

HACKATHON

AI for Particle Accelerators Hackathon

Challenge — Surrogate Models

Team Laser Focused · 16 October 2026

Benat Alberdi Esuain¹ · Sara Dastan² · Philippe Fazilleau³ · Netan Mangal⁴ ·
Marceau Pintonato⁵ · Damien Minenna³

¹Helmholtz-Zentrum Berlin ²Elettra Sincrotrone ³CEA ⁴Disobedient Geometry Labs ⁵IJCLab,
Amplitude

Meet Team Laser Focused



**Benat Alberdi
Esuain**

Helmholtz-Zentrum
Berlin

Investigator



Sara Dastan

Elettra Sincrotrone

Investigator



**Philippe
Fazilleau**

CEA

Fuzzy Logic Manager



Netan Mangal

Disobedient Geometry
Labs

AI Research Engineer



**Marceau
Pintonato**

IJCLab / Amplitude

Captain and physics expert

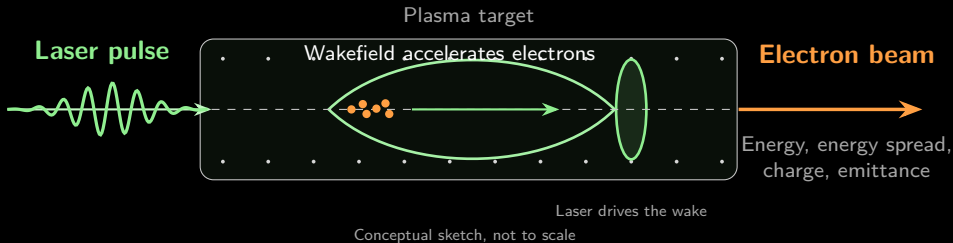


**Damien
Minenna**

CEA

Mentor

Laser-plasma acceleration — why surrogate models?



The bottleneck

High-fidelity simulations track coupled laser, plasma and beam dynamics.

Repeating them for many machine settings makes parameter scans and optimization expensive.

Our approach

Train a surrogate on simulated examples to predict beam observables quickly.

Use fast predictions to explore settings and guide the search for a desired beam.

Simulate a training set once; use the learned model for repeated predictions.

The dataset - example for first problem

Metadata		Laser + target settings (inputs)				Electron beam (outputs)		
config	injected	p_1	a_0	c_N2	x_of	E_med_MeV	dE_mad	q_end
11	True	51.97	1.329	0.0319	740.02	176.89	0.0627	1.51×10^{-10}
101	True	81.18	1.278	0.0414	914.82	117.44	0.1871	2.72×10^{-10}
1001	False	33.71	1.100	0.0613	731.21	NaN	NaN	NaN

13073 simulations, 35 columns. First three rows, selected columns (rounded). NaN = missing beam observable.

What we control

Laser: a_0, normalized laser amplitude; x_of, focal-position offset.

Gas target: p_1, gas pressure; c_N2, nitrogen dopant fraction.

config: ID; injected: injection flag.

Inverse A selection: 7 208 injected random-scan rows; 5 766 training / 1 442 validation (80/20).

What the beam does

E_med_MeV: median energy in MeV.

dE_mad: energy-spread descriptor (MAD).

q_end / i_peak: charge / peak current.

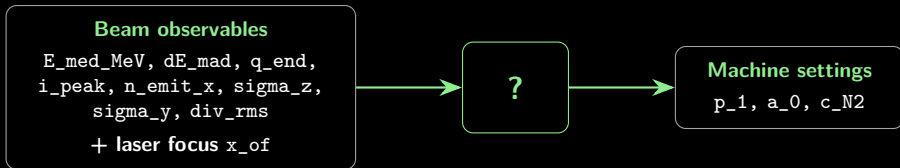
n_emit_x: normalized emittance.

sigma_z, sigma_y: RMS beam sizes.

div_rms: RMS angular divergence.

Forward model: settings $\rightarrow$ beam. Inverse A: beam + known x_of $\rightarrow$ p_1, a_0, c_N2.

