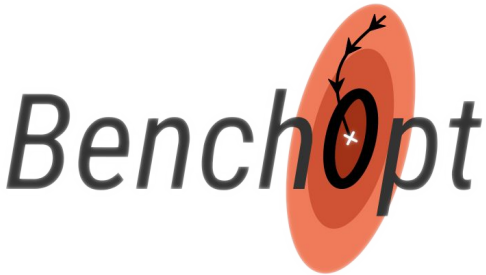
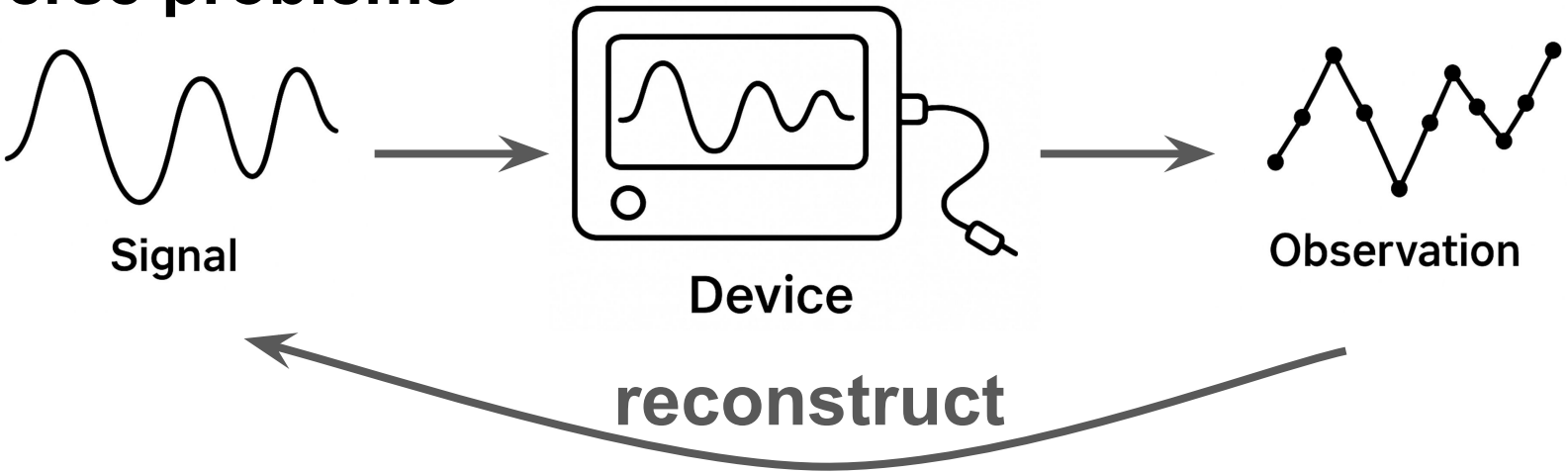
Benchmarking Large Scale Inverse Problems Resolution

Implementation & benchmarking



Inverse problems



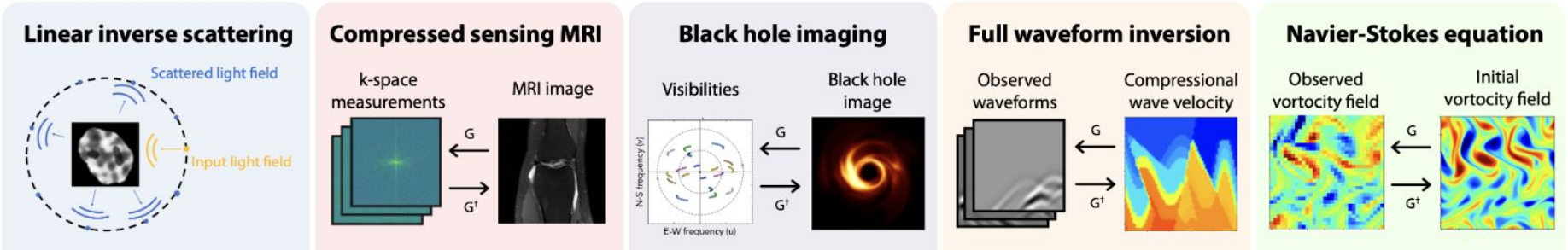
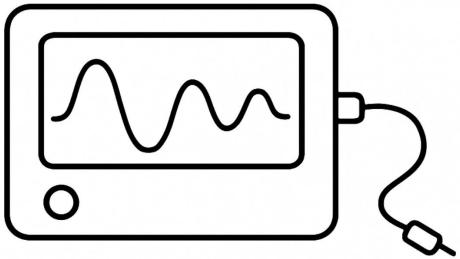


