Machine Learning-Enhanced MOGA for Ultrahigh-Brightness Lattices

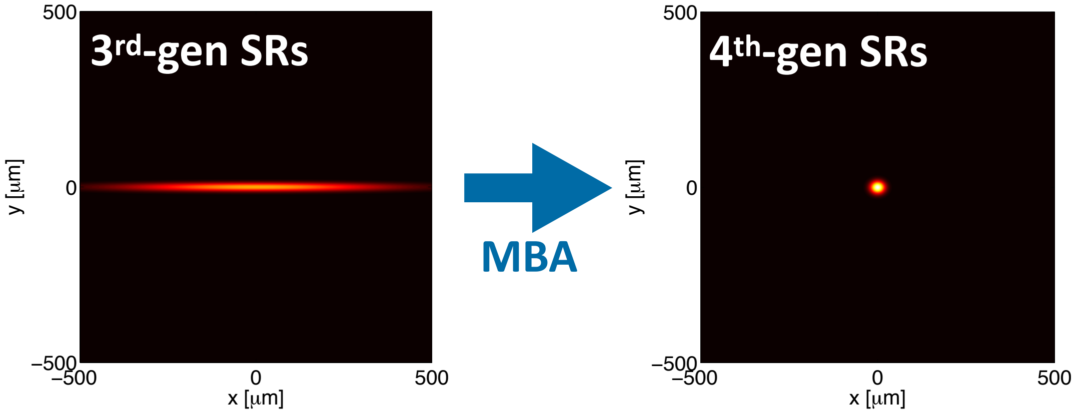
Simon C. Leemann

ALS Accelerator Physics, ALS & ATAP Divisions, Lawrence Berkeley National Laboratory

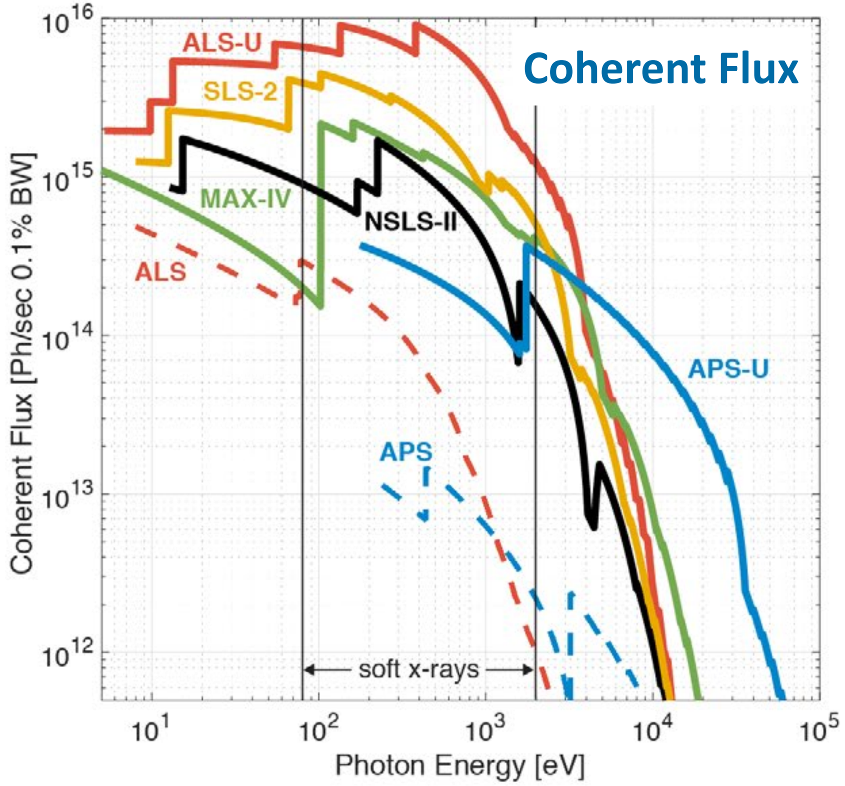
3rd Workshop on Low Emittance Lattice Design, ALBA, Barcelona, Spain, June 26-29, 2022

Introduction

- **4th-generation storage rings** (4GSRs) leverage MBA lattices to render ultra-high brightness with large coherent fraction

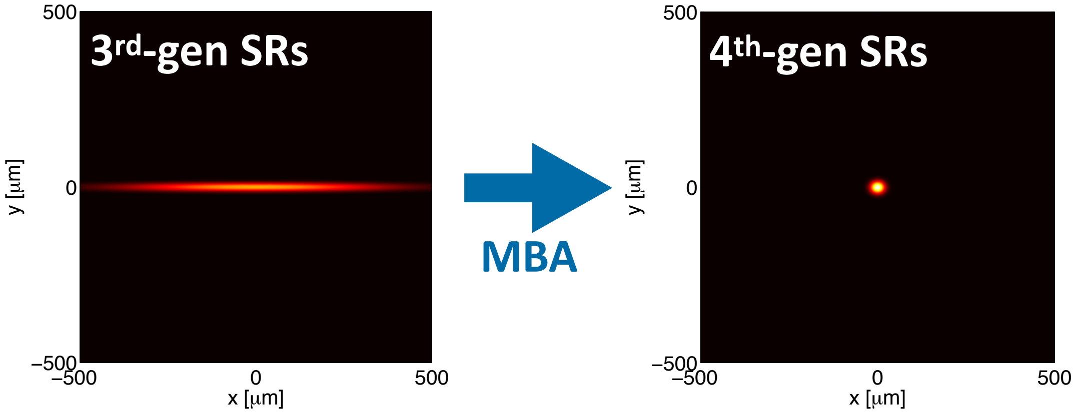


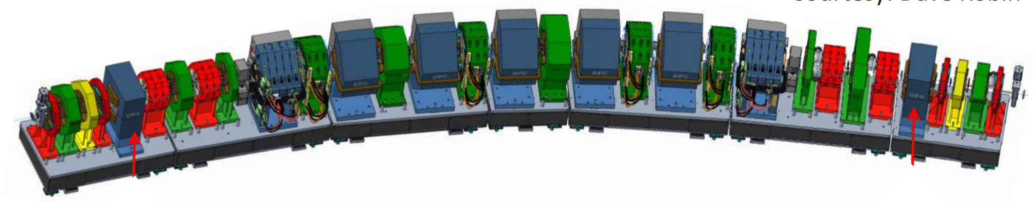
Courtesy: Dave Robin



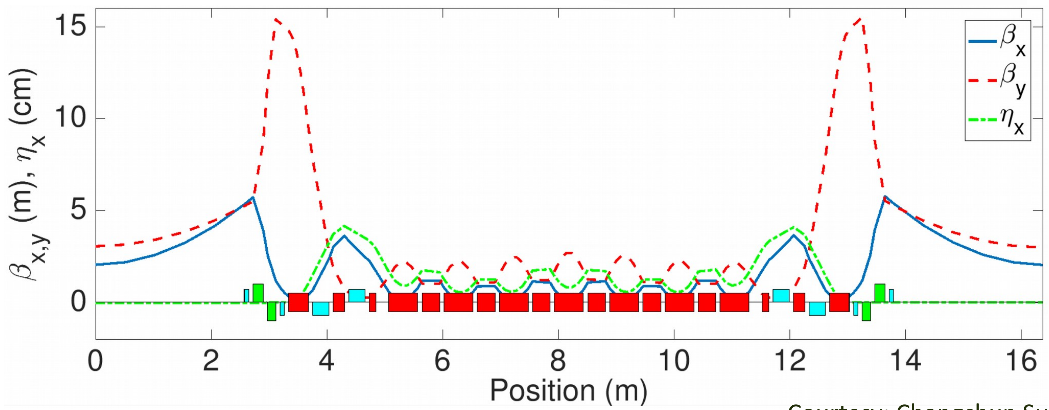
Introduction: The Problem

- **4th-generation storage rings** (4GSRs) leverage MBA lattices to render ultra-high brightness with large coherent fraction
- **MBA lattices** are very challenging: dense & exploit very strong focusing → drives strong chromatic & higher-order corrections
- Solutions often highly nonlinear & involve many degrees of freedom (DoF) → demanding **optimization**:
    - tough objectives, many of which often in direct competition
    - large number of parameters, many boundary constraints





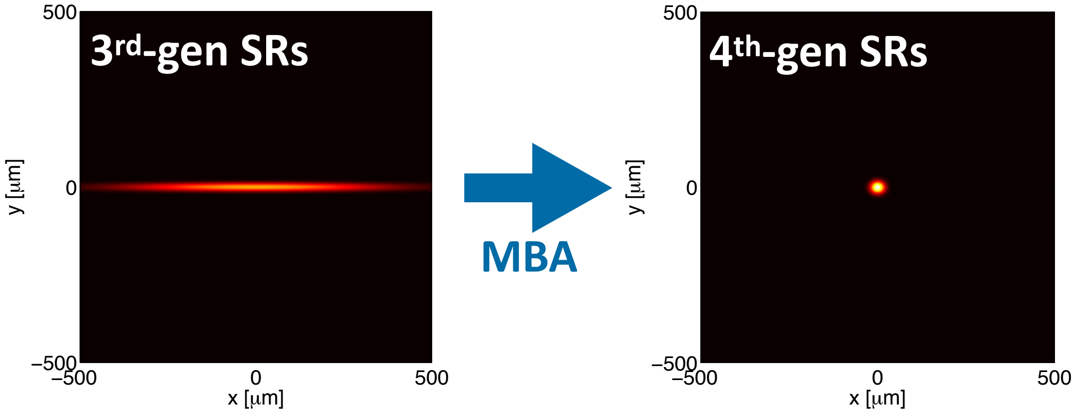
Courtesy: Dave Robin



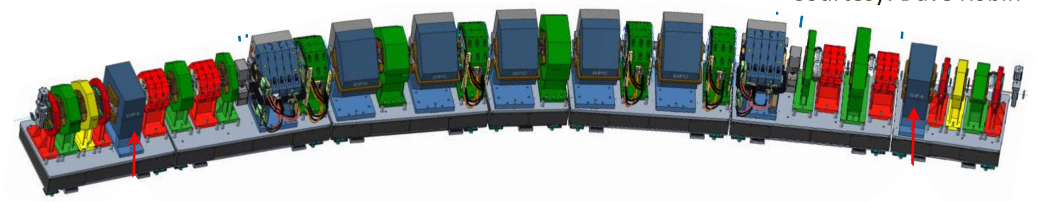
Courtesy: Changchun Sun

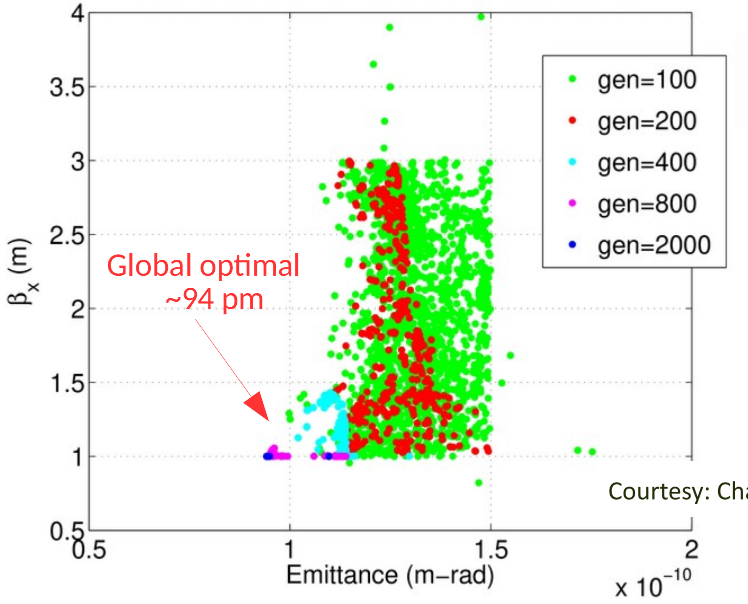
Introduction: The Problem

- **4th-generation storage rings** (4GSRs) leverage MBA lattices to render ultra-high brightness with large coherent fraction
- **MBA lattices** are very challenging: dense & exploit very strong focusing ➔ drives strong chromatic & higher-order corrections
- Solutions often highly nonlinear & involve many degrees of freedom (DoF) ➔ demanding **optimization**:
  - tough objectives, many of which often in direct competition
  - large number of parameters, many boundary constraints
- **Multi-objective genetic algorithms (MOGA)** are highly successful at such optimization & have become tool of choice
- However, stochastic nature is inherent weakness ➔ need to evaluate vast number of lattice candidates, most ultimately rejected
- Do not want to artificially limit DoF, search ranges, or make many initial assumptions about attractive solutions ➔ *so what can we do?*



