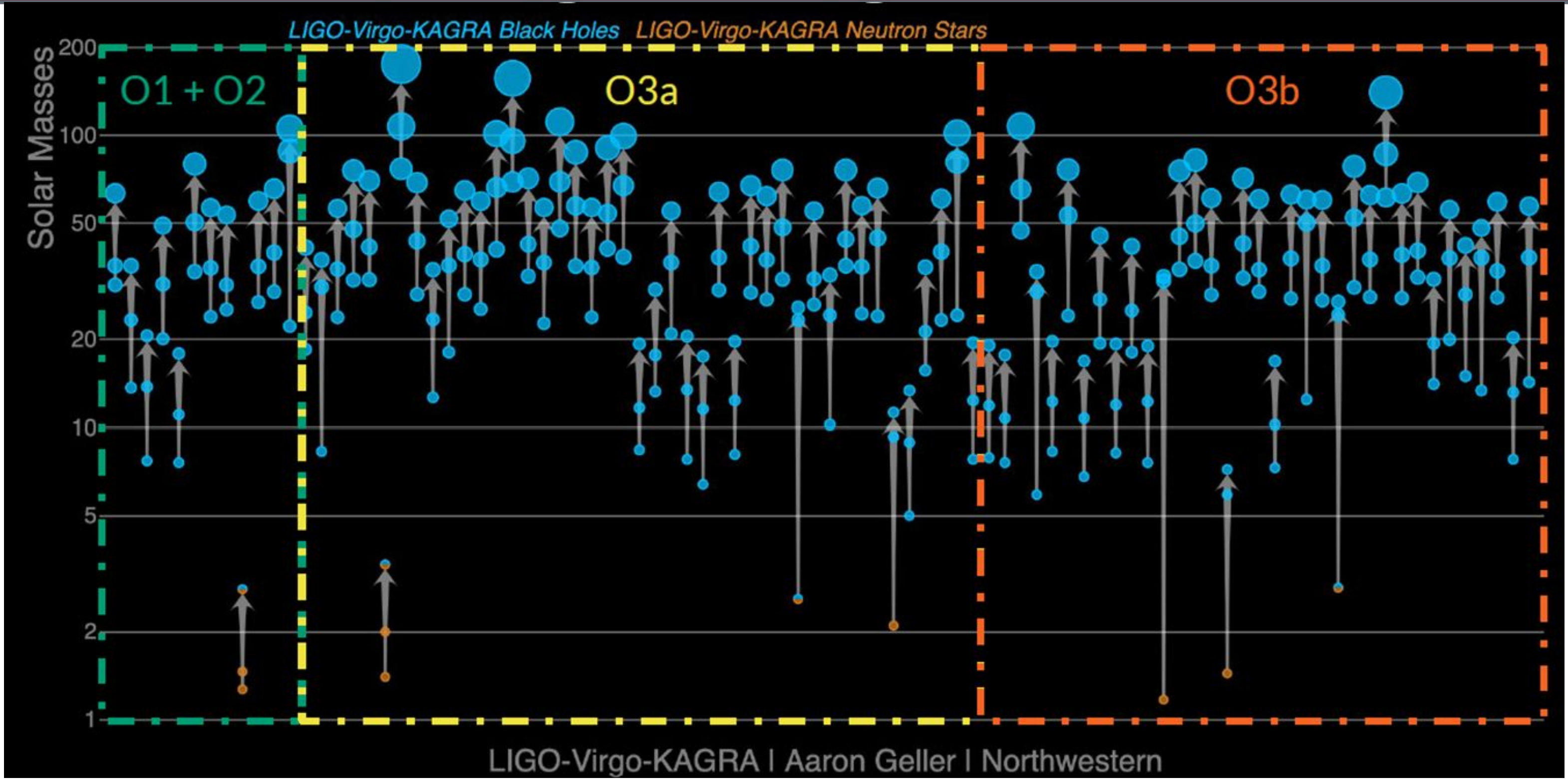
Hanford, Washington

Livingston, Louisiana



Network of observatories all over the world

Third transient event catalog: GWTC-3



11 events from O1+O2

44 events in O3a, 55 total
1041 "subthreshold" events in O1,O2,O3a

35 events in O3b, 90 total
(catalogs are cumulative)