Figure adapted from Zheng et al., *InverseBench*, ICLR 2025.

Inverse problems

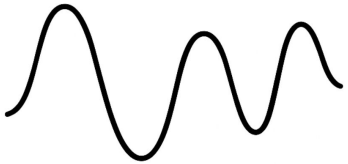
Challenges:



Device

The device loses information.

⟹ Ill-posed inversion problem

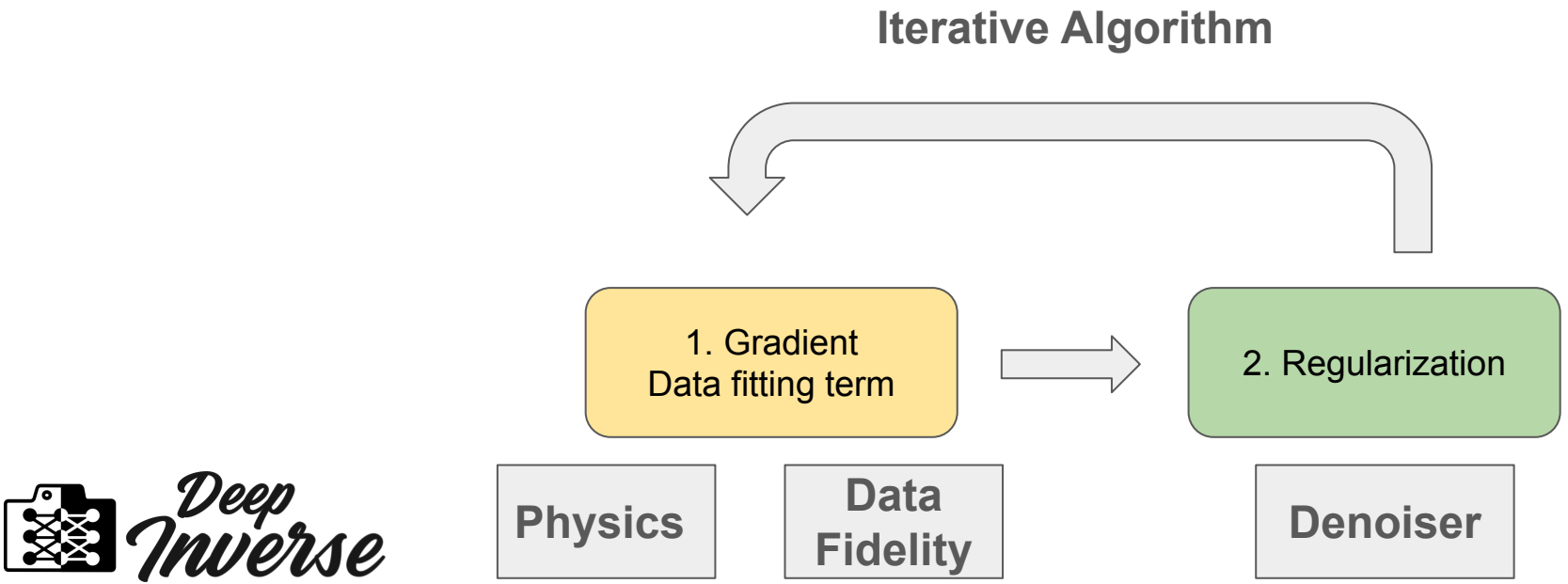


Signal

The reconstructed signal must be realistic.

