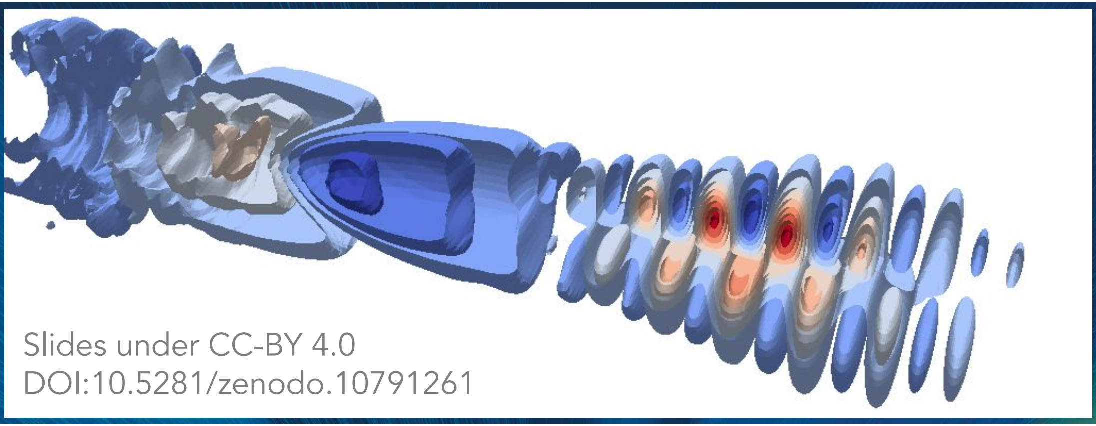
# AI/ML Coupling & Surrogates in BLAST Accelerator Modeling Codes

Axel Huebl, Ryan T Sandberg, Remi Lehe, Chad E Mitchell,
Marco Garten, Ji Qiang, and Jean-Luc Vay
*Lawrence Berkeley National Laboratory*




DOI:10.5281/zenodo.10791261

RT Sandberg et al.,
*accepted to PASC24,*
arXiv:2402.17248 (2024)

4th ICFA Beam Dynamics Mini-Workshop on Machine Learning Applications for Particle Accelerators – Gyeongju, March 5-8, 2024

BERKELEY LAB LDRD
SciDAC Scientific Discovery through Advanced Computing
BERKELEY LAB
ACCELERATOR TECHNOLOGY & APPLIED PHYSICS DIVISION ATAP
U.S. DEPARTMENT OF ENERGY | Office of Science

# Abstract (20' incl. Q&A)

Detailed modeling of particle accelerators can benefit from parallelization on modern compute hardware such as GPUs and can often be distributed to large supercomputers. Providing production-quality implementations, the Beam, Plasma & Accelerator Simulation Toolkit (BLAST) provides multiple modern codes to cover the widely different time and length scales between conventional accelerator elements and advanced, plasma-based elements. The Exascale code WarpX provides electromagnetic and -static, t-based particle-in-cell routines, advanced algorithms and is highly scalable. For beam-dynamics, the s-based ImpactX code provides an efficient implementation for tracking relative to a nominal reference trajectory, including space charge. Integrated modeling of "hybrid" beamlines – integrating both detailed plasma models and large-scale transport at full detail – requires exchange between codes and is limited by the computational speed of the most-detailed element, usually the plasma element.

In this work, we present an alternative approach to coupling particle-in-cell models and codes beyond direct data exchange or reduced details for accelerator modeling. In particular, we investigate and demonstrate detailed data-driven modeling based on high-quality WarpX simulations that were used to train surrogate models for the beam transport code ImpactX. We describe new workflows, illuminate predictive quality, performance and applicability to central research topics in advanced accelerator research, such as staging of laser-wakefield accelerators.

# Outline

## BLAST Codes for Exascale

Our Background: WarpX and ImpactX

## GPU-accelerated ML surrogates

Approach: establishing rapid, fully accelerated, "in-the-loop" ML

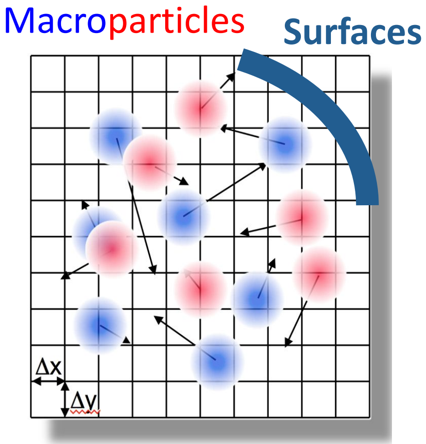
## Staging of LWFA for future HEP colliders

Demo: Hybrid beamlines - plasma-transport modeling

# BLAST Codes for Exascale

## WarpX and ImpactX

# First Principle Particle-in-Cell Modeling of Particle Accelerators




electromagnetic (EM) fields on a grid

Involves the modeling of the intricate interactions of

- **relativistic particles**: beams, plasmas, halo, stray electrons
- **EM fields**: accelerating/focusing fields, beam self-fields, laser/plasma fields
- **structures**: metals, dielectrics.

Typical computer representations:

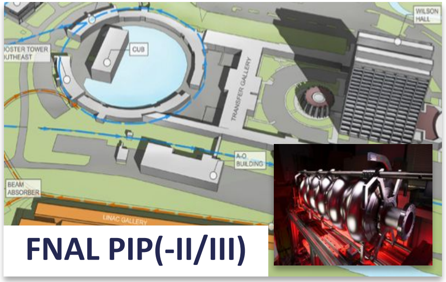
- **particles:** macro particles representing each 1-$10^6$ particles
- **fields:** electromagnetic, on a grid
- **structures:** surfaces interacting with grid and macroparticles

Many space- and time scales to cover:

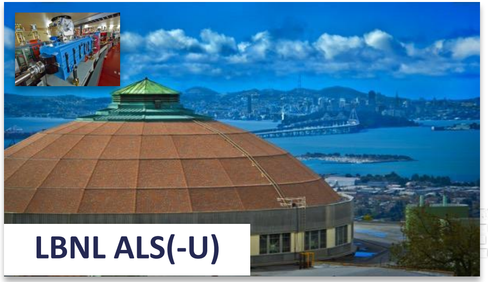
- from **µm** (e.g., plasma structures, $e^-$-surface interactions) **to km** (e.g., LHC)
- from **ns** (beam passing one element) **to seconds or more** (beam lifetime)

**⇒ needs best algorithms on largest & fastest computers**

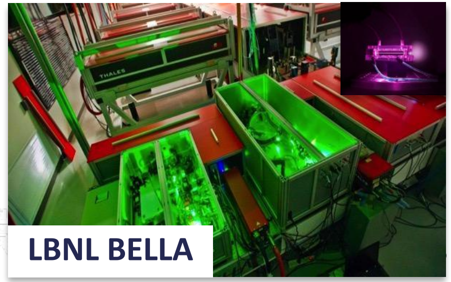
CERN (HL-)LHC



FNAL PIP(-II/III)

