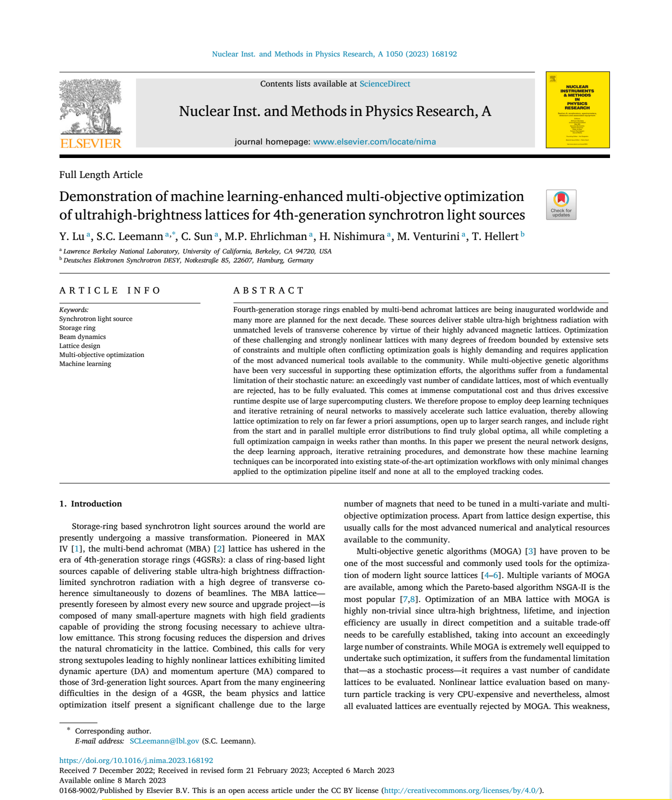


Nuclear Inst. and Methods in Physics Research, A 1050 (2023) 168192

Contents lists available at ScienceDirect

Nuclear Inst. and Methods in Physics Research, A

journal homepage: www.elsevier.com/locate/nima

Full Length Article

Demonstration of machine learning-enhanced multi-objective optimization of ultrahigh-brightness lattices for 4th-generation synchrotron light sources

Y. Lu [a], S.C. Leemann [a,*], C. Sun [a], M.P. Ehrlichman [a], H. Nishimura [a], M. Venturini [a], T. Hellert [b]

[a] Lawrence Berkeley National Laboratory, University of California, Berkeley, CA 94720, USA
[b] Deutsches Elektronen Synchrotron DESY, Notkestraße 85, 22607, Hamburg, Germany

ARTICLE INFO

Keywords:
Synchrotron light source
Storage ring
Beam dynamics
Lattice design
Multi-objective optimization
Machine learning

ABSTRACT

Fourth-generation storage rings enabled by multi-bend achromat lattices are being inaugurated worldwide and many more are planned for the next decade. These sources deliver stable ultra-high brightness radiation with unmatched levels of transverse coherence by virtue of their highly advanced magnetic lattices. Optimization of these challenging and strongly nonlinear lattices with many degrees of freedom bounded by extensive sets of constraints and multiple often conflicting optimization goals is highly demanding and requires application of the most advanced numerical tools available to the community. While multi-objective genetic algorithms have been very successful in supporting these optimization efforts, the algorithms suffer from a fundamental limitation of their stochastic nature: an exceedingly vast number of candidate lattices, most of which eventually are rejected, has to be fully evaluated. This comes at immense computational cost and thus drives excessive runtime despite use of large supercomputing clusters. We therefore propose to employ deep learning techniques and iterative retraining of neural networks to massively accelerate such lattice evaluation, thereby allowing lattice optimization to rely on far fewer a priori assumptions, open up to larger search ranges, and include right from the start and in parallel multiple error distributions to find truly global optima, all while completing a full optimization campaign in weeks rather than months. In this paper we present the neural network designs, the deep learning approach, iterative retraining procedures, and demonstrate how these machine learning techniques can be incorporated into existing state-of-the-art optimization workflows with only minimal changes applied to the optimization pipeline itself and none at all to the employed tracking codes.

1. Introduction

Storage-ring based synchrotron light sources around the world are presently undergoing a massive transformation. Pioneered in MAX IV [1], the multi-bend achromat (MBA) [2] lattice has ushered in the era of 4th-generation storage rings (4GSRs): a class of ring-based light sources capable of delivering stable ultra-high brightness diffraction-limited synchrotron radiation with a high degree of transverse coherence simultaneously to dozens of beamlines. The MBA lattice—presently foreseen by almost every new source and upgrade project—is composed of many small-aperture magnets with high field gradients capable of providing the strong focusing necessary to achieve ultra-low emittance. This strong focusing reduces the dispersion and drives the natural chromaticity in the lattice. Combined, this calls for very strong sextupoles leading to highly nonlinear lattices exhibiting limited dynamic aperture (DA) and momentum aperture (MA) compared to those of 3rd-generation light sources. Apart from the many engineering difficulties in the design of a 4GSR, the beam physics and lattice optimization itself present a significant challenge due to the large number of magnets that need to be tuned in a multi-variate and multi-objective optimization process. Apart from lattice design expertise, this usually calls for the most advanced numerical and analytical resources available to the community.

