Task 3.3 — High-performance coupling of CFD solvers to deep neural networks

Status and next steps

Corentin Lapeyre, Anass Serhani, Gabriel Staffelbach (CERFACS)
Léo Nicoletti, Luca Marradi (ATOS-BULL)

CoE RAISE All-Hands Meeting

August 28-30, 2023 — Hveragerði, Iceland

Outline

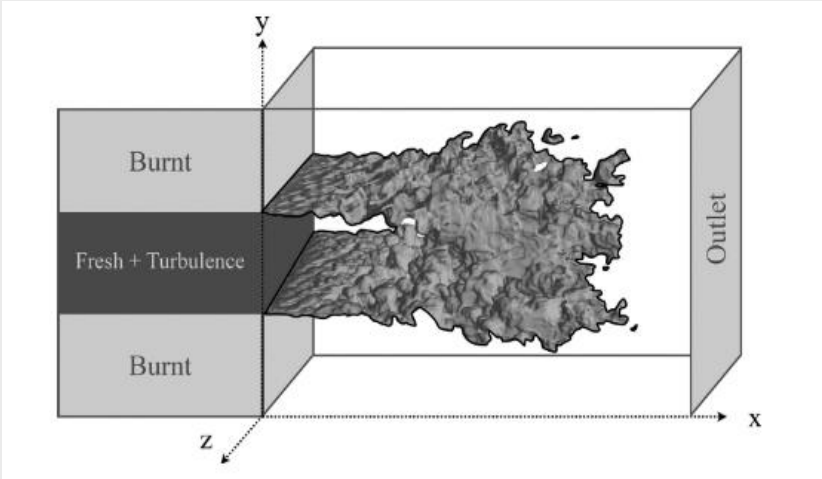
- Origin: data-driven models in reacting flows.
- PhyDLL: a Physics - Deep Learning coupler.
- Coupling GNNs with solvers.
- Next steps.

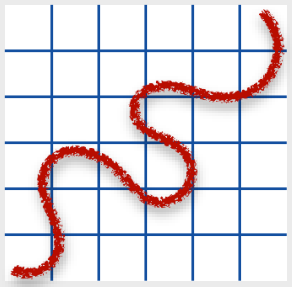
Where we come from:

Data-driven closure for turbulent combustion

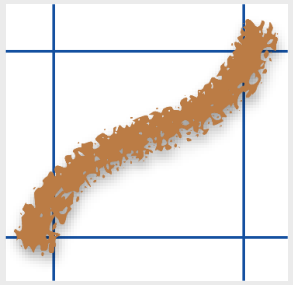
Data-driven models in reacting flows

Slot-jet burner Use-case





DNS : resolved flame front



LES : missing wrinkling

SGS slot-burner

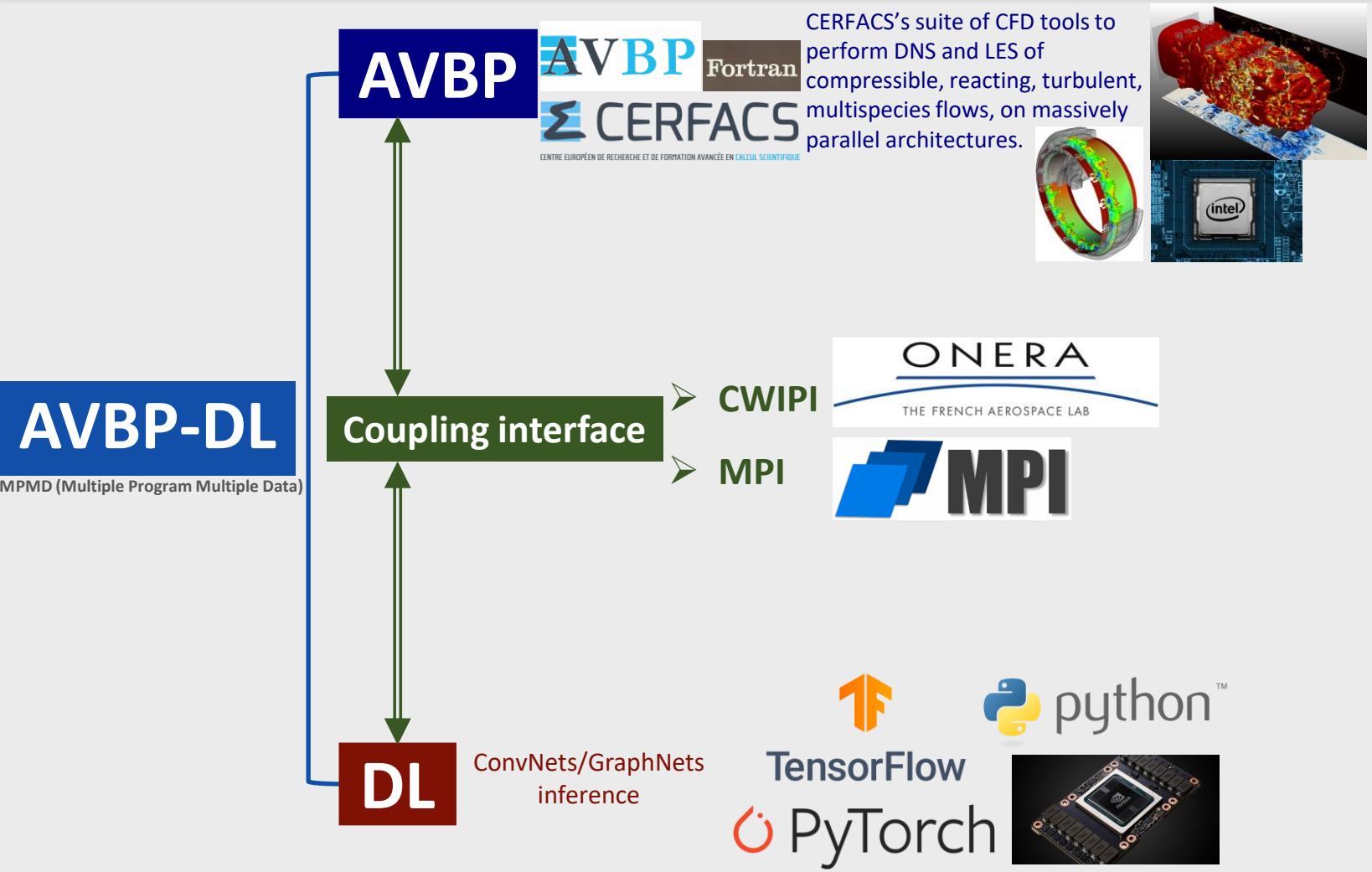
$(\dots, \bar{c}, \dots)$ Progress-variable → Classical computations ~~Charlette model~~ → $(\dots, \dot{\bar{\omega}}, \dots)$ Reaction-rate

Update $\dot{\bar{\omega}}$

$\bar{c}$ input → **Convolutional Neural Net (CNN)** → output Ξ effeciency

Note: This *must* be *online!*

AVPB-DL



Status:

- CNN/GNN inference.
- MPI/CWIPI coupling.
- Parallel inference for multi-GPUs setting.
- Hierarchical mesh partitioning.
- Good scaling results.
- Cheap DL overhead.