⟹ Need for regularization

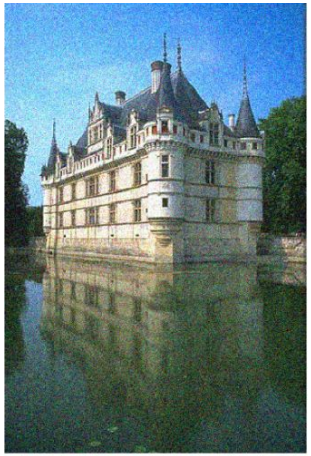
Solving Inverse Problems



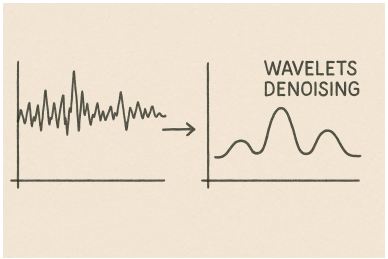
Deep Inverse

Regularization

Transform a low quality signal into
a close enough realistic version

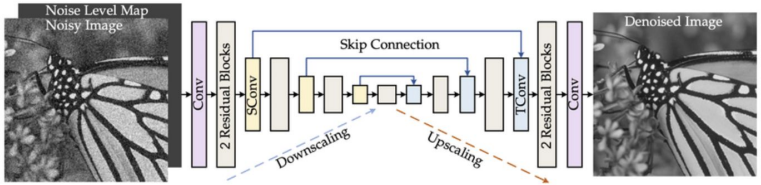


Hand-crafted priors



OR

Learned from signals



Possible challenges on large data

Costly gradient of data fidelity term because of the forward structure

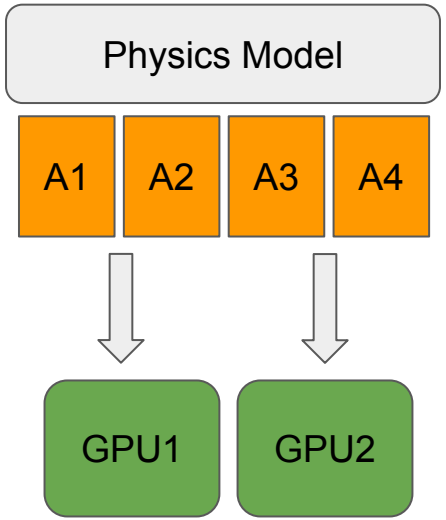
ex: numerous visibilities in radio interferometry + nufft

