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

The authors acknowledge NSF grant OAC-2117997

Machine learning infrastructure for gravitational wave physics

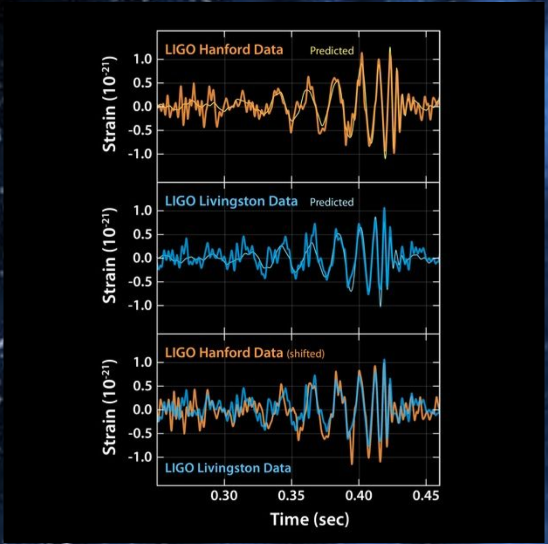
Alec Gunny[*,1], Ethan Marx[1], William Benoit[2], Deep Chatterjee[1], Michael Coughlin[2], Philip Harris[1], Erik Katsavounidis[1], Eric Moreno[1], Dylan Rankin[1], Ryan Raikman[1], Muhammed Saleem[2]

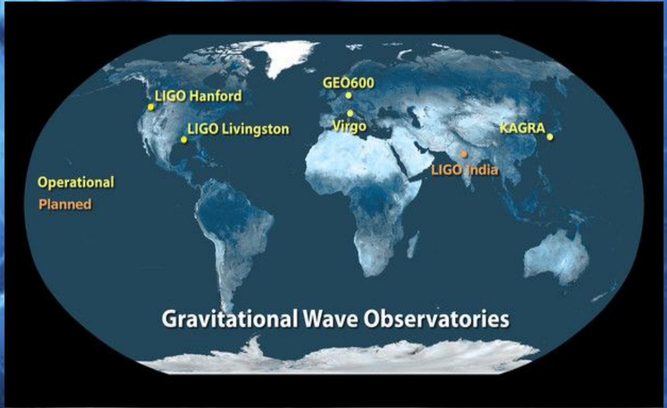
Accelerated AI Algorithms for Data-Driven Discovery

Gravitational wave physics

Large scale astrophysical events ripple the fabric of spacetime

International Gravitational Wave Observatory Network (IGWN) set up to detect, locate, and characterize events





Measure timeseries of unitless quantity - gravitational wave **strain**

Requirements for ML deployment

Training

- Load timeseries data from disk and efficiently move it to GPU
- Leverage simulations to create robust datasets
- Implement signal processing operations on GPU

Inference

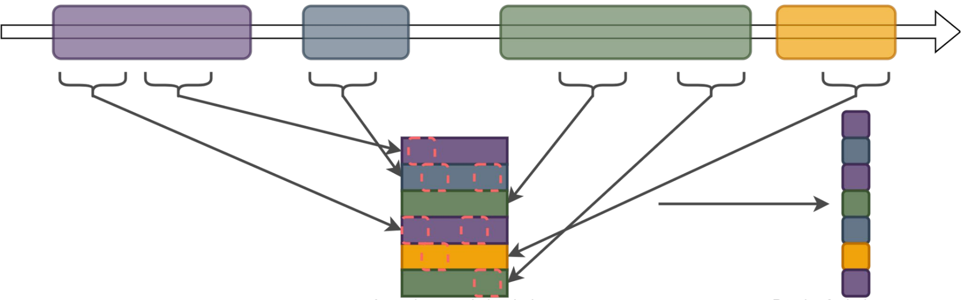
- Offline - produce predictions on O(months-years) of data
- Online - produce predictions on real-time data in O(ms)
- Stream timeseries data into NN
- Heterogeneous backends/dtypes