Courtesy: Dave Robin

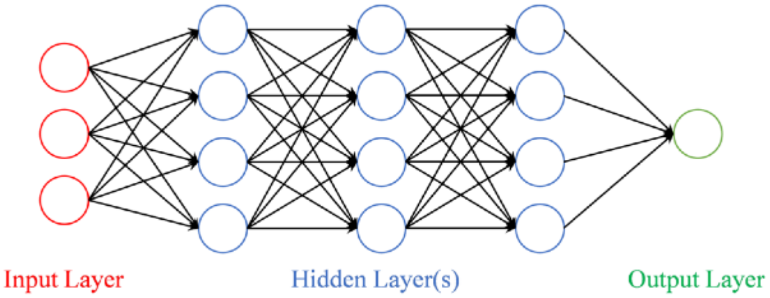


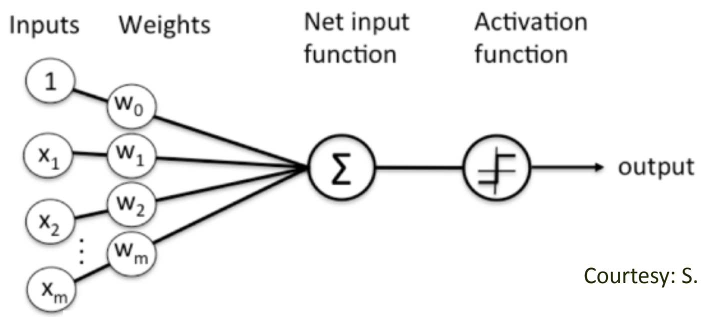


Courtesy: Changchun Sun

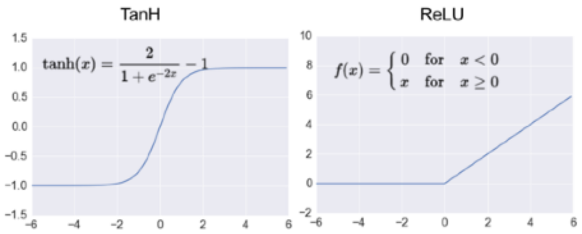
Introduction: Machine Learning (ML) to the Rescue

- ML can be employed to render **neural networks (NNs)** → surrogate models used in lieu of computationally expensive evaluation (e.g. many-turn nonlinear tracking)
- Lattice candidate evaluation becomes near instantaneous → ideally, want to speed up MOGA without modifying MOGA/tracking tools or existing workflows & <u>without</u> sacrificing physics fidelity
- Previous attempts [1-3] have focused on various aspects, but we set out with a different emphasis:
  - Direct optimization of relevant physics quantities (ε_0, DA, MA)
  - Combined linear/nonlinear optimization involving all free parameters (quadrupoles & sextupoles)





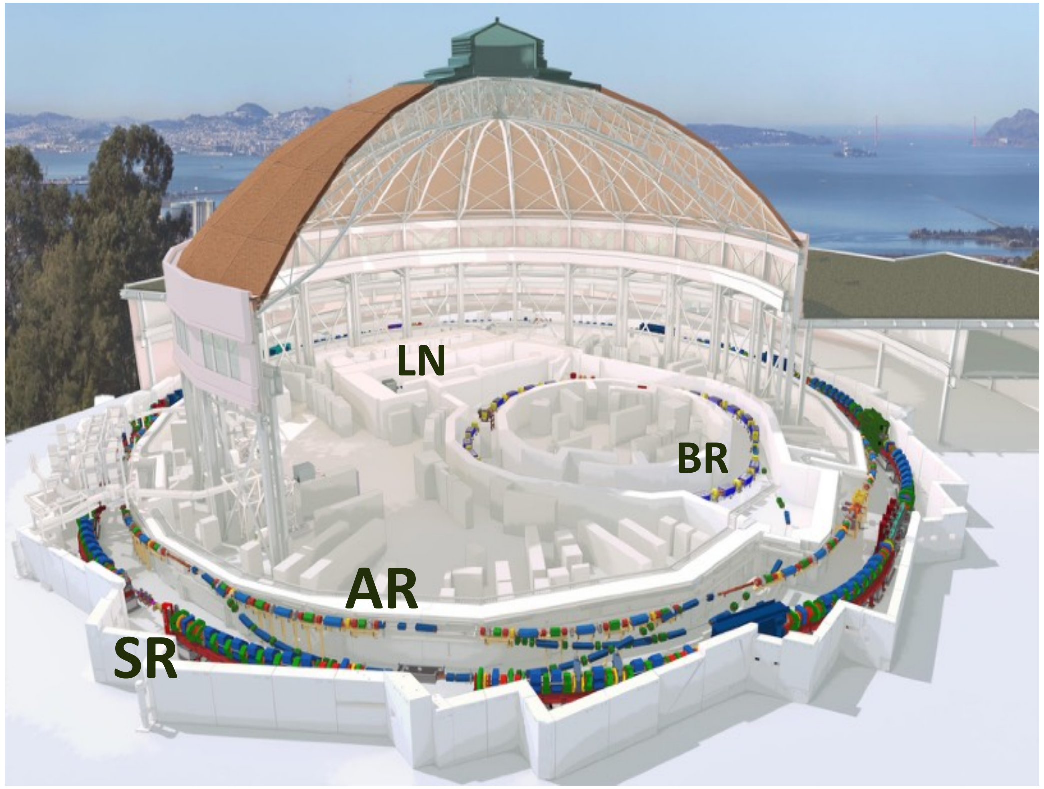
Courtesy: S. Liu

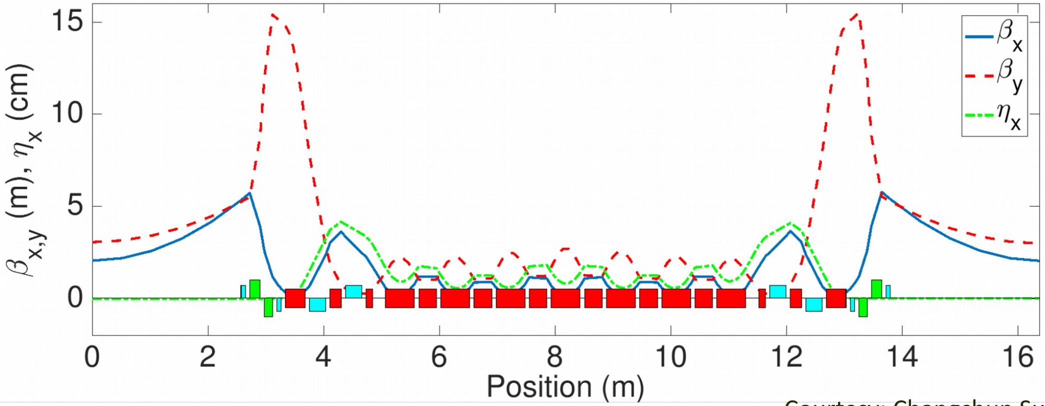


[1] M. Kranjčević, B. Riemann, A. Adelmann, A. Streun, PRAB **24** 014601, 2021.
[2] Y. Li, W. Cheng, L.Yu, R. Rainer, PRAB **21** 054601, 2018.
[3] J. Wan, P. Chu, Y. Jiao, PRAB **23** 081601, 2020.

ALS-U as a Test Case

- **ALS-U storage ring** (SR) calls for a challenging 9BA in order to achieve ≈75 pm rad (round beam) at 2 GeV in <200 m circumference
- But retain booster (BR) & linac (LN) ➔ build accumulator ring (AR) to damp & top off
- 9BA SR lattice tailored for highest soft x-ray brightness ➔ dense, strong, very strained

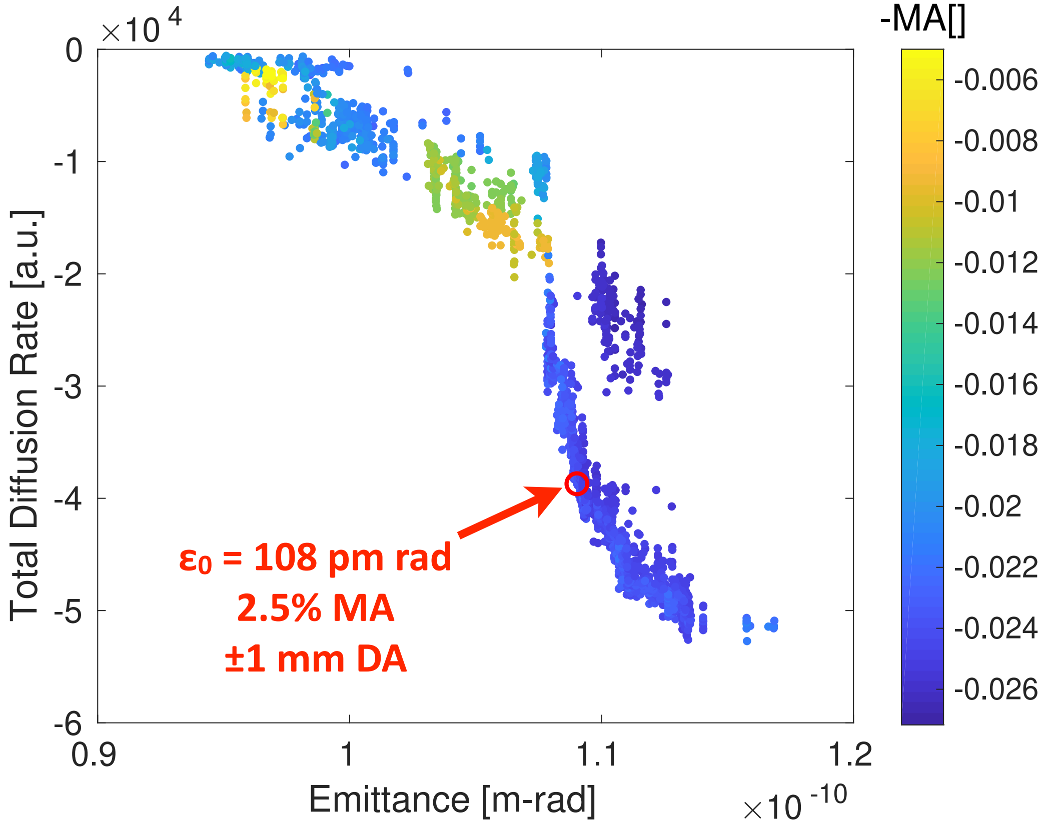




Courtesy: Changchun Sun

ALS-U as a Test Case

- **ALS-U storage ring** (SR) calls for a challenging 9BA in order to achieve ≈75 pm rad (round beam) at 2 GeV in <200 m circumference
- But retain booster (BR) & linac (LN) → build accumulator ring (AR) to damp & top off
- 9BA SR lattice tailored for highest soft x-ray brightness → dense, strong, very strained
- Highly staged **MOGA** approach resulted in
  - ±1 mm DA (on-axis swap-out injection with AR)
  - ≈1 hr lifetime (with 3HCs)

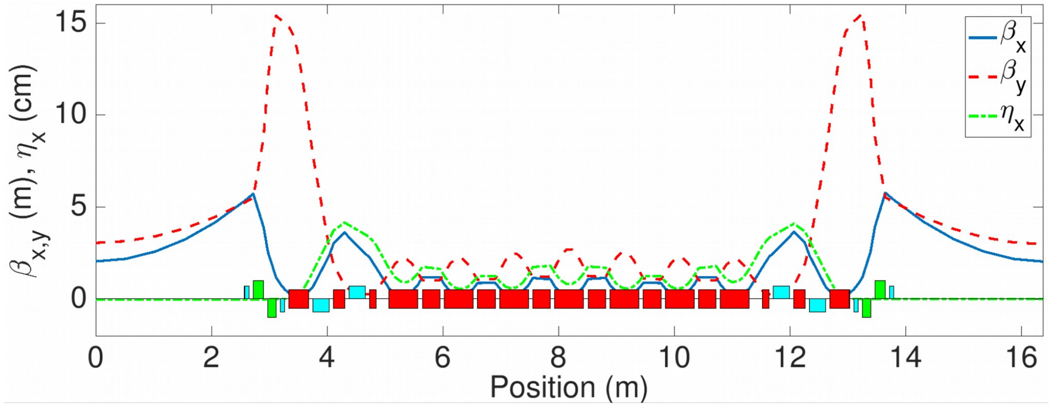


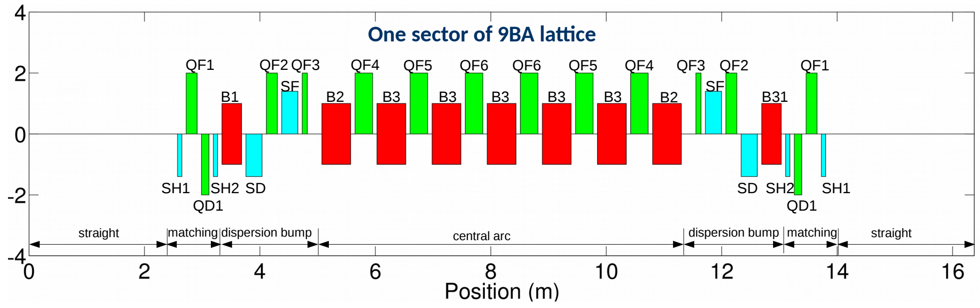
Courtesy: Changchun Sun

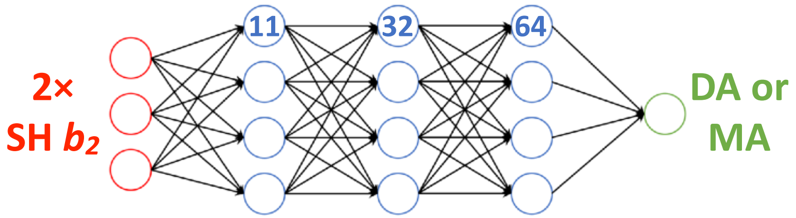
A First Simple NN for Sextupole Optimization

- ALS-U 9BA has **4 sextupole families**: 2 required for chromatic corrections ➔ leaves **2 harmonic families** (SH1 & SH2) for optimization of DA & MA
- Small & simple **3-layer NN** renders accurate prediction of DA/MA as a function of 2 SH variables [4] instead of many-turn tracking with TRACY

