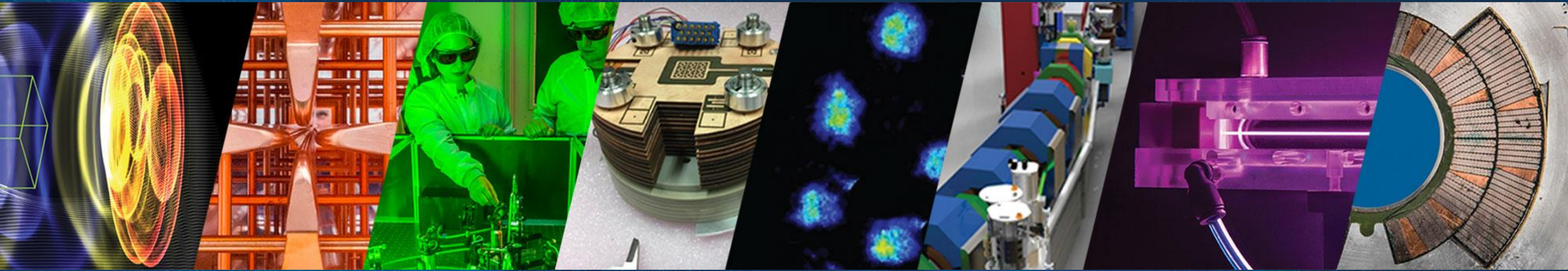
A next generation, integrated community toolset for the modeling of linear colliders

Jean-Luc Vay, Arianna Formenti, Marco Garten, Axel Huebl, Remi Lehe, Chad Mitchell, Ji Qiang, Olga Shapoval, Edoardo Zoni



International Workshop on Future Linear Colliders, LCWS2024

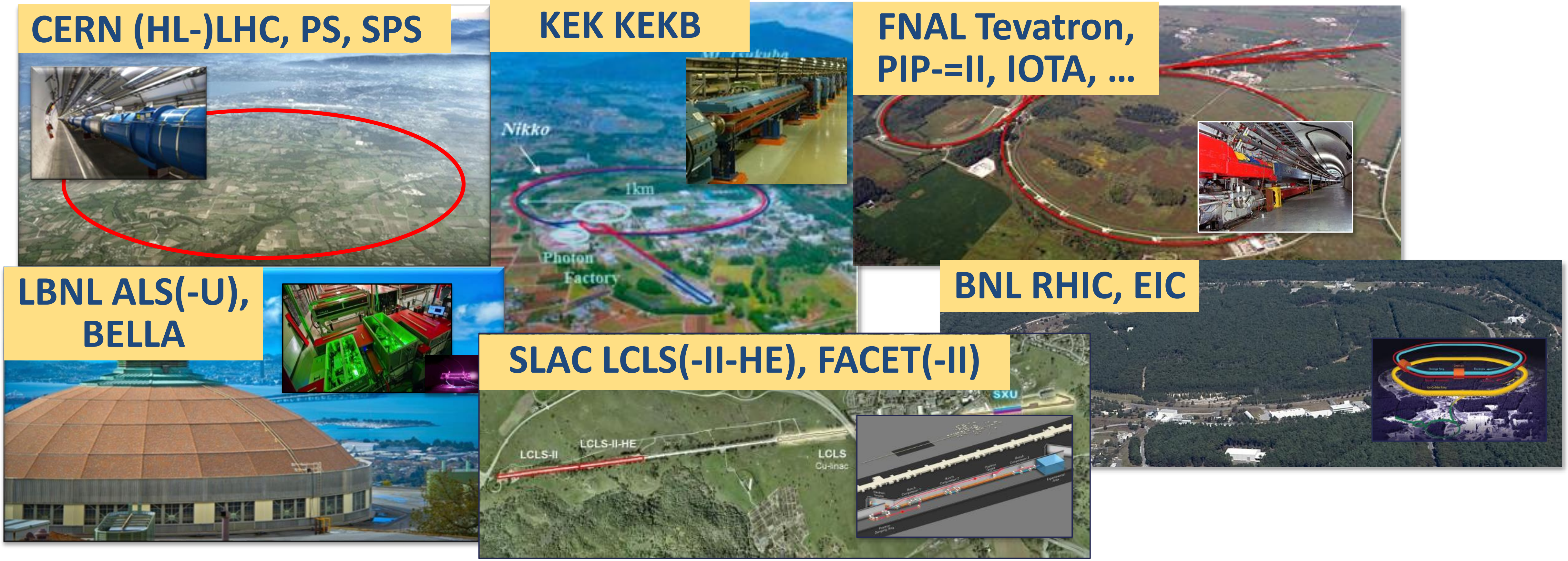
July 8-11, 2024 – The University of Tokyo, Japan

BERKELEY LAB

Advanced Modeling Program
ACCELERATOR TECHNOLOGY & APPLIED PHYSICS DIVISION ATAP

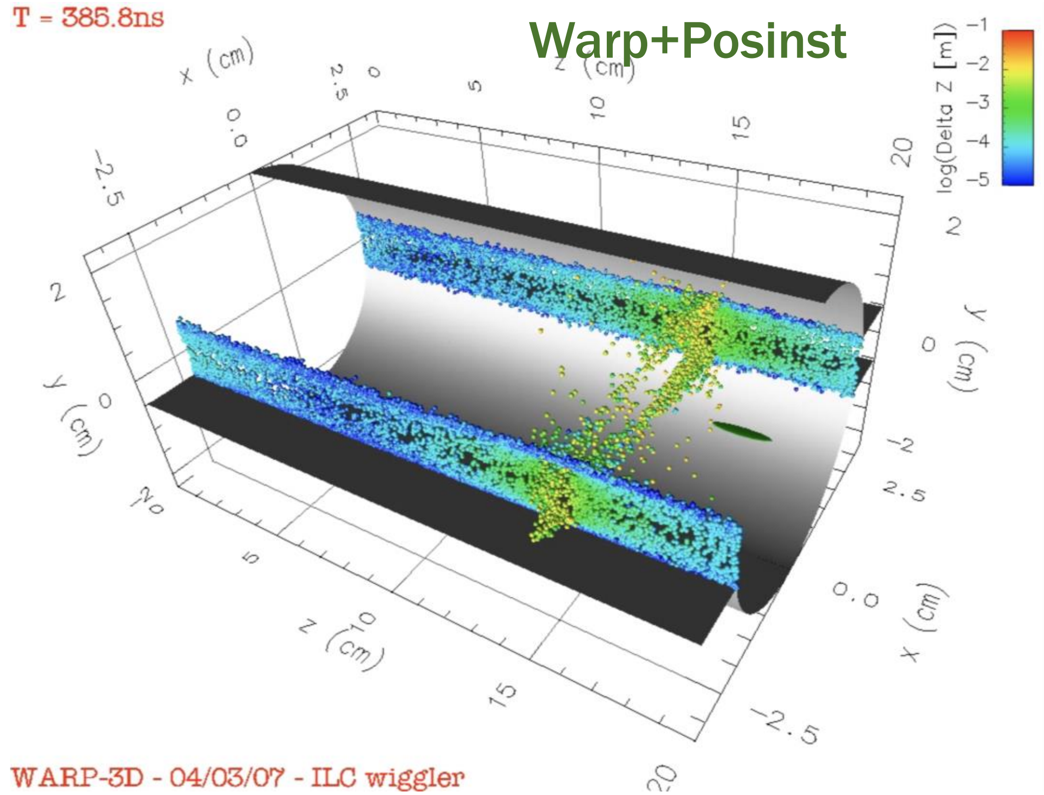
U.S. DEPARTMENT OF ENERGY | Office of Science

Berkeley Lab has long been home to state-of-the-art modeling of particle accelerators



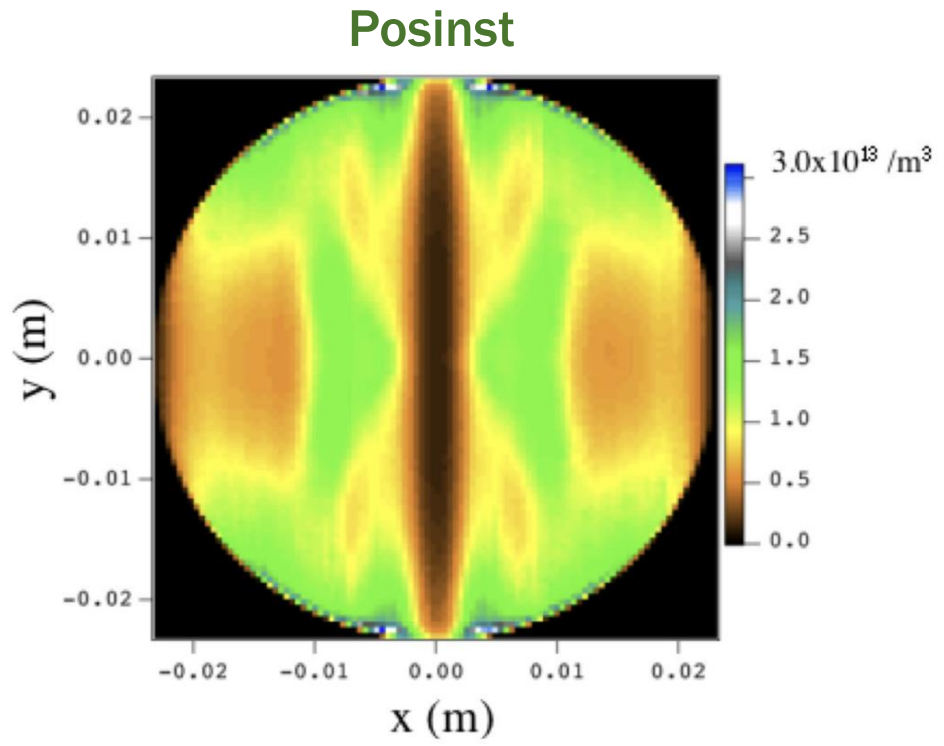
Sample of applications

Berkeley Lab has long been home to state-of-the-art modeling of particle accelerators



PIC calculations of the e-cloud in the ILC positron damping ring wigglers

C. M. Celata, M. A. Furman, J.-L. Vay, D. P. Grote, *Proc. PAC07* (2007)
C. M. Celata, M. A. Furman, J.-L. Vay, D. P. Grote, *Proc. ECLOUD07* (2007)

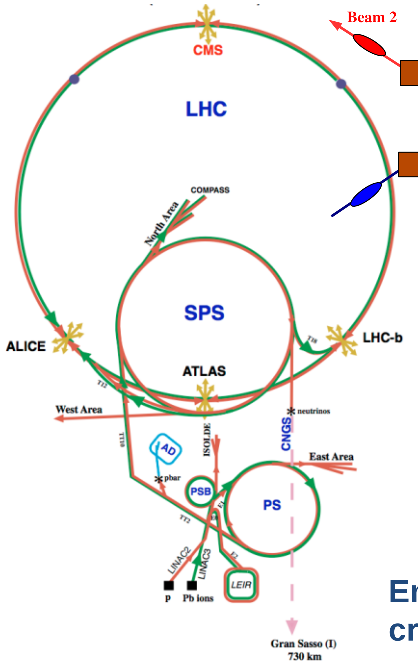


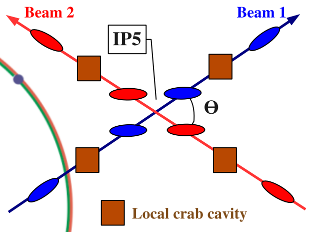
Electron cloud cyclotron resonances in the presence of a short-bunch-length relativistic beam

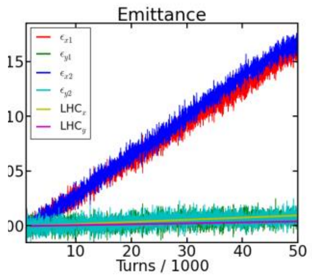
C. M. Celata, M. A. Furman, J.-L. Vay, J. W. Yu, *PRAB* **11**, 091002 (2008)

Berkeley Lab has long been home to state-of-the-art modeling of particle accelerators

BeamBeam3D





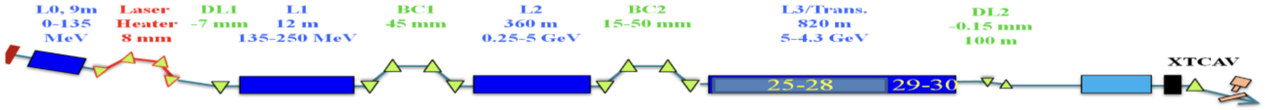


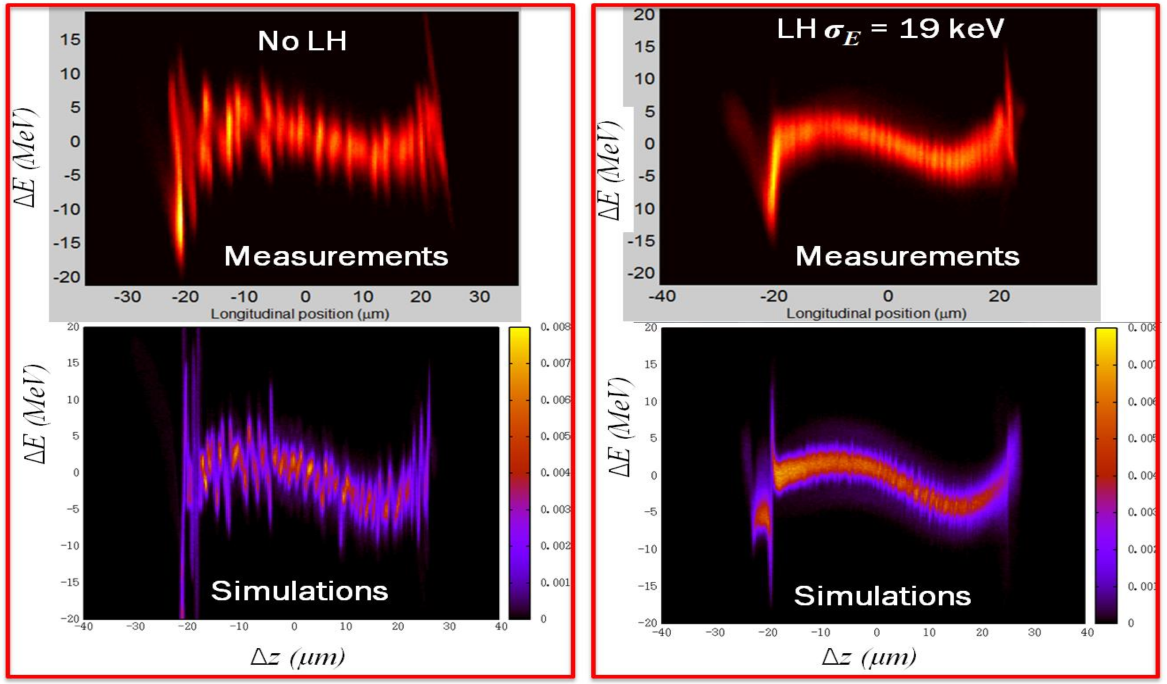
Emittance growth due to crab cavity RF noise.

Strong beam-beam simulation of interactions in the LHC upgrade.

J. Qiang *et al.*, *Proc. IPAC 2015*, Richmond, VA (2015).

Impact-T + Impact-Z (+ Genesis)





Start-to-end, one-to-one modeling reproduces microbunching in the LCLS X-ray FEL.

J. Qiang *et al.*, *Phys. Rev. Accel. Beams* **20**, 054402 (2017).

The Berkeley Lab Accelerator Simulation Toolkit was created to coordinate Berkeley Lab codes

BLAST
BERKELEY LAB ACCELERATOR SIMULATION TOOLKIT

blast.lbl.gov
Open Source

2014

Codes:
- BeamBeam3D
- Impact-T, Impact-Z
- Marylie/Impact
- Posinst
- Warp

Applications:

Start-to-end accelerators
- beams, plasmas, lasers, structures
- rings, linacs, sources, injectors
- RF, plasma, dielectric acceleration
- conventional & plasma-based focusing
- e-cloud
- CSR
- cooling
- collisions
- beam-beam @ IP
- …

Codes were successfully used by accelerator community on major projects but:
- coordination of development was limited by legacy
- amount of duplication was growing
- number of additional physics modules, codes & contributions (from various institutions) was increasing
- need to modernize codes for increasing number of levels of parallelism and GPUs

➔ new (more inclusive) name & coordination of codes development.

The *B*eam, p*L*asma & *A*ccelerator *S*imulation *T*oolkit followed to better coordinate & modernize codes

blast.lbl.gov
Open Source

2024

Codes:

- BeamBeam3D
- Impact-T, Impact-Z
- Marylie/Impact
- Posinst
- Warp
- FBPIC
- HiPACE++
- ImpactX
- LW3D
- Wake-T
- WarpX

Standards:

data	openPMD
input	PICMI Standard
lasers	LASY
optimization	optimas

Applications:

Start-to-end accelerators
- beams, plasmas, lasers, structures
- rings, linacs, sources, injectors
- RF, plasma, dielectric acceleration
- conventional & plasma-based focusing
- e-cloud
- CSR
- cooling
- collisions
- beam-beam @ IP
- QED effects
- ...

Plasma & fusion devices

and more (astrophysics, thermionics, microelectronics, ...)

Codes were successfully used by accelerator community on major projects but:

- coordination of development was limited by legacy
- amount of duplication was growing
- number of additional physics modules, codes & contributions (from various institutions) was increasing
- need to modernize codes for increasing number of levels of parallelism and GPUs

➔ **new (more inclusive) name & coordination of codes development.**

BLAST is a unique suite of interoperable codes & a multi-institutional international collaboration (>80 contributors, incl. from private sector).

BLAST
BEAM PLASMA & ACCELERATOR SIMULATION TOOLKIT

Developed by an international, multidisciplinary team

physicists + applied mathematicians + computational scientists + software engineers

open source initiative
THE LINUX FOUNDATION
HIGH PERFORMANCE SOFTWARE FOUNDATION

BLAST: a cutting-edge open-source simulation toolkit for end-to-end accelerator modeling

blast.lbl.gov

BLAST — BEAM PLASMA & ACCELERATOR SIMULATION TOOLKIT

Group	Code	Year started	Dimensionality	Independent variable	Solver	Symplectic maps	ML surrogate elements	Language	Parallel (multinode)	CPU and GPU	Multi-vendor GPU	Linac	Ring	Source	Wakefield accelerators	Beam-Beam	QED	E-cloud	IBS	CSR	Spin tracking	...
General purpose (legacy)	Warp	1989	2/3/RZ++	t/s/ξ	ES/EM/QS			For./Python	✓			✓	✓	✓	✓			✓		3D		...
	Impact-Z	1999	3	s	ES	✓		Fortran	✓			✓	✓						✓	1D		...
	Impact-T	2002	3	t	ES			Fortran	✓			✓		✓					✓			...
	Marylie/IMPACT	2006	3	s	ES	✓		Fortran	✓			✓	✓									...
Specialized	Posinst	2002	2	t	ES			Fortran										✓				...
	BeamBeam3D	2003	2.5	t,s	ES	✓		Fortran	✓				✓			✓						...
	FBPIC	2015	RZ++	t	EM			Python	✓	✓					✓						✓	...
	LW3D	2018	3	t	LW			Fortran	✓											3D		...
	Wake-T	2019	RZ	ξ	QS			Python				✓			✓					1D		...
New integrated	WarpX	2016	1/2/3/RZ++	t	ES/EM		**	C++/Python	✓	✓	✓	*	**	✓	✓	✓	✓		**	3D	**	...
	HiPACE++	2022	3	ξ	QS			C++/Python	✓	✓	✓				✓						✓	...
	ImpactX	2022	3	s	ES	✓	✓	C++/Python	✓	✓	✓	✓	✓						**	1D*, ML*	**	...

* in development
** planned, seeking additional funding

ES=Electrostatic; EM=Electromagnetic; QS=Quasistatic; LW=Lienard-Wiechert; ML=Machine Learning Model

standards & workflows

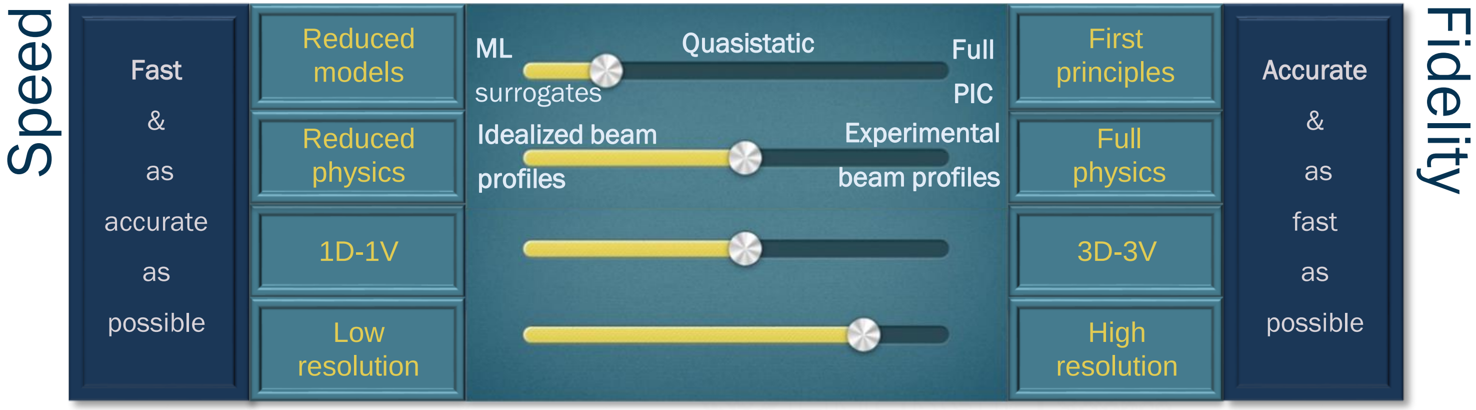
Data	openPMD
Input	PICMI Standard
Lasers	LASY
Optimize	optimas

GitHub

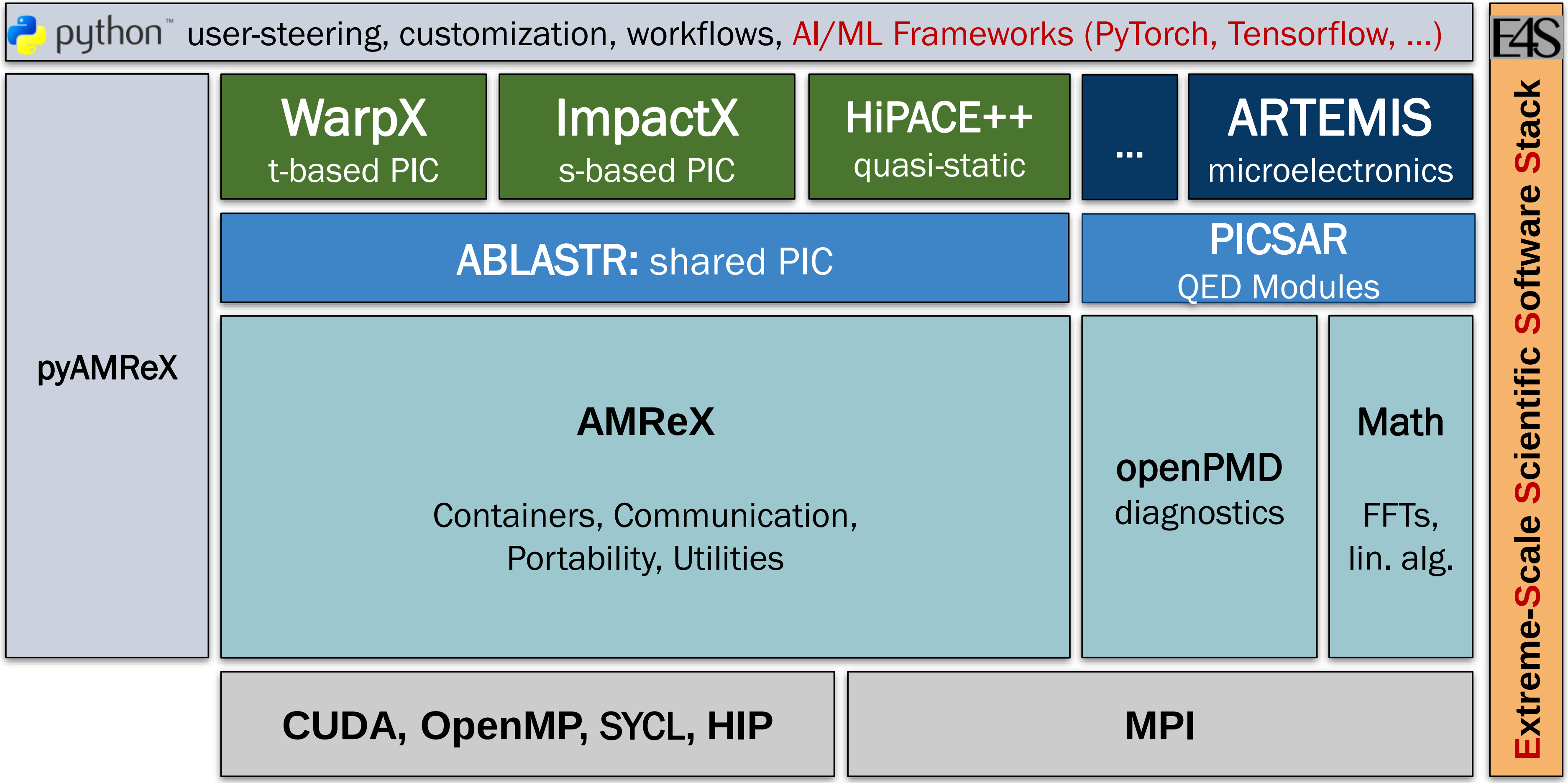
New codes are integrated around a common high-performance infrastructure w/CPU+GPU+ML support

- all types of colliders: Higgs factory, 10 TeV parton, muon, plasma-based, …
- tunability from **fast modeling** to **detailed physics** studies for collider design.

E.g., for a plasma-based collider:
ML surrogate ➔ Wake-T ➔ HiPACE++➔WarpX



AMP leverages the ECP technology (E4S) underpinning WarpX
➔high-performance, integrated suite for particle accelerator modeling (& more)



E4S is **unique in the world:**
➔ advantage of **unparallel performance & portability.**

Python interface:
- **Modular approach pioneered by Warp 20+ years ago**
  ➔Coupling to other codes (e.g., Posinst, ICOOL, …)
- **Access to powerful AI/ML tools**

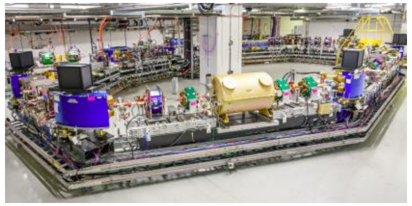
EXASCALE COMPUTING PROJECT
Laboratory Directed Research & Development
BERKELEY LAB
SciDAC
Scientific Discovery through Advanced Computing

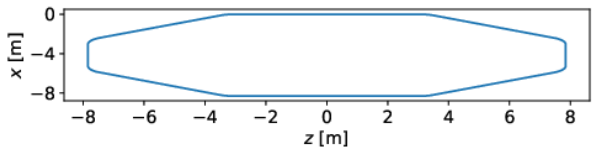
➔ Propose to leverage for faster & larger scale modeling for colliders (all types) R&D

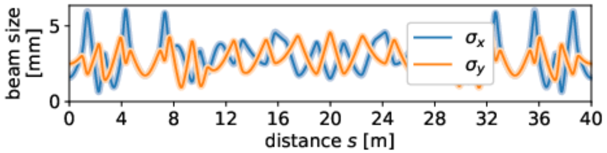
ImpactX aims at high(er) performance modeling of RF Accelerator Modeling

Beam-Dynamics in Linacs, Rings, Colliders

- intense beams, long-term dynamics
- **HEP colliders: Higgs factory, 10 TeV parton, ee, hh, $\gamma\gamma$, muons, …**
- Example of benchmark against Impact-Z on IOTA ring beam dynamics





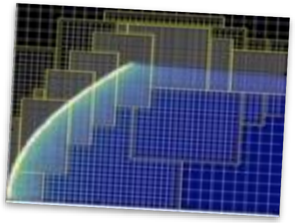


Advanced Numerics

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

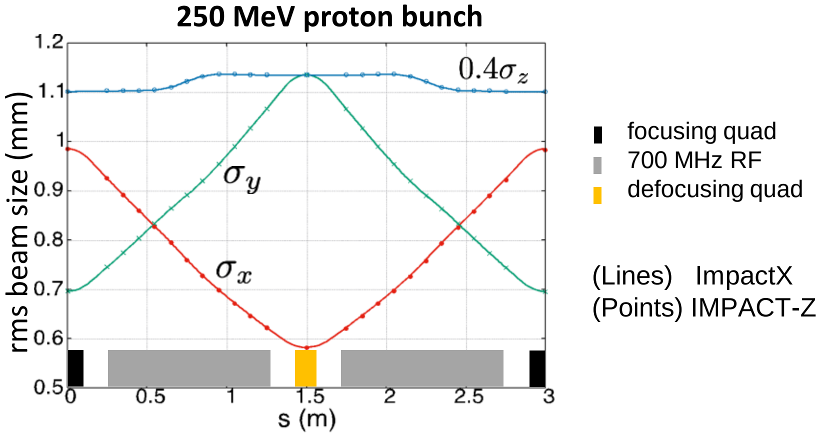
Triple Acceleration Approach

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



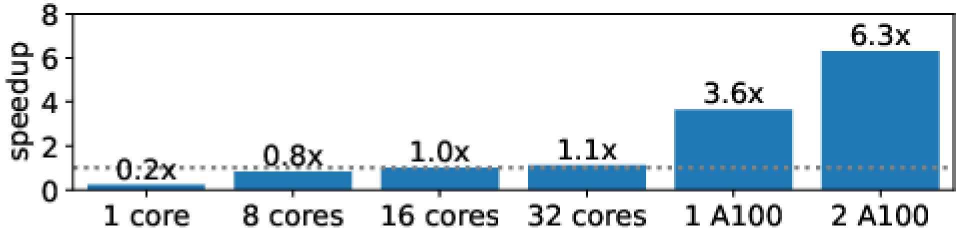
Benchmarks & Validations

- 86 continuously run benchmarks
- code-to-code comparisons



Performance

- order-of-magnitude perf. ↗ from GPUs

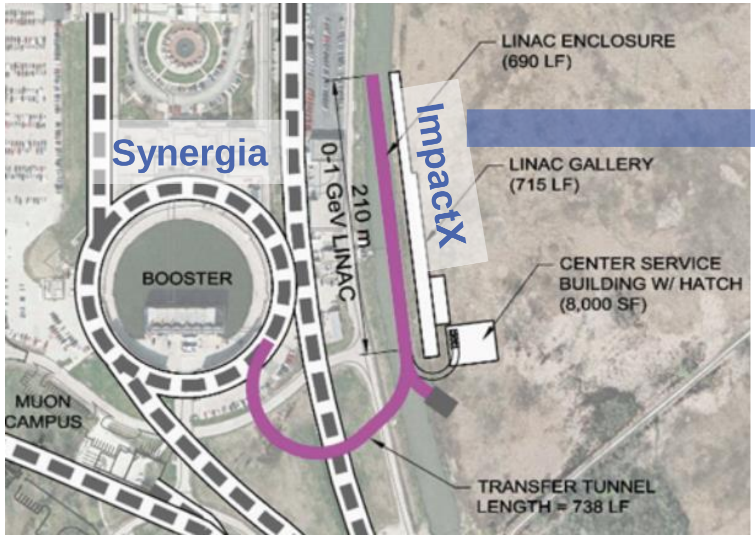


C Mitchell et al., HB2023, THBP44 and TUA2I2 (2023); A Huebl et al., NAPAC22 and AAC22 (2022); J Qiang et al., PRSTAB (2006); RD Ryne et al., ICAP2006 ICAP2006 (2006)

BERKELEY LAB LDRD SciDAC Scientific Discovery through Advanced Computing

ImpactX being applied to (& benchmarked on) high intensity beams in PIP-II

Proton Improvement Plan II (Fermilab)



Goal:
start-to-end modeling and design tuning (virtual test stand)

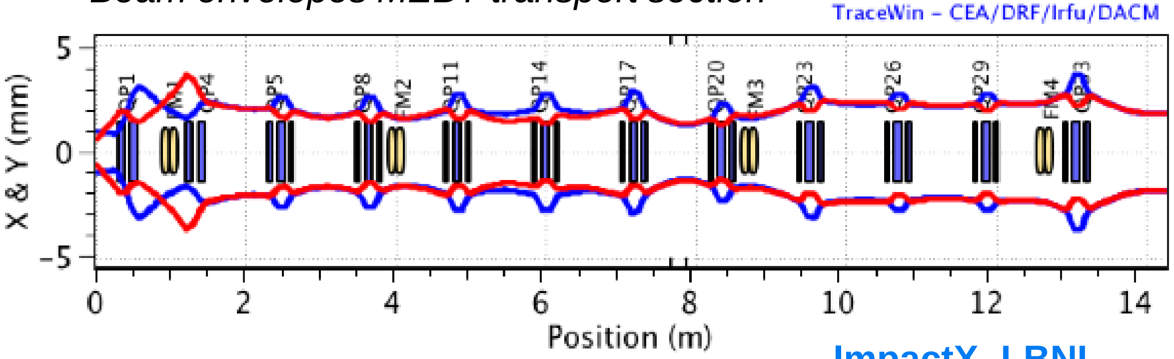
to Synergia for
modeling the Booster

SciDAC
Scientific Discovery through Advanced Computing

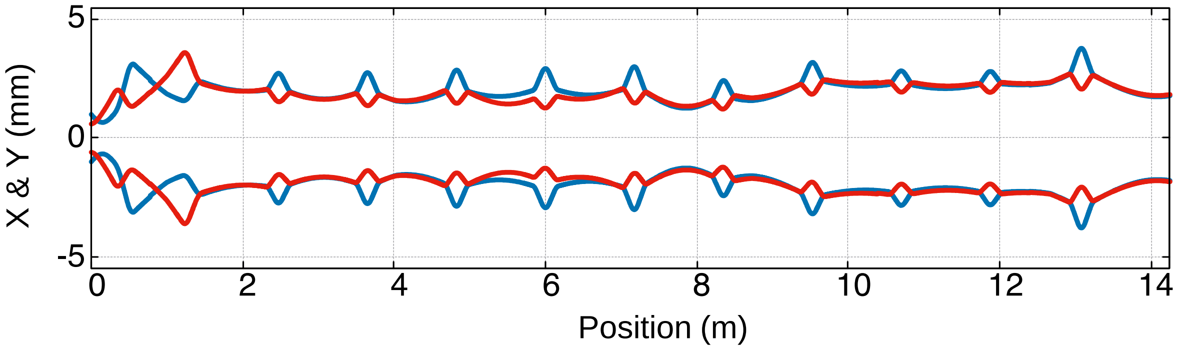
Fermilab SLAC NATIONAL ACCELERATOR LABORATORY
CAMPA
UCLA
Argonne NATIONAL LABORATORY
OAK RIDGE National Laboratory

5 mA beam modeling of the PIP-II linac

Beam envelopes MEBT transport section



ImpactX- LBNL

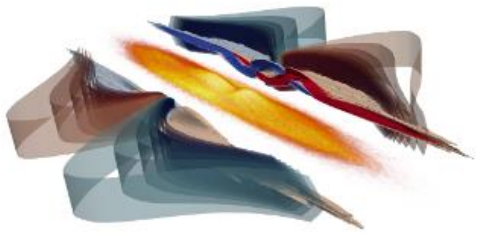


- Now: Benchmark against existing results using new parallel GPU-capable tools (ImpactX) for validation.
- Goal: Higher speed, resolution, and fidelity modeling.

Samples of ongoing engagement in using next-generation BLAST tools to model future HEP colliders

Collider interaction point modeling using WarpX:

- collaboration with SLAC, CEA Saclay
- benchmarked against GUINEA-PIG



WarpX simulation of beam-beam crossing at the interaction point of the ILC

See next talk by A. Formenti

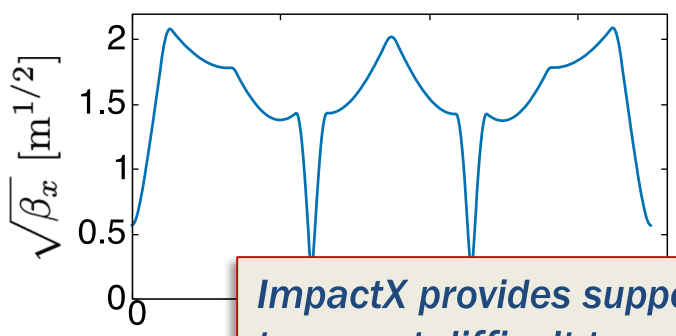
Beam transport challenges for a laser plasma acceleration collider:

- Focusing of beams with large energy spread
- Compact transport ➡ plasma lenses for focusing
- Preservation of beam emittance (100's of stages)
- Insensitivity to beam jitter and pointer errors

See *Advanced Accelerator Concepts 2 session: this afternoon*

Exploring gap transport designs using ImpactX:

- collaboration with HALHF group (Univ. Oslo)
- apochromatic transport concept
- transversely-tapered plasma lens concept



Initial 1% energy spread

Emittance growth:

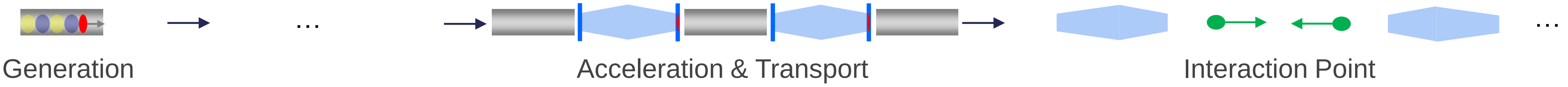
$$\Delta\epsilon_x/\epsilon_x \approx 0.1\%$$

ImpactX provides support for accelerator gap transport difficult to model using, e.g., ELEGANT

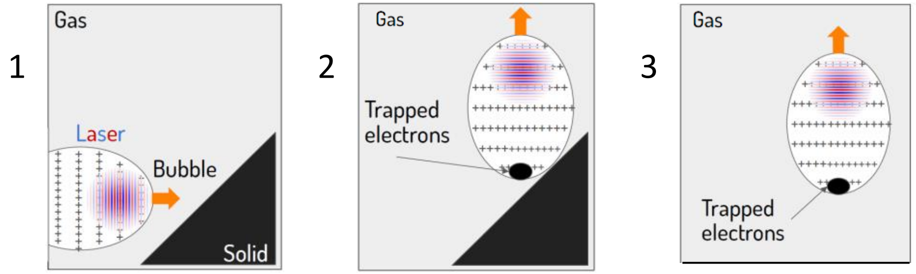
C. Lindstrom and E. Adli, Phys. Rev. Accel. Beams 19, 071002 (2016)
C. Lindstrom, EuroNNAc Special Topics Workshop 2022 (2022)
C. Lindstrom *et al* incl. A. Huebl & C. Mitchell, paper in preparation

See talks on HALHF at this workshop

BLAST codes also cover plasma-based collider modeling from source to interaction point



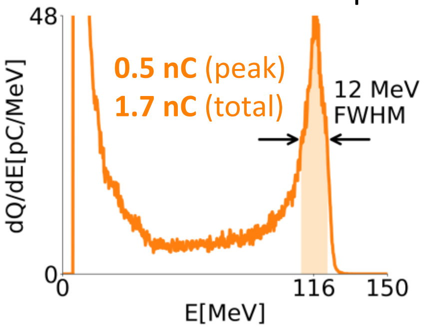
Two-stage injection+acceleration w/ plasma mirror

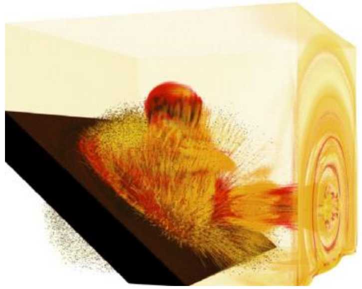


1st stage

Interaction with the plasma mirror

2nd stage

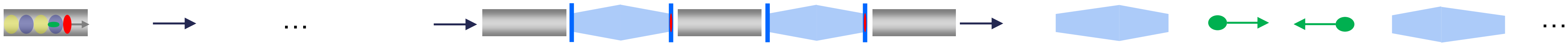




L Fedeli, A Huebl et al., SC22, **ACM Gordon Bell Prize for WarpX** (2022)

A success story of a multidisciplinary, multi-institutional team!

BLAST codes also cover plasma-based collider modeling from source to interaction point

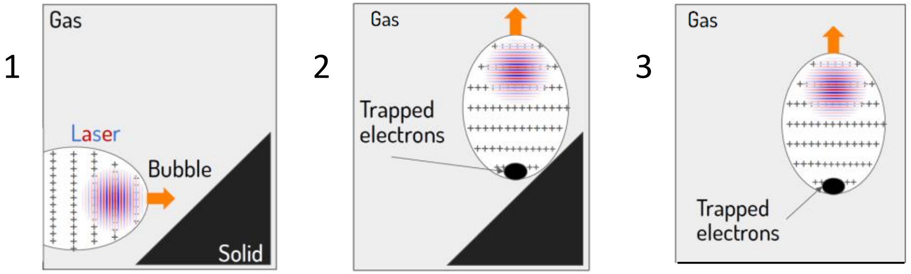


Generation

Acceleration & Transport

Interaction Point

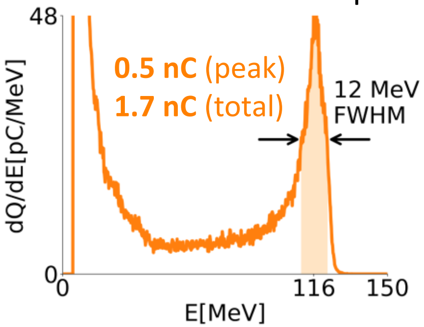
Two-stage injection+acceleration w/ plasma mirror

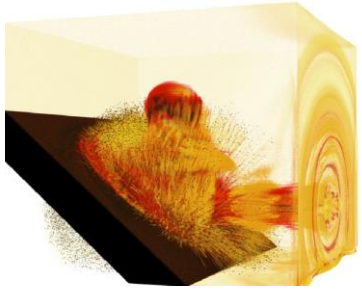


1st stage

Interaction with the plasma mirror

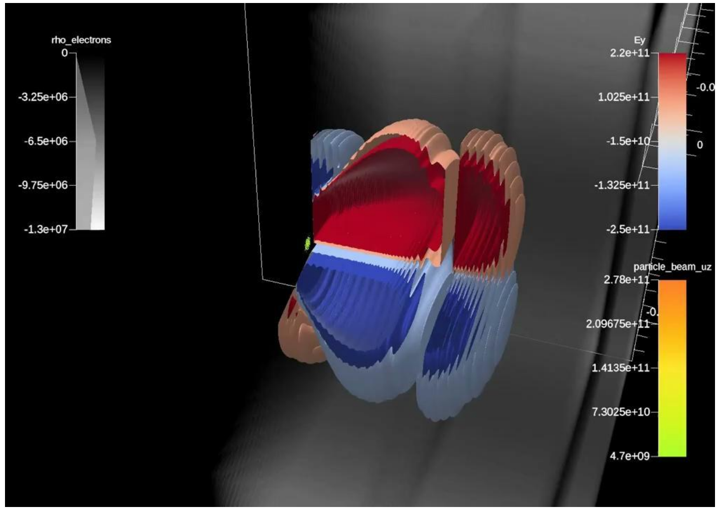
2nd stage

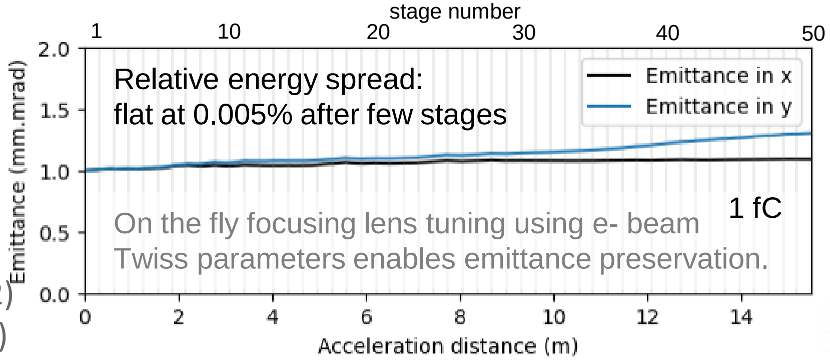




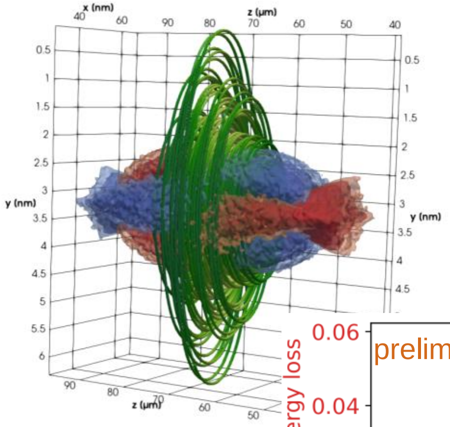
50 Multi-GeV LPA Stages in 3D

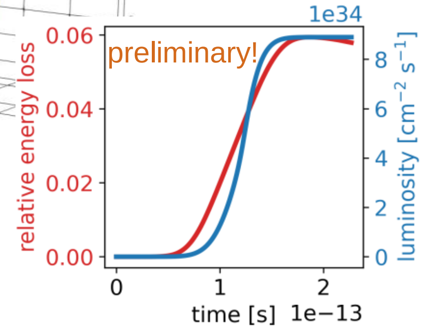
Ascent VTK-m *In Situ* Visualization of the first 15 stages:





10 TeV COM e^-e^+





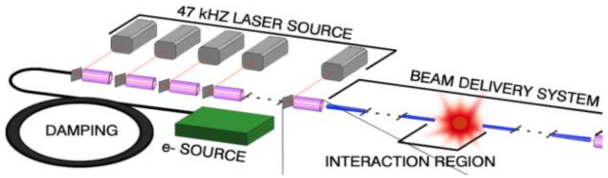
WarpX can now simulate flat, spherical, round and asymmetric beams in linear colliders:
ILC, C^3, wakefield, HALHF, …
Next are more QED physics &
circular colliders: FCC-ee, Muons

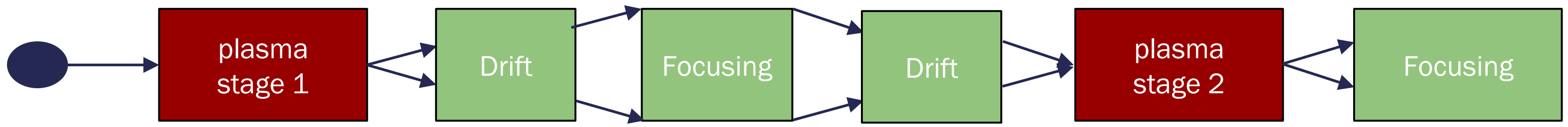
P5 Report: Exploring the Quantum Universe (2023)
J-L Vay et al., ISAV20 Keynote (2020) & PoP 28.2, 023105 (2021)
L Fedeli, A Huebl et al., SC22, ACM Gordon Bell Prize for WarpX (2022)
WarpX ECP MS FY23.1 & FY23.2 (2023); T Barklow et al., JINST (2023)
A Ferran Pousa et al., IPAC23, *TUPA093 & PRAB* (2023); CB Schroeder et al., JINST (2023)

See next talk by Arianna Formenti

We also explore the training of ML surrogate models to speed-up simulations

- In a given lattice, some elements can be more **computationally-expensive** to model.
  Extreme example: laser-plasma acceleration stages



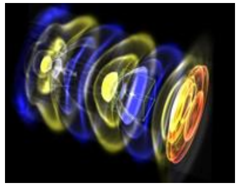


few pC
e- beam

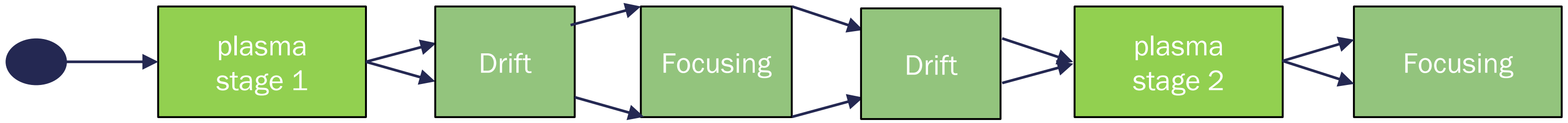
WarpX
(full EM-PIC)

ImpactX
(tracking code)

WarpX
(full EM-PIC)



- Under certain conditions (here: negligible collective effects, specific range of parameters), computationally-expensive elements can be replaced by **ML models**, trained over **past simulations.**



WarpX-trained
neural network
PyTorch

ImpactX
(tracking code)

WarpX-trained
neural network
PyTorch

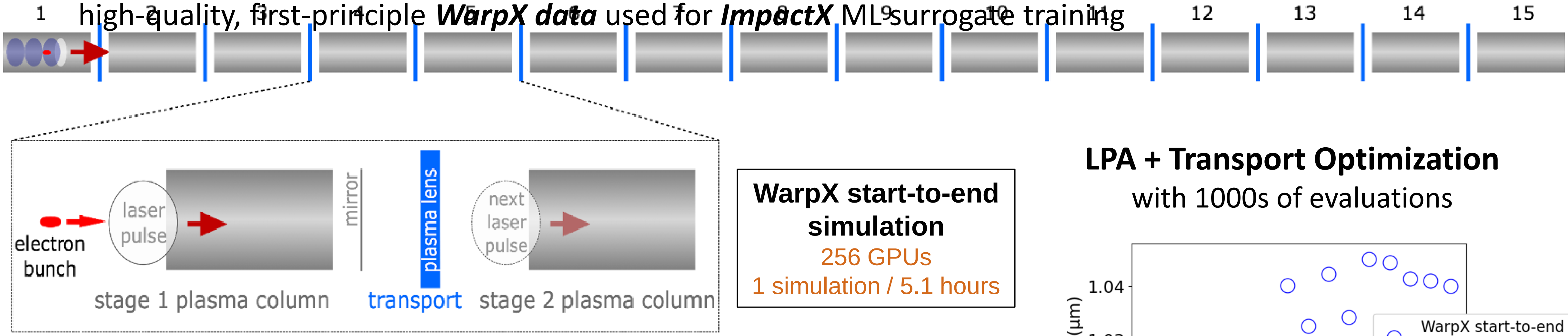
Simulation time: (with full geometry/physics)

hrs on several GPUs	<sec on 1 GPU

We Exploit our High-Quality HPC Data for ML-Boosted Collider Design

Central BLAST Code Interoperability: Combine Plasma & RF Accelerator Elements for start-to-end modeling

high-quality, first-principle ***WarpX data*** used for ***ImpactX*** ML surrogate training



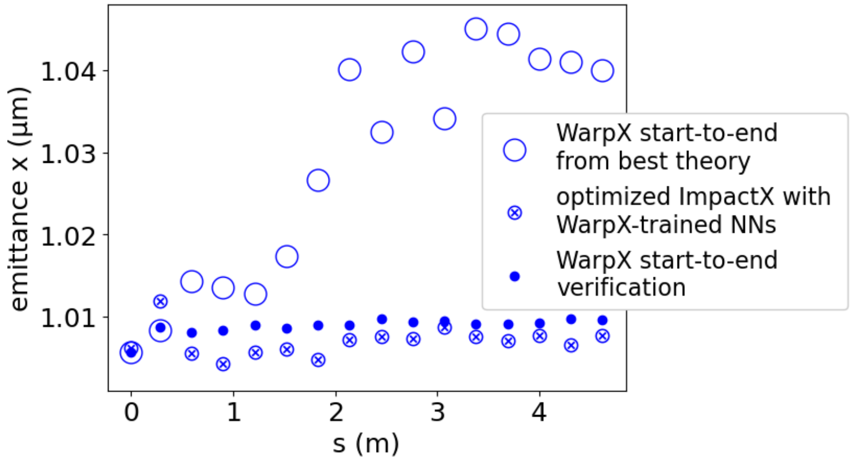
WarpX start-to-end simulation
256 GPUs
1 simulation / 5.1 hours

⇩

ImpactX with WarpX-trained NNs
1 GPU
2-4 simulations / sec

tightly-coupled LPA-*neural networks* inside **ImpactX**

LPA + Transport Optimization
with 1000s of evaluations



RT Sandberg et al., IPAC23, DOI:10.18429/JACoW-IPAC2023-WEPA101 (2023)
RT Sandberg et al., ***PASC24 Best Paper*** (2024)

Summary & Outlook

- Berkeley Lab has a long history in high performance modeling of particle accelerators

- We are leading the Beam, pLasma & Accelerator Simulation Toolkit (BLAST)
  - a coordinated effort that develops an integrated suite for start-to-end modeling of colliders
  - 80+ multidisciplinary team of contributors from labs, universities & private sector
  - leverages unique technology from the US DOE Exascale Computing Project that enables codes to be built on a common core that enables efficient simulations on CPUs and GPUs
  - common Python front-end enables user steering and direct coupling with optimization & AI/ML

- We are offering to use the new set of tools in support of the modeling of any type of future collider: Higgs factory, 10 TeV parton, muon, plasma-based, …

Questions?

We are fostering interoperability across open-source optimization software.

- Several **open-source optimization frameworks** are being used in the accelerator community (each with their respective strengths)

Xopt
github.com/ChristopherMayes/Xopt
Optimization algorithms

optimas
github.com/optimas-org/optimas
Optimization algorithms

rsopt
github.com/radiasoft/rsopt
Optimization algorithms

SLAC NATIONAL ACCELERATOR LABORATORY

DESY

BERKELEY LAB

Argonne NATIONAL LABORATORY

radiasoft

Argonne NATIONAL LABORATORY

powered by **libensemble**
github.com/Libensemble/libensemble

- Ongoing efforts by the developers to **standardize optimizers** and **foster interoperability.**

SciDAC Scientific Discovery through Advanced Computing

Even tighter ML integration can be achieved with differentiable simulations

Regular simulation code

Input: $\boldsymbol{X}$ → **Simulation code** → Output: $f(\boldsymbol{X})$

Differentiable simulation code

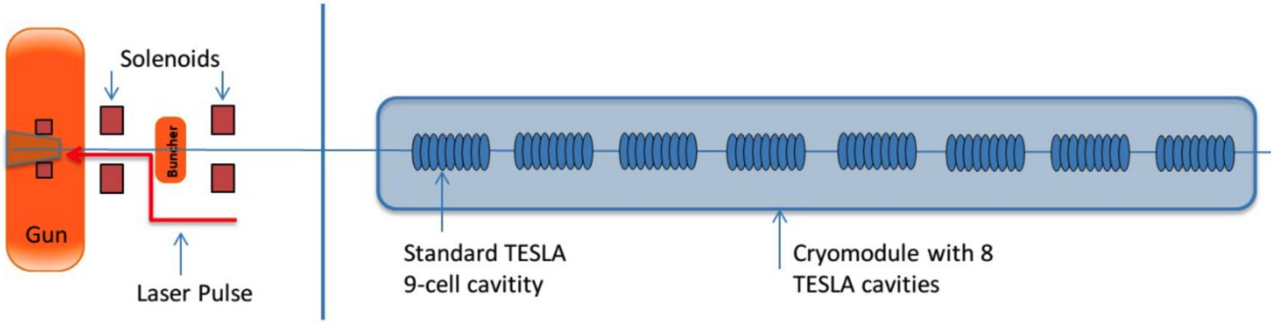
Input: $\boldsymbol{X}$ → **Simulation code** → Output: $f(\boldsymbol{X})$, $\frac{\partial f}{\partial \boldsymbol{X}}(\boldsymbol{X})$

Example:

$$f = \epsilon_{\perp}$$

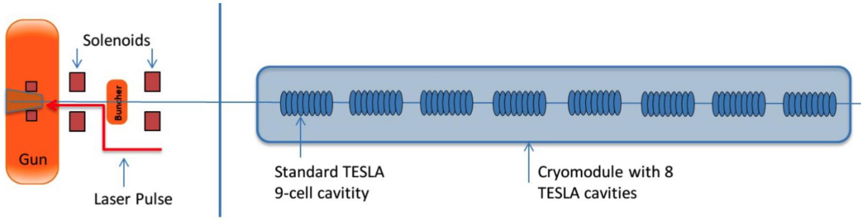
Input:
accelerator parameters

$$\boldsymbol{X} = \begin{pmatrix} B_{solenoid} \\ \varphi_{RF\ cavity} \\ E_{RF\ cavity} \\ \sigma_{beam,i} \end{pmatrix}$$



$$\frac{\partial f}{\partial \boldsymbol{X}} = \begin{pmatrix} \frac{\partial \epsilon_{\perp}}{\partial B_{solenoid}} \\ \frac{\partial \epsilon_{\perp}}{\partial \varphi_{RF\ cavity}} \\ \frac{\partial \epsilon_{\perp}}{\partial E_{RF\ cavity}} \\ \frac{\partial \epsilon_{\perp}}{\partial \sigma_{beam,i}} \end{pmatrix}$$

Differentiable codes have several advantages.

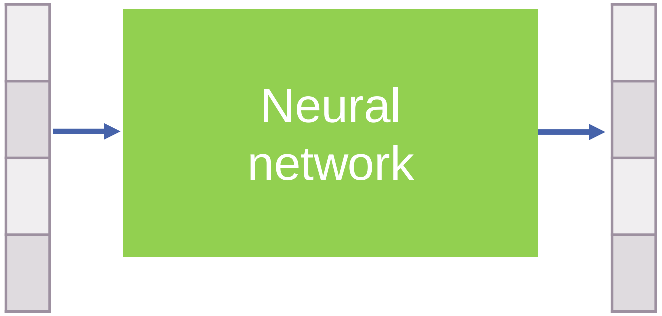


- **Sensitivity studies**
  $\frac{\partial f}{\partial X}$ quantifies how sensitive the output is to the input.

- **Optimization in high-dimensional space** (e.g. of accelerator designs)
  $\frac{\partial f}{\partial X}$ can be used in **gradient-based** optimizers, which often converge faster

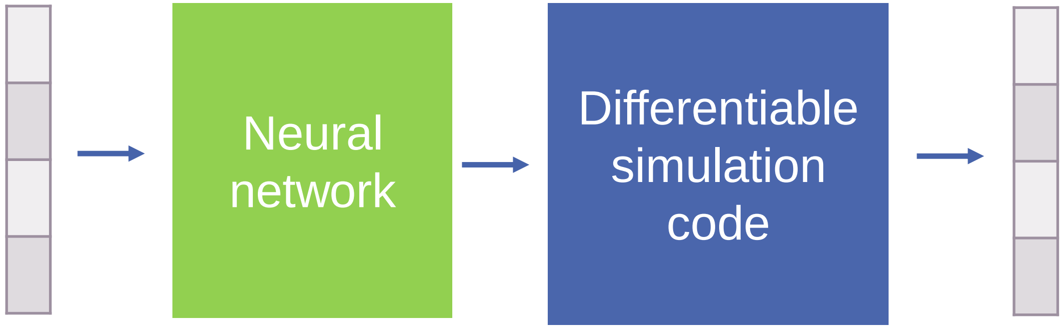
- **Allows training of a neural network that is combined with a differentiable code**

Traditional training of neural network



Input/output pairs, from a data set

Training of neutral network combined with a code



Example: *R. Roussel et al., Phase Space Reconstruction from Accelerator Beam Measurements Using Neural Networks and Differentiable Simulations, PRL (2023)*

We are exploring frameworks for differentiable codes.

- Several **algorithms** are available to make a code differentiable. e.g.
  J. Qiang, Differentiable self-consistent space-charge simulation for accelerator design, PRAB (2023)

- Several efforts to build differentiable **accelerator simulation codes**,
  based on **auto-differentiation frameworks.**
  - `pytorch`
    Cheetah: accelerator code based on `pytorch`
    github.com/desy-ml/cheetah

  - `Julia`
    Bmad-Julia: proposal to implement Bmad algorithms in Julia
    github.com/bmad-sim

  - Enzyme AD
    Takes existing C code and makes it auto-differentiable at compile time.
    Could be leveraged to make BLAST codes (ImpactX, WarpX, …) differentiable.