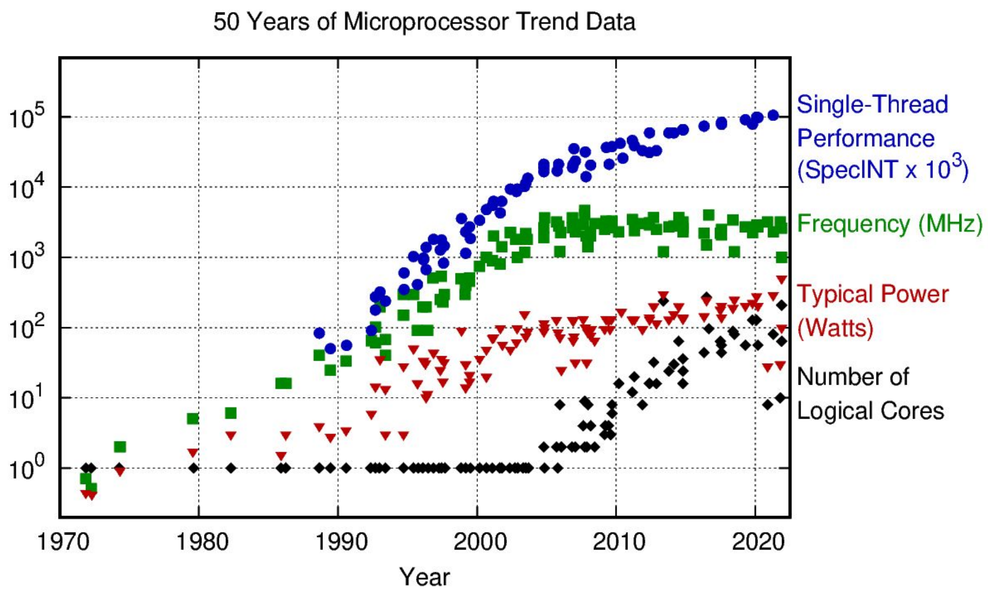


LBNL ALS(-U)



LBNL BELLA

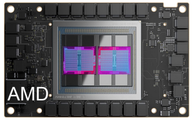
# Power-Limits Seeded a Cambrian Explosion of Compute Architectures

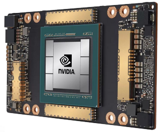



Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2021 by K. Rupp




ARM



CPUs

GPUs

## Top 500


Frontier (USA): 1.2 EFlops
- AMD GPUs

Fugaku (Japan): 0.44 EFlops
- Fujitsu ARM CPUs


Lumi (Finland): 0.3 EFlops
- AMD GPUs


Leonardo (Italy): 0.24 EFlops
- Nvidia GPUs

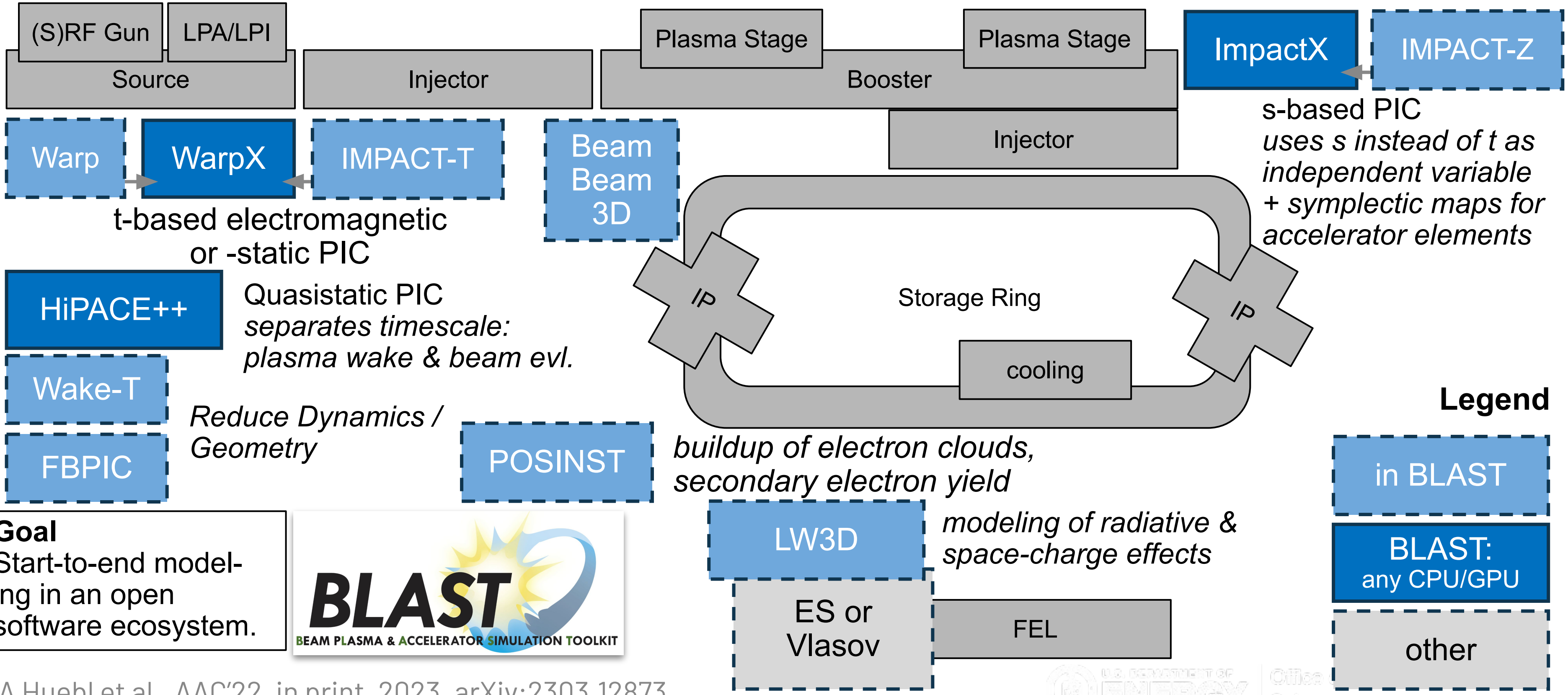

Summit (USA): 0.15 EFlops
- Nvidia GPUs

Upcoming (under acceptance testing)


Aurora (USA): ~2 EFlops
- Intel GPUs

# Beam, Plasma and Accelerator Simulation Toolkit (BLAST) at Exascale

Imagine a future, ***hybrid* particle accelerator**, e.g., with conventional and plasma elements.



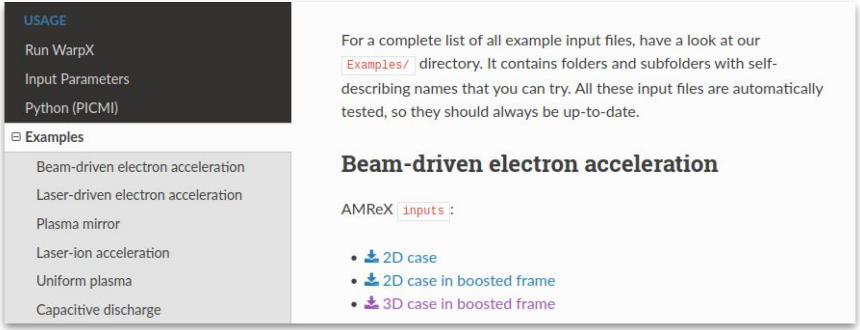

**Goal**
Start-to-end model-ing in an open software ecosystem.

A Huebl et al., AAC'22, in print, 2023. arXiv:2303.12873

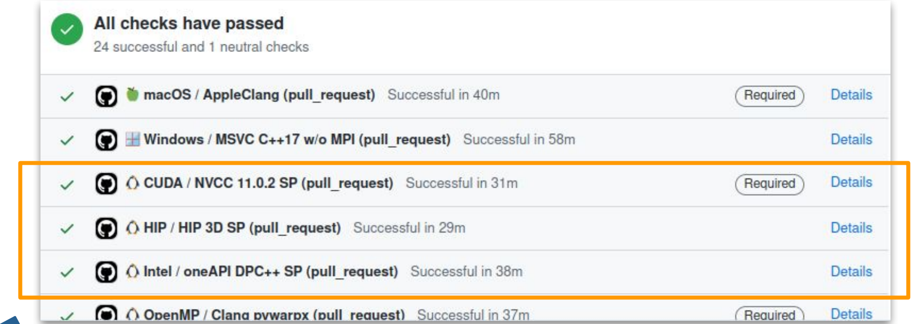
# We Develop Openly with the Community

BLAST
BEAM PLASMA & ACCELERATOR SIMULATION TOOLKIT

## Online Documentation:
warpx|hipace|impactx.readthedocs.io




## Open-Source Development & Benchmarks:
github.com/ECP-WarpX




**230 physics benchmarks** *run on every code change* of WarpX
**34 physics benchmarks** for ImpactX

## Rapid and easy installation on any platform:

**conda install -c conda-forge warpx**

**spack install warpx**
**spack install py-warpx**

**cmake -S . -B build**
**cmake --build build --target install**

**python3 -m pip install .**

**brew tap ecp-warpx/warpx**
**brew install warpx**


**module load warpx**
**module load py-warpx**

# BLAST Codes: Easy to Use, Extent, Tested and Documented

```python
from impactx import ImpactX, elements

sim = ImpactX()
# ...

# design the accelerator lattice)
ns = 25  # number of slices per ds in the element
fodo = [
    elements.Drift(ds=0.25, nslice=ns),
    elements.Quad(ds=1.0, k=1.0, nslice=ns),
    elements.Drift(ds=0.5, nslice=ns),
    elements.Quad(ds=1.0, k=-1.0, nslice=ns),
    elements.Drift(ds=0.25, nslice=ns),
    monitor,
]
# assign a fodo segment
sim.lattice.extend(fodo)

# run simulation
sim.evolve()
```