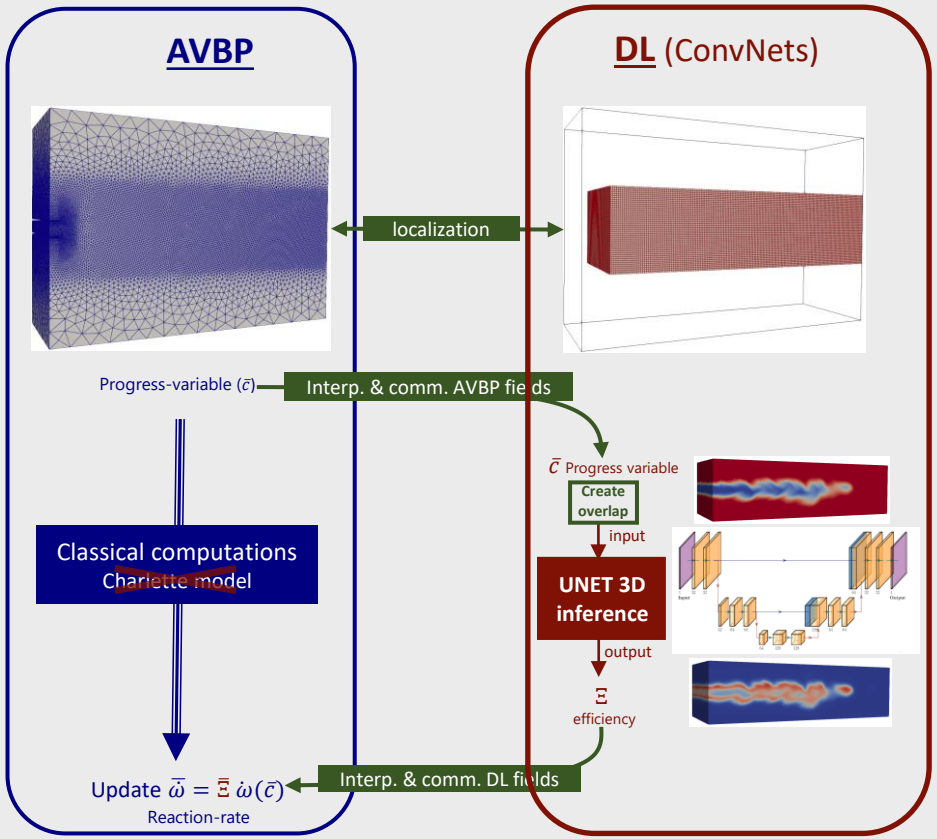
Ported & tested on:

- Local CERFACS supercomputer
- JEAN-ZAY (IDRIS)
- JUWELS-BOOSTER (Juelich)

Data-driven models in reacting flows

AVBP-DL (ConvNet): Slot-jet burner Use-case

❖ Slot-burner + ConvNet (CWIPI coupling)



❖ Slot-burner + ConvNet (CWIPI coupling)