Gravitational Wave Astronomy - LIGO Data

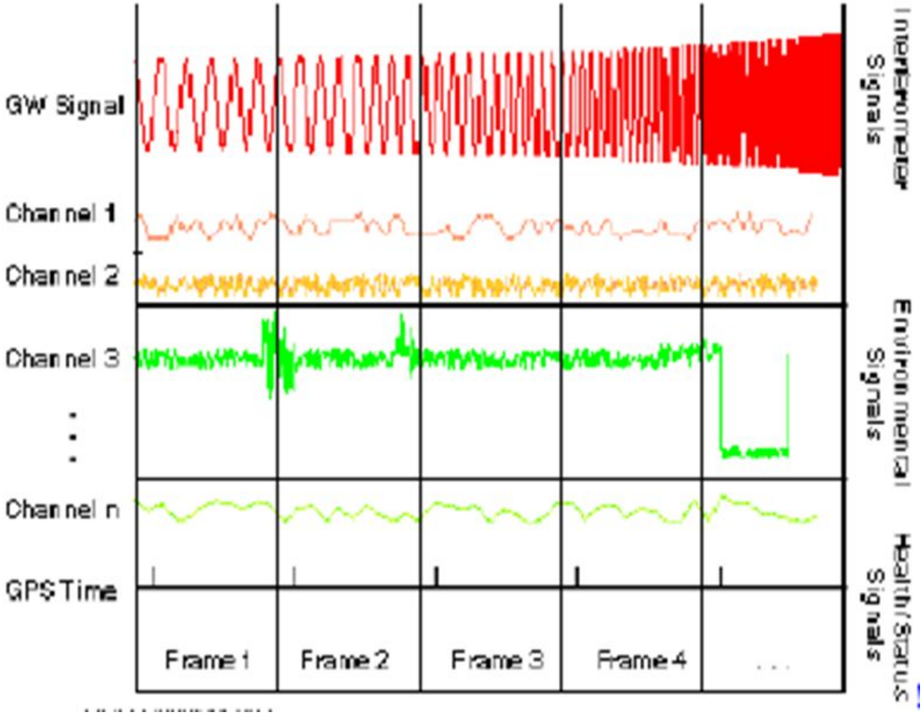
Continuous **time series** (1Hz, 128Hz … 16kHz)

Gravitational Wave channel: ~20GB/day (per instrument)

Physical Environment Monitors (seismometers, accelerometers, magnetometers, microphones etc)

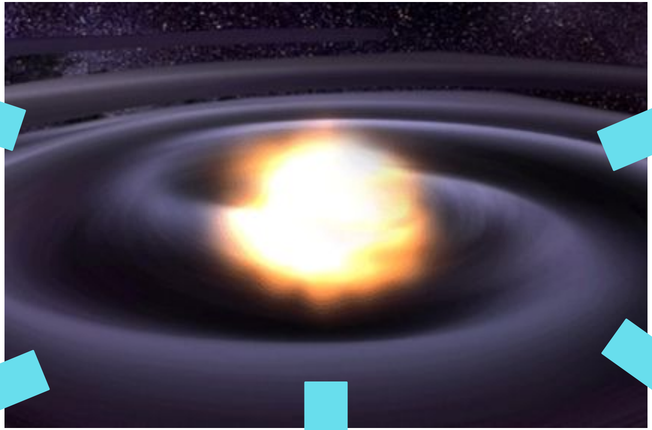
Internal Engineering Monitors (sensing, housekeeping, status etc)

Together with various intermediate data products >2TB/day (per instrument)

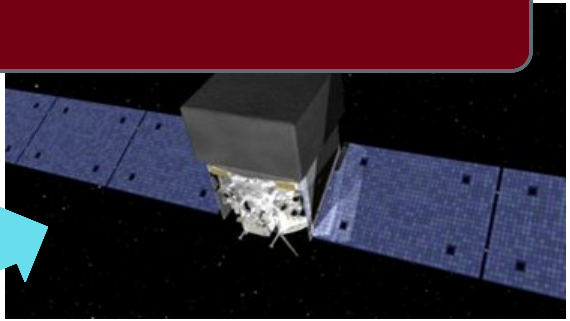


Initial and Enhanced LIGO archive (2002-2010) exceeds 1PB of data

Multi-messenger Astronomy



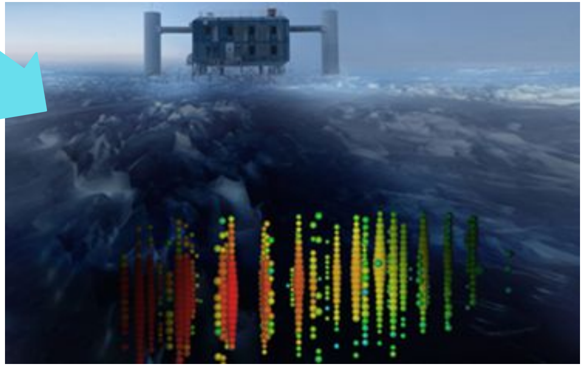
Gravitational waves



X-rays/Gamma-rays

Visible/infrared light

Radio waves



Neutrinos

Machine Learning in GW Astronomy

Online

Real-time analysis with goal of alerting electromagnetic astronomers (MMA) of significant events

Detect events → Localize on Sky → Send public alerts

Main consideration is ***latency***

Offline

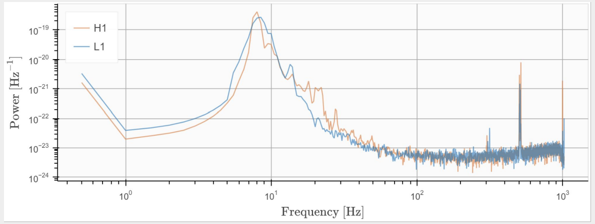
Large scale analysis of archival data for