Courtesy: Changchun Sun





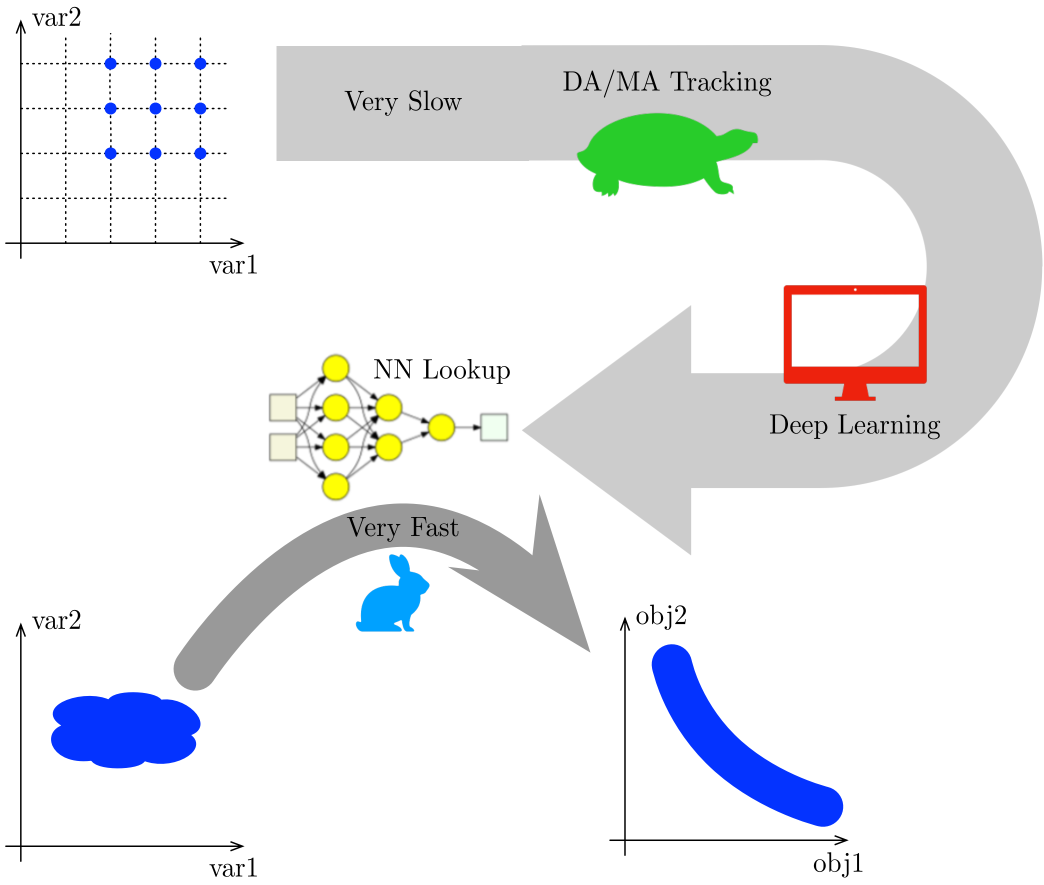


Fully-connected (FC) NN, using ReLU as activation function, # = node depth

[4] Y. Lu, S.C. Leemann, C. Sun, et al., IPAC2021, MOPAB106, p.387.

A First Simple NN for Sextupole Optimization

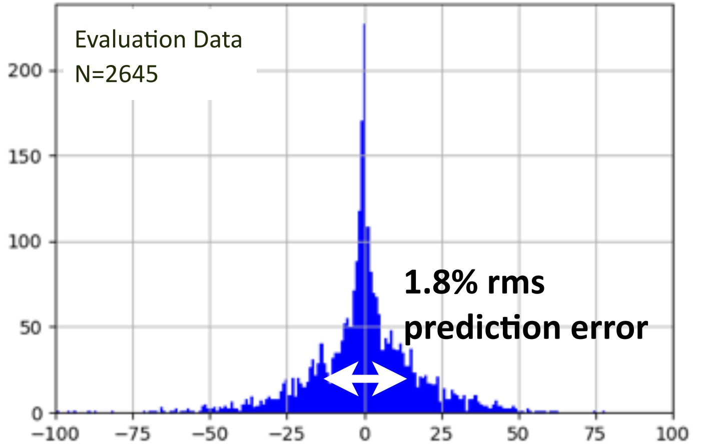
- ALS-U 9BA has **4 sextupole families**: 2 required for chromatic corrections → leaves **2 harmonic families** (SH1 & SH2) for optimization of DA & MA
- Small & simple **3-layer NN** renders accurate prediction of DA/MA as a function of 2 SH variables [4] instead of many-turn tracking with TRACY
  - Training involves low density sampling of 2D input space
  - 20^2 samples tracked for training data → predictions accurate to within ≈2% rms



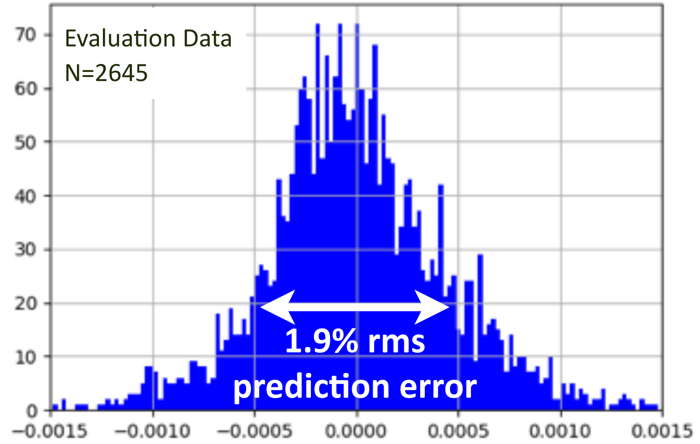
[4] Y. Lu, S.C. Leemann, C. Sun, et al., IPAC2021, MOPAB106, p.387.

A First Simple NN for Sextupole Optimization

- ALS-U 9BA has **4 sextupole families**: 2 required for chromatic corrections ➔ leaves **2 harmonic families** (SH1 & SH2) for optimization of DA & MA
- Small & simple **3-layer NN** renders accurate prediction of DA/MA as a function of 2 SH variables [4] instead of many-turn tracking with TRACY
  - Training involves low density sampling of 2D input space
  - 20^2 samples tracked for training data ➔ predictions accurate to within ≈2% rms



DA NN Prediction Error Histogram

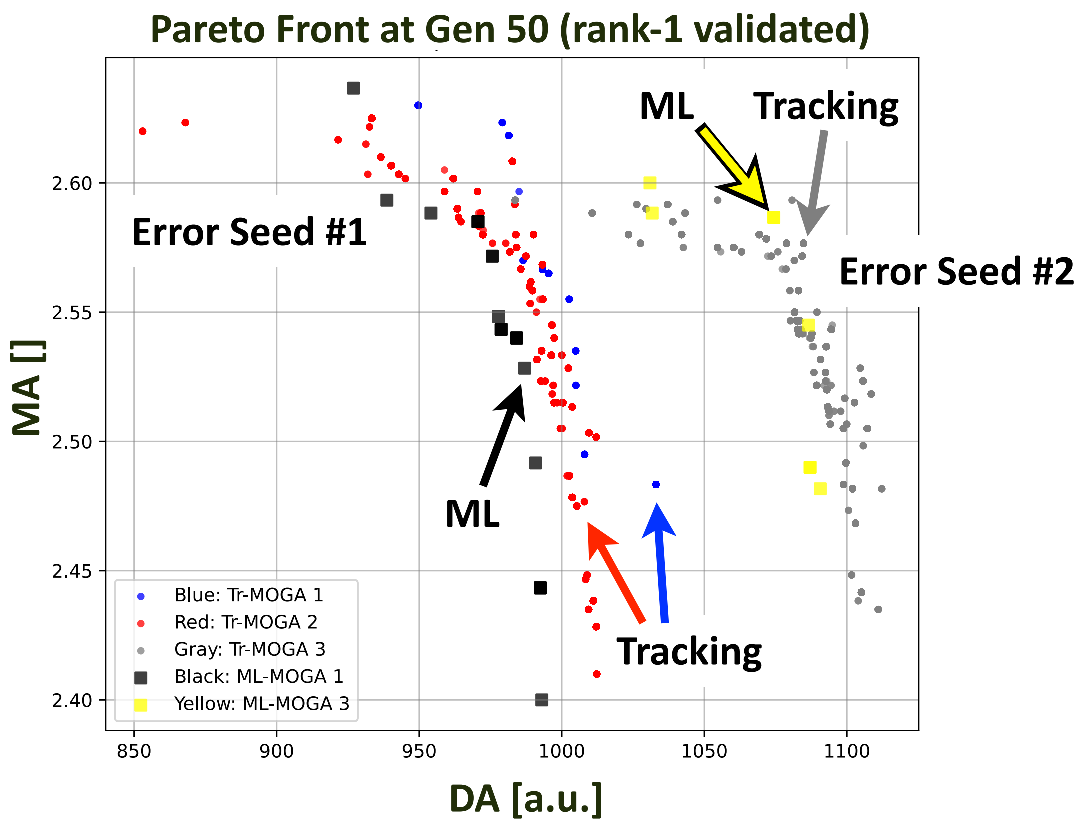


MA NN Prediction Error Histogram

[4] Y. Lu, S.C. Leemann, C. Sun, et al., IPAC2021, MOPAB106, p.387.

A First Simple NN for Sextupole Optimization

- ALS-U 9BA has **4 sextupole families**: 2 required for chromatic corrections ➔ leaves **2 harmonic families** (SH1 & SH2) for optimization of DA & MA
- Small & simple **3-layer NN** renders accurate prediction of DA/MA as a function of 2 SH variables [4] instead of many-turn tracking with TRACY
  - Training involves low density sampling of 2D input space
  - 20^2 samples tracked for training data ➔ predictions accurate to within ≈2% rms
  - Overall **speedup ≈ factor 625** (vs. traditional MOGA requiring 250,000 lattices tracked)

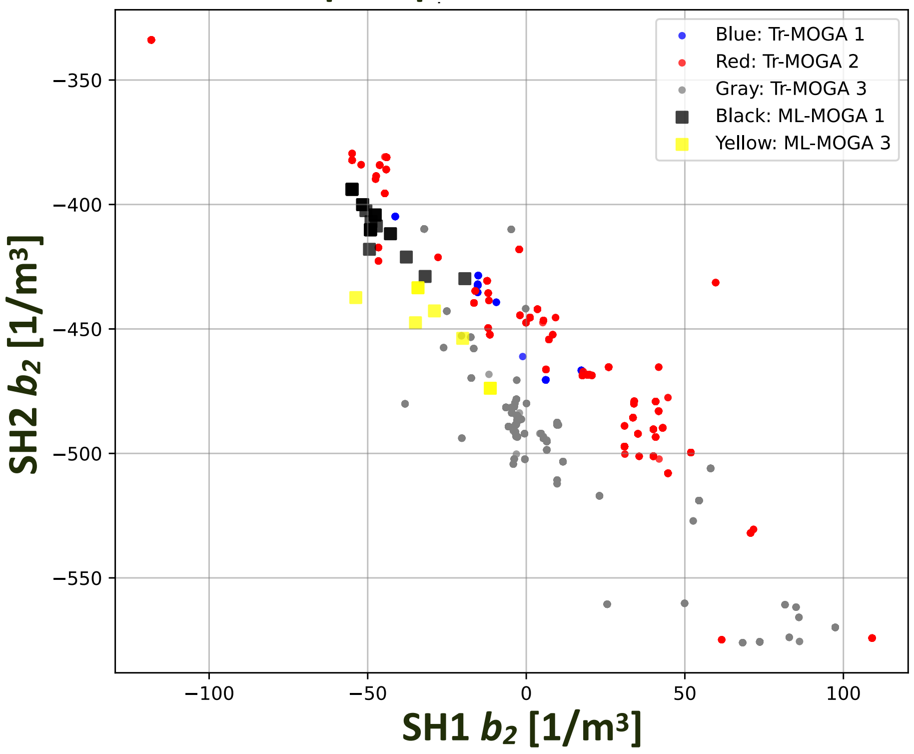


[4] Y. Lu, S.C. Leemann, C. Sun, et al., IPAC2021, MOPAB106, p.387.

A First Simple NN for Sextupole Optimization

- ALS-U 9BA has **4 sextupole families**: 2 required for chromatic corrections ➔ leaves **2 harmonic families** (SH1 & SH2) for optimization of DA & MA
- Small & simple **3-layer NN** renders accurate prediction of DA/MA as a function of 2 SH variables [4] instead of many-turn tracking with TRACY
  - Training involves low density sampling of 2D input space
  - 20^2 samples tracked for training data ➔ predictions accurate to within ≈2% rms
  - Overall **speedup ≈ factor 625** (vs. traditional MOGA requiring 250,000 lattices tracked)
  - NN design & training can be **automated**, 2 lines of code modified in MOGA optimization code

Input Space at Gen 50

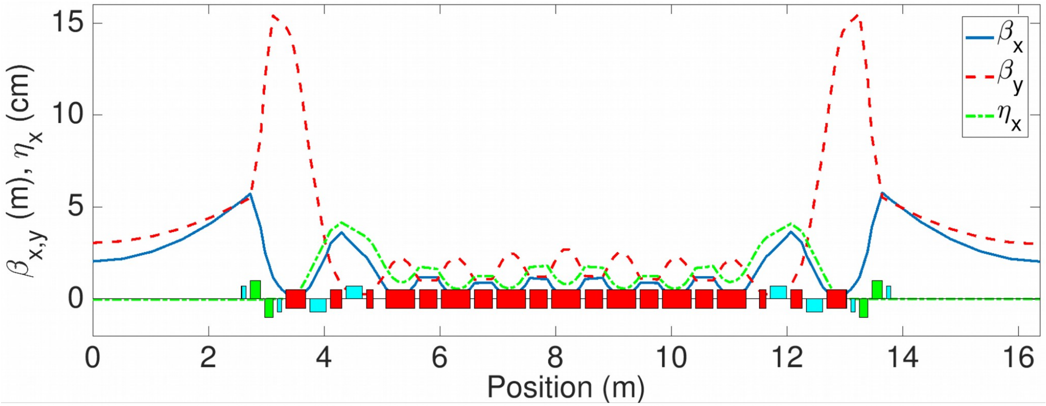


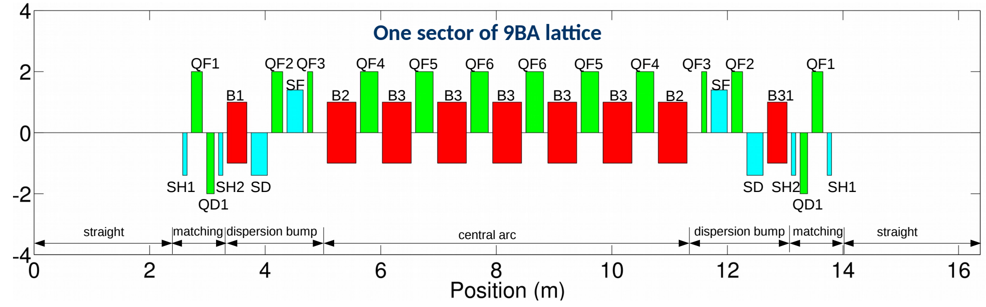
[4] Y. Lu, S.C. Leemann, C. Sun, et al., IPAC2021, MOPAB106, p.387.