mpirun –n 32 AVBP.EXE : -n 4 python dl.py

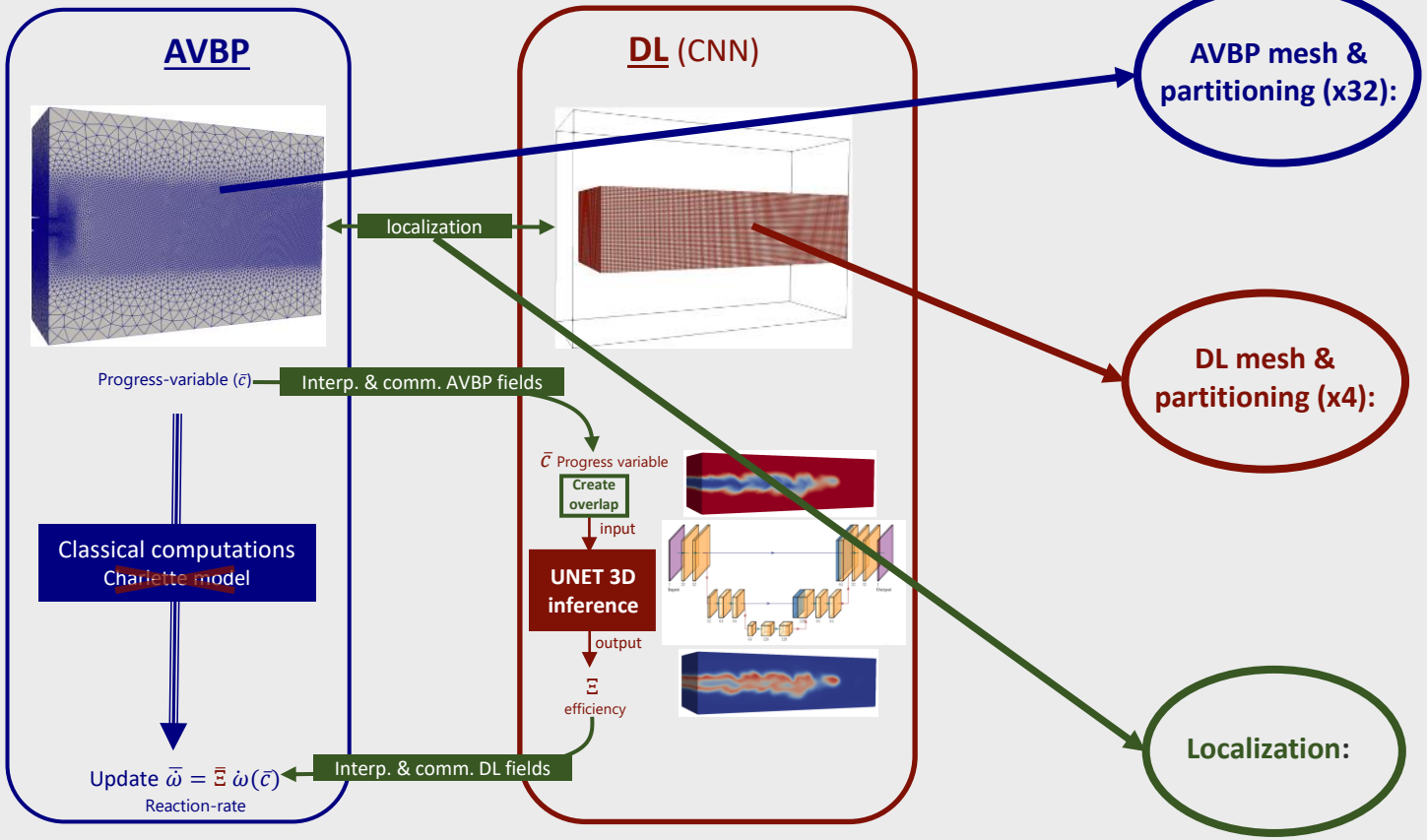
x32 Intel Skylake

x4 Nvidia V100

Data-driven models in reacting flows

AVBP-DL (ConvNet): Slot-jet burner Use-case

❖ **Slot-burner + ConvNet (CWIPI coupling)**



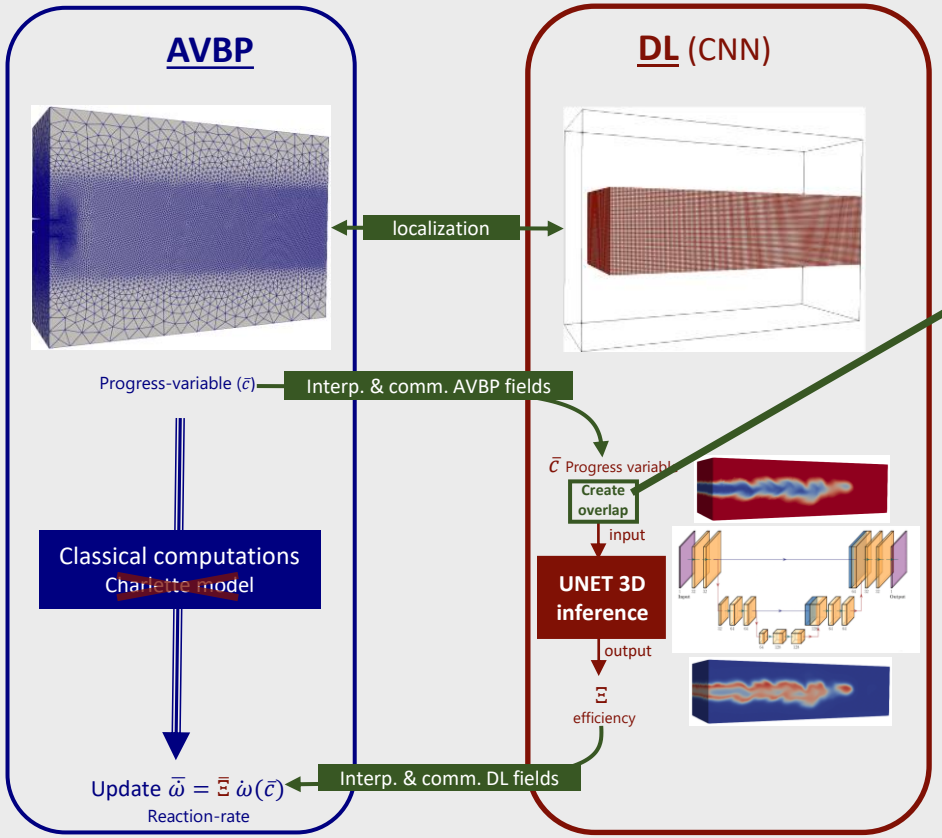
❖ **Meshes & localization results:**

Data-driven models in reacting flows

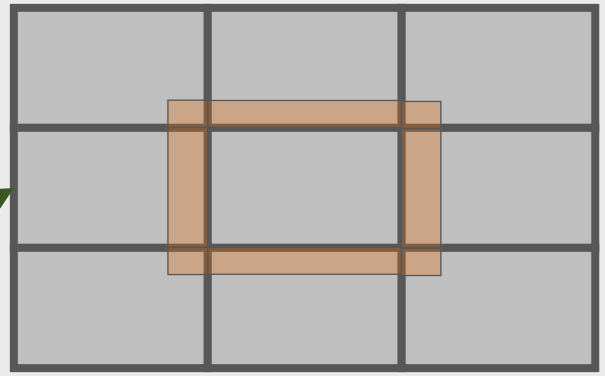
AVBP-DL (ConvNet): Slot-jet burner Use-case

RAISE Center of Excellence

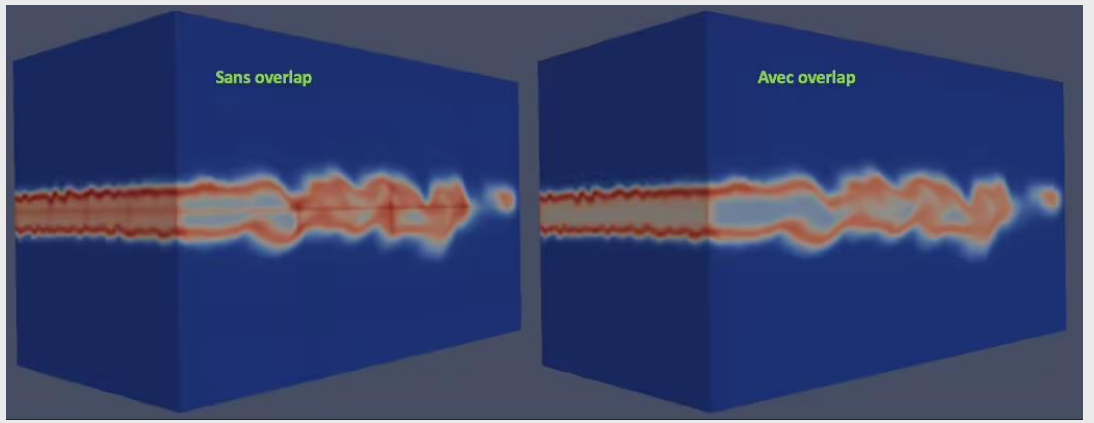
❖ **Slot-burner + ConvNet (CWIPI coupling)**