- End to end searches
- Validating new methods, performing new research

Main consideration is ***throughput***

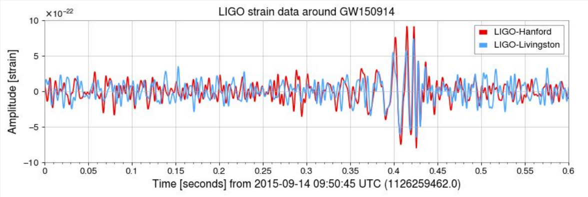
Machine Learning in GW Astronomy

Detector Characterization



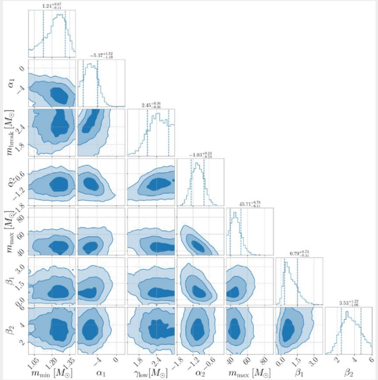
***[DeepClean](#)*: Noise regression from auxiliary channels using autoencoders**

Event Detection



[Aframe](#): Detecting CBC events in low latency with supervised neural networks

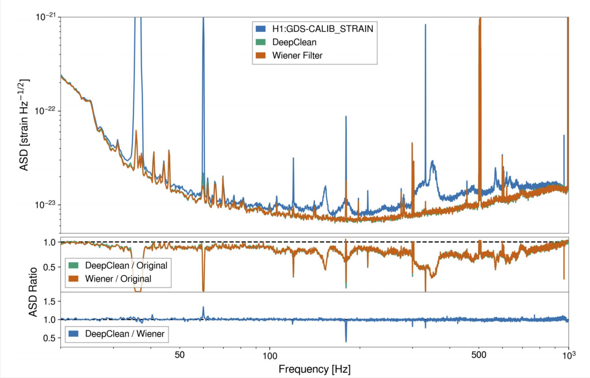
Event Characterization



***[Parameter estimation](#)*: Characterizing source parameters with Normalizing Flows**

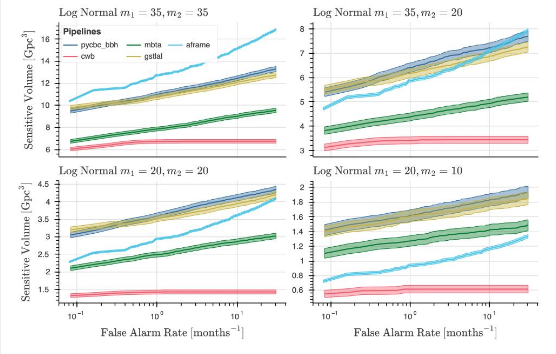
Initial Success

DeepClean



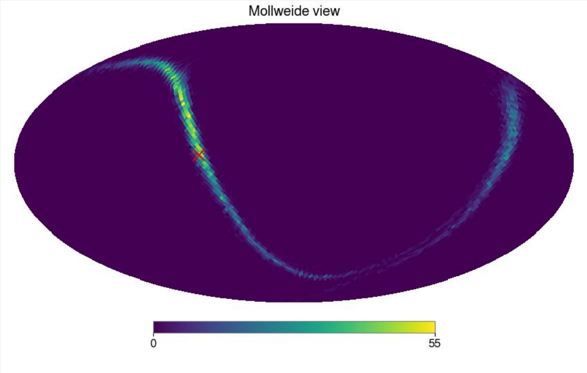
Offline regression of 60Hz power line

Aframe



Comparable Sensitivities with matched filtering pipelines over the O3 observing run