Multi-objective genetic algorithms (MOGA) [3] have proven to be one of the most successful and commonly used tools for the optimization of modern light source lattices [4–6]. Multiple variants of MOGA are available, among which the Pareto-based algorithm NSGA-II is the most popular [7,8]. Optimization of an MBA lattice with MOGA is highly non-trivial since ultra-high brightness, lifetime, and injection efficiency are usually in direct competition and a suitable trade-off needs to be carefully established, taking into account an exceedingly large number of constraints. While MOGA is extremely well equipped to undertake such optimization, it suffers from the fundamental limitation that—as a stochastic process—it requires a vast number of candidate lattices to be evaluated. Nonlinear lattice evaluation based on many-turn particle tracking is very CPU-expensive and nevertheless, almost all evaluated lattices are eventually rejected by MOGA. This weakness,

* Corresponding author.
E-mail address: SCLeemann@lbl.gov (S.C. Leemann).

https://doi.org/10.1016/j.nima.2023.168192
Received 7 December 2022; Received in revised form 21 February 2023; Accepted 6 March 2023
Available online 8 March 2023

NIM-A **1050**, 168192 (2023)

Machine Learning-Enhanced MOGA for Ultrahigh-Brightness Lattices in 4th-generation Synchrotron Light Sources

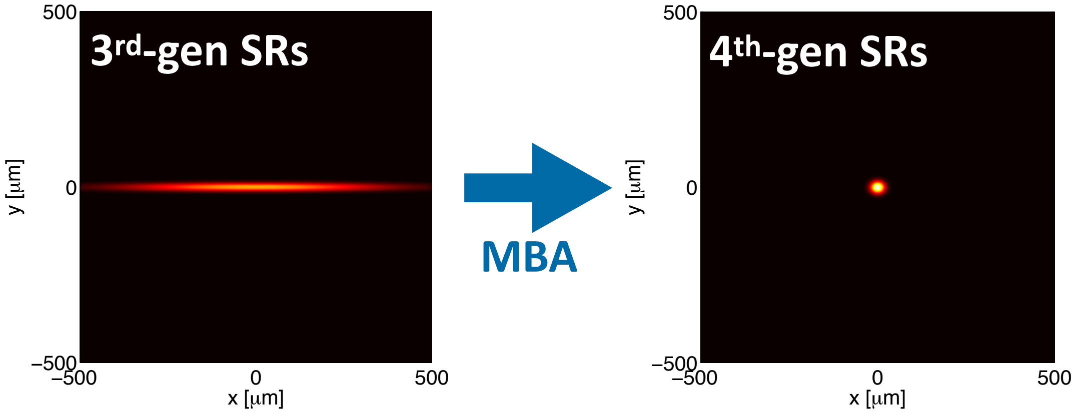
Simon C. Leemann for the ALS ML Team

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

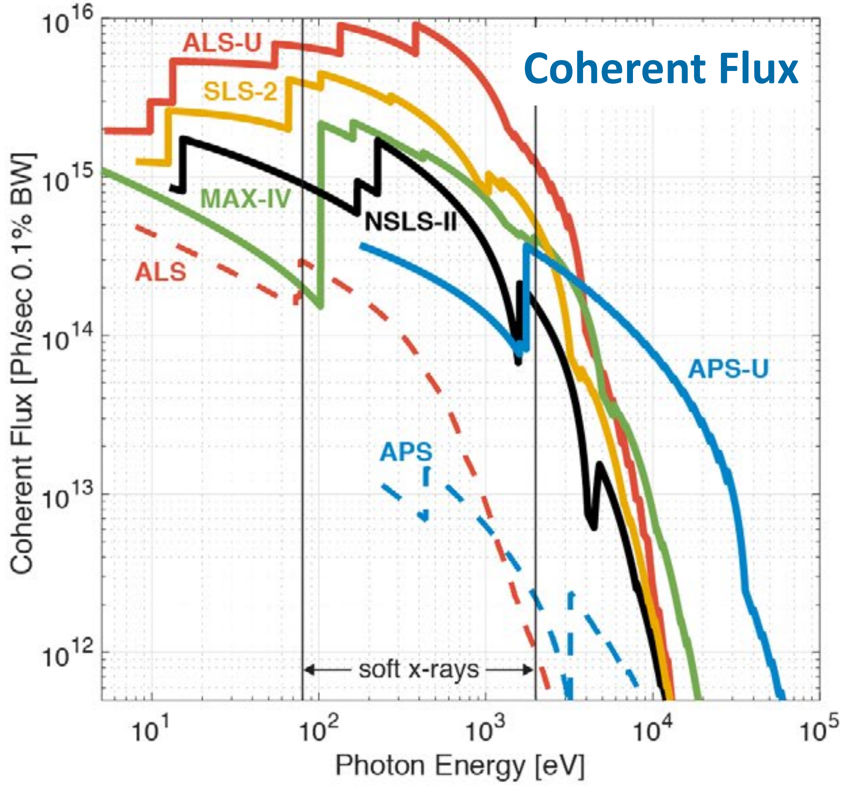
AMP Seminar, April 3, 2023

Introduction

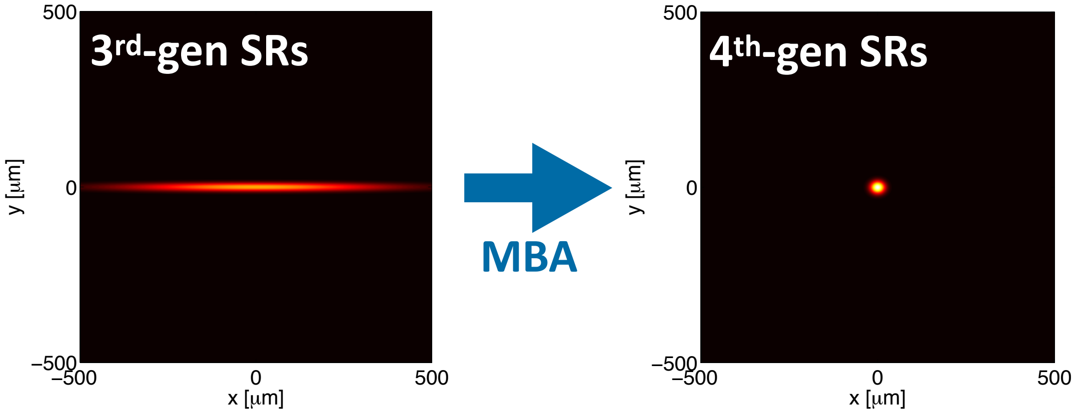
- **4th-generation storage rings** (4GSRs) leverage MBA lattices to render ultra-high brightness with large coherent fraction