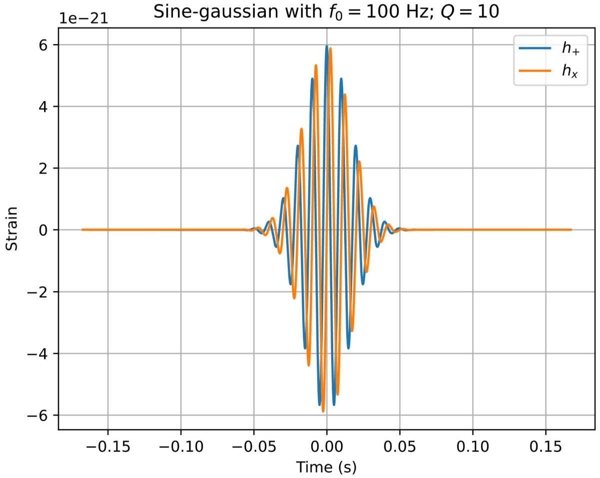
Design Goals

Intuitive - maps on to familiar, physically meaningful concepts
Composable - hierarchical layers of abstraction support new use cases seamlessly
Integrated - ecosystem of tools following same standards and nomenclature
Efficient - make the most out of parallel computing resources

`ml4gw` - Torch training utilities

Transitioning to larger datasets





Chunked loading of background data

Tradeoff between memory, I/O, and randomness

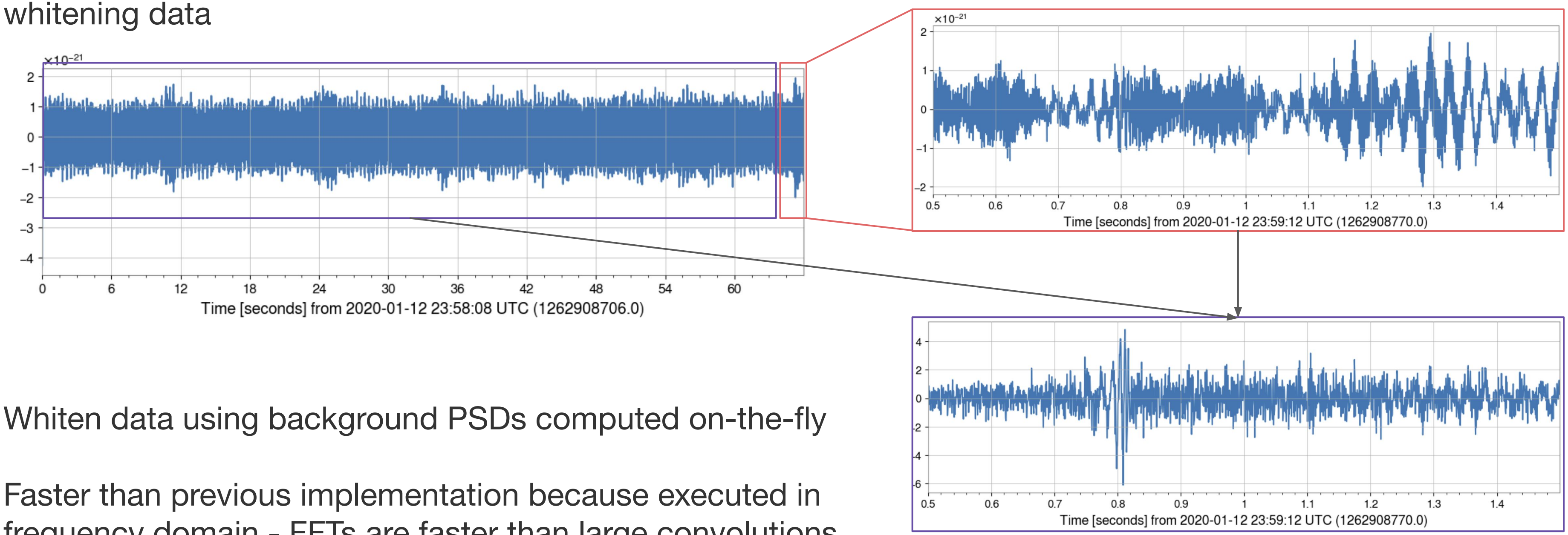
Fully on-the-fly generation of waveforms for unlimited training signal data

