Transverse phase space reconstruction methods with traditional and machine learning techniques

Emre Cosgun
JPARC/KEK

2026/04/10

가속기 빔진단 협력 workshop
KOMAC, KAERI, Gyeongju, South Korea

Table of Content

Phase Space

Transverse beam dynamics provides the framework for understanding how charged particles move through focusing lattices and how beam quality is quantified.

- A particle cannot be fully described by its position alone.
- Two particles may have the same position and yet move differently because their momenta are different.
- For this reason, beam dynamics is formulated in phase space.
- **The most complete single-particle description is the six-dimensional phase space,**

$$(x,\, p_x,\, y,\, p_y,\, z,\, p_z)$$

x, y, z → spatial coordinates

p_x, p_y, p_z → momentum components

- The six-dimensional dynamics may be decomposed into transverse and longitudinal parts:

$$(x,\, p_x,\, y,\, p_y,\, z,\, p_z) \longrightarrow (x,\, p_x,\, y,\, p_y) + (z,\, p_z)\,.$$

Transverse motion

- In a linear focusing lattice, the transverse motion of a particle is governed by the Hill equation;

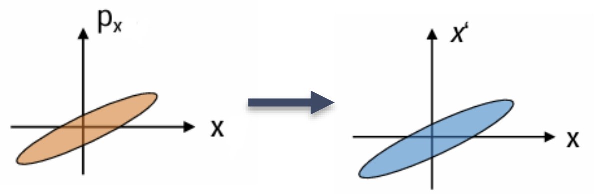
$$x''(s) + K(s)x(s) = 0,$$

$x(s)$ is the transverse displacement,
(s) is the longitudinal coordinate along the reference orbit,
$K(s)$ is the focusing function determined by the lattice.

- The solution of the Hill equation is conventionally written in the form:

$$x(s) = \sqrt{\varepsilon\,\beta(s)}\,\cos\big(\psi(s) + \phi\big),$$

- This expression shows that the particle executes a transverse oscillation while moving along the beamline. The oscillation amplitude is not constant. Instead, it depends on the product $\varepsilon\beta(s)$.
- Canonical phase space uses variables such as → $(\boldsymbol{x}, \boldsymbol{p_x})$
- In accelerator physics, it is often more practical to use *trace space.* →$(\boldsymbol{x}, \boldsymbol{x'})$



- The transverse momentum is small compared with the longitudinal momentum, one may write:

$$x' = \frac{dx}{ds}.$$

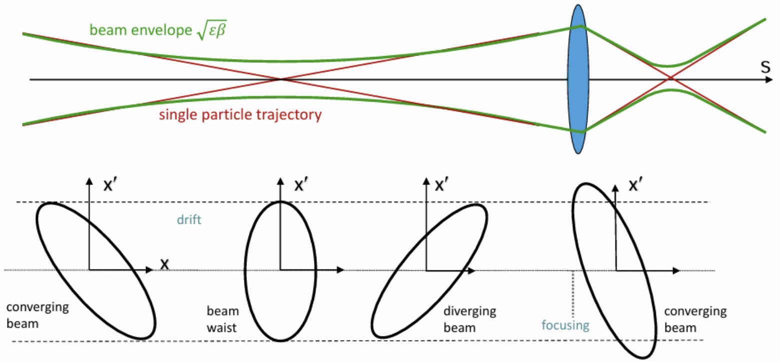
Single particle motion

- Once both (x) and (x') are considered, the motion of a single particle can be represented geometrically in trace space.
- Transverse motion satisfies the invariant quadratic form:

$$\gamma(s)x^2 + 2\alpha(s)xx' + \beta(s)x'^2 = \varepsilon,$$

$$\alpha(s) = -\frac{1}{2}\beta'(s), \qquad \gamma(s) = \frac{1+\alpha^2(s)}{\beta(s)}.$$

This is called the Courant--Snyder ellipse.



- β controls the width of the ellipse in position,
- γ controls the extent in angle,
- α controls the tilt of the ellipse.

- If $\alpha = 0$, the ellipse is upright, corresponding to a beam waist. If $\alpha \neq 0$, position and angle are correlated, and the ellipse is tilted.

A real beam

- A real beam is not a single particle moving on a single ellipse. It is an ensemble of many particles.
- In trace space, the beam therefore appears as a distribution of points rather than as a single closed curve. For a real beam, it is usually more meaningful to work with second moments.

$$\Sigma = \begin{pmatrix} \langle x^2 \rangle & \langle x x' \rangle \\ \langle x x' \rangle & \langle x'^2 \rangle \end{pmatrix}.$$

- From this matrix, the rms emittance is defined by,

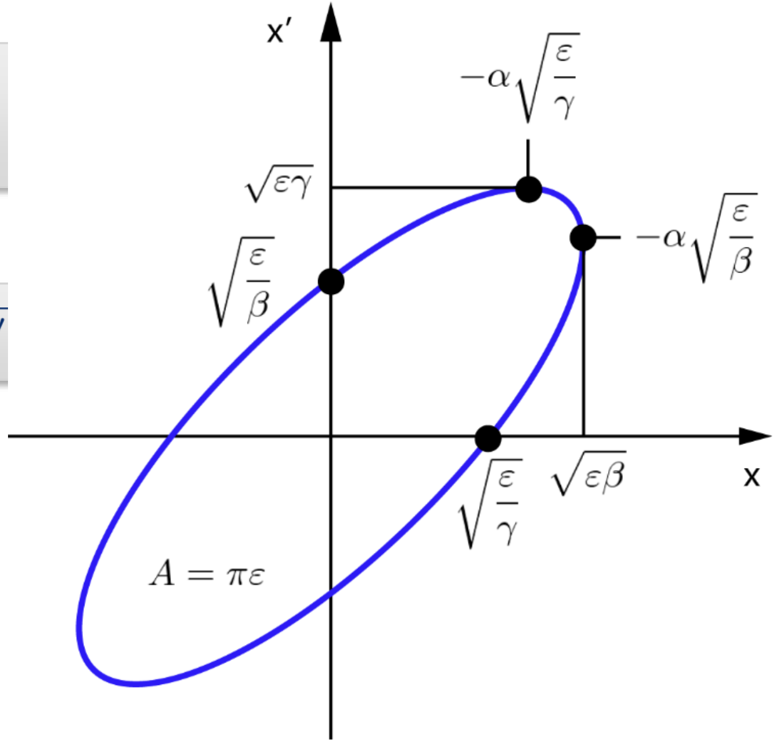
$$\varepsilon_{\rm rms} = \sqrt{\langle x^2 \rangle \langle x'^2 \rangle - \langle x x'}$$

- The Twiss parameters of the beam:

$$\beta = \frac{\langle x^2 \rangle}{\varepsilon_{\rm rms}},$$

$$\alpha = -\frac{\langle x x' \rangle}{\varepsilon_{\rm rms}},$$

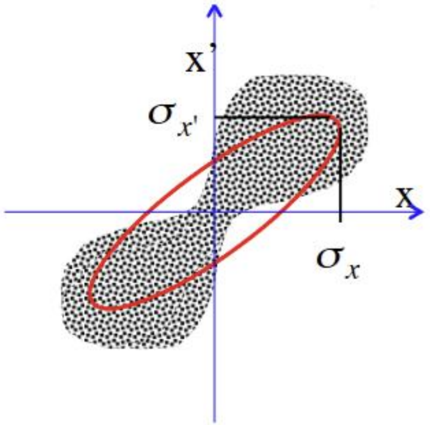
$$\gamma = \frac{\langle x'^2 \rangle}{\varepsilon_{\rm rms}}.$$



Statistical definition of emittance

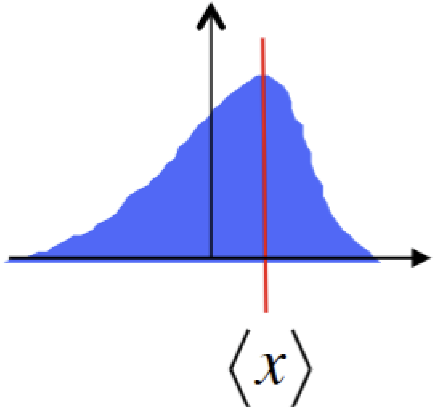
- Real beams can be non-Gaussian, distorted, filamented, or halo-dominated.
- The statistical definition based on second moments remains meaningful and reproducible, even when the beam shape is complicated.
- Let $f_x(x, x')$ denote the normalized beam distribution in the horizontal trace space (x, x').

$$\iint_{-\infty}^{+\infty} f_x(x,x')\,dx\,dx' = 1.$$



- **The first-order** moments define the beam centroid;

$$\langle x\rangle = \iint_{-\infty}^{+\infty} x\, f_x(x,x')\,dx\,dx',$$
$$\langle x'\rangle = \iint_{-\infty}^{+\infty} x'\, f_x(x,x')\,dx\,dx'.$$



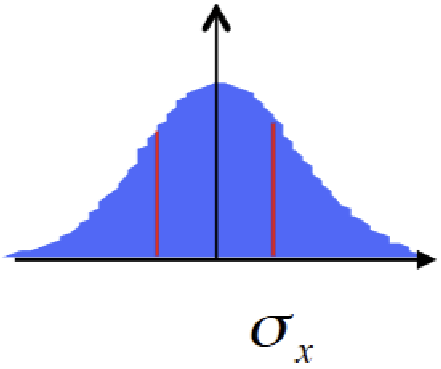
Statistical definition of emittance

» **The second moments** measure the spread of the beam:

$$\sigma_x^2 = \left\langle (x - \langle x \rangle)^2 \right\rangle = \iint_{-\infty}^{+\infty} (x - \langle x \rangle)^2 f_x(x, x')\, dx\, dx'.$$

$$\sigma_{x'}^2 = \left\langle (x' - \langle x' \rangle)^2 \right\rangle = \iint_{-\infty}^{+\infty} (x' - \langle x' \rangle)^2 f_x(x, x')\, dx\, dx'.$$

$$\sigma_{xx'} = \langle (x - \langle x \rangle)(x' - \langle x' \rangle) \rangle = \iint_{-\infty}^{+\infty} (x - \langle x \rangle)(x' - \langle x' \rangle) f_x(x, x')\, dx\, dx'.$$



» The second central moments are collected into the beam matrix

$$\Sigma = \begin{pmatrix} \sigma_x^2 & \sigma_{xx'} \\ \sigma_{xx'} & \sigma_{x'}^2 \end{pmatrix}.$$

» Rms emittance;

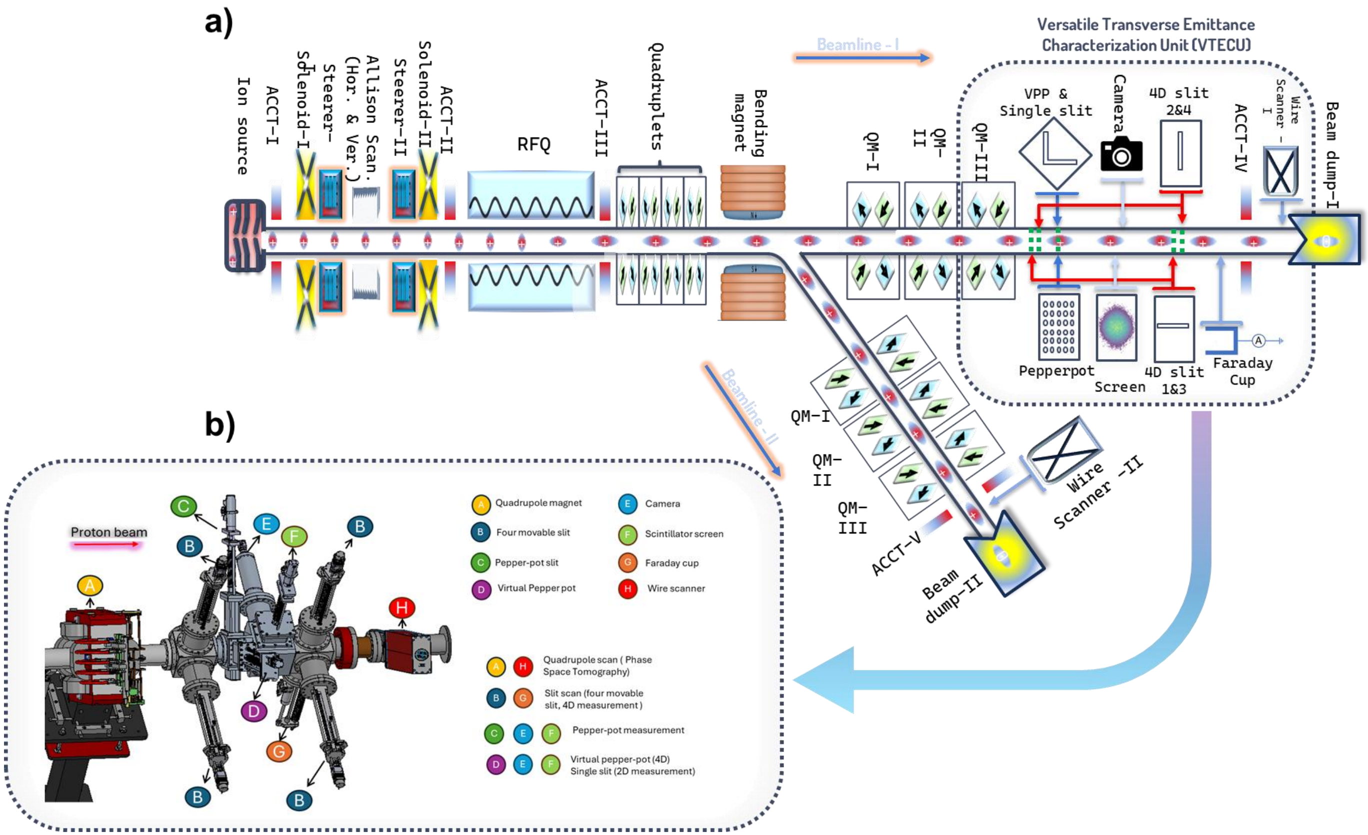
$$\varepsilon_{\mathrm{rms}} = \sqrt{\det \Sigma} = \sqrt{\sigma_x^2 \sigma_{x'}^2 - \sigma_{xx'}^2}.$$