ML for Full Linear & Nonlinear ALS-U Optimization

- **ALS-U 9BA** @ 2nd stage MOGA:
  9 quadrupoles, 4 sextupoles → **11 free knobs**

Courtesy: Changchun Sun





ML for Full Linear & Nonlinear ALS-U Optimization

Linear Opt.

2 Objectives:
- Emittance
- Beta

9 Knobs:
- 9 quad gradient

To explore input parameter and objective spaces

Linear & nonlinear opt.

3 Objectives:
- Emittance
- MA
- Total diffusion rate

11 Knobs:
- 9 quad gradient
- 2 harmonic sext.

Many runs were carried out; hyper-parameters and input parameter ranges are tuned

Linear & nonlinear Opt. with reverse bend

3 Objectives:
- Emittance
- MA
- Total diffusion rate

14 Knobs:
- 9 quad gradient
- 2 harmonic sext.
- 3 reverse bend ang.

Reduce emit by about 20% but similar DA

Linear & nonlinear Opt. using alternative objectives

3 Objectives:
- Brightness
- Lifetime
- Dynamic acceptance

14 Knobs:
- 9 quad gradient
- 2 harmonic sext.
- 3 reverse bend ang.

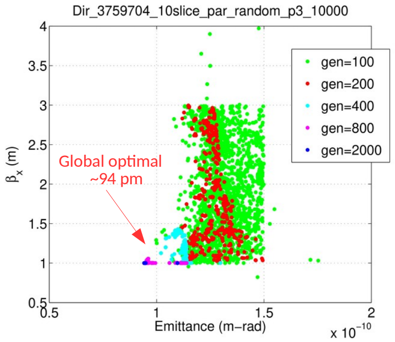
Lifetime is further improved and lattice variants are identified

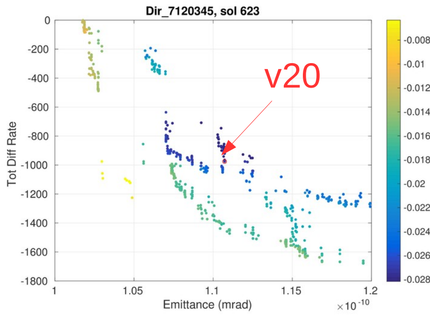
Introduce 3.2T HBend by matching

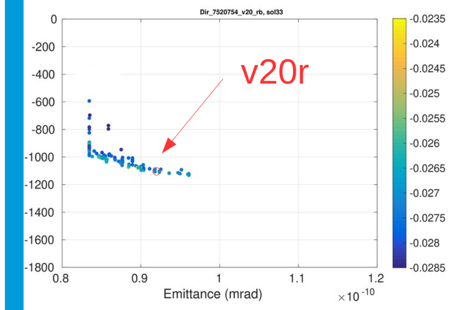
Matching Objectives:
- Twiss functions
- Phase advance between SFs

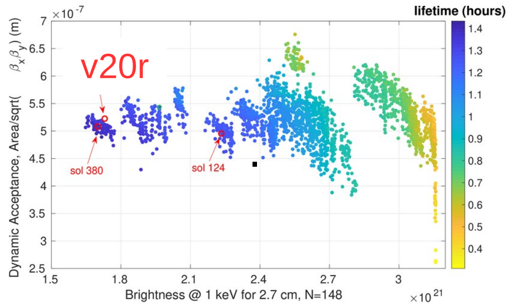
6 Knobs:
- 5 quad gradient
- 1 dipole gradient

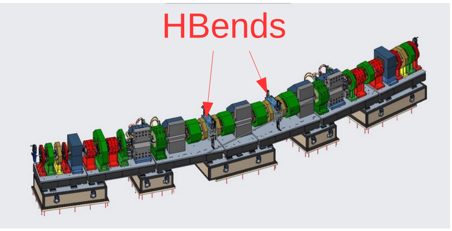
Increase natural emit by 18% and lifetime by 10% but similar DA











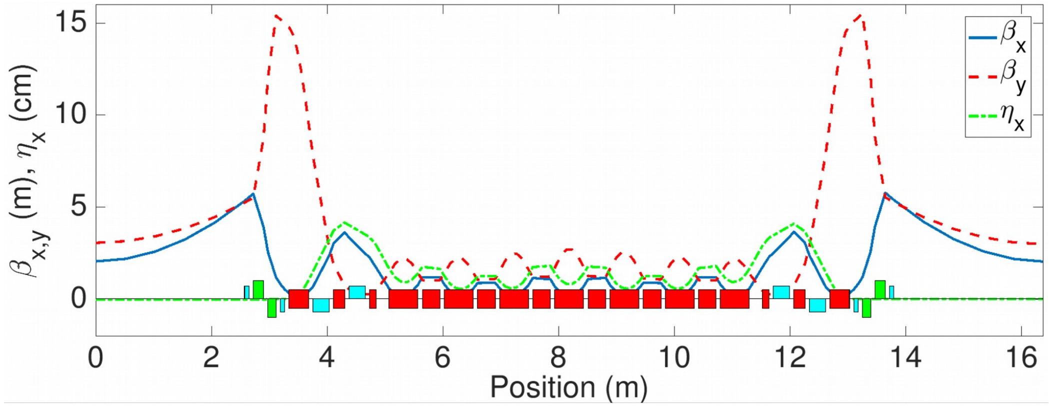
Courtesy: Changchun Sun

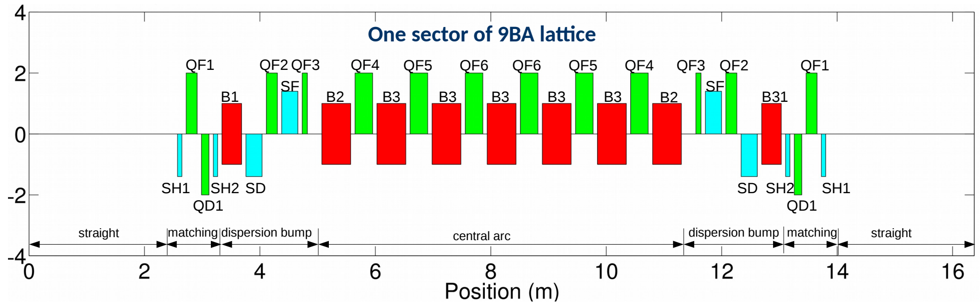
This stage will be focus here

ML for Full Linear & Nonlinear ALS-U Optimization

- **ALS-U 9BA** @ 2nd stage MOGA:
  9 quadrupoles, 4 sextupoles ➔ **11 free knobs**
- Roughly a dozen magnet/lattice constraints on top of quadrupole ranges (from 1st stage)
- ***Objectives:*** ε_0, MA, and on-momentum DA (modeled as integrated diffusion rate)
- **Training data** for 11D problem can no longer be acquired through equidistant sampling of input space
- Do not want to make too many assumptions or “wise choices” ➔ retain generality of approach…

Courtesy: Changchun Sun

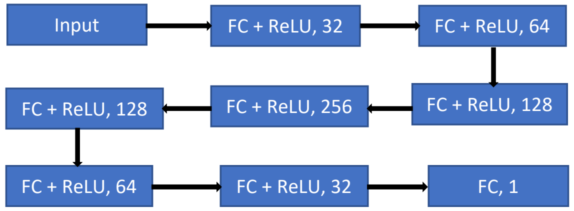




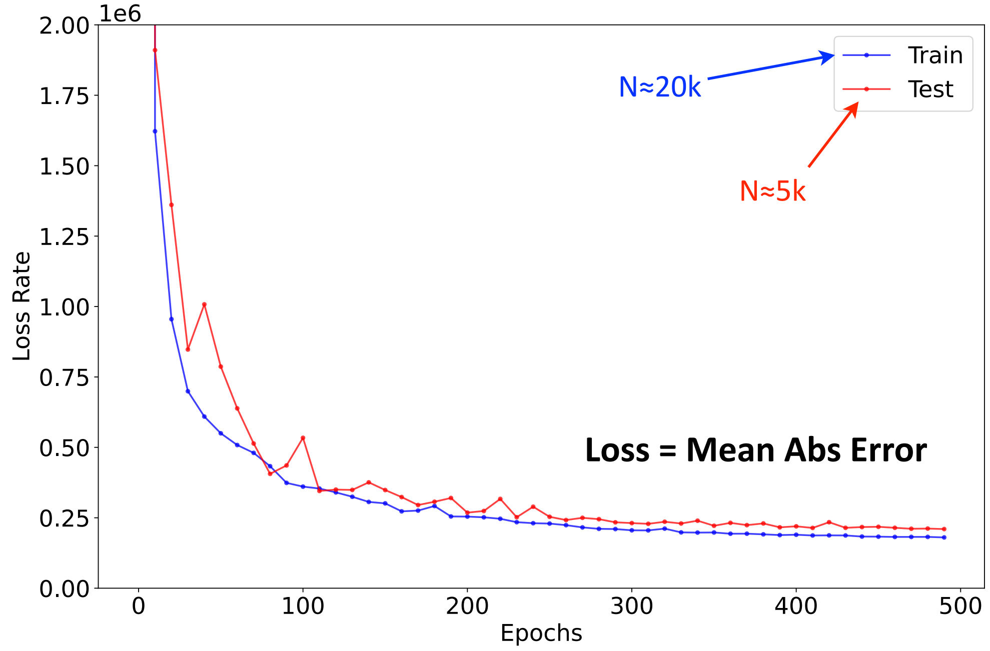
Natural emittance	$\varepsilon_0 < 155\,\mathrm{pm\,rad}$
Maximum beta	$\beta_{x,y} < 30\,\mathrm{m}$
Maximum dispersion	$\eta_x < 15\,\mathrm{cm}$
Fractional tunes	$0.1 < \nu_{x,y} < 0.4$
Dispersion at center of straight	$\lvert\eta_x^*\rvert < 1\,\mathrm{mm}$
Beta at center of straight	$1\,\mathrm{m} < \beta_{x,y}^* < 5\,\mathrm{m}$
Beta in central arc bends (B3)	$\beta_{x,y}^{\mathrm{B3}} < 4\,\mathrm{m}$
Fractional tune difference	$\lvert\nu_x - \nu_y\rvert < 0.01$
Chromatic sextupole strength (SF, SD)	$b_2 < 900\,\mathrm{m}^{-3}$

ML for Full Linear & Nonlinear ALS-U Optimization (cont.)

- Instead: use first generations of **MOGA data** as training data for deep neural networks (DNNs)
- Use two **8-layer DNNs** in lieu of MOGA calls to TRACY for DA and MA (via many-turn tracking)
- Traditional MOGA requires about 640 gen (5000 children/gen) ➔ ≈8 days on 1000-core cluster
- Training 2 DNNs to get DA/MA predictions ≈1% rms requires about 10 gen (of which only ≈5 used due to rejection of candidates with violated constraints)

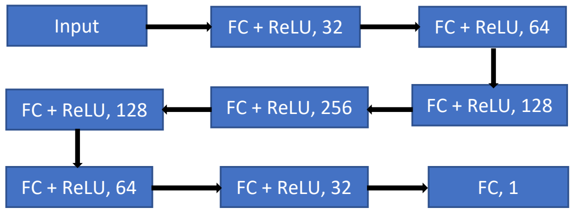


Fully-connected (FC) NN, using ReLU as activation function, # = node depth

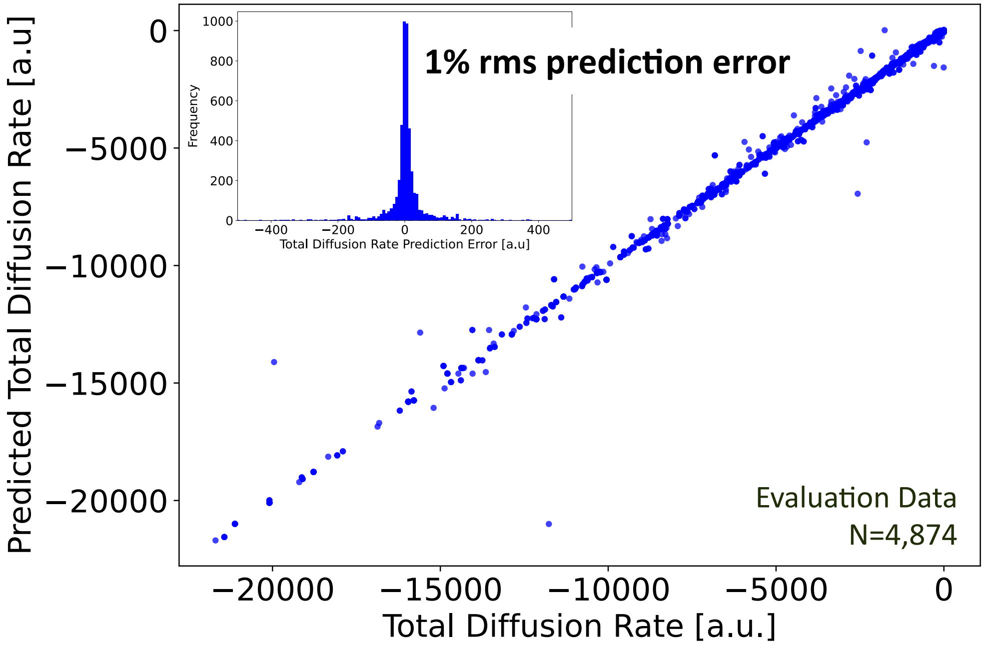


ML for Full Linear & Nonlinear ALS-U Optimization (cont.)

