

Modeling & Optimizing Plasma Accelerators



Andreas Döpp^{1,2,3}

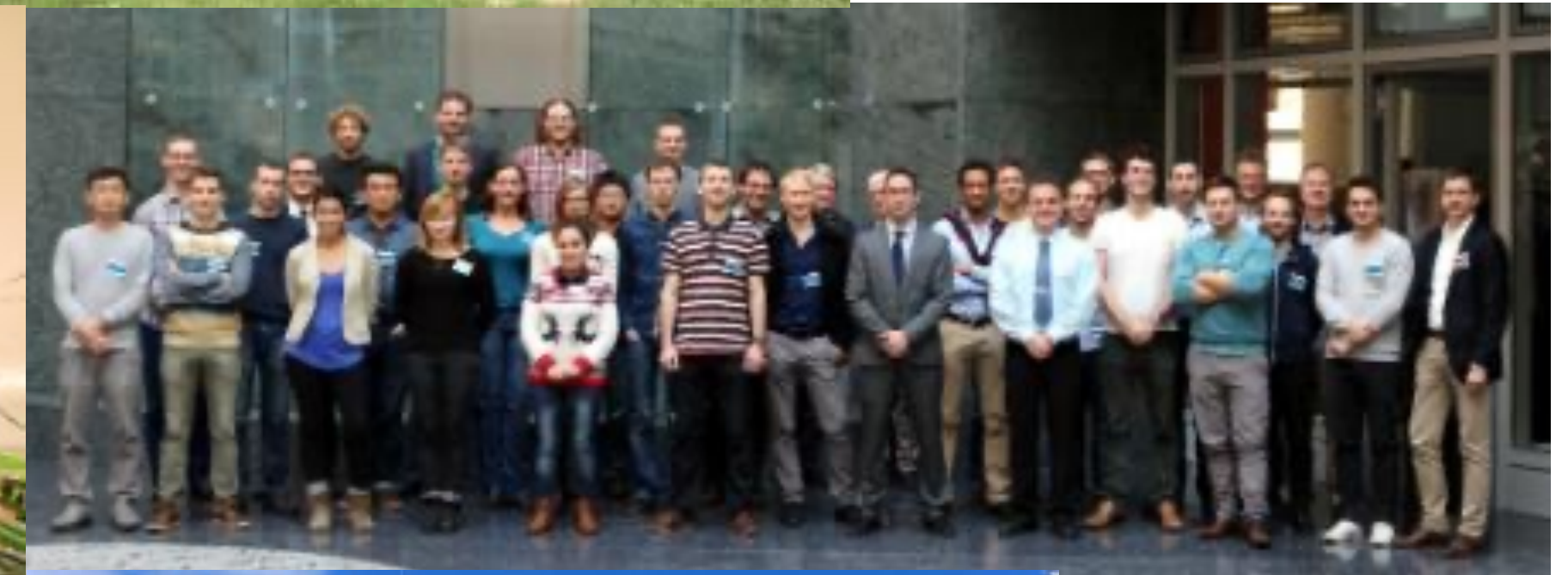
¹ Centre for Advanced Laser Applications, Faculty of Physics, LMU Munich

² Max Planck Institute for Quantum Optics (MPQ)

³ Faculty of Physics, Oxford University



It's a pleasure to be here!



I also used to be a Marie-Curie fellow ... almost a decade ago 😅

Team DOLPHIN

Data-driven Optimization of Laser Physics and Interactions



PhD students

Sunny

Jannik

Chris

Jakob

Collaborators

in CALA

Stefan Karsch, Jörg Schreiber

Faran, Nils, Jinpu

in Salamanca

Iñigo Sola,

Benjamin Alonso

in Oxford

Peter Norreys,

Robin Wang



Modeling & Optimizing

Plasma Accelerators



Modeling

- Analytical Models
- Particle-In-Cell Models
- Machine-learning Models

Optimization

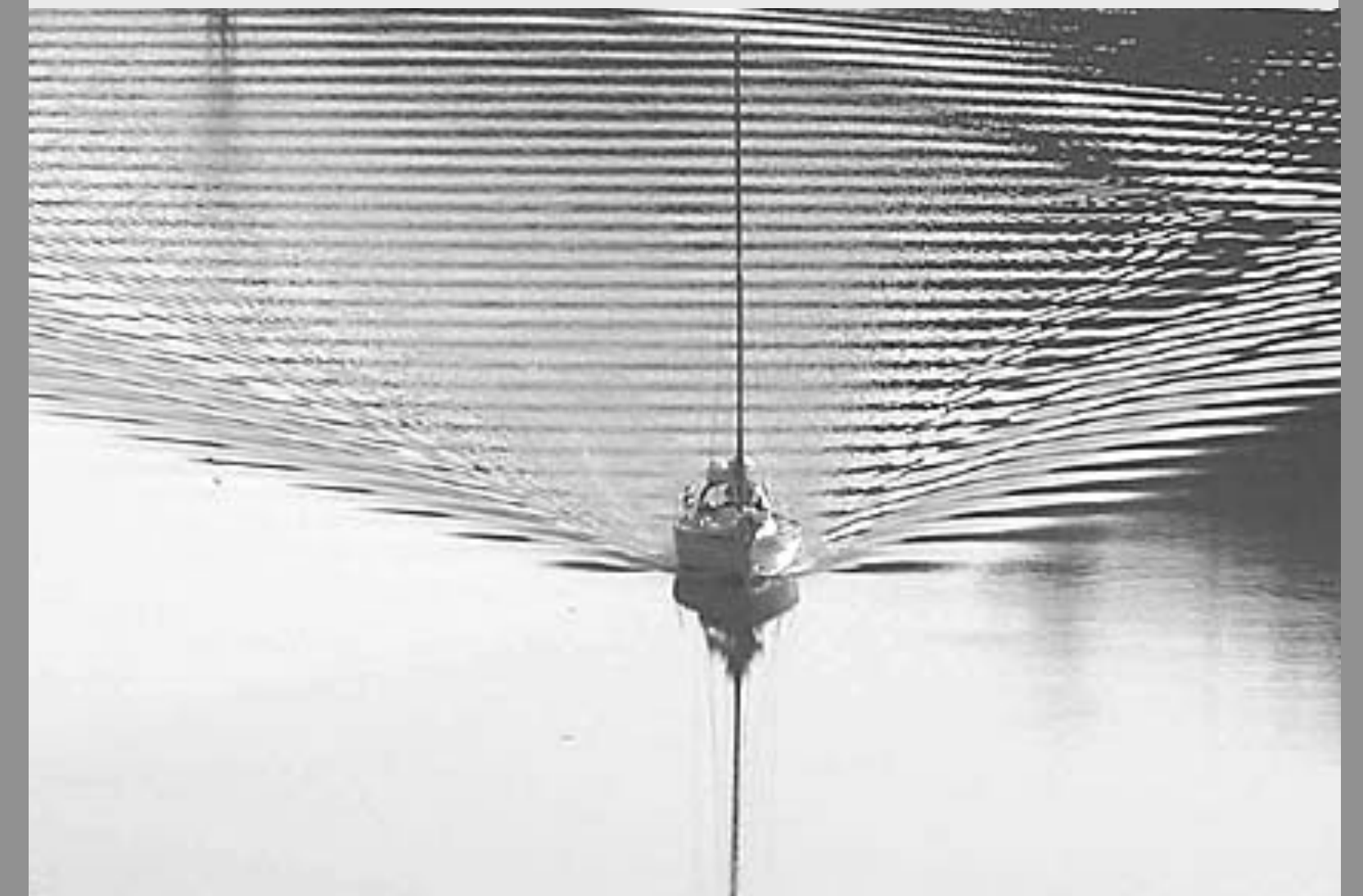
- Objective functions
- Multi-objective optimization

Modeling laser-plasma interaction

The linear wakefield regime

$$a_0 \sim E_0 [\text{TV/m}] / 4$$

$$\left(\frac{\partial^2}{\partial \xi^2} + k_p^2 \right) \phi = k_p^2 \frac{\langle a^2 \rangle}{2}$$



Plasma density n_e [cm^{-3}]

10^{20}
 10^{19}
 10^{18}
 10^{17}

λ_p
Longitudinal motion

Modeling laser-plasma interaction

The linear wakefield regime

Consider impulse response:

$$\left(\frac{\partial^2}{\partial \xi^2} + k_p^2 \right) \phi = \delta(\xi)$$



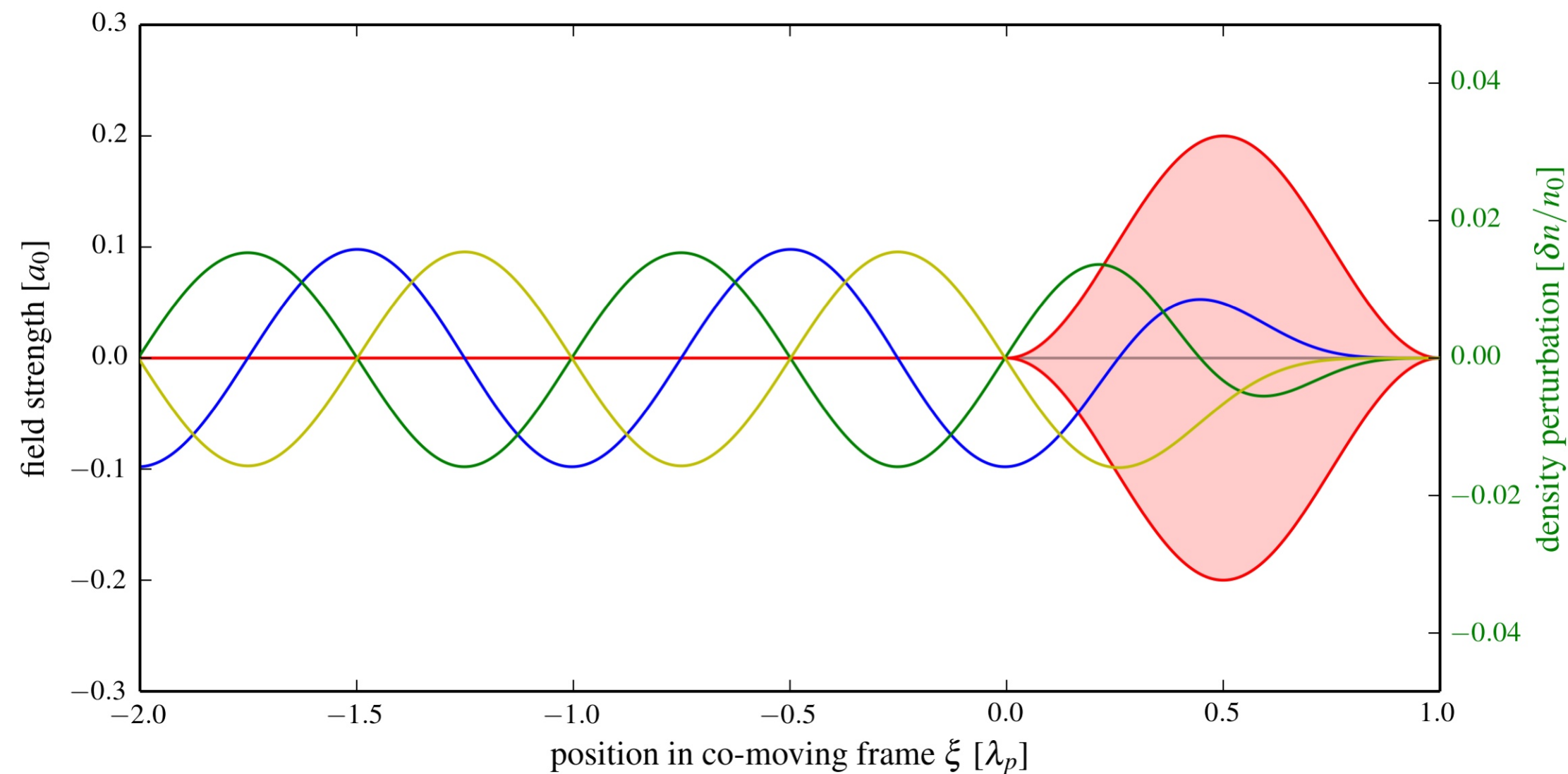
Find the Green's function

$$G(\xi) = \frac{1}{2} \sin[k_p(\xi - \xi')]$$



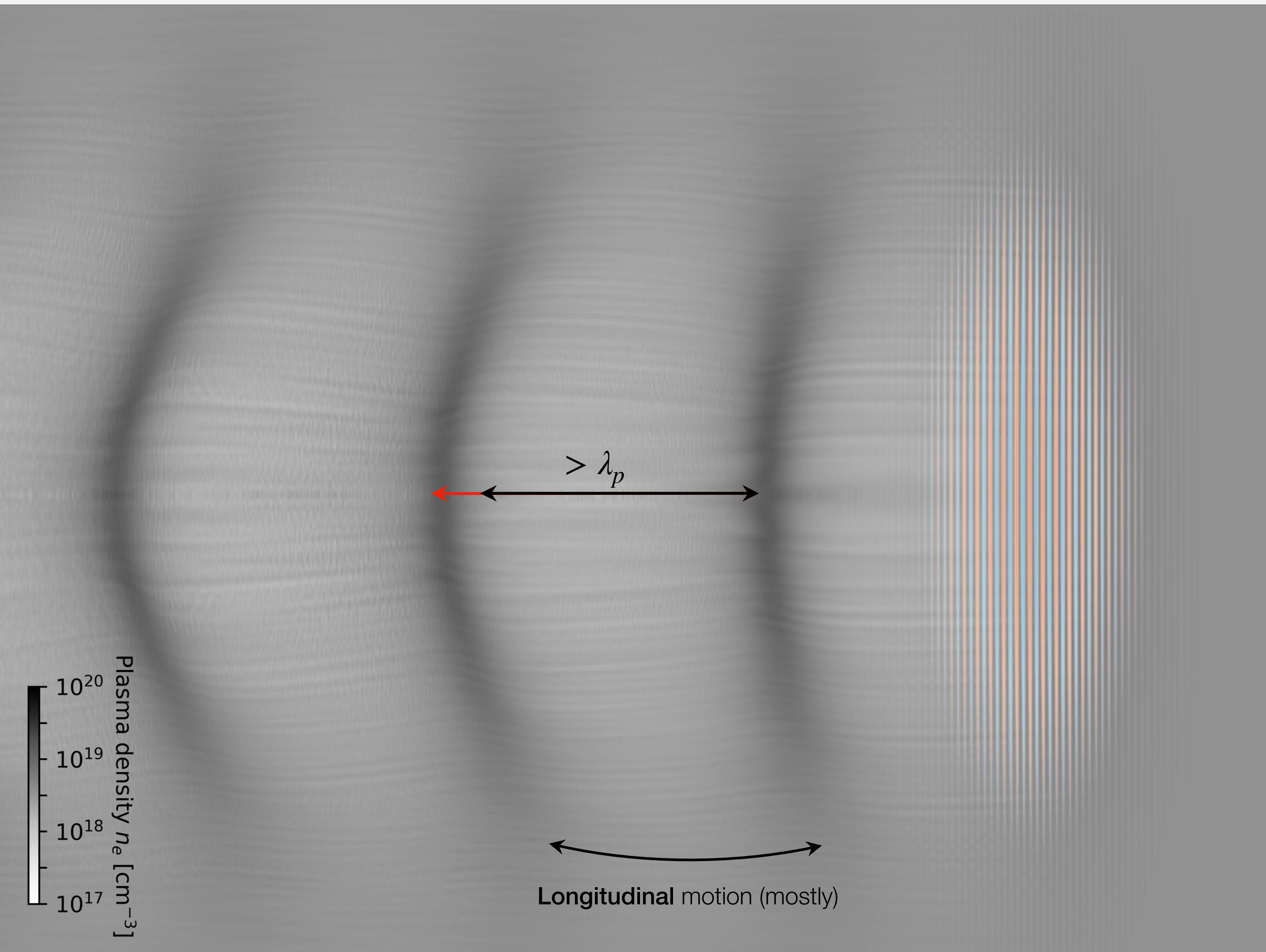
Convolve the actual excitation with the Green's function:

$$\phi = \int_{-\infty}^{\xi} f(\xi') G(\xi - \xi') d\xi'$$

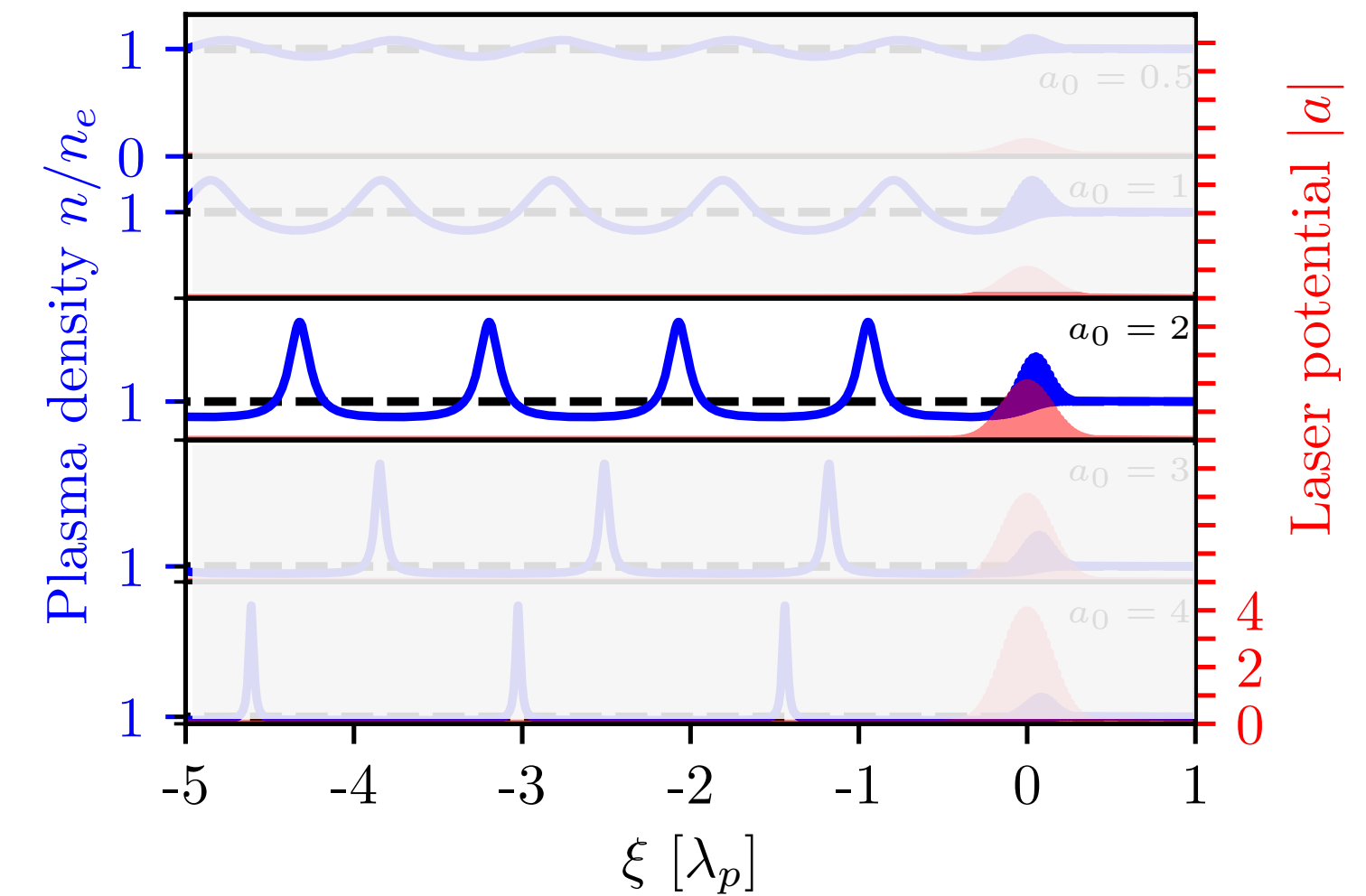


Modeling laser-plasma interaction

The non-linear wakefield regime

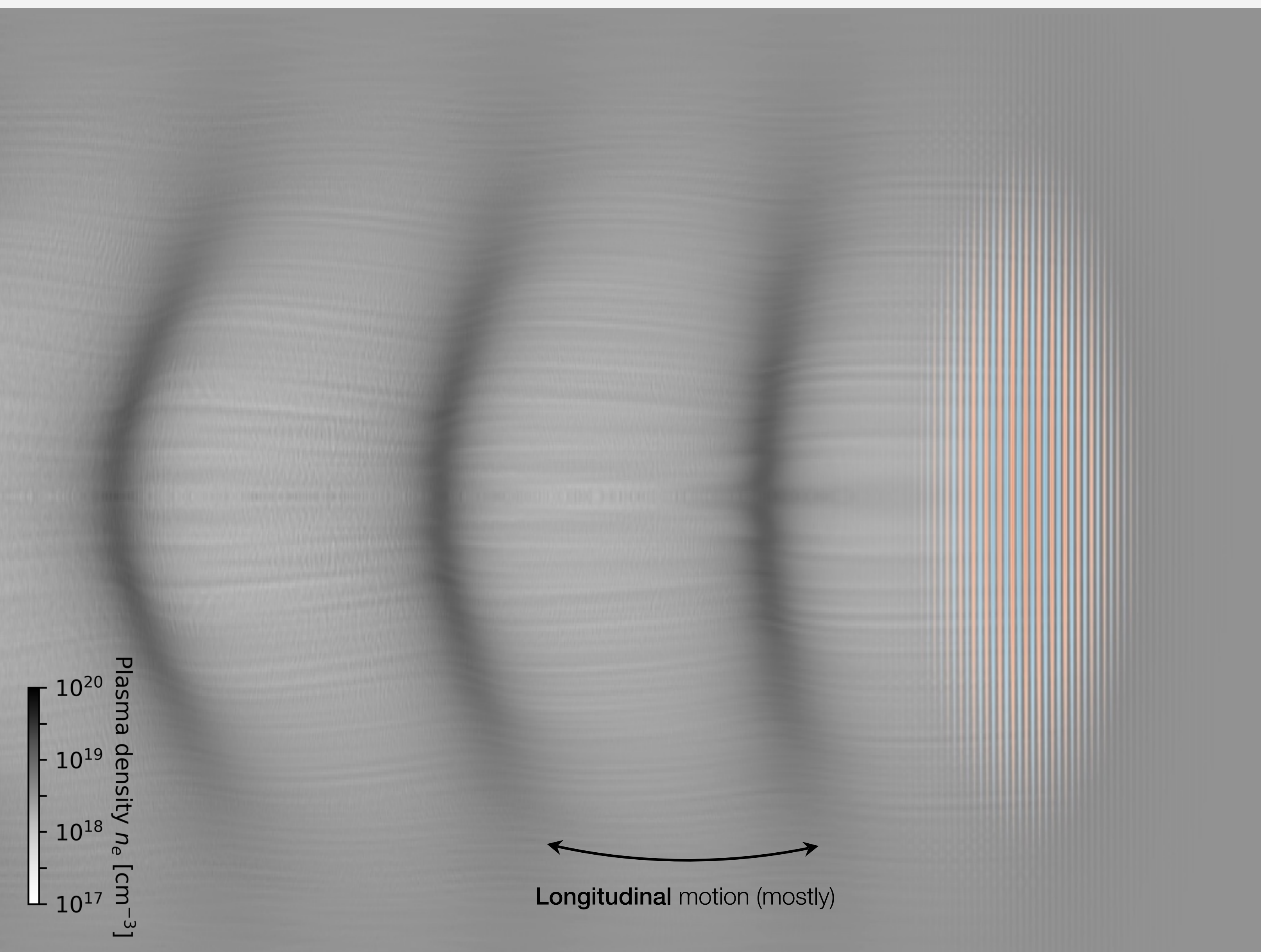


$$\frac{\partial^2}{\partial \xi^2} \Phi = \frac{1}{2} \left(\frac{1 + a^2}{(1 + \Phi)^2} - 1 \right) k_p^2$$