💡 **Same Script**
CPU/GPU & multi-node

Example: ImpactX FODO Cell Lattice

INSTALLATION
- Users
- Developers
- HPC

USAGE
- Run ImpactX
- Parameters: Python
- Parameters: Inputs File

⊟ Examples
- FODO Cell
- Chicane
- Constant Focusing Channel
- Constant Focusing Channel with Space Charge
- Expanding Beam in Free Space
- Kurth Distribution in a Periodic Focusing Channel
- Kurth Distribution in a Periodic Focusing Channel with Space Charge
- Acceleration by RF Cavities
- FODO Cell with RF
- FODO Cell, Chromatic
- Chain of thin multipoles
- A nonlinear focusing channel based on the IOTA nonlinear lens
- The "bare" linear lattice of the Fermilab IOTA storage ring

/ Examples — Edit on GitHub

## Examples

This section allows you to **download input files** that correspond to different physical situations or test different code features.

- FODO Cell
- Chicane
- Constant Focusing Channel
- Constant Focusing Channel with Space Charge
- Expanding Beam in Free Space
- Kurth Distribution in a Periodic Focusing Channel
- Kurth Distribution in a Periodic Focusing Channel with Space Charge
- Acceleration by RF Cavities
- FODO Cell with RF
- FODO Cell, Chromatic
- Chain of thin multipoles
- A nonlinear focusing channel based on the IOTA nonlinear lens
- The "bare" linear lattice of the Fermilab IOTA storage ring
- Solenoid channel
- Drift using a Pole-Face Rotation
- Soft-edge solenoid
- Soft-Edge Quadrupole
- Positron Channel
- Cyclotron
- Combined Function Bend
- Ballistic Compression Using a Short RF Element
- Test of a Transverse Kicker

github.com/**ECP-WarpX/impactx**

EXASCALE COMPUTING PROJECT — BERKELEY LAB — LDRD

# Toward an integrated ecosystem of codes with on-the-fly tunability

## Speed

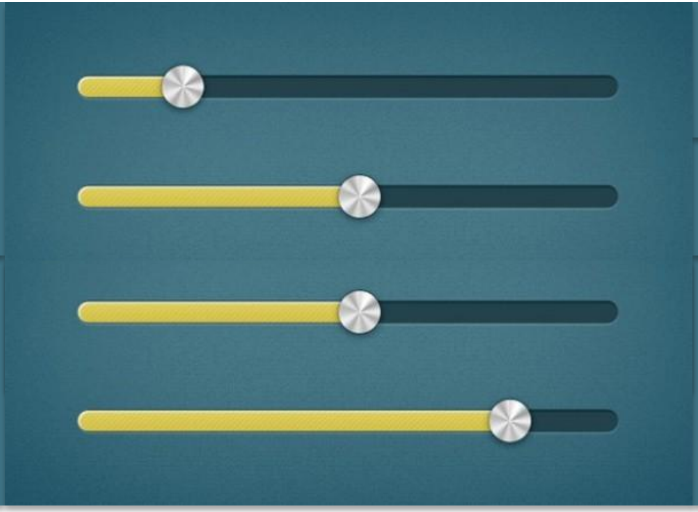
**Fast**
&
as
accurate
as
possible

Reduced physics

Reduced models

1D-1V

Low resolution



## Fidelity

Full physics

First principles

3D-3V

High resolution

**Accurate**
&
as
fast
as
possible

e.g., optimization & operations

e.g., exploration, training data

Rapid optimization with multi-fidelity models:
Talk seen on Wed by **Remi Lehe** (LBNL)

Ecosystem of codes
- share models & data between codes
- works best when standardized

# We Standardize - Let's Work Together

**Simulation code**

## Particle-In-Cell Modeling Interface

PICMI Standard

- Code A
- Code B
- ...

## open Particle Mesh Data standard

openPMD

as of 03/2024: **37+ projects**

## Facilitates:

- Chaining of codes for multiphysics workflow.
- Cross-benchmarking, verification, comparison.
- Interfacing with ensemble optimization, AI/ML software.
- Integration into frameworks.

optimas LASY


A Huebl et al., DOI:10.5281/zenodo.591699 (2015)
DP Grote et al., *Particle-In-Cell Modeling Interface (PICMI)* (2021)
LD Amorim et al., *GPos* (2021); M Thévenet et al., DOI:10.5281/zenodo.8277220 (2023)
A Ferran Pousa et al., DOI:10.5281/zenodo.7989119 (2023)
RT Sandberg et al., IPAC23, DOI:10.18429/JACoW-IPAC-23-WEPA101 (2023)


SciDAC Scientific Discovery through Advanced Computing

# GPU-accelerated ML surrogates

establishing rapid, fully accelerated, "in-the-loop" ML

# Augmenting & GPU-accelerating PIC Simulations & ML Models

**GPU Workflows are blazingly fast**

- PIC simulations
- Machine learning

*Can we augment & accelerate on-GPU PIC simulations with on-GPU ML models?*

```python
from pywarpx import picmi
import torch
# ...

# iterate all density boxes
for i in rho_device:
    rho = torch.as_tensor(
        rho_device.array(i),
        device="cuda")

    # apply ML in-memory
    with torch.no_grad():
        surrogate_model(rho)
```

## Compatible ecosystem between:


## Persistent GPU data placement

- read+write access, no CPU transfer

*Cross-Ecosystem, In Situ Coupling:*
Consortium for Python Data API Standards *data-apis.org*

# Use-Cases of these Python Bindings

Designed with two fundamental workflows in mind:

**expand BLAST codes**
from Python

- optimization workflows
- numerical prototyping
- modular code coupling
- in situ analysis
- interactive steering
- …
- ***data-science and AI/ML***
  - ***incl. AI in the loop***

write your own
**benchmarks**, **tests**

- interactive tutorials
- education
- app prototyping
- testing
- …
- Python purists ;-)