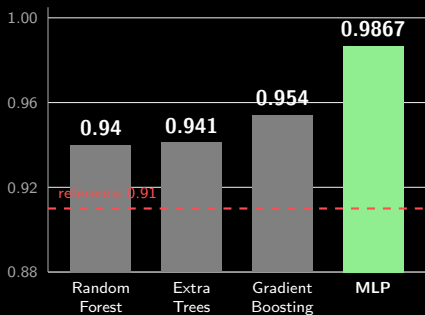
Inverse A — predict the machine backwards



- ▶ **Data:** Campaign A, injected rows (≥ 3 pC) — 7 208 simulations
- ▶ **Score:** R^2 per target and their mean
- ▶ **Bar:** reference ensemble, mean $R^2 = 0.9100$

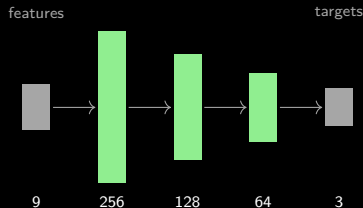
Inverse A — our solution: an MLP

Model comparison (mean R^2)



Linear baseline: 0.69

Architecture

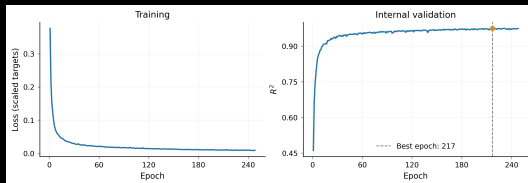


Adam, ReLU, 256-128-64

- ▶ $\log_{10}$ on `q_end`, `i_peak`, `n_emit_x`, `sigma_y`
- ▶ Standardized inputs *and* targets
- ▶ Hyperparameter optimization w/ gridsearchCV: $\text{lr} = 10^{-3}$, $\text{L2 } \alpha = 10^{-3}$, early stopping.
- ▶ 5-fold CV for architecture choice:
small 0.967 · medium 0.976 · large **0.982**

Inverse A — surrogate model results

Learning across epochs

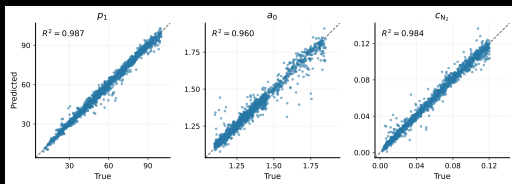


Left: training loss decreases as the model learns.

Right: internal-validation R^2 rises and then levels off.

Best validation epoch: 217.

Prediction accuracy

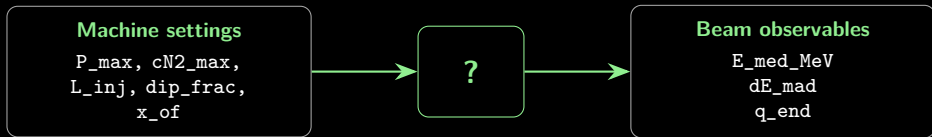


Surrogate predictions versus reference simulation results.

Each dot is one example. The dashed line marks perfect agreement.

Points close to the line indicate accurate predictions.

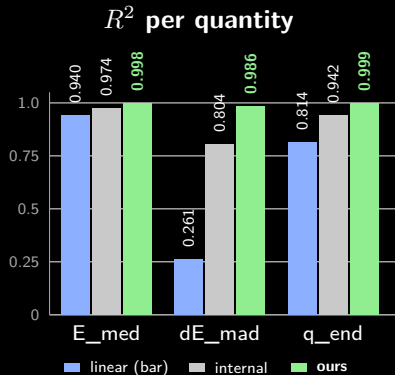
Direct B — read the machine forwards



forward model = the inverse of exercise 1

- ▶ **Data:** Campaign B, injected rows (2882 simulations)
- ▶ **Deliverable:** submission_direct.csv (to be scored in 357 hidden configs)
- ▶ **Score:** R^2 per quantity
- ▶ **Bar:** linear baseline 0.94 / 0.26 / 0.81
the room is in the energy spread
- ▶ **Stretch target:** internal emulator 0.974 / 0.804 / 0.942