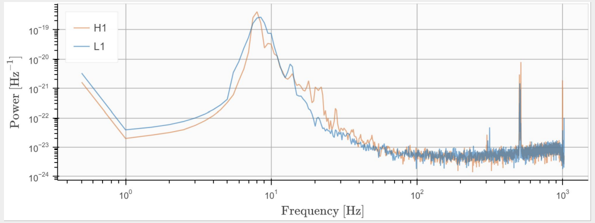
Parameter Estimation



Success estimating sky localization using generic templates

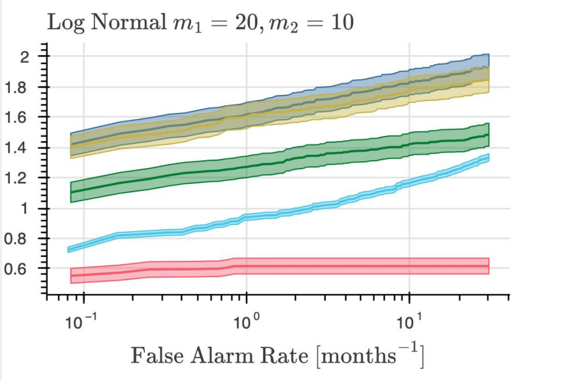
Not Without Limitations

DeepClean



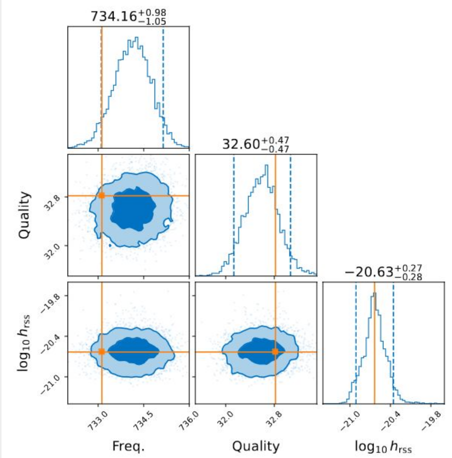
Can DeepClean solve other known noise coupling problems?

Aframe



Reduced sensitivities at lower mass ranges

Parameter Estimation



Wider error bars than standard Bayesian methods

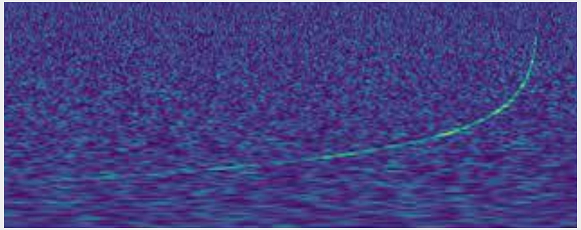
Many Ideas

DeepClean

```python
@dataclass
class Coupling:
    freq_low: float
    freq_high: float
    witnesses: list[str]
```

Investigate complex couplings beyond standard 60Hz problem

Aframe

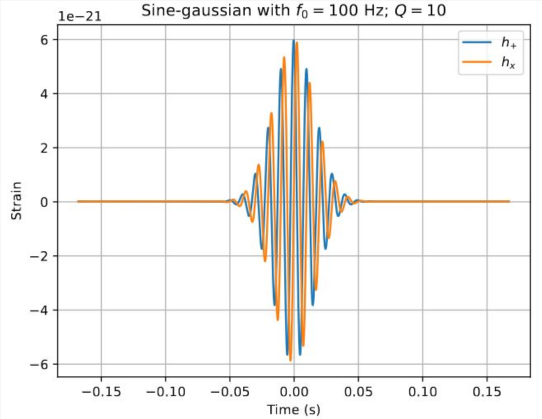


Curriculum learning emphasizing lower mass ranges

Different architectures

Spectrograms to reduce data dimensionality

Parameter Estimation



Larger models

Frequency domain vs time domain

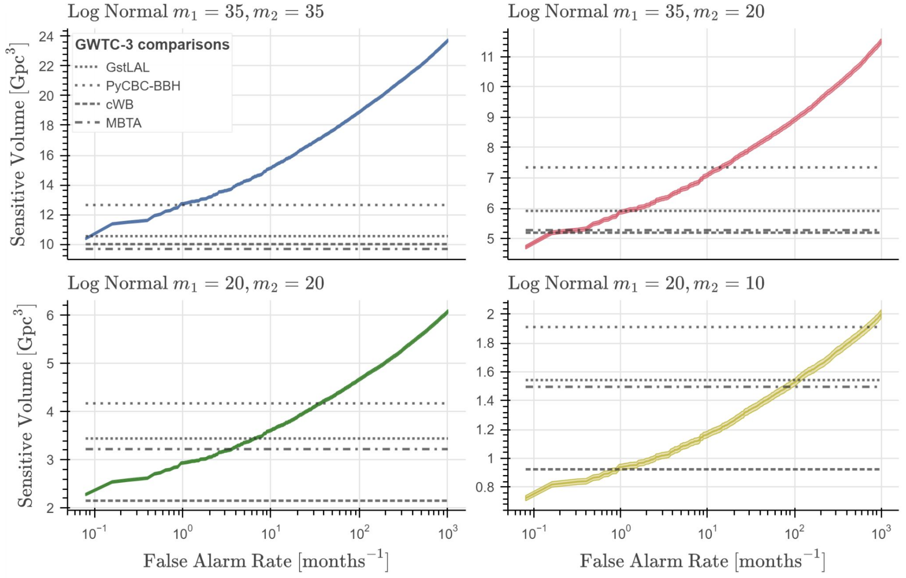
Aframe - Sensitive Volume

Events are assigned a false alarm rate (FAR) by analyzing background timeslides

$$\frac{\max(\sum_i^{N_b} \mathbb{I}[\eta_i \geq \eta], 1)}{T_b}$$

Sensitive Volume -
detection algorithms effective "reach" to some population of sources at a given false alarm rate (FAR)