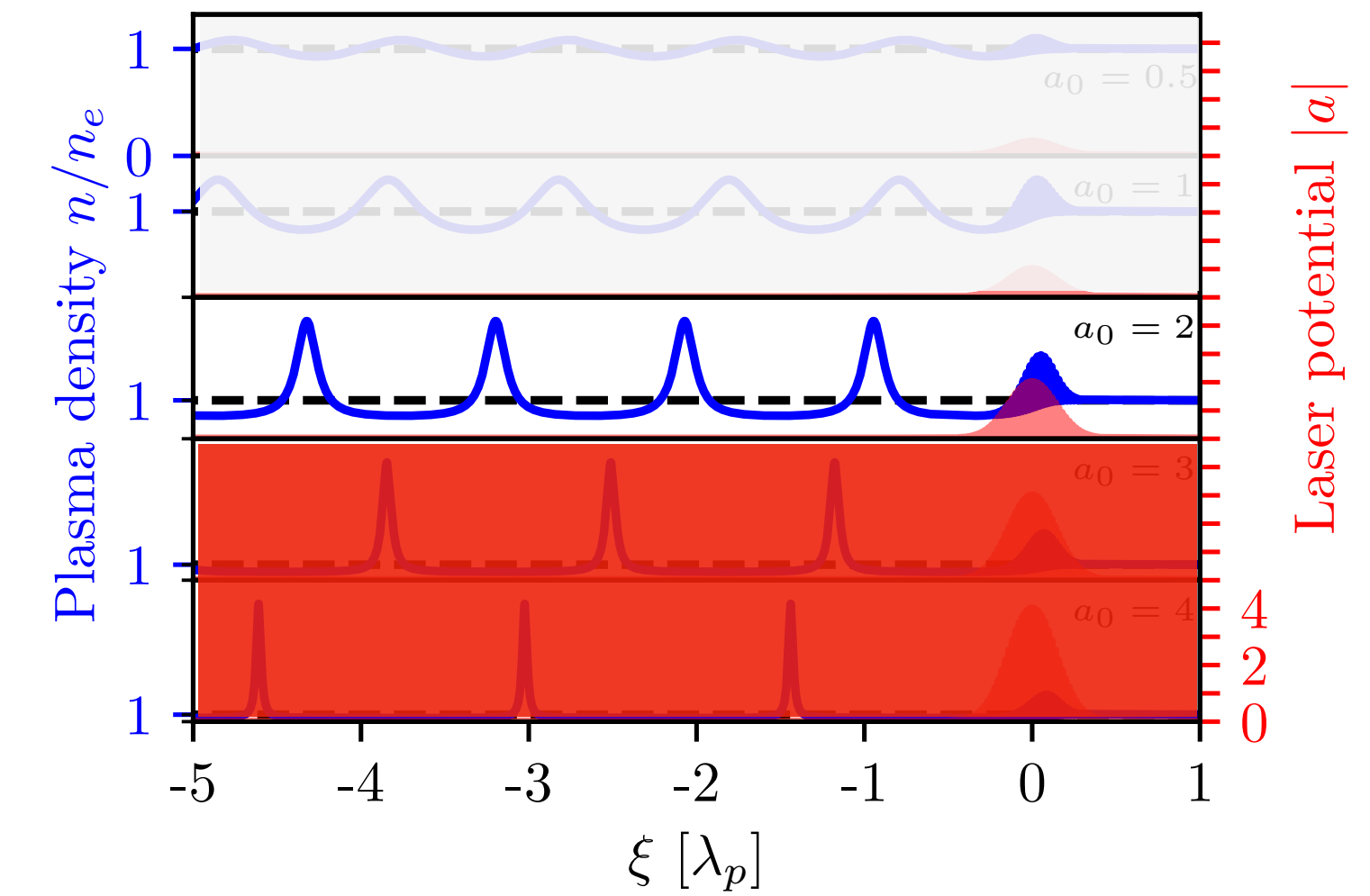
Modeling laser-plasma interaction

The non-linear wakefield regime



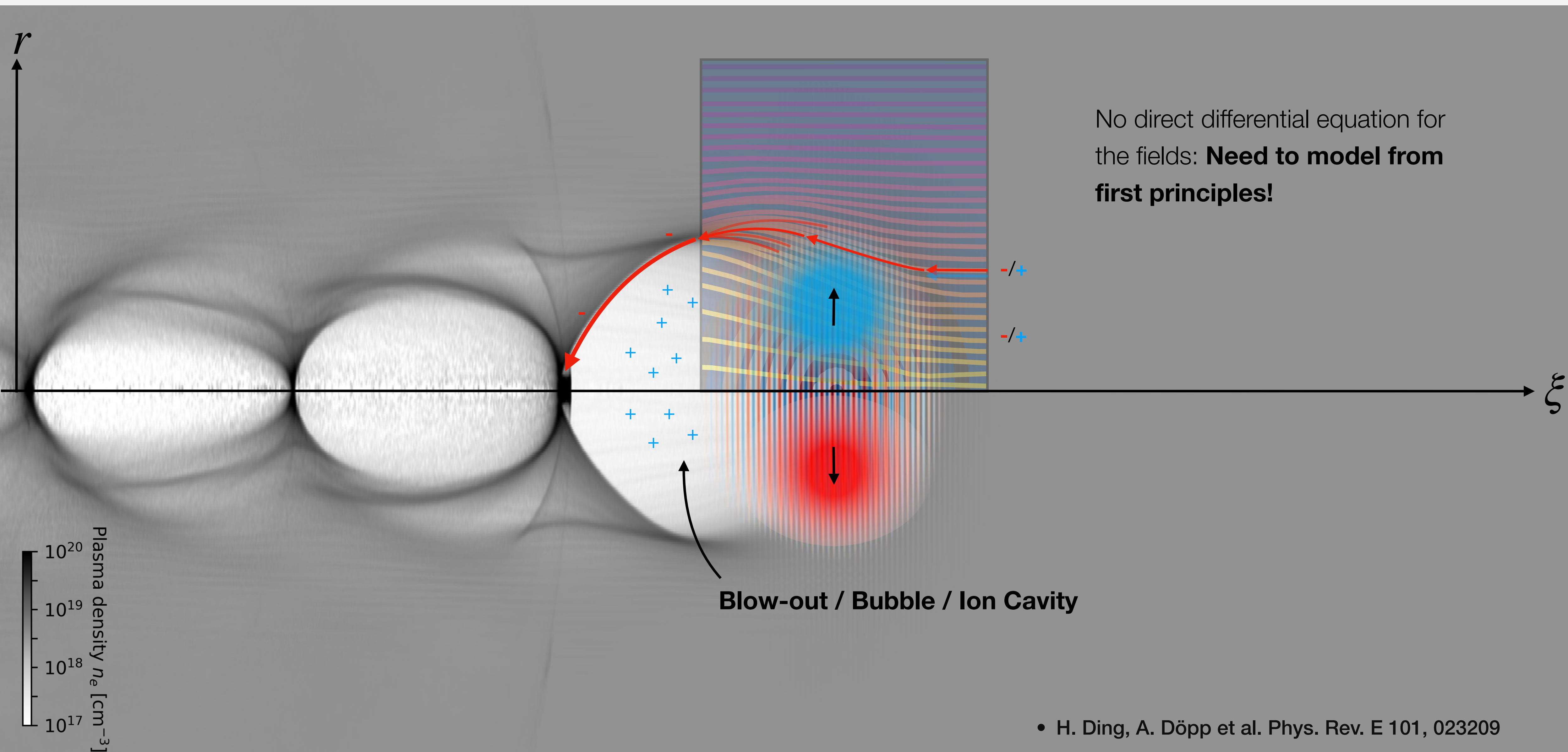
$$\frac{\partial^2}{\partial \xi^2} \Phi = \frac{1}{2} \left(\frac{1 + a^2}{(1 + \Phi)^2} - 1 \right) k_p^2$$

Approximation only valid if **mainly longitudinal motion!**
 This is the case for large focus when the transverse ponderomotive force is small.



Modeling laser-plasma interaction

The highly non-linear “blow-out” regime



Modeling laser-plasma interaction

The Vlasov equation

- We are interested in the evolution of a particle distribution in space, velocity and time $\rho(\vec{x}, \vec{v}, t)$
- In a collisionless system, where particles are neither created nor destroyed, the continuity equation is valid:

$$\frac{df}{dt} = \frac{\partial f}{\partial t} + \vec{v}_x \cdot (\nabla_x f) + \vec{v}_p \cdot \left(\frac{\vec{v}}{c} \nabla_p f \right) = 0$$

How does distribution change to movement of particles?

How does distribution change due to momentum changes (forces)?

Electrons

$$\frac{\partial f_e}{\partial t} + \vec{v}_e \cdot \nabla_x f_e - e \left(\vec{E} + \frac{\vec{v}_e}{c} \times \vec{B} \right) \cdot \nabla_p f_e = 0$$

Ions

$$\frac{\partial f_i}{\partial t} + \vec{v}_i \cdot \nabla_x f_i + Z_i e \left(\vec{E} + \frac{\vec{v}_i}{c} \times \vec{B} \right) \cdot \nabla_p f_i = 0$$

Modeling laser-plasma interaction

The Maxwell equations



- At the same time we need to fulfill the Maxwell equations

$$\nabla \times \vec{B} = \frac{4\pi}{c} \vec{j} + \frac{1}{c} \frac{\partial \vec{E}}{\partial t}$$

$$\nabla \times \vec{E} = -\frac{1}{c} \frac{\partial \vec{B}}{\partial t}$$

$$\nabla \cdot \vec{B} = 0$$

$$\nabla \cdot \vec{E} = 4\pi\rho$$

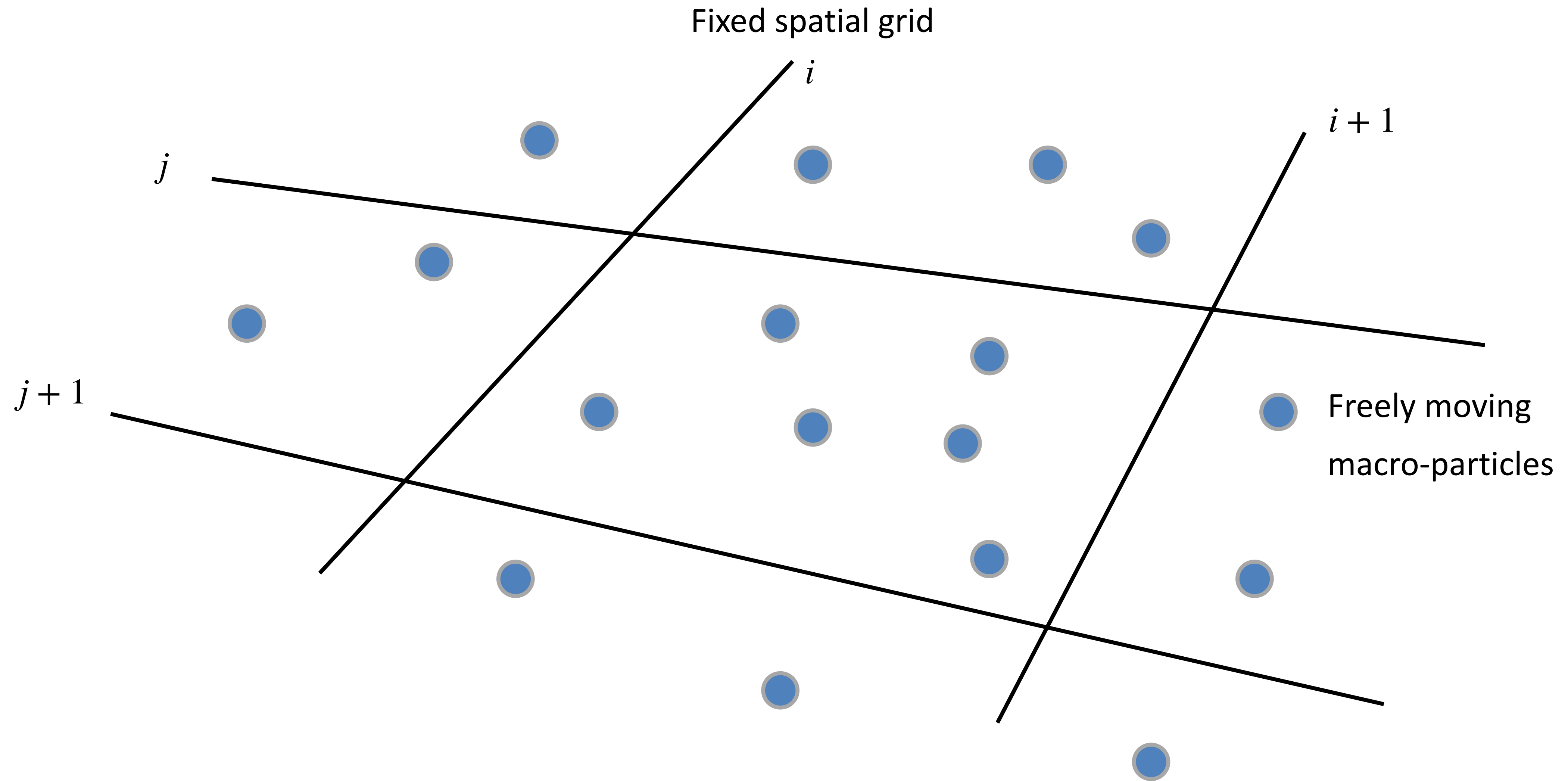
- With the charge density ρ and current \vec{j} defined as

$$\rho = e \int (Z_i f_i - f_e) d^3p$$

$$\vec{j} = e \int (Z_i \vec{v}_i f_i - \vec{v}_e f_e) d^3p$$

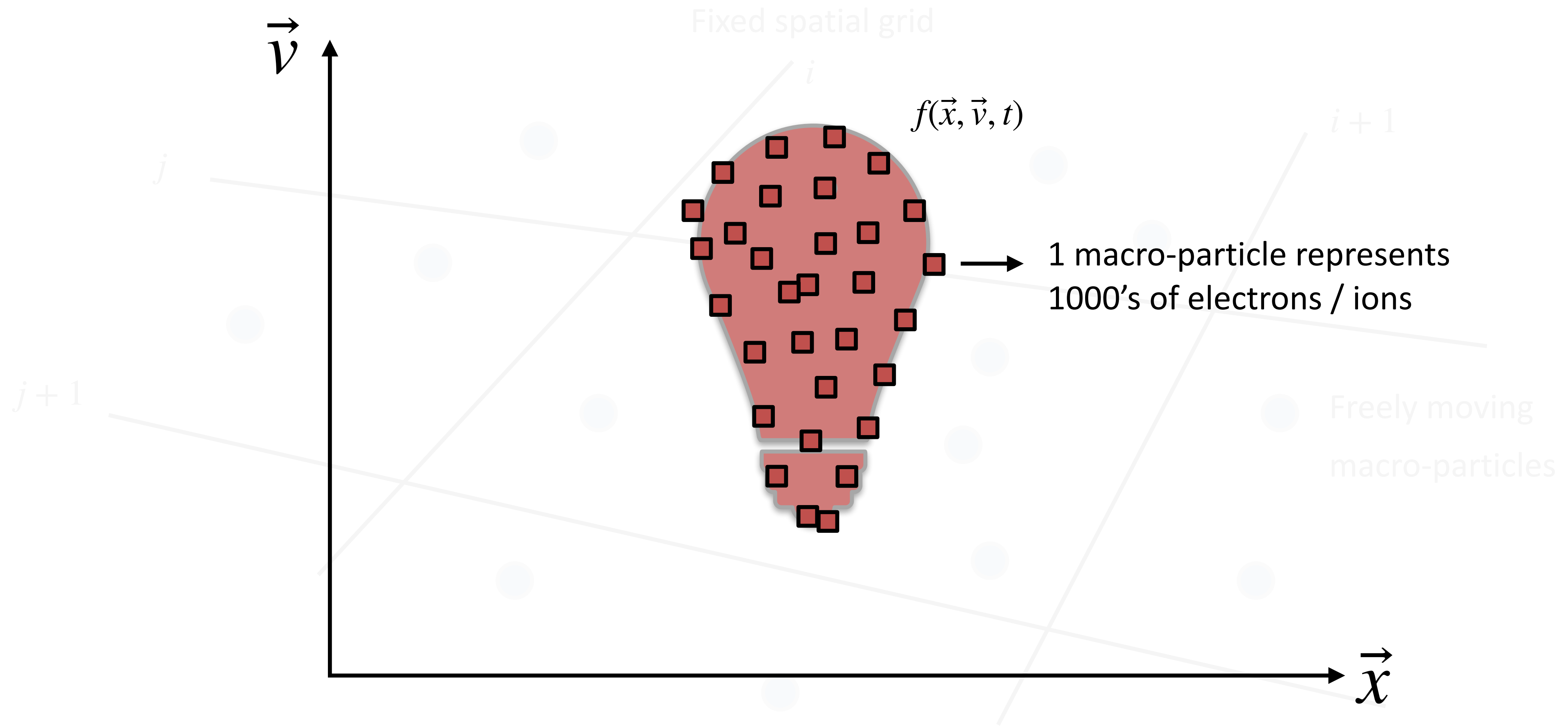
Modeling laser-plasma interaction

The Particle-In-Cell Method



Modeling laser-plasma interaction

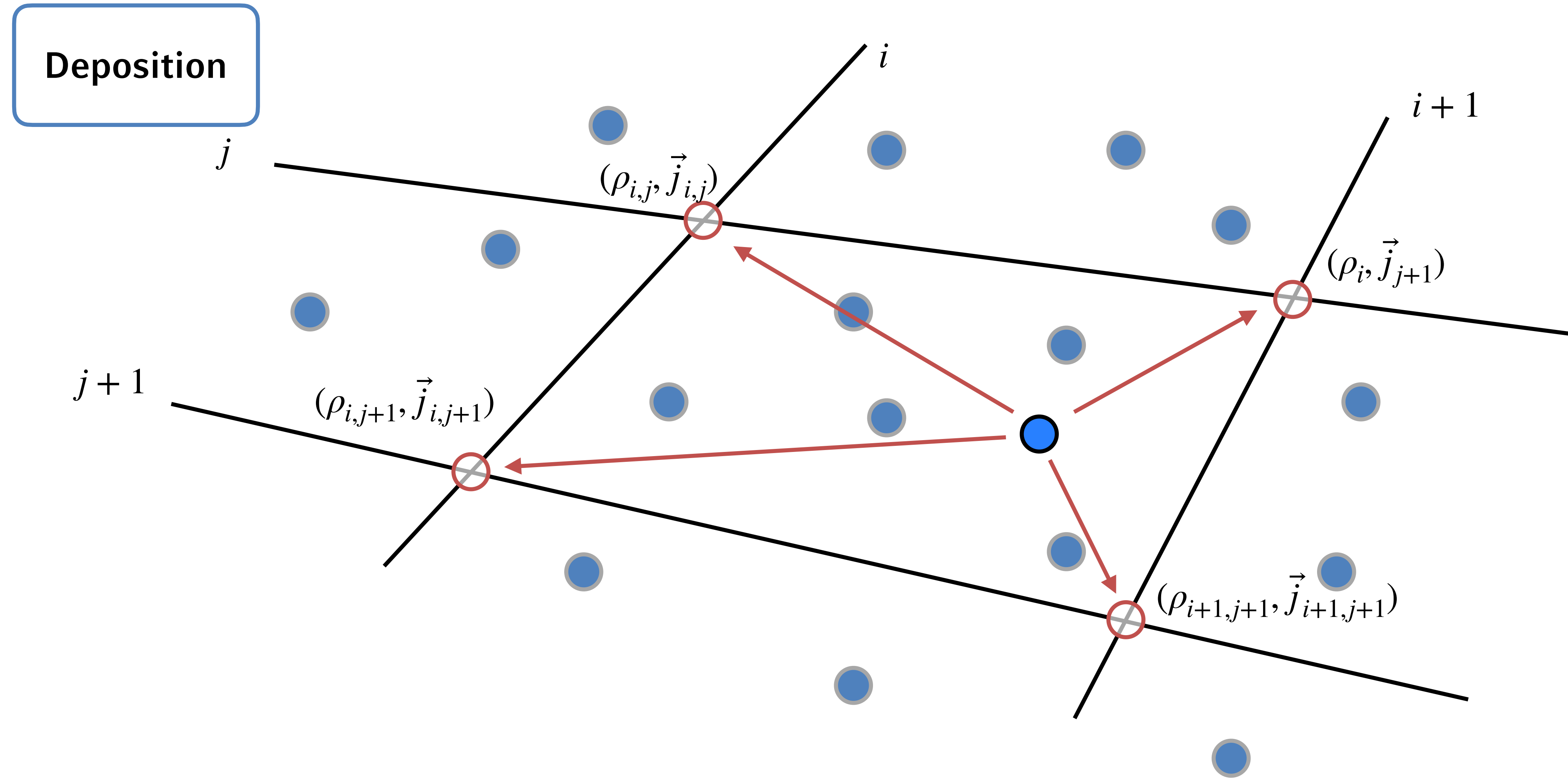
The Particle-In-Cell Method



The PIC assumption is basically that particles initially close to another stay close to another