Direct B — our solution



RF 0.85 · XGBoost 0.92 · **MLP 0.994**

MLP: mean $R^2 = 0.9943$ — energy spread 0.26 → 0.986

Physics-informed features

- ▶ $n_e, n_{N_2}, L_{\text{plasma}} = L_{\text{inj}} + 3.2$
- ▶ $E_0 L \propto \sqrt{n_e} L_{\text{plasma}}, \text{ dip} \times P_{\text{max}}$
- ▶ **Energy gain integrated along the plasma** ($\eta = 0.27$)

MLP 256-512-64

- ▶ ReLU, Adam ($\text{lr} = 10^{-3}$), **Huber loss**
- ▶ Cosine annealing, 300 epochs, batch 32
- ▶ 3-seed ensemble, final fit on all rows
- ▶ 5-fold CV: **0.994 ± 0.0007**

Trajectory — the bunch along the accelerator



- **Physics features:** integrated energy gain, n_e , n_{N_2} , dz , in_plasma
- **Training:** Huber loss, Adam, cosine schedule, 3-seed ensemble
- **Data:** 2855 configurations; the submission model uses all of them

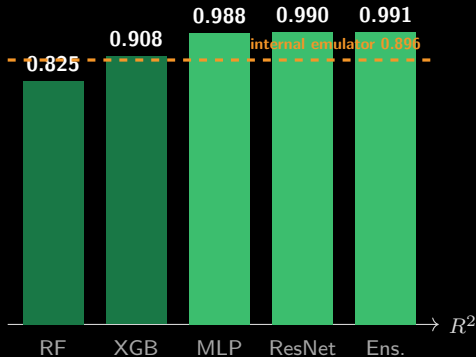
Trajectory — the physics we put into the inputs

Rebuilt from the five knobs, so the hidden test can use them

feature	formula	what it encodes
dz	$z - (L_{inj} + 3.2)$	distance to the plasma end
n_{N2}	$P_{\max} c_{N_2}$	nitrogen K-shell electrons: the charge reservoir
$gain(z)$	$\eta e \int_0^z E_0(n_e(z')) dz'$	energy gained riding a fraction $\eta = 0.27$ of the wave-breaking field

The linear model nearly doubles when it is given the physics, from 0.35 to 0.62, and its energy score goes from negative to 0.67.

Trajectory — results



Our score
0.9849

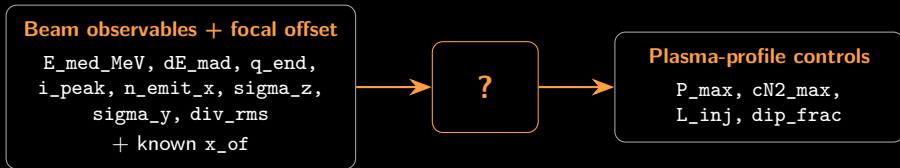
mean R^2 , Trajectory (Easy)

Robustness

- ▶ 5-fold CV: 0.9897 ± 0.0009
- ▶ Drift zone: 0.990, no degradation

The hidden test is denser (38% above the OOD cut vs 20% of the public set): a random holdout reads optimistic. See the out-of-distribution slides (gap 0.030).

Inverse B — infer the machine settings

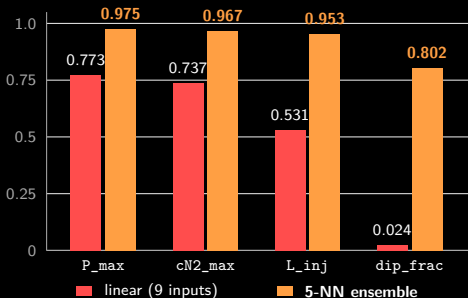


Inverse surrogate: 9 measured / known inputs, 4 predicted controls

- ▶ **Data:** Campaign B, 2882 successfully injected simulations
- ▶ **Split:** 2305 training / 577 validation rows (80/20)
- ▶ **Score:** R^2 per control, plus their unweighted mean
- ▶ **Linear baseline:** 0.773 / 0.737 / 0.531 / 0.024
Order: `P_max` / `cN2_max` / `L_inj` / `dip_frac`
- ▶ **The hardest control is `dip_frac`.**

Inverse B — our solution

Validation R^2 per control



Mean R^2 : 0.5161 $\rightarrow$ **0.9241**

dip_frac: $R^2 = 0.8017$, MAE = 0.0389

Preprocessing

Add two ratios to the nine inputs:

$$\frac{q_{\text{end}}}{i_{\text{peak}}}, \quad \frac{dE_{\text{mad}}}{E_{\text{med_MeV}}}$$

Log-transform $n_{\text{emit_x}}$, dE_{mad} , div_rms , sigma_z ; then standardize inputs and targets.

A simple NN with five layers: input layer + 3 hidden layers + output layer

Each layer: $11 \rightarrow 256 \rightarrow 128 \rightarrow 64 \rightarrow 4$.

tanh activation + Adam, learning rate 5×10^{-4} .

L2 = 10^{-3} regularization penalty; up to 700 epochs.

Early stopping on an internal 15% split.

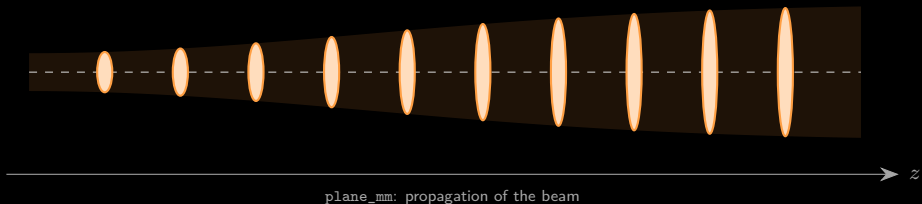
Five seeds; average in original target units.

Same 577 validation rows for both models.

Moments — the bunch at each plane



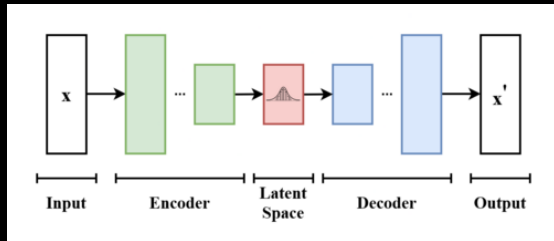
one stored plane = one bunch snapshot to model



- ▶ **Data:** electron bunch moments at each plane stored along the simulation
- ▶ **Score:** R^2 per column of the 13 targets
- ▶ **The goal:** a model of the bunch at each plane.

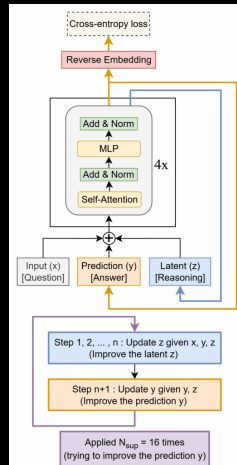
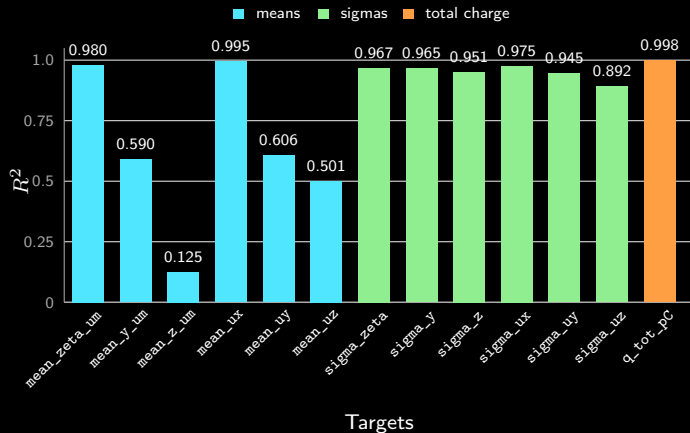
Variational Auto Encoder (VAE)

This model yield very good results :
Mean R^2 : 0.87, before finding a **leak** in
the **data pipeline** ...