Regularization takes time or don’t even fit on 1 gpu

ex: deep network to denoise a 3D volume in tomography or a high resolution image

Distributing the forward / gradient

Assumption: $\mathcal{A} = \sum_{k=1}^{K} \mathcal{A}_k$

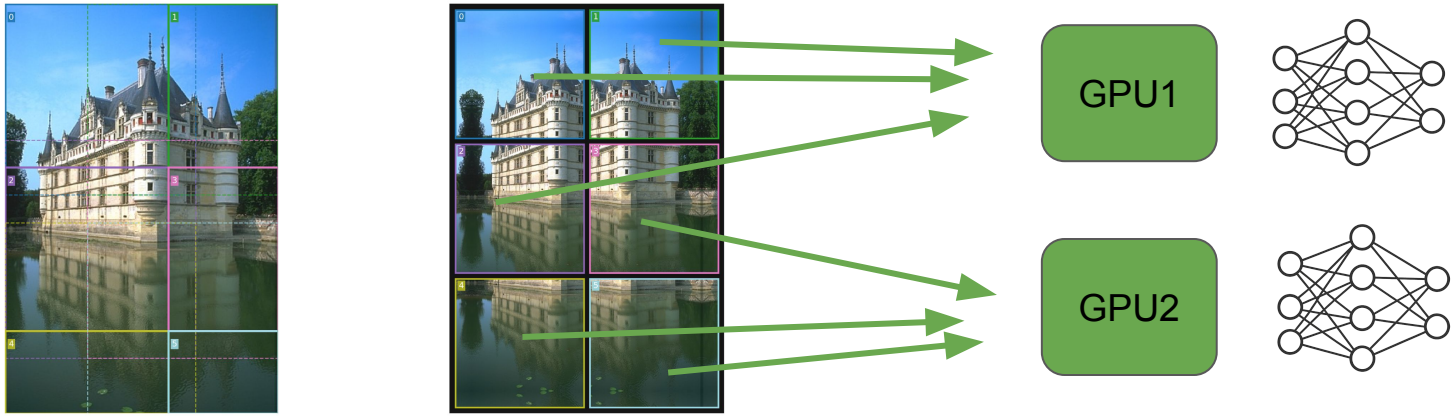


DistributedStackedPhysics - DistributedStackedLinearPhysics
Manages forward, adjoint and vector jacobian products in parallel

Distributed Data Fidelity
Manages gradient computation

Future work: distributing Physics spatially

Distributing the prior



DistributedProcessor - DistributedStrategy
Manage prior computation with respect to a given tiling and reduce strategy

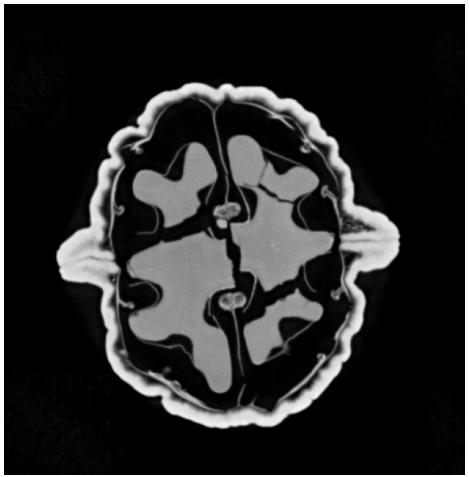
Current work: distributing backward pass

Use cases and benchmarking

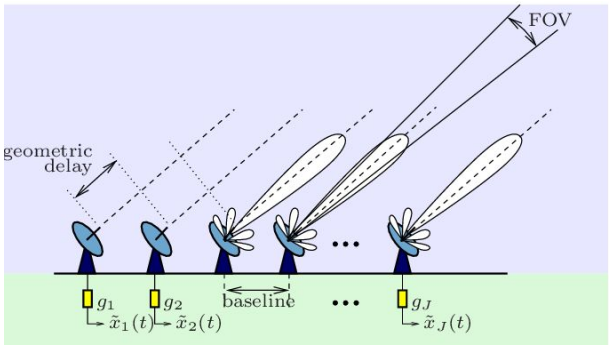
Large images and volumes (~10M-1G Pixels)

High resolution images

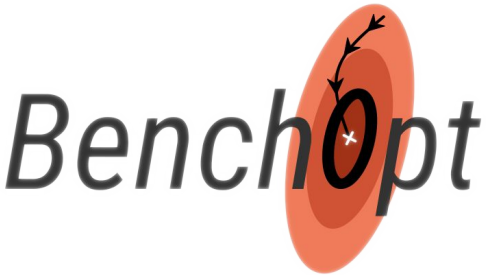
3D computed tomography



Radio-interferometry



Benchmarking

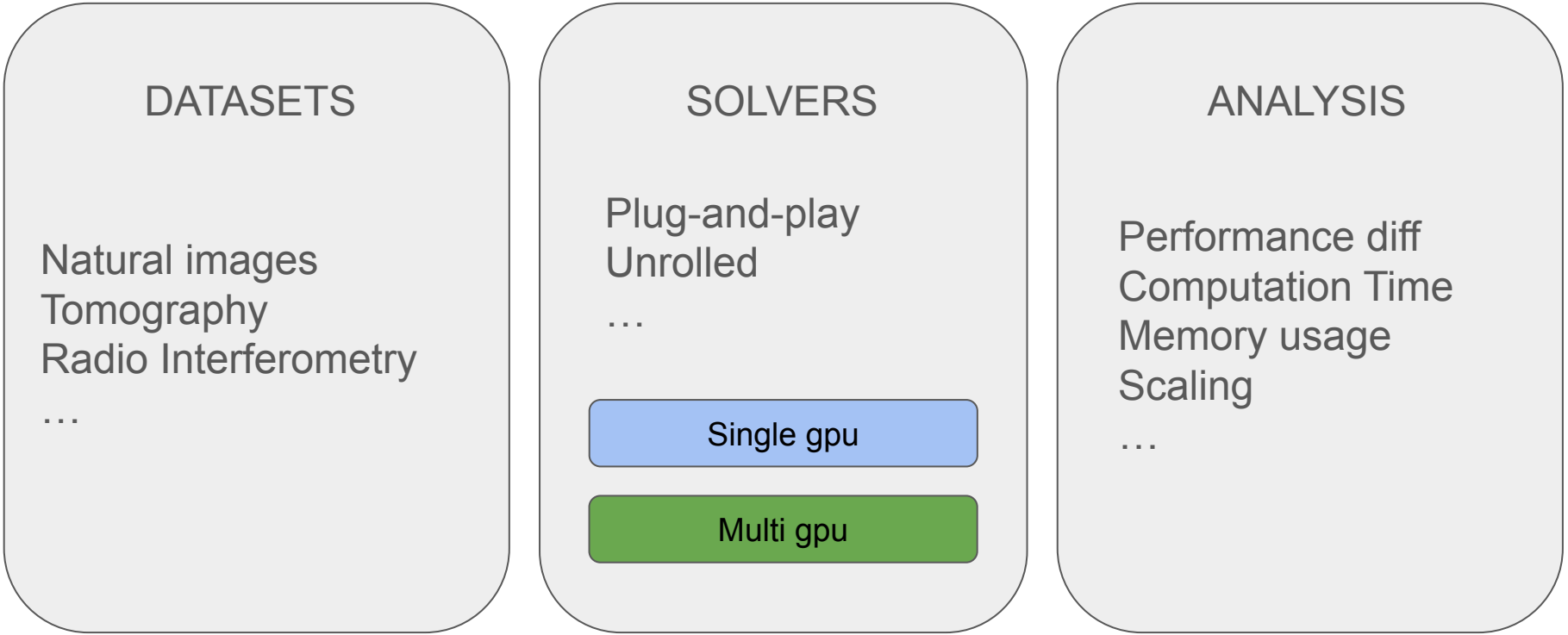


Goal: benchmark distributed inverse problems resolution (inference)

Metrics:

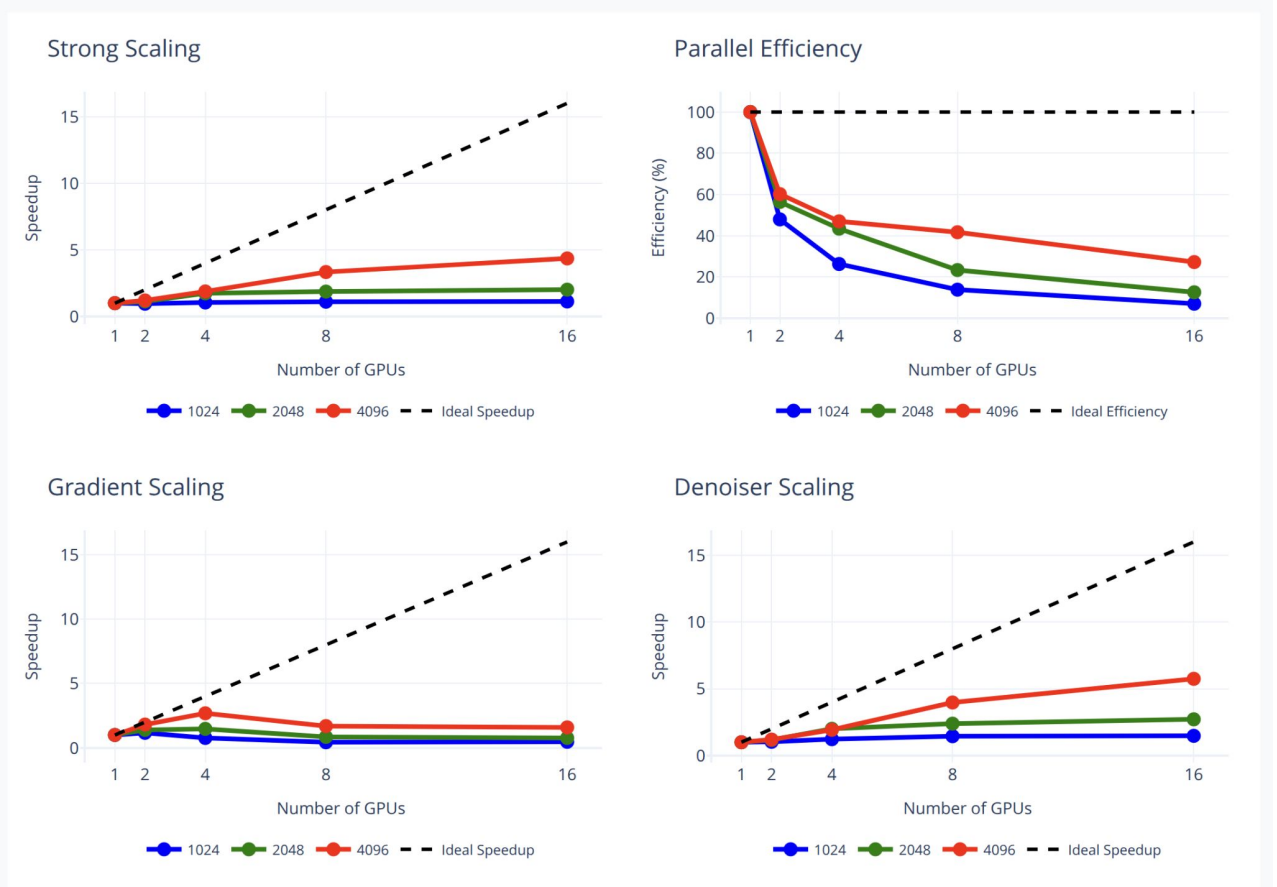
- reconstruction quality (PSNR, …)
- computation time, memory usage, communication overhead…