❖ **Overlap description**



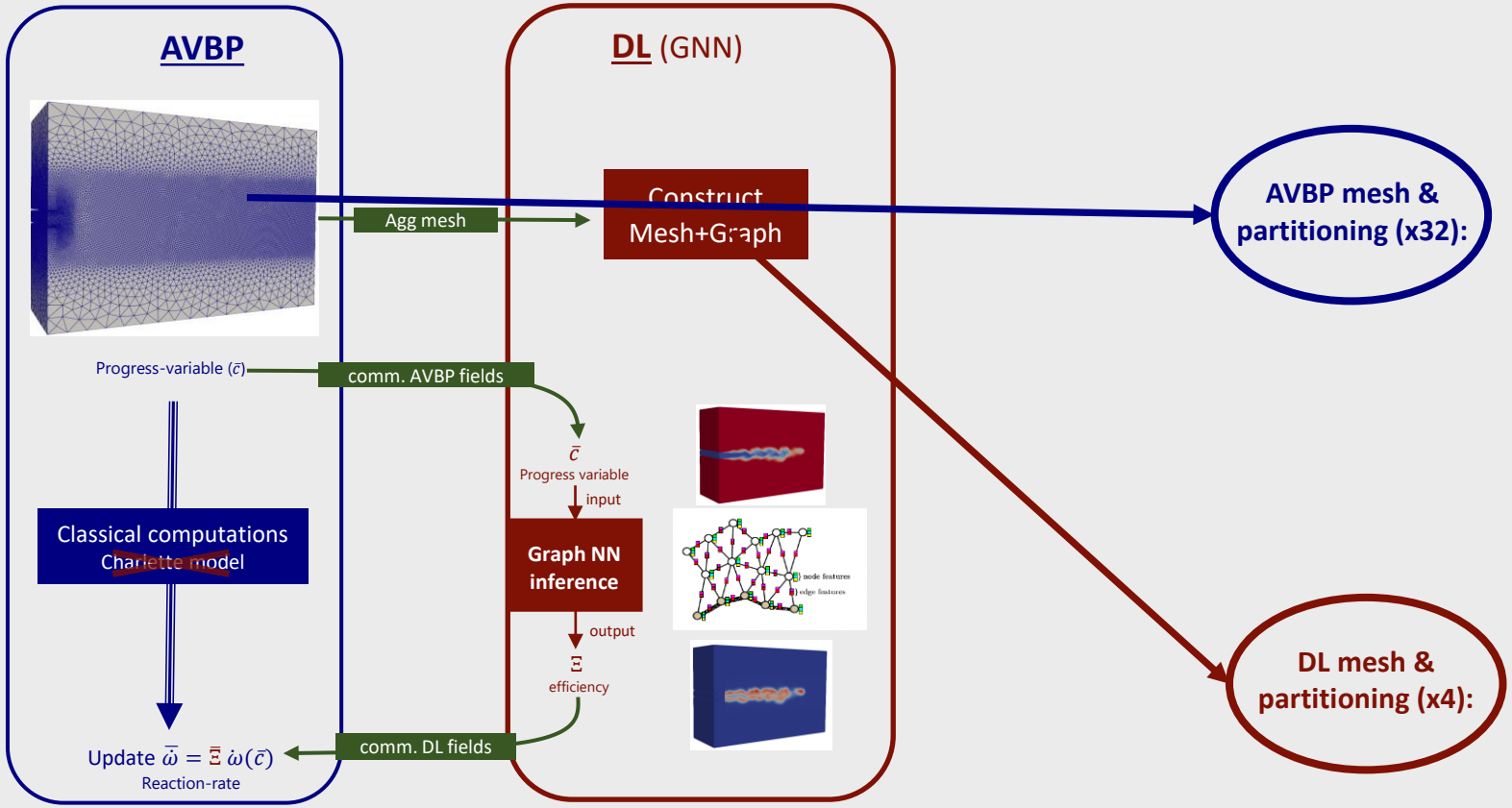
❖ **Overlap results**



Data-driven models in reacting flows

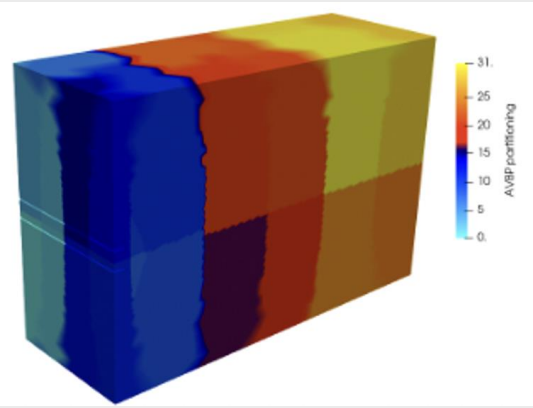
AVBP-DL (GraphNet): Slot-jet burner Use-case

❖ **Slot-burner + GraphNet (MPI coupling)**

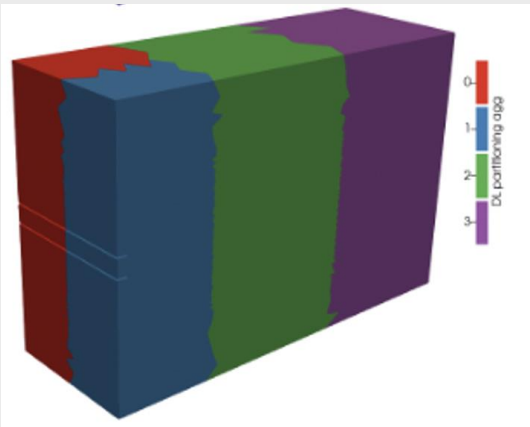


❖ **Meshes aggregation:**

AVBP mesh & partitioning (x32):



DL mesh & partitioning (x4):



PhyDLL:
Phy sics
Deep Learning
coupLer

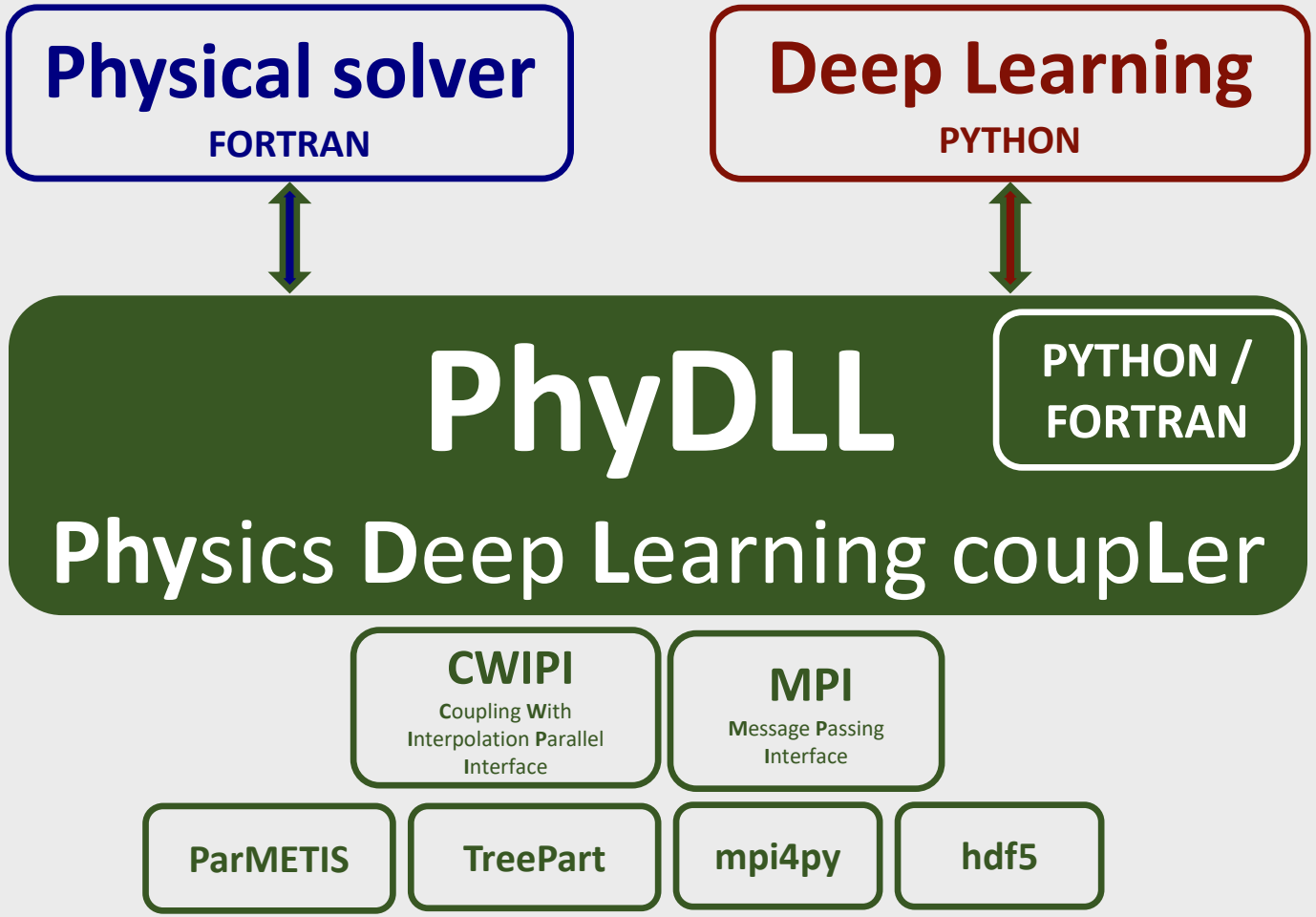
PhyDLL
Outline

PhyDLL (**Phy**sic **D**eep **L**earning Coup**L**er):

- Open-source high-performance coupling library.
- To be compiled with numerical simulation solvers.
- Written in Fortran and Python (future: C/C++).
- It has all AVBP-DL features:
  - CWIPI support.
  - MPI communications.
  - Distributed NN inference (multi-GPU).
  - CNN/GNN inference.