» The beam matrix may be written in the Twiss form:

$$\Sigma = \varepsilon_{\mathrm{rms}} \begin{pmatrix} \beta & -\alpha \\ -\alpha & \gamma \end{pmatrix}. \qquad \beta = \frac{\sigma_x^2}{\varepsilon_{\mathrm{rms}}}, \qquad \alpha = -\frac{\sigma_{xx'}}{\varepsilon_{\mathrm{rms}}}, \qquad \gamma = \frac{\sigma_{x'}^2}{\varepsilon_{\mathrm{rms}}}.$$

Table of Content

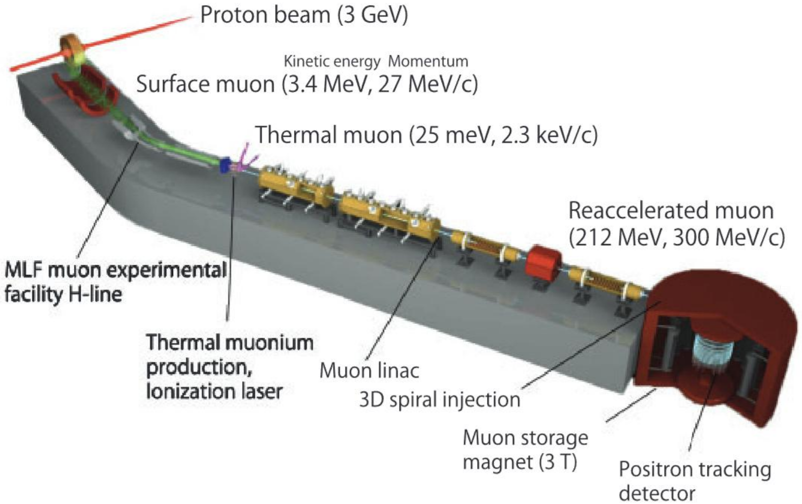
Facility-I KOMAC Beam Test Stand



Transverse phase space characterization methods;

- Allison Scanner – 2D
- Quadrupole Scan – 2D
- Tomography – 2D
- Single Slit – 2D
- 4 slit method – 4D
- Pepper-pot -4D
- Virtual Pepper-pot – 4D

Facility-II J-PARC Muon Facility



New approach to measure the muon magnetic moment anomaly $a_\mu = (g - 2)/2$ and the muon electric dipole moment (EDM) d_μ at the J-PARC muon facility.

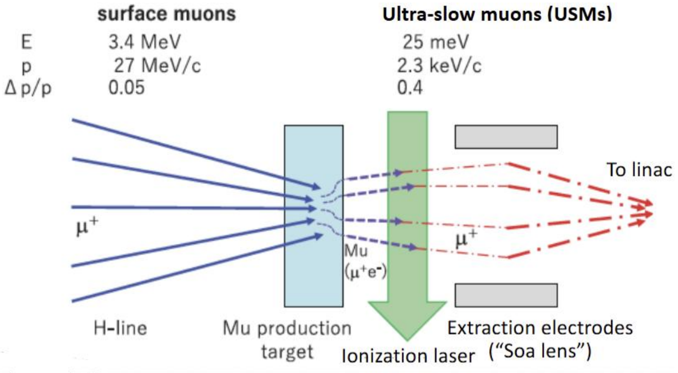
i. A 3 GeV proton beam hits a graphite target at J-PARC. This produces pions, and some of those pions stop near the surface and decay into **surface muons**.
ii. The surface muons are sent into **silica aerogel**. Inside the aerogel, many muons form **muonium.**
iii. Muon Linac together bring the thermal muon beam up to **300 MeV/c** momentum.
iv. Inject the beam into a compact **3 T** storage solenoid.

- The **oscillation in positron count versus time** gives the anomalous precession frequency ω_a which leads to $\boldsymbol{a_\mu}$.
- The **up-down asymmetry** of positrons gives sensitivity to the EDM.

Surface muon → **Thermal muon** → **Acceleration** → **Transport & Injection** → **Storage** → **Detection** → **Reconstruction**

Facility-II J-PARC Muon Facility - Muon Linac

- Generation of the "ultra-slow muons" (USMs).
- Acceleration of the USMs. 4-stage linac, from the p-like acceleration to the e-like one.



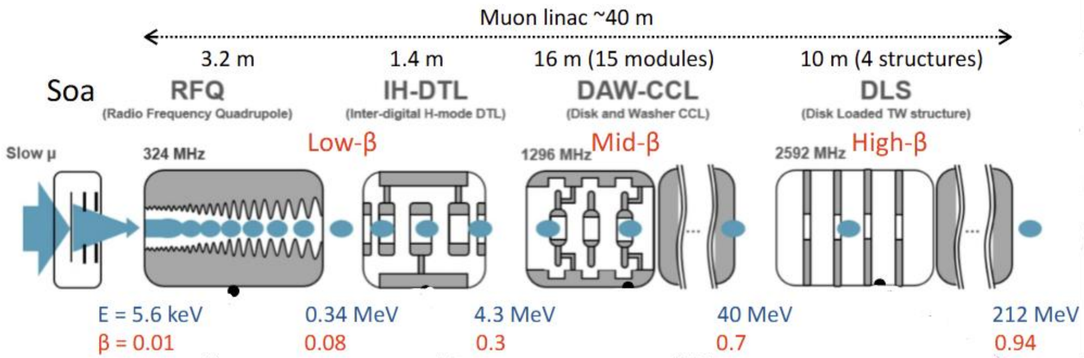


Table of Content

Allison type emittance meter

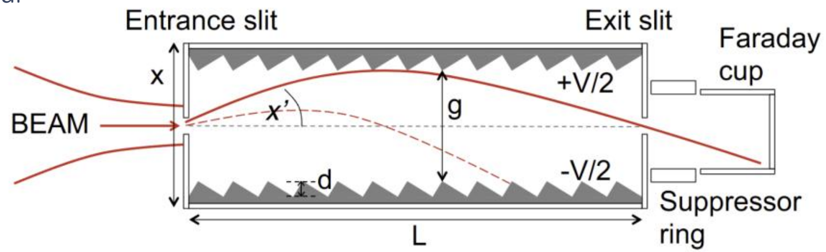
- A very narrow entrance slit selects a thin slice of the beam at one transverse position.
- The particles then pass through a short region with an electrostatic deflection field between plates.
- Only particles with a particular transverse angle are deflected so they can pass through an exit slit.
- A collector measures the transmitted current.
- The entrance slit defines the transverse position (x_s is the scanner position.): $x = x_s$
- Transverse angle: $$x' = \frac{\Delta v_x}{v_z} = \frac{qV_dL}{mg\,v_z^2} = \frac{qV_dL}{2gW}$$
  deflection voltage: V_d
  the plate gap: g
  the plate length: L
  kinetic energy: W
- Selected angle is proportional to the deflection voltage: $x' = KV_d$ where $K = \frac{qL}{2gW}$
- The collector current at scanner position x_s and voltage V_d is proportional to the beam phase-space density:
  $$I_c(x_s, V_d) \propto f(x, x') \qquad x = x_s, \qquad x' = K_{\mathrm{cal}} V_d$$
- Thus the reconstructed phase-space map is:
  $$f(x, x') \propto I_c\left(x, \frac{x'}{K_{\mathrm{cal}}}\right)$$

$$\sigma_{xx} = \langle (x - \langle x \rangle)^2 \rangle$$

$$\sigma_{x'x'} = \langle (x' - \langle x' \rangle)^2 \rangle$$

$$\sigma_{xx'} = \langle (x - \langle x \rangle)(x' - \langle x' \rangle) \rangle$$

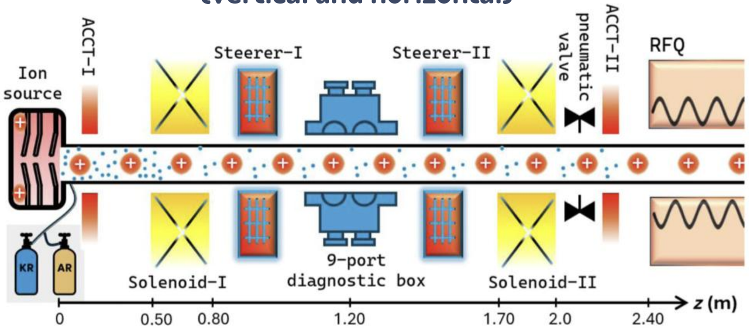
$$\varepsilon_{x,\mathrm{rms}} = \sqrt{\sigma_{xx}\sigma_{x'x'} - \sigma_{xx'}^2}$$



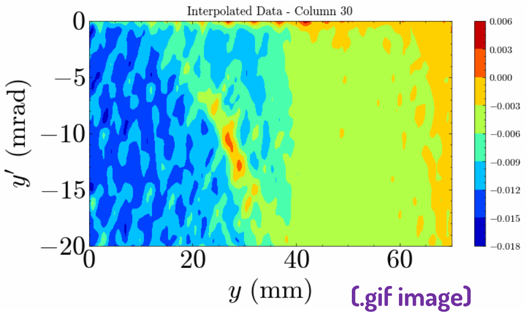
From: *Alvarez, M., Gaynier, J., Prost, L., Scarpine, V. and Shemyakin, A., 2016. Characterisation of the PXIE Allison-type emittance scanner. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, 815, pp.7-17.*

Allison type emittance meter – Example

KOMAC BTS LEBT line has two Allison scanner (vertical and horizontal)

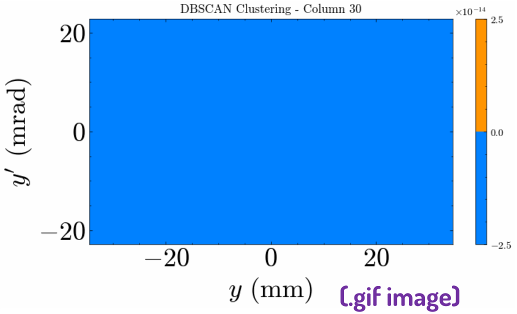


Original data



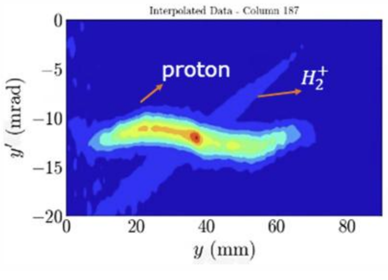
(.gif image)

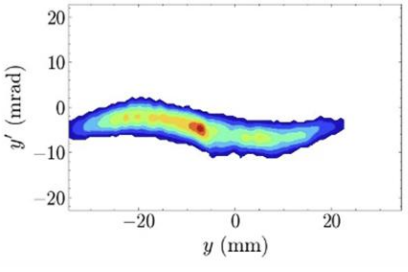
Clustered data



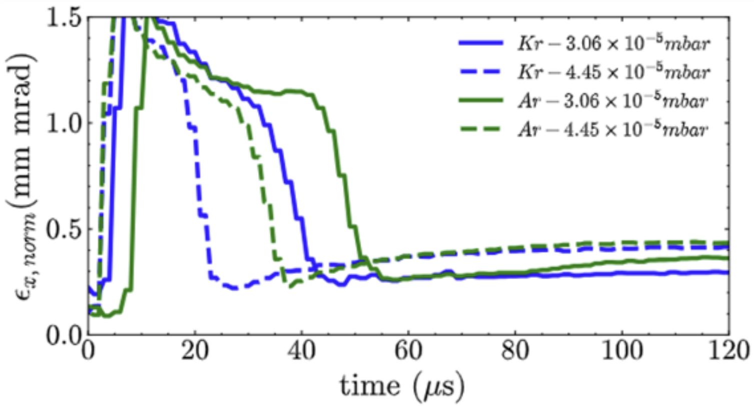
(.gif image)

- Density-based spatial clustering algorithm (DBSCAN) to the raw phase-space data to eliminate noise and outliers.

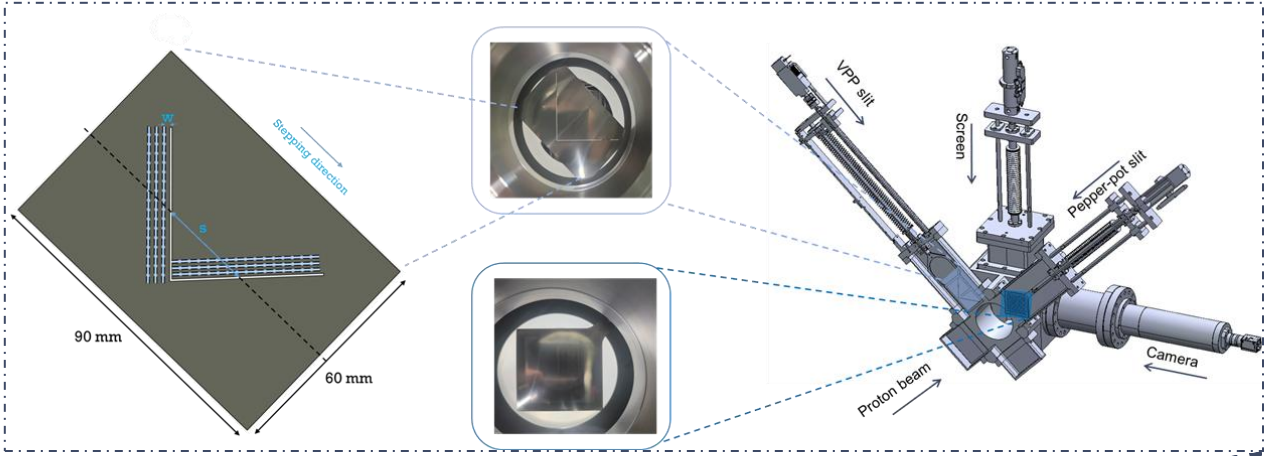


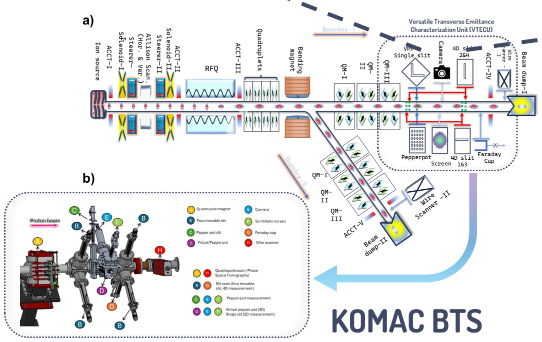


Emittance vs. time



Single slit – Pepper Pot – VPP





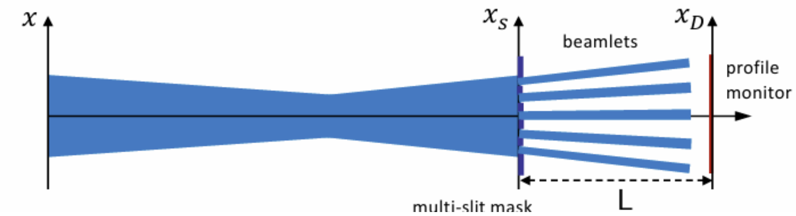
Parameter	Value	Description
Slit width (d)	0.20 mm	Width of each slit opening
Slit spacing (w)	1.00 mm	Step length or motor action distance (optional)
Perpendicular slit spacing (s)	56 mm	The perpendicular center position distance of slits
Slit length	60 mm	Total horizontal length of VPP slit pattern
Slit height	90 mm	Total vertical height of VPP slit pattern
Slit thickness (L_{th})	0.10 mm	Thickness of the VPP plate
Slit-to-screen distance (L_d)	$\sim$110 mm	Distance from slit to scintillator screen

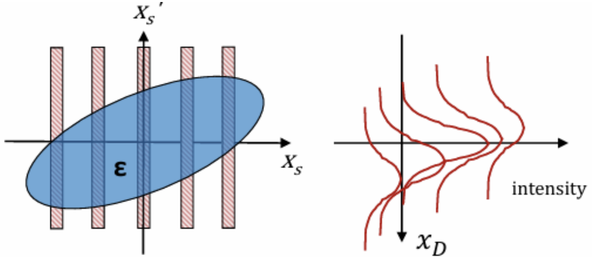
Slit scan method

- Measure the 2D transverse phase-space distribution
- Useful for low-energy beams, where space-charge effects are often significant.
- A narrow slit selects particles from a small transverse position interval.
- These selected particles drift freely over a known distance.
- Their positions at a downstream detector are measured.
- The downstream position distribution is converted into an angular distribution.
- Repeating this for many slit positions reconstructs the full transverse phase space.
- The transport matrix of a drift of length (L) is;

$$\begin{pmatrix} x_D \\ x'_D \end{pmatrix} = \begin{pmatrix} 1 & L \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_s \\ x'_s \end{pmatrix}.$$

$$x_D = x_s + L x'_s$$
$$x'_D = x'_s.$$



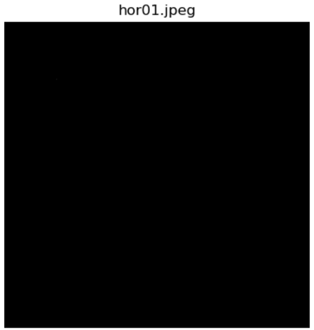


- If the x_s, L is known, and the downstream detector position (x_D) is measured, then the initial divergence (x'_s) of the particles passing through the slit can be determined.

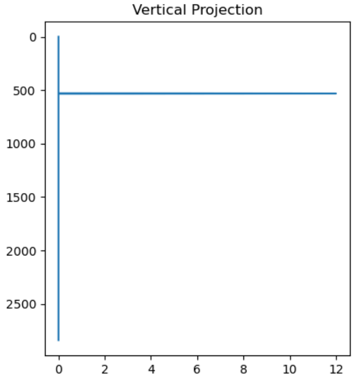
$$x'_s = \frac{x_D - x_s}{L}.$$

Slit scan method – Example

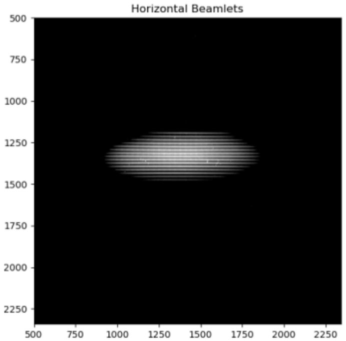
1. **Load and Preprocess Image**
   (a) Read the beam profile image and convert it to grayscale.
   (b) Filter the image and eliminate the beam spots (in this case noise).
   (c) Apply a median filter to remove noise while preserving beamlet structures.
2. **Extract Beamlet Projections**
   (a) Compute the horizontal and vertical projections by summing intensity along the horizontal and vertical axis.
   (b) Detect beamlet peaks using Gaussian fitting.
   (c) Match the peak beamlet locations to the slit center locations.
3. **Convert Pixel Positions to Physical Units**
   (a) Define conversion factors to transform pixel positions into millimeters.
   (b) Compute beam divergence angles from slit positions.
4. **Compute RMS Emittance**
   (a) Compute mean beamlet positions and divergences.
   (b) Calculate RMS values for beam size and angular spread.
   (c) Compute RMS emittance and Twiss parameters.

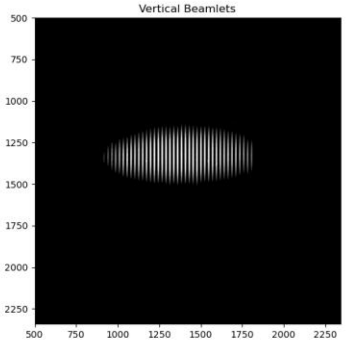


(.gif image)



(.gif image)





Emittance analysis

» Rigid formula for slits and pepper-pot emittance measurement is derived by **M. Zhang** *(Emittance formula for slits and pepper-pot measurement. No. FNAL-TM--1988. 1996.)*

$$\varepsilon^2_{x,\mathrm{rms}} = \frac{1}{N^2}\left\{\left[\sum_{j=1}^{p} n_j(x_j-\bar{x})^2\right]\left[\sum_{j=1}^{p}\left(n_j\sigma^2_{x'_j}+n_j(x'_j-\bar{x}')^2\right)\right]-\left[\sum_{j=1}^{p} n_j x_j x'_j - N\bar{x}\bar{x}'\right]\right\}$$

- $N = \sum_{j=1}^{p} n_j$ is the total number of protons passing through all holes.
- n_j is the number of protons passing through the j^{th} hole.
- x_j is the position of the j^{th} hole.
- The mean hole position is given by:

$$\bar{x} = \frac{\sum_{j=1}^{p} n_j x_j}{N}$$

- The divergence of the j^{th} beamlet is:

$$x'_j = \frac{X_j - x_j}{L_{md}}$$

- X_j is the peak position of the detected j^{th} beamlet.
- $\bar{x}'$ is the mean divergence of all the beamlets.
- L_{md} is the distance from the mask to the detector.
- p is the total number of beamlets.

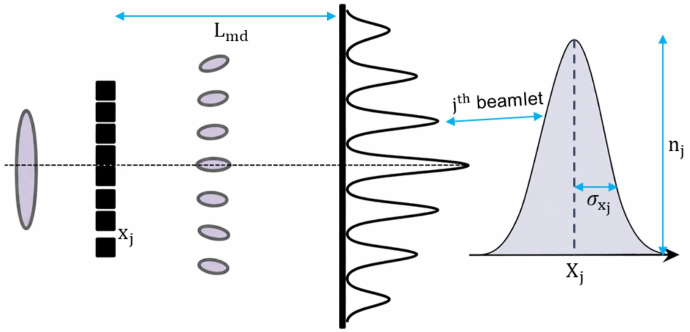
- The squared r.m.s. divergence of the j^{th} beamlet for the **Pepper-Pot** method is:

$$\sigma'^2_{x'_j} = \frac{\sigma^2_{x'_j}}{L^2_{md}}.$$

- For the **Virtual Pepper-Pot** and **Single-Slit** methods, it is given by:

$$\sigma'^2_{x'_j} = \frac{\sigma^2_{x'_j} - \left(\frac{d}{\sqrt{12}}\right)^2}{L^2_{md}}.$$

where d is finite slit width.



4D beam matrix

$$\Sigma^{4D} \equiv \begin{pmatrix} \langle x^2\rangle & \langle xx'\rangle & \langle xy\rangle & \langle xy'\rangle \\ \langle xx'\rangle & \langle x'^2\rangle & \langle x'y\rangle & \langle x'y'\rangle \\ \langle xy\rangle & \langle x'y\rangle & \langle y^2\rangle & \langle yy'\rangle \\ \langle xy'\rangle & \langle x'y'\rangle & \langle yy'\rangle & \langle y'^2\rangle \end{pmatrix} = \begin{pmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{xy}^T & \Sigma_{yy} \end{pmatrix}$$

The **4D beam matrix** can be expressed as

$$M^T \Sigma^{4D} M = \begin{pmatrix} \epsilon_1 & 0 & 0 & 0 \\ 0 & \epsilon_1 & 0 & 0 \\ 0 & 0 & \epsilon_2 & 0 \\ 0 & 0 & 0 & \epsilon_2 \end{pmatrix},$$

where ϵ_1 and ϵ_2 are the **intrinsic eigen-emittances** and M is a **symplectic matrix**. ϵ_1 and ϵ_2 can be calculated:

$$\epsilon_1 = \frac{1}{2}\sqrt{-\mathrm{tr}[(\Sigma^{4D}J)^2] + \sqrt{\mathrm{tr}^2[(\Sigma^{4D}J)^2] - 16\det(\Sigma^{4D})}},$$

$$\epsilon_2 = \frac{1}{2}\sqrt{-\mathrm{tr}[(\Sigma^{4D}J)^2] - \sqrt{\mathrm{tr}^2[(\Sigma^{4D}J)^2] - 16\det(\Sigma^{4D})}}.$$

where

$$J \equiv \begin{pmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{pmatrix}.$$

- x, y : The horizontal and vertical positions of the beam.
- x', y' : Divergences of the beam in the x and y directions.
- $\langle x^2\rangle, \langle y^2\rangle$: Second moments of the beam.
- $\langle xx'\rangle, \langle yy'\rangle$: Correlations between position and angle.
- $\langle xy\rangle, \langle xy'\rangle, \langle x'y\rangle, \langle x'y'\rangle$: Cross-plane correlations.
- Σ_{xx} : Beam properties in the x-plane.
- Σ_{yy} : Beam properties in the y-plane.
- Σ_{xy} : Coupling between the x and y planes.
- Σ_{xy}^T : Transpose of Σ_{xy}.

- When all Σ_{xy} terms are zero, the apparent and intrinsic emittances are equal.
- When there are xy correlation terms, the projections onto the x and y axes, which yield the **apparent emittances**, are not the same as the **intrinsic emittances**.
- It can be shown that

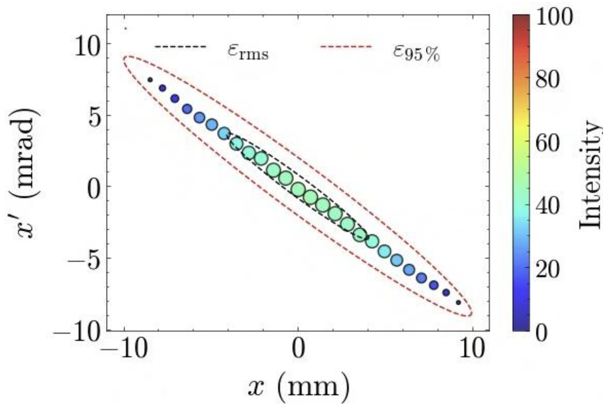
$$\boldsymbol{\epsilon_1\epsilon_2 \leq \epsilon_x\epsilon_y}$$

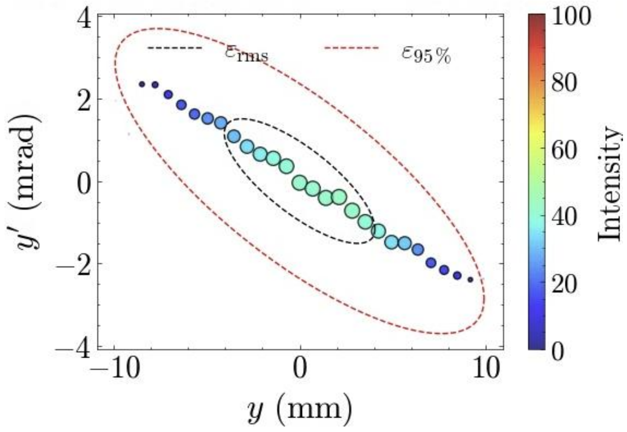
The **coupling parameter** t is introduced to quantify interplane coupling and is defined as

$$t = \frac{\varepsilon_x\varepsilon_y}{\varepsilon_1\varepsilon_2} - 1 \geq 0.$$

If t is equal to zero, there are no interplane correlations, and the projected RMS emittances are equal to the eigen emittances.

Slit scan method – Example





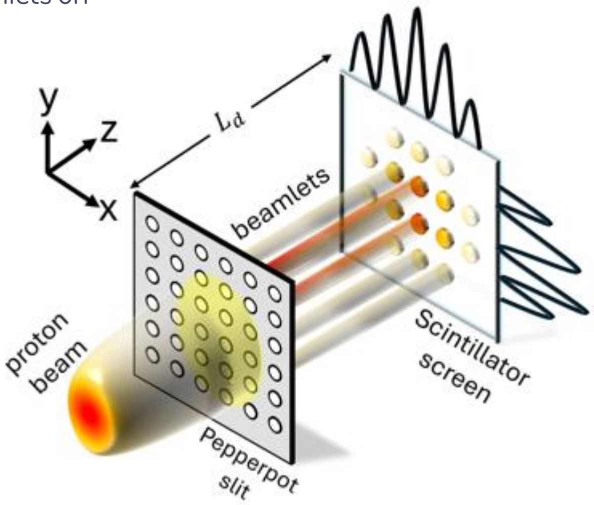
Parameter	Value
ε_x (m · rad)	3.380×10^{-6}
ε_y (m · rad)	3.914×10^{-6}
$\varepsilon_{n,x}$ (mm · mrad)	0.155
$\varepsilon_{n,y}$ (mm · mrad)	0.180
α_x	4.353
β_x (m/rad)	4.905
γ_x (rad/m)	4.067
α_y	1.204
β_y (m/rad)	4.199
γ_y (rad/m)	0.584

$$\Sigma_{4D} = \begin{bmatrix} 1.658 \times 10^{-5} & -1.471 \times 10^{-5} & 0 & 0 \\ -1.471 \times 10^{-5} & 1.374 \times 10^{-5} & 0 & 0 \\ 0 & 0 & 1.644 \times 10^{-5} & -4.714 \times 10^{-6} \\ 0 & 0 & -4.714 \times 10^{-6} & 2.284 \times 10^{-6} \end{bmatrix}$$

Pepper-pot method

- The pepper-pot method is an interceptive diagnostic used to measure the transverse emittance of a charged particle beam in both transverse planes simultaneously.
- Each hole transmits a small fraction of the beam, producing an array of beamlets.
- The hole positions provide the transverse coordinates, while the displacement of the beamlets on the screen gives the corresponding angular information.
- For a drift length (L), the beam angles are estimated from the beamlet displacement as

$$x' \approx \frac{x_{\text{screen}} - x_{\text{hole}}}{L}, \qquad y' \approx \frac{y_{\text{screen}} - y_{\text{hole}}}{L}.$$

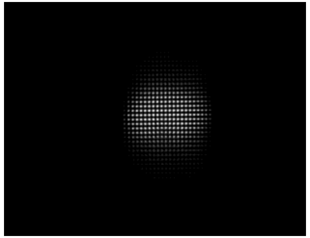


- Single-shot measurement in both transverse planes without mechanical scanning.
- It can also provide information on transverse coupling when a full two-dimensional analysis is performed.
- Its main disadvantages are limited resolution compared with slit-based methods, possible overlap of neighboring beamlets on the screen, reduced signal when very small holes are used, and distortion from scattering or diffraction effects.

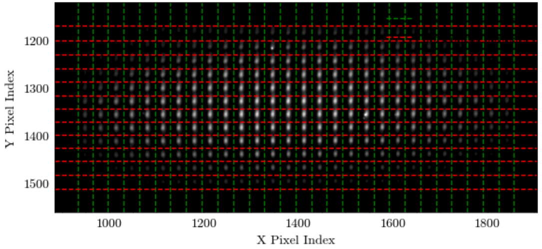
Pepper-pot method – Example

1. **Upload and Preprocess Data**
   - Load the original image data for both horizontal and vertical beamlets.
2. **Apply Median Filtering**
   - Use a median filter to remove noise and artifacts from the image.
3. **Detect Region of Interest (ROI) and Remove Background Noise**
   - Identify the beamlet region by applying thresholding or adaptive intensity filtering.
4. **Automatically Generate Gridlines and Isolate Individual Beamlets**
   - Use automated grid detection to separate beamlets efficiently.
   - Segment individual beamlets for precise measurements.
5. **Find Peak Intensity and Peak Position for Each Beamlet**
   - Identify peak pixel positions for each beamlet in both x and y directions.
6. **Match Beamlet Peak Points to Slit Hole Centers**
   - Align detected peak points with corresponding slit hole positions.
7. **Compute Beam Divergence**
   - Calculate the beamlet divergence for each beamlet.
8. **Calculate RMS and Normalized Emittance**
   - Compute RMS emittance of the beam.

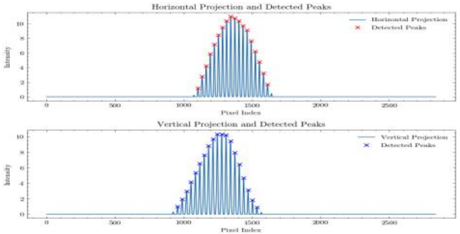
Step - 1



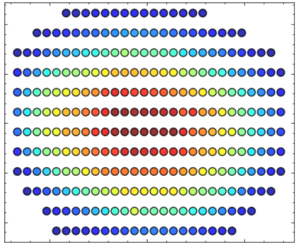
Step - 4



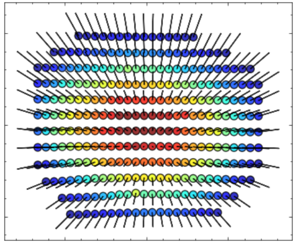
Step - 5



Step - 6

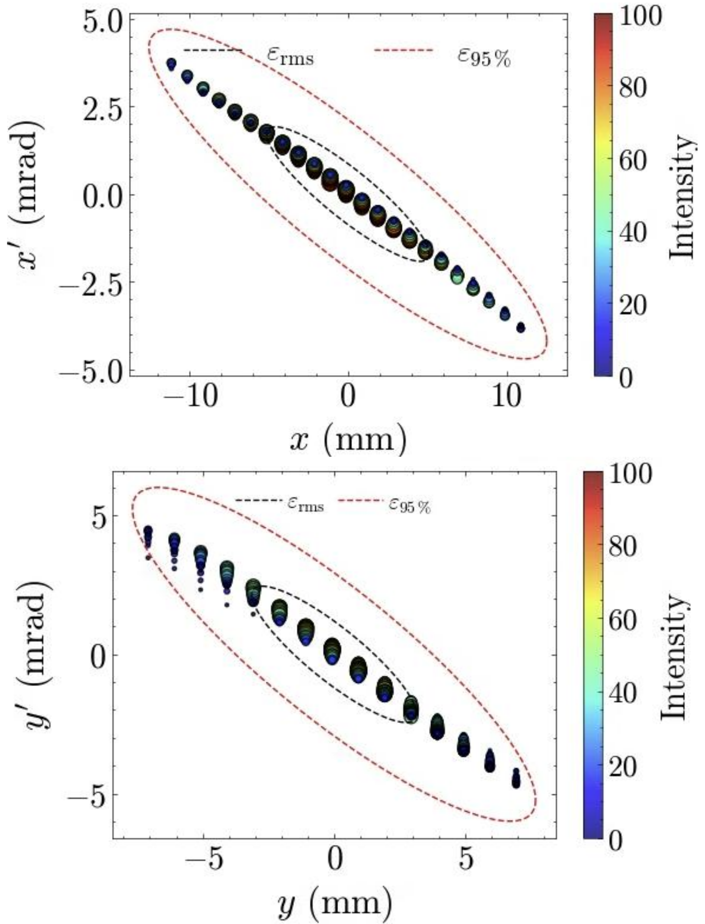


(Identified slit holes)



(corresponding beamlets on screen)

Pepper-pot method – Example

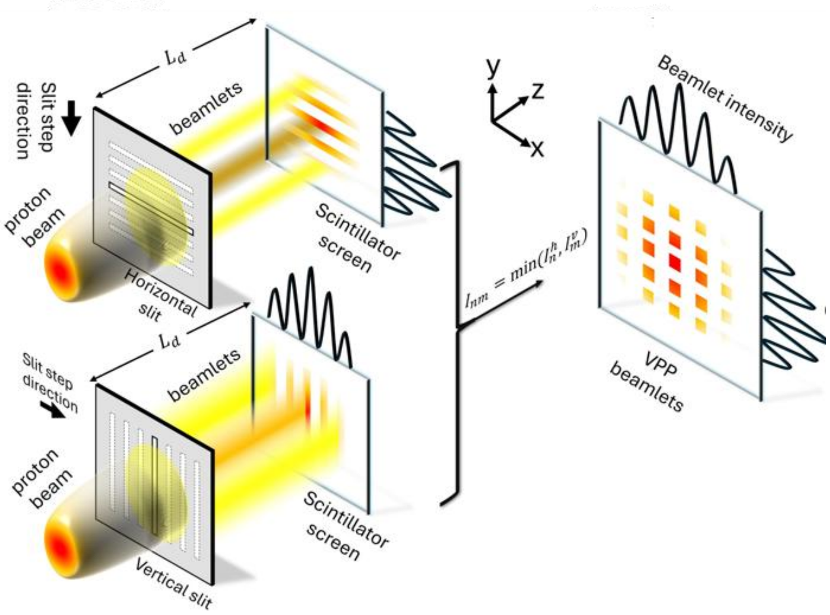


Parameter	Value
ε_x (m · rad)	4.437×10^{-6}
ε_y (m · rad)	3.746×10^{-6}
ε_{4D} (m · rad)	1.645×10^{-11}
$\varepsilon_{n,x}$ (mm · mrad)	0.204
$\varepsilon_{n,y}$ (mm · mrad)	0.172
α_x	1.969
β_x (m/rad)	5.894
γ_x (rad/m)	0.828
α_y	1.784
β_y (m/rad)	2.615
γ_y (rad/m)	1.599
ε_1 (m · rad)	4.608×10^{-6}
ε_2 (m · rad)	3.571×10^{-6}
$\varepsilon_1\varepsilon_2$ ($m^2 \cdot rad^2$)	1.645×10^{-11}
t	1.028×10^{-2}

$$\Sigma_{4D} = \begin{bmatrix} 2.615 \times 10^{-5} & -8.739 \times 10^{-6} & -1.007 \times 10^{-6} & 4.985 \times 10^{-8} \\ -8.739 \times 10^{-6} & 3.673 \times 10^{-6} & 5.380 \times 10^{-7} & -1.556 \times 10^{-7} \\ -1.007 \times 10^{-6} & 5.380 \times 10^{-7} & 9.794 \times 10^{-6} & -6.682 \times 10^{-6} \\ 4.985 \times 10^{-8} & -1.556 \times 10^{-7} & -6.682 \times 10^{-6} & 5.992 \times 10^{-6} \end{bmatrix}$$

Virtual Pepper-pot (VPP)

- The **Virtual Pepper-Pot (VPP)** works by replacing a physical pepper-pot mask with a **numerically reconstructed pepper-pot pattern** built from separate horizontal and vertical slit scans.
- VPP measures the beam with orthogonal slit scans, combines the two datasets into a virtual array of beamlets, and uses their positions and intensities to reconstruct the full 4D transverse phase space without a physical pepper-pot plate.
- By combining the two scan datasets, the crossings of horizontal and vertical beamlets define a matrix of **virtual holes.**
- The main function: **Minimum-bit principle.**
- The full **4D transverse phase space** is synthesized from the complete set of VPP beamlets.



Virtual Pepper-pot (VPP)

- Slit plane: coordinates $(\mathbf{x_i}, \mathbf{y_i})$. Screen plane: coordinates $(\mathbf{x_f}, \mathbf{y_f})$.
- A screen pixel is denoted by:

$$u = (x_f, y_f).$$

- Each slit intersection $(\mathbf{x_n}, \mathbf{y_m})$ produces a distinct beamlet spot on the screen. We denote a representative pixel of this beamlet by $\mathbf{u_{n,m}}$.
- A horizontal slit positioned at $\mathbf{x} = \mathbf{x_n}$ produces 2D screen image:

$$H_n(u) \equiv H_n(x_f, y_f),$$

- A vertical slit positioned at $\mathbf{y} = \mathbf{y_m}$ produces 2D screen image:

$$V_m(u) \equiv V_m(x_f, y_f),$$

- The VPP method constructs a *virtual intersection image* for each slit intersection $(\mathbf{x_n}, \mathbf{y_m})$ by applying a pixel-wise minimum operation:

$$W_{n,m}(u) = \min\big(H_n(u),\ V_m(u)\big).$$

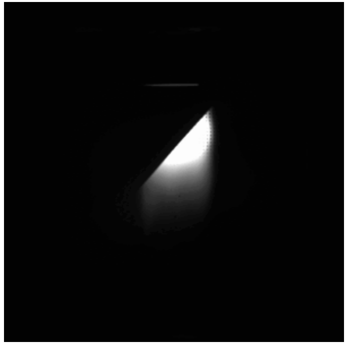
- The intensity of the virtual beamlet is obtained by integrating over a small region $\mathbf{R_{n,m}}$:

$$I_{\text{VPP}}(n, m) = \int_{R_{n,m}} W_{n,m}(u)\, du.$$

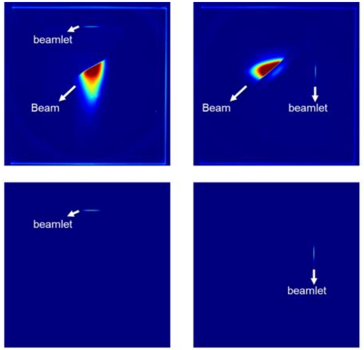
Virtual Pepper-pot (VPP) – Example

1. **Load and Preprocess Image**
   (a) Read the beam profile image and convert it to grayscale.
   (b) Filter the image and eliminate the beam spots (in this case noise).
   (c) Apply a median filter to remove noise while preserving beamlet structures.
2. **Define the Virtual Pepper Pot Grid**
   (a) Perform horizontal and vertical slit scans.
   (b) Construct a virtual slit mask with grid positions.
   (c) Apply the minimum intensity principle to obtain a 2D beamlet distribution.
3. **Extract Beamlet Projections**
   (a) Compute the horizontal and vertical projections by summing intensity along the horizontal and vertical axis.
   (b) Detect beamlet peaks using Gaussian fitting.
   (c) Match the peak beamlet locations to the slit center locations.
4. **Convert Pixel Positions to Physical Units**
   (a) Define conversion factors to transform pixel positions into millimeters.
   (b) Compute beam divergence angles from slit positions.
5. **Compute RMS Emittance**
   (a) Compute mean beamlet positions and divergences.
   (b) Calculate RMS values for beam size and angular spread.
   (c) Compute RMS emittance and Twiss parameters.

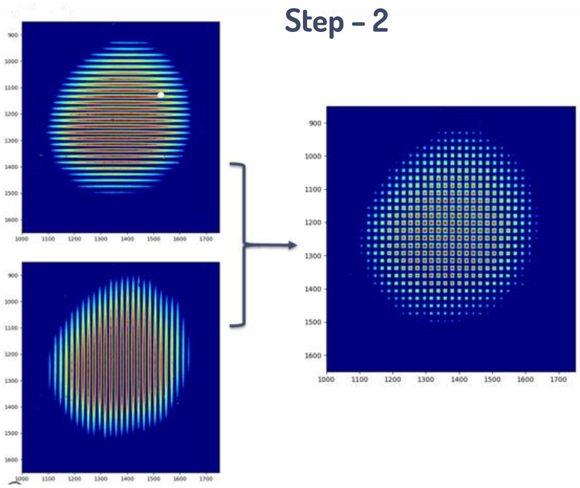
Step – 1



(.gif image)

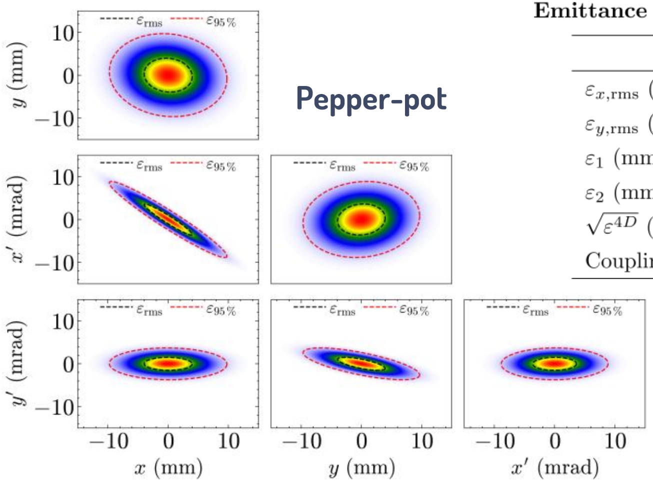


Step – 2



Benchmarking

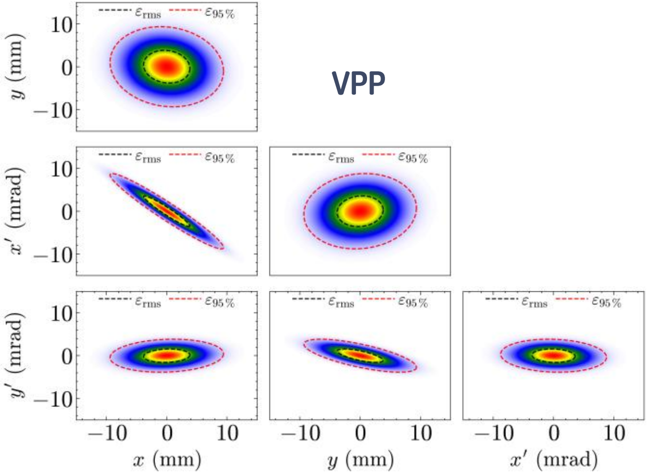
- The comparison involves the phase space distribution for **QM3 = 15 A** for Pepper-pot and VPP.
- Despite these small variations, the overall distributions remain consistent, confirming the validity of both methods.



Pepper-pot

Emittance and coupling factor between Pepperpot and VPP.

	Pepper pot	Virtual Pepper pot
$\varepsilon_{x,\rm rms}$ (mm mrad)	3.937	3.755
$\varepsilon_{y,\rm rms}$ (mm mrad)	4.267	4.270
ε_1 (mm mrad)	4.489	4.430
ε_2 (mm mrad)	3.700	3.583
$\sqrt{\varepsilon^{4D}}$ (mm mrad)	4.075	3.983
Coupling factor (t)	0.0112	0.0101



VPP

$$\Sigma^{4D}_{PP} = \begin{bmatrix} 1.588\times10^{-5} & -1.388\times10^{-5} & -1.381\times10^{-6} & -1.260\times10^{-7} \\ -1.388\times10^{-5} & 1.311\times10^{-5} & 1.342\times10^{-6} & 7.175\times10^{-8} \\ -1.381\times10^{-6} & 1.342\times10^{-6} & 1.562\times10^{-5} & -4.184\times10^{-6} \\ -1.260\times10^{-7} & 7.175\times10^{-8} & -4.184\times10^{-6} & 2.286\times10^{-6} \end{bmatrix}$$

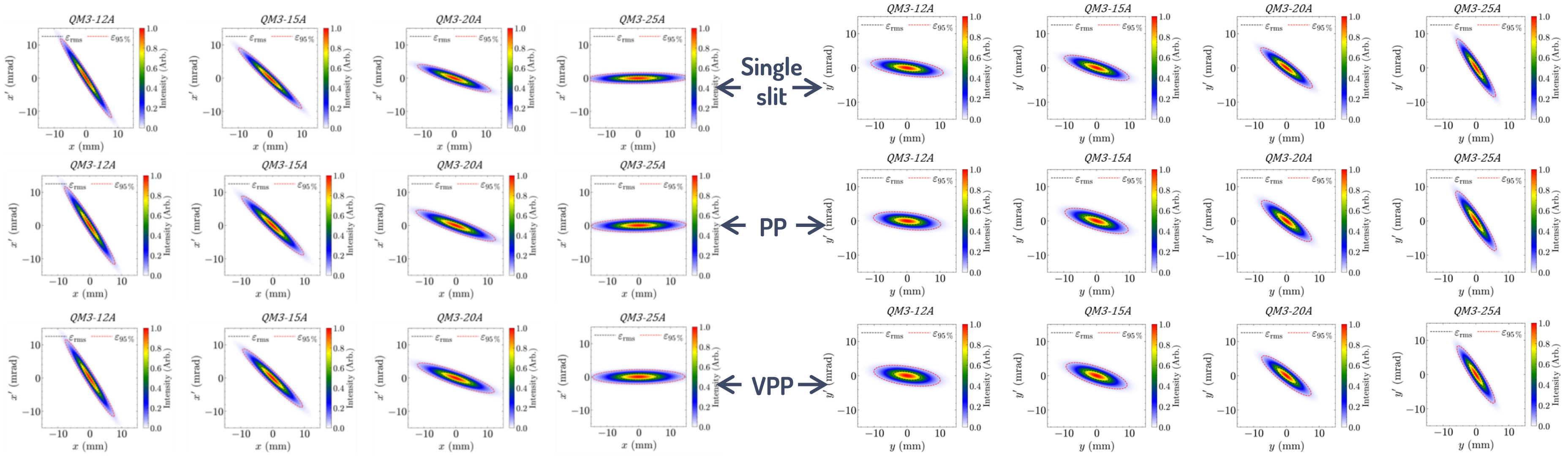
$$\Sigma^{4D}_{VPP} = \begin{bmatrix} 1.474\times10^{-5} & -1.323\times10^{-5} & -1.611\times10^{-6} & 6.897\times10^{-7} \\ -1.323\times10^{-5} & 1.283\times10^{-5} & 1.179\times10^{-6} & -5.449\times10^{-7} \\ -1.611\times10^{-6} & 1.179\times10^{-6} & 1.453\times10^{-5} & -4.181\times10^{-6} \\ 6.897\times10^{-7} & -5.449\times10^{-7} & -4.181\times10^{-6} & 2.458\times10^{-6} \end{bmatrix}$$

Benchmarking

- Varying quadrupole magnet strengths: 12A, 15A, 20A, 25A.

Horizontal phase space

Vertical phase space

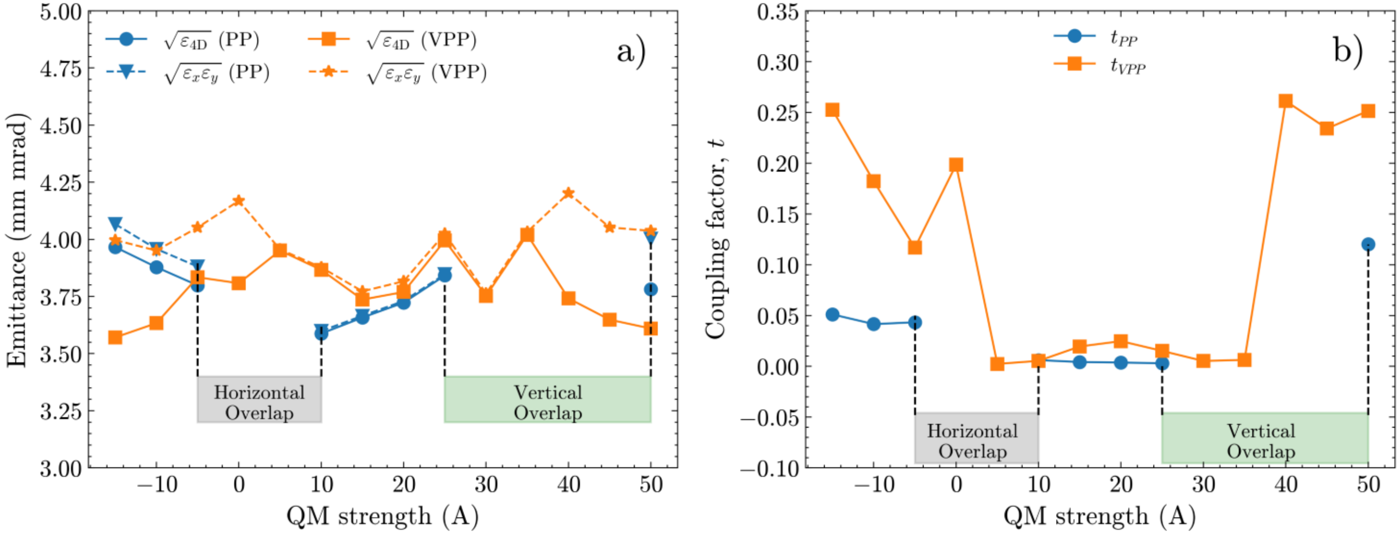


- Higher quadrupoles currents leads to **anti clockwise** rotation
- Horizontal defocusing effect

- Higher quadrupoles currents leads to **clock-wise** rotation
- Vertical focusing effect

Benchmarking

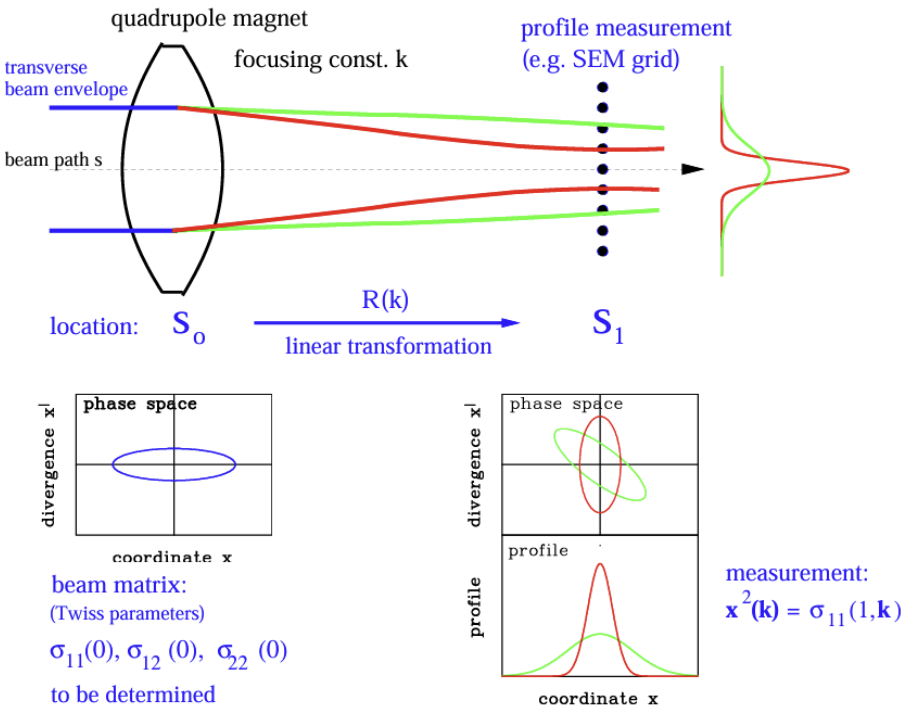
- While both methods capture similar trends across the quadrupole scan, slight discrepancies are present in certain regions.
- As expected, the coupling factor remains close to zero within the central range of the quadrupole scan (QM-III: 5–35 A), where the beam conditions are near-ideal and the transverse profiles are approximately round.
- However, at the extremes of the scan (QM-III: −15 A and 50 A), the coupling factor increases significantly, indicating stronger inter-plane coupling.
- PP data are either missing in some overlap regions.
- In contrast, the VPP method does not suffer from this limitation, as it reconstructs the phase space distribution.



Quadrupole Scan method

- The quadrupole scan method is a standard beam diagnostic technique used to determine the transverse beam properties at a **chosen reconstruction point** in a beamline.
- Its main purpose is to reconstruct the **second-order beam moment.**

1. Choose a reconstruction point where the beam parameters are to be determined.
2. Select a quadrupole whose strength will be varied.
3. Choose a downstream screen or profile monitor where the beam size will be measured.
4. Define a scan range of quadrupole strengths.
5. For each quadrupole setting, compute the corresponding transport matrix from the reconstruction point to the screen.
6. Measure or simulate the rms beam size at the screen.
7. Fit the scan equation to recover the initial second moments.
8. Convert the fitted second moments into emittance and Twiss parameters.

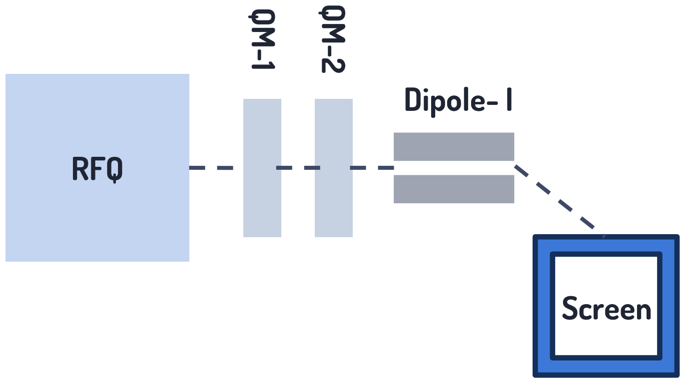


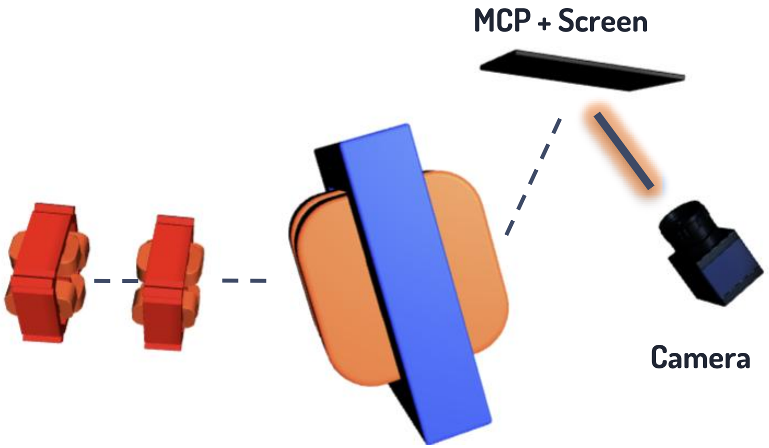
From: Forck, Peter. "Lecture notes on beam instrumentation and diagnostics." Joint Universities Accelerator School (JUAS 2010) (2011).

Quadrupole Scan – Example

Muon Linac RFQ will be commissioned soon

- We will conduct Quadrupole scan method at the RFQ exit reconstruction point.
- Beamline includes 2 quadrupole magnet, 1 dipole magnet, MCP based screen.
- The data will be collected in the screen by varying the QM-1 strength.





Quadrupole Scan – Example

The beamline: $D_1 \rightarrow Q1(k) \rightarrow D_2 \rightarrow Q2 \rightarrow D_3 \rightarrow B \rightarrow D_4 \rightarrow \text{Screen}.$

The scan equation is: $\sigma^2(k) = R_{11}^2(k)\Sigma_{11} + 2R_{11}(k)R_{12}(k)\Sigma_{12} + R_{12}^2(k)\Sigma_{22},$

The total first-order transport matrix from the beamline entrance to the screen:

$$R(k) = D_4\, B\, D_3\, Q2\, D_2\, Q1(k)\, D_1 .$$

If the initial beam matrix is Σ_0 then the beam matrix at the screen is:

$$\Sigma_s(k) = R(k)\, \Sigma_0\, R^T(k).$$

The rms beam size at the screen: $\sigma^2(k) = [\Sigma_s(k)]_{11} .$

Beam size squared at the screen is approximately a quadratic function of the quadrupole strength:

$$\sigma^2(k) \approx ak^2 + bk + c.$$

Quadrupole Scan – Example

Quadrupole Scan method is adapted to Cheetah code

```
def make_segment_q1_scan(k1_q1_scan):

return Segment(elements=[
*[Drift(length=ld1, name=f"drift1_{i}") for i in range(
   nslice_drift)],

*[Quadrupole(length=lq1, k1=torch.as_tensor(k1_q1_scan,
   dtype=dtype), name=f"q1_{i}") for i in range(
   nslice_quad)],

*[Drift(length=ld2, name=f"drift2_{i}") for i in range(
   nslice_drift)],

*[Quadrupole(length=lq2, k1=torch.tensor(k1_q2_nom, dtype
   =dtype), name=f"q2_{i}") for i in range(nslice_quad)],

*[Drift(length=ld3, name=f"drift3_{i}") for i in range(
   nslice_drift)],

*[Dipole(length=lb, angle=ab, k1=torch.tensor(0.0, dtype=
   dtype), tracking_method="linear", name=f"bend_{i}")
   for i in range(nslice_bend)],

*[Drift(length=ld4, name=f"drift4_{i}") for i in range(
   nslice_drift)],

Screen(name="mcp_screen"),
])
```

```
def total_first_order_map(segment, beam_in):
R_total = torch.eye(7, dtype=beam_in.energy.dtype
   , device=beam_in.energy.device)

energy = beam_in.energy
species = beam_in.species

for elem in segment.elements:
R_elem = elem.first_order_transfer_map(energy,
   species)
R_total = R_elem @ R_total

return R_total
```

This segment configuration allows:

- Slicing technique of elements,
- Importing 1D field map of elements,
- Applying of SC kick modules.

Quadrupole Scan – Example

```
def fit_second_moments_from_scan(sigma_screen_m,
    R11, R12):
y = sigma_screen_m**2

A = np.column_stack([
R11**2,
2.0 * R11 * R12,
R12**2
])

sol, residuals, rank, svals = np.linalg.lstsq(A,
    y, rcond=None)
S11, S12, S22 = sol

emit_sq = S11 * S22 - S12**2
emit = np.sqrt(emit_sq)

beta = S11 / emit
alpha = -S12 / emit
gamma = S22 / emit
```

Lines 4–8:

```
A = np.column_stack([
R11**2,
2.0 * R11 * R12,
R12**2
])
```

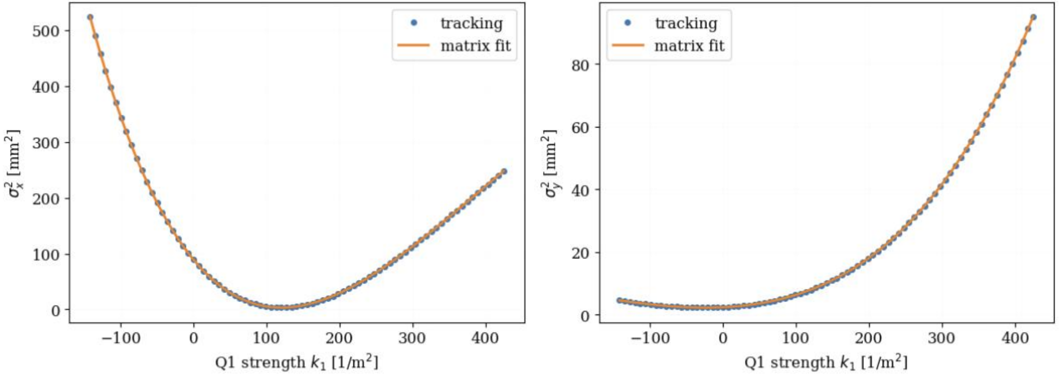
This builds the linear system corresponding to

$$\sigma^2(k) = R_{11}^2(k)S_{11} + 2R_{11}(k)R_{12}(k)S_{12} + R_{12}^2(k)S_{22}. \quad (4)$$

Lines 10–11:

```
sol, residuals, rank, svals = np.linalg.lstsq(A,
    y, rcond=None)
S11, S12, S22 = sol
```

This solves the least-squares problem and returns the fitted second moments S_{11}, S_{12}, and S_{22}.



Parameter	Ground truth	Direct x-fit
ε_x [m rad]	2.990094×10^{-6}	3.409688×10^{-6}
$\varepsilon_{n,x}$ [mm mrad]	0.239566	0.273183
β_x [m]	0.211239	0.188602
α_x	−1.743599	−1.494757

Parameter	Ground truth	Fitted y-plane
ε_y [m rad]	2.895983×10^{-6}	2.896416×10^{-6}
$\varepsilon_{n,y}$ [mm mrad]	0.232025	0.232060
β_y [m]	0.071923	0.071927
α_y	0.542955	0.542851

Dispersion effects?

Quadrupole Scan – Example

The horizontal coordinate at the screen can be written as:

$$x_{\text{screen}} = \underbrace{R_{11}(k)x_{\beta 0} + R_{12}(k)p_{x\beta 0}}_{\text{betatron part}} + \underbrace{[R_{11}(k)D_{x0} + R_{12}(k)D_{p_x 0} + R_{16}(k)]\,\delta_0}_{\text{dispersive part}} .$$

Total horizontal dispersion at the screen:

$$D_x^{\text{screen}}(k) = R_{11}(k)D_{x0} + R_{12}(k)D_{p_x 0} + R_{16}(k).$$

Dispersion-corrected horizontal beam size:

$$\sigma_{x,\beta}^2(k) = \sigma_x^2(k) - \left(D_x^{\text{screen}}(k)\,\sigma_p\right)^2 .$$

Results:

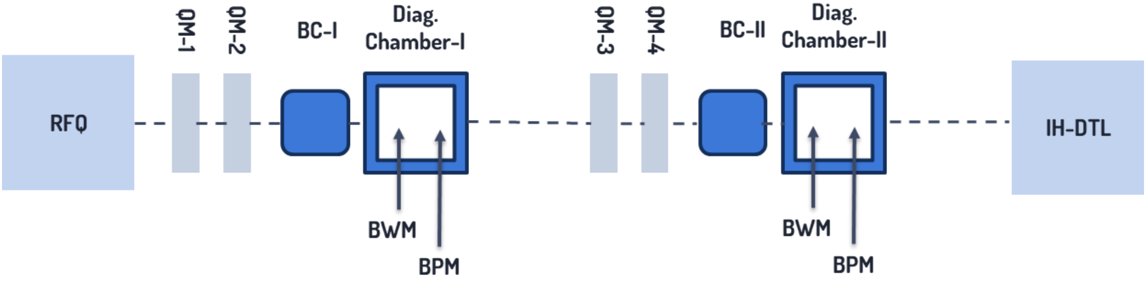
Parameter	Ground truth	Direct x-fit	Dispersion-corrected x-fit
ε_x [m rad]	2.990094×10^{-6}	3.409688×10^{-6}	2.990110×10^{-6}
$\varepsilon_{n,x}$ [mm mrad]	0.239566	0.273183	0.239567
β_x [m]	0.211239	0.188602	0.211238
α_x	−1.743599	−1.494757	−1.743588
D_x [m]	-3.894431×10^{-3}	—	-3.894431×10^{-3}
D_{p_x}	2.636621×10^{-1}	—	2.636621×10^{-1}

Parameter	Ground truth	Fitted y-plane
ε_y [m rad]	2.895983×10^{-6}	2.896416×10^{-6}
$\varepsilon_{n,y}$ [mm mrad]	0.232025	0.232060
β_y [m]	0.071923	0.071927
α_y	0.542955	0.542851

Table of Content

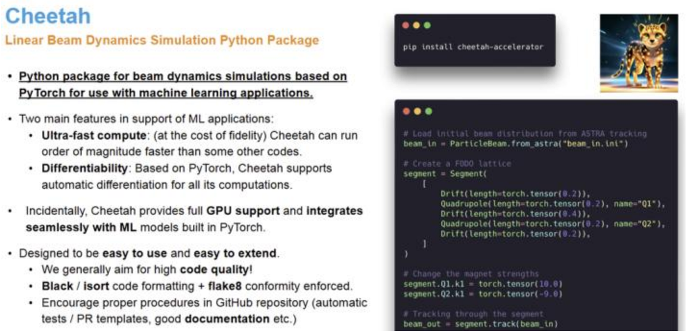
Gradient based optimization

- Instead of adjusting magnets by trial and error, you define a **goal** for the beam, such as a target spot size, position, or shape, and let an optimization algorithm change the magnet strengths to reduce the error.
- A gradient tells how sensitive the objective is to each tuning parameter. So instead of guessing how to change magnets, computing how each magnet affects the final beam quality and use that information to move in a better direction.
- **Main objective**: Tune the quadrupole parameters and match the **muon beam** to the IH-DTL.



- start with a muon beam coming from the RFQ,
- pass it through several beamline elements,
- adjust the **quadrupole magnets** automatically,
- make the beam match the requirements of the **IH-DTL** at the end.

- We need to use a differentiable simulation tool for gradieant based optimization .
- Cheetah, a PyTorch-based highspeed differentiable linear beam dynamics code.
- Beamline can be modeled in PyTorch, and gradients with respect to quadrupole strengths can be obtained automatically.

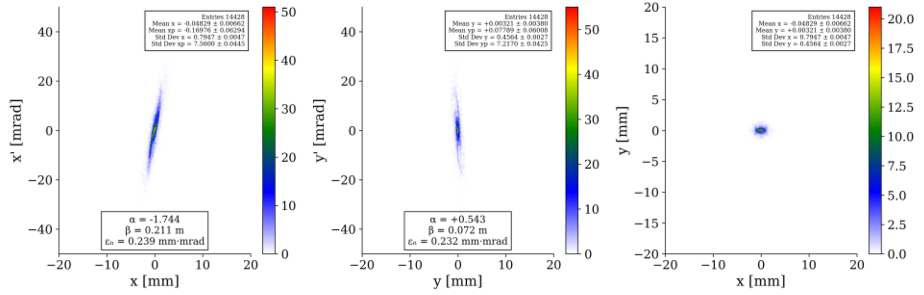


Gradient based optimization

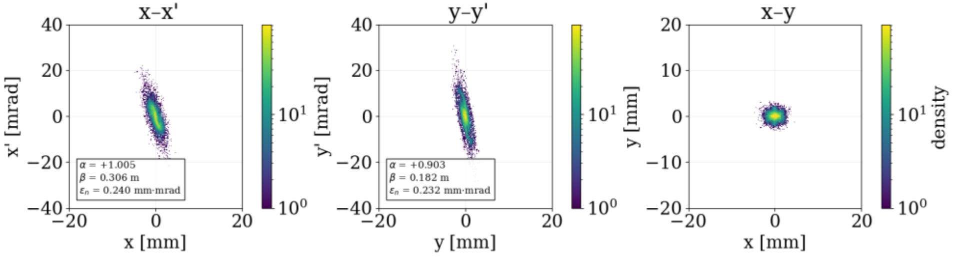
Manual-tuning benchmark

- The quadrupole settings are varied in simulation and the resulting beam transport is evaluated.
- **Objective: Match the beam to the IH-DTL**

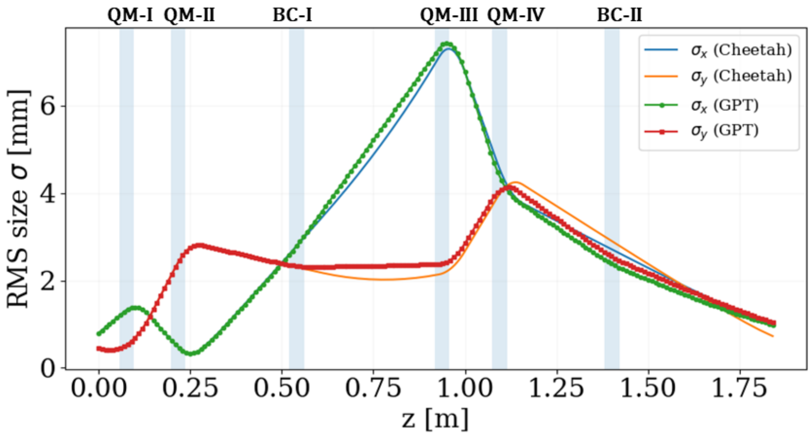
RFQ exit phase space



IH-DTL entrance phase space



Beam trajectory along the beamline



Target Twiss parameters		Best observed results	
$\alpha_x = 0.701$	$\alpha_y = 0.966$	$\alpha_x = 1.005$	$\alpha_y = 0.903$
$\beta_x = 0.320$	$\beta_y = 0.378$	$\beta_x = 0.306$	$\beta_y = 0.182$

- **Gradient based optimization can tune beam to IH-DTL easily?**

Gradient based optimization

- The Twiss parameters at the end of the lattice become a deterministic function of the quadrupole settings:

$$(\beta_x, \alpha_x, \beta_y, \alpha_y) = f(\mathbf{k}),$$

$$\mathbf{k} = (k_{1,\mathrm{QM1}}, k_{1,\mathrm{QM2}}, k_{1,\mathrm{QM3}}, k_{1,\mathrm{QM4}}) \,.$$

- The optics matching objective is to select $(\mathbf{k})$ such that these outgoing parameters reproduce the desired targets $(\boldsymbol{\beta}_{\mathbf{x}}^*, \boldsymbol{\alpha}_{\mathbf{x}}^*, \boldsymbol{\beta}_{\mathbf{y}}^*, \boldsymbol{\alpha}_{\mathbf{y}}^*)$ at the diagnostic position.
- We define the observed (predicted) outgoing beam-parameter vector as;

$$\mathbf{y}(\mathbf{k}) = \begin{pmatrix} \beta_x(\mathbf{k}) \\ \alpha_x(\mathbf{k}) \\ \beta_y(\mathbf{k}) \\ \alpha_y(\mathbf{k}) \end{pmatrix}, \qquad \mathbf{y}^* = \begin{pmatrix} \beta_x^* \\ \alpha_x^* \\ \beta_y^* \\ \alpha_y^* \end{pmatrix}.$$

- The optimization objective is defined through a mean-square-error loss function:

$$\mathcal{L}(\mathbf{k}) = \frac{1}{4} \sum_{i=1}^{4} (y_i(\mathbf{k}) - y_i^*)^2 = \frac{1}{4} \|\mathbf{y}(\mathbf{k}) - \mathbf{y}^*\|_2^2 \,.$$

- The full optics matching becomes a continuous nonlinear optimisation:

$$\mathbf{k}^{\mathrm{opt}} = \arg\min_{\mathbf{k} \in \mathbb{R}^4} \mathcal{L}(\mathbf{k}).$$

Gradient based optimization

» A key advantage of this framework is that the beamline simulation is implemented using **torch. Tensor** operations. Therefore, the output Twiss parameters and the loss function remain differentiable with respect to quadrupole strengths. PyTorch computes gradients via reverse-mode automatic differentiation (backpropagation):

$$\nabla_{\mathbf{k}}\mathcal{L}(\mathbf{k}) = \left(\frac{\partial\mathcal{L}}{\partial k_{1,\mathrm{QM1}}}, \frac{\partial\mathcal{L}}{\partial k_{1,\mathrm{QM2}}}, \frac{\partial\mathcal{L}}{\partial k_{1,\mathrm{QM3}}}, \frac{\partial\mathcal{L}}{\partial k_{1,\mathrm{QM4}}}\right).$$

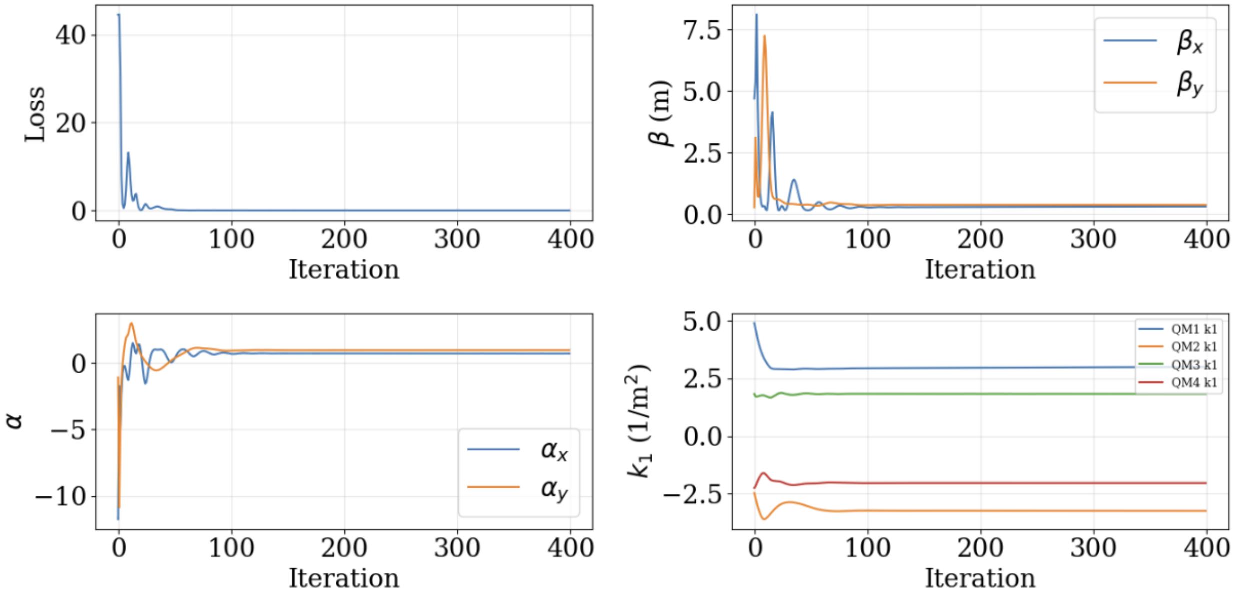
» The optimisation loop therefore follows the standard machine-learning "training" structure:

1. Track the beam through the lattice for the current magnet settings: $\mathbf{y}(\mathbf{k_t})$.
2. Compute the mismatch loss .
3. Backpropagate to compute $(\nabla_{\mathbf{k}}\mathcal{L})$.
4. Update the quadrupole strengths using Adam.
5. Repeat until convergence.

```python
def train_model(model, beam, target, num_steps=300, lr=0.05, print_every=10):
    target = target.to(dtype=model.k1.dtype, device=model.k1.device).view(-1)
    assert target.numel() == 4

    opt = torch.optim.Adam(model.parameters(), lr=lr)    # (4)

    loss_hist, beam_hist, k1_hist = [], [], []
    pbar = tqdm(range(num_steps), desc="Training", unit="iter")
    beam0 = beam.clone()

    final_beam = None
    for step in pbar:
        opt.zero_grad()
        final_beam = model(beam0.clone())    # (1)

        observed = torch.stack([final_beam.beta_x, final_beam.alpha_x,
                                final_beam.beta_y, final_beam.alpha_y])

        loss = F.mse_loss(observed, target)    # (2)
        loss.backward()    # (3)
        opt.step()

        loss_val = float(loss.detach().cpu())
        loss_hist.append(loss_val)
        beam_hist.append(observed.detach().cpu().numpy())
        k1_hist.append(model.k1.detach().cpu().numpy())
```

Gradient based optimization

Training process:



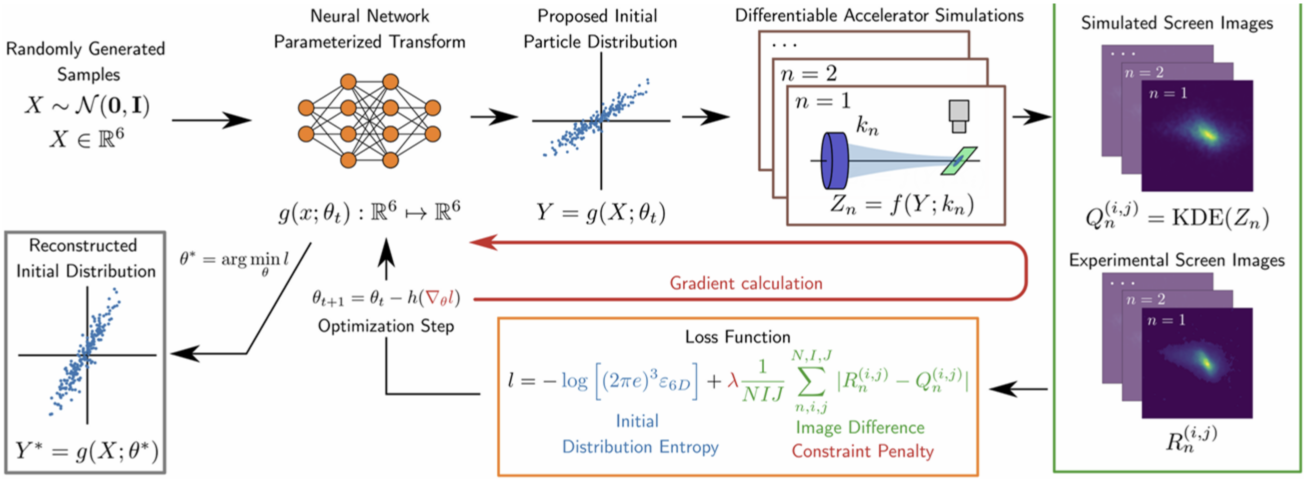
IH-DTL matching results:

Case	Horizontal		Vertical	
	α_x	β_x	α_y	β_y
Target parameters	0.701	0.320	0.966	0.378
Manual tuning	1.005	0.306	0.903	0.182
Gradient-based optimization	0.704	0.315	0.965	0.379

~ 15 seconds at a GPU machine

Generative Phase Space Reconstruction

- Generative Phase Space Reconstruction (GPSR) combines **neural network** with **differentiable particle tracking/beam dynamics** to infer high-dimensional (4D/6D) phase-space beam distributions.
- The method reconstructs detailed initial beam distributions using only standard diagnostics (e.g., a quadrupole scan and a downstream screen), without requiring specialized measurement devices.



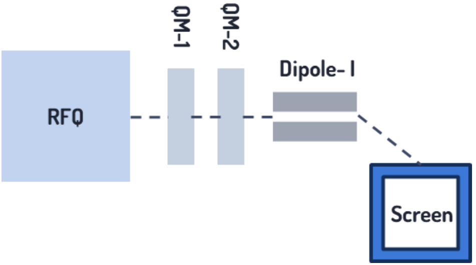
Overview of the GPSR workflow for reconstructing particle-beam phase-space distributions (taken from Roussel et al., PRL 130, 145001 (2023)).

- generate a candidate initial beam,
- propagate it through the differentiable accelerator model,
- render the predicted diagnostic image,
- compare prediction to measurement,
- backpropagate the discrepancy through the diagnostic, transport, and generator,
- update the generator parameters,
- repeat until the predicted measurements match the real data.

Generative Phase Space Reconstruction

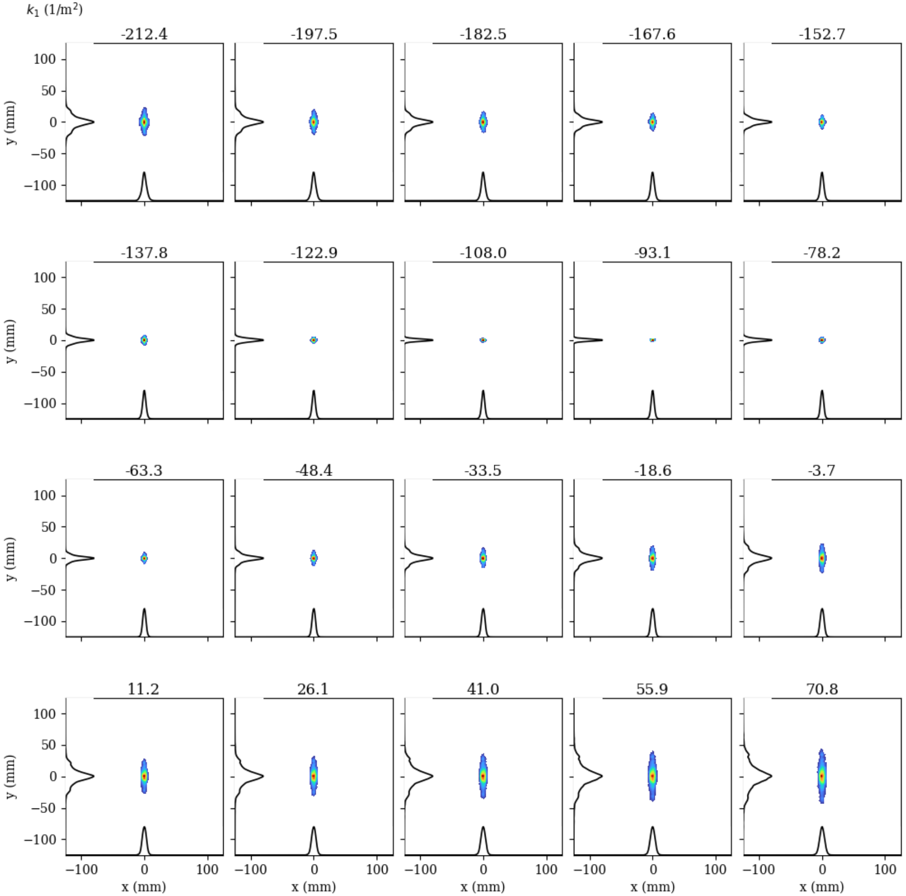
1. Generate train and test dataset on screen.