Courtesy: Dave Robin

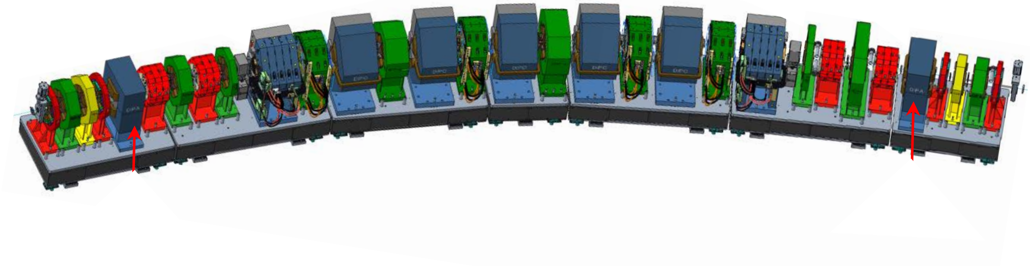


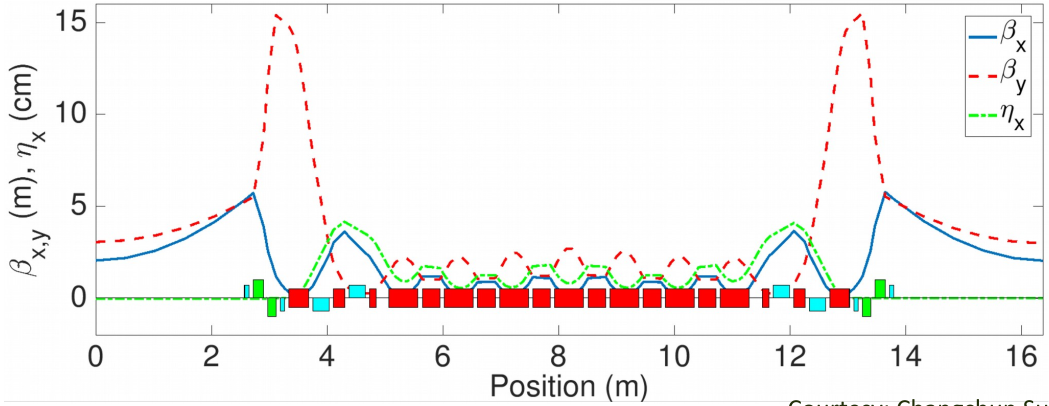
Introduction: The Problem



- **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

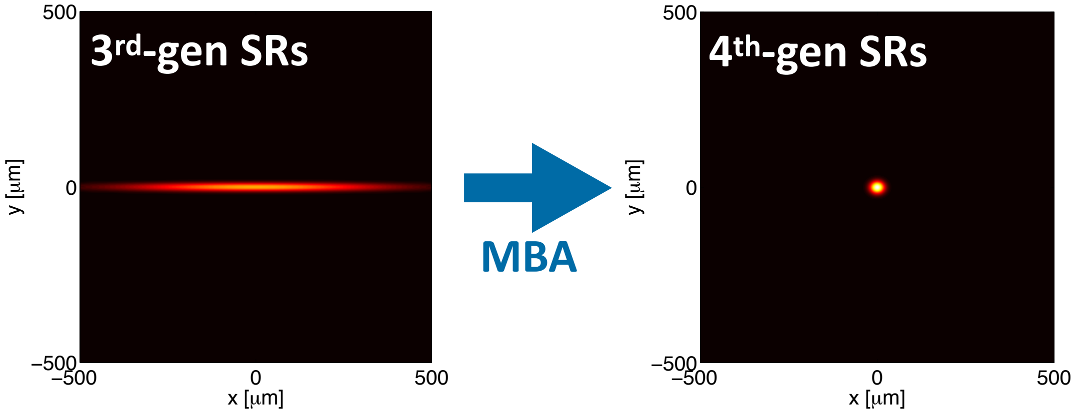


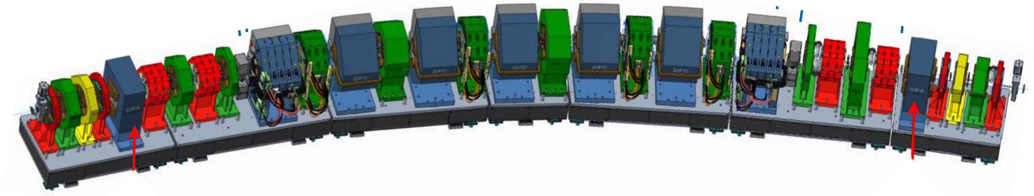


Courtesy: Changchun Sun

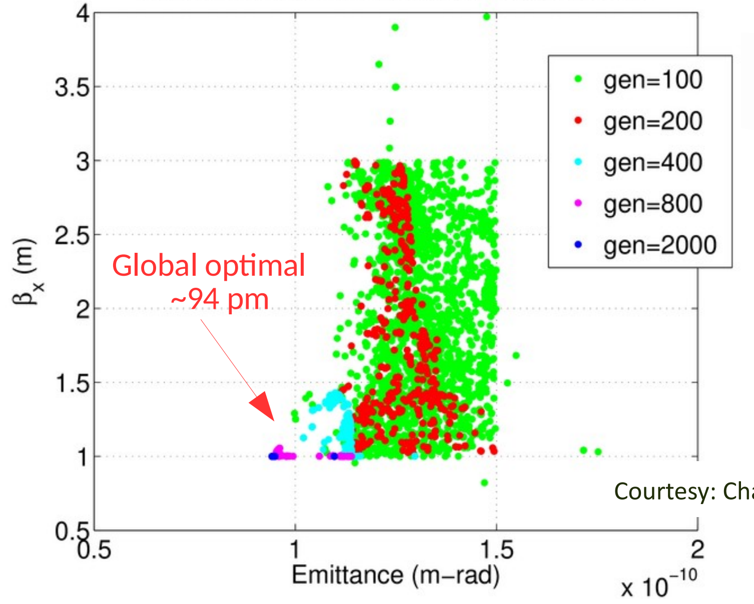
Introduction: The Problem

- **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?*





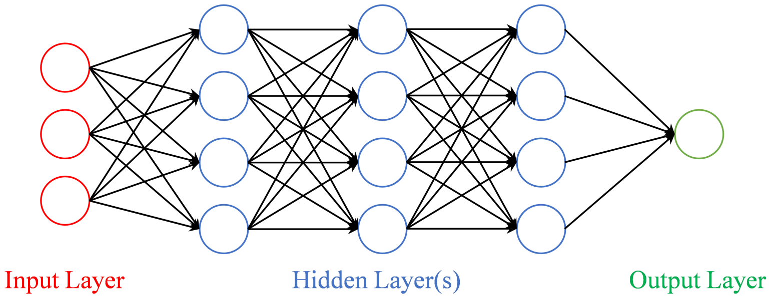
Courtesy: Dave Robin

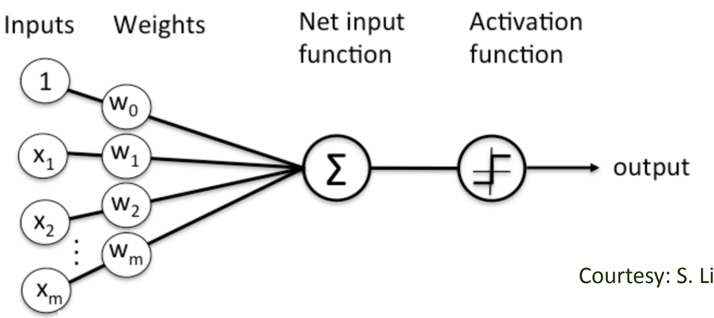


Courtesy: Changchun Sun

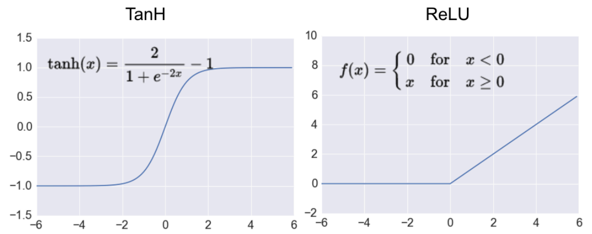
Introduction: Machine Learning (ML) to the Rescue

- 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 [1-4] 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)





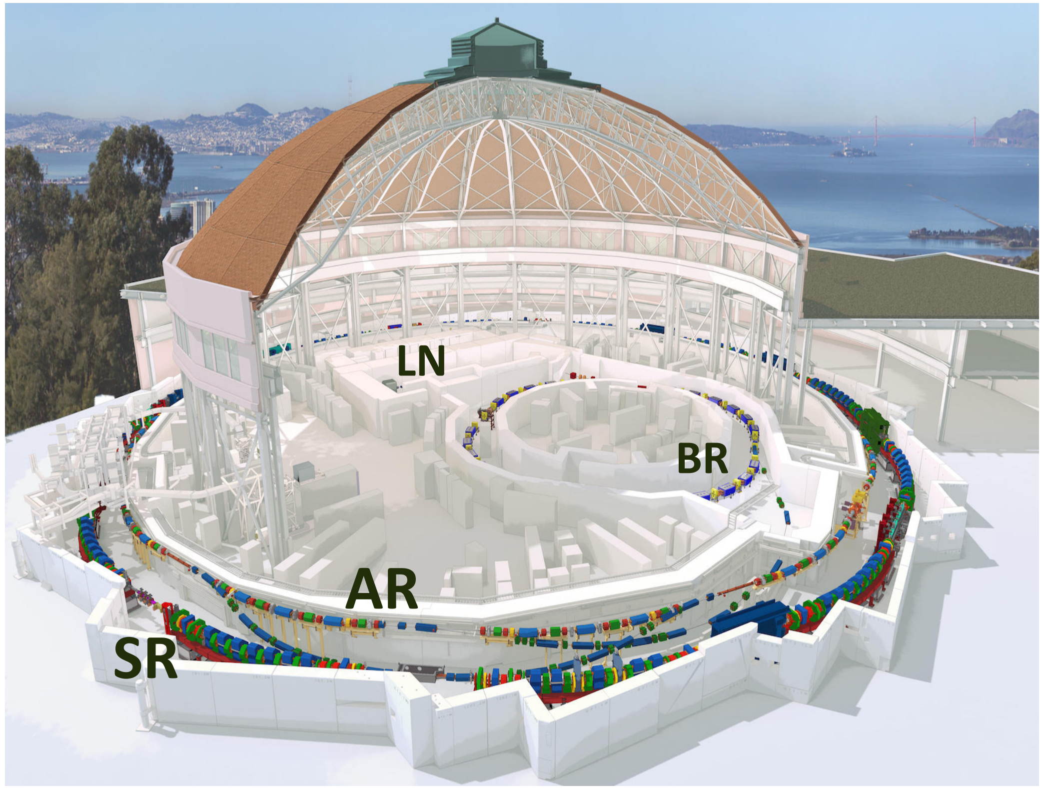
Courtesy: S. Liu

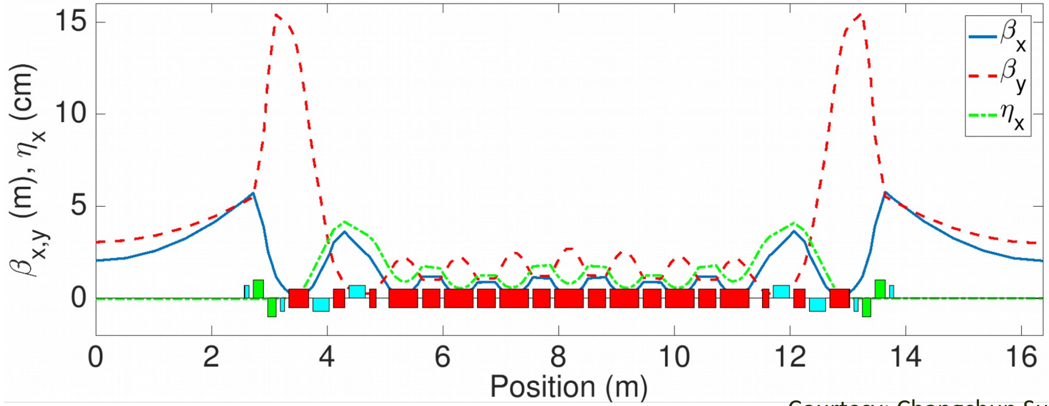


[1] M. Kranjčević, B. Riemann, A. Adelmann, A. Streun, PRAB **24** 014601, 2021.
[2] M. Song, X. Huang, L. Spentzouris, Z. Zhang, NIM-A **976** 164273, 2020.
[3] Y. Li, W. Cheng, L.Yu, R. Rainer, PRAB **21** 054601, 2018.
[4] J. Wan, P. Chu, Y. Jiao, PRAB **23** 081601, 2020.

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

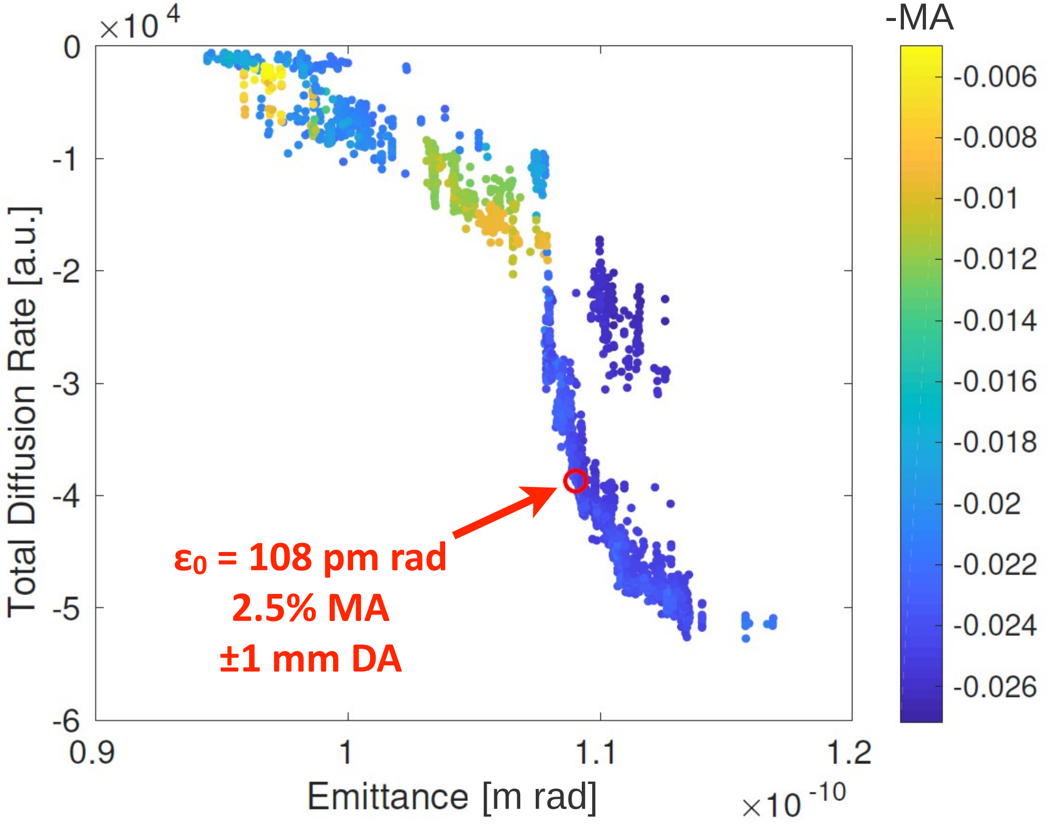




Courtesy: Changchun Sun

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)

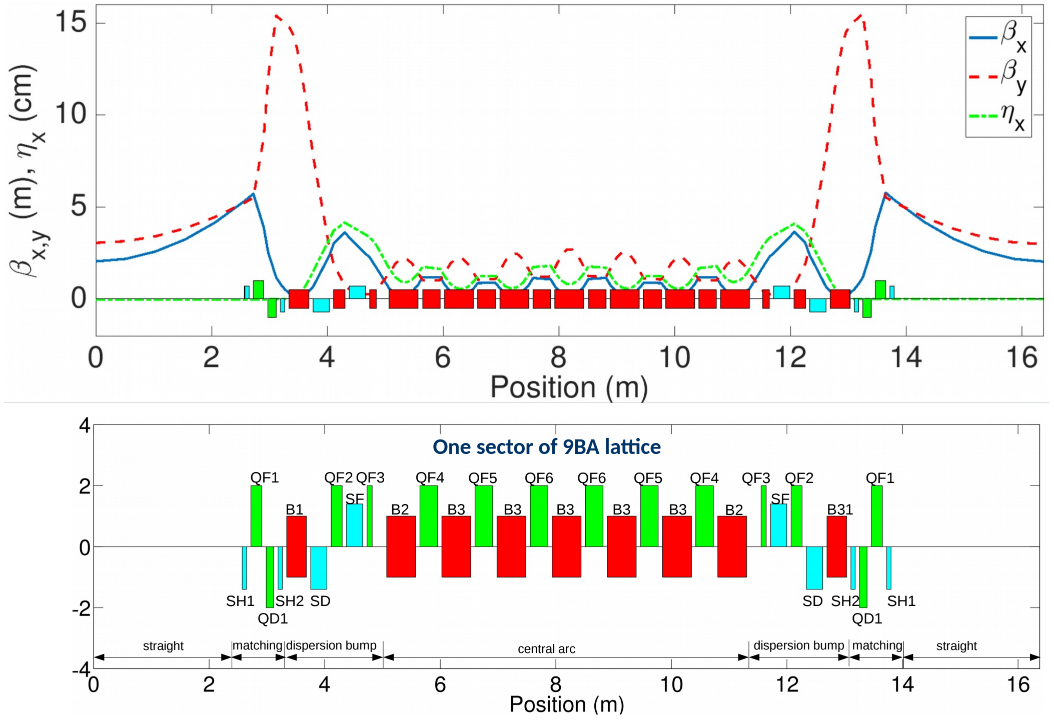


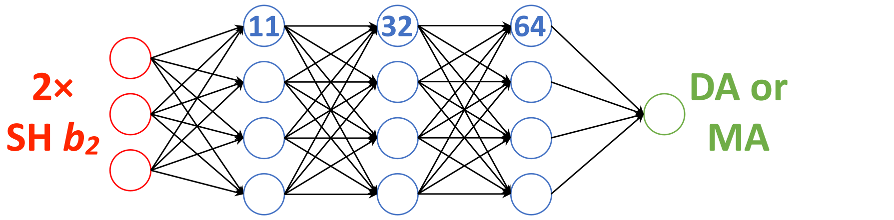
Courtesy: Changchun Sun

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 [5] instead of many-turn tracking with TRACY

Courtesy: Changchun Sun



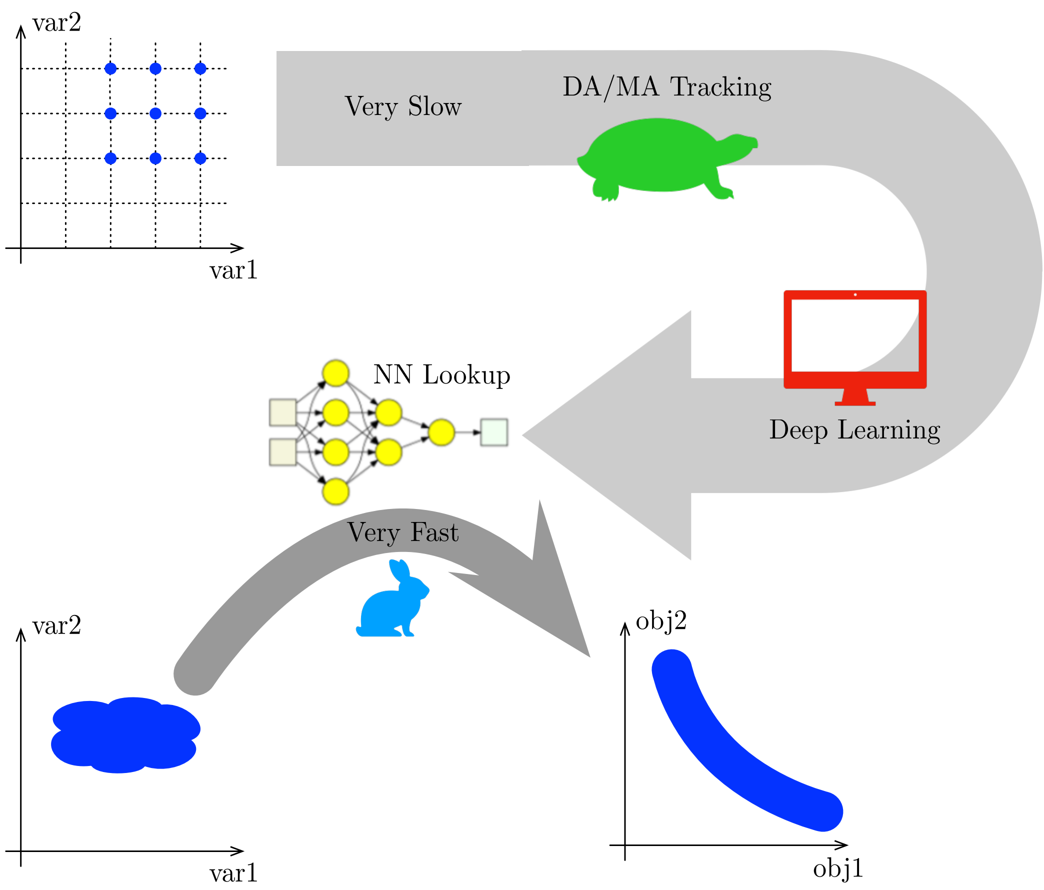


Fully-connected (FC) NN, using ReLU as activation function, # = node depth

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

A First Simple NN for Sextupole Optimization

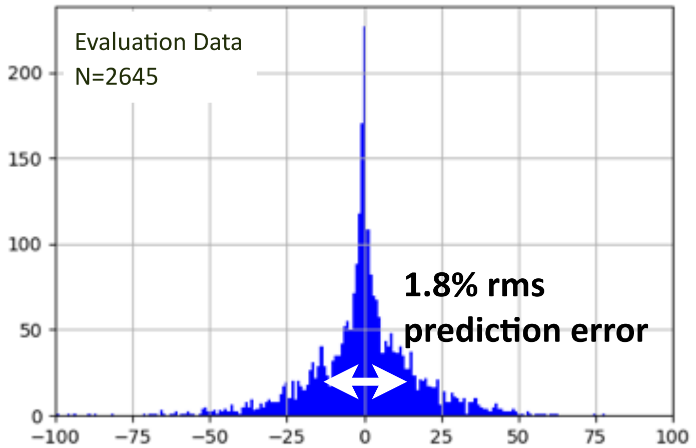
- ALS-U 9BA has **4 sextupole families**: 2 required for chromatic corrections ➔ leaves **2 harmonic families** (SH1 & SH2) for optimization of DA & MA
- Small & simple **3-layer NN** renders accurate prediction of DA/MA as a function of 2 SH variables [5] instead of many-turn tracking with TRACY
  - Training involves low density sampling of 2D input space
  - 20^2 samples tracked for training data ➔ predictions accurate to within ≈2% rms



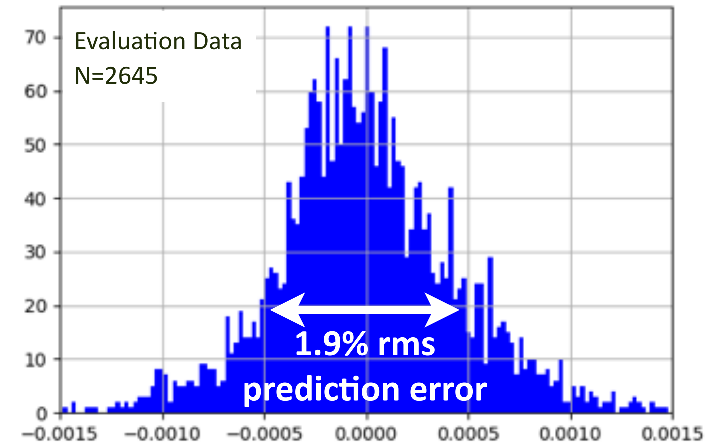
[5] Y. Lu, S.C. Leemann, C. Sun, et al., IPAC2021, MOPAB106, p.387.

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 [5] 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

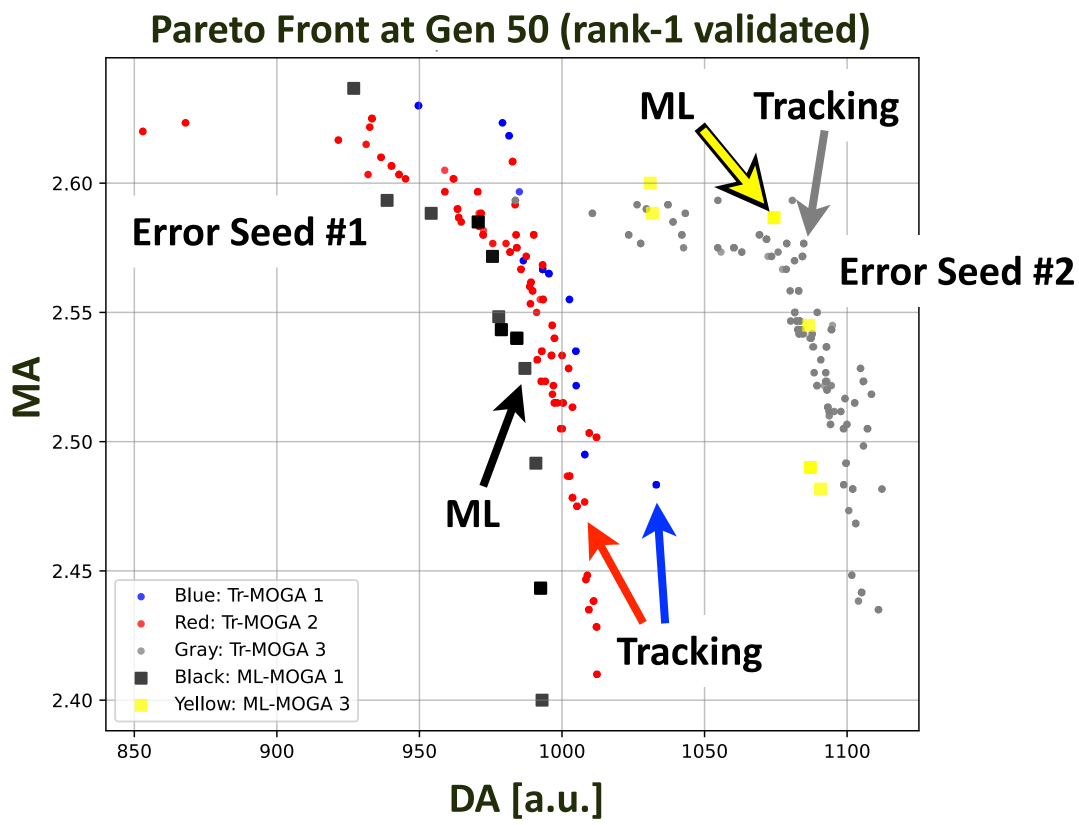


MA NN Prediction Error Histogram

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

A First Simple NN for Sextupole Optimization

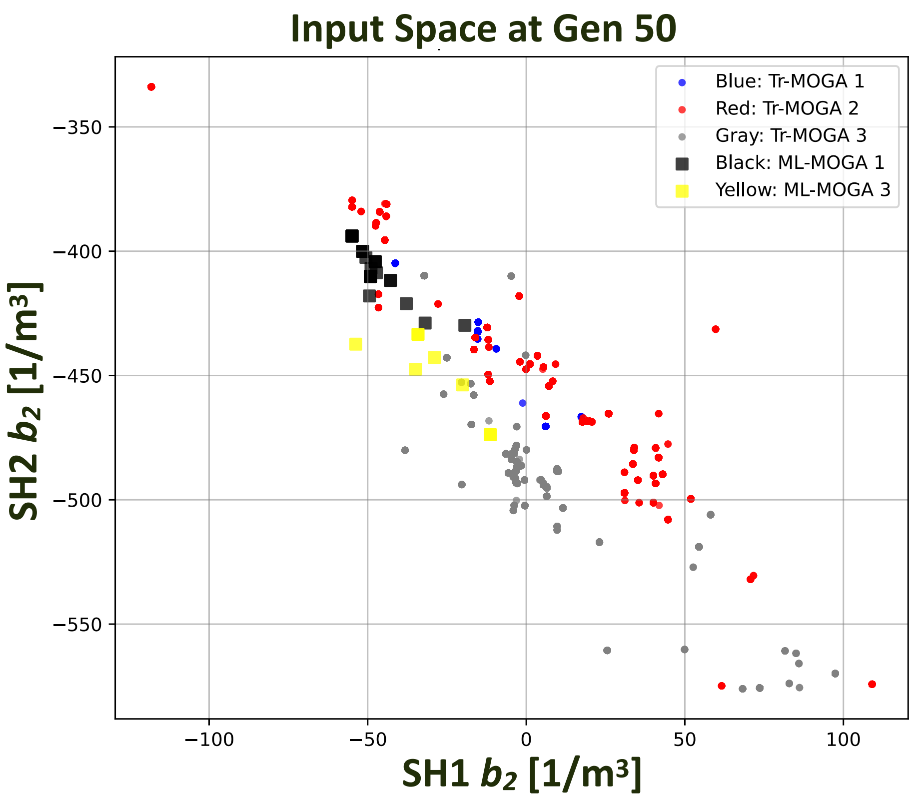
- ALS-U 9BA has **4 sextupole families**: 2 required for chromatic corrections ➔ leaves **2 harmonic families** (SH1 & SH2) for optimization of DA & MA
- Small & simple **3-layer NN** renders accurate prediction of DA/MA as a function of 2 SH variables [5] instead of many-turn tracking with TRACY
  - 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)



[5] 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 [5] 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

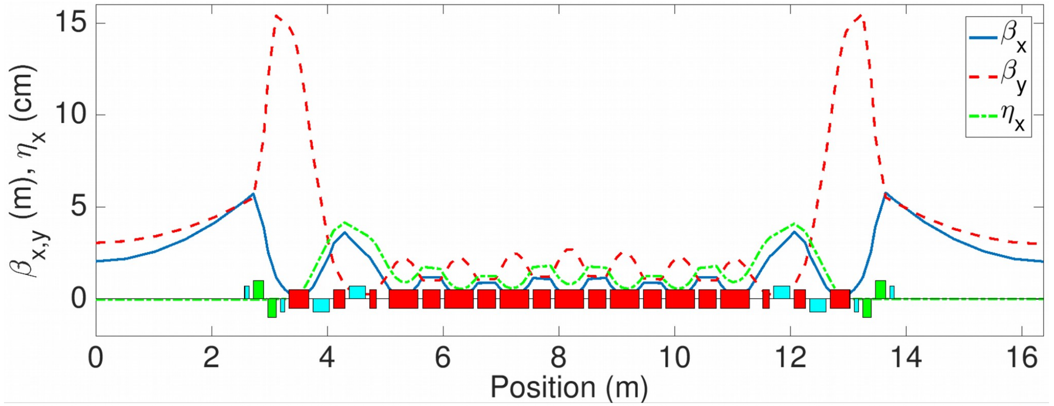