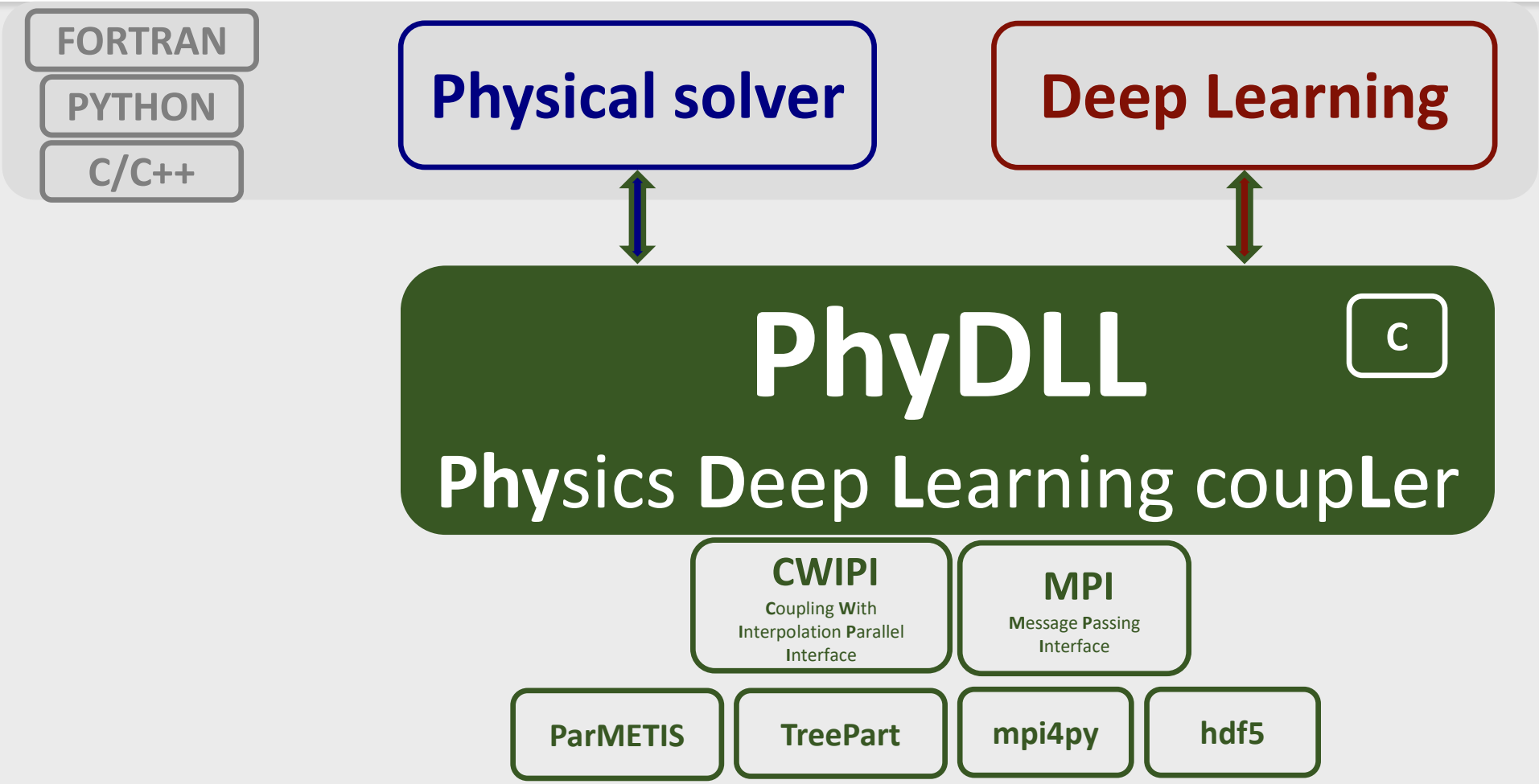
- ❑ Official website: *gitlab.com/cerfacs/phydll*
- ❑ Sources: *phydll.readthedocs.io*

PhyDLL 0.1
Status

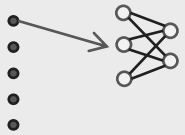
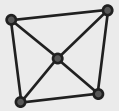
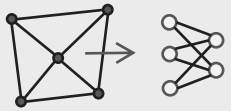
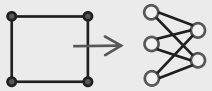
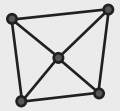
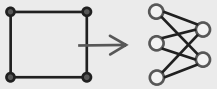


PhyDLL 0.2

Final tests ongoing



PhyDLL's thought process

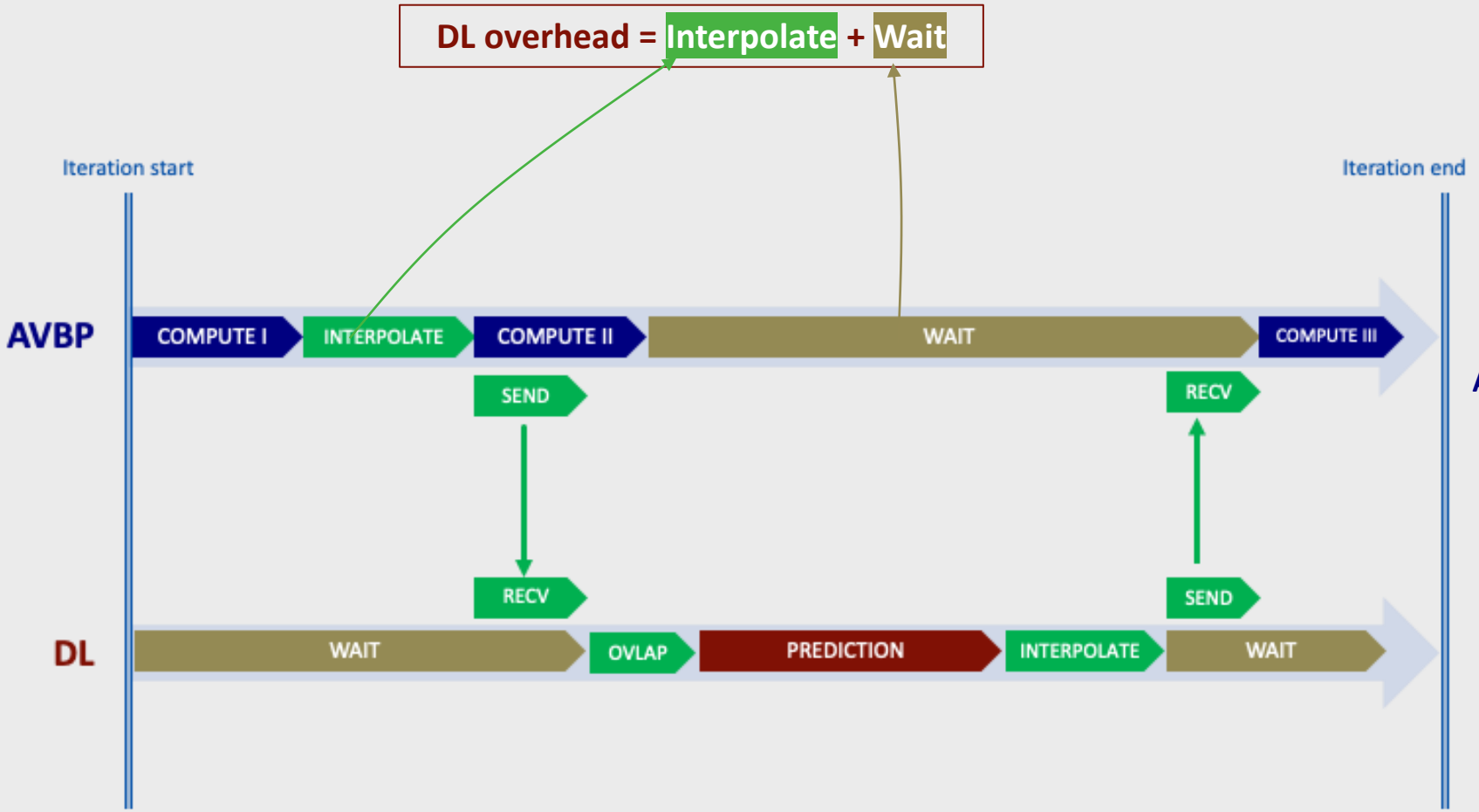
Physics Solver	Deep Learning		
		Algebraic formulation, no context needed	No-Context Scheme
		Spatial formulation, manually rebuilt on both sides	No-Context Scheme
		Spatial formulation, matching meshes	Direct Scheme
		Spatial formulation, **non**-matching meshes	Interpolation Scheme