See impact in upcoming PE talk/poster

Only have sine gaussian implemented for now

`ml4gw` - Torch training utilities

More training background requires more flexibility when whitening data



Whiten data using background PSDs computed on-the-fly

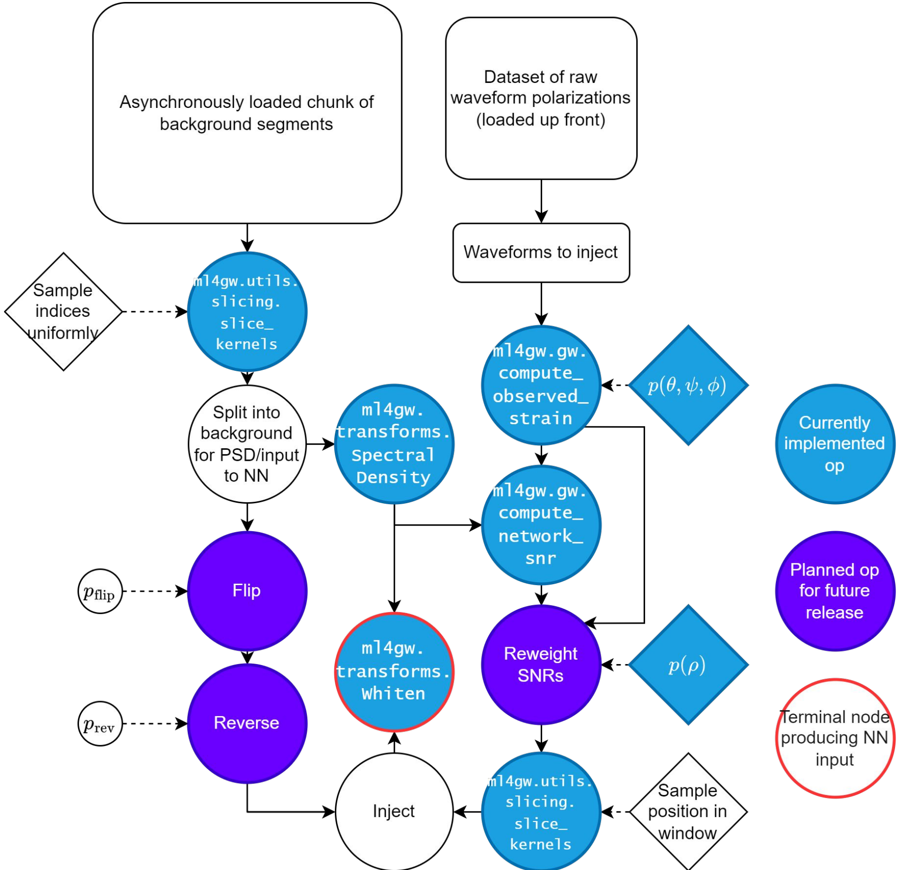
Faster than previous implementation because executed in frequency domain - FFTs are faster than large convolutions

`ml4gw` - Torch training utilities

Example use case: dataloading for binary black hole detection (`aframe`)

Complex data flow simplified by intuitive transform `Modules`

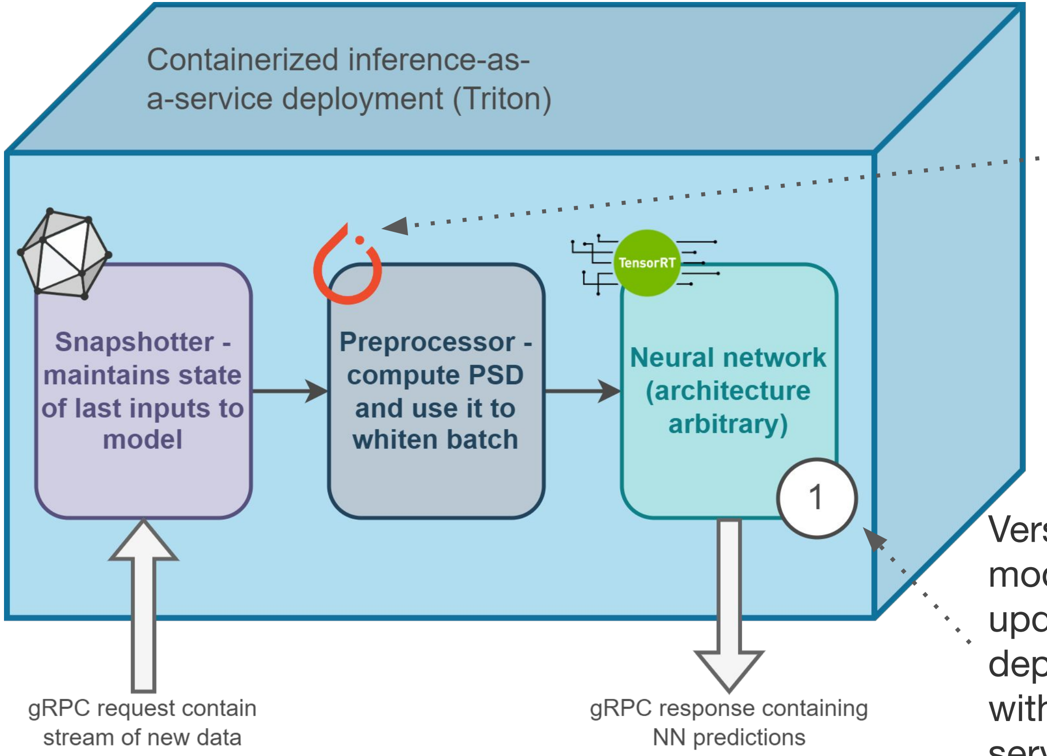
Efficient GPU implementations ensure strong utilization, shift bottleneck to NN forward/backward step