- We use same beamline in the example of quadrupole scan of Muon Linac
- RFQ exit is again reconstruction point



- $\widetilde{I_n^{\text{meas}}}$ is the measured 2D image on the screen for setting s_n.
- s_n may include quantities such as quadrupole strength, TDC voltage, dipole state, or other beamline controls.
- The training set becomes

$$\widetilde{\mathcal{D}} = \left\{ \left(s_n, \widetilde{I}_n^{\text{meas}} \right) \right\}_{n=1}^{N}.$$



Generative Phase Space Reconstruction

2. Initialize the generative beam model

- A latent random vector is drawn from a simple prior distribution,

$$\boldsymbol{Z} \sim p_0(\boldsymbol{Z}),$$

- A neural generator maps latent samples into phase-space particles:

$$\boldsymbol{X} = g_{\boldsymbol{\theta}}(\boldsymbol{Z}),$$

$\boldsymbol{\theta}$ is trainable parameters of the generator.

- Forward propagate the beam through the lattice.
- For each machine setting s_n, the initial particles are transported through the accelerator beamline:

$$\boldsymbol{X}_i^{(n)} = F_{s_n}\left(\boldsymbol{X}_i^{(0)}\right), \qquad i = 1, \ldots, M.$$

F_{s_n} is the differentiable transport map corresponding to setting s_n.

- At the screen, the coordinates are obtained by projection:

$$(u_i^{(n)}, v_i^{(n)}) = P\boldsymbol{X}_i^{(n)},$$

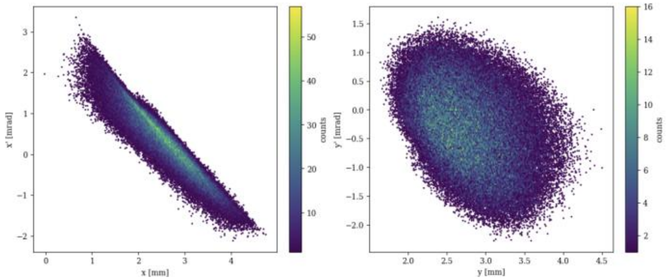
- The predicted image is generated by kernel density estimation:

$$\widetilde{I}_n^{\text{pred}}(u, v) = \frac{1}{M} \sum_{i=1}^{M} K_h\left(u - u_i^{(n)}, v - v_i^{(n)}\right),$$

- A Gaussian kernel is commonly used:

$$K_h(\Delta u, \Delta v) = \frac{1}{2\pi h^2} \exp\left[-\frac{\Delta u^2 + \Delta v^2}{2h^2}\right].$$

Latent projection



Generative Phase Space Reconstruction

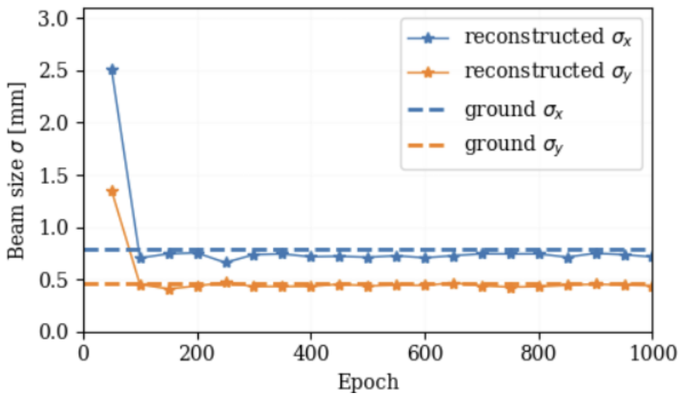
3. Compute the data-matching loss

» The total loss:

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{n=1}^{N} D\left(\widetilde{I}_n^{\text{pred}}, \widetilde{I}_n^{\text{meas}}\right),$$

» Mean absolute error:

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{NIJ} \sum_{n=1}^{N} \sum_{i=1}^{I} \sum_{j=1}^{J} \left| \widetilde{I}_n^{\text{meas}}(i,j) - \widetilde{I}_n^{\text{pred}}(i,j) \right|.$$



4. Backpropagate gradients

» Because the forward model is differentiable, the loss gradient with respect to the generator parameters can be written;

$$\nabla_{\boldsymbol{\theta}} \mathcal{L} = \sum_{n=1}^{N} \frac{\partial \mathcal{L}}{\partial \widetilde{I}_n^{\text{pred}}} \frac{\partial \widetilde{I}_n^{\text{pred}}}{\partial \boldsymbol{X}^{(n)}} \frac{\partial \boldsymbol{X}^{(n)}}{\partial \boldsymbol{X}^{(0)}} \frac{\partial \boldsymbol{X}^{(0)}}{\partial \boldsymbol{\theta}}.$$

$\frac{\partial \mathcal{L}}{\partial \widetilde{I}_n^{\text{pred}}}$: image sensitivity, $\frac{\partial \boldsymbol{X}^{(n)}}{\partial \boldsymbol{X}^{(0)}}$: beamline sensitivity,

$\frac{\partial \widetilde{I}_n^{\text{pred}}}{\partial \boldsymbol{X}^{(n)}}$: screen-particle sensitivity, $\frac{\partial \boldsymbol{X}^{(0)}}{\partial \boldsymbol{\theta}}$: generator sensitivity.

» The parameters are updated using gradient-based optimization:

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \eta_k \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}_k),$$

η_k : the learning rate.

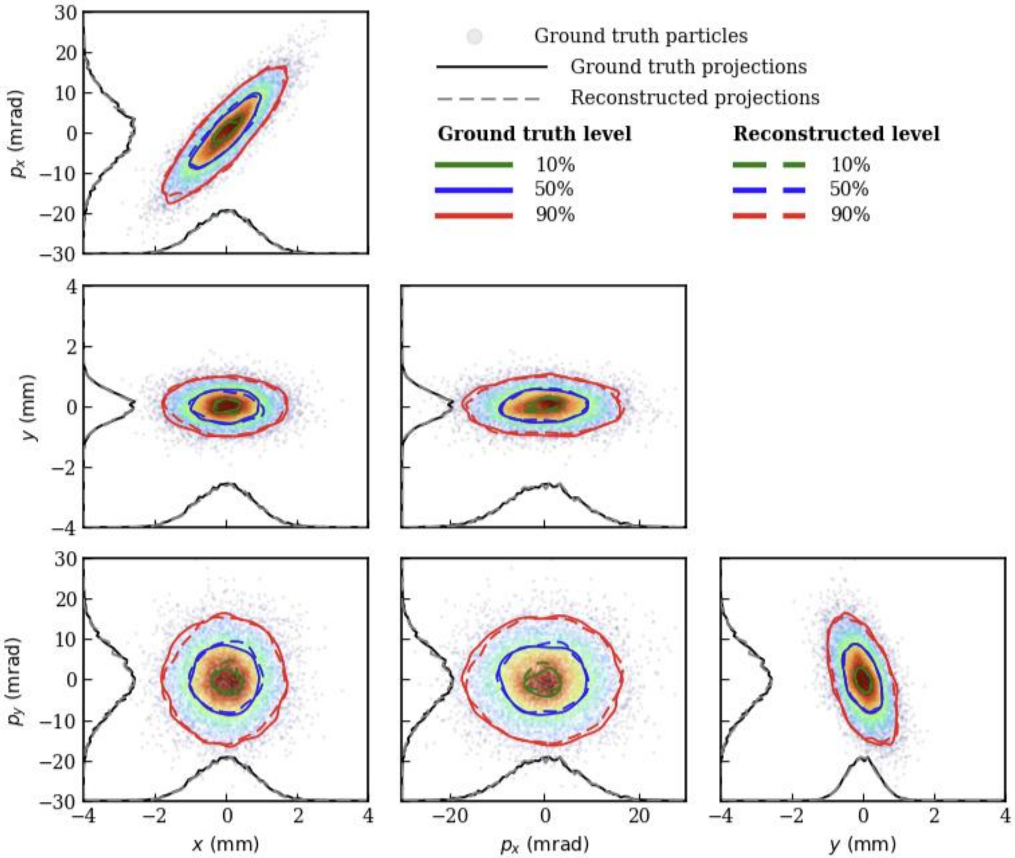
» The forward and backward steps are repeated until the loss converges.

$$\mathcal{L}(\boldsymbol{\theta}_{k+1}) \approx \mathcal{L}(\boldsymbol{\theta}_k).$$

Generative Phase Space Reconstruction

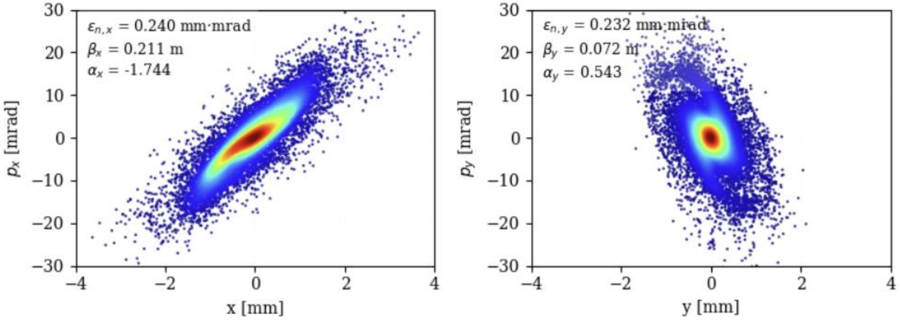
3. Evaluate the results

	Initial Beam	Reconstructed Beam
$\epsilon_{n,x}$	0.2396 mm · mrad	0.2331 mm · mrad
$\epsilon_{n,y}$	0.2320 mm · mrad	0.2359 mm · mrad
α_x	-1.7436	-1.5674
β_x	0.2112 mm/rad	0.2185 mm/rad
α_y	0.5430	0.3421
β_y	0.0719 mm/rad	0.0793 mm/rad
σ_x	0.7947 mm	0.7972 mm
σ_y	0.4564 mm	0.4833 mm
σ_{p_x}	7.5626 mrad	6.7844 mrad
σ_{p_y}	7.2205 mrad	6.4405 mrad

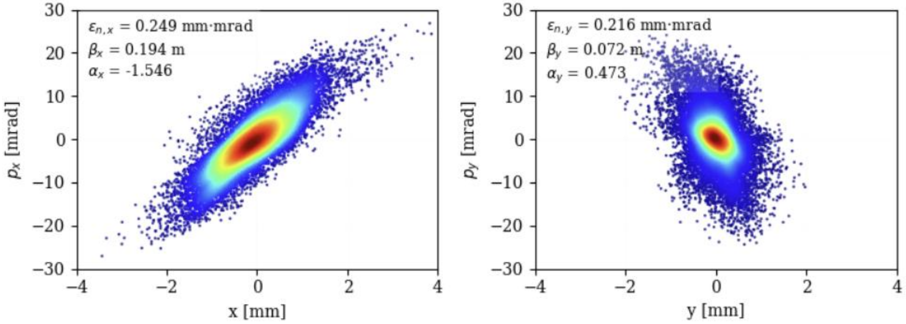


GPSR & Quadrupole Scan

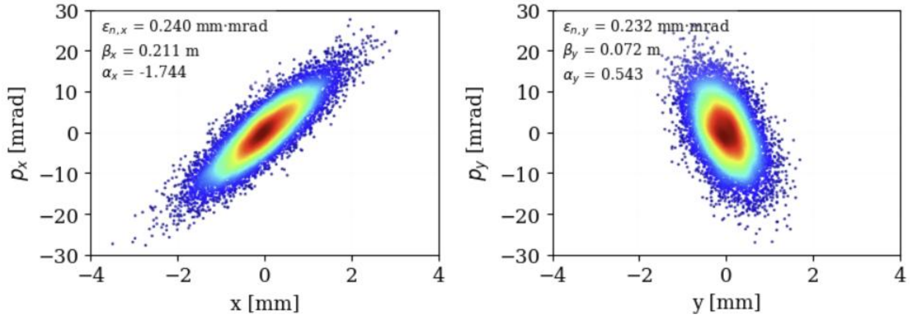
Ground truth



GPSR prediction



Quad scan reconstruction



Conclusion

Transverse phase-space reconstruction methods have been developed.

- The single-slit, pepper-pot, and virtual pepper-pot reconstruction methods were developed at KOMAC BTS.
- An Allison-type emittance scanner was also investigated at KOMAC BTS.
- The quadrupole scan method is planned for phase-space reconstruction during the commissioning of the Muon Linac.

We also present beamline optimization and phase-space prediction methods.

- We show that gradient-based beamline optimization can be applied to beam matching in the Muon Linac IH-DTL.
- The GPSR method can be used to reconstruct the initial phase space during the commissioning of the Muon Linac MEBT section.

THANKS!