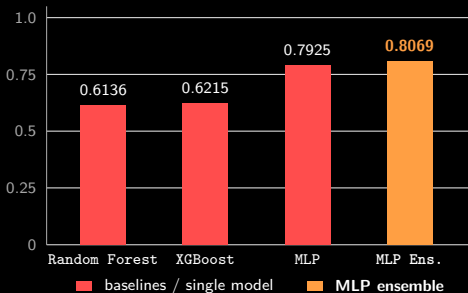
Tiny Reasoning Model (TRM)

This architecture allows for reasoning, but is quite big and hard to train.



Model comparison — our solution

Mean validation R^2 per model



Mean R^2 : 0.6136 (RF) → **0.8069**

MLP ensemble: +0.0143 over single MLP, +0.1853 over
XGBoost

For this task we tried a variety of models, and we found out the best model to be an ensemble of MLP.

In the final evaluation test it scored :
 $R^2 = 0.7574$

Out of distribution in density



- ▶ **Signature:** trajectory task (unchanged)
- ▶ **Train only** on $P_{\max} < 6\,528.6$ Pa; the 578 public configs above the cut are dropped
- ▶ **Scored:** 221 hidden in-range configs and 136 above the cut
- ▶ **Deliverable:** the gap between the two scores, honestly measured

“A model tuned until the gap looks small has usually learned the cut.”

Out of distribution — the gap



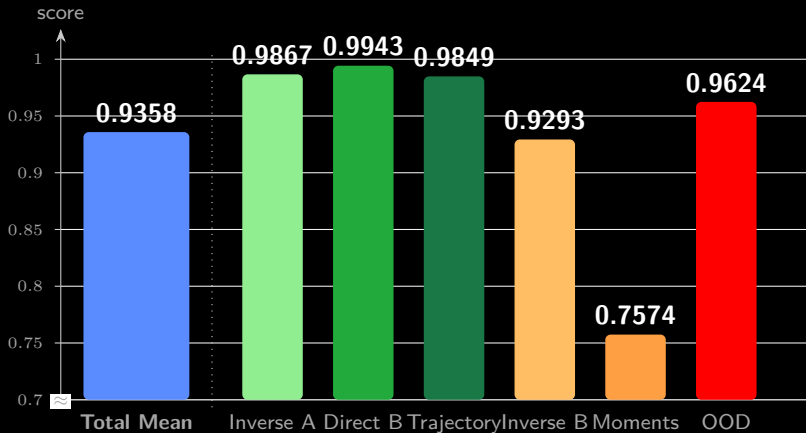
Protocol

- ▶ 578 public configs above the cut: **never trained on**
- ▶ Early stopping on in-range data only

Why the MLP extrapolates

- ▶ Trees are piecewise constant: they cannot leave the trained range
- ▶ Physics features: integrated energy gain, n_e , n_{N_2}
- ▶ Predicting E/gain helps: $0.984 \rightarrow 0.987$ above cut

Overall Results



Long-term vision

Architectures to try

- ▶ **Neural ODE / state-space:** $\frac{d\mathbf{m}}{dz} = f_{\theta}(\mathbf{m}, \text{settings})$, continuous in `plane_mm`
- ▶ **Neural operators** (DeepONet, FNO): `settings` $\rightarrow$ the whole profile along z

Gaussian statistics

- ▶ **Gaussian processes** (sparse GP): uncertainty for free, good with few data
- ▶ **Deep ensembles / VAE**: confidence on each prediction

Experimental data

- ▶ **Real data:** see of the models performs on real data

Now

MLP: `settings` + `plane`

Next

evolution along z :
TRM, Neural ODE, operators

Vision

full Gaussian $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$
with uncertainty

Thank you.

Questions / ideas / pull requests?

AI for Particle Accelerators Hackathon

Benat Alberdi Esuain · Sara Dastan · Philippe Fazilleau · Netan Mangal · Marceau
Pintonato · Damien Minenna