# Modular Software Architecture

BLAST
BEAM PLASMA & ACCELERATOR SIMULATION TOOLKIT

Python: Modules, PICMI interface, Workflows

**WarpX**
full PIC,
LPA/LPI

**ImpactX**
accelerator lattice design

**...**

**HiPACE++**
quasi-static,
PWFA

**ARTEMIS**
microelectronics

**ML Frameworks**
PyTorch,
Tensorflow, …

**pyAMReX**

**PICSAR**
QED Modules

**ABLASTR:** shared PIC

**AMReX**
Containers, Communication,
Portability, Utilities

**openPMD**
diagnostics

**Math**
FFTs,
lin. alg.

**CUDA, OpenMP, SYCL, HIP**

**MPI**

mac OS

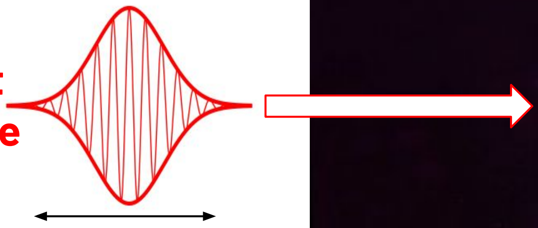
Desktop to HPC

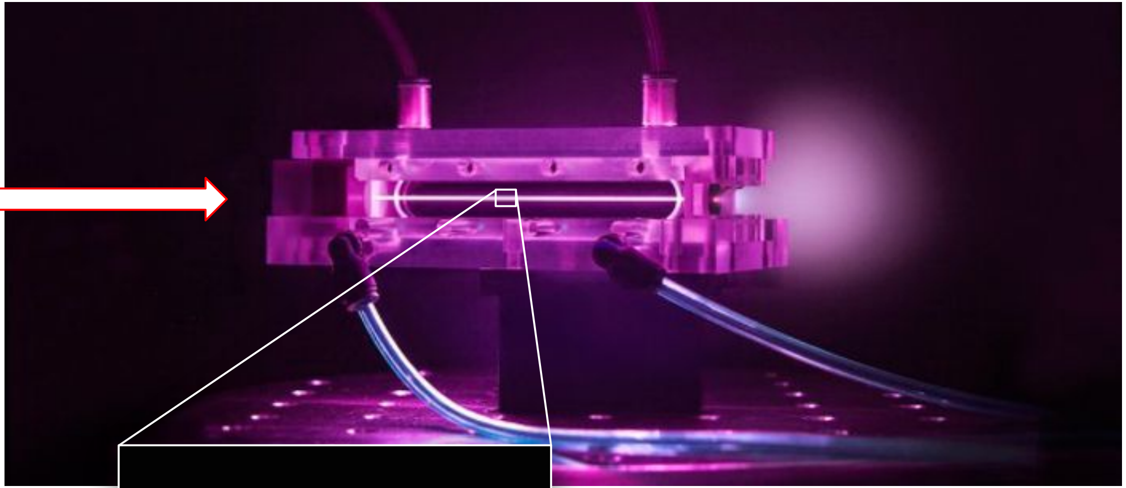
# Staging of LWFA for future HEP colliders

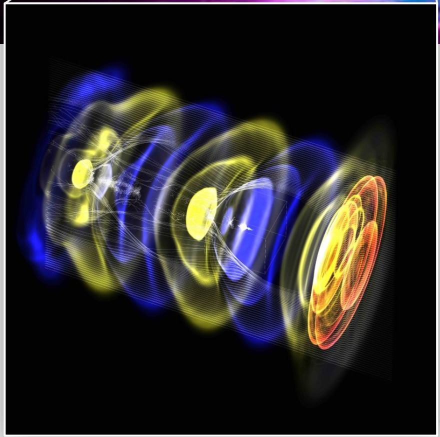
## Hybrid beamlines: plasma-transport modeling

# Laser-Wakefield Acceleration

**Petawatt laser pulse**





*RF* **<200 MV / m**

***LPA*** **100 000 MV / m**

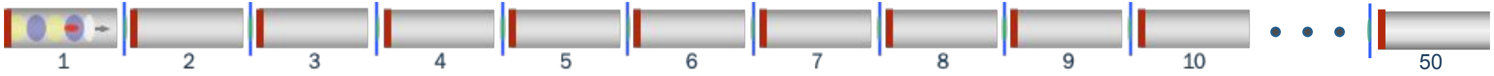
AJ Gonsalves et al., "Petawatt Laser Guiding and Electron Beam Acceleration to 8 GeV in a Laser-Heated Capillary Discharge Waveguide", Phys. Rev. Lett. 122, 084801 (2019)

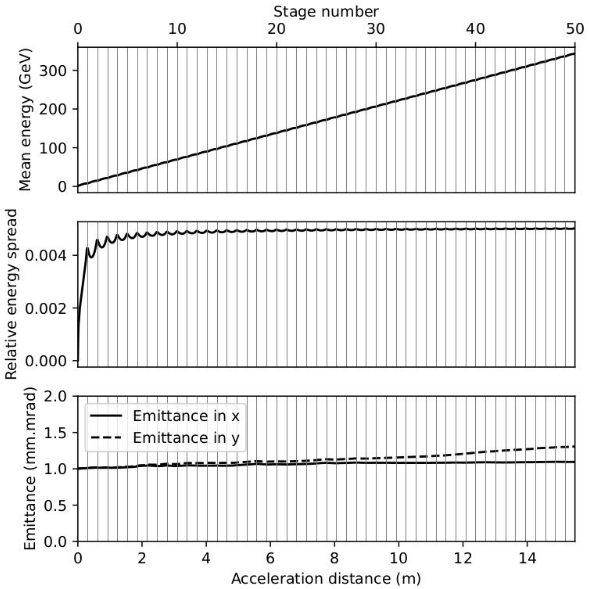
BELLA

BERKELEY LAB

# Future Collider Concept: Staging of LWFAs

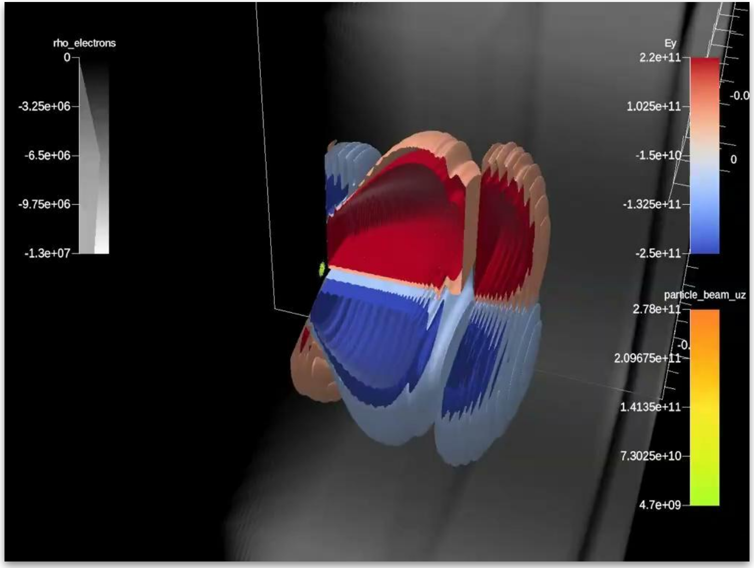
first 3D simulation of a chain of 50 plasma accelerator stages
for future colliders




**simulated transverse electric field**

WarpX on Frontier:
*Ascent & VTK-m*
*on 552 GPUs/GCDs*

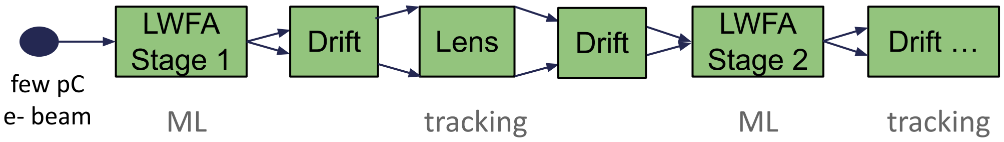




J-L Vay, A Huebl et al., PoP 28.2, 023105 (2021); N Marsaglia, et al., DOI:10.5281/zenodo.8226853 (2023)
J-L Vay et al., ECP WarpX MS FY23.1; A Ferran Pousa et al., IPAC23, DOI:10.18429/JACoW-IPAC2023-TUPA093 (2023)

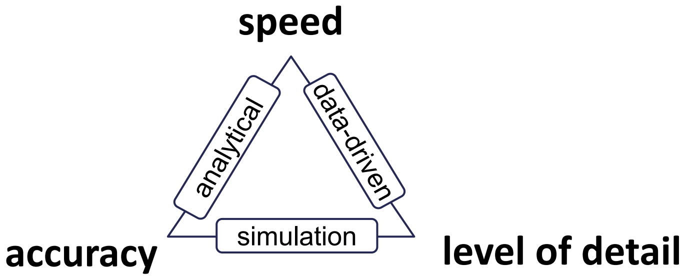
# Modeling Hybrid, Conventional + Plasma Beamlines