Modeling laser-plasma interaction

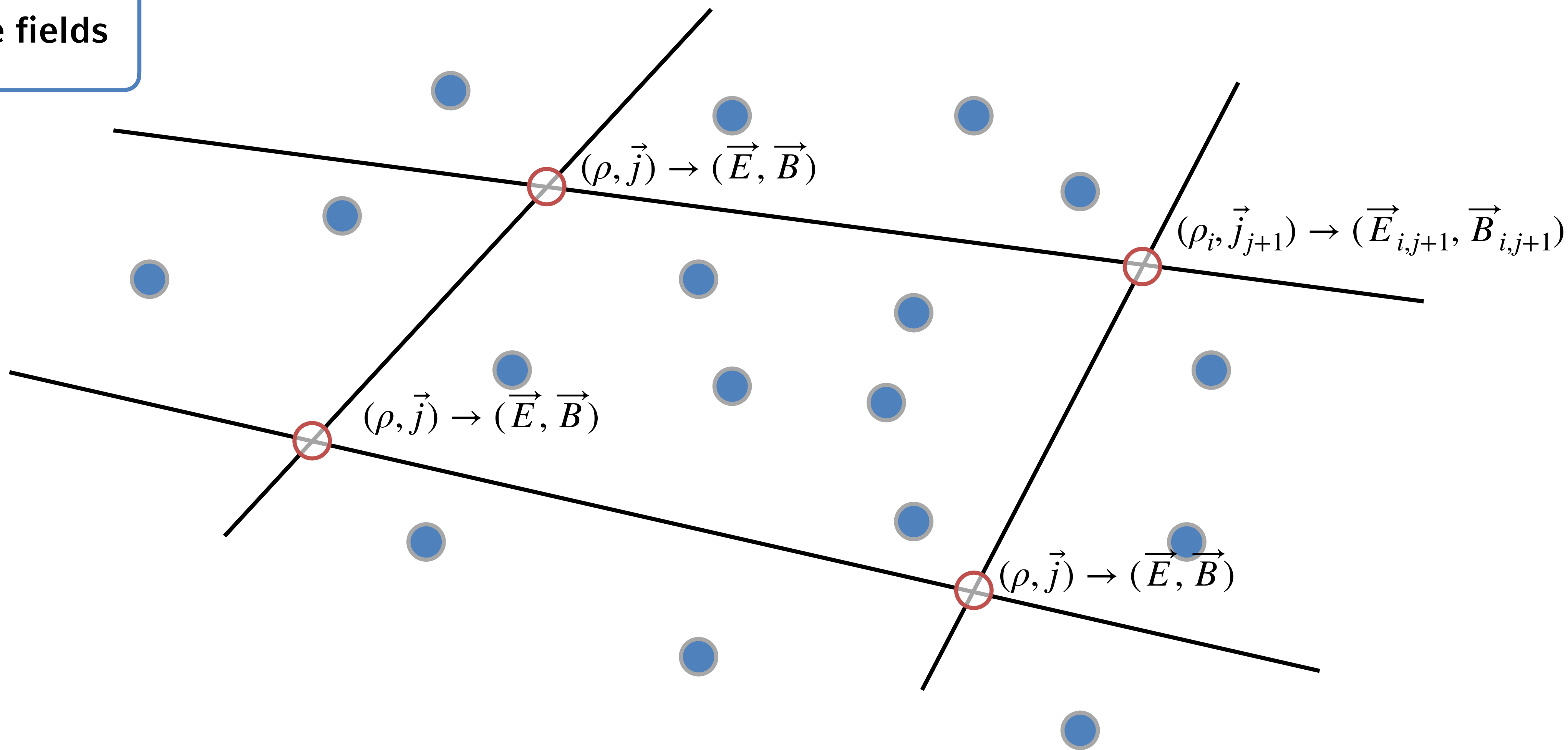
The Particle-In-Cell Method



Modeling laser-plasma interaction

The Particle-In-Cell Method

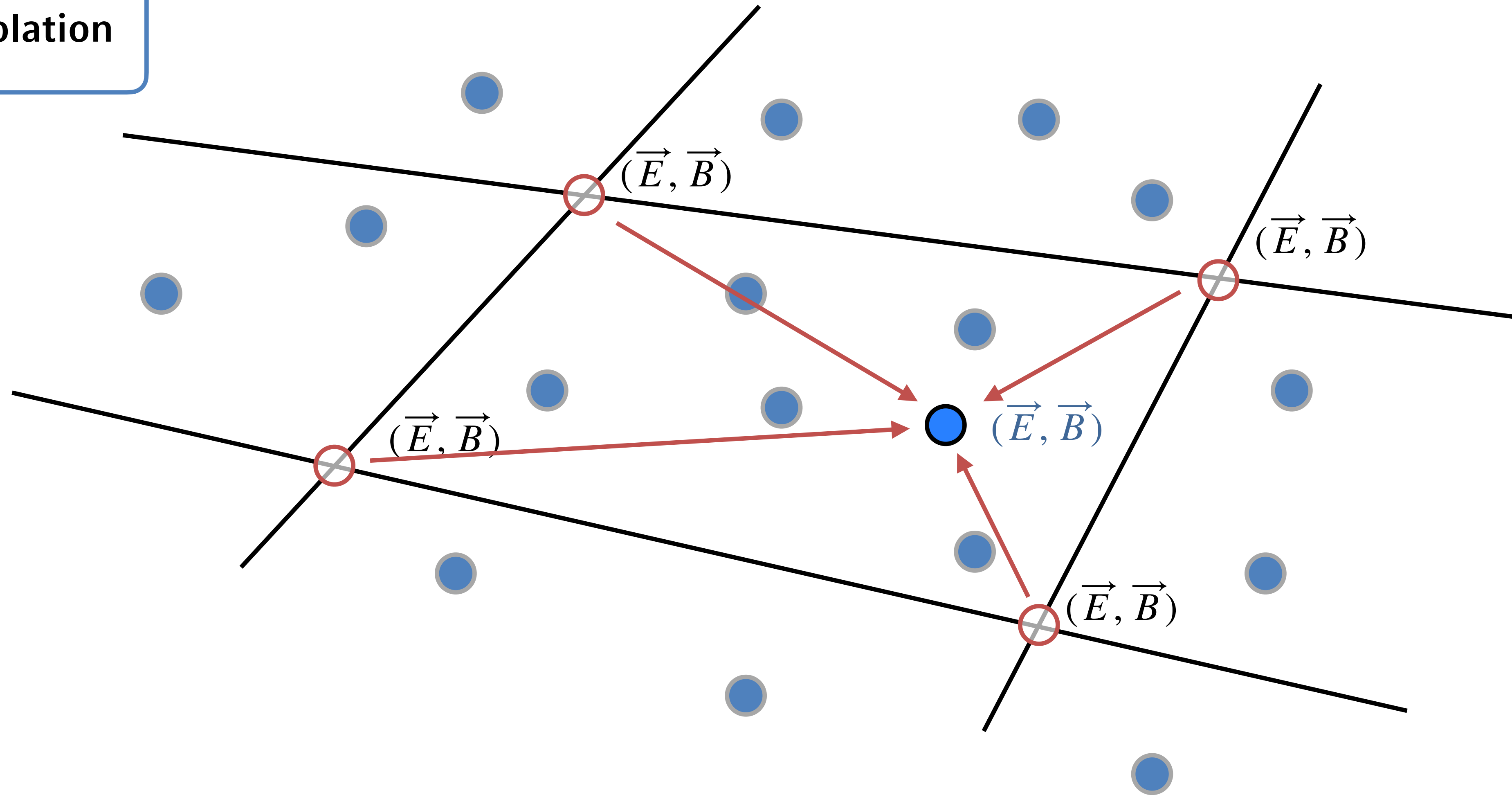
Update fields



Modeling laser-plasma interaction

The Particle-In-Cell Method

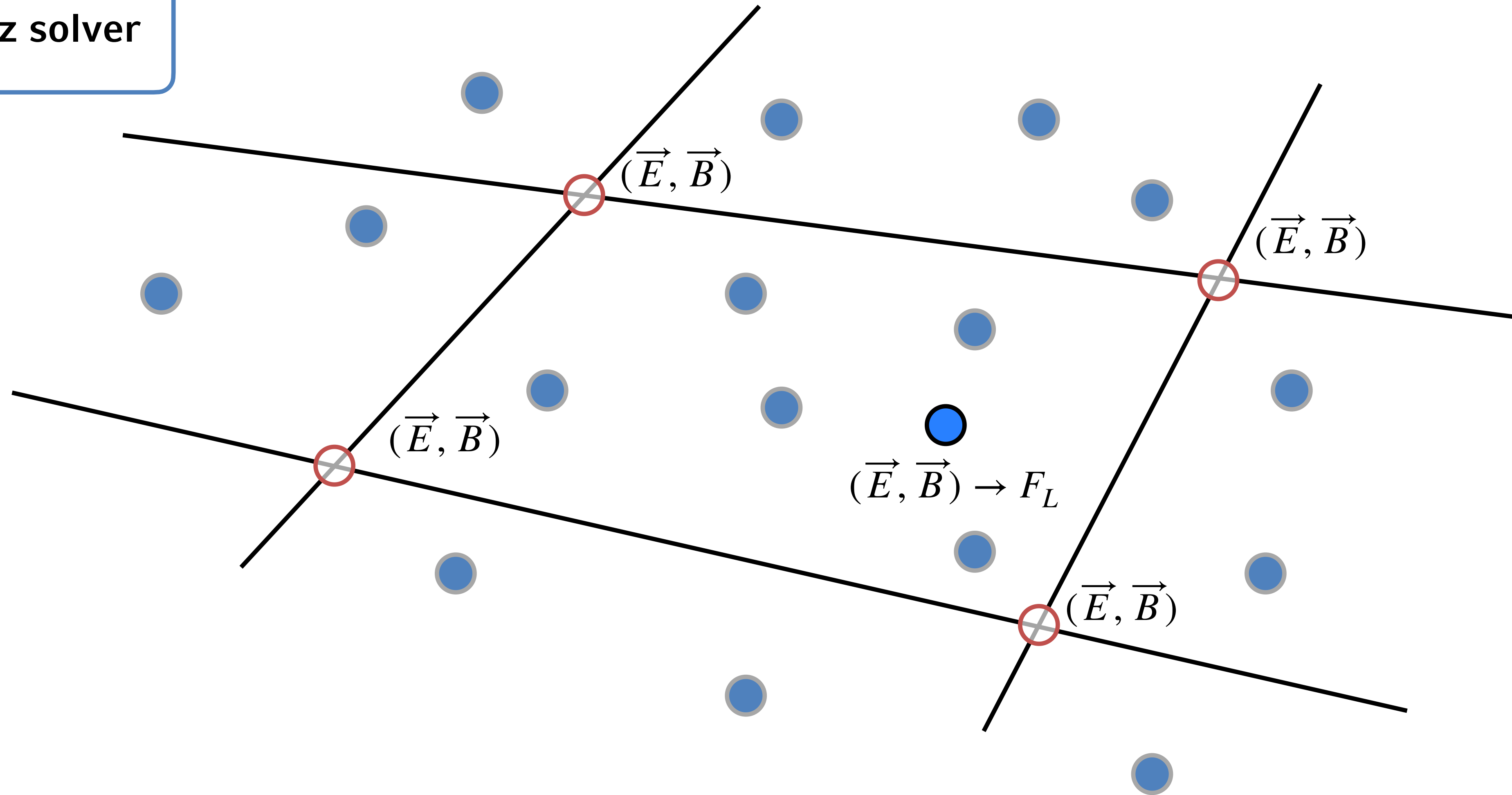
Interpolation



Modeling laser-plasma interaction

The Particle-In-Cell Method

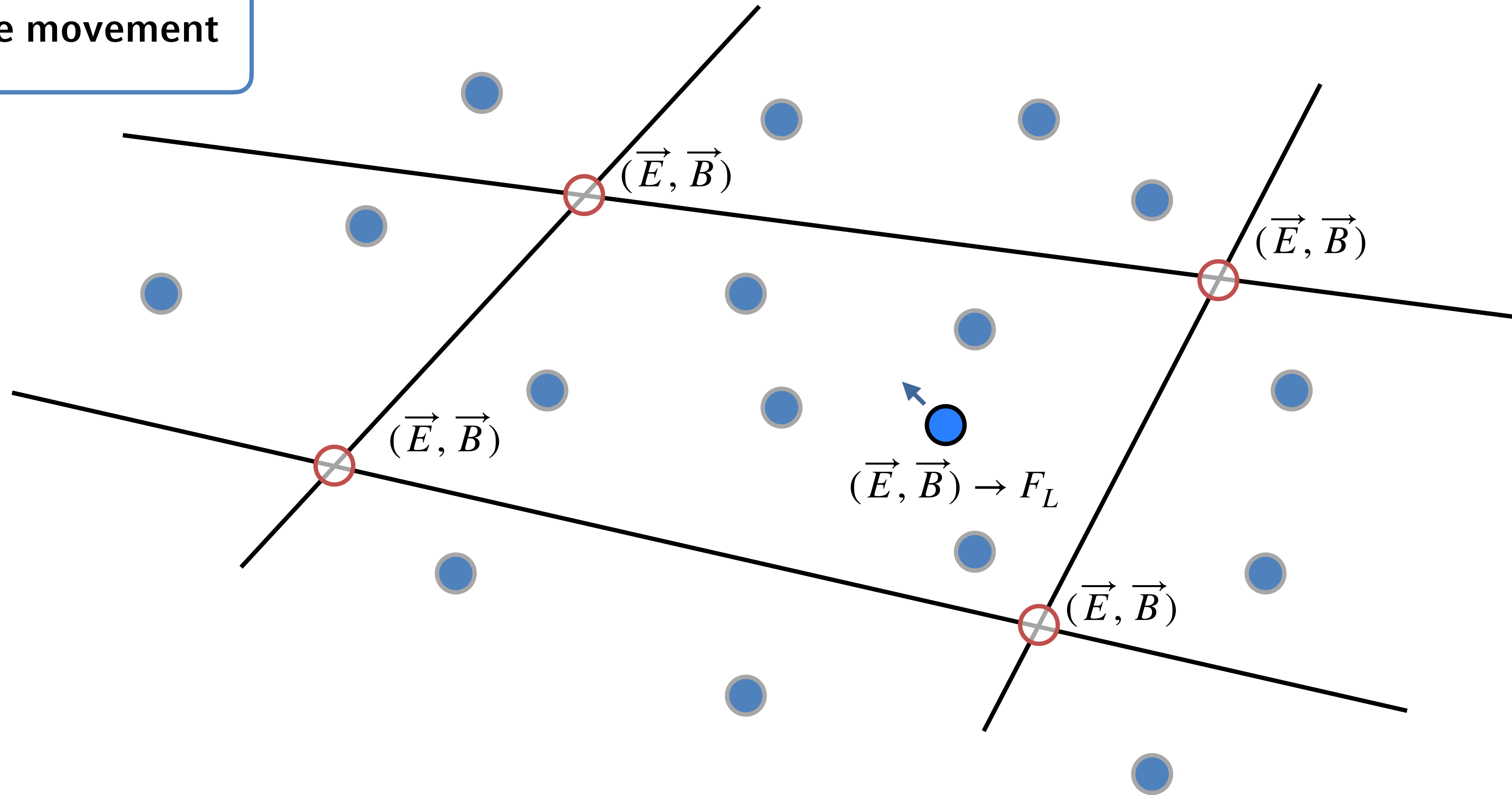
Lorentz solver



Modeling laser-plasma interaction

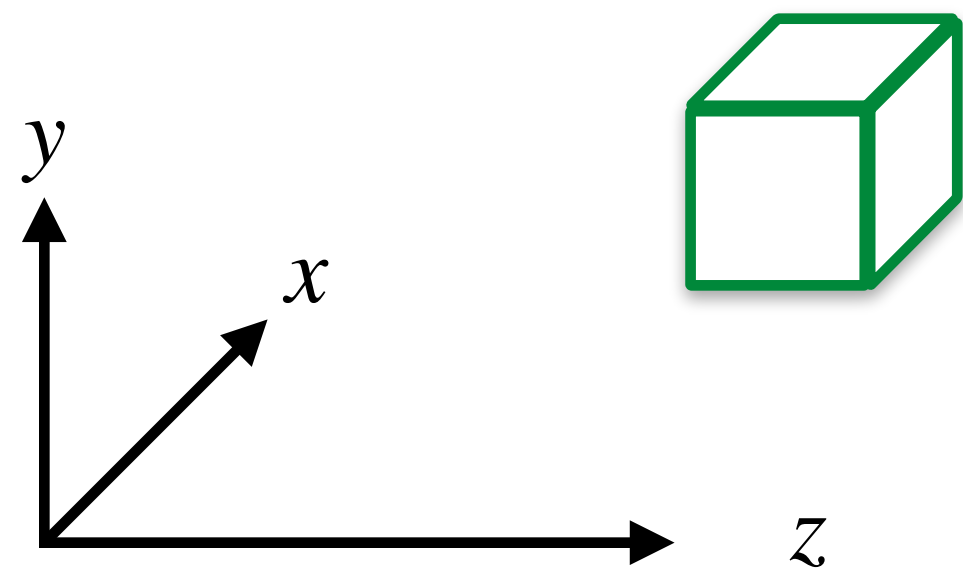
The Particle-In-Cell Method

Particle movement

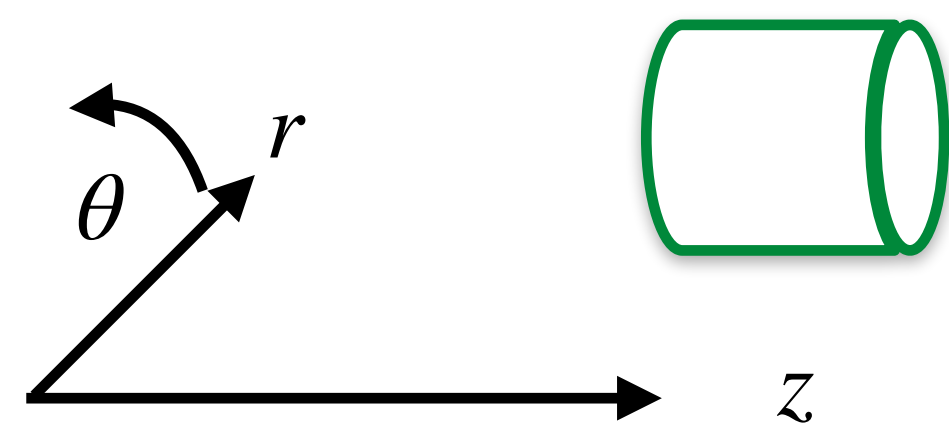
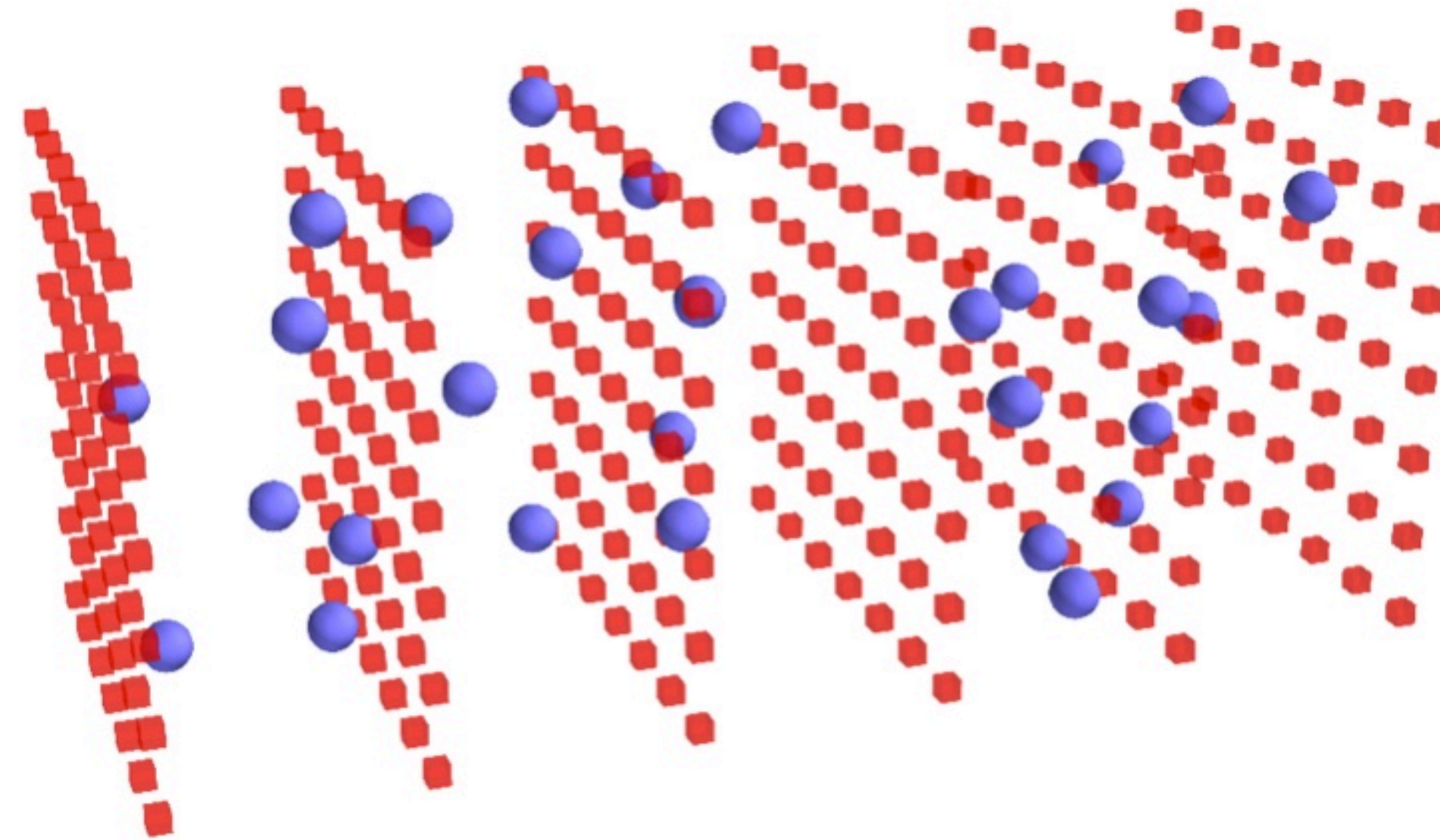


Modeling laser-plasma interaction

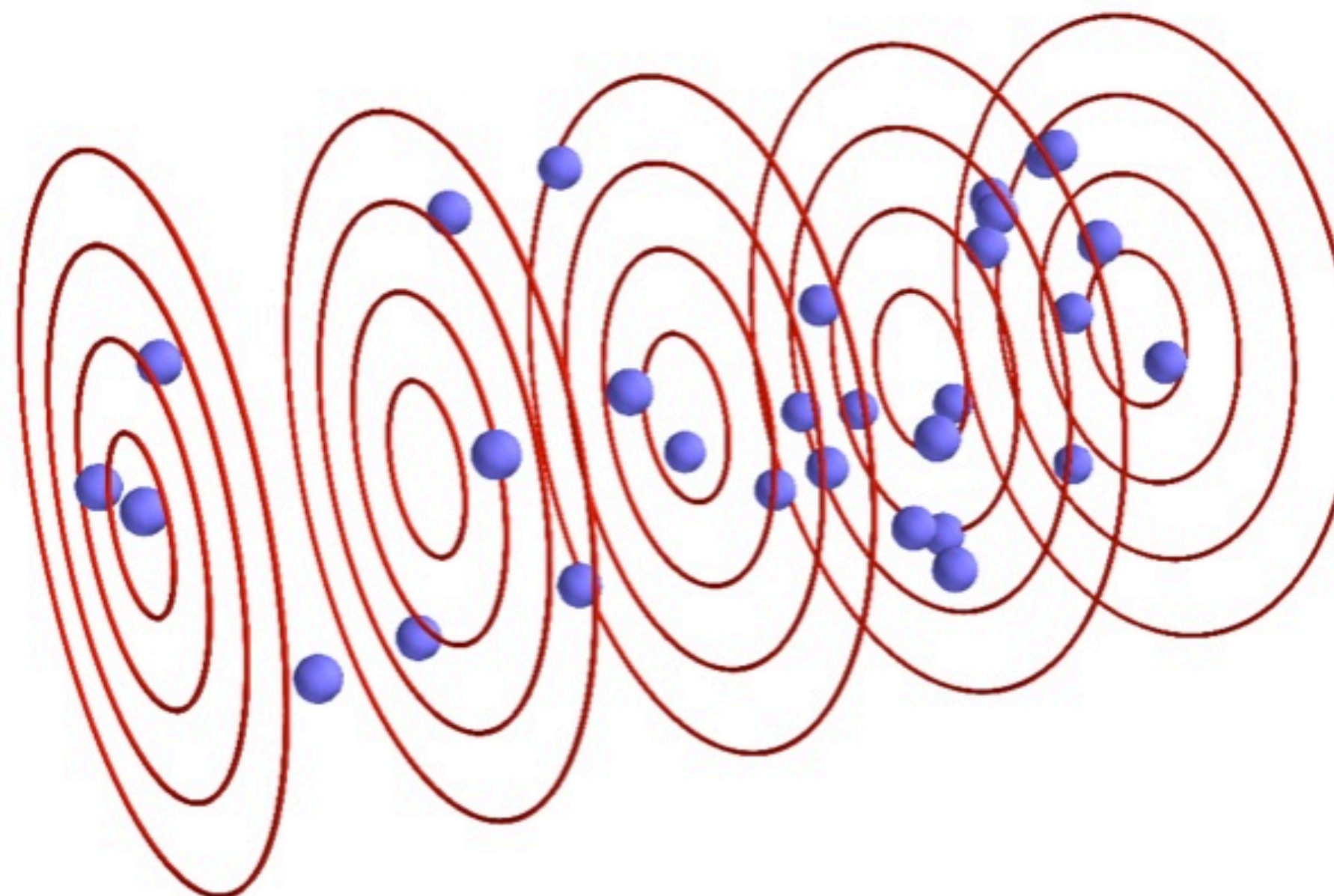
The Particle-In-Cell Method | Geometries



Cartesian



Cylindrical



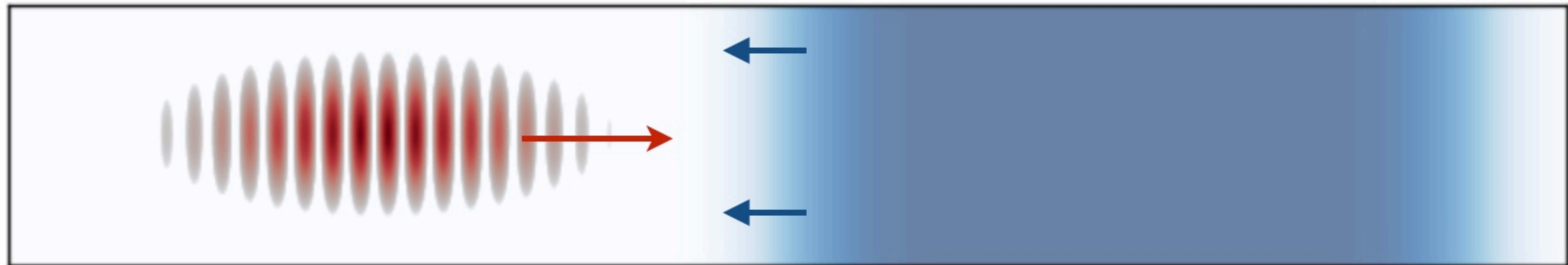
Modeling laser-plasma interaction

The Particle-In-Cell Method | Reference frames

Laboratory frame (can be co-moving)