hermes - Inference-as-a-Service deployment tools

Example deployment: binary black hole detection (aframe)

hermes is a set of APIs for assisting in the acceleration, export, serving, and requesting of models using Triton Inference Server. New features include:

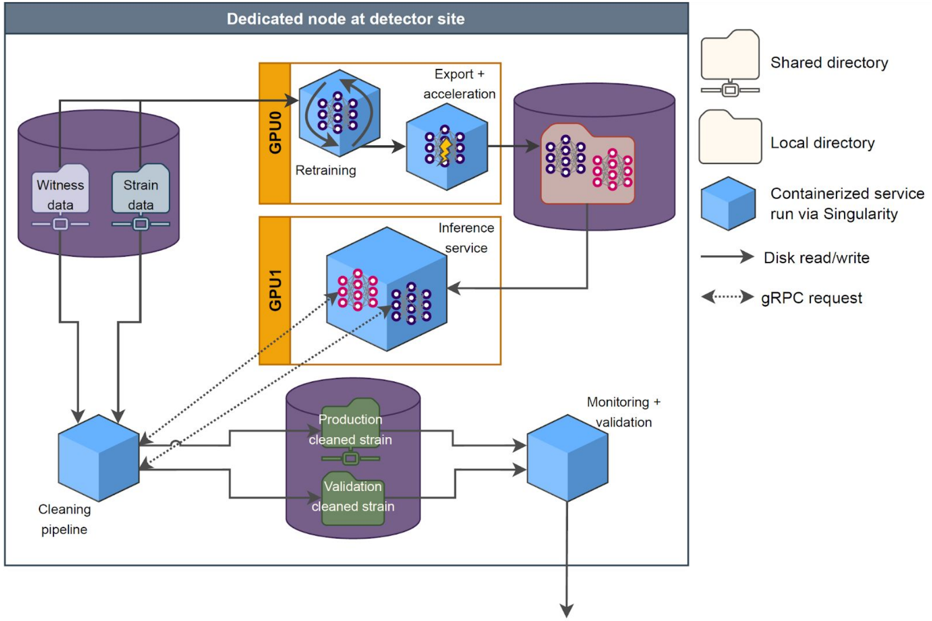


Support for Torchscript export format due to lack of support for FFT ops in ONNX

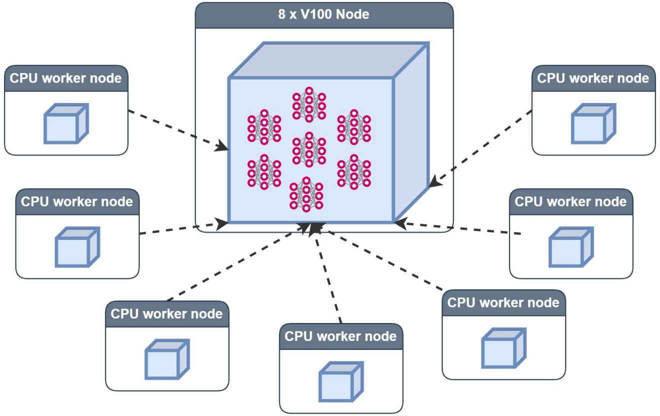
Version specification in model ensembles - update individual deployed models without interrupting service