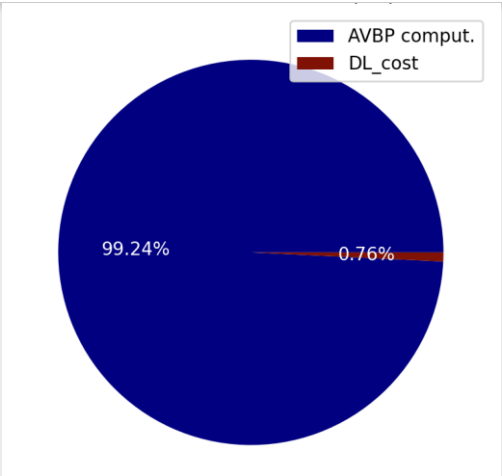
Data-driven models in reacting flows:

Performance result

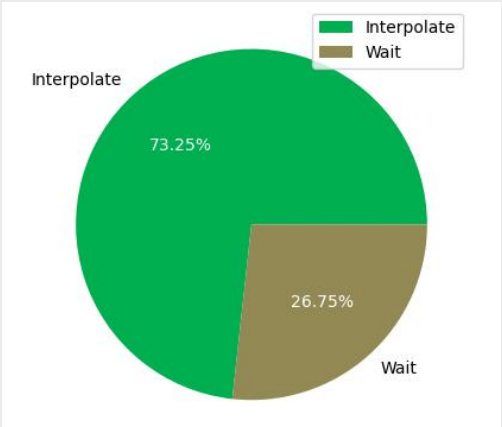
Performance result

Slot-jet burner Use-case: DL overhead





AVBP-DL iteration time:
AVBP computations and deep learning overhead

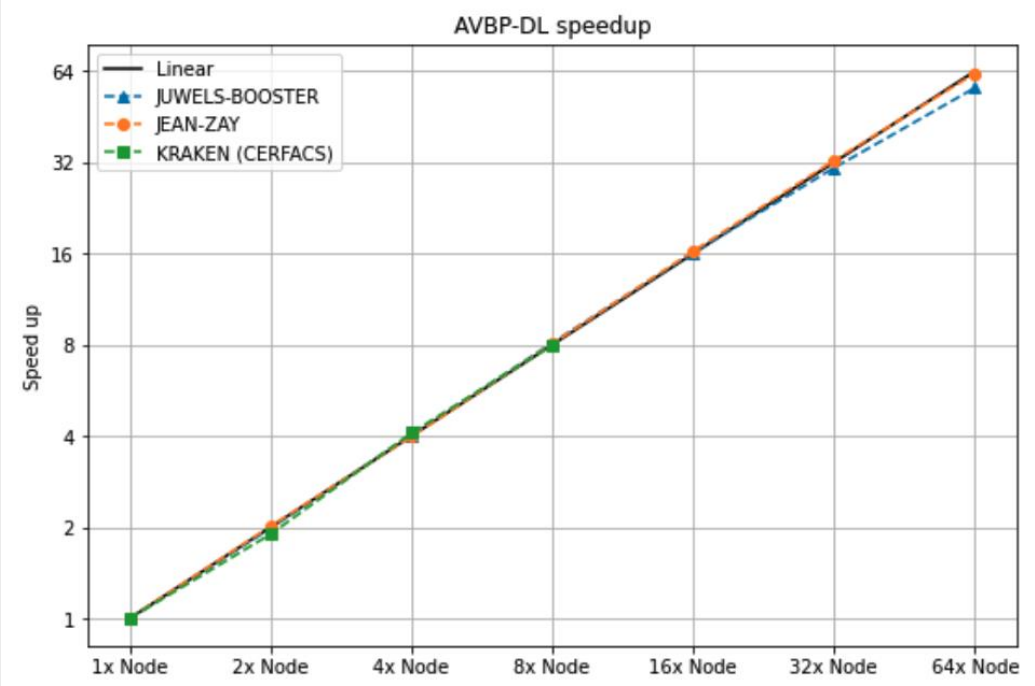


Zoom on DL cost: Interp + Wait

Performance result

Slot-jet burner Use-case: Scaling results

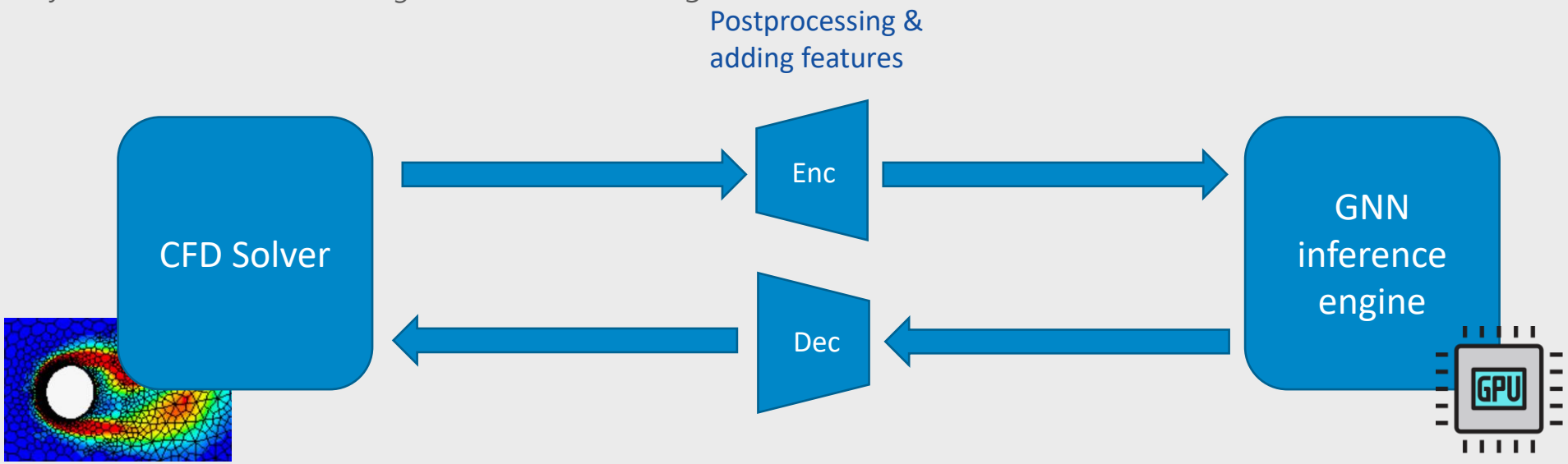
- **KRAKEN** : 1x Node := **AVBP:** 28x INTEL Skylake + **DL:** 4x NVIDIA A30 24GB
- **JEAN-ZAY** : 1x Node := **AVBP:** 36x INTEL Cascade Lake + **DL:** 4x NVIDIA V100 16GB
- **JUWELS-BOOSTER** : 1x Node := **AVBP:** 44x AMD Rome Epyc + **DL:** 4x NVIDIA A100 40GB