Boosted frame



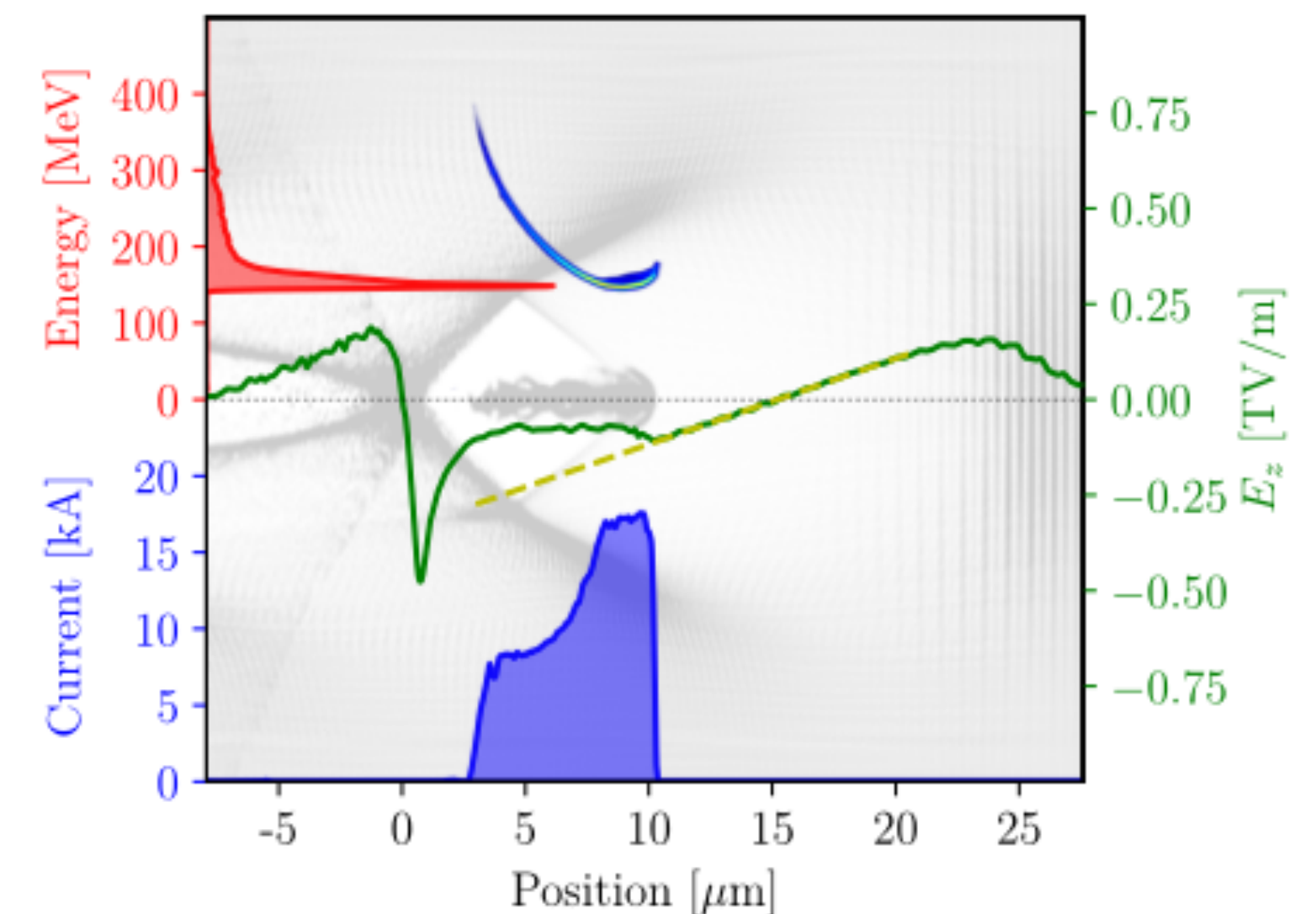
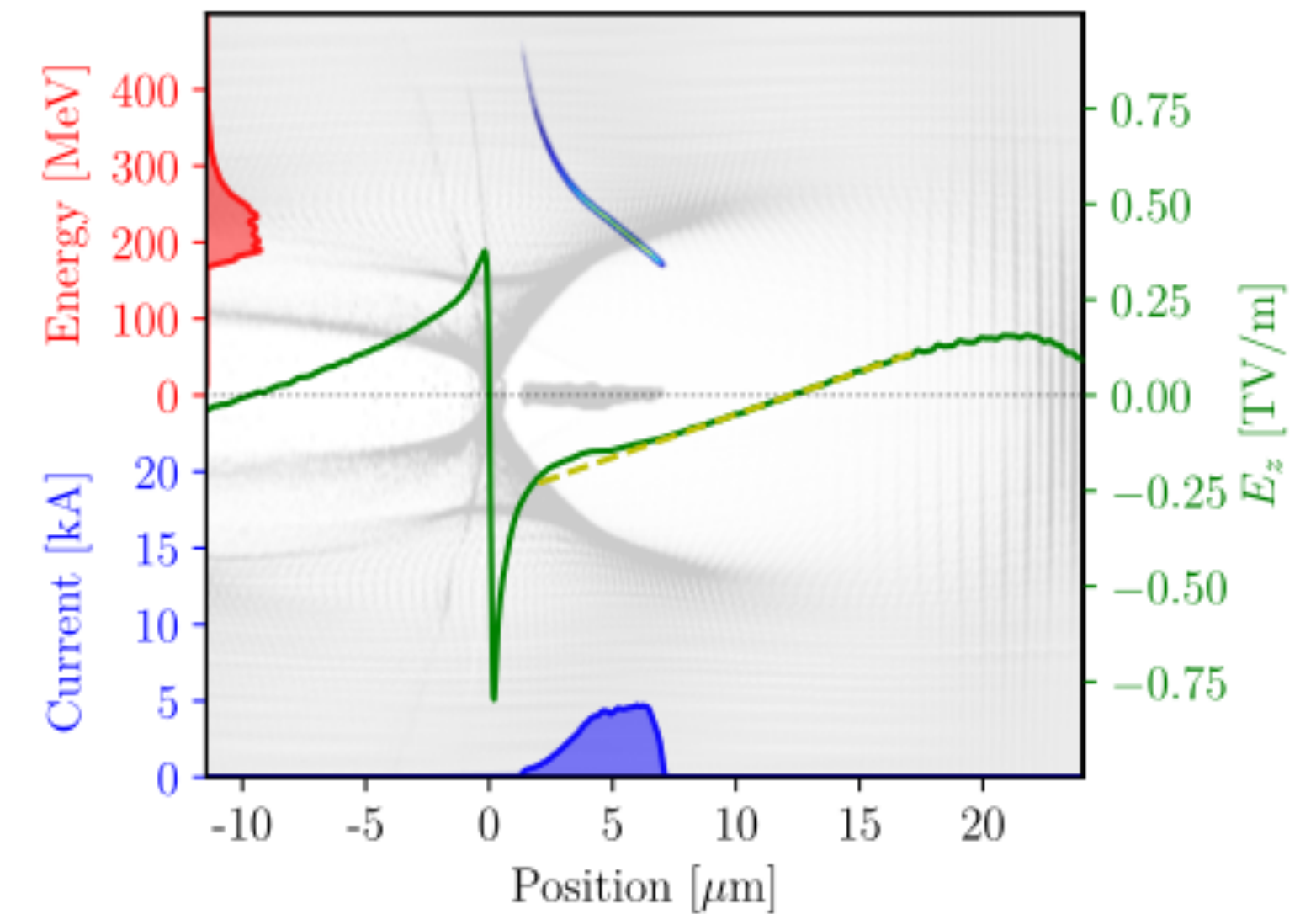
γ_p

Modeling laser-plasma interaction

The Particle-In-Cell Method | The Output

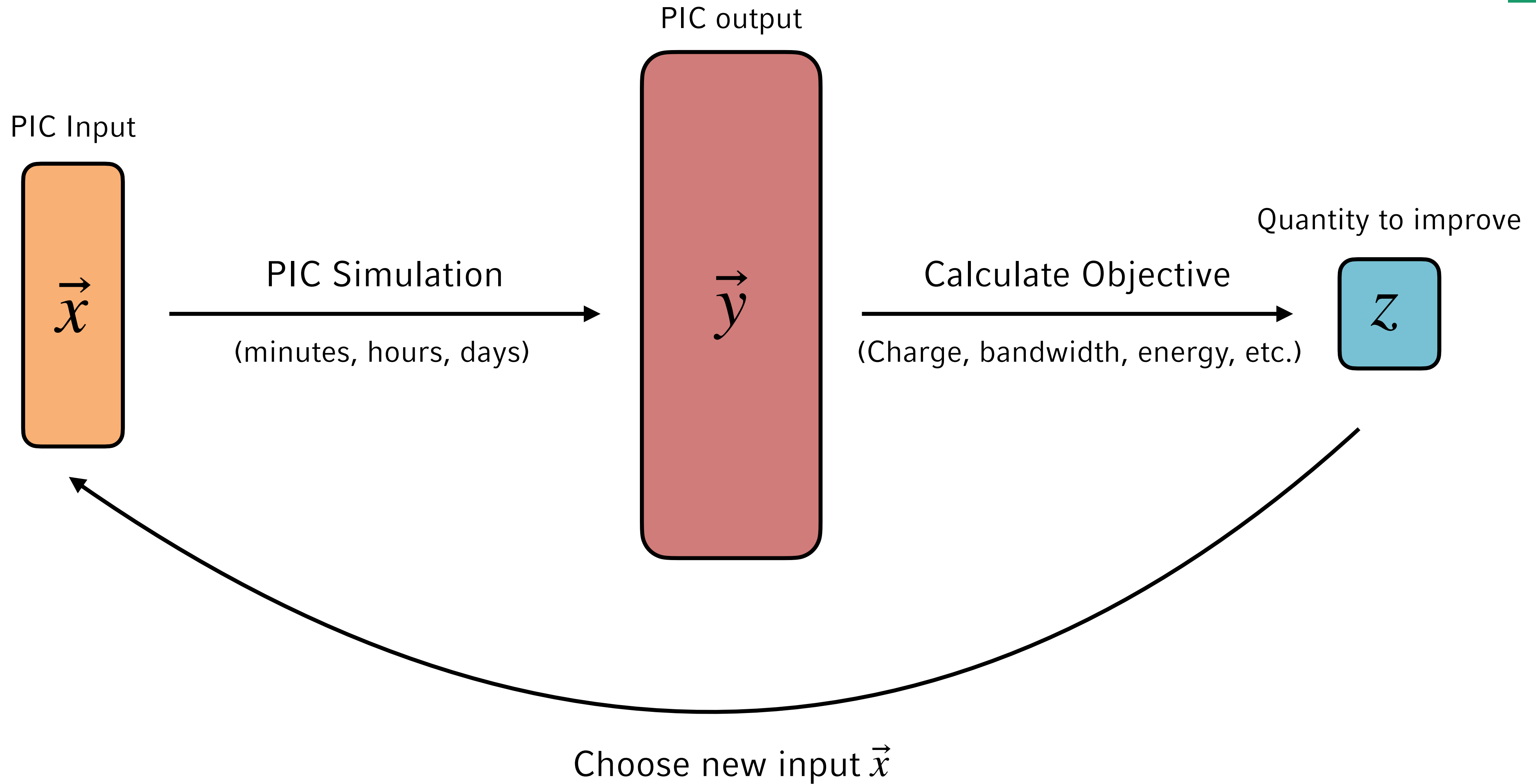


- **Field data:** Meshes of all fields, as well as the deposited density and currents
- **Particle data:** List of macro particles with their individual weights and the phase space values (position, momentum).
- **We are talking about GBs of data.** But often we are actually **only interested in a single number that we can optimize.**
- We do this via an **objective function** that **reduces the entire distribution function to a single scalar number**
 $O[f(\vec{x}, \vec{v}, t)] = z.$



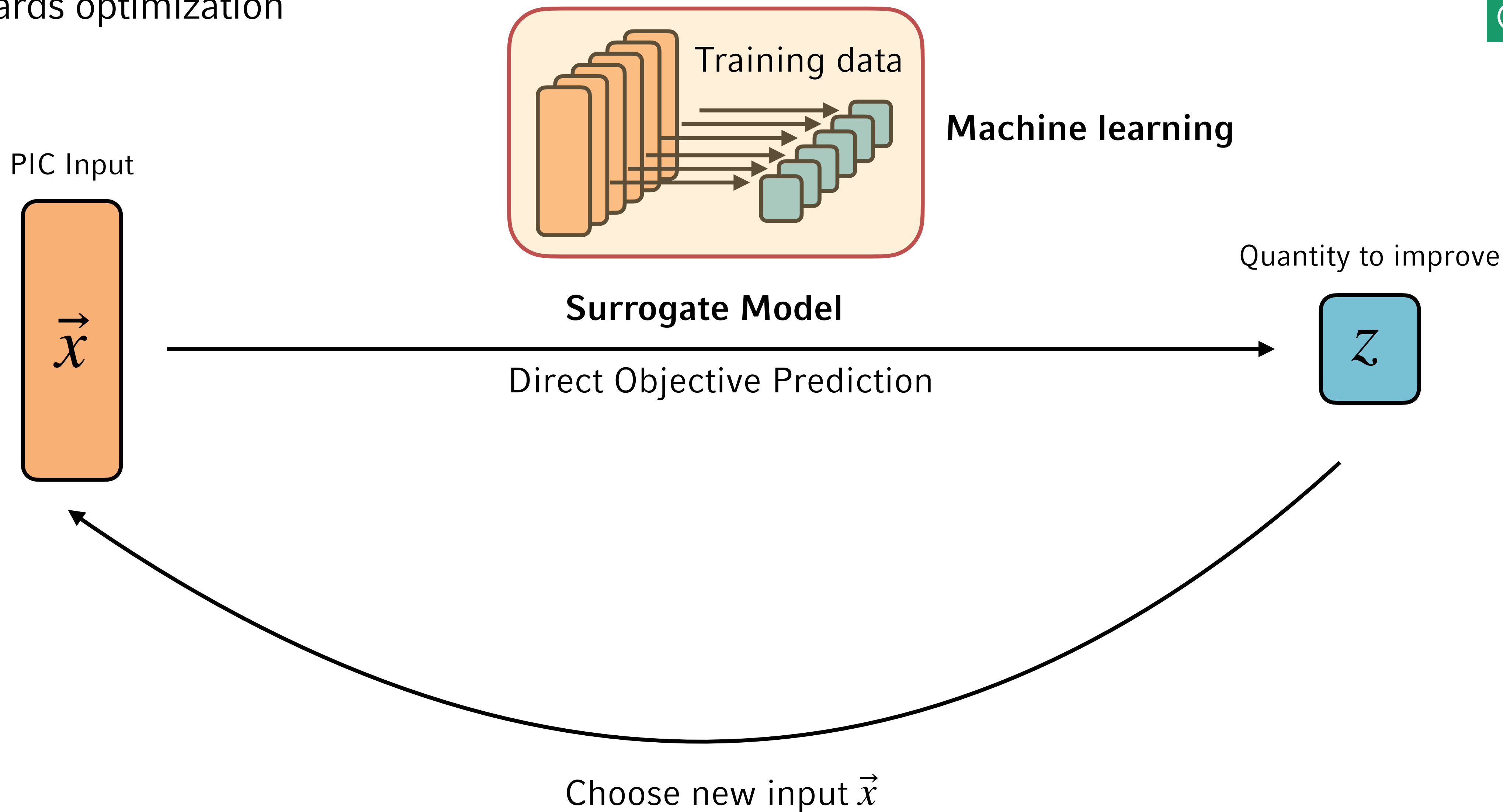
Modeling laser-plasma interaction

Towards optimization



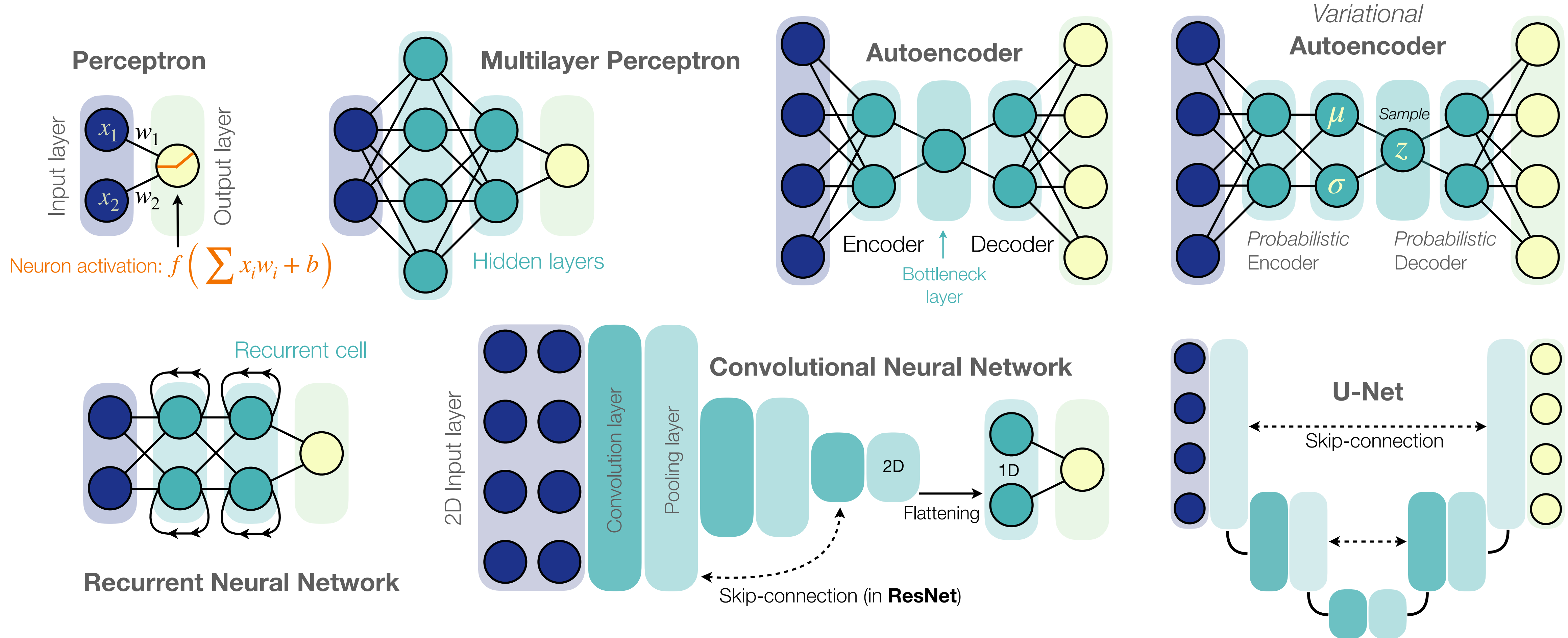
Modeling laser-plasma interaction

Towards optimization



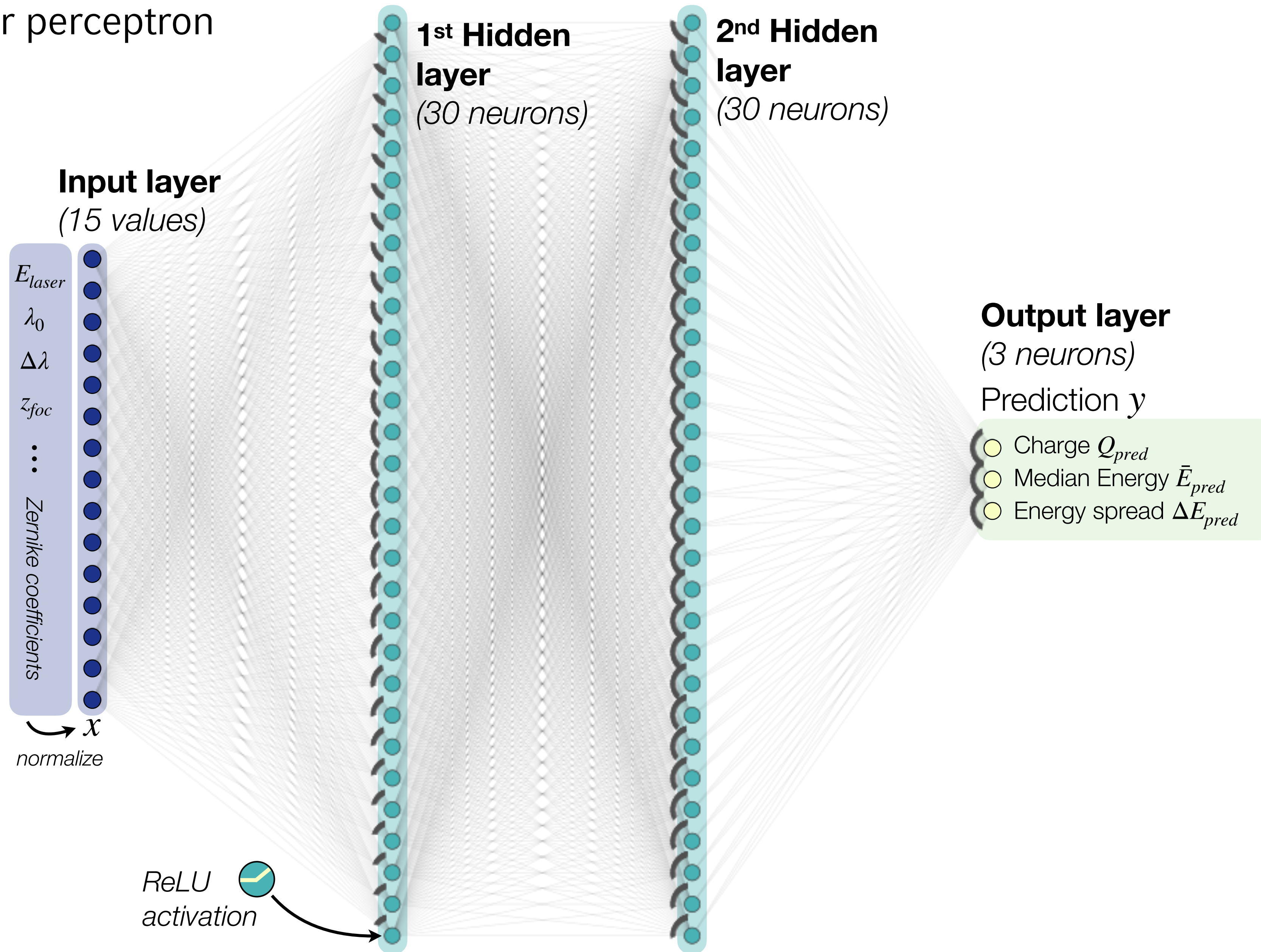
Modeling laser-plasma accelerators

Which neural network architecture to choose?



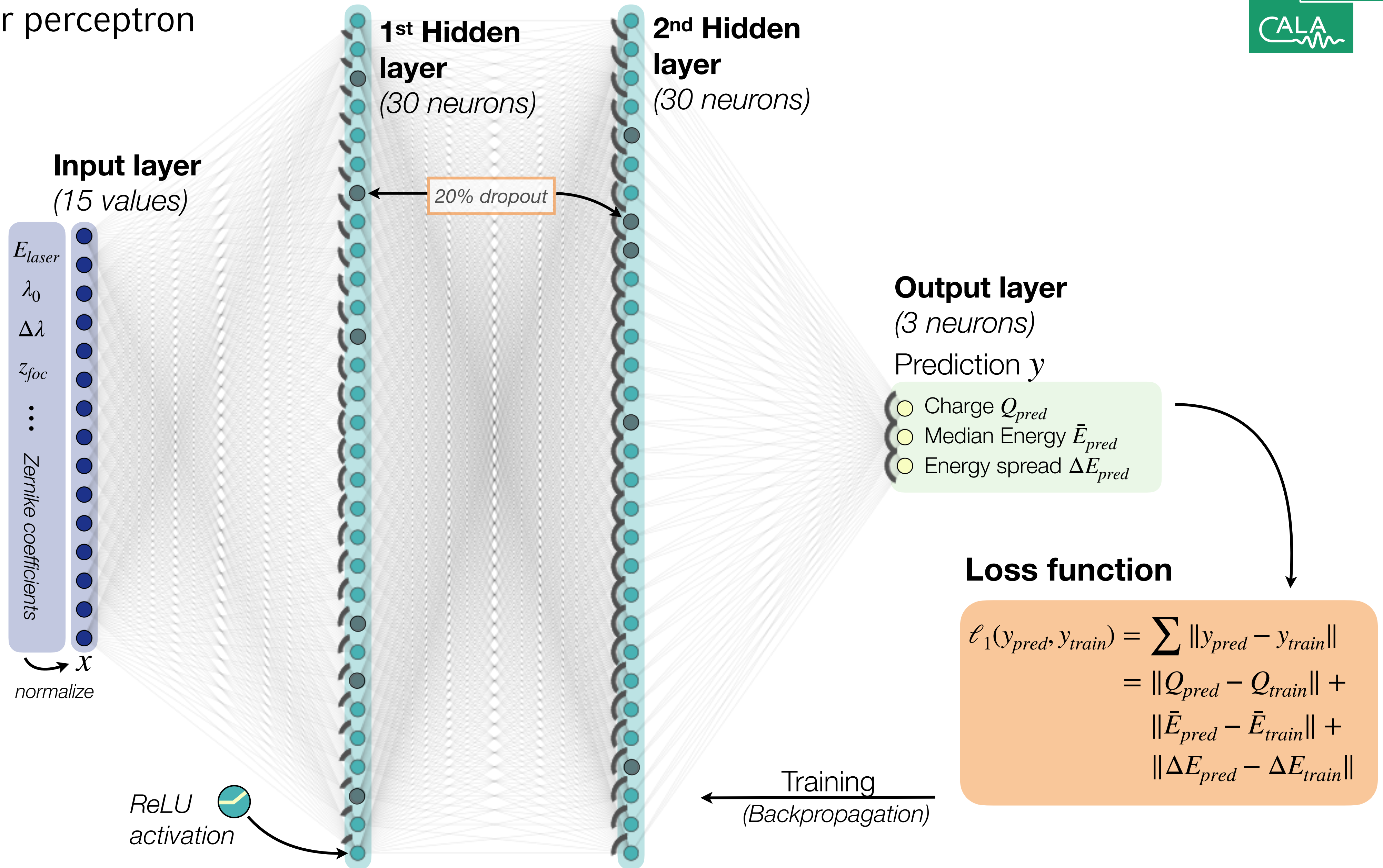
Modeling laser-plasma accelerators

The multilayer perceptron



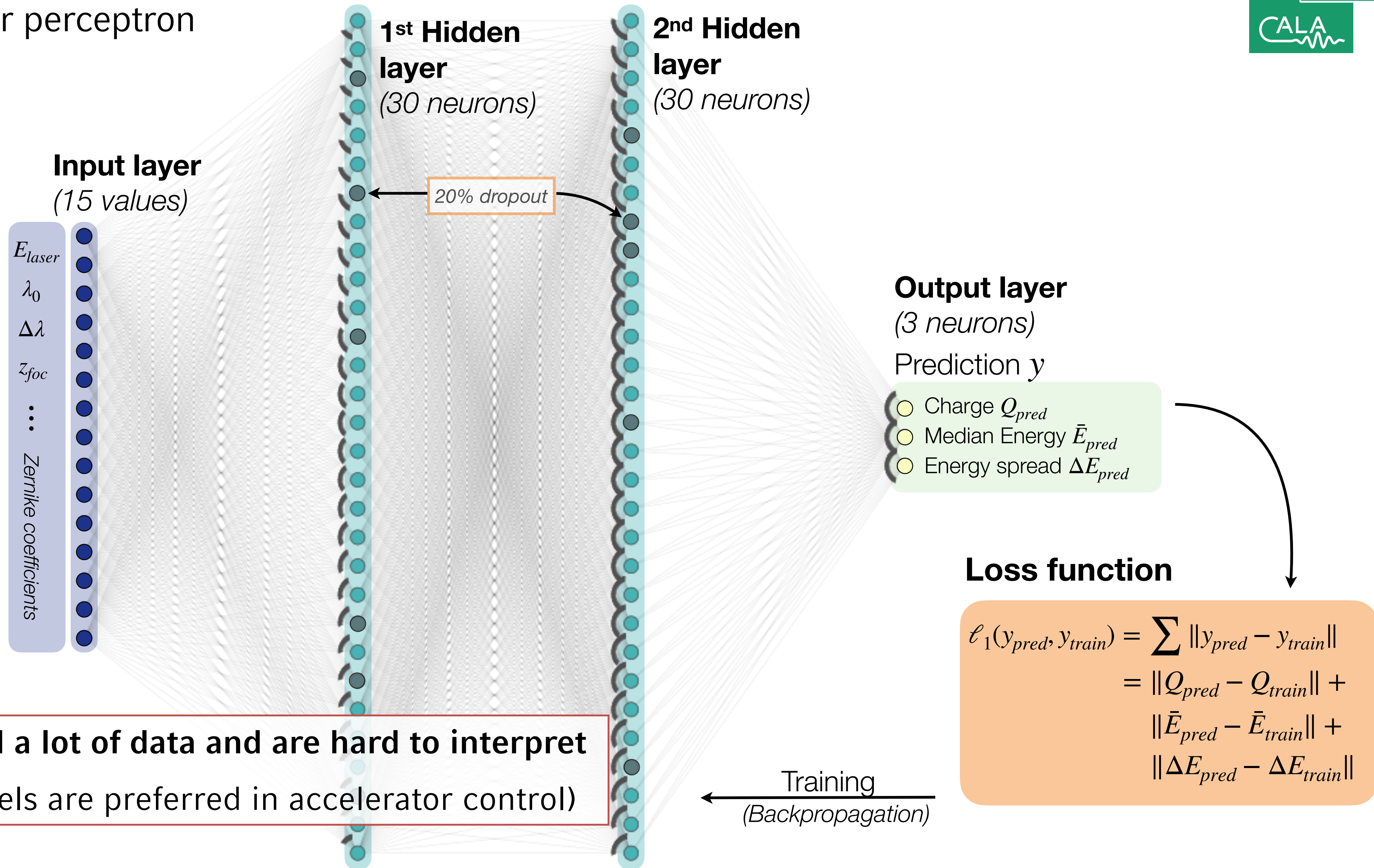
Modeling laser-plasma accelerators

The multilayer perceptron



Modeling laser-plasma accelerators

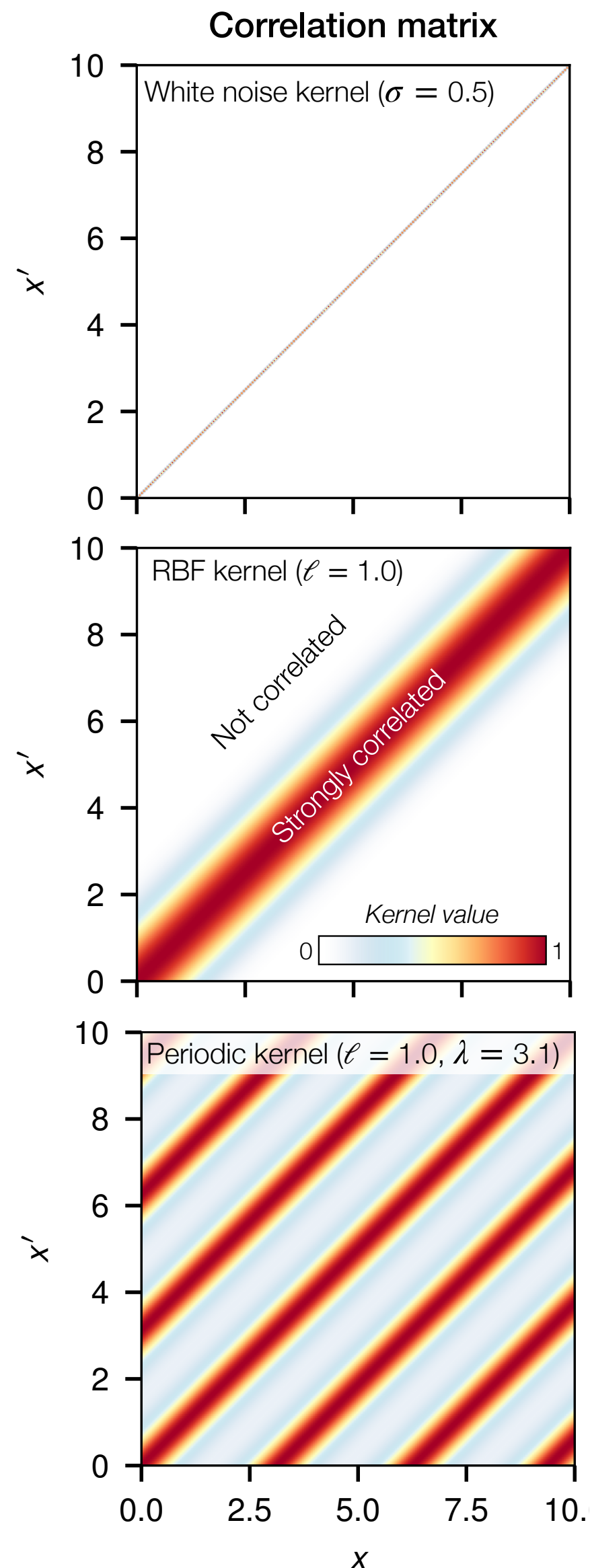
The multilayer perceptron



Neural nets need a lot of data and are hard to interpret
(explainable models are preferred in accelerator control)

Modeling laser-plasma accelerators

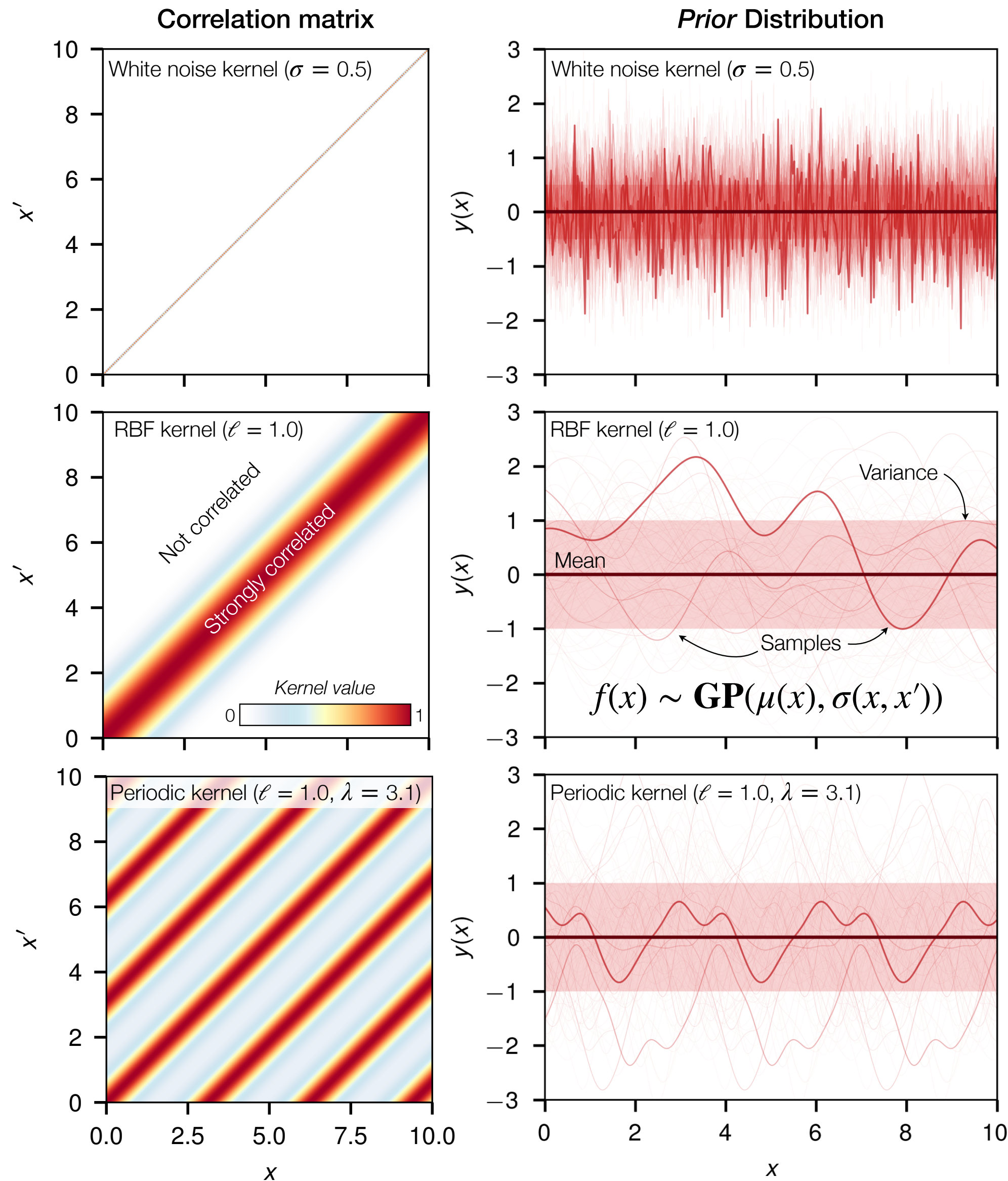
Gaussian Process Regression



- **Non-parametric method:** Predicts function values based on observed data without a predetermined model.
- **Covariance function/Kernel:** Defines the relationship between points, capturing their correlations.
- **Probabilistic description:** Provides a full description of the function, including mean and uncertainty.

Modeling laser-plasma accelerators

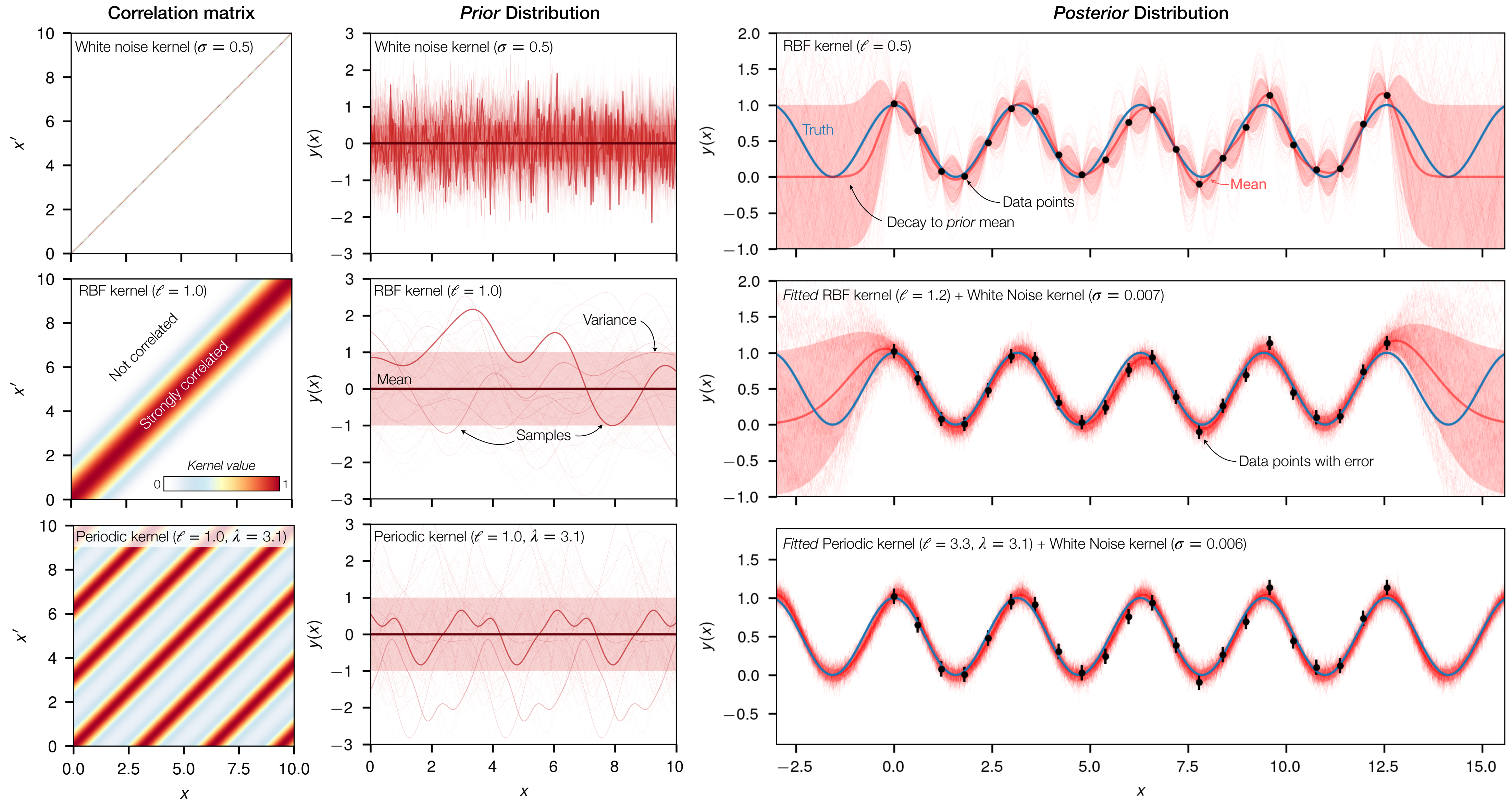
Gaussian Process Regression



- A. Döpp et al. **Data-driven Science and Machine Learning Methods in Laser-Plasma Physics**, High Power Laser Science and Engineering **11** 55 (2023) | [arXiv:2212.00026](https://arxiv.org/abs/2212.00026) (2022)

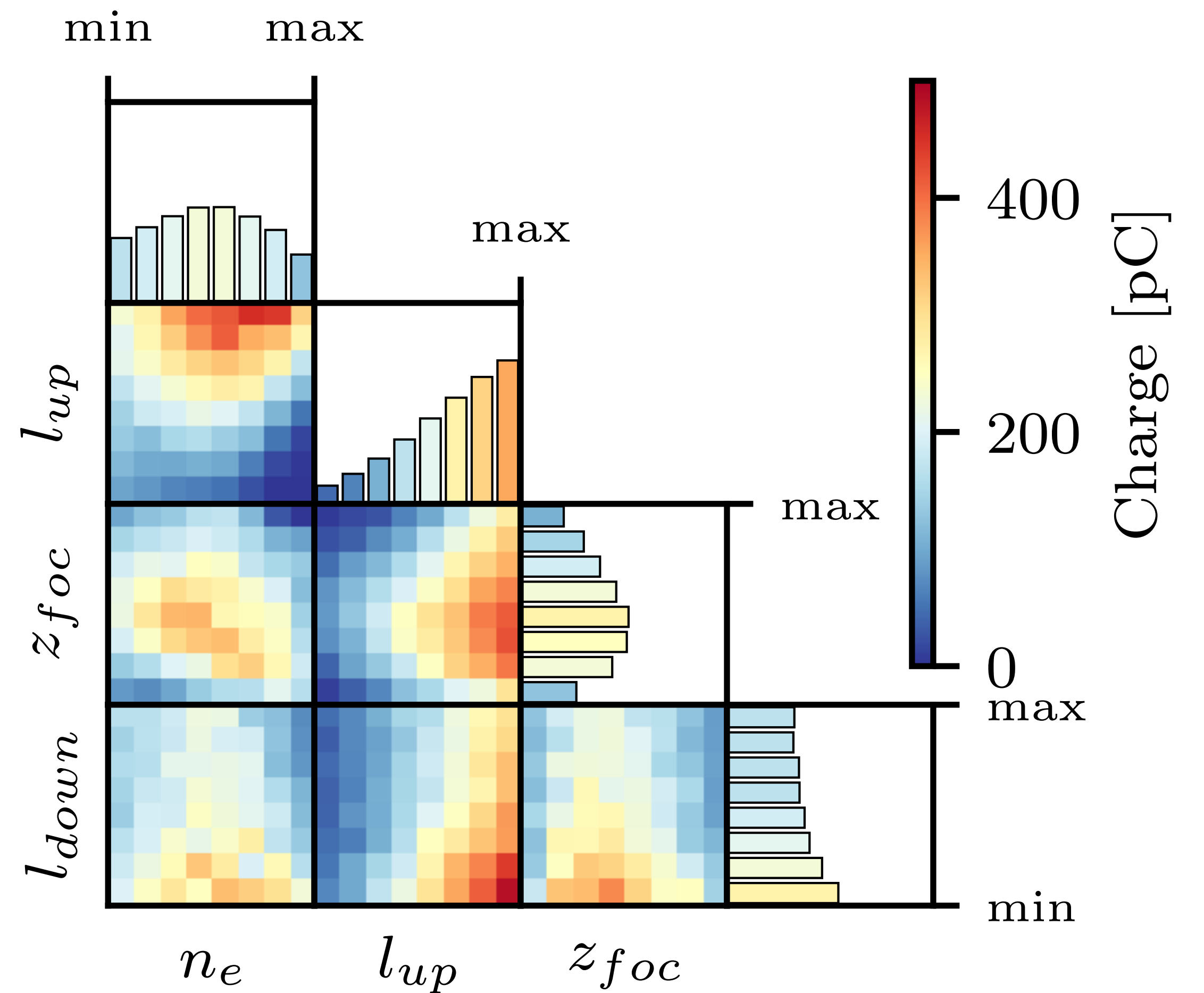
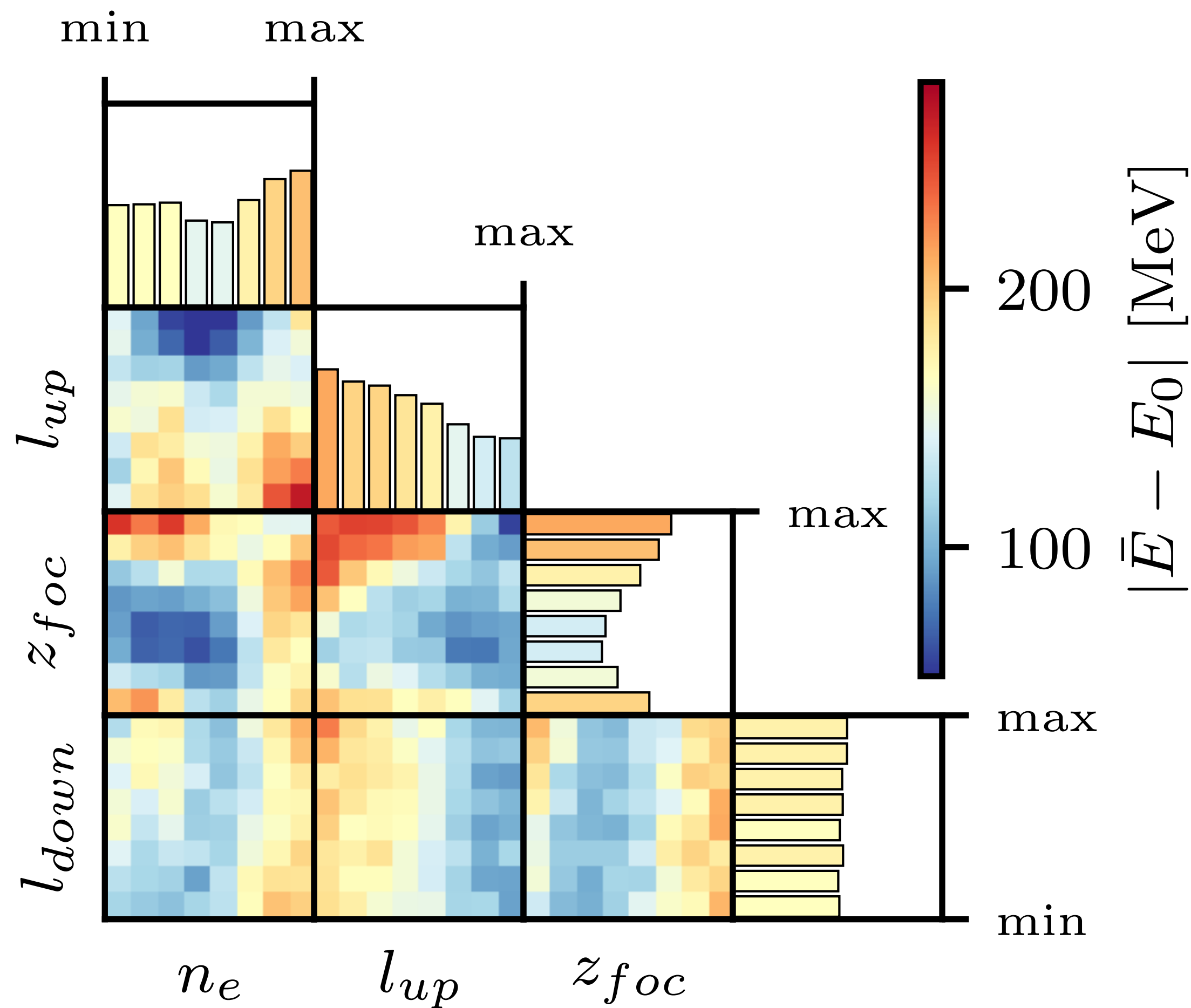
Modeling laser-plasma accelerators

Gaussian Process Regression



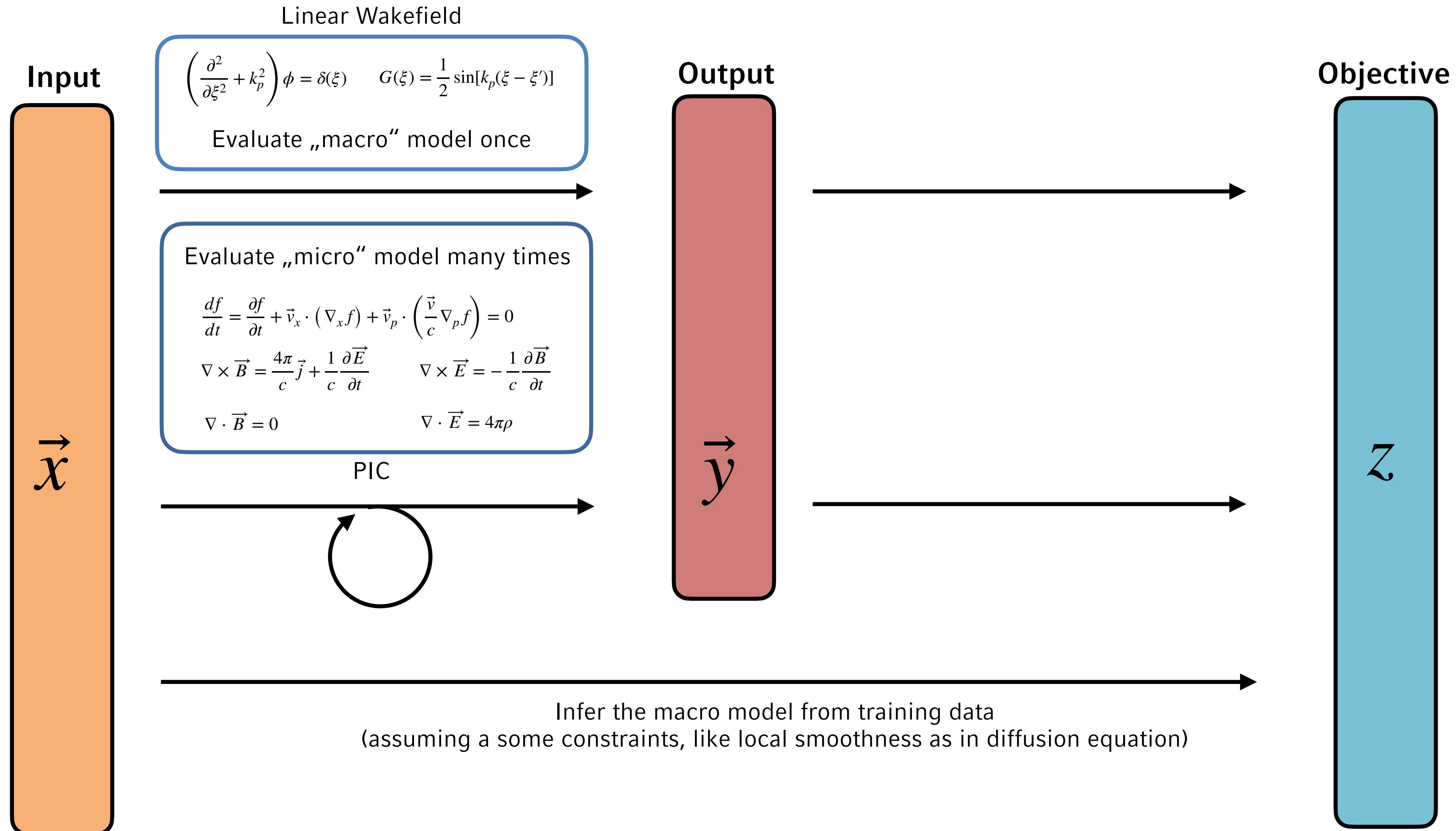
Modeling laser-plasma accelerators

Gaussian Process Regression



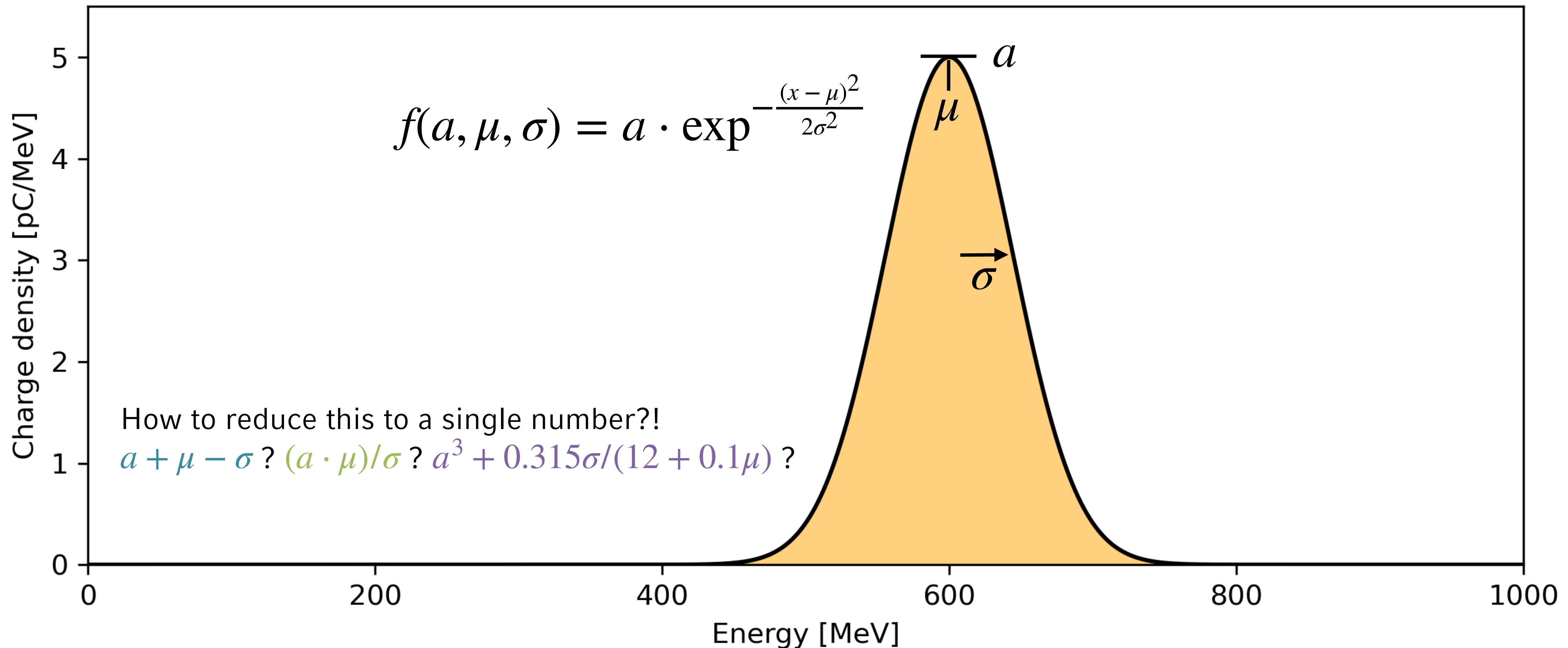
Modeling laser-plasma accelerators

Summarizing the different approaches



Optimizing laser-plasma accelerators

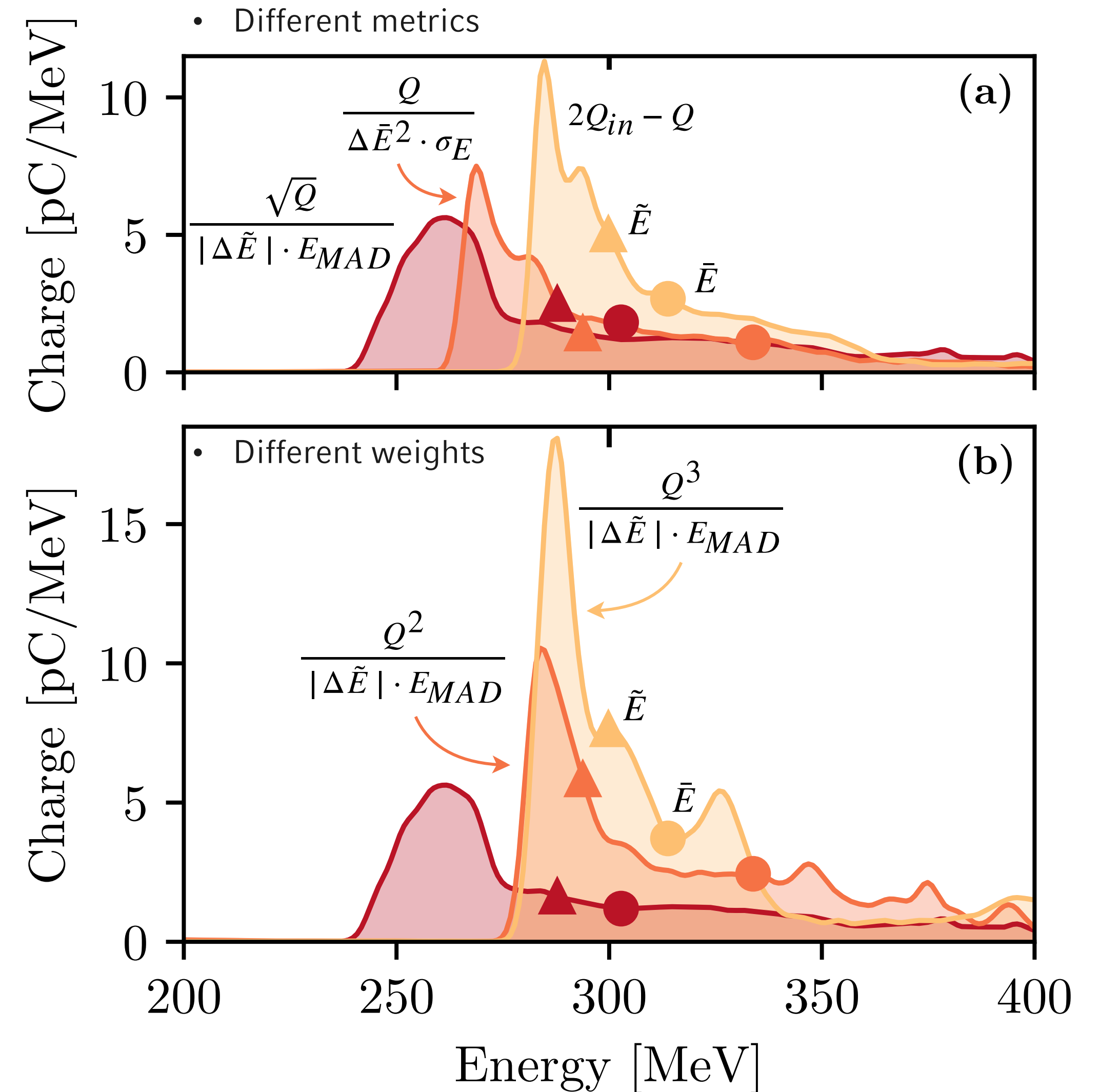
Choosing the right objective



Optimizing laser-plasma accelerators

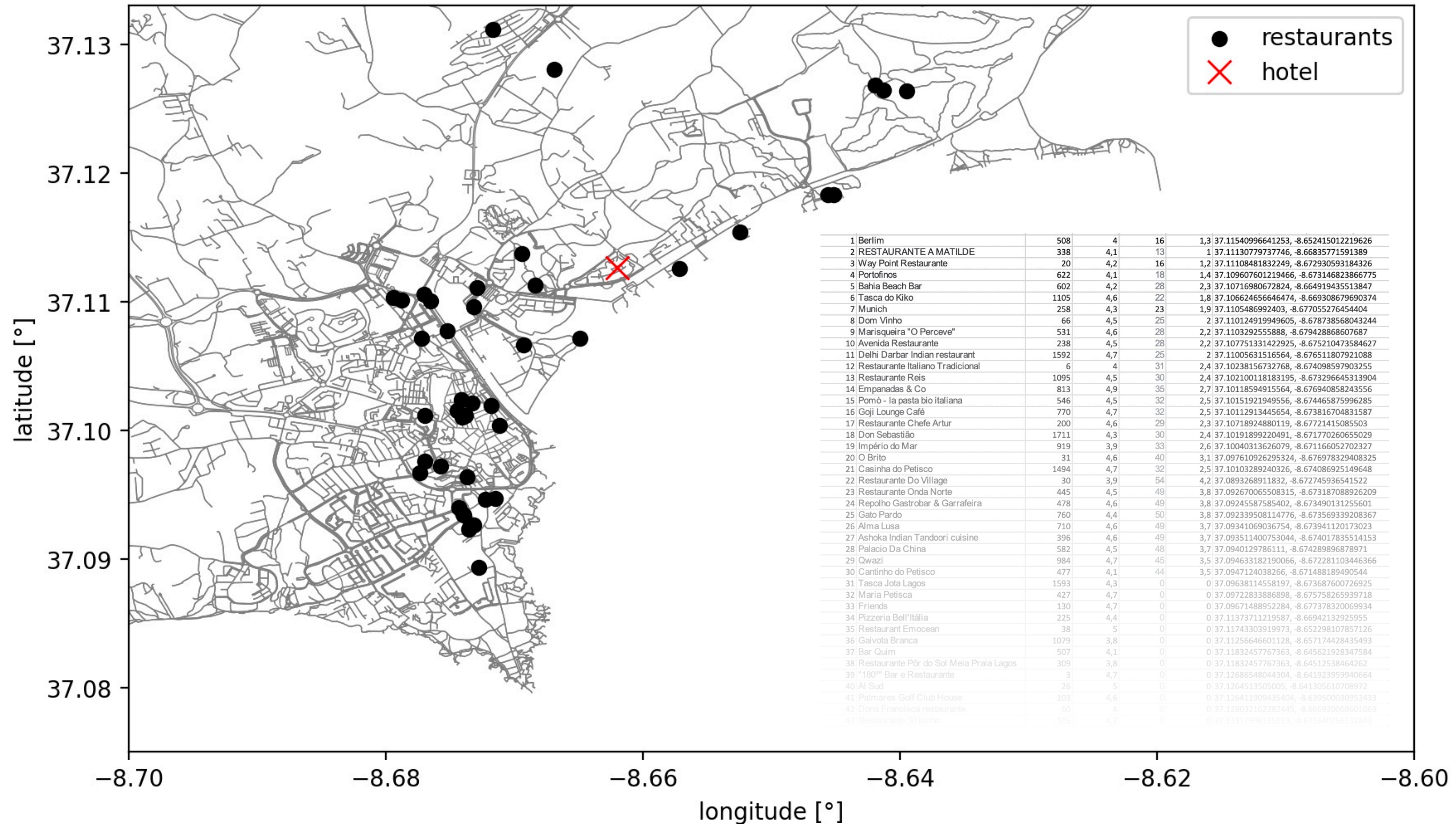
Choosing the right objective

- Say we want to optimize three electron beam parameters:
 - Charge Q (total charge, charge within FWHM, etc.)
 - Bandwidth (standard deviation σ_E , median absolute deviation E_{MAD} , etc.)
 - Distance to a target energy $|E_{target} - E|$ (using mean energy, median energy, peak energy, etc.)
- Choosing **different metrics or weights** for each objective **changes the outcome in an a priori unknown way!**
- How to solve this problem?



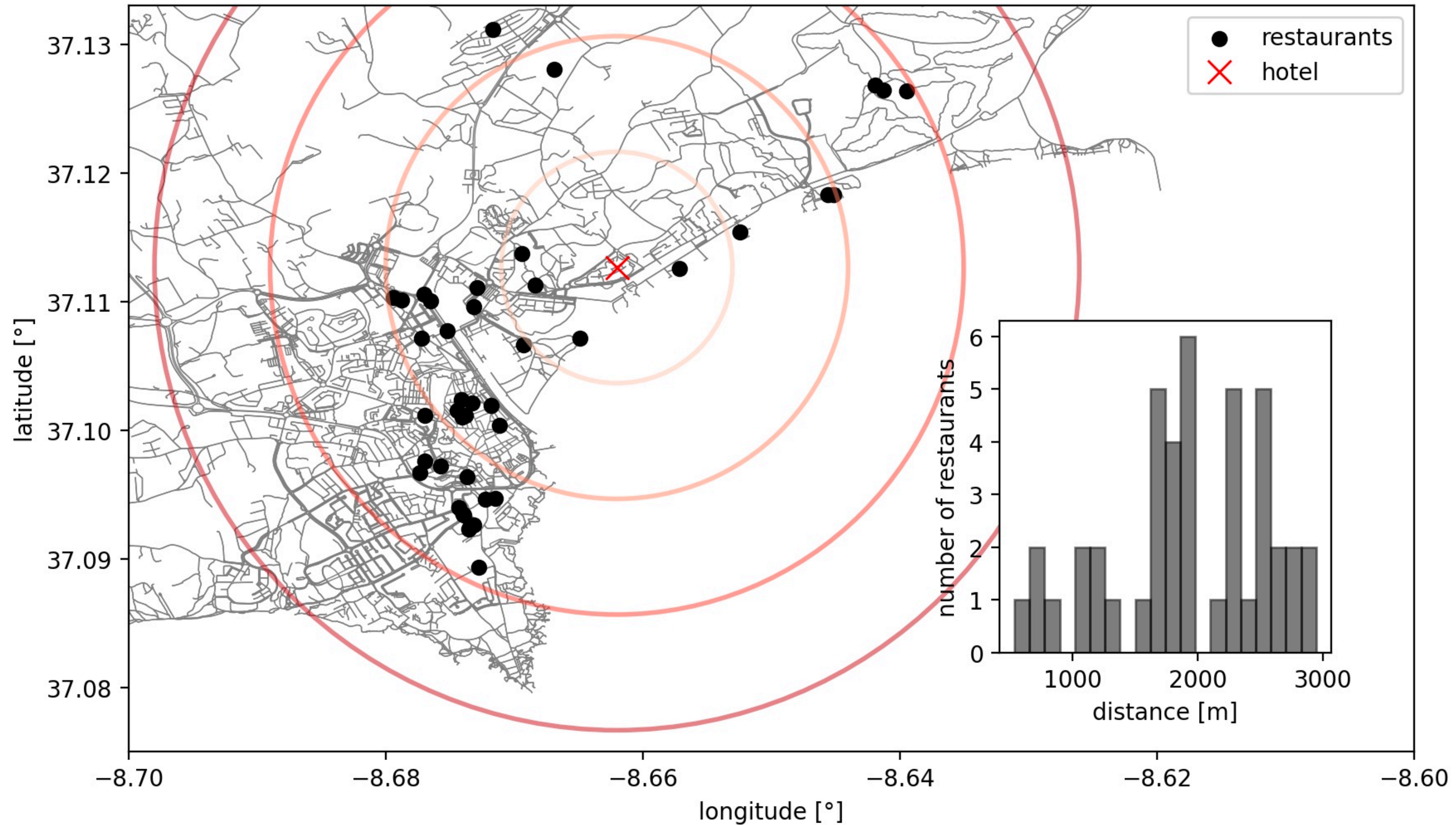
Optimizing laser-plasma accelerators

Multi-objective optimization | The conference dinner problem



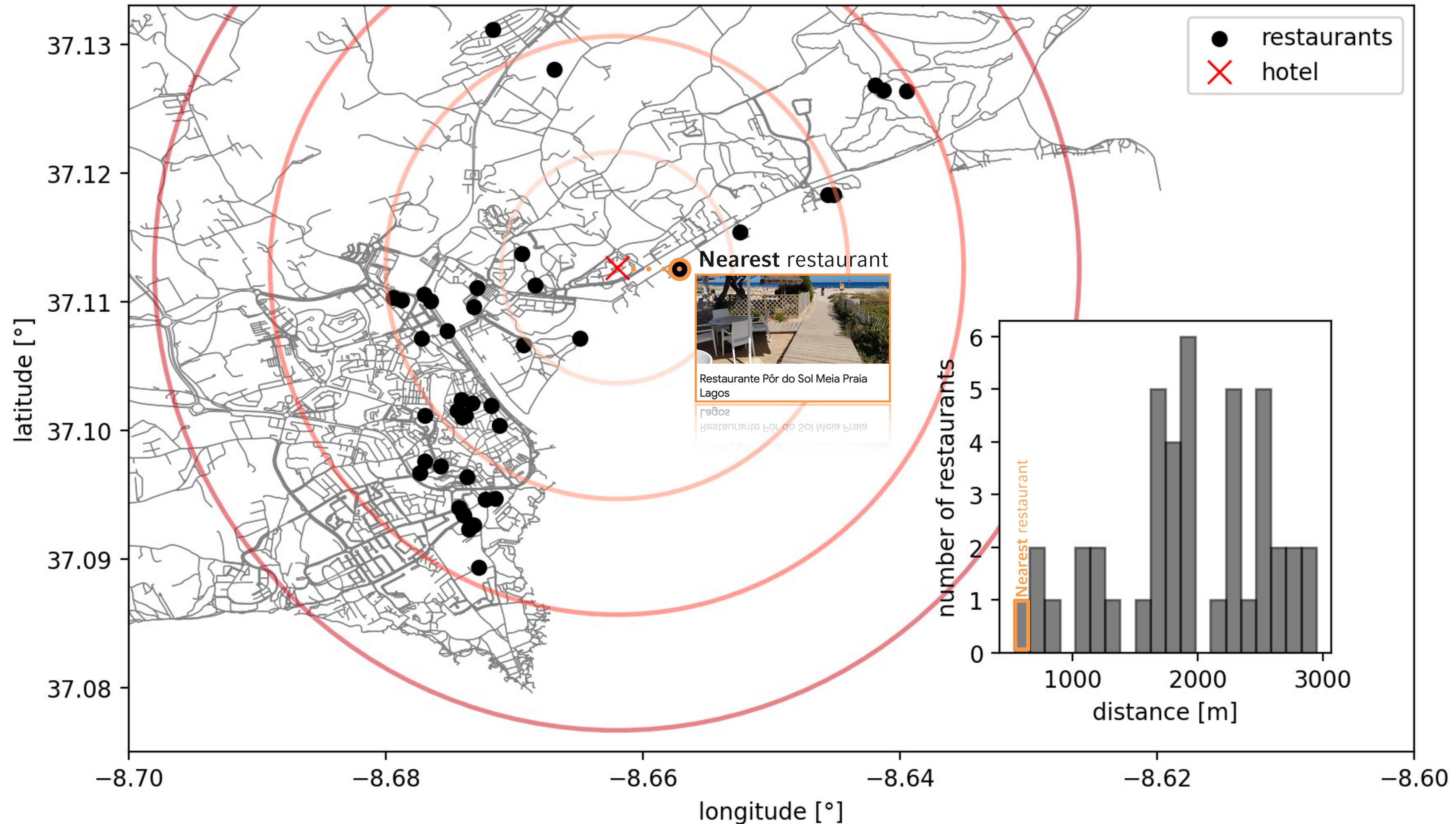
Optimizing laser-plasma accelerators

Multi-objective optimization | The conference dinner problem



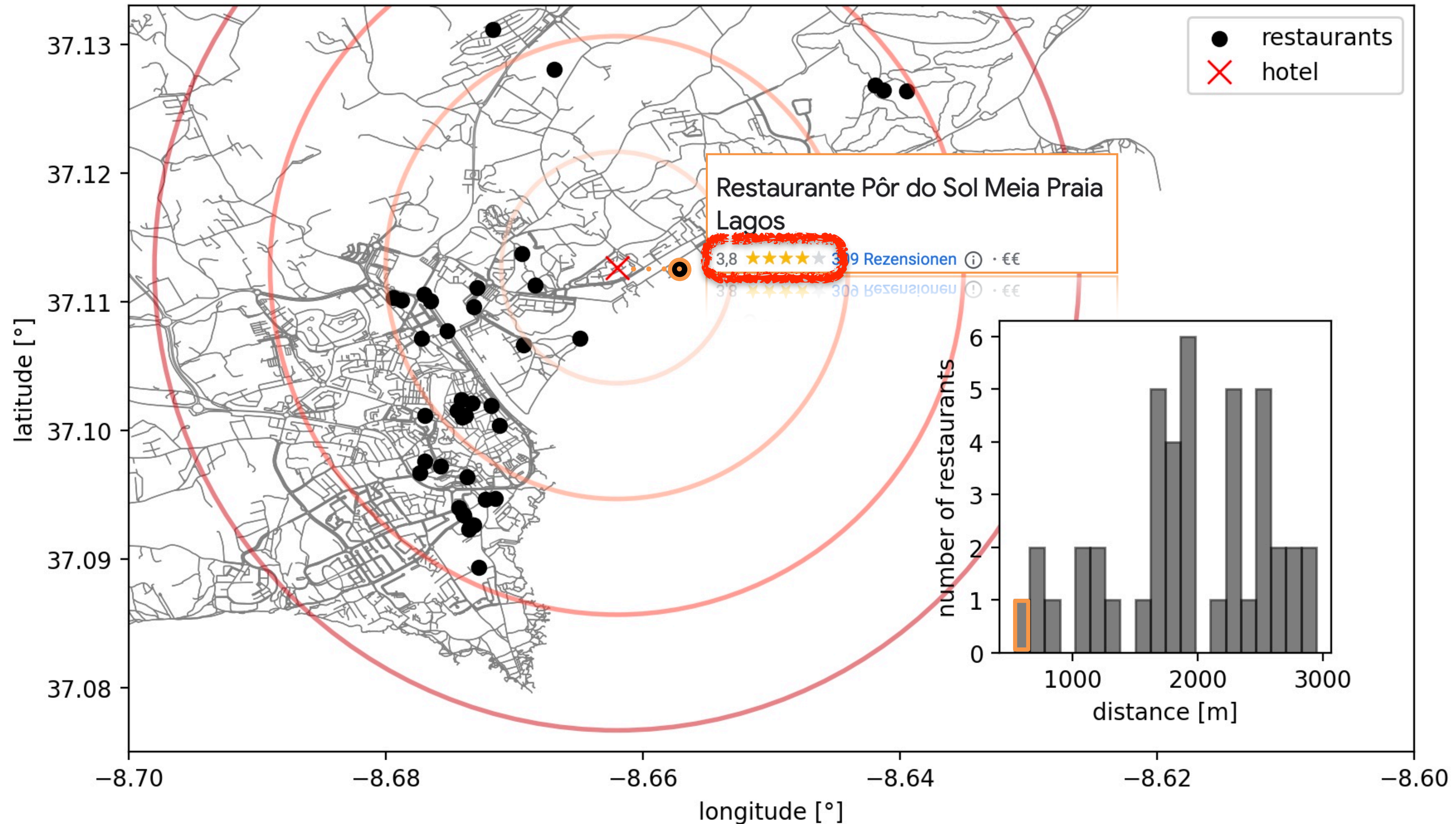
Optimizing laser-plasma accelerators

Multi-objective optimization | The conference dinner problem



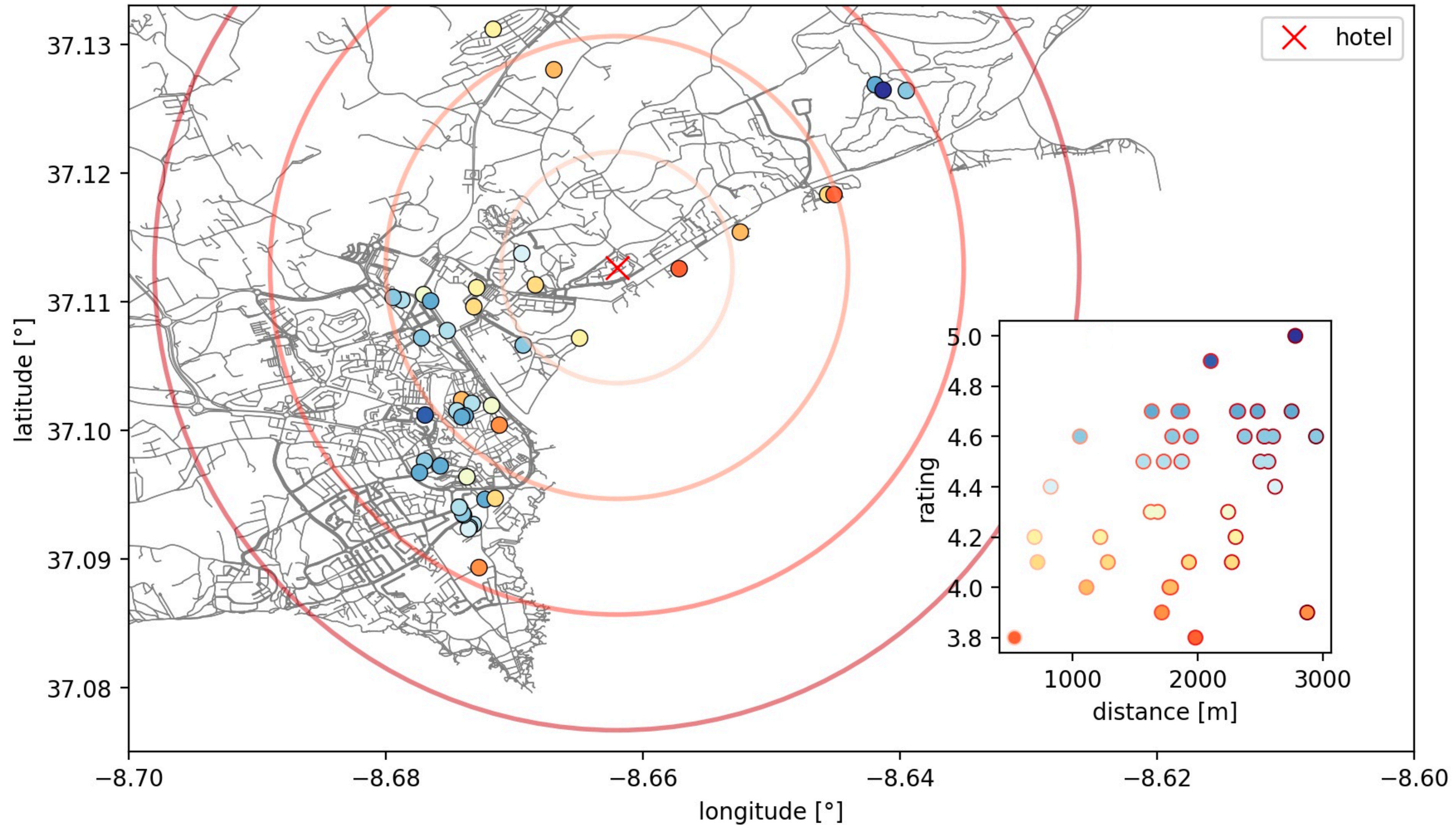
Optimizing laser-plasma accelerators

Multi-objective optimization | The conference dinner problem



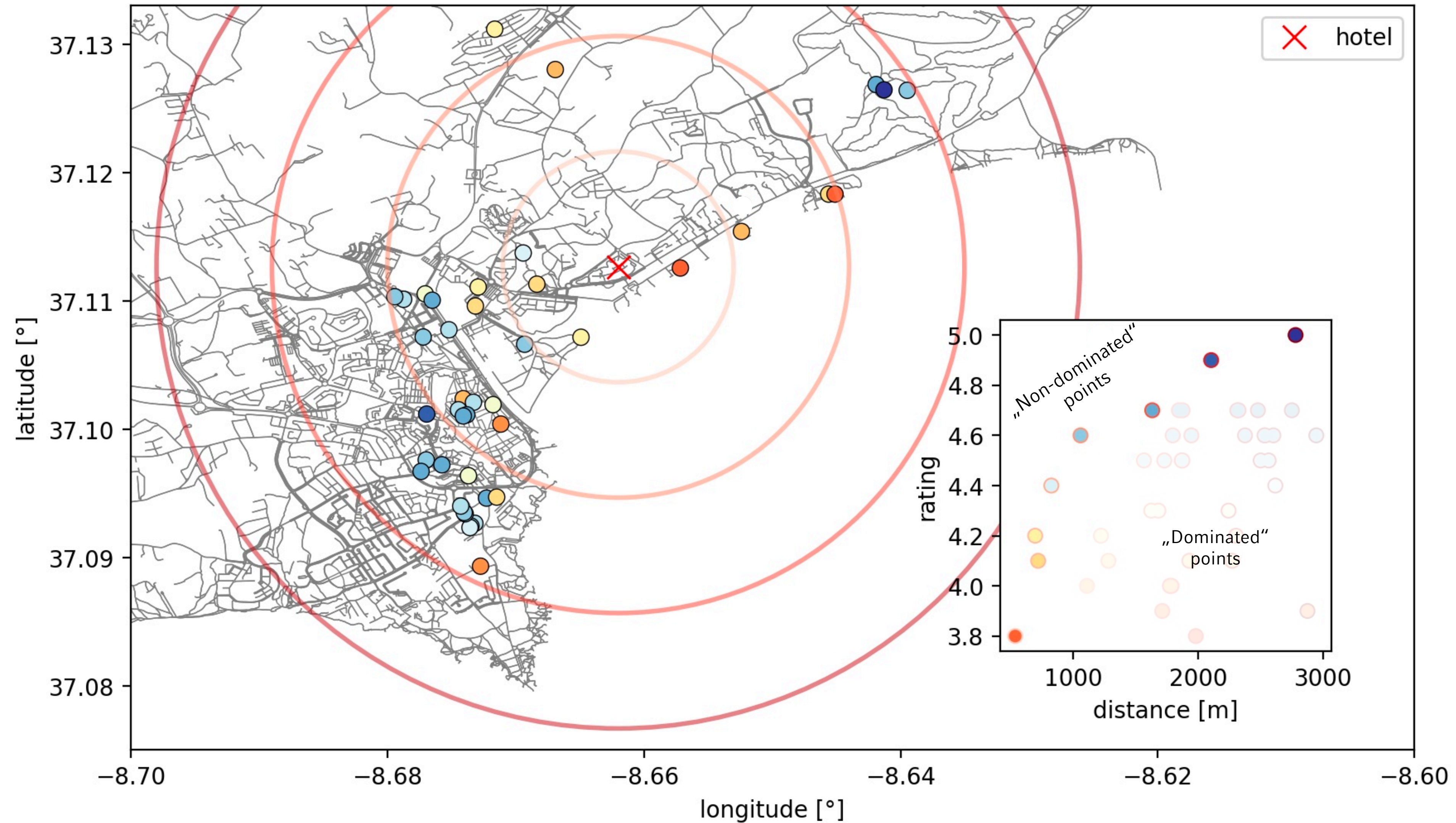
Optimizing laser-plasma accelerators

Multi-objective optimization | The conference dinner problem



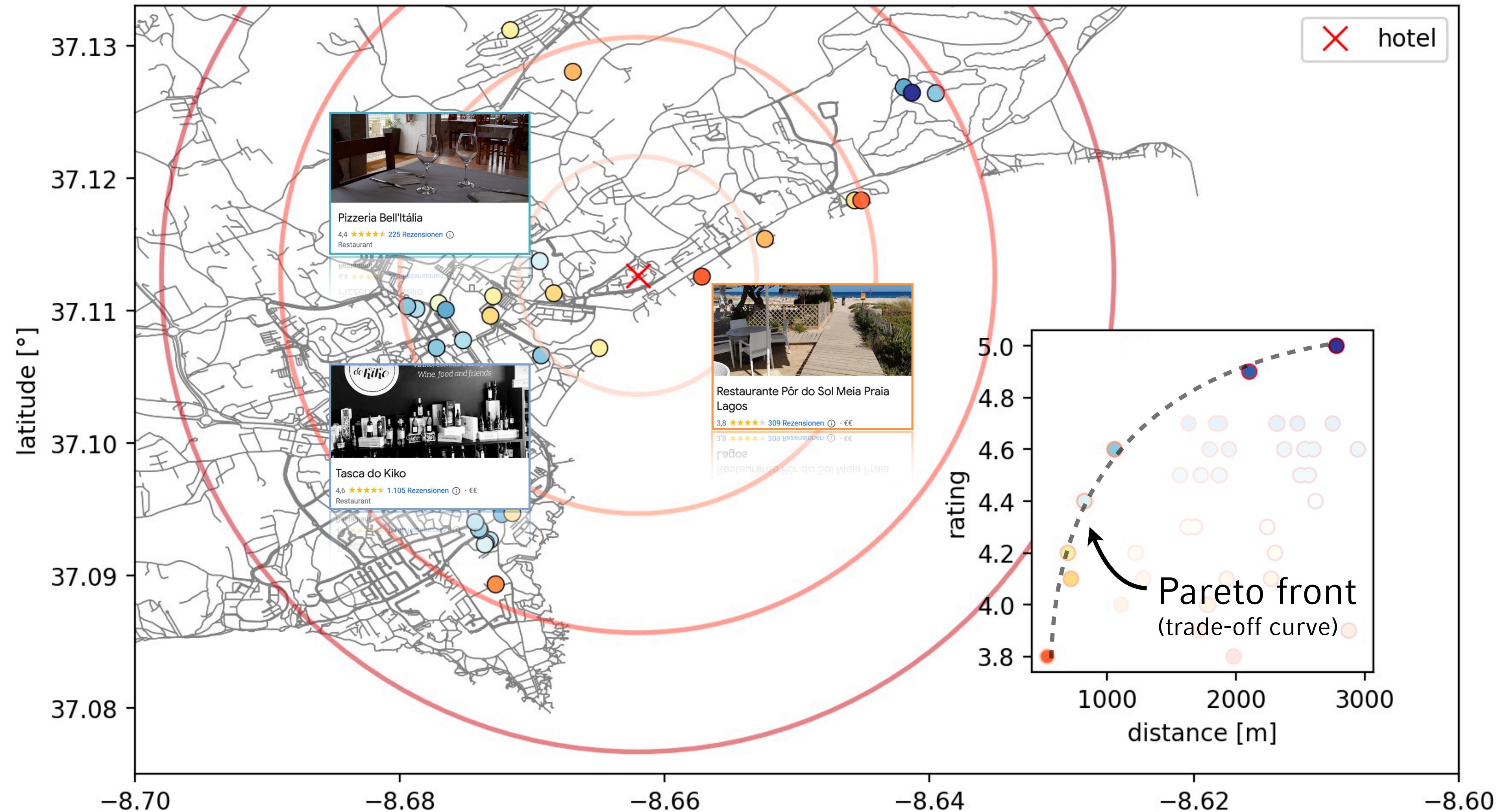
Optimizing laser-plasma accelerators

Multi-objective optimization | The conference dinner problem



Optimizing laser-plasma accelerators

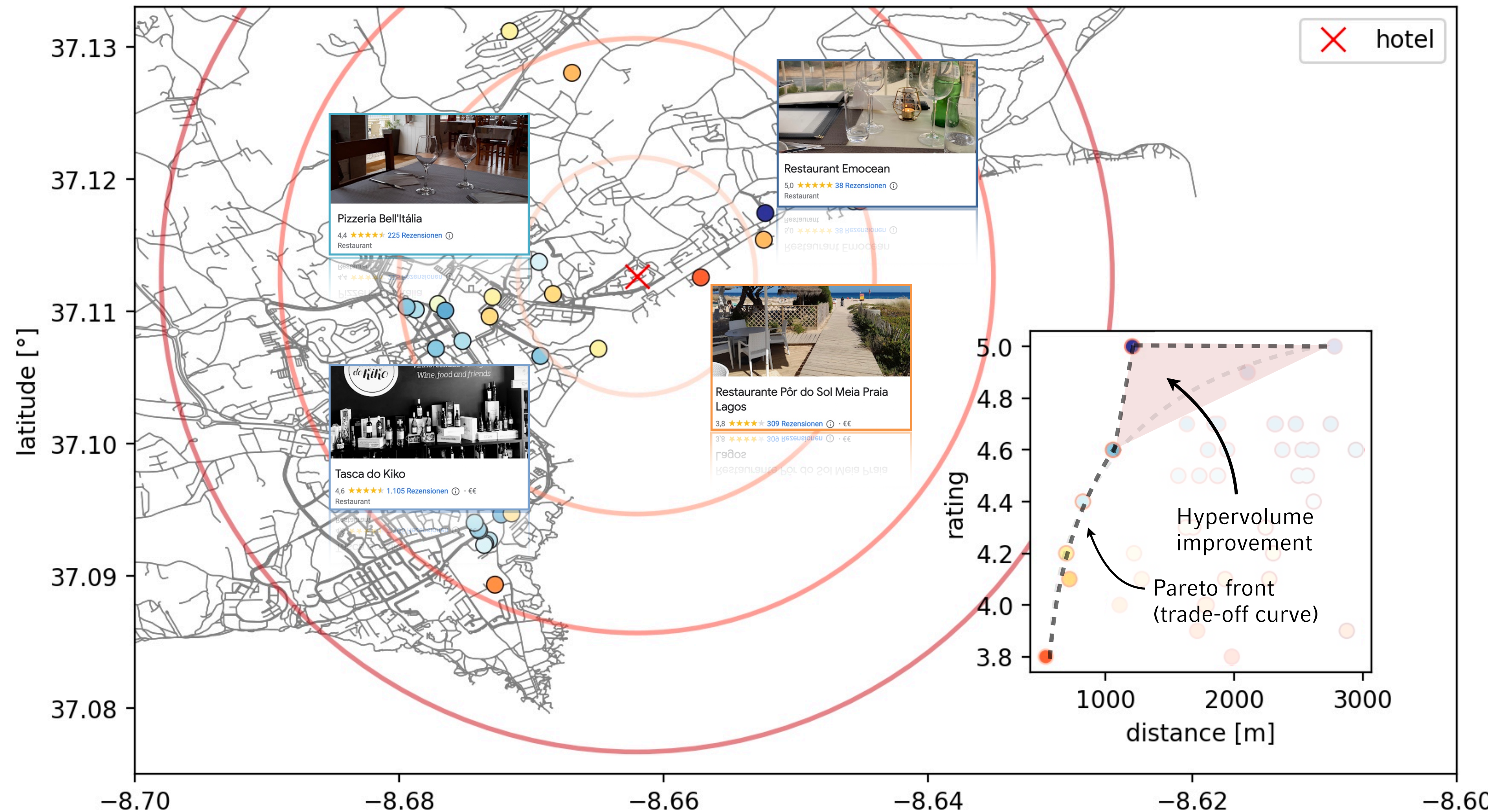
Multi-objective optimization | The conference dinner problem



In multi-objective optimization we have **multiple (competing) goals with different trade-offs.**

Optimizing laser-plasma accelerators

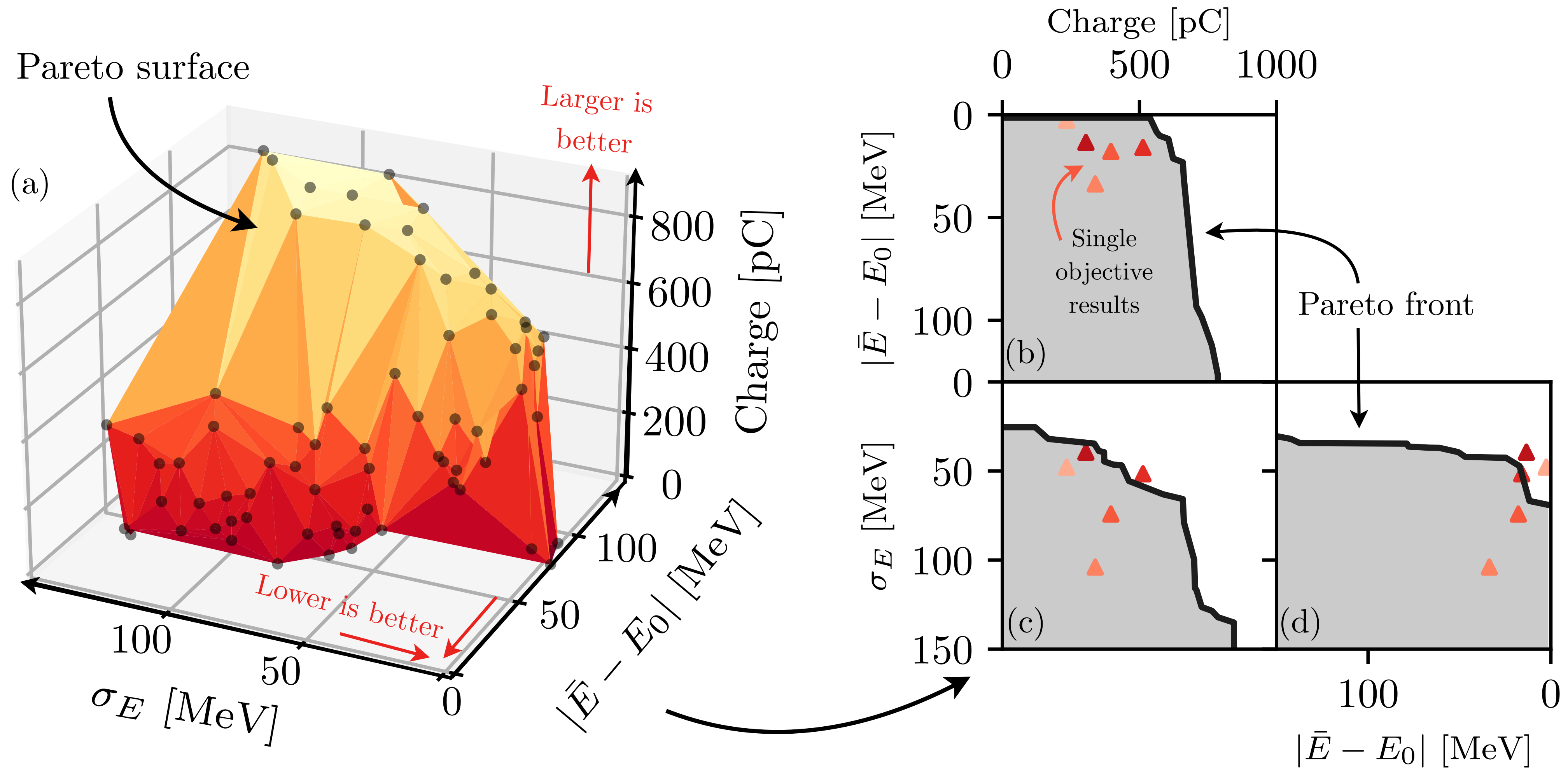
Multi-objective optimization | The conference dinner problem



In multi-objective optimization we have **multiple (competing) goals** with **different trade-offs**.

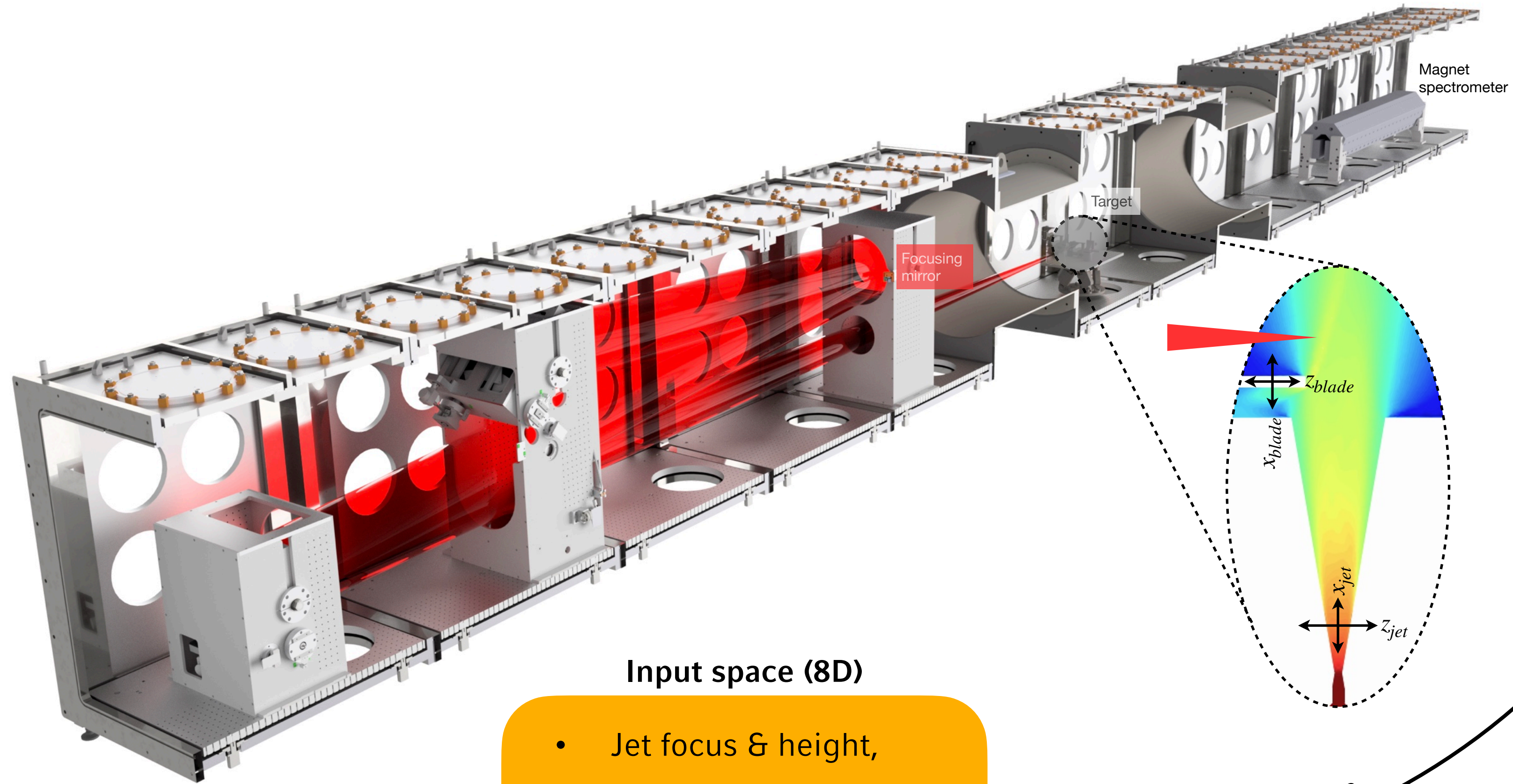
Optimizing laser-plasma accelerators

Multi-objective optimization | PIC simulations



Optimizing laser-plasma accelerators

Multi-objective optimization | Experiments



Input space (8D)

- Jet focus & height,
- Blade focus & height,
- Dispersion (ϕ_2, ϕ_3, ϕ_4)
- Gas Pressure

Output space (3D)

- Charge
- Energy
- Bandwidth

Observation

Optimizing laser-plasma accelerators

Multi-objective optimization | Experiments

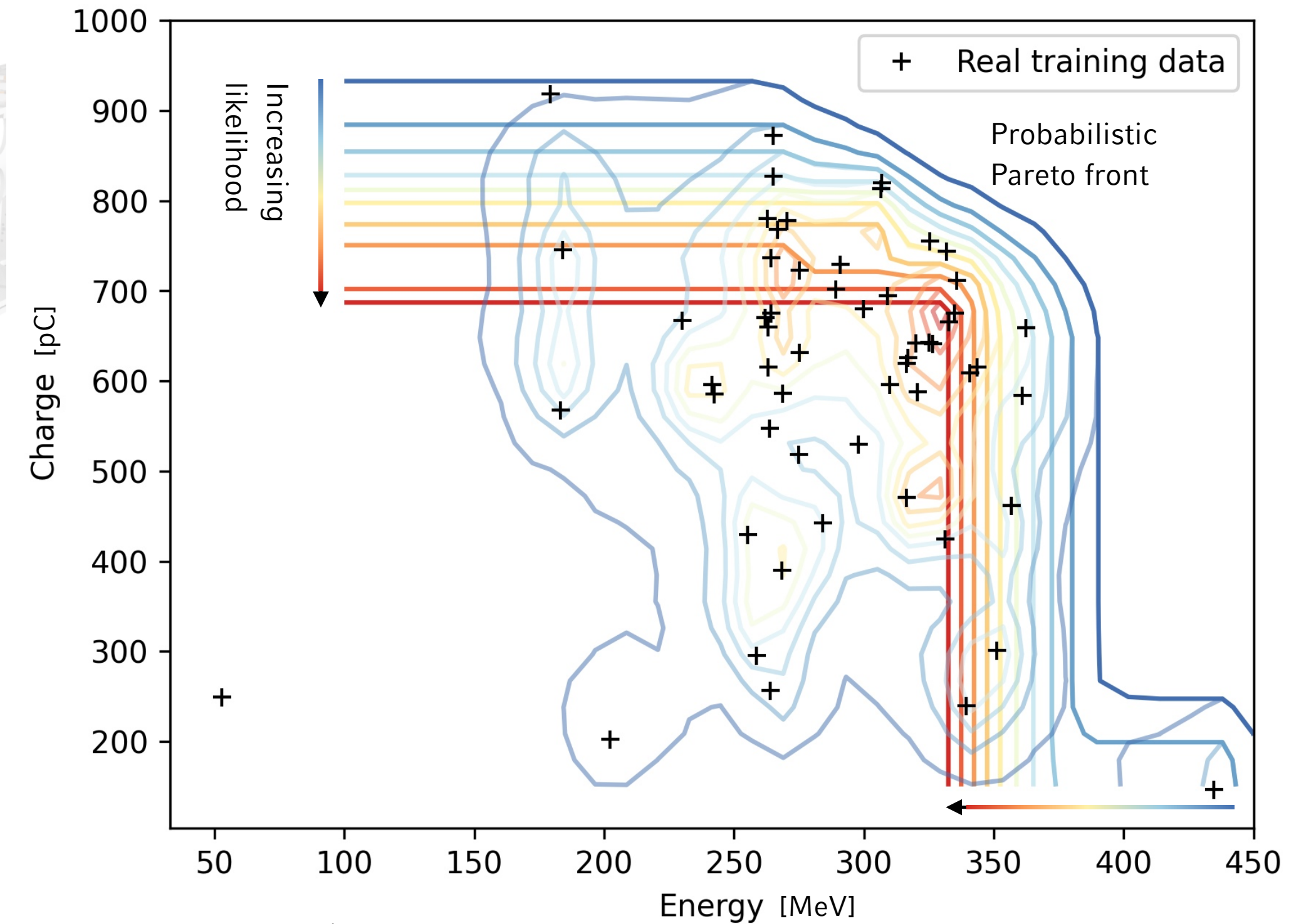


Input space (8D)

- Jet focus & height,
- Blade focus & height,
- Dispersion (ϕ_2, ϕ_3, ϕ_4)
- Gas Pressure

Observation

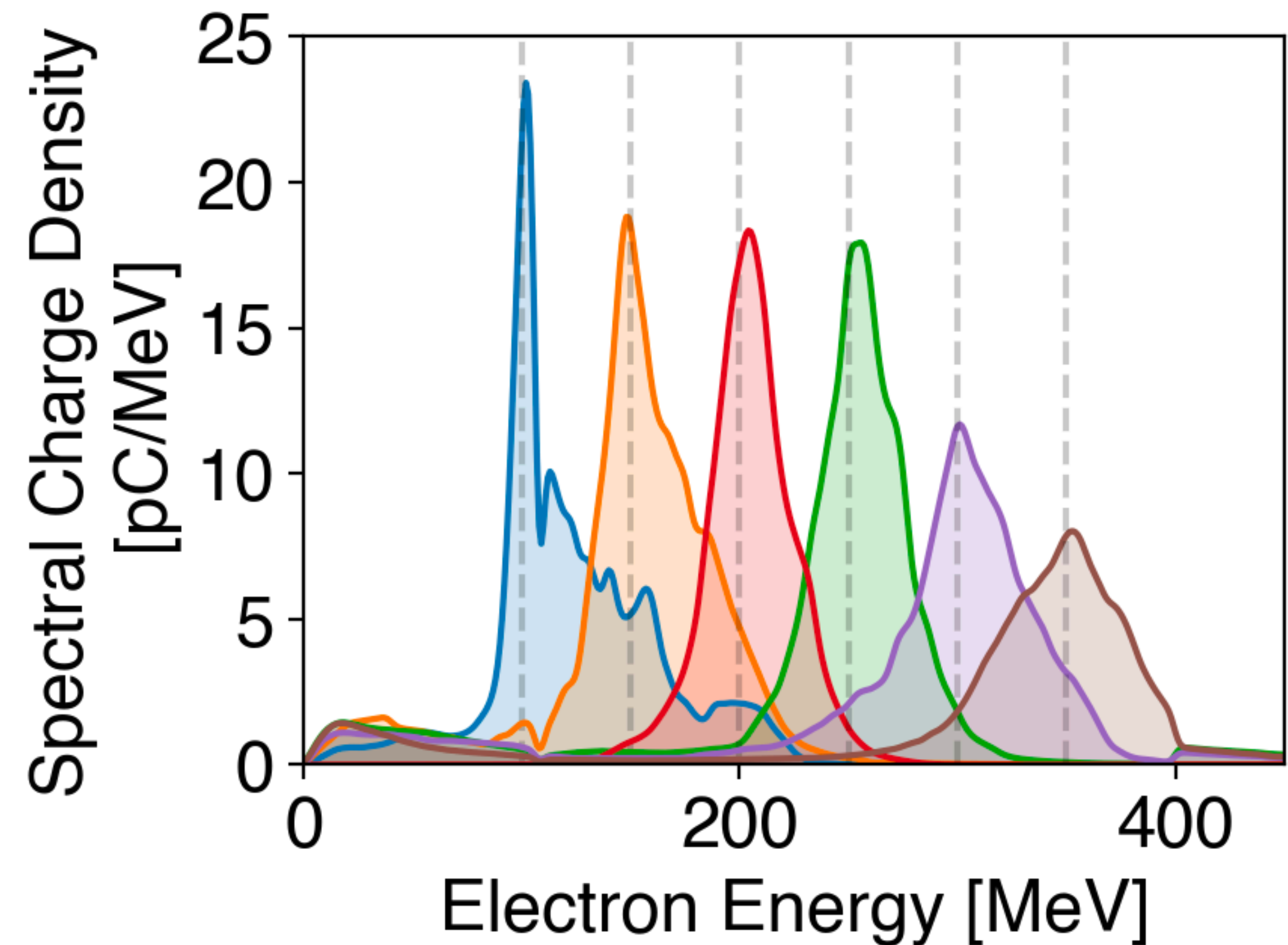
Model



Optimizing laser-plasma accelerators

Multi-objective optimization | Experiments

- **Inverse Optimization:** Given a target (e.g. energy), we can search the model for most likely positions in input space.
- Tricky part: Inverse from 1D to 8D has no unique solution. Use **Gaussian Mixture Network** to predict tuning curves.
- Tuning also works in experiments!



Further reading

Review paper

- What to do with my data?
- What are established machine learning techniques?
- Which method is suitable for my application?
- Extensive **review / tutorial paper** (30+ pages) on data-driven science and machine learning methods in laser-plasma physics

- A. Döpp et al. **Data-driven Science and Machine Learning Methods in Laser-Plasma Physics**, High Power Laser Science and Engineering **11** 55 (2023) | *arXiv:2212.00026* (2022)



arXiv:submit/4626985 [physics.plasm-ph] 30 Nov 2022

Data-driven Science and Machine Learning Methods in Laser-Plasma Physics

Andreas Döpp,^{1,*} Christoph Eberle,¹ Sunny Howard,^{1,2} Faran Irshad,¹ Jinpu Lin,¹ and Matthew Streeter³

¹Ludwig-Maximilians-Universität München, Am Coulombwall 1, 85748 Garching, Germany

²Department of Physics, Clarendon Laboratory, University of Oxford, Parks Road, Oxford OX1 3PU, United Kingdom

³Centre for Plasma Physics, Queens University Belfast, Belfast BT7 1NN, United Kingdom

Laser-plasma physics has developed rapidly over the past few decades as lasers have become both more powerful and more widely available. Early experimental and numerical research in this field was dominated by single-shot experiments with limited parameter exploration. However, recent technological improvements make it possible to gather data for hundreds or thousands of different settings in both experiments and simulations. This has sparked interest in using advanced techniques from mathematics, statistics and computer science to deal with, and benefit from, big data. At the same time, sophisticated modeling techniques also provide new ways for researchers to deal effectively with situation where still only sparse data are available. This paper aims to present an overview of relevant machine learning methods with focus on applicability to laser-plasma physics and its important sub-fields of laser-plasma acceleration and inertial confinement fusion.

CONTENTS

I. Introduction	2	C. Downhill simplex method and gradient-based algorithms	22
A. Laser-Plasma Physics	2	D. Genetic algorithms	23
B. Why data-driven techniques?	3	E. Bayesian optimization	23
C. Prediction and Feedback	14	F. Reinforcement learning	26
II. Modeling & prediction	4	V. Unsupervised Learning	27
A. Predictive models	4	A. Clustering	27
1. Spline Interpolation	5	1. Centroid-based clustering	27
2. Regression	5	2. Distribution-based clustering	27
3. Probabilistic models	5	B. Correlation analysis	27
4. Gaussian process regression	6	C. Dimensionality reduction	27
5. Decision trees and forests	7	1. Principal component analysis	27
6. Neural networks	9	2. Autoencoders	28
7. Physics-informed machine learning models	11	VI. Image analysis	29
B. Time series forecasting	12	A. Classification	29
1. Classical models	12	1. Support vector machines	29
2. State-Space Models	13	2. Convolutional neural networks	29
3. Forecasting networks	13	B. Object detection	30
C. Prediction and Feedback	14	C. Segmentation	31
III. Inverse problems	15	VII. Conclusions	31
A. Least squares solution	16	Acknowledgements	32
B. Statistical inference	16	References	32
C. Regularization	16		
D. Compressed sensing	17		
E. End-to-end deep learning methods	18		
F. Deep unrolling	19		
IV. Optimization	20		
A. General concepts	20		
1. Objective functions	20		
2. Pareto optimization	21		
B. Grid search and random search	22		

* a.doep@lmu.de

Further reading

Review paper

- What to do with my data?
- What are established machine learning techniques?
- Which method is suitable for my application?
- Extensive review / tutorial paper (30+ pages) on data-driven science and machine learning methods in laser-plasma physics

- A. Döpp et al. **Data-driven Science and Machine Learning Methods in Laser-Plasma Physics**, High Power Laser Science and Engineering **11** 55 (2023) | *arXiv:2212.00026* (2022)

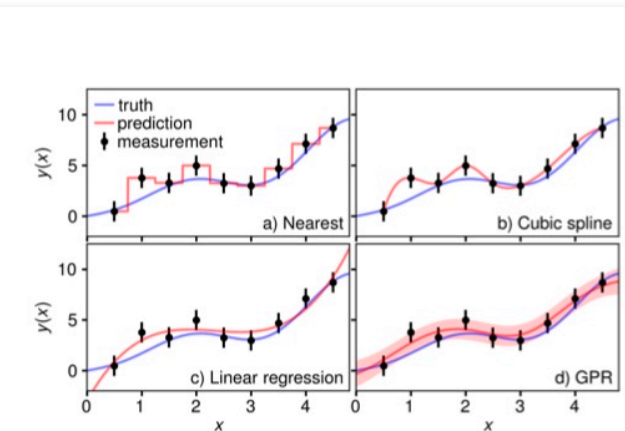


FIG. 2. Illustration of standard approaches to making predictive models in machine learning. The data was sampled from the function $y = x(1 + \sin x^2) + \epsilon$ with random Gaussian noise, ϵ , for which $\langle \epsilon^2 \rangle = 1$. The data has been fitted by a) nearest neighbour interpolation, b) cubic spline interpolator and d) Gaussian process regression (GPR).

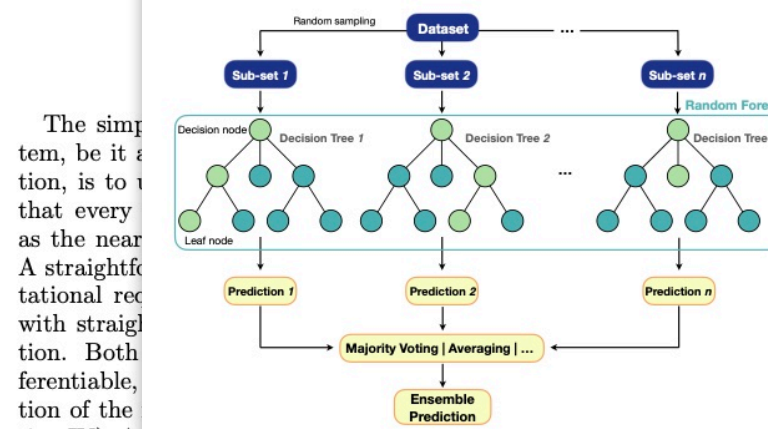


FIG. 4. Sketch of a random forest, an architecture for regression or classification consisting of multiple decision trees, whose individual predictions are combined using into an ensemble prediction e.g. via majority voting or averaging.

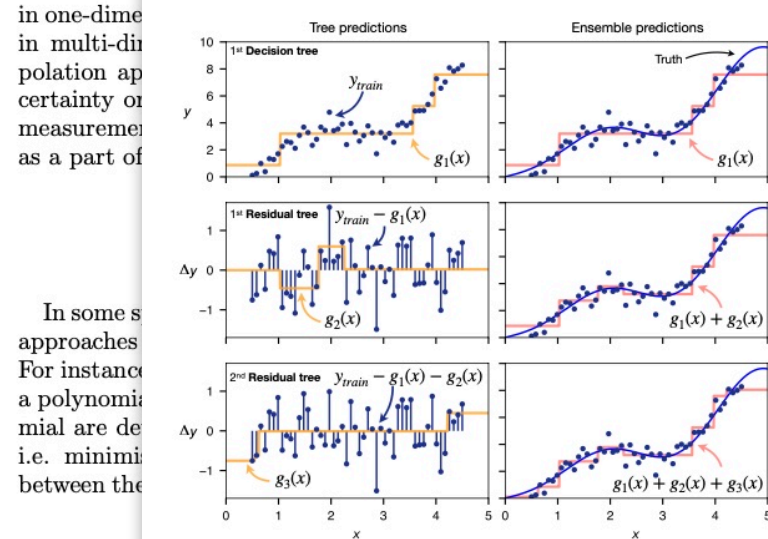


FIG. 5. Example of gradient boosting with decision trees. First, a decision tree g_1 is fitted to the data. In the next step, the residual difference between training data and the prediction of this tree is calculated and used to fit a second decision tree g_2 . This process is repeated n times, with each new tree g_n learning to correct only the remaining difference to the training data. Data in this example sampled from same function as in Fig. 2 and each tree has a maximum depth of two decision layers.

in regression settings or entropy and information gain in a classification setting. At each decision point the data set is split and subsequently the metric is re-evaluated for the resulting groups, generating the next layer of decision nodes. This process is repeated until the leaves are reached. The more layers decision layers are used, called the depth of the tree, the more complex relationships can

generally in nominal but the model does not capture actual trends or unseen data, but often motivated by physical intuition. Crucially, a limitation of analytical models is that this is one of the most important factors for the goodness of fit, the R^2 , the χ^2 .

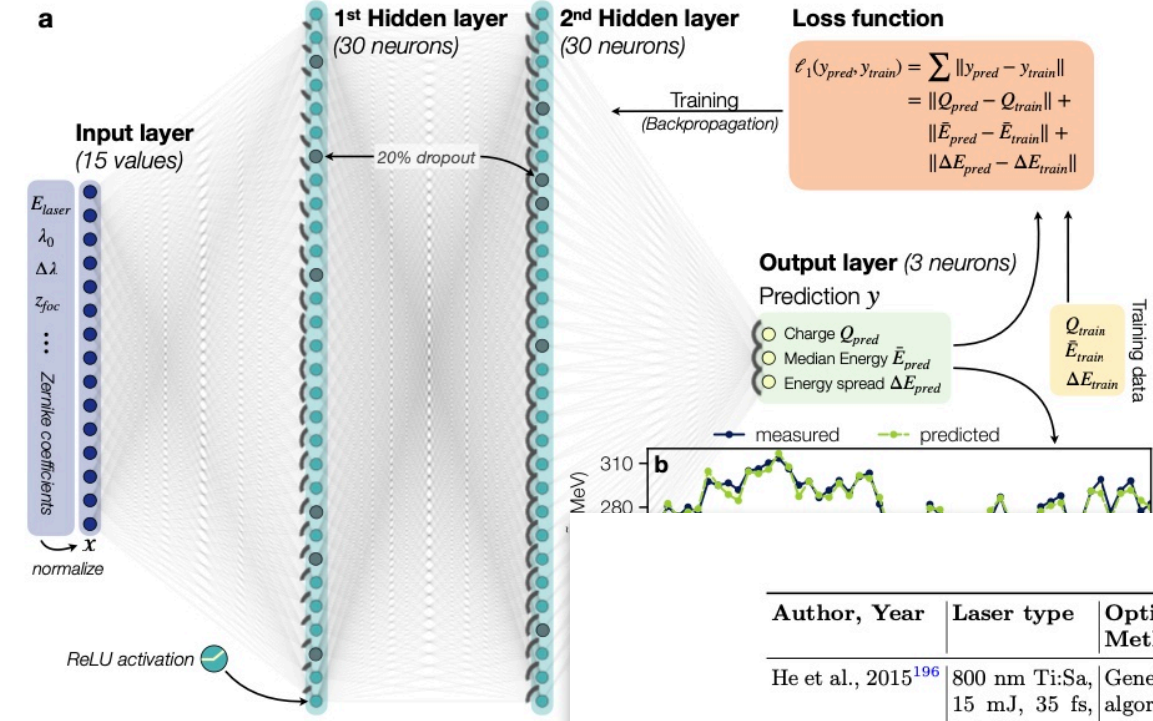


FIG. 7. Real-world example of a multilayer perceptron consists of 15 input neurons, two hidden layers with 30 neurons each, and an output layer with 3 neurons. The input is derived from parasitic laser diagnostics (laser power $\Delta\lambda$, longitudinal focus position z_{loc} and Zernike coefficients λ_0 , $\Delta\lambda$, z_{loc} , \dots). 20% of neurons drop out for regularization during training. We evaluate the accuracy of the model, in this case using the median energy (\bar{E}) and (c) measured and predicted energy spread (ΔE). The loss function is then propagated back through the network to update the weights. b-c adapted from Kirchen *et al.*²⁹

model incorporating a trained neural network was used to provide an additional computation package to the Geant4 particle physics platform. Neural networks are also trained to assist hohlraum design for ICF experiments by predicting the time evolution of the radiation temperature, in the recent work by McClarren *et al.*¹¹². In the work by Simpson *et al.*¹¹³, a fully-connected neural network with three hidden layers is constructed to assist the analysis of a x-ray spectrometer, which measures the x-rays driven by MeV electrons produced from high-power laser-solid interaction.

7. Physics-informed machine learning models

The ultimate application of machine learning for modeling physics systems would arguably be to create an “artificial intelligence physicist”, as coined by Wu and Tegmark¹¹⁴. One prominent idea at the backbone of how to train a deep neural network. An example of using a decision tree as an initializer are Deep Jointly-Informed Neural Networks (DJINN) developed by Humbird *et al.*⁹⁵, which have been widely applied in the high power laser community, especially in analyzing inertial confinement fusion datasets. The algorithm first constructs a tree or a random forest with tree depth set as a tunable hyperparameter. It then maps the tree to a neural network, or maps the forest to an ensemble of networks. The structure of the network (number of neurons and hidden layer, initial weights, etc.) reflects the structure of the tree. The neural network is then trained using back-propagation. The use of decision trees for initialization largely reduces the computational cost while maintaining comparable performance to optimized neural network architectures. The DJINN algorithm has been applied to several classification and regression tasks

Author, Year	Laser type	Optimization Method(s)	Free Parameters	Optimization goals
He et al., 2015 ¹⁹⁶	800 nm Ti:Sa, 15 mJ, 35 fs, 0.5 kHz	Genetic algorithm	deformable mirror (37 actuator voltages)	Electron angular profile, energy distribution & transverse emittance, optical pulse compression
Dann et al., 2019 ¹⁹⁷	800 nm Ti:Sa, 450 mJ, 40 fs, 5 Hz	Genetic & Nelder-Mead algorithms	deformable mirror or acousto-optic programmable dispersive filter	Electron beam charge, total charge within energy range, electron beam divergence
Shaloo et al., 2020 ¹⁹⁸	800 nm Ti:Sa, 0.245 J, 45 fs (bandwidth limit), 1 Hz	Bayesian optimization	Gas cell flow rate & length, laser dispersion ($\partial_x^2 \phi$, $\partial_z^2 \phi$), focus position	Total electron beam energy, Electron charge within acceptance angle, Betatron X-ray counts
Jalas et al., 2021 ¹⁹⁹	800 nm Ti:Sa, 2.6 J, 39 fs, 1 Hz	Bayesian optimization	Gas cell flow rates (H_2 front and back, N_2); focus position and laser energy	Spectral charge density

TABLE I. Summary of a few representative papers on machine-learning-aided optimization in the context of laser-plasma acceleration and high-power laser experiments.

distributions, in this case the electron energy distribution. While simple at first glance, these objectives need to be properly defined and there are often different ways to do so²⁰¹. In the example above, energy and bandwidth are examples for the central tendency and the statistical dispersion of the energy distribution, respectively. These can be measured using different metrics such as weighted arithmetic or truncated mean, the median, mode, percentiles and so forth for the former; and full width at half maximum, median absolute deviation, standard deviation, maximum deviation, etc. for the latter. Each of these seemingly similar measures emphasises different features of the distribution they are calculated from, which can affect the outcome of optimization tasks. Sometimes one might also want to include higher order momenta as objectives, such as the skewness, or use integrals, e.g. the total beam charge.

2. Pareto optimization

In practice, optimization problems often constitute multiple sometimes competing objectives g_i . As the objective function should only yield a single scalar value, one has to condense these objectives in a process known as *scalarization*. Scalarization can for instance take the form of a weighted product $g = \prod g_i^{\alpha_i}$ or sum $g = \sum \alpha_i g_i$ of the individual objectives g_i , with the hyperparameters α_i describing its weight. Another common scalarization technique is ϵ -constraint scalarization, where one seeks to reformulate the problem of optimizing multiple objectives into a problem of single-objective optimization conditioned on constraints. In this method the goal is to optimize one of the g_i given some bounds on the other objectives. All of these techniques introduce some explicit bias in the optimization which may not necessarily repre-

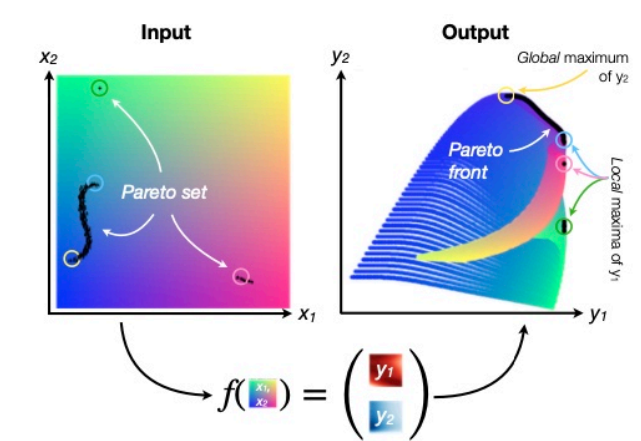


FIG. 12. Pareto front. Illustration how a multi-objective function $f(x) = y$ acts on a two-dimensional input space $x = (x_1, x_2)$ and transforms it to the objective space $y = (y_1, y_2)$ on the right. The entirety of possible input positions is uniquely color-coded on the left and the resulting position in the objective space is shown in the same color on the right. The Pareto-optimal solutions form the Pareto front, indicated on the right, whereas the corresponding set of coordinates in the input space is called the Pareto set. Note that both Pareto front and Pareto set may be continuously defined locally, but can also contain discontinuities when local maxima get involved. Adapted from Irshad *et al.*²⁰².

sent the desired outcome. Because of this, the hyperparameters of the scalarization may have to be optimized themselves by running optimizations several times. A more general approach is *Pareto optimization*, where the entire vector of individual objectives $g = (g_1, \dots, g_N)$ is optimized. To do so, instead of optimizing individual objectives, it is based on the concept of *dominance*. A

Thank you for your attention!

Dr. Andreas Döpp
Am Coulombwall 1 · 85748 Garching · Germany
a.doepp@lmu.de · www.pulse.physik.lmu.de