**ML boosted:** for a *specific* problem

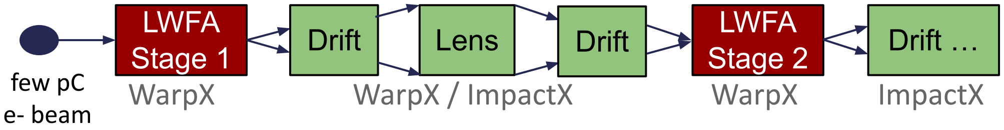



- start-to-end collider modeling
- digital twin / ‘real-time’

**Model Choice:** for complex, nonlinear, many-body systems ***pick two*** of the following




**Model Speed:** for accelerator elements




**Simulation time:** full geometry, full physics

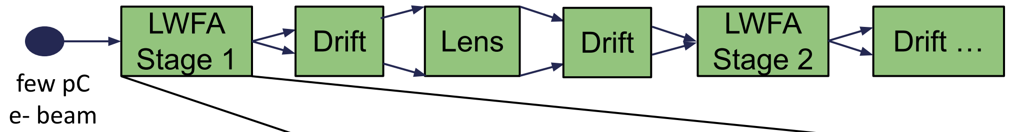
| | |
|---|---|
| hrs | <sec |
| 256 GPUs | 1 GPU |

**Fast surrogates:** Data-driven modeling is a potential middle ground between
- analytical modeling and
- full-fidelity simulations.

RT Sandberg et al and A Huebl, IPAC23, DOI:10.18429/JACoW-IPAC-23-WEPA101 (2023)
RT Sandberg et al. and A Huebl, PASC24 *accepted* (2024)

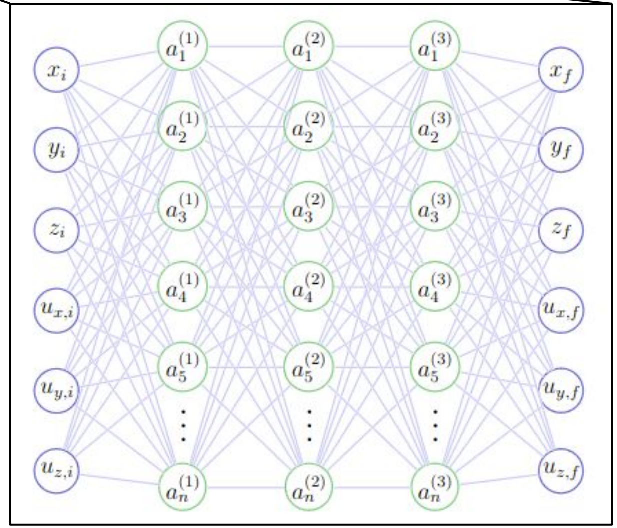
# We Trained a Neural Net with WarpX for Staging of Electrons

**one-time cost:** few hr WarpX sim + 10min training

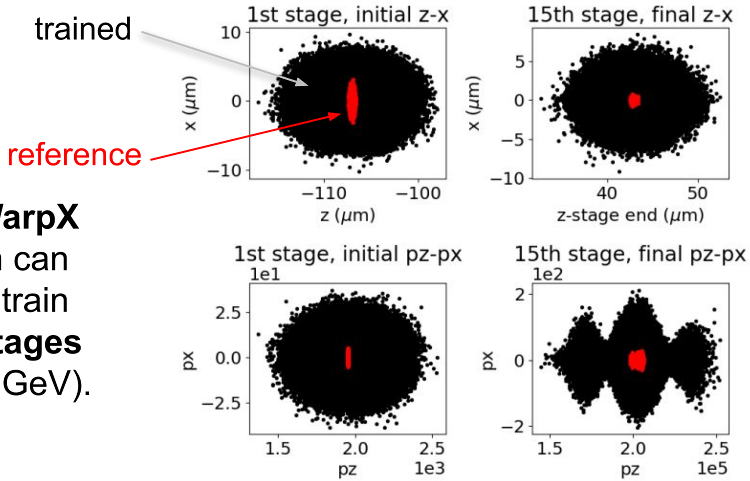


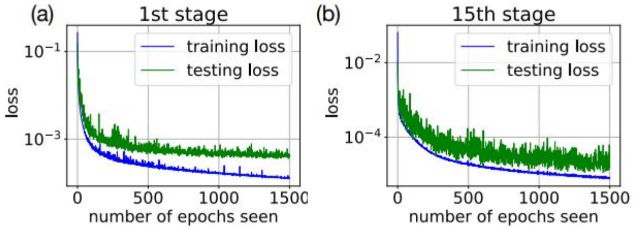

**A Neural Net is a non-linear transfer map!**

Assumption: purely tracking




A **single WarpX simulation** can be used to train **multiple stages** (7,14,21,…GeV).




**Hyperparameters**

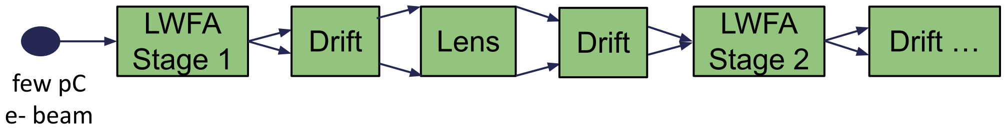
- 6D in 6D out
- 3-5 hidden layers with 700-900 nodes each are sufficient

Training data: 1M particles / beam
Training time: 2-2.2 hrs on 1 GPU

RT Sandberg et al and A Huebl, IPAC23, DOI:10.18429/JACoW-IPAC-23-WEPA101 (2023)
RT Sandberg et al. and A Huebl, PASC24 *accepted* (2024)

# Modeling + Inference are Fully GPU Accelerated

**one-time cost:** few hr WarpX sim + 10min training

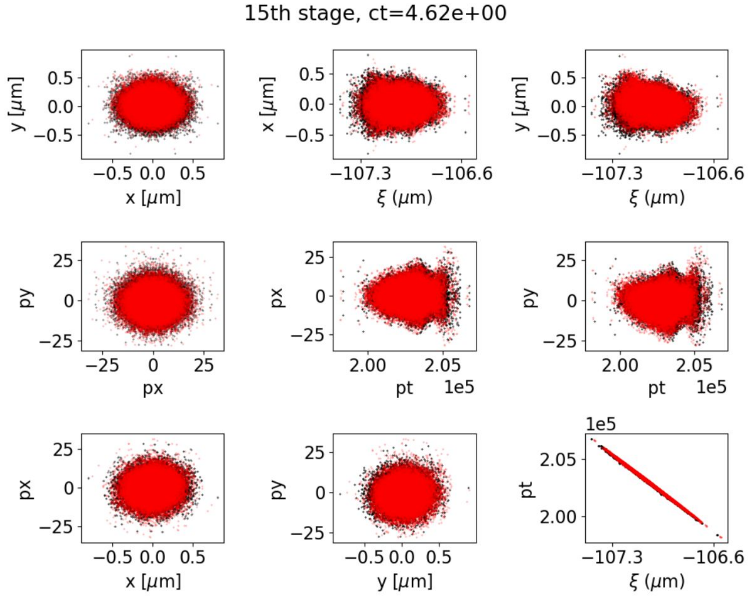



**ImpactX tracking 10M particles:** 10s on 1 GPU
**Inference time:** 63ns / particle / stage

**1st & 2nd order beam moments**
~0.1-1%-lvl error

ImpactX: 10 GPU sec
after 15 surrogates

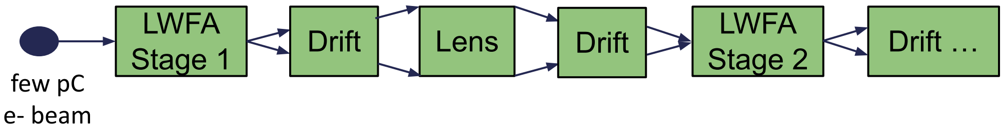
WarpX: 1,316 GPU hrs
15 stage simulation