Benchmark structure



Examples of results

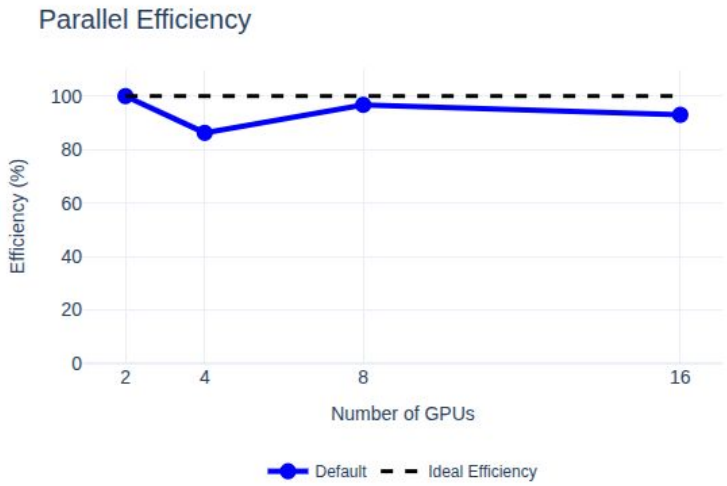
Natural 2D images



Doc available at:
https://bmalezieux.github.io/benchmark_invprob_inference/

Examples of results

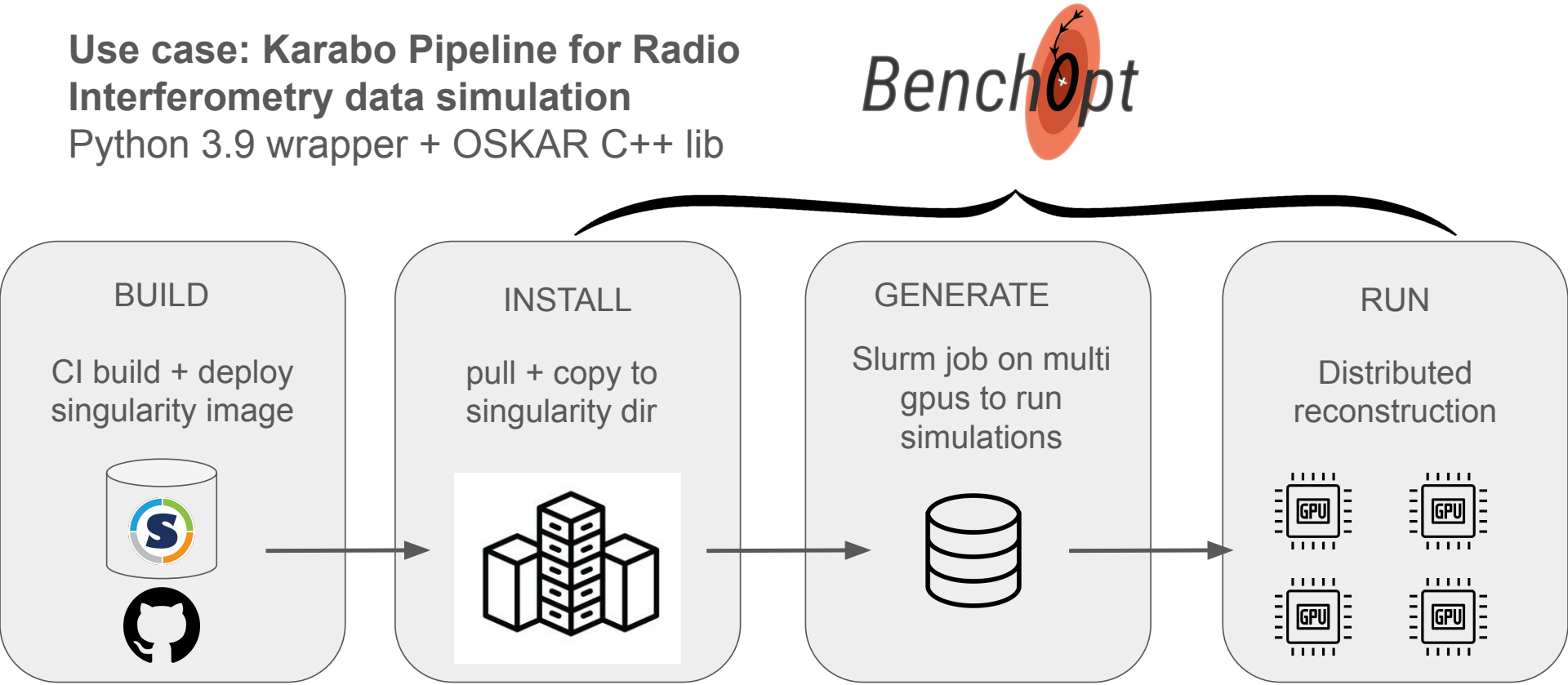
3D computed tomography (512^3)



Doc available at:
https://bmalezieux.github.io/benchmark_invprob_inference/

Bringing External Simulators into the Benchmark

Use case: Karabo Pipeline for Radio Interferometry data simulation
Python 3.9 wrapper + OSKAR C++ lib



Areas for Improvement - Next Steps

Performance monitoring → bottlenecks → optimizations

Standardize the end-to-end workflow (external simulators ↔ Torch distributed)

Reproducibility, easiness of use and extensibility

Start working on training (not just inference)