Coupling GNNs with solvers

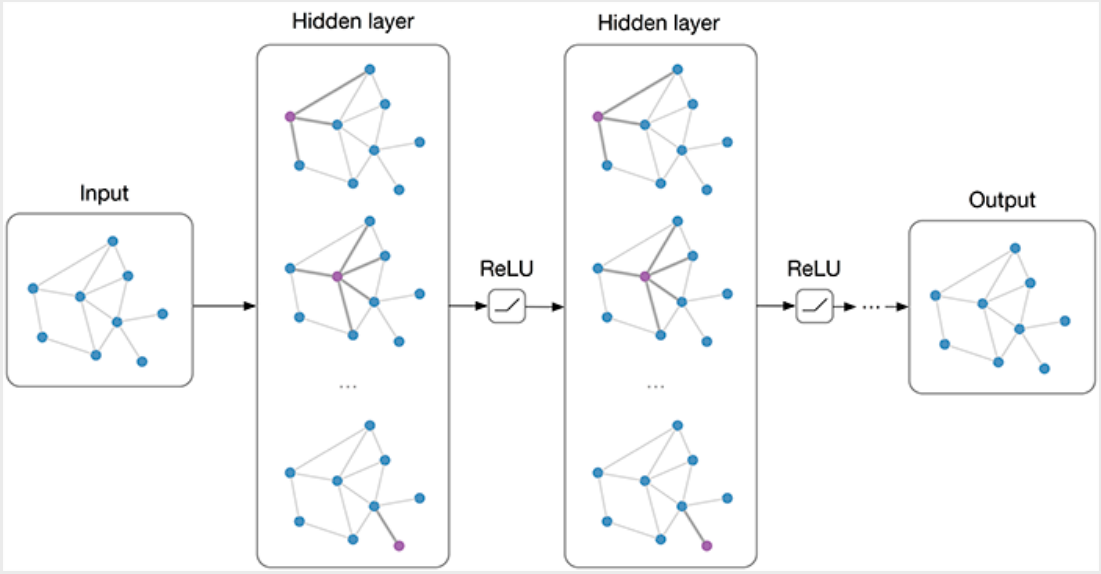
Coupling GNNs with solvers
Mesh to graph relation

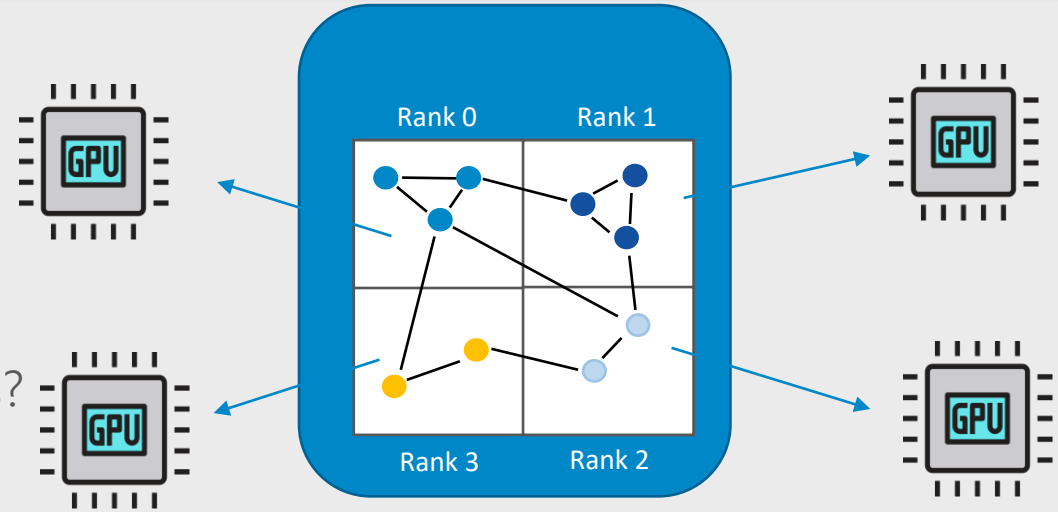
- CFD solvers rely on irregular meshes
  - Motivates the use of GNNs as data-supervized surrogates
  - Avoids interpolation by directly feeding the mesh as input
- Lightweight encoding/decoding transforms on the same graph structure
  - Potentially add tailored features, *e.g.* distance on the edges

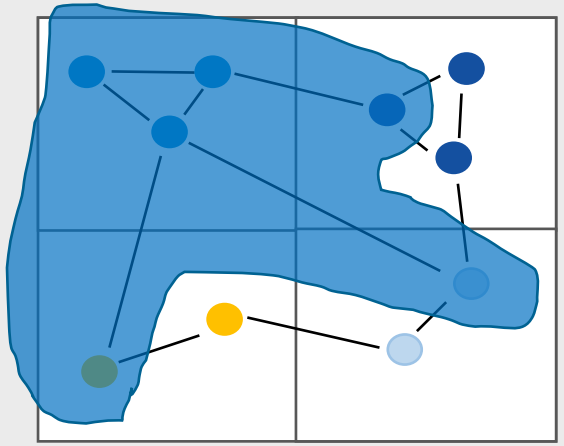


Coupling GNNs with solvers
Spatial parallelization

- Spatial parallelization is needed for inference:
  - over large meshes
  - with deep spatial correlations
- But how to handle the growing *receptive field* of the GNN filters?



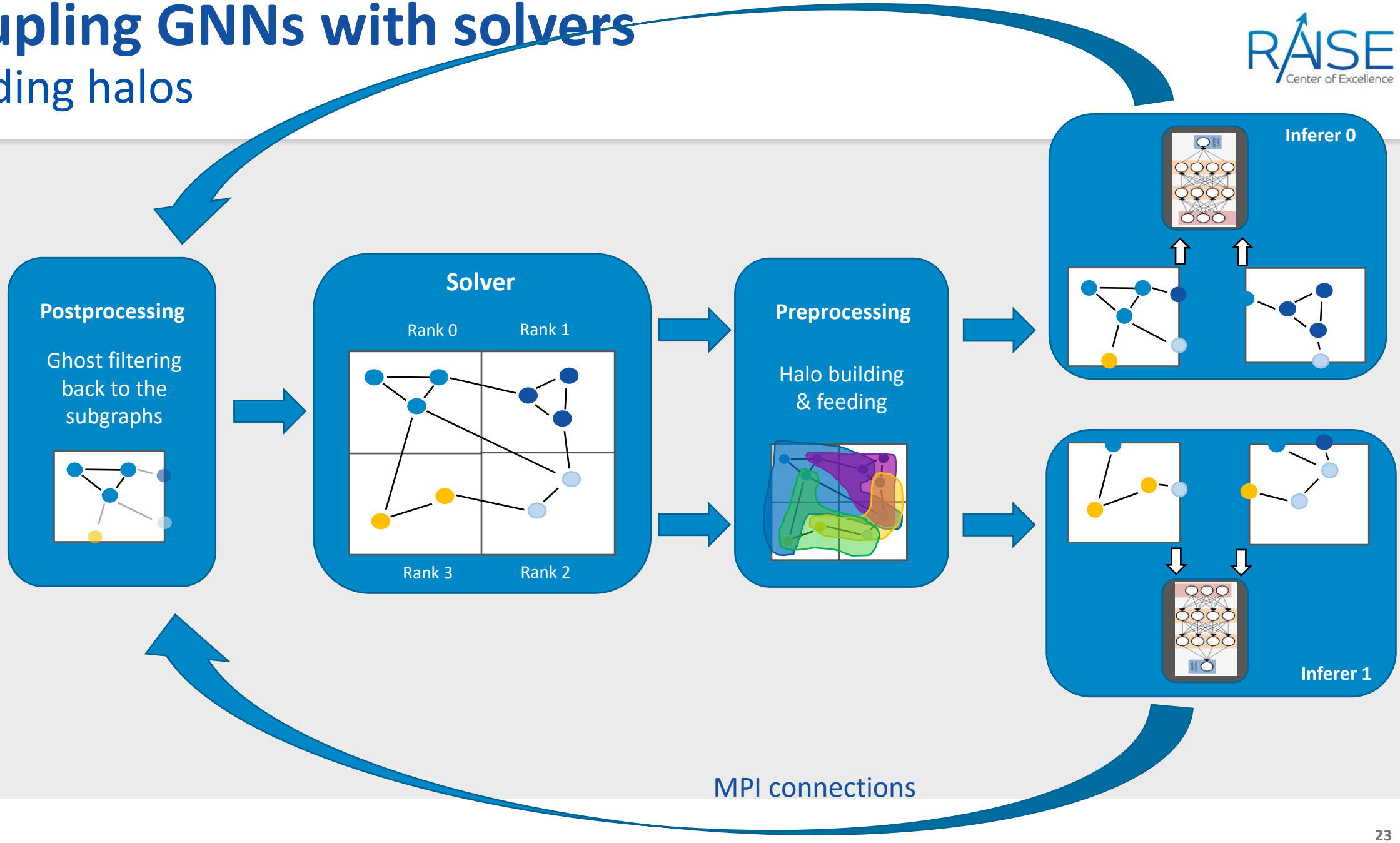




First-order receptive field of the rank 0 subgraph

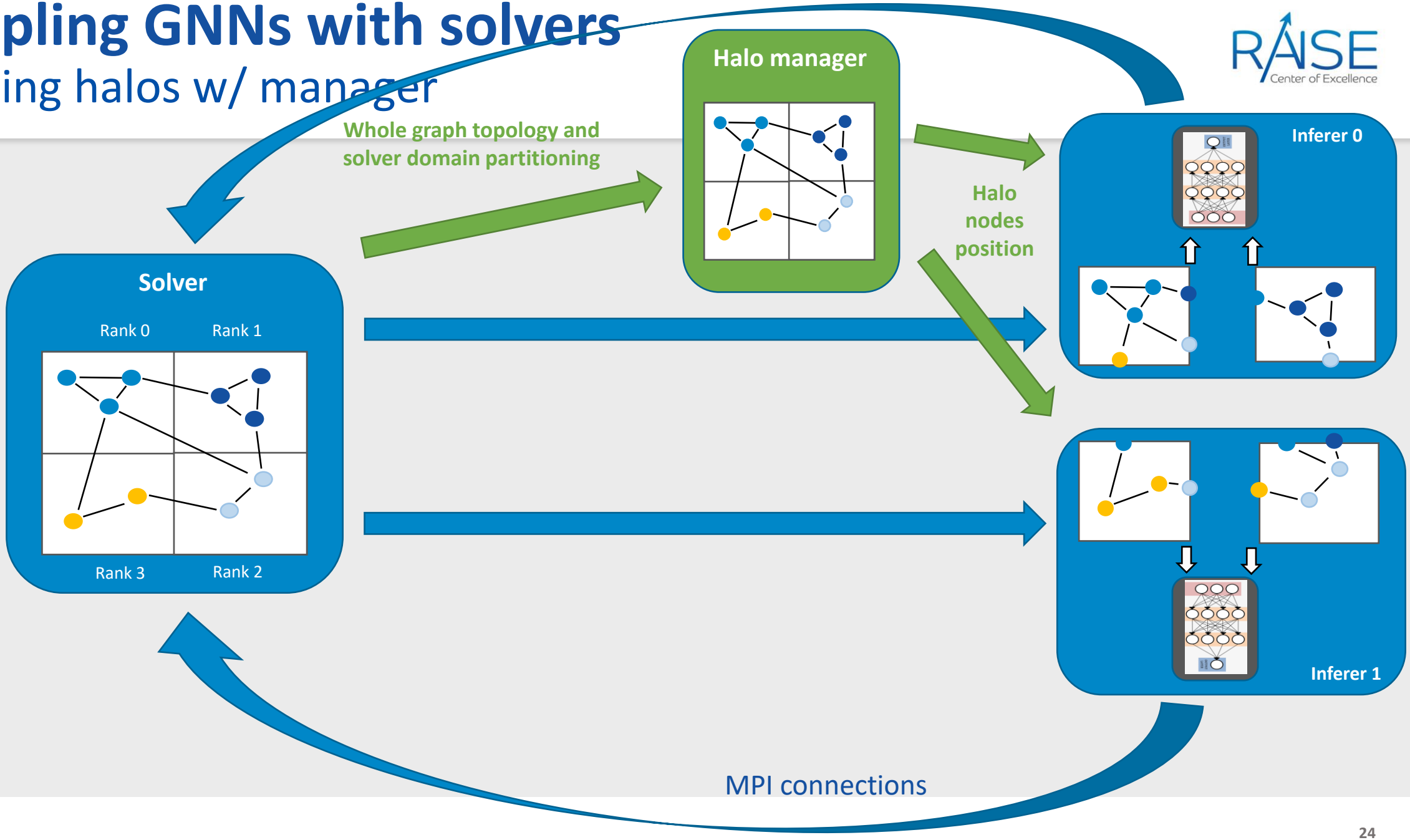
Coupling GNNs with solvers

Building halos



Coupling GNNs with solvers

Building halos w/ manager



Next steps

Next steps

- Continue the development of PhyDLL 0.2 and release
- Consolidate interfaces for Python, Fortran, C/C++
- Offer continued support to all RAISE partners wishing to work with PhyDLL

drive. enable. innovate.

RAISE Center of Excellence

The CoE RAISE project has received funding from
the European Union's Horizon 2020 –
Research and Innovation Framework Programme
H2020-INFRAEDI-2019-1 under grant agreement no. 951733

Follow us: