ICFA Mini-Workshop on AI/ML
Chicago, Nov, 2022

Transverse 2D phase-space tomography using BPMs

Kilean Hwang
FRIB

Chad Mitchell, Robert Ryne
LBNL

This material is based upon work supported by the U.S. Department of Energy Office of Science under Cooperative Agreement DE-SC0000661, the State of Michigan and Michigan State University. Michigan State University designs and establishes FRIB as a DOE Office of Science National User Facility in support of the mission of the Office of Nuclear Physics.

Theoretical marginal profile estimation using turn-by-turn kicked beam centroid data

kicked beam centroid in normal form

- Time evolution of the 2D canonical variables in normal form read

$$x(t) - ip(t) = (x_0 - ip_0)e^{i\omega t} = \sqrt{2J_0}e^{i(\omega t - \theta)}$$

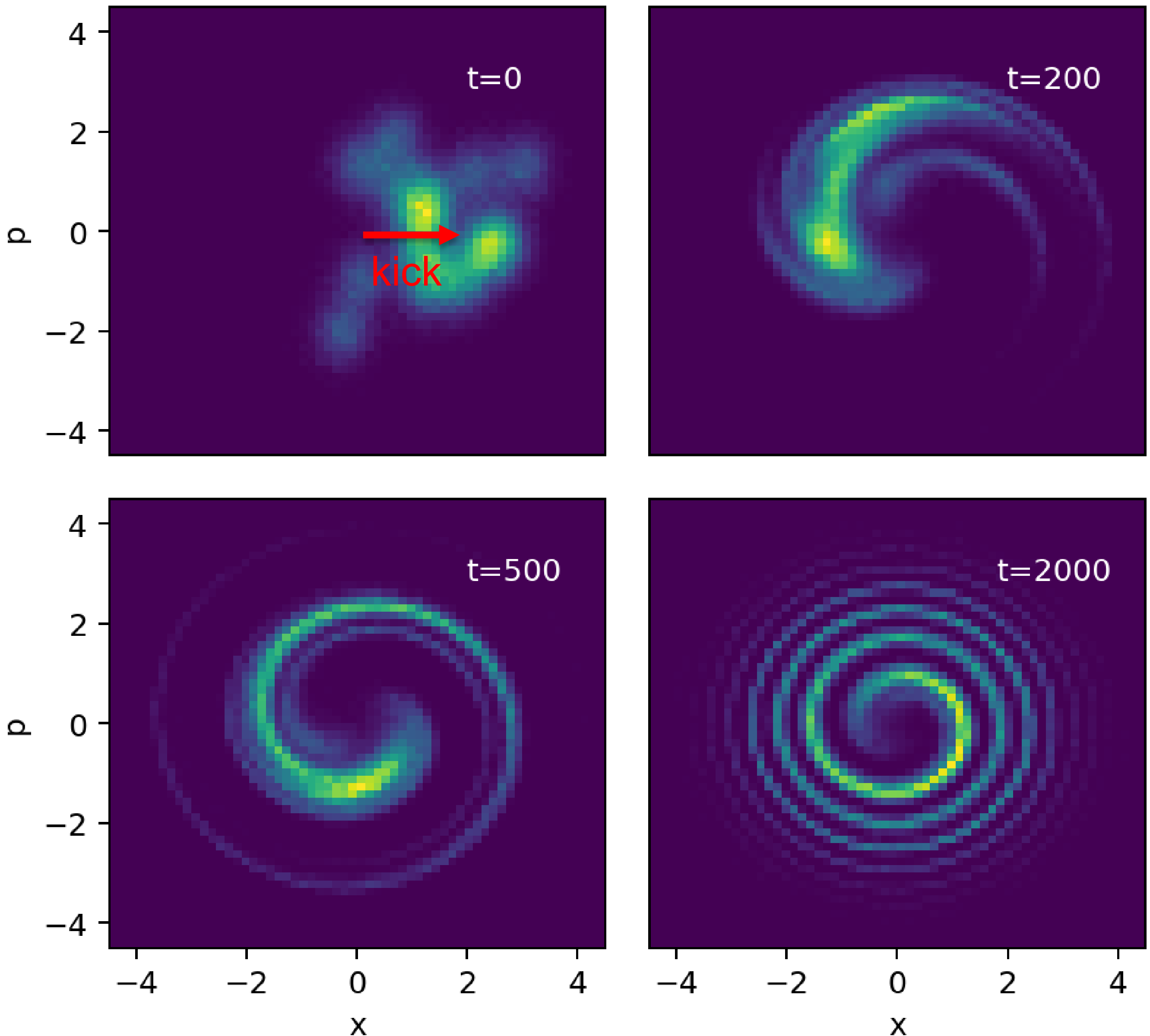
$$\omega(J_0), \qquad J_0 = (x_0^2 + p_0^2)/2$$

- TBT(turn-by-turn) beam centroid data is function of initial (before kick) beam distribution

$$\langle x \rangle_t = \Re \int (x - ip)e^{i\omega t}\rho(x - x_0, p - p_0)dxdp$$

- *We may able to reconstruct beam phase-space using a BPM*

illustration of nonlinear decoherence (due to phase mixing)



Theoretical estimation of the marginal beam profile

- If assume
  1. slowly varying betatron frequency over the beam area s.t.

$$\omega(\Delta J) = \mu_0 + \mu_1 \Delta J \cdots \qquad \Delta J \equiv J - J_0$$

  2. large kick strength s.t. $J_0/\epsilon \gg 1$

- Marginal beam profile (along the kick angle) can be analytically expressed

$$\lambda_\theta(x) = 2\frac{|\mu_1|}{\pi}\Re\sum_{t=0}^{T} e^{i\theta} e^{-i\left(\mu_1\sqrt{2J_0}x+\mu_0\right)t}\langle x\rangle_t - \frac{|\mu_1|}{\pi}\sqrt{2J_0}\cos^2\theta$$

$$p_0\cos\theta = x_0\sin\theta$$

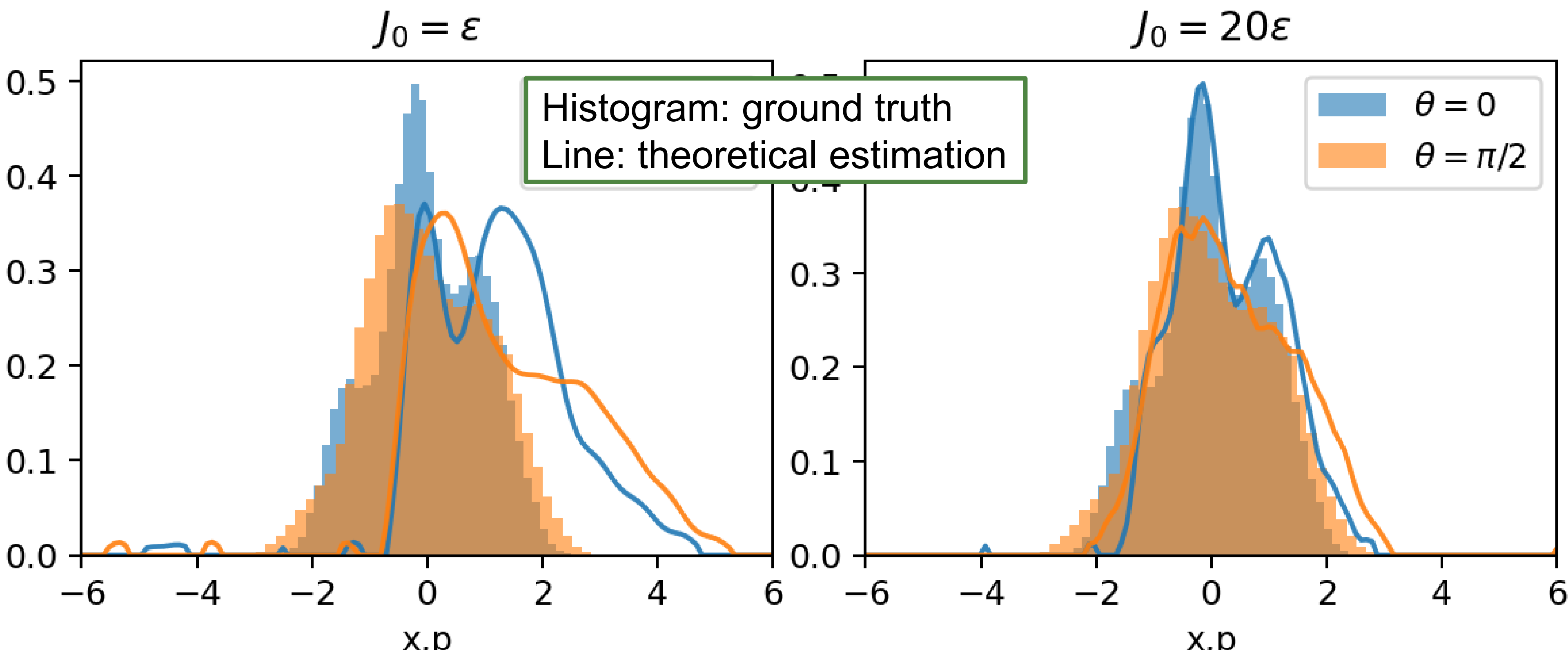
$$\lambda_\theta(x) \equiv \int \rho_\theta(x,p)\,dp$$

$$\rho_\theta(x,p) = \rho\left(x\cos\theta - p\sin\theta, p\cos\theta + x\sin\theta\right)$$

however, the large kick strength assumption can be limited due to beam pipe aperture

- Illustration on a toy model:

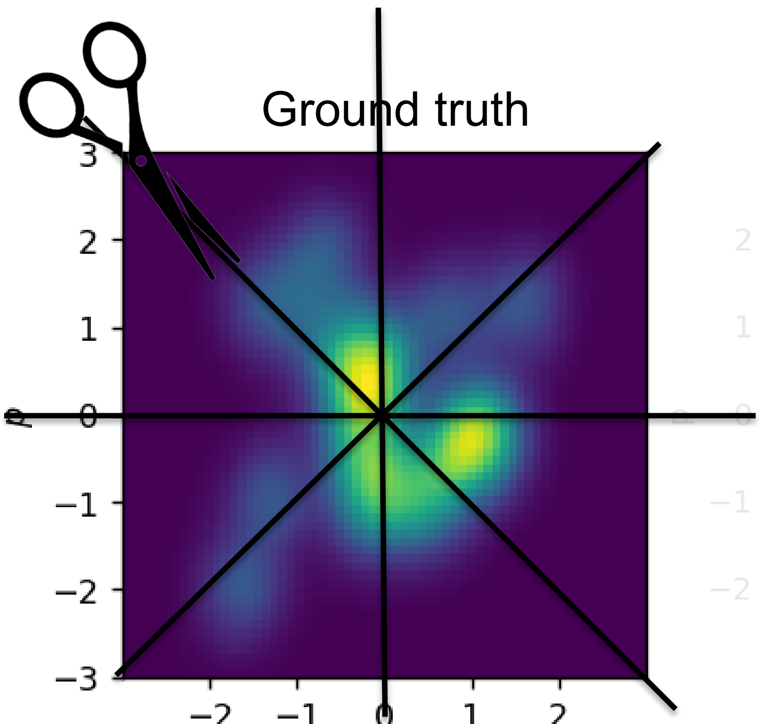
$$\omega = \omega_0 + \omega_1 J + \omega_2 \frac{J^2}{2}$$

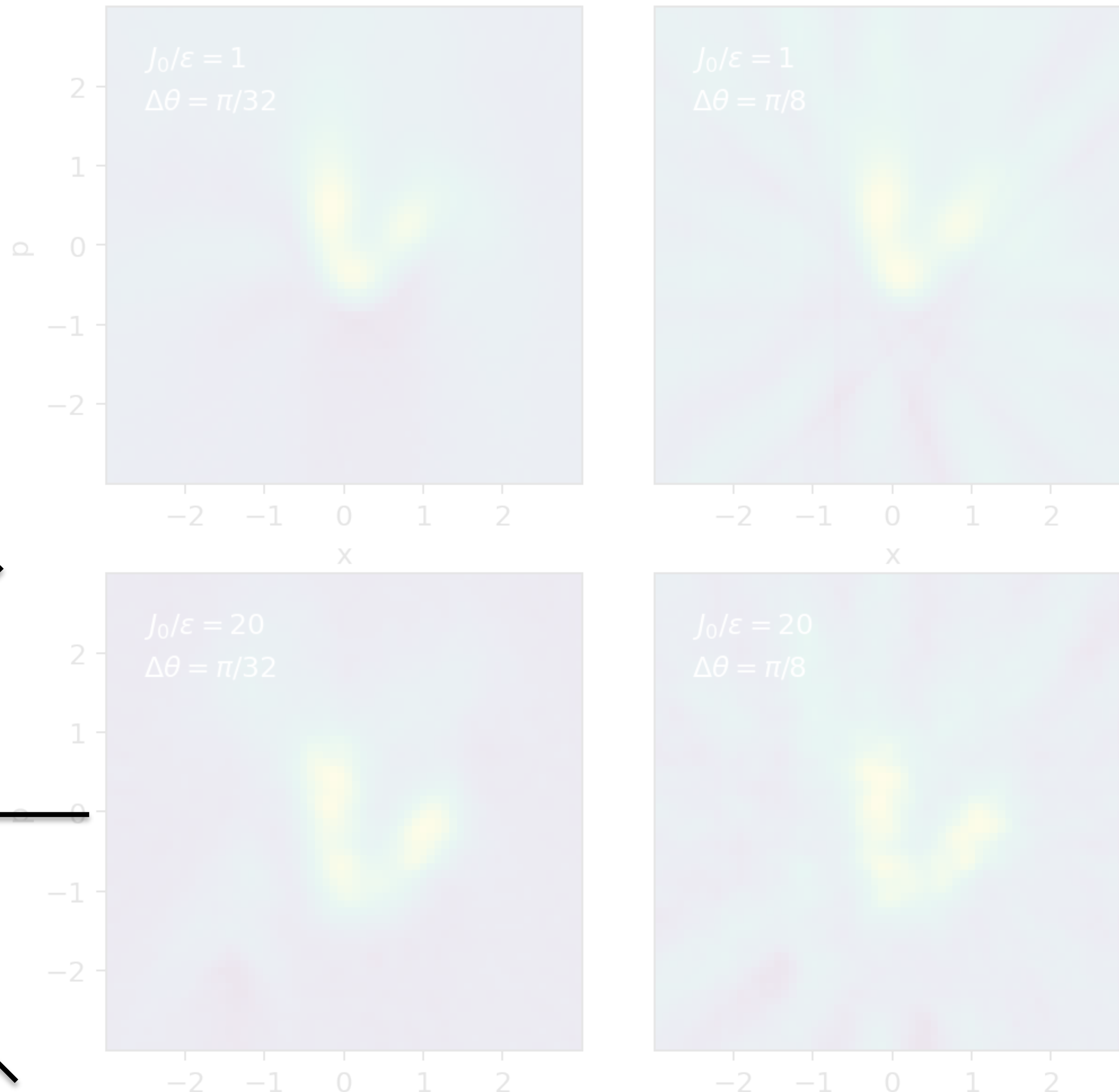


Inverse Radon transformation (an algebraic tomography method) using theoretically estimated marginal profiles

Inverse Radon transformation (an algebraic method) using theoretically estimated marginal profiles

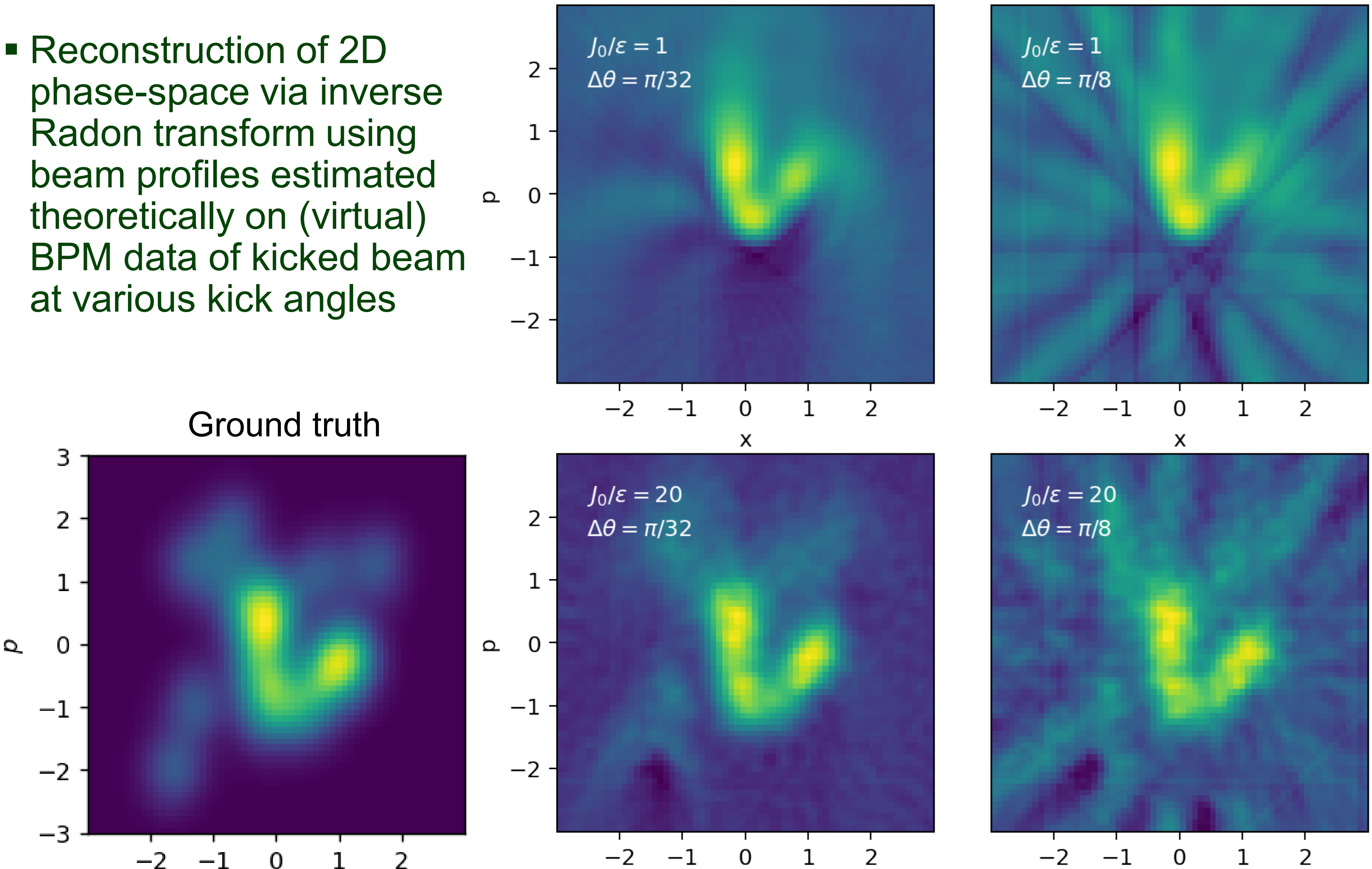
- Reconstruction of 2D phase-space via inverse Radon transform using beam profiles estimated theoretically on (virtual) BPM data of kicked beam at various kick angles





Inverse Radon transformation (an algebraic method) using theoretically estimated marginal profiles

- Reconstruction of 2D phase-space via inverse Radon transform using beam profiles estimated theoretically on (virtual) BPM data of kicked beam at various kick angles



Summary of inverse Radon with theoretical profile estimation

- Requires knowledge of optics parameters
  - transformation from physics to normal coordinate
  - oscillation frequency at kick action
  - nonlinear frequency detuning at kick action
  - kick action and angle in normal coordinate

- Requires strong kicks
  - generally not possible in presence of tight physical aperture

- Requires enough angular resolution
  - needs many kicks of different angles

Gaussian Mixture Model (GMM)

use Gaussian Mixture Model (GMM)

- TBT beam centroid is analytically integrable for isotropic Gaussian

$$\langle x \rangle_{\mathcal{N},t} = \Re \int (x - ip)\, e^{i\omega t} \mathcal{N}\left(\boldsymbol{x} - \boldsymbol{x}_0, \sigma \boldsymbol{I}\right)\, dxdp$$

$$= \frac{x_0\left(1-\tau^2\right) + 2p_0\tau}{\left(1+\tau^2\right)^2} \exp\left(-\frac{J_0}{\epsilon}\frac{\tau^2}{1+\tau^2}\right)\cos\Psi$$

$$-\frac{2x_0\tau - p_0\left(1-\tau^2\right)}{\left(1+\tau^2\right)^2} \exp\left(-\frac{J_0}{\epsilon}\frac{\tau^2}{1+\tau^2}\right)\sin\Psi$$

$$\omega(\Delta J) = \mu_0 + \mu_1 \Delta J \cdots$$

$$\Psi \equiv \mu_0 t - \frac{(J_0/\epsilon)\tau^3}{1+\tau^2}$$

$$\tau \equiv \epsilon \mu_1 t$$

- Therefore, one can model the beam density by GMM

$$\rho_{GMM}(x,p) = \frac{1}{G}\sum_{g=1}^{G} \mathcal{N}\left(\boldsymbol{x} - \boldsymbol{m}_g, \sigma_g \boldsymbol{I}\right)$$

$$\langle x \rangle_{GMM,t} = \Re \int (x - ip)\, e^{i\omega t} \rho_{GMM}\left(x - x_0, p - p_0\right)\, dxdp$$

use Gaussian Mixture Model (GMM)

- Virtual BPM data $\langle x \rangle_{BPM,k,t}$ is generated from the toy model

$$\omega = \omega_0 + \omega_1 J + \omega_2 \frac{J^2}{2}$$

- Fit parameters by minimizing

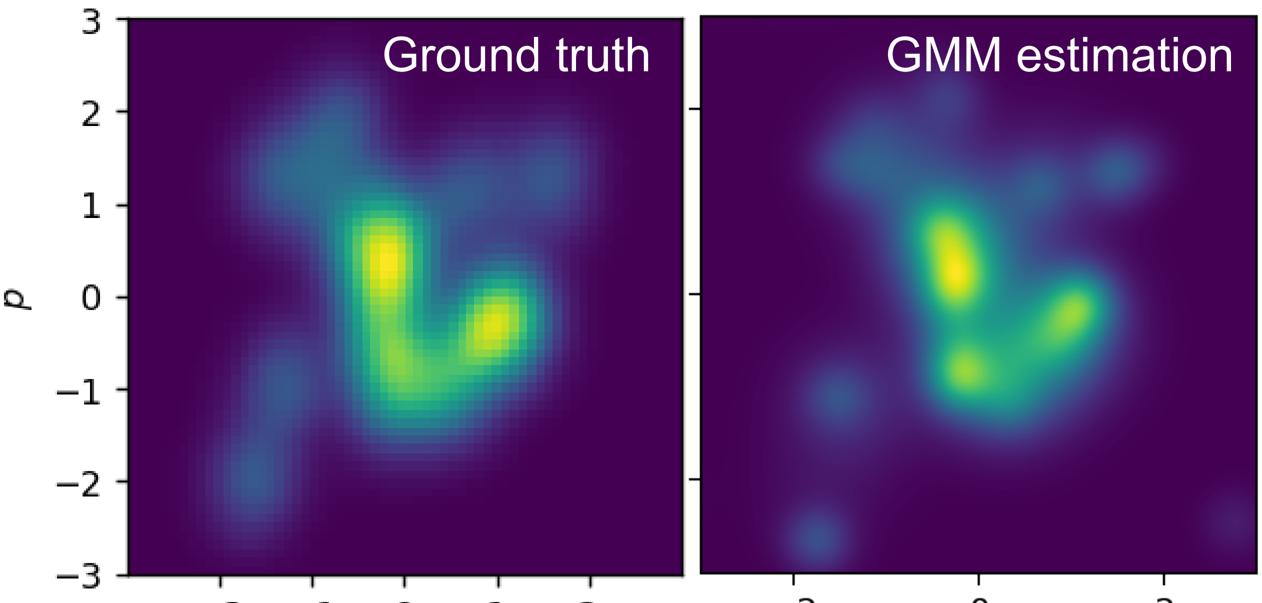
$$\sum_{k=1}^{K} \sum_{t=0}^{T} \left(\langle x \rangle_{BPM,k,t} - \langle x \rangle_{GMM,k,t} \right)^2 + \alpha$$

some regularization loss

- $K(=8)$ kick actions from $[2\epsilon, 4\epsilon]$ and kick angles equally spaced in $[0, \pi]$

In order to reduce computational complexity of optimization, optics parameters are first optimized using single Gaussian kernel model and global optimization (differential evolution method).

Then, all parameters are optimized through Nelder-Mead method.



Summary of GMM

- Computational complexity
  - There are too many parameters to fit :
    - multiple Gaussian kernels (used 100),
    - optics parameters (e.g. $\mu_0, \ \mu_1$),
    - kick strengths and angles
  - Using a black-box optimization tool took 12 hours using single core
  - hard to estimate model uncertainty due to computational complexity
  - Maybe relieved with GPU and differentiable implementation.
- Resolution is limited by number of Gaussian kernels.
- May have better fit for optics parameter (compared to single Gaussian beam model)

Differentiable Particle simulation Model (DPM)

use Differentiable Particle Model

- Use particle model and simple differentiable simulation model
  - Motivated by [1] to reduce computational complexity with
  - Simple betatron frequency model:

$$\omega = \mu_{0,k} + \mu_1 \Delta J$$

  - Particle locations, optics parameters are updated through gradient decent of the objective minimizing difference between prediction and true BPM TBT data with 8 different kicks *~corresponds to negative log likelihood*

$$\sum_{k=1}^{K}\sum_{t=0}^{T}\left(\langle x\rangle_{BPM,k,t} - \langle x\rangle_{PM,k,t}\right)^2 + \alpha$$

some regularization loss

  - Approximate Bayesian ensemble by anchoring (with L2 regularization) [2,3] model parameters to the randomized particle locations and prior optics parameter estimations using single Gaussian model. *~corresponds to negative log prior*

[1] Ryan Roussel et al, "Phase Space Reconstruction from Accelerator Beam Measurements Using Neural Networks and Differentiable Simulations"

[2] Ian Osband et al, "Randomized Prior Functions for Deep Reinforcement Learning", NeurIPS 2018

[3] Tim Pearce et al, "Bayesian Inference with Anchored Ensembles of Neural Networks, and Application to Exploration in Reinforcement Learning", ICML2018

use Differentiable Particle Model

- Use particle model and simple differentiable simulation model
  - Motivated by [1] to reduce computational complexity with
  - Simple betatron frequency model:

$$\omega = \mu_{0,k} + \mu_1 \Delta J$$

  - Particle locations, optics parameters are updated through gradient decent of the objective minimizing difference between prediction and true BPM TBT data with 8 different kicks *~corresponds to negative log likelihood*

$$\sum_{k=1}^{K} \sum_{t=0}^{T} \left(\langle x \rangle_{BPM,k,t} - \langle x \rangle_{PM,k,t} \right)^2 + \alpha$$

some regularization loss

  - Approximate Bayesian ensemble by anchoring (with L2 regularization) [2,3] model parameters to the randomized particle locations and prior optics parameter estimations using single Gaussian model. *~corresponds to negative log prior*

[1] Ryan Roussel et al, "Phase Space Reconstruction from Accelerator Beam Measurements Using Neural Networks and Differentiable Simulations"

[2] Ian Osband et al, "Randomized Prior Functions for Deep Reinforcement Learning", NeurIPS 2018

[3] Tim Pearce et al, "Bayesian Inference with Anchored Ensembles of Neural Networks, and Application to Exploration in Reinforcement Learning", ICML2018

use Differentiable Particle Model

- Use particle model and simple differentiable simulation model
  - Motivated by [1] to reduce computational complexity with
  - Simple betatron frequency model:

$$\omega = \mu_{0,k} + \mu_1 \Delta J$$

  - Particle locations, optics parameters are updated through gradient decent of the objective minimizing difference between prediction and true BPM TBT data with 8 different kicks *~corresponds to negative log likelihood*

$$\sum_{k=1}^{K} \sum_{t=0}^{T} \left(\langle x \rangle_{BPM,k,t} - \langle x \rangle_{PM,k,t} \right)^2 + \alpha$$

some regularization loss

  - Approximate Bayesian ensemble by anchoring (with L2 regularization) [2,3] model parameters to the randomized particle locations and prior optics parameter estimations using single Gaussian model. *~corresponds to negative log prior*

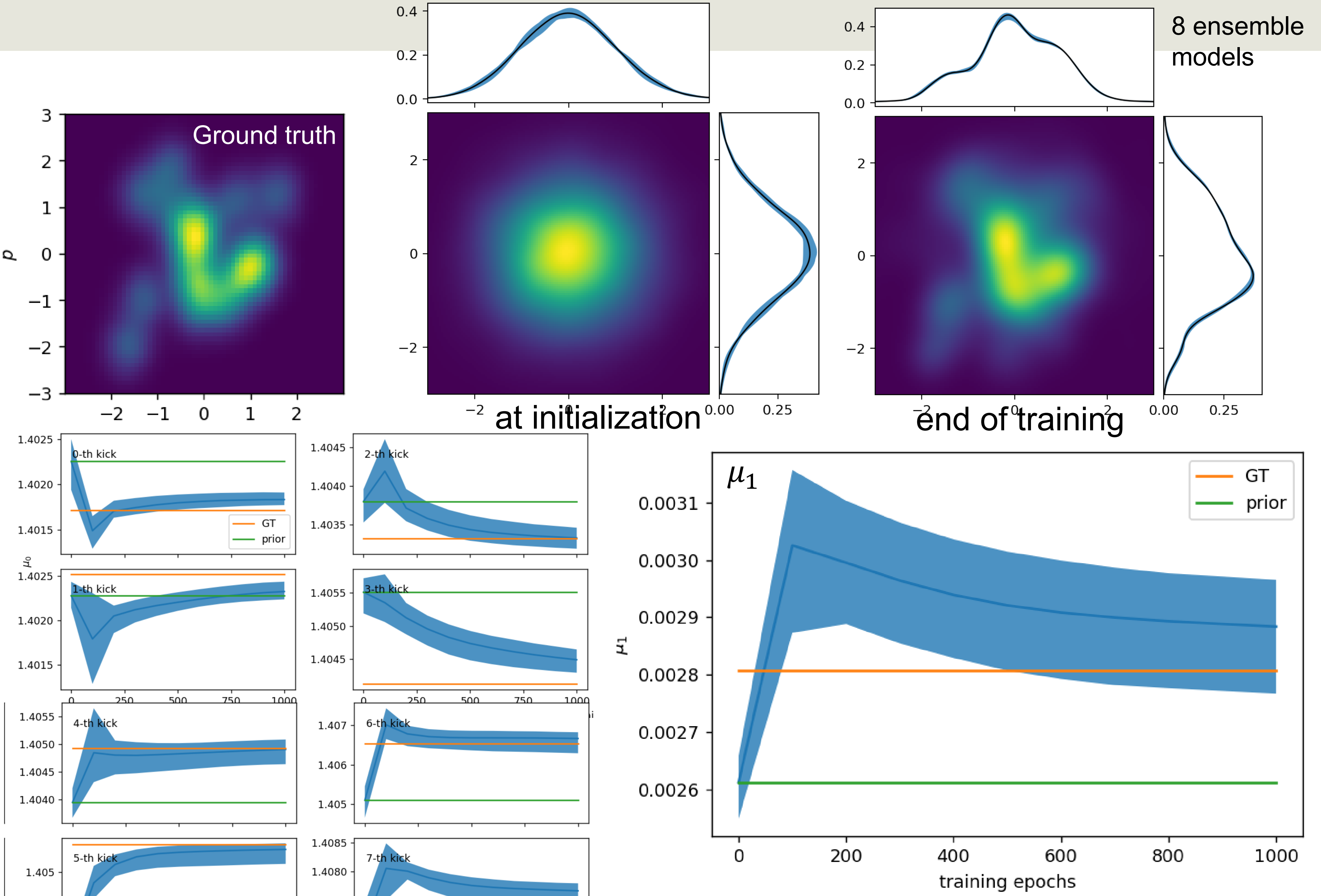
---

[1] Ryan Roussel et al, "Phase Space Reconstruction from Accelerator Beam Measurements Using Neural Networks and Differentiable Simulations"

[2] Ian Osband et al, "Randomized Prior Functions for Deep Reinforcement Learning", NeurIPS 2018

[3] Tim Pearce et al, "Bayesian Inference with Anchored Ensembles of Neural Networks, and Application to Exploration in Reinforcement Learning", ICML2018

use Particle Model



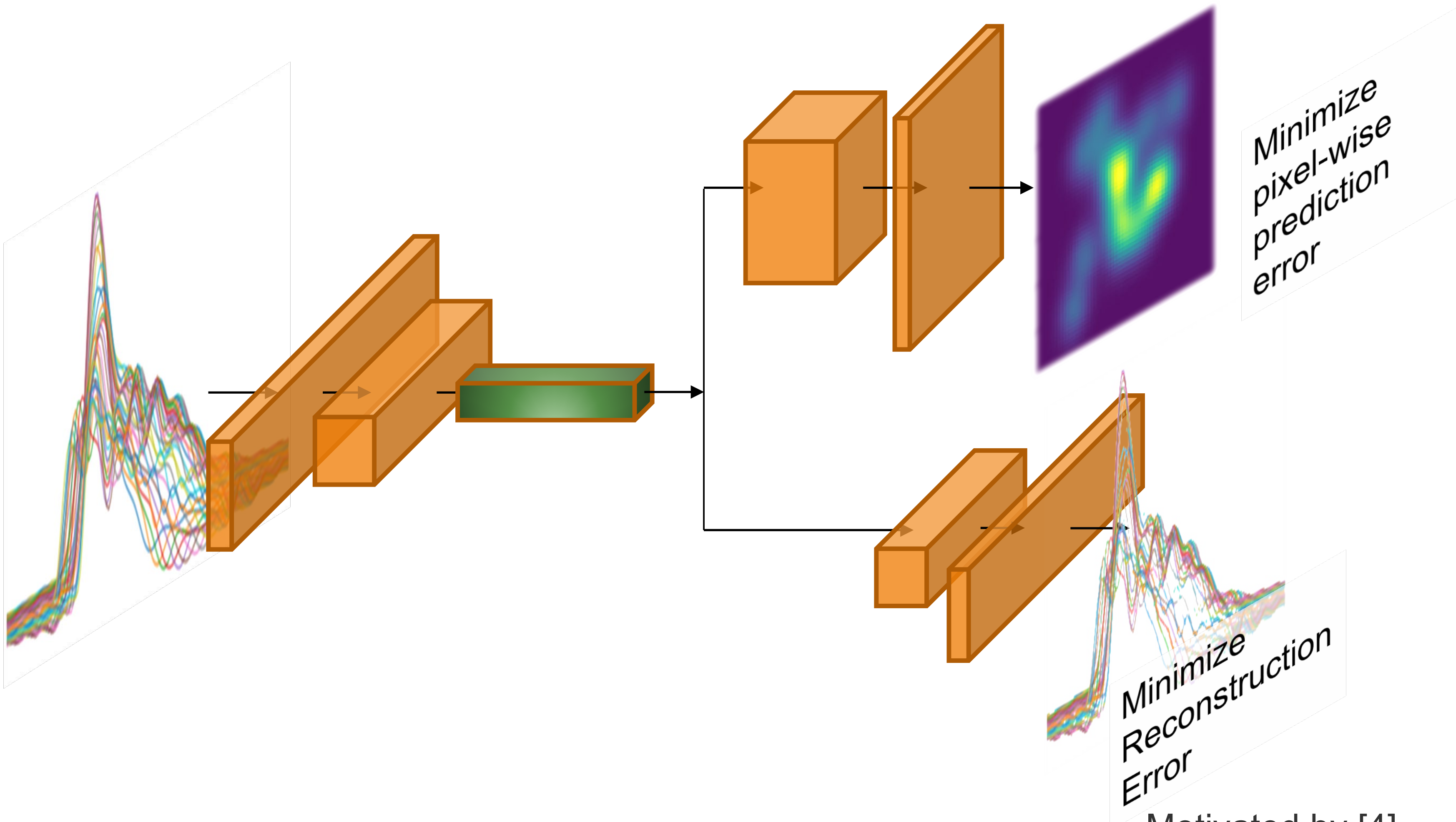
Summary of PM

- Computational complexity
  - With gradient decent (ADAM) optimization, it still took 10 hours (using one CPU core) for 8 models (~ 1h for each model)
  - Maybe relieved with implementation.
- UQ may be feasible due to faster training (compared to GMM without differentiable model)
- Better fit for optics parameter (compared to single Gaussian beam model)

Neural Network (NN) Model with supervised learning and model uncertainty

Use NN

- Input: theoretically estimated marginal profiles
- Output: beam density plot and reconstructed marginal profiles



[4] Alex Scheinker, “Adaptive machine learning”, information 2021

Motivated by [4]

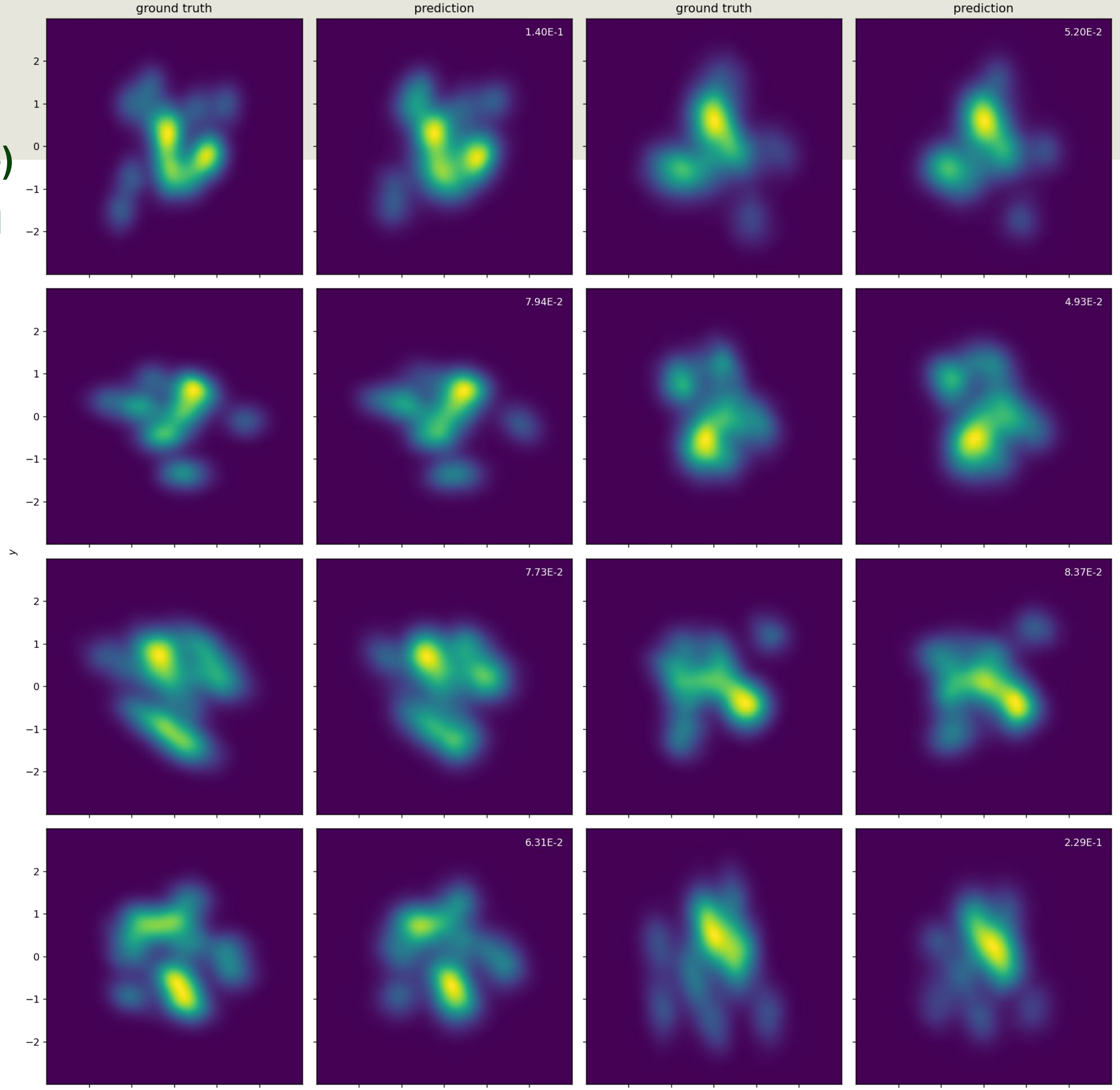
How training data generated

- Virtual BPM TBT data from toy-simulation-model

$$\omega = \omega_0 + \omega_1 J + \omega_2 \frac{J^2}{2}$$

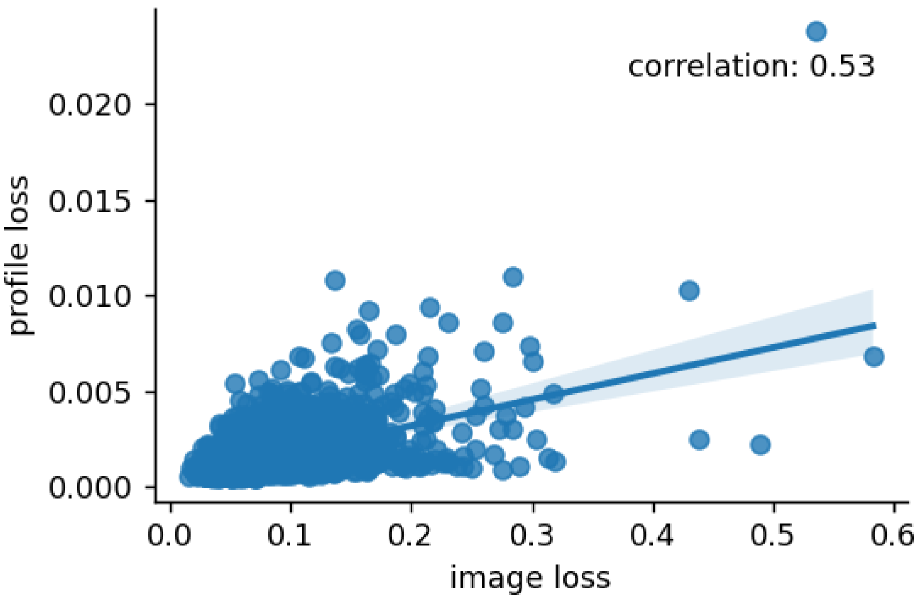
- Each data is generated using randomly sampled model parameters of simulation ("*domain randomization*" for "*sim-to-real*")
  - randomized model parameters need to cover the expected real machine behavior
    - » beam emittance $\epsilon \sim \mathcal{N}(1, \sigma_\epsilon)$
    - » frequency parameters $\omega_i \sim \mathcal{N}\left(\overline{\omega}_i, \sigma_{\omega_i}\right), \quad i \in [0-2]$
    - » kick strengths and angles $J_{0,k} \sim \mathcal{N}\left(\bar{J}_{0,k}, \sigma_{J_0}\right),\ \theta_k \sim \mathcal{N}(\bar{\theta}_k, \sigma_\theta), \quad k \in [1,8]$

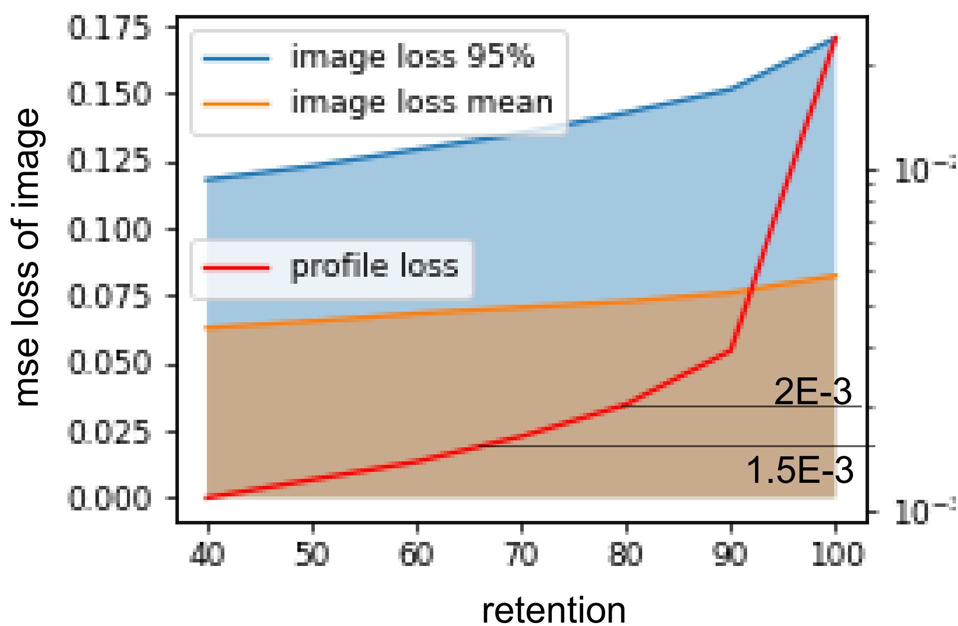
Some of random (except the 1st one) prediction samples



UQ using reconstruction loss of input profile

- UQ using ensemble
  - As it is expected that the model predictions in the training distribution are close each other while predictions in out-of-distribution show large variance for each model
- The reconstruction loss can also be used for UQ
  - As it is also expected to be small in the training distribution while large in out-of-distribution

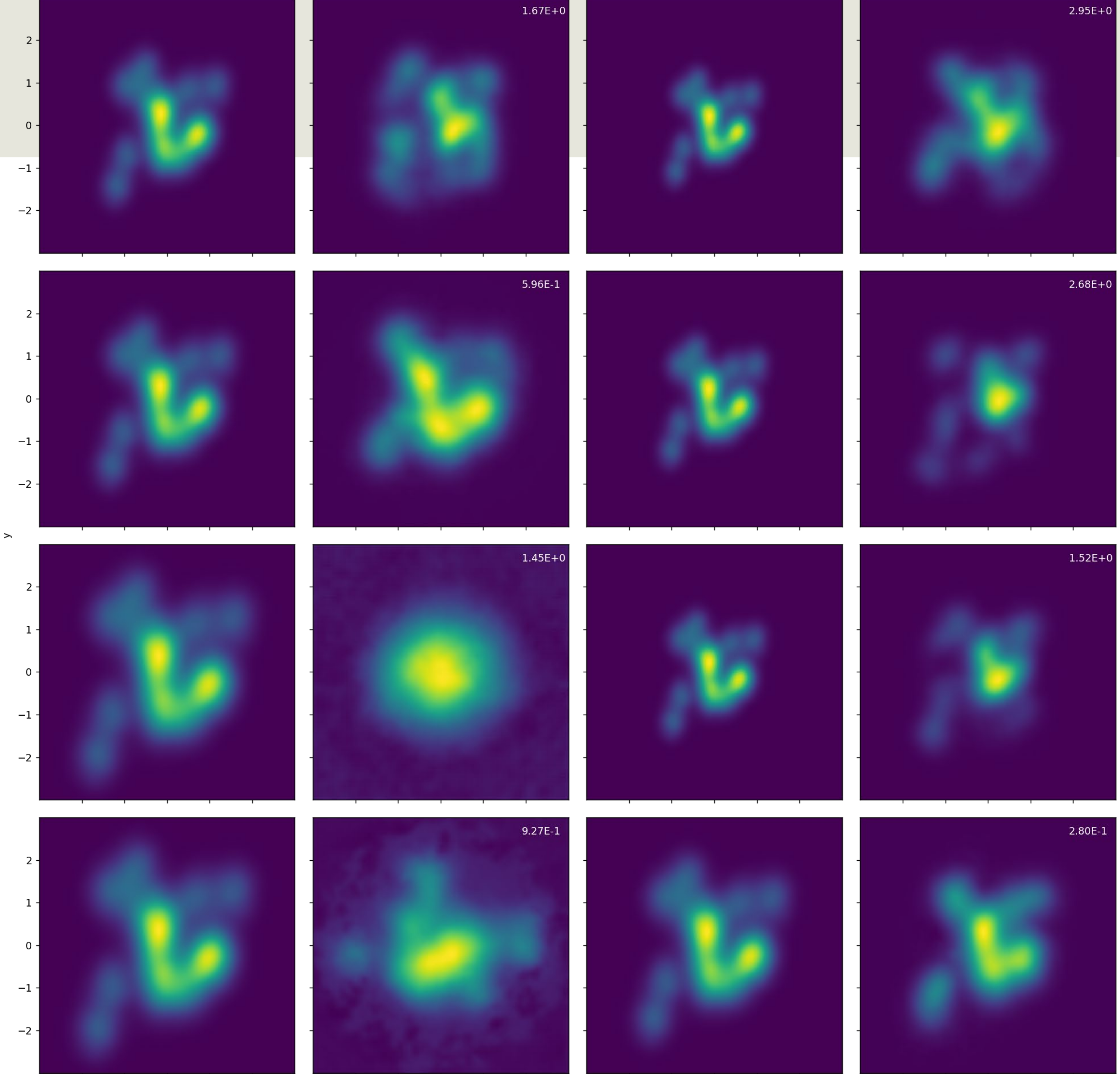




Out of distribution samples

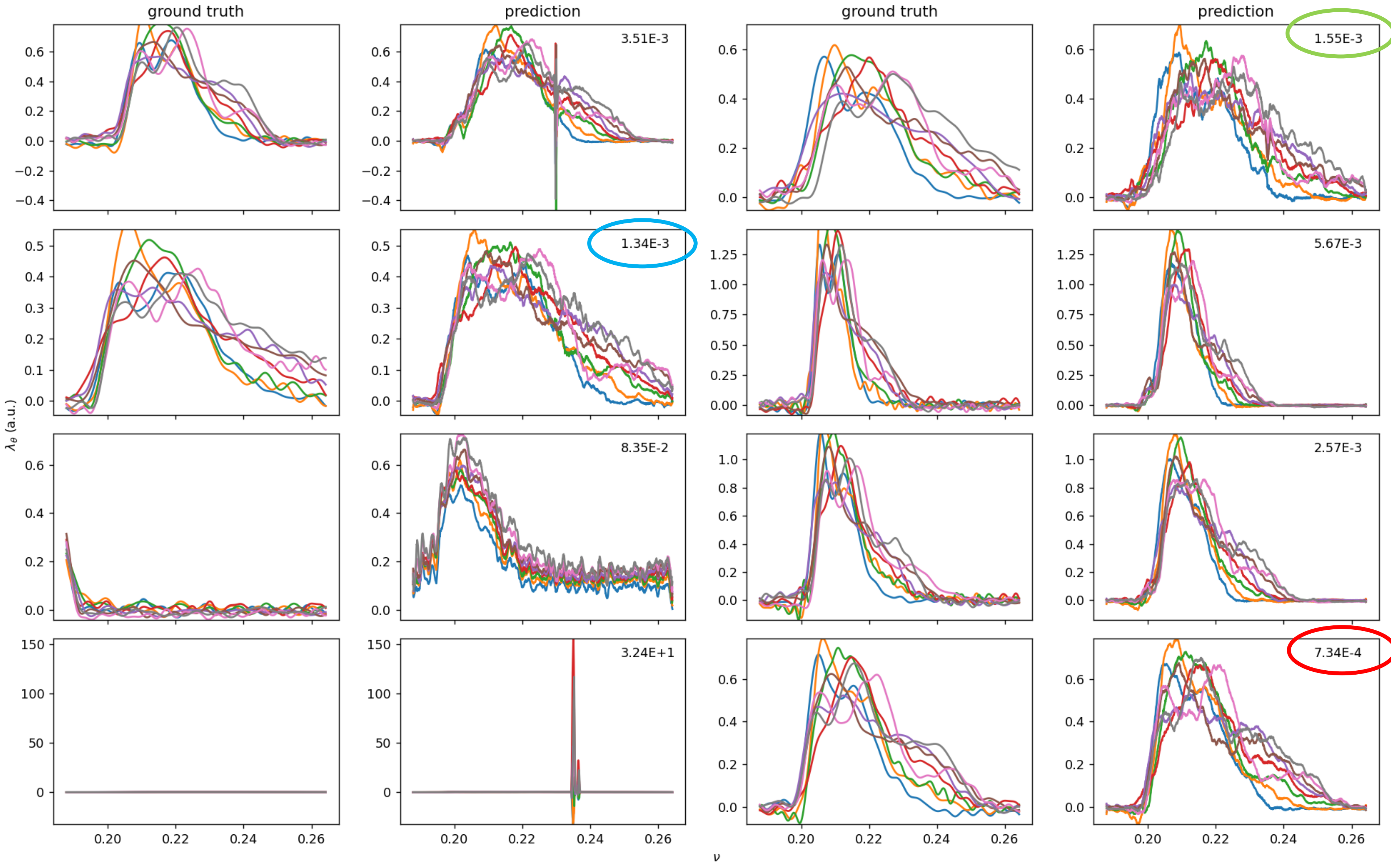
- OOD samples are generated by sampling from 4 times larger parameter variances (than the variance used for the training data generation)