- Instead: use first generations of **MOGA data** as training data for deep neural networks (DNNs)
- Use two **8-layer DNNs** in lieu of MOGA calls to TRACY for DA and MA (via many-turn tracking)
- Traditional MOGA requires about 640 gen (5000 children/gen) ➔ ≈8 days on 1000-core cluster
- Training 2 DNNs to get DA/MA predictions ≈1% rms requires about 10 gen (of which only ≈5 used due to rejection of candidates with violated constraints)
- But once DNNs trained ➔ quasi-instantaneous lookup (16 ms) vs. DA/MA tracking (88 sec)

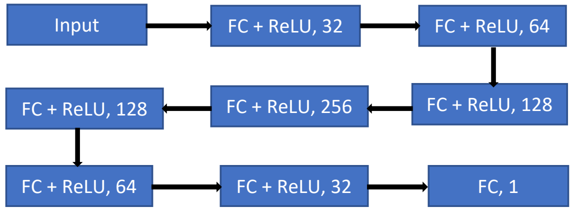


Fully-connected (FC) NN, using ReLU as activation function, # = node depth

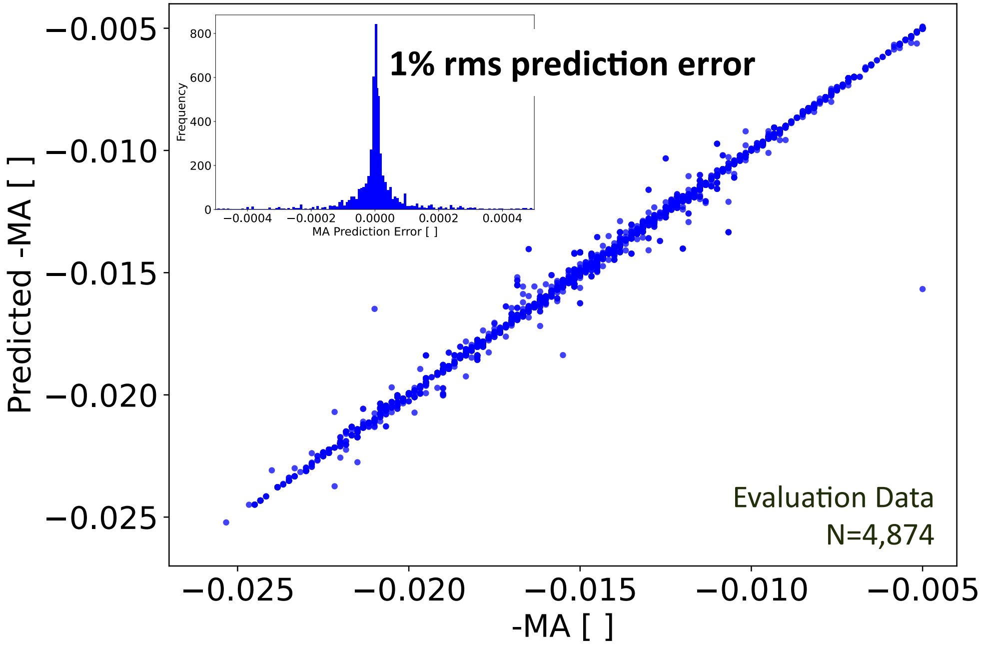


ML for Full Linear & Nonlinear ALS-U Optimization (cont.)

- Instead: use first generations of **MOGA data** as training data for deep neural networks (DNNs)
- Use two **8-layer DNNs** in lieu of MOGA calls to TRACY for DA and MA (via many-turn tracking)
- Traditional MOGA requires about 640 gen (5000 children/gen) ➔ ≈8 days on 1000-core cluster
- Training 2 DNNs to get DA/MA predictions ≈1% rms requires about 10 gen (of which only ≈5 used due to rejection of candidates with violated constraints)
- But once DNNs trained ➔ quasi-instantaneous lookup (16 ms) vs. DA/MA tracking (88 sec)

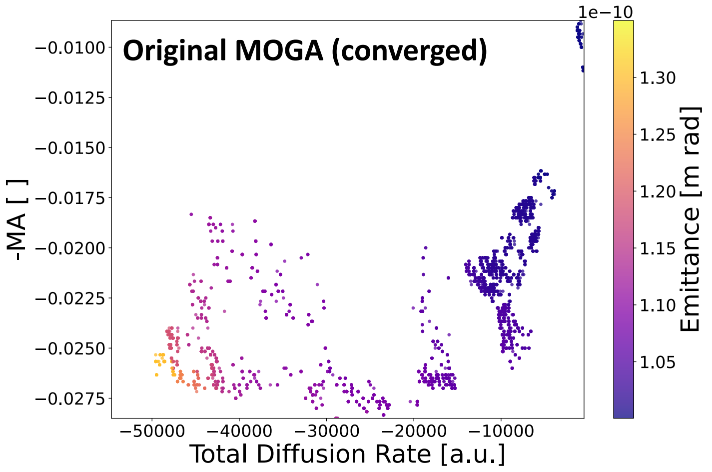


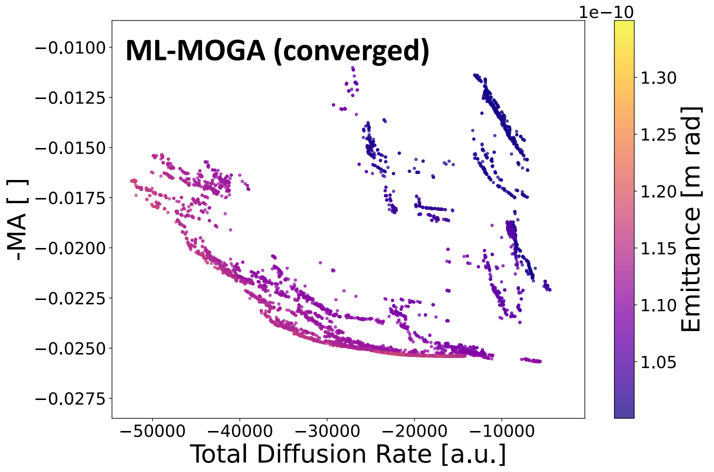
Fully-connected (FC) NN, using ReLU as activation function, # = node depth

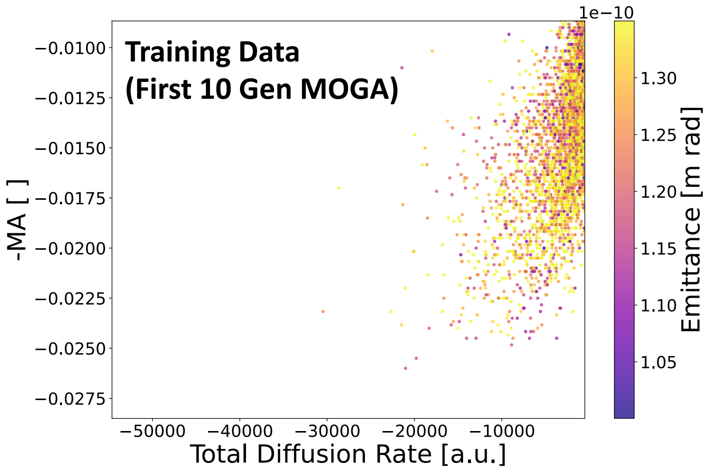


But of course it's a bit more complicated…

- ML predictions are not 100% accurate (training data based on initial optimization data ➔ potentially far from Pareto-optimal areas in input space)

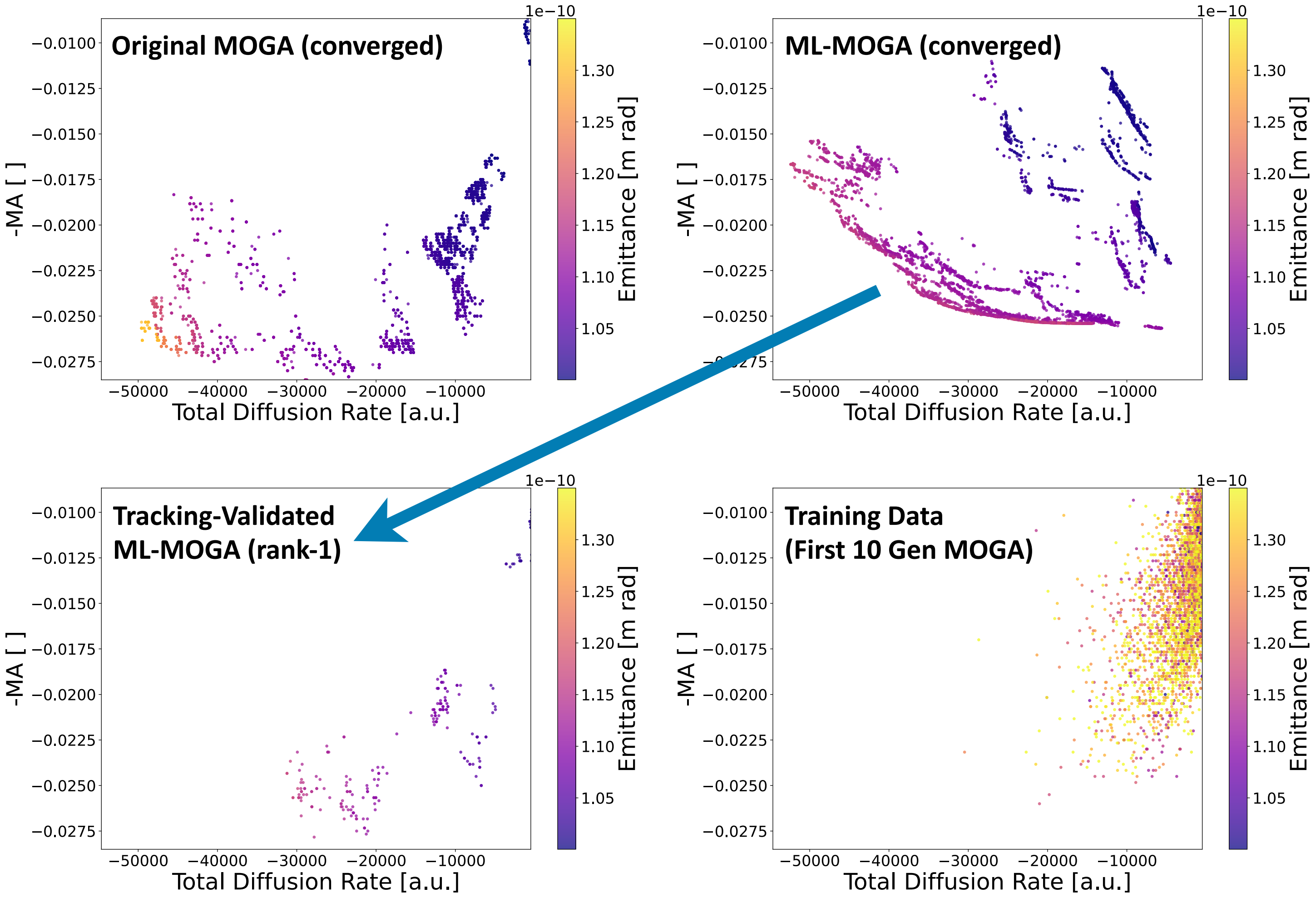






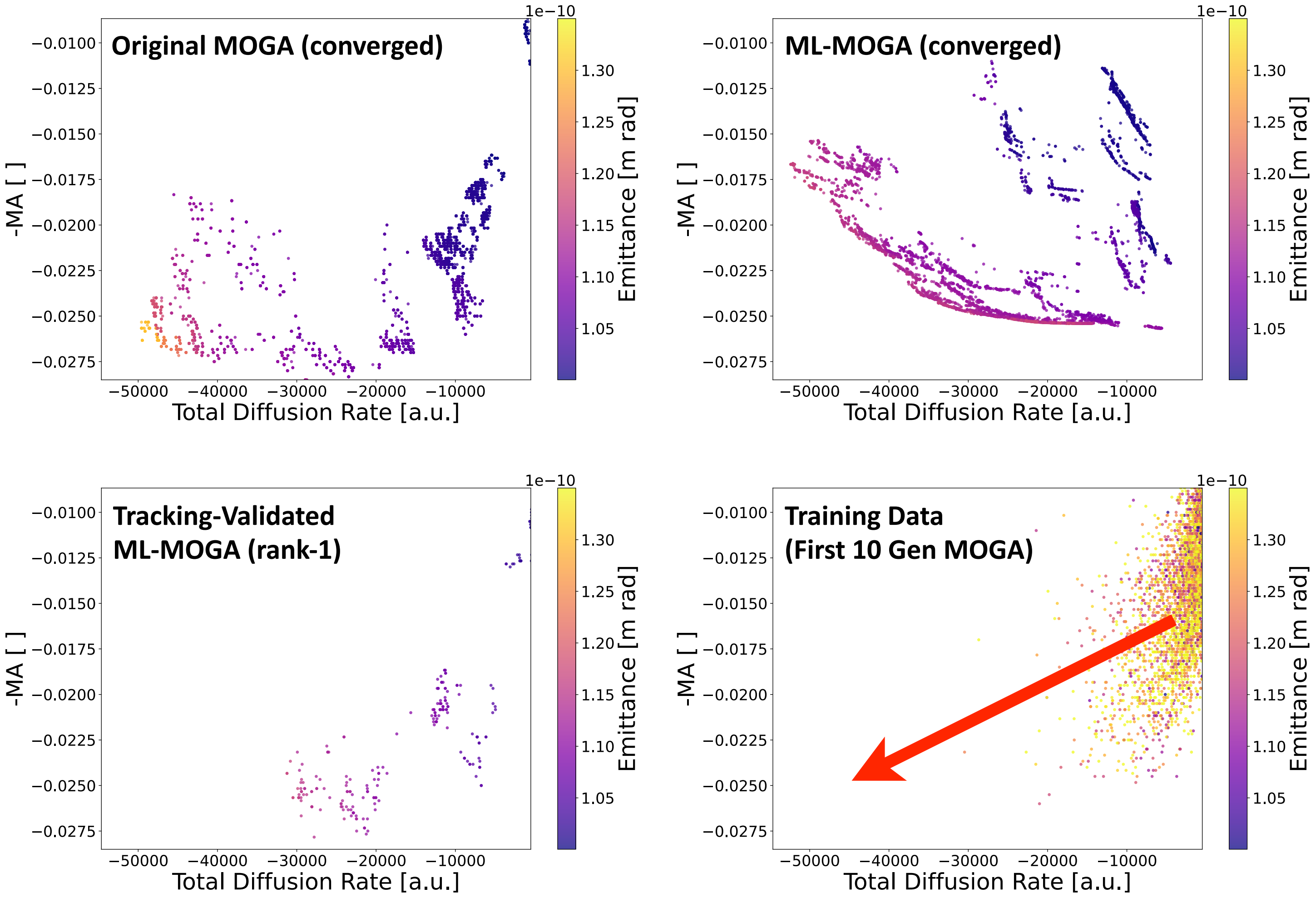
But of course it's a bit more complicated…

- ML predictions are not 100% accurate (training data based on initial optimization data ➔ potentially far from Pareto-optimal areas in input space)
- ML-MOGA solutions show disagreement to **tracking validation** ➔ converged solution front is not entirely non-dominated



But of course it’s a bit more complicated…

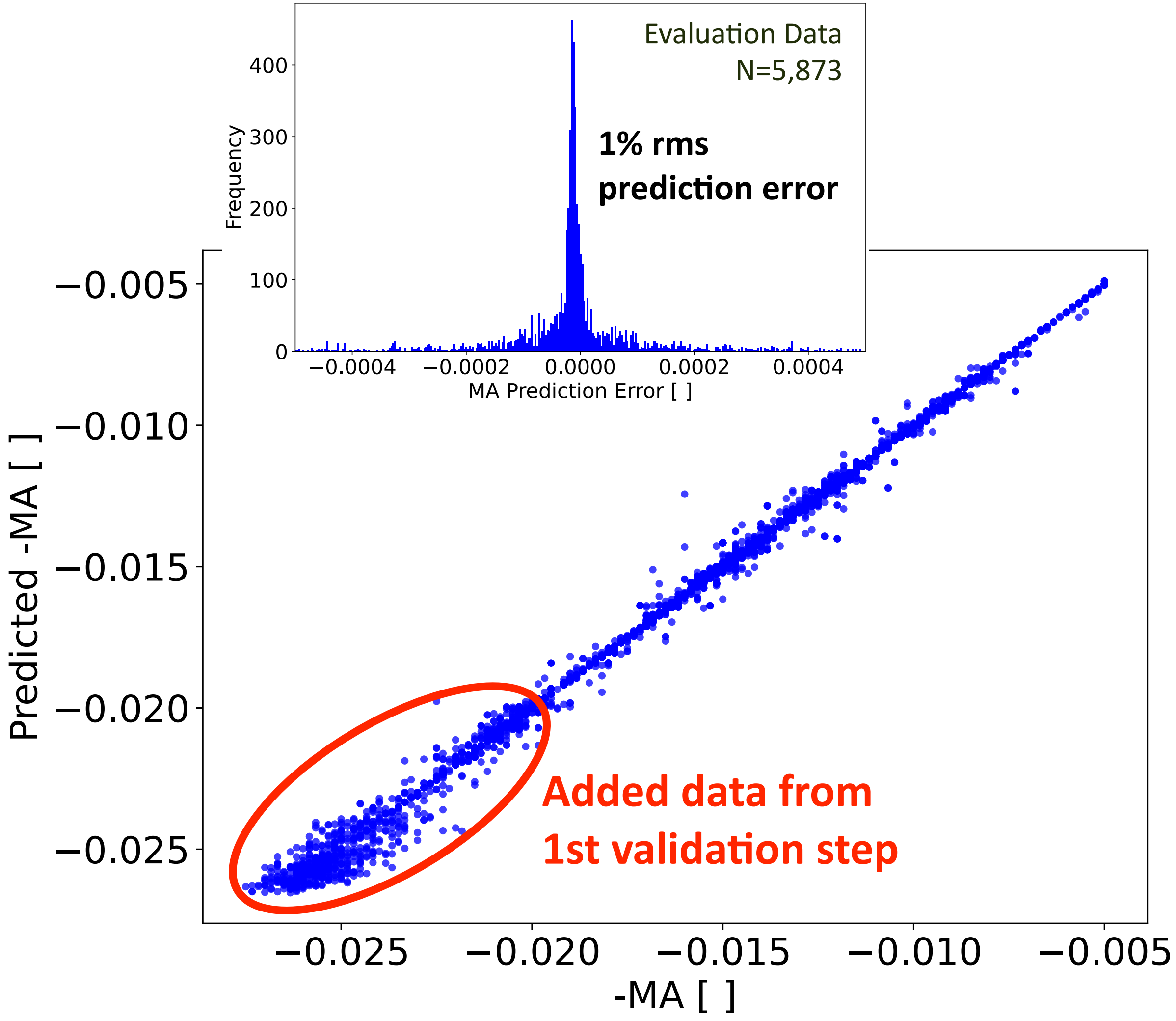
- ML predictions are not 100% accurate (training data based on initial optimization data ➔ potentially far from Pareto-optimal areas in input space)
- ML-MOGA solutions show disagreement to **tracking validation** ➔ converged solution front is not entirely non-dominated
- Want to **retrain DNNs** with an improved resampling of input space ➔ more samples closer to optimal solutions as in [5], …
- …but here for a many-dimensional input space without making any assumptions on smoothness of distributions



[5] A. Edelen, N. Neveu, M. Frey, et al., PRAB **23** 044601, 2020.

Repeated Retraining Improves ML-MOGA

- **Retraining DNNs** with tracking validation data is computationally inexpensive & makes no assumptions on distributions
- Retrained DNN is used for next run starting with inputs from final gen of last run
- **Iterate** this ML–validation–retraining process until ML-MOGA results reach the true Pareto-optimal front
  - But when is that? How do we know our predictions have become accurate enough and our ML-MOGA derived Pareto front is the actual Pareto front?
  - Also, traditional MOGA requires ≈640 gen, ML-MOGA trained on 10 gen → minimizing no. of additional required iterations is crucial to maintaining **large overall speedup**



Distance Metrics & Convergence

- Introduce two distance metrics for **input/objective space**
- Euclidean norms normalized in each variable → single unit-free relative **measure for movement of distribution** in input/objective space
- Metrics can inform us when
  - MOGA can be considered truly converged (required for full automation)
  - there is no more added benefit from an additional iteration of retraining–ML–validation
- For objective space, choice of **"golden target"** leaves some freedom to lattice designer (not sensitive as long as chosen aggressively)
- MOGA considered converged when for large *m*

$$\Delta\delta_{i,o}(m) \to 0$$

- Consider retraining–ML–validation process converged once Δ_f no longer reduces with additional iterations

Input Space

Gen *m*

Dimensions of input space ***N***

$$\delta_i(m) = \frac{1}{n(n-1)} \sum_{j=1}^{n} \sum_{k=1}^{n} \sqrt{\sum_{l=1}^{N} \left(\frac{a_{jl}^{(m)} - a_{kl}^{(m-1)}}{c_l} \right)^2}$$

Pop size ***n***

Input ***l*** of child ***j*** at gen ***m***

Parameter range for input ***l***

Objective Space

ε_0 of child ***j*** at gen ***m***

DA of child ***j*** at gen ***m***

MA of child ***j*** at gen ***m***

$$\delta_o(m) = \frac{1}{n} \sqrt{\sum_{j=1}^{n} \left[\left(\frac{\varepsilon_{mj} - \varepsilon_0}{\varepsilon_0} \right)^2 + \left(\frac{D_{mj} - D_0}{D_0} \right)^2 + \left(\frac{M_{mj} - M_0}{M_0} \right)^2 \right]}$$

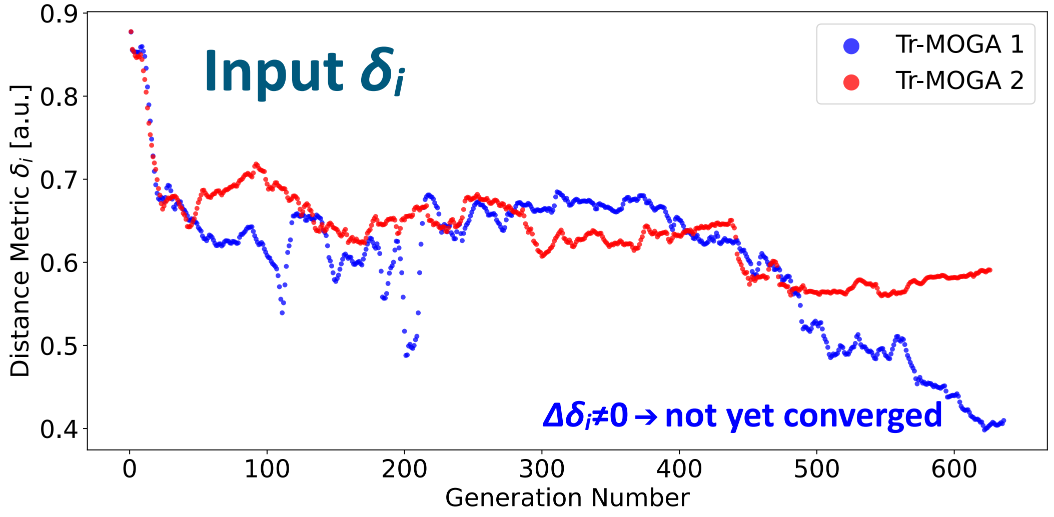
Reference Values $\{\varepsilon_0, D_0, M_0\}$

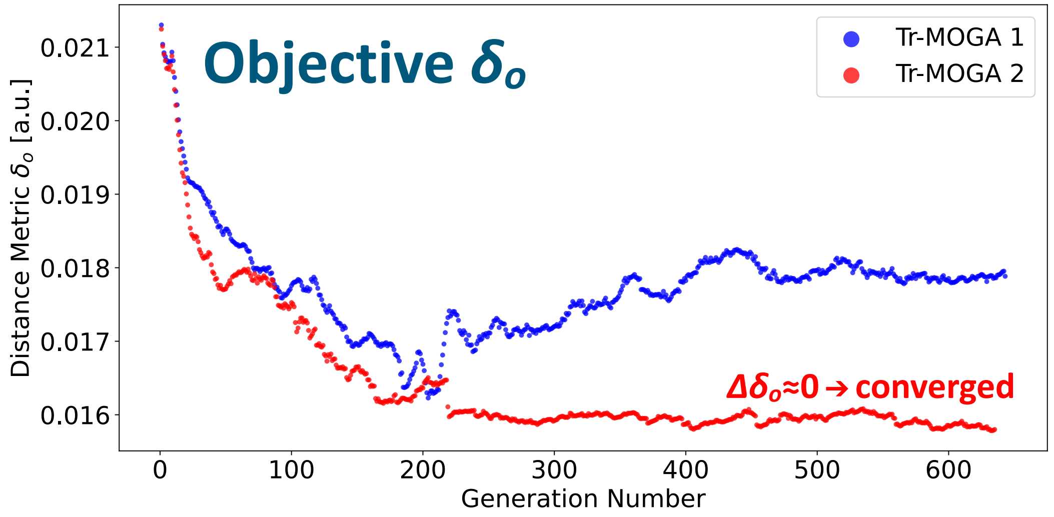
$$\Delta_f = \delta_o^v(m_f) - \delta_o(m_f)$$

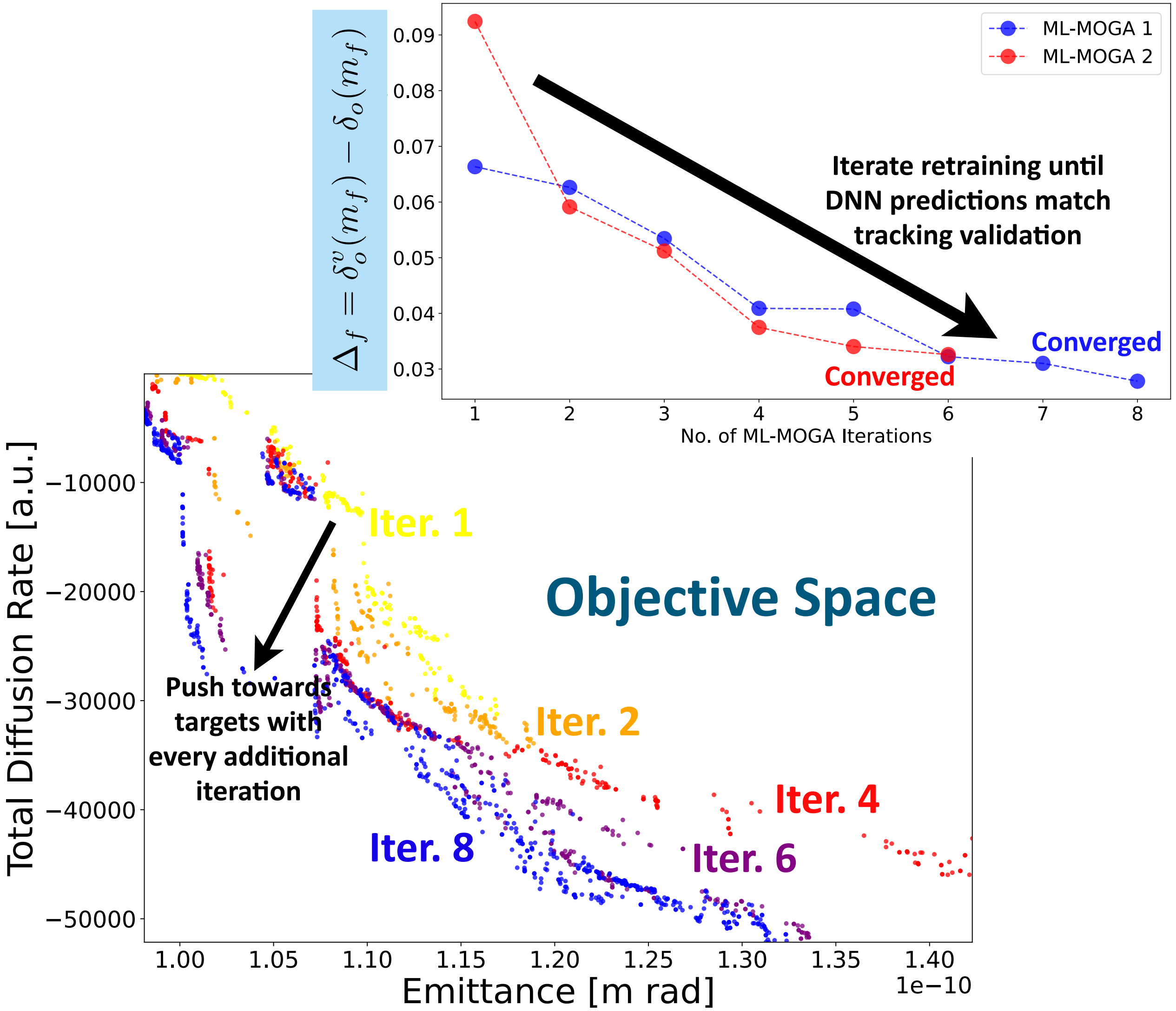
Tracking Validated δ_o

Final gen m_f

Distance Metrics & Convergence (cont.)

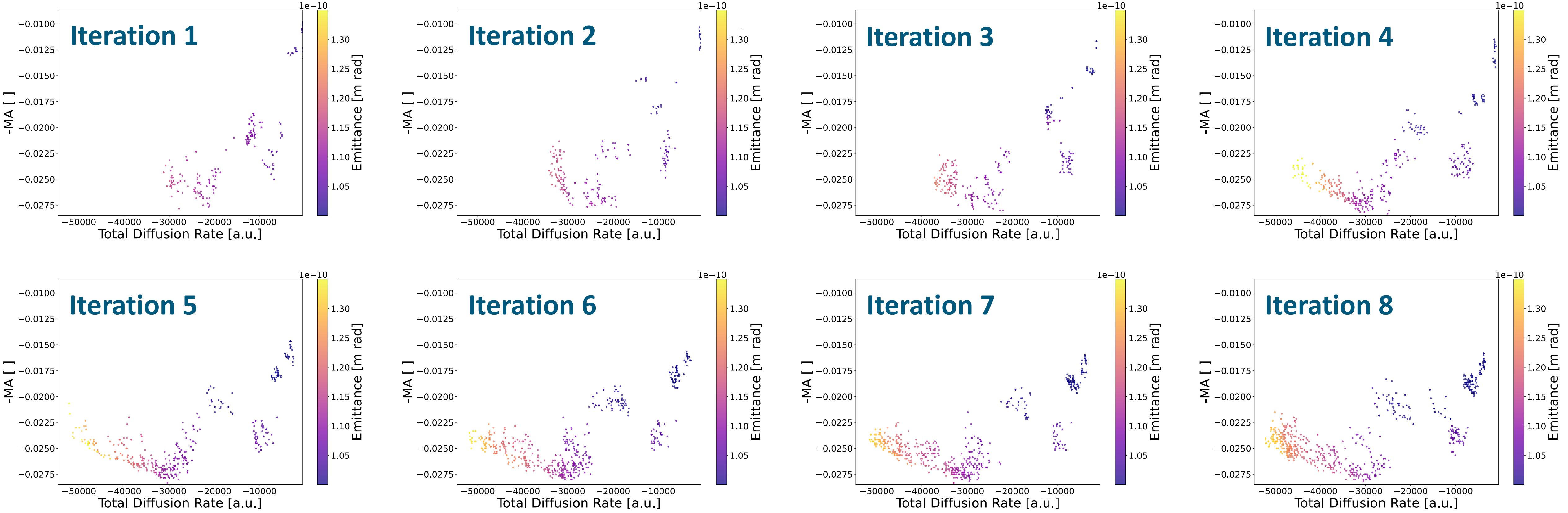






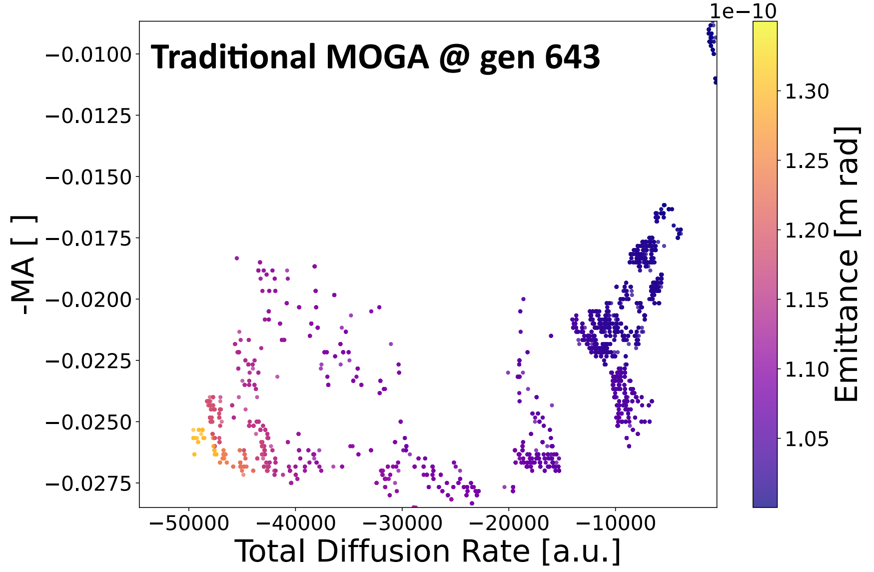
Results

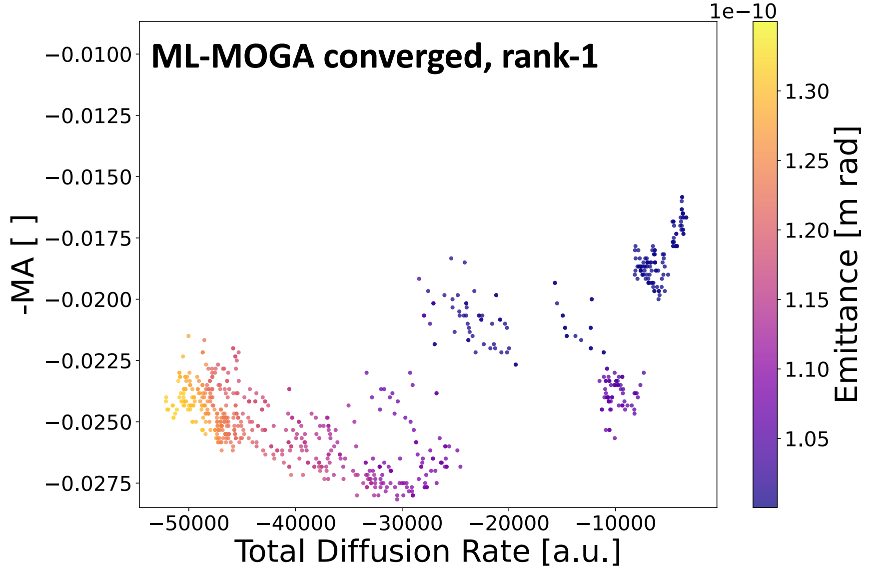
- Retraining shows very quick convergence
  (6-8 iterations)



Results

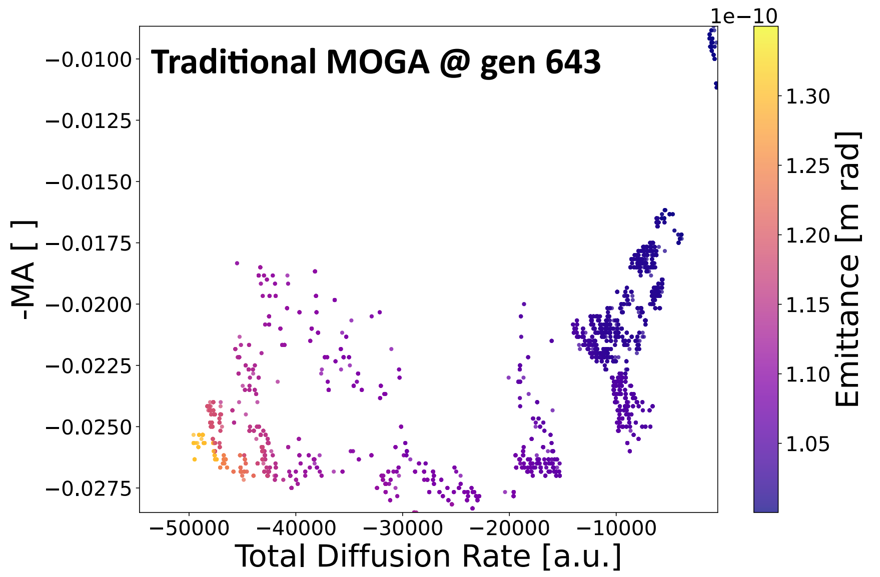
- Retraining shows very quick convergence
  (6-8 iterations)
- Once fully converged, ML-MOGA inputs & objectives match those of traditional MOGA to within "noise floor"

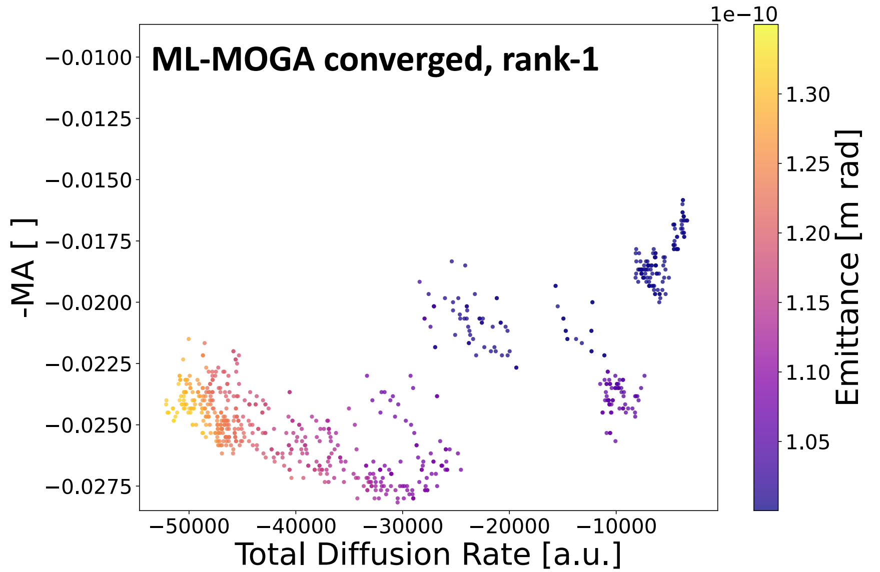




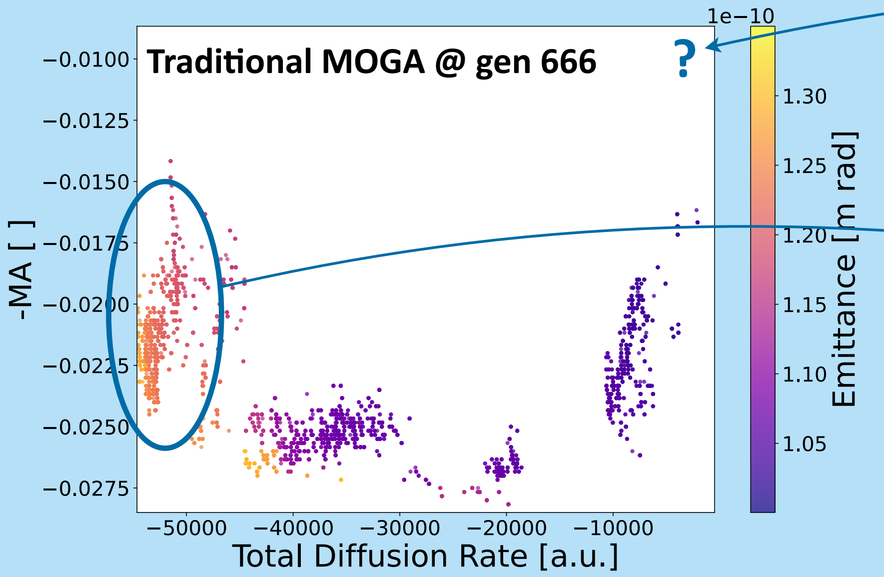
Results

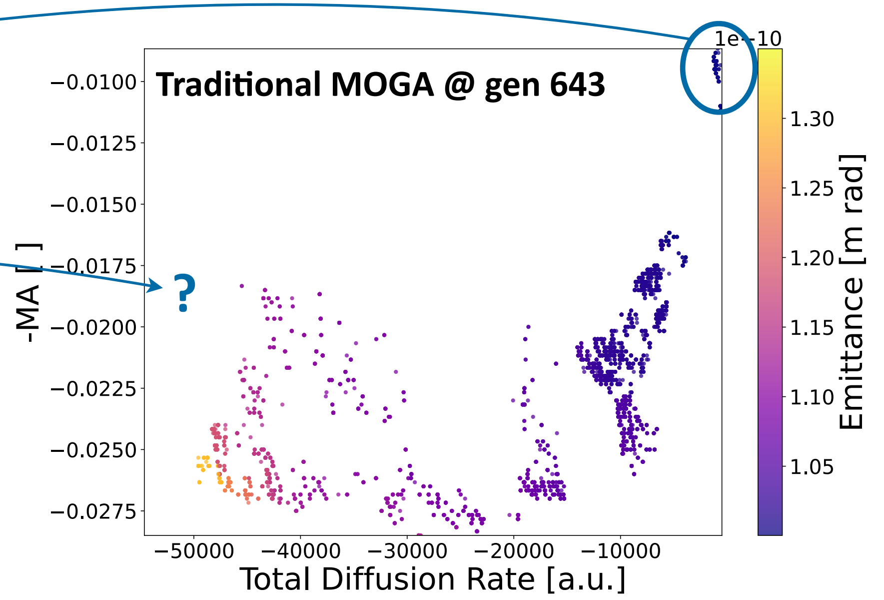
- Retraining shows very quick convergence
  (6-8 iterations)
- Once fully converged, ML-MOGA inputs & objectives match those of traditional MOGA to within “noise floor”
- Stochastic noise in MOGA process accounts for bulk of discrepancy in objective space
  (input space shows excellent agreement)



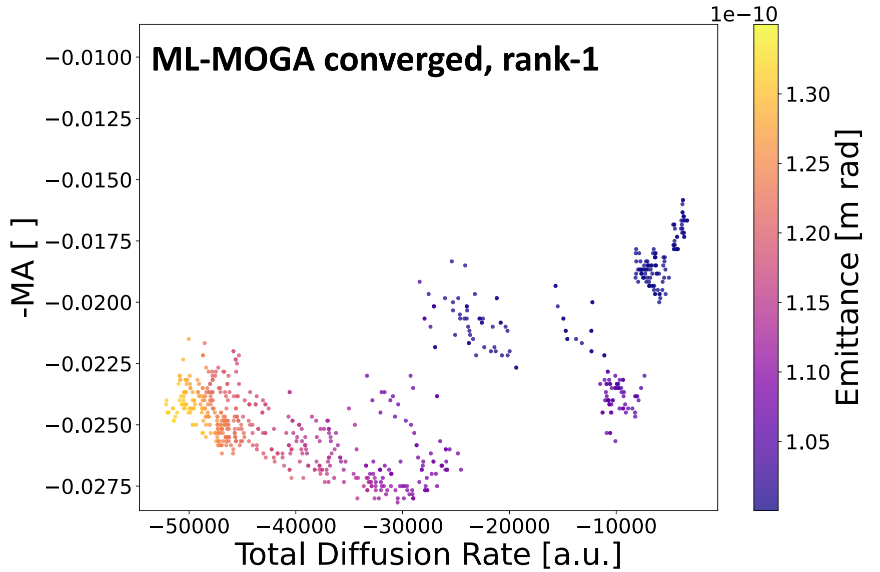


Results



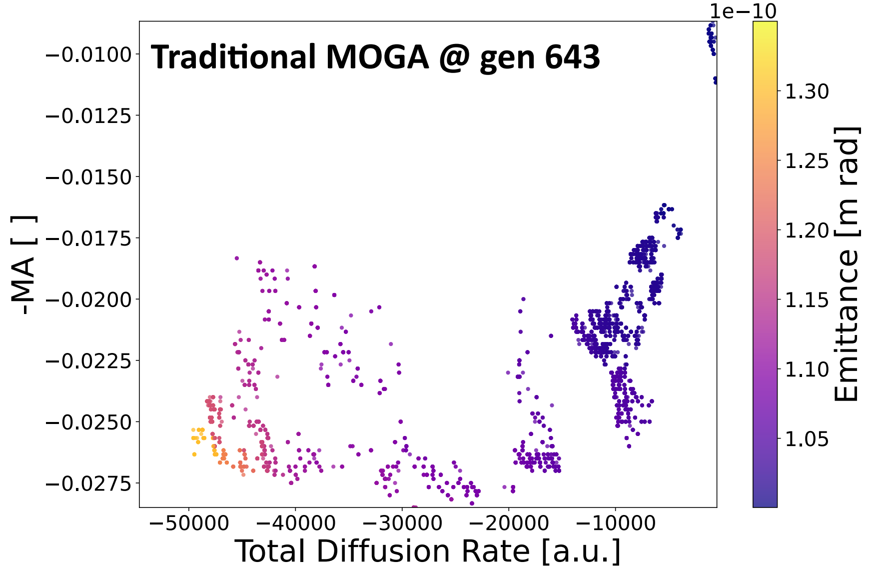


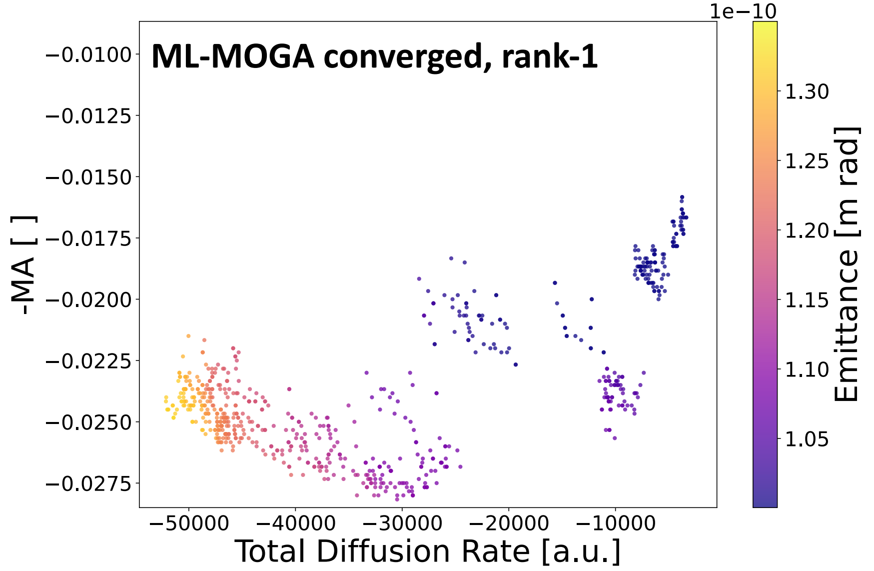
Only MOGA random seed changed → same physics



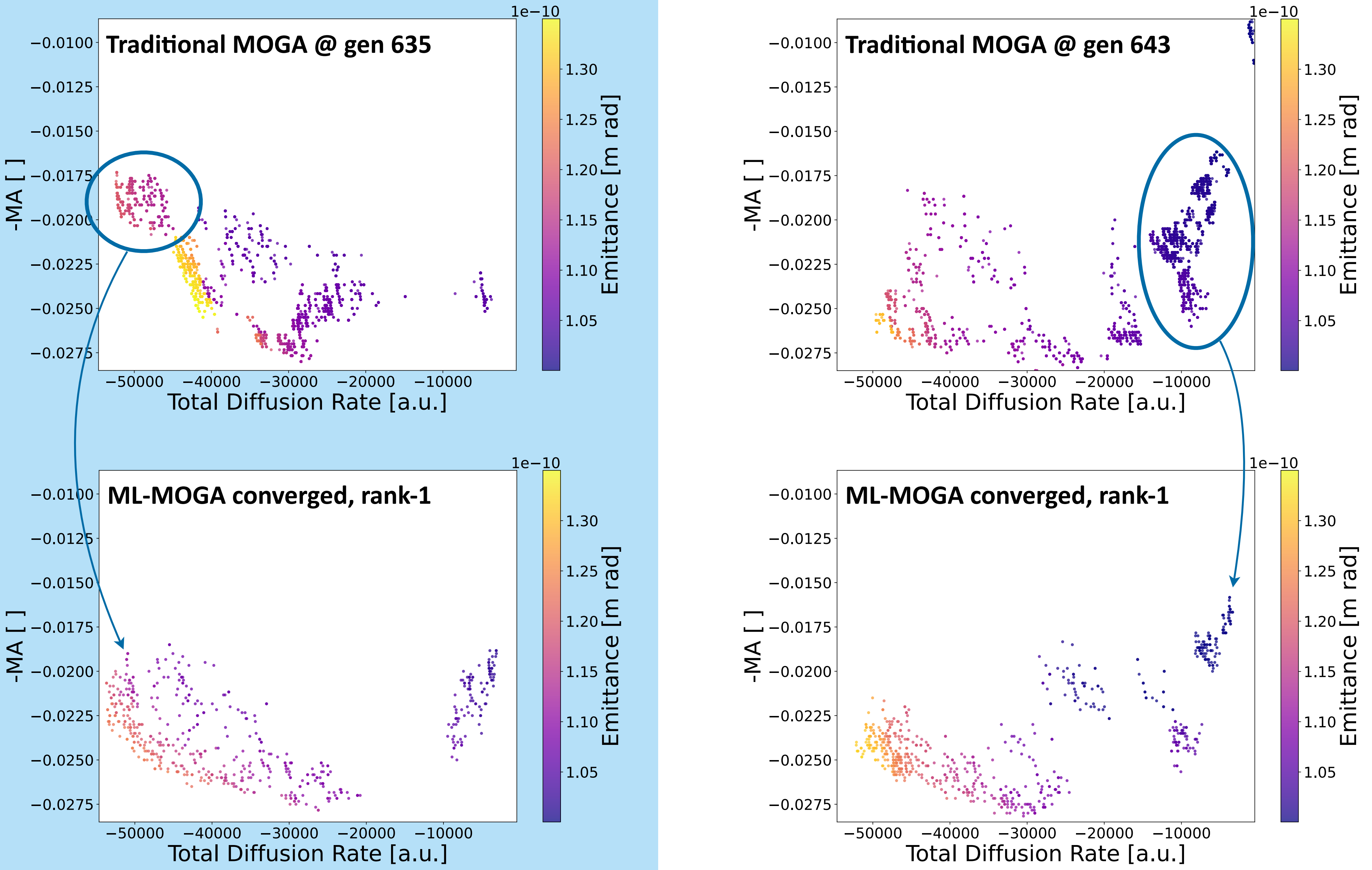
Results

- Retraining shows very quick convergence
  (6-8 iterations)
- Once fully converged, ML-MOGA inputs & objectives match those of traditional MOGA to within “noise floor”
- Stochastic noise in MOGA process accounts for bulk of discrepancy in objective space
  (input space shows excellent agreement)
- ML-MOGA results remain true to underlying physics changes (changes in error distribution, random error seed)





Results



Error distribution changed
→ different physics

Conclusions

- ML-MOGA requires **≈16 gen tracked vs. ≈640** for traditional MOGA → **overall speedup ≈40** (depending on exact choice of cutoff Δ_f)
- Only **very minor modifications required** to existing MOGA workflow/tools
- Convergence defined in **model-independent** way → process can be automated & adapted to other optimization problems (eg. other lattices, or adding additional DoF such as reverse bending or superbends)
  - Only requirement: DNN prediction errors need to remain small (≲2% rms)
  - Note, hyperparameter tuning & DNN architecture modifications can also be automated by a non-ML expert (eg. AutoML) → **focus remains on lattice design and beam dynamics**
- Vast speedup allows for **optimization of multiple error lattices in parallel** → resulting lattice candidate consists of inputs that are common to all error seeds → likeliest to produce Pareto-optimal solutions for *as-built* machine's error distribution
- **Potential to fully automate entire workflow is highly attractive**

Thank You!

Questions?

Acknowledgments: Yuping Lu, Changchun Sun, Mike Ehrlichman, Hiroshi Nishimura, Marco Venturini, Thorsten Hellert, Rob Ryne, Fernando Sannibale, DOE Office of Science (BES/ASCR) Contract No. DEAC02-05CH11231

Come join ALS Accelerator Physics & Berkeley Lab!

Opening for accelerator physicists at the level of Research Scientist (career/career-track) or Staff Scientist

https://bit.ly/ALS-AP-Jobs