To get O(years) significance, need O(years) background



Aframe - Vanilla Inference Deployment

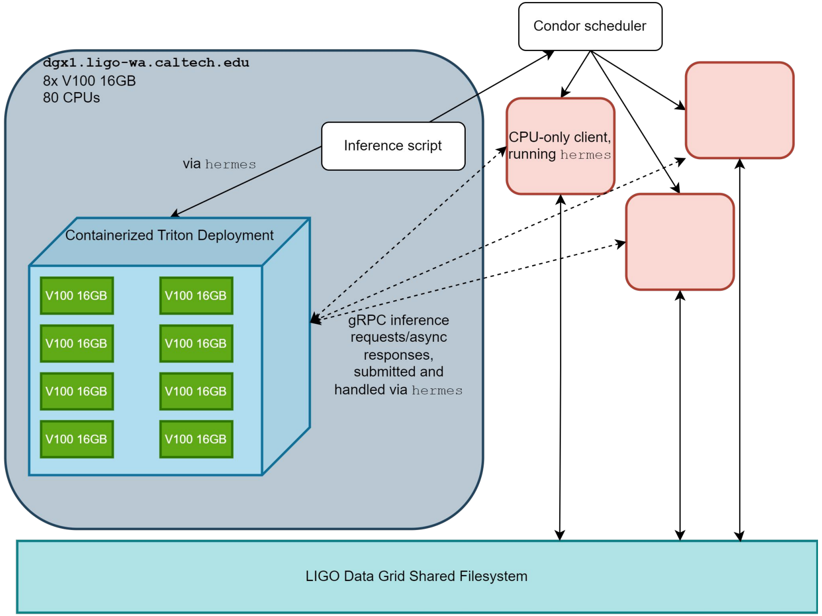
- Load `torch` model in memory, shove data through it
- Can process ~512 seconds of data per second (s' / s) on single 16GB V100
- 1yr of background = 17hrs of compute → 70 days to get 100 yrs!
- Not quick enough for iterating on ideas

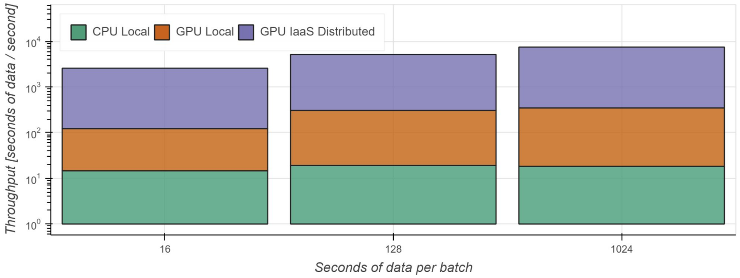
Aframe - Local IaaS

Deploy inference service locally, bombard with requests from clients

Throughput scales nearly linearly to ~3800 s'/s

Suboptimal due to FP16 issues, lazy client:GPU ratio strategy





Streaming IaaS - Snapshotter

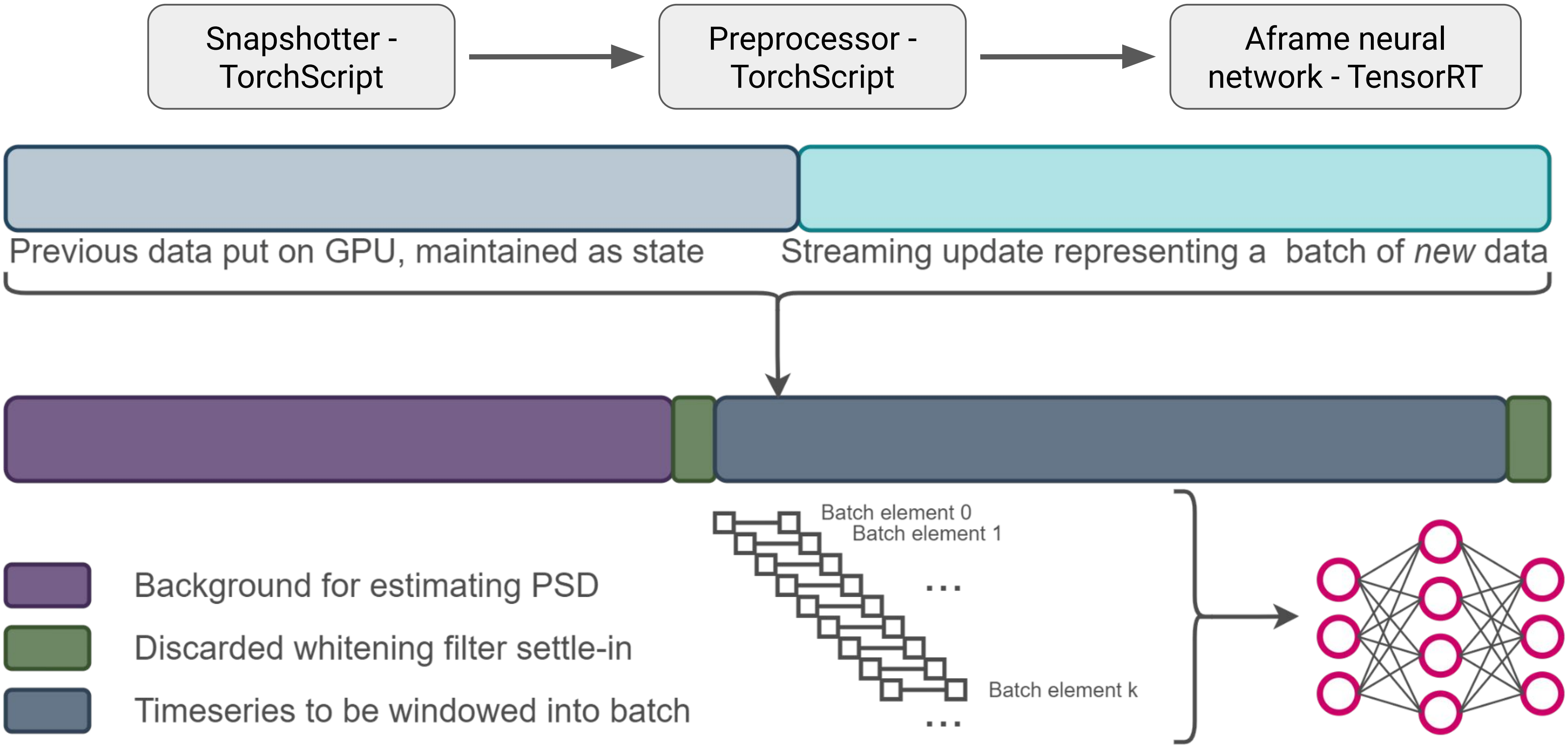
Most GW use cases benefit from inference on overlapping data

Creates redundant network I/O that can bottleneck IaaS deployments

Snapshotter maintains state → only send required updates

ml4gw library offers implementations of some basic stateful steps easy to build off for more custom needs

Aframe - IaaS Deployment



hermes IaaS made simple

https://github.com/ML4GW/hermes

Export → Acceleration → Deployment → Inference

Export

- Managing model repository
- Pythonic interfaces to protobuf configs
- Simple support for stateful streaming models
- Supports Torch and TensorFlow export

Acceleration

- Conversion of Torch models to ONNX
- ONNX → TensorRT conversion with FP16 support

Deployment

- Python contexts for deploying a local inference service
- Throughput and latency metrics monitoring service

Inference

- Asynchronous inference request submission and response handling
- Input/output shape/dtype inference

Nautilus Computing Cluster

LIGO Data Grid (LDG)

LIGOs computing ecosystem of mostly CPU resources

Limited GPUs, workloads not scalable

GPUs (currently) not exposed to condor scheduling system

Wild west: Submit GPU jobs from head nodes, first come first serve

Nautilus HyperCluster

Collection of computing clusters containing 1000s of GPUs

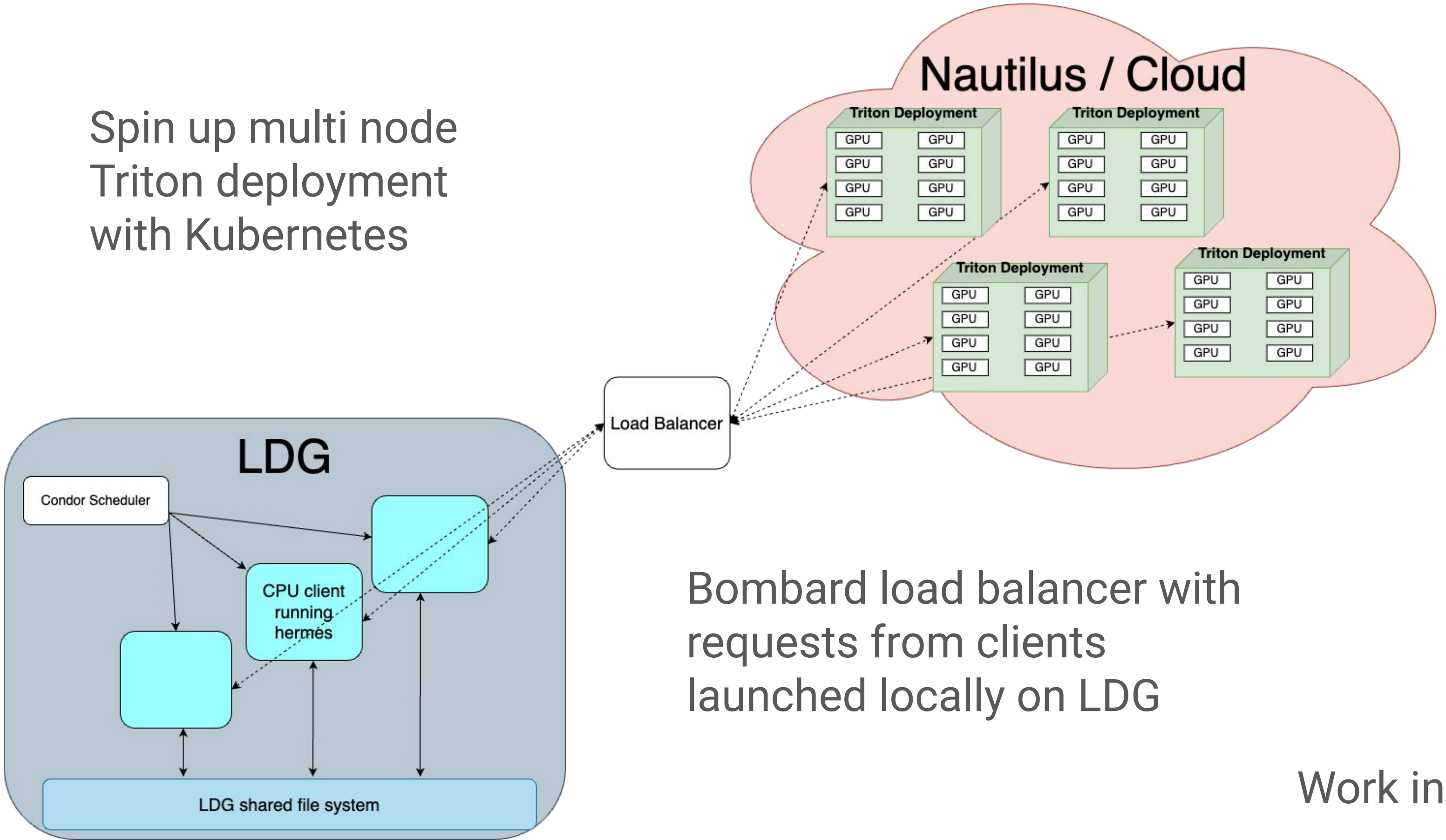
Containerized workloads

Trivially scalable with Kubernetes

With Kubernetes infra, can easily migrate to other cloud resources

Looking Ahead - Remote Distributed Inference

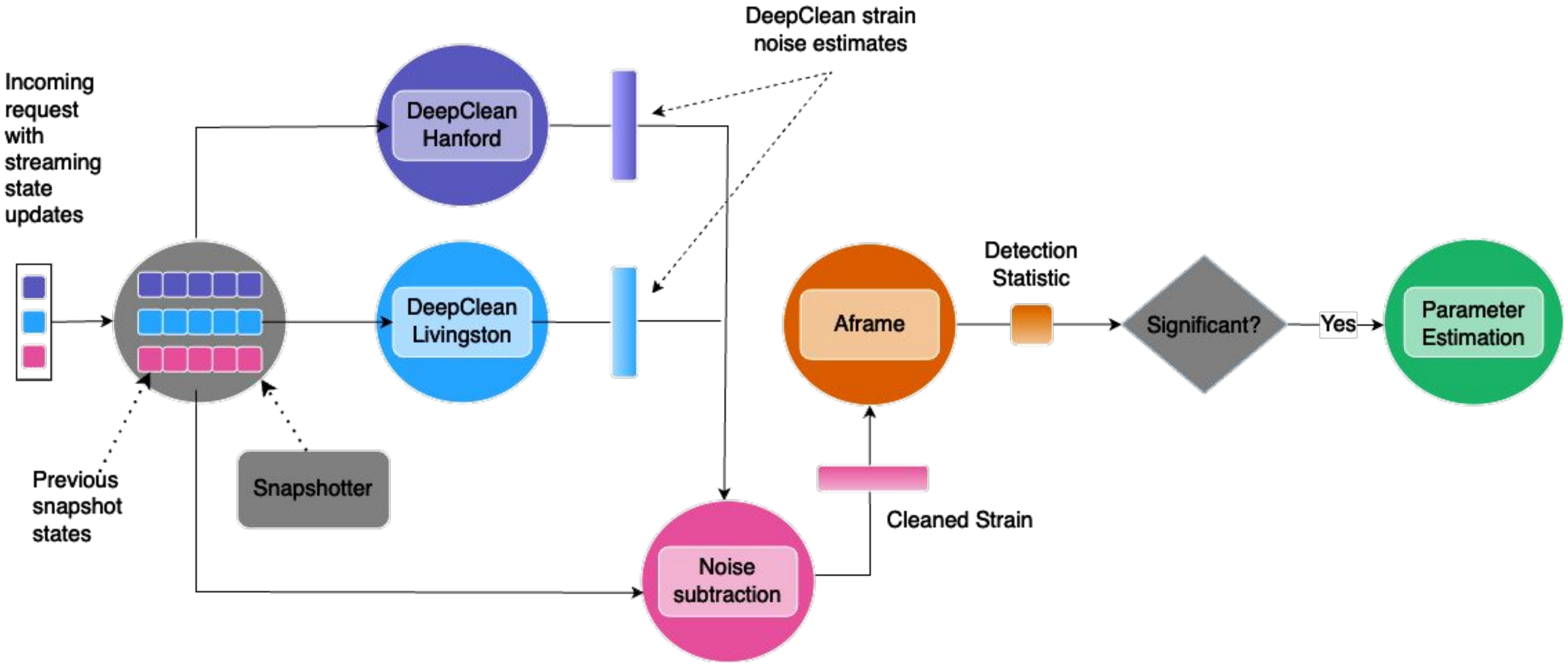
Spin up multi node Triton deployment with Kubernetes



Bombard load balancer with requests from clients launched locally on LDG

Work in Progress

Looking Ahead - Data Analysis Ensemble



Conclusion

ML applications in GW astronomy are becoming production ready

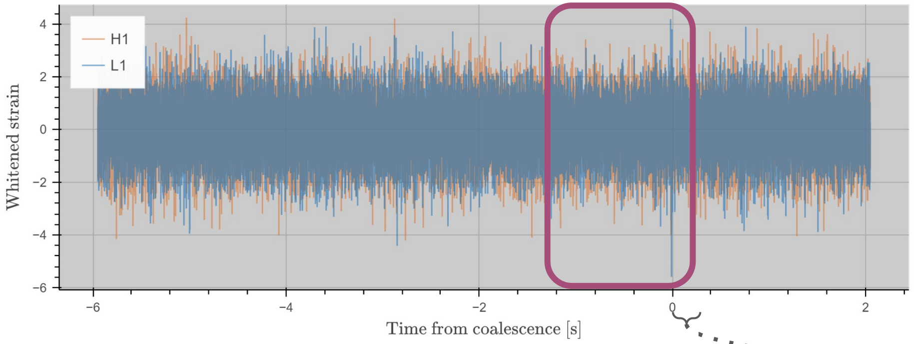
IaaS will play a critical role enabling online and offline use cases

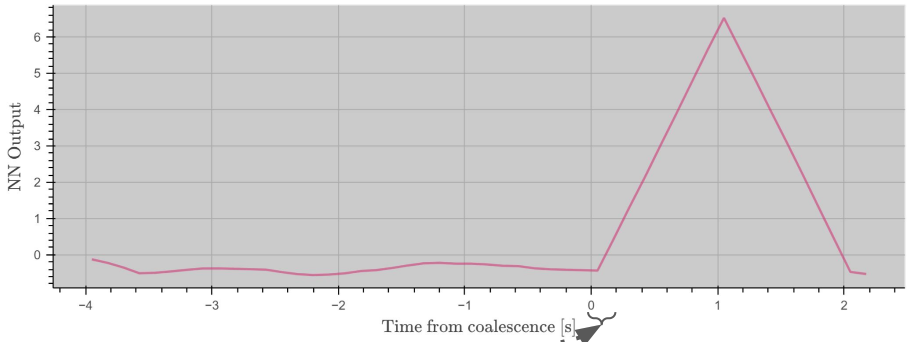
Scaling IaaS deployments will expedite research and time to solution

Thank You

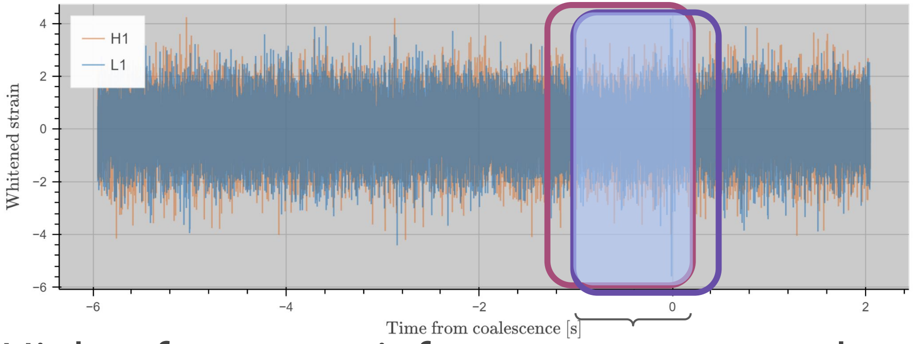
Backups

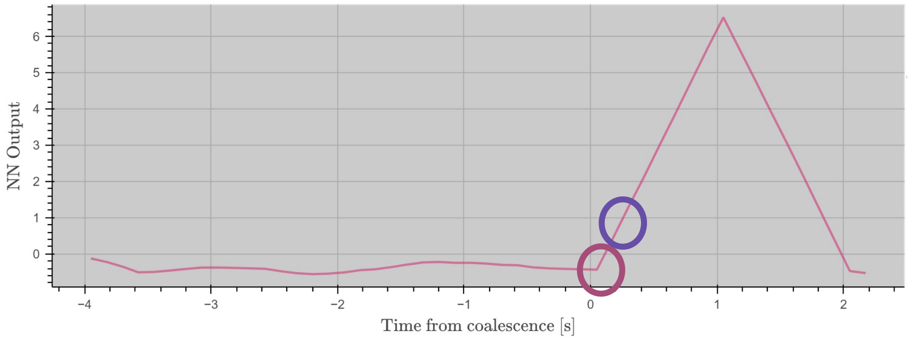
Aframe - Inference





Inference frequency determines resolution of coalescence time recovery





Higher frequency inference means much of input data is overlapping

Inference bottlenecked by data transfer