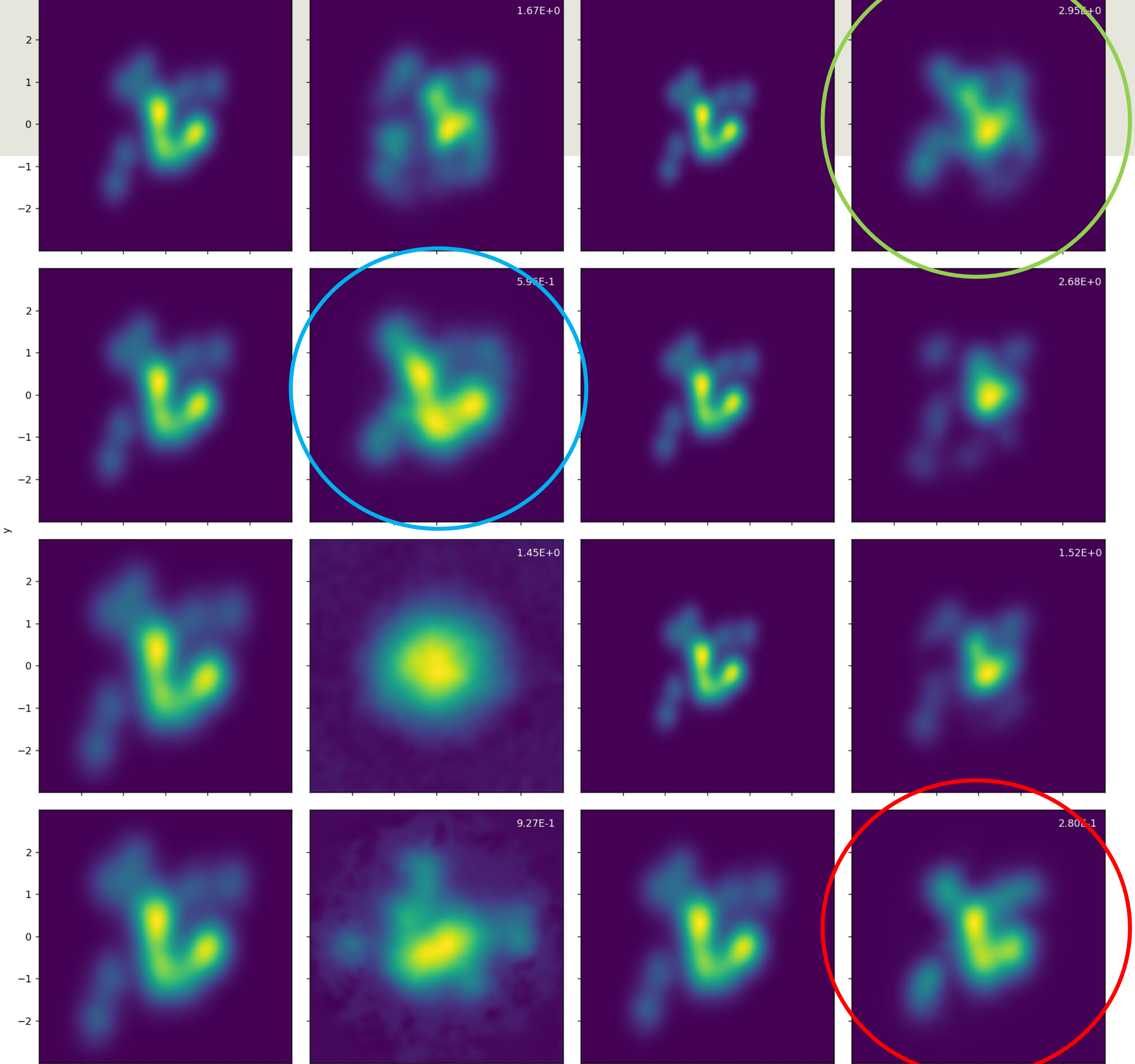
  - » beam emittance $\epsilon \sim \mathcal{N}(1, 4\sigma_\epsilon)$
  - » frequency parameters $\omega_i \sim \mathcal{N}\left(\bar{\omega}_i, 4\sigma_{\omega_i}\right), \quad i \in [0-2]$
  - » kick strengths and angles $J_{0,k} \sim \mathcal{N}\left(\bar{J}_{0,k}, 4\sigma_{J_0}\right), \ \theta_k \sim \mathcal{N}(\bar{\theta}_k, 4\sigma_\theta), \quad k \in [1,8]$



1.67E+0
2.95E+0
5.96E-1
2.68E+0
1.45E+0
1.52E+0
9.27E-1
2.80E-1
y
2
1
0
−1
−2

profile reconstruction losses




1.67E+0
2.95E+0
5.96E-1
2.68E+0
1.45E+0
1.52E+0
9.27E-1
2.80E-1
y
2
1
0
−1
−2

Summary of NN

- Computational complexity is no longer a problem but the collection of training data and training time can be problematic

- The reconstruction loss looks promising alternative method for UQ

- Domain randomization is used for “sim-to-real” adaption.
  - However, (as is the case of GMM or PM) the simulation model may not enough for real machine

Conclusion

Conclusion

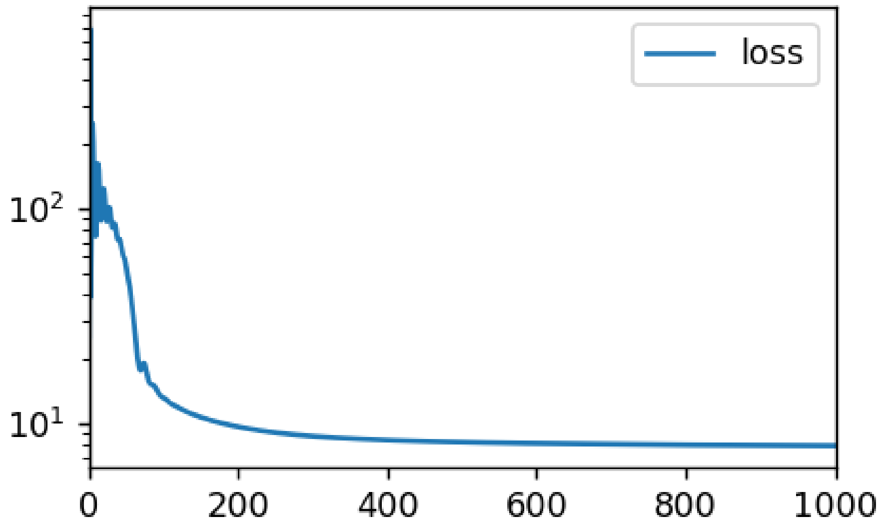
- TBT(turn-by-turn) beam centroid data is function of initial (before kick) beam distribution
  - Beam profile can be theoretically estimated when $J_0/\epsilon \gg 1$
    - » ML methods does not require large kick
- We investigated various method for 2D phase-space tomography using kicked beam turn-by-turn (virtual) BPM data
- GMM, PM methods are promising as long as
  - Computational complexity problem can be improved through differentiable simulation model and GPU implementation
- NN method is promising as long as
  - Enough (simulational) training data that can represent the real machine can be collected
- In all cases, simple but well-representative simulational model plays important role

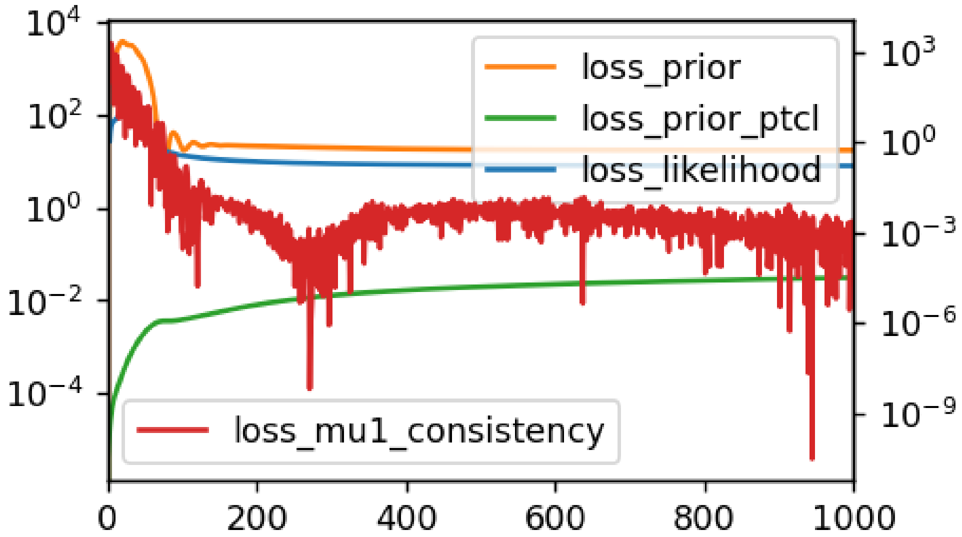
Acknowledgements

- Work supported by the Director of the Office of Science of the US Department of Energy under Cooperative Agreement DE-SC0000661, the State of Michigan and Michigan State University

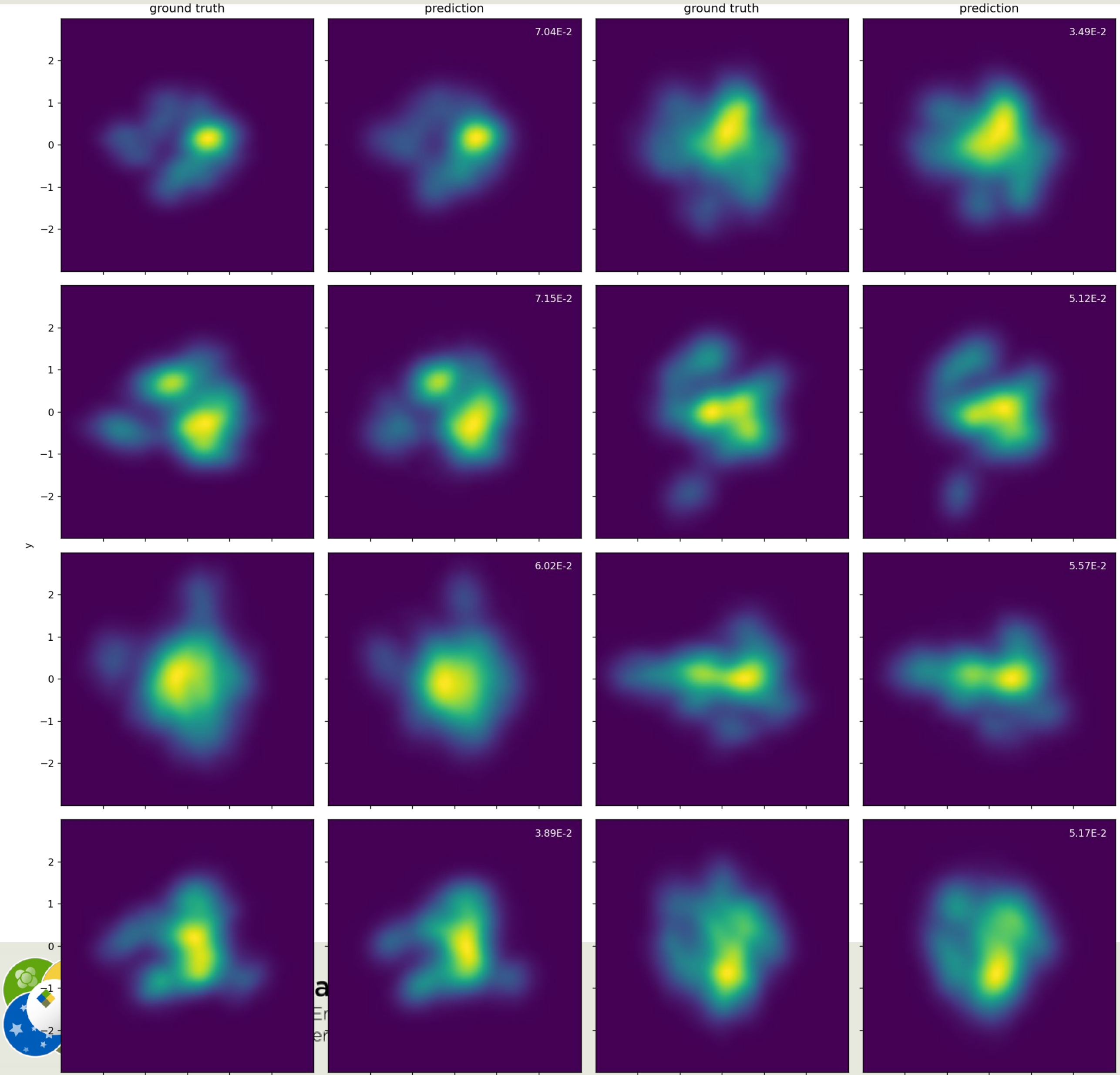
Backup slide

PM training loss

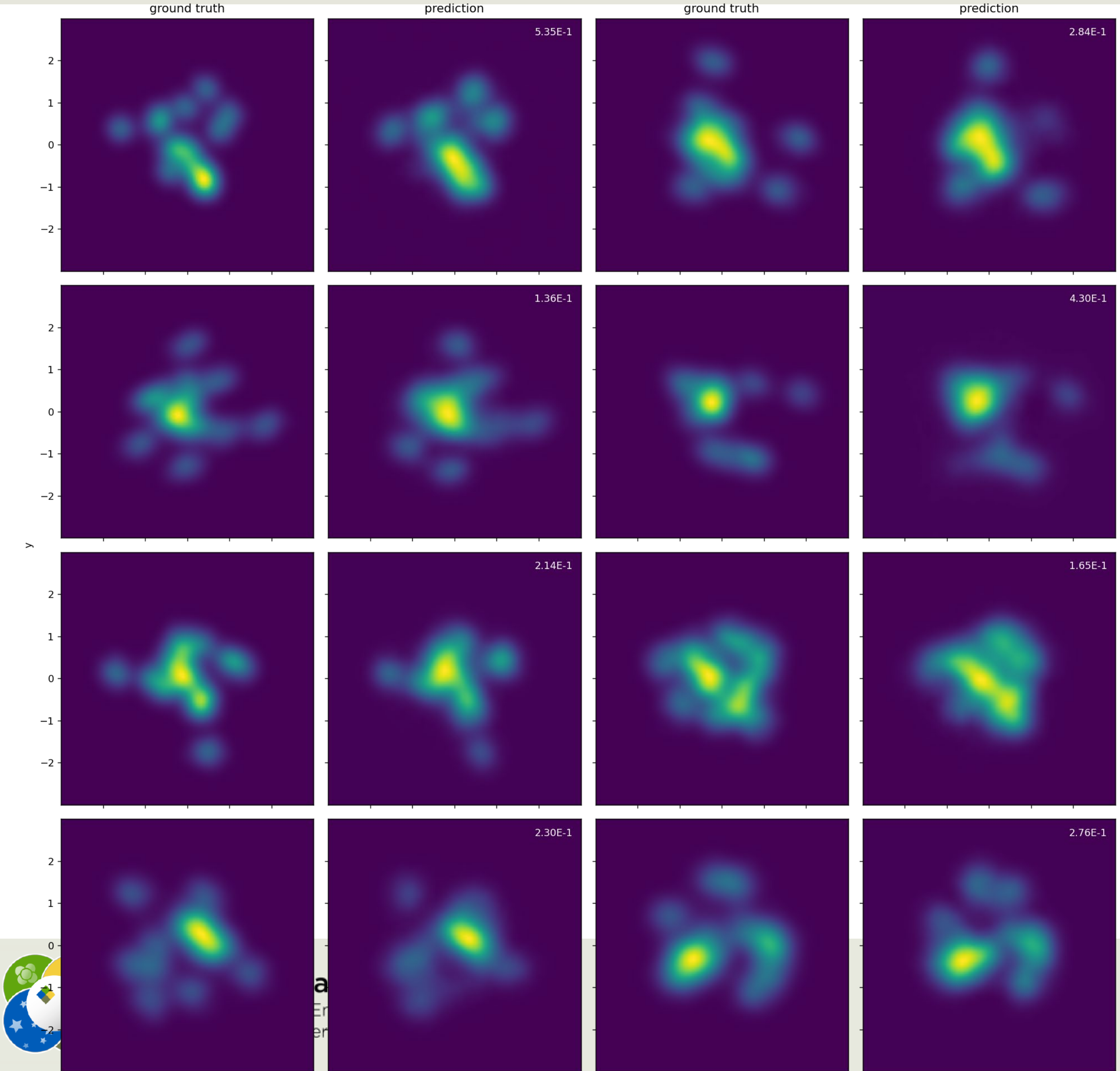




NN test samples of best profile loss



NN test samples of worst profile loss



FRIB