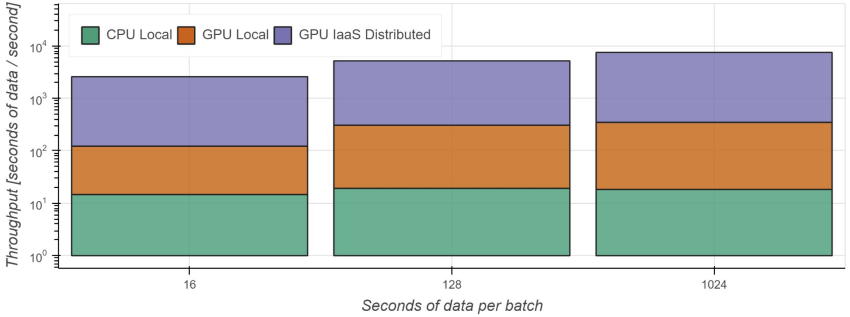
hermes - Inference-as-a-Service deployment tools

Example use case: online deployment of DeepClean noise subtraction algorithm



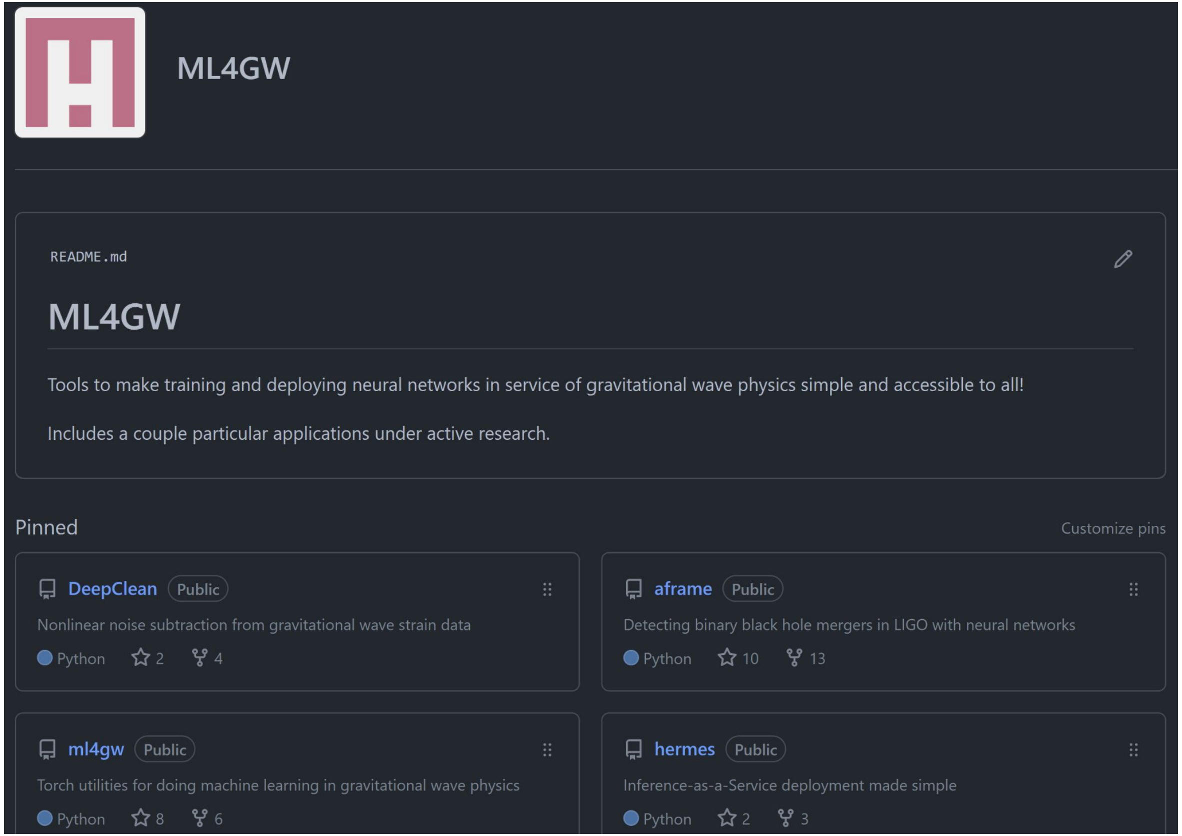
Example use case: offline deployment of `aframe`





Ensemble versioning allows newly trained models to be validated/deployed asynchronously

ML4GW



https://github.com/ML4GW

New releases coming this week

Lots more to be done - always looking for collaborators!

More use cases → more robust tools for everyone

Thank you!