RT Sandberg et al and A Huebl, IPAC23, DOI:10.18429/JACoW-IPAC-23-WEPA101 (2023)
RT Sandberg et al. and A Huebl, *accepted to PASC24*, arXiv:2402.17248 (2024)

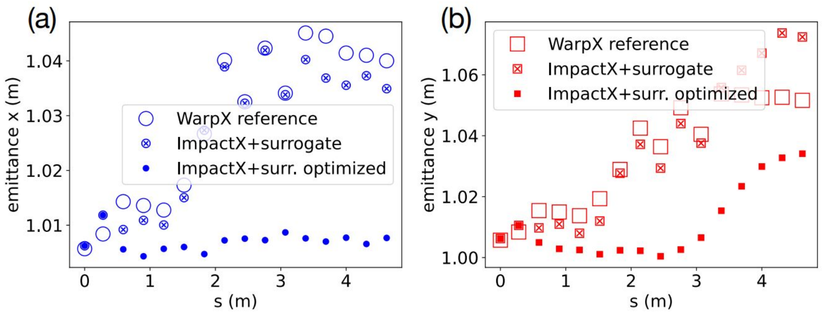
# Modeling + Inference are Fully GPU Accelerated

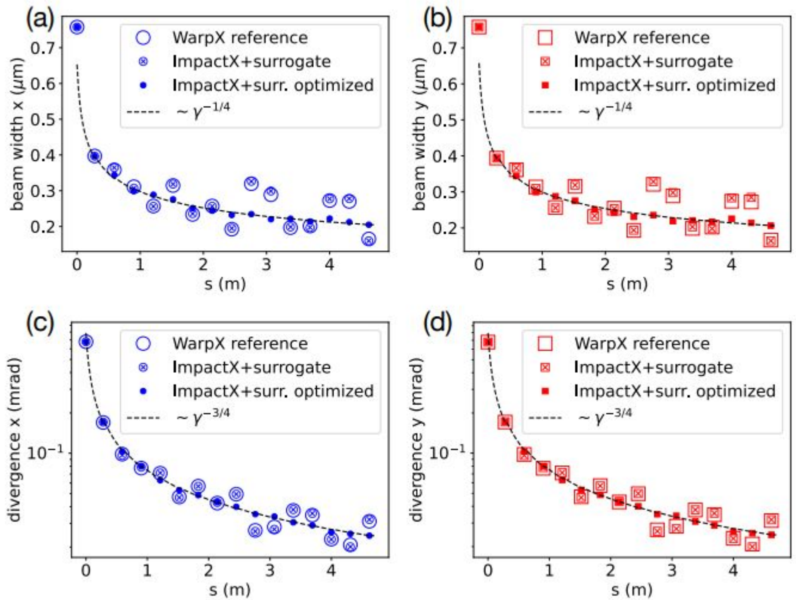
**one-time cost:** few hr WarpX sim + 10min training




**ImpactX tracking 10M particles:** 10s on 1 GPU
**Inference time:** 63ns / particle / stage

## Rapid Start-to-End Optimization for Transport Design




**Crucial, Open Challenges**
- microscopic *and* collective effects together: space charge
- better conserve beam moments

**feedback & collabs wanted**

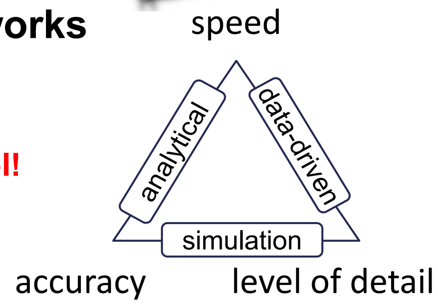
RT Sandberg et al and A Huebl, IPAC23, DOI:10.18429/JACoW-IPAC-23-WEPA101 (2023)
RT Sandberg et al. and A Huebl, *accepted to PASC24*, arXiv:2402.17248 (2024)

# Summary

- **BLAST** is a modular, fully open suite of **Exascale** PIC codes for **beam, laser-plasma & accelerator modeling**.
  - **WarpX** for time-based integration, e.g., injectors, LWFAs
  - **ImpactX** for s-based beam dynamics, e.g., linacs, rings, start2end


- Seamless, **GPU-Accelerated Coupling** of **AMReX/BLAST & ML Frameworks**
  - **zero-copy GPU data access**: in situ ML elements
  - Scripted: easy to **vary & research** new data models

**bring your own lattice & ML model!**




- **Vibrant Ecosystem and Contributions**
  - Runs on any platform: Linux, macOS, Windows - Laptop to HPC
  - Public development, automated testing, review & documentation
  - Friendly, open & helpful community

github.com/**ECP-WarpX**
github.com/**openPMD**
github.com/**AMReX-Codes**
github.com/**picmi-standard**

open source initiative®

# Contacts & Funding Acknowledgements


## Presenter

- Axel Huebl, axelhuebl@lbl.gov


## CAMPA Project (SciDAC-HEP)

- PI: Jean-Luc Vay, jlvay@lbl.gov
- campa.lbl.gov, blast.lbl.gov

open source initiative®

github.com/**ECP-WarpX**
github.com/**openPMD** **www.openPMD.org**
github.com/**AMReX-Codes**
github.com/**picmi-standard**
github.com/**UCLA-Plasma-Simulation-Group**
github.com/**fnalacceleratormodeling**


This research was supported by the **Exascale Computing Project** (17-SC-20-SC), a collaborative effort of two **U.S. Department of Energy organizations (Office of Science and the National Nuclear Security Administration)** responsible for the planning and preparation of a capable exascale ecosystem, including software, applications, hardware, advanced system engineering and early testbed platforms, in support of the nation's exascale computing imperative. This work was also performed in part by the **Laboratory Directed Research and Development Program of Lawrence Berkeley National Laboratory** under U.S. Department of Energy Contract No. DE-AC02-05CH11231, **Lawrence Livermore National Laboratory** under Contract No. DE-AC52–07NA27344 and **SLAC National Accelerator Laboratory** under Contract No. AC02–76SF00515. Supported by the **CAMPA collaboration**, a project of the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research and Office of High Energy Physics, **Scientific Discovery through Advanced Computing (SciDAC)** program. This research used resources of the **Oak Ridge Leadership Computing Facility**, which is a DOE Office of Science User Facility supported under Contract DE-AC05-00OR22725, the **National Energy Research Scientific Computing Center (NERSC)**, a U.S. Department of Energy Office of Science User Facility located at Lawrence Berkeley National Laboratory, operated under Contract No. DE-AC02-05CH11231, and the supercomputer Fugaku provided by **RIKEN**.

# Backup Slides

# Ad Hoc Uncertainty Quantification Attempt: Testing the Network

## Error of Beam Moments

| | combined beamline error | combined beamline relative error | stage 1 relative error | stage 2 relative error |
|---|---|---|---|---|
| $\langle x \rangle$ | -2.015e-08 | -6.337e-02 | 5.179e-02 | -3.916e-02 |
| $\sigma_x$ | 2.723e-09 | 8.565e-03 | -4.381e-03 | 4.288e-03 |
| $\langle u_x \rangle$ | -3.319e-01 | -9.887e-02 | -8.609e-02 | 2.814e-02 |
| $\sigma_{ux}$ | 1.710e-02 | 5.094e-03 | 1.047e-02 | 7.716e-03 |
| $\epsilon_x$ | 1.844e-08 | 1.747e-02 | 7.740e-03 | 9.912e-03 |
| $\langle y \rangle$ | -6.882e-10 | -2.155e-03 | 5.228e-02 | 1.585e-02 |
| $\sigma_y$ | 9.245e-09 | 2.895e-02 | -8.687e-04 | 6.412e-03 |
| $\langle u_y \rangle$ | -4.540e-01 | -1.328e-01 | -1.089e-02 | -1.243e-01 |
| $\sigma_{uy}$ | 9.856e-02 | 2.884e-02 | 3.411e-02 | 2.491e-03 |
| $\epsilon_y$ | 5.932e-08 | 5.509e-02 | 3.334e-02 | 5.899e-03 |
| $\langle z \rangle$ | -7.686e-09 | -7.506e-02 | -9.746e-04 | -2.561e-02 |
| $\sigma_z$ | -1.900e-11 | -1.855e-04 | -3.943e-04 | 2.927e-03 |
| $\langle u_z \rangle$ | 1.797e+00 | 6.148e-05 | 4.151e-04 | -3.769e-05 |
| $\sigma_{uz}$ | -1.088e+01 | -8.394e-02 | -8.186e-02 | -3.944e-02 |

Training data: 50,000 particles / beam

9 stage simulation (pre-review) for:
RT Sandberg et al. and A Huebl, *accepted to PASC24*, arXiv:2402.17248 (2024)

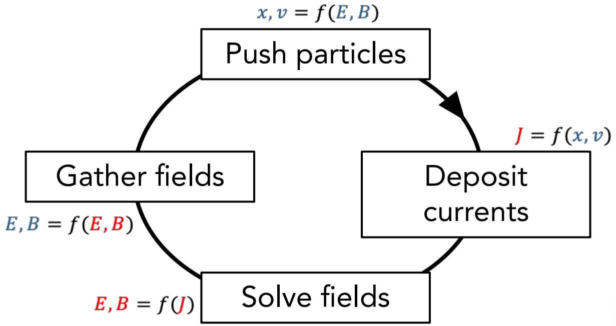
# Developed by a multidisciplinary, multi-institution team

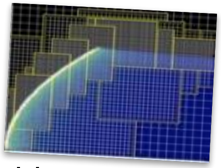
# WarpX is a GPU-Accelerated PIC Code for Exascale

BLAST
BEAM PLASMA & ACCELERATOR SIMULATION TOOLKIT

## Multiple Particle-in-Cell Loops

- electromagnetic or -static (time integration)



## Advanced algorithms

boosted frame, spectral solvers, Galilean frame, embedded boundaries + CAD, MR, …

## Multi-Physics Modules

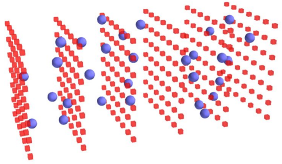
field ionization of atomic levels, Coulomb collisions, QED processes (e.g. pair creation), macroscopic materials, secondary emission

EXASCALE COMPUTING PROJECT

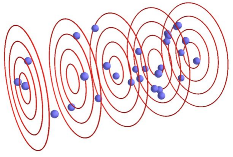
SciDAC Scientific Discovery through Advanced Computing

## Geometries

- 1D3V, 2D3V, 3D3V and RZ (quasi-cylindrical)



3D Cartesian grid



Cylindrical grid (schematic)

## Multi-Node parallelization

- MPI: 3D domain decomposition
- dynamic load balancing

## On-Node Parallelization

- GPU: CUDA, HIP and SYCL
- CPU: OpenMP

mac OS

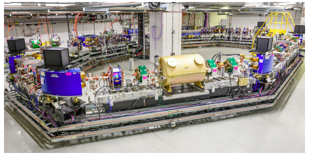
## Scalable & Standardized

- PICMI input
- openPMD (HDF5 or ADIOS)
- in situ: diagnostics & Python APIs

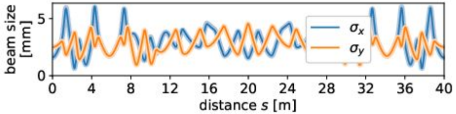
# ImpactX: GPU-, AMR- & AI/ML-Accelerated Beam Dynamics

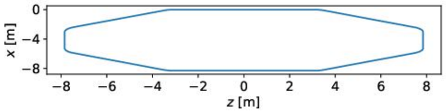
BLAST
BEAM PLASMA & ACCELERATOR SIMULATION TOOLKIT

## Particle-in-Cell Loop



- electrostatic
  - with space-charge effects
- s-based
  - relative to a reference particle
  - elements: <u>symplectic maps</u>



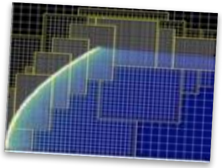




## Fireproof Numerics

based on IMPACT suite of codes, esp. IMPACT-Z and MaryLie

## Triple Acceleration Approach

- GPU support
- Adaptive Mesh Refinement
- AI/ML & Data Driven Models



BERKELEY LAB LDRD

SciDAC Scientific Discovery through Advanced Computing

## User-Friendly

- single-source C++, full Python control
- fully tested
- fully documented

## Multi-Node parallelization

- MPI: domain decomposition
- dynamic load balancing (in dev.)

## On-Node Parallelization

- GPU: CUDA, HIP and SYCL
- CPU: OpenMP

## Scalable & Standardized

- openPMD (HDF5 or ADIOS)
- in situ: diagnostics & Python APIs

# Active Standardization Efforts

## Standardization…

- Inputs
- Data
- Reference Implementations

strong int. partnerships

Particle-In-Cell Modeling Interface

PICMI Standard

↓

Code A

Code B

...

↓

open Particle Mesh Data standard

openPMD

## … Accelerates Innovation

- *LASY*
  github.com/**LASY-org**
- optimas
  github.com/**optimas-org**
- BLAST + Geant4
  github.com/LDAmorim/**GPos**
- **easy ML training**

open source initiative®

A Huebl et al., DOI:10.5281/zenodo.591699 (2015)
DP Grote et al., *Particle-In-Cell Modeling Interface (PICMI)* (2021)
LD Amorim et al., *GPos* (2021); M Thévenet et al., DOI:10.5281/zenodo.8277220 (2023)
A Ferran Pousa et al., DOI:10.5281/zenodo.7989119 (2023)
RT Sandberg et al., IPAC23, DOI:10.18429/JACoW-IPAC-23-WEPA101 (2023)

# ML Surrogates: A Sensible Target for T/PBytes of Data

## *Bridging model time scales with data-driven methods.*

**Things that run very fast on GPU:**

- our PIC simulations
- machine learning

*Can we augment & accelerate on-GPU PIC simulations with on-GPU ML models?*

**A) Training** (slow)

- Offline: WarpX openPMD → Neural Network
- Online (*in situ*): advanced ML methods

**B) Inference:** *in situ* to codes (fast)

- Zero-copy data access: <u>*persistently on GPU*</u>
- Example: an *ML map* in beam dynamics

**Model Speed:** for accelerator elements

| Plasma Source | Trans-port | Plasma Stage | Plasma Stage | Injector |
|---|---|---|---|---|
| WarpX | ImpactX | WarpX | HiPACE++ | WarpX-ES |

**Simulation time:** full geometry, full physics

| hrs | sec | hrs | hrs | min |
|---|---|---|---|---|

**ML boosted:** for a *specific* problem

| LWFA w/ iinj. | Trans-port | LWFA Stage | PWFA Stage | Kicker Magnet |
|---|---|---|---|---|
| ML | ImpactX | ML | ML | ML |

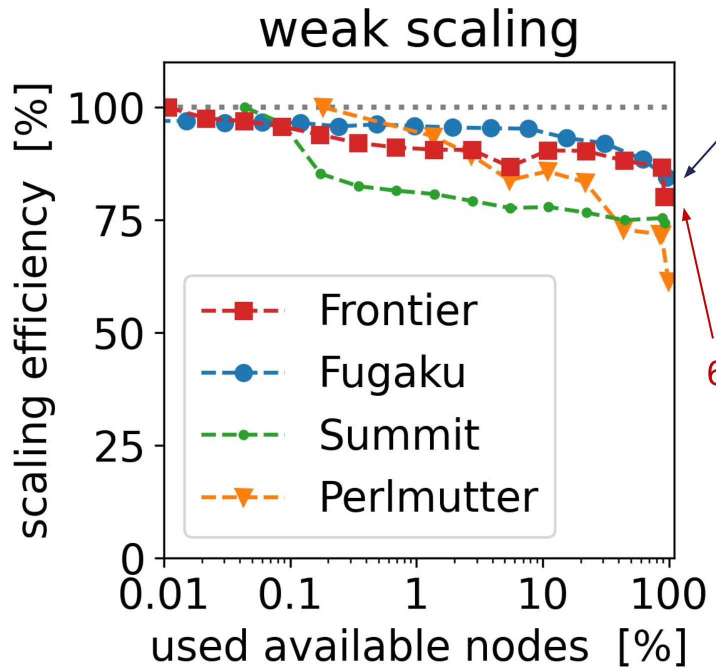
- start-to-end collider modeling
- digital twin / ‘real-time’

A Huebl et al., NAPAC22, DOI:10.18429/JACoW-NAPAC2022-TUYE2 (2022)
RT Sandberg et al and A Huebl, IPAC23, DOI:10.18429/JACoW-IPAC-23-WEPA101 (2023)
A Huebl et al., AAC22, arXiv:2303.12873 (2023); RT Sandberg et al. and A Huebl, *accepted, PASC24* (2024)

# WarpX Scales to the World's Largest HPCs

## April-July 2022: WarpX on **world's largest HPCs**

L. Fedeli, A. Huebl et al., *Gordon Bell Prize Winner* at SC'22, 2022




7,299,072 CPU Cores

68,608 GPUs of *First Exascale* Machine

*Note: Perlmutter & Frontier were pre-acceptance measurements!*

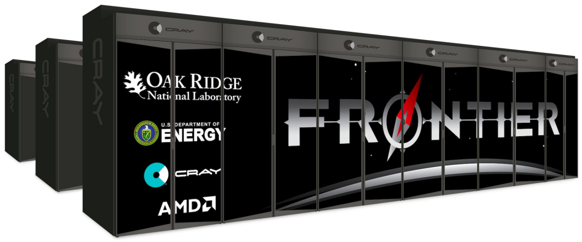


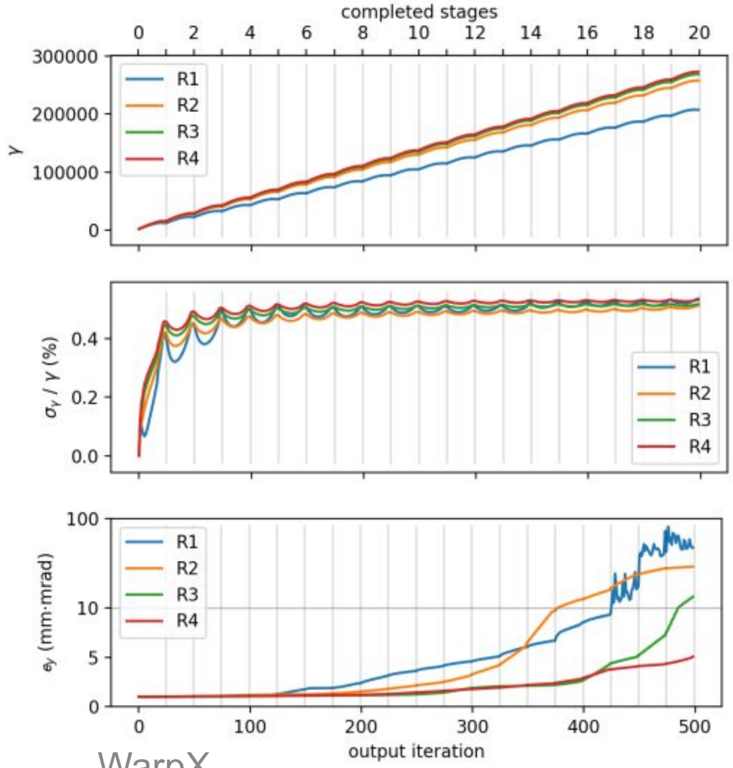

**Figure-of-Merit**: weighted updates / sec

| Date | Code | Machine | $N_c$/Node | Nodes | FOM |
|---|---|---|---|---|---|
| 3/19 | Warp | Cori | 0.4e7 | 6 625 | 2.2e10 |
| 3/19 | WarpX | Cori | 0.4e7 | 6 625 | 1.0e11 |
| 6/19 | WarpX | Summit | 2.8e7 | 1 000 | 7.8e11 |
| 9/19 | WarpX | Summit | 2.3e7 | 2 560 | 6.8e11 |
| 1/20 | WarpX | Summit | 2.3e7 | 2 560 | 1.0e12 |
| 2/20 | WarpX | Summit | 2.5e7 | 4 263 | 1.2e12 |
| 6/20 | WarpX | Summit | 2.0e7 | 4 263 | 1.4e12 |
| 7/20 | WarpX | Summit | 2.0e8 | 4 263 | 2.5e12 |
| 3/21 | WarpX | Summit | 2.0e8 | 4 263 | 2.9e12 |
| 6/21 | WarpX | Summit | 2.0e8 | 4 263 | 2.7e12 |
| 7/21 | WarpX | Perlmutter | 2.7e8 | 960 | 1.1e12 |
| 12/21 | WarpX | Summit | 2.0e8 | 4 263 | 3.3e12 |
| 4/22 | WarpX | Perlmutter | 4.0e8 | 928 | 1.0e12 |
| 4/22 | WarpX | Perlmutter† | 4.0e8 | 928 | 1.4e12 |
| 4/22 | WarpX | Summit | 2.0e8 | 4 263 | 3.4e12 |
| 4/22 | WarpX | Fugaku† | 3.1e6 | 98 304 | 8.1e12 |
| 6/22 | WarpX | Perlmutter | 4.4e8 | 1 088 | 1.0e12 |
| 7/22 | WarpX | Fugaku | 3.1e6 | 98 304 | 2.2e12 |
| 7/22 | WarpX | Fugaku† | 3.1e6 | 152 064 | 9.3e12 |
| 7/22 | WarpX | Frontier | 8.1e8 | 8 576 | 1.1e13 |

110x

500x

# libEnsemble: Design Optimization I




WarpX

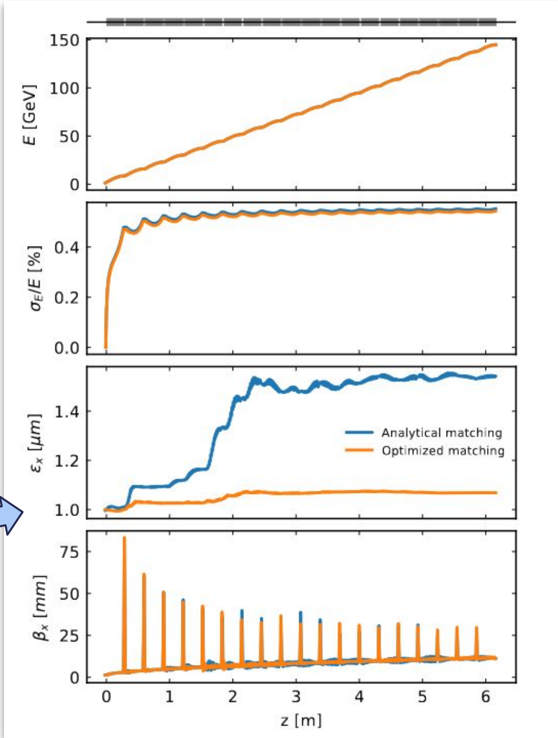
## Staged LPA

Beam Emittance Preservation

3. converge
   3D
4. optimize

1. optimize
   low-D, redu.

2. inform
   3D




Wake-T, libEnsemble

J.-L. Vay et al., ECP WarpX MS FY23.1; A. Ferran Pousa et al., IPAC23, DOI:10.18429/JACoW-IPAC2023-TUPA093 (2023)

# GPU Performance In Practice: Highly Asynchronous

Nvidia Nsight Systems trace files of ImpactX under DOI:10.5281/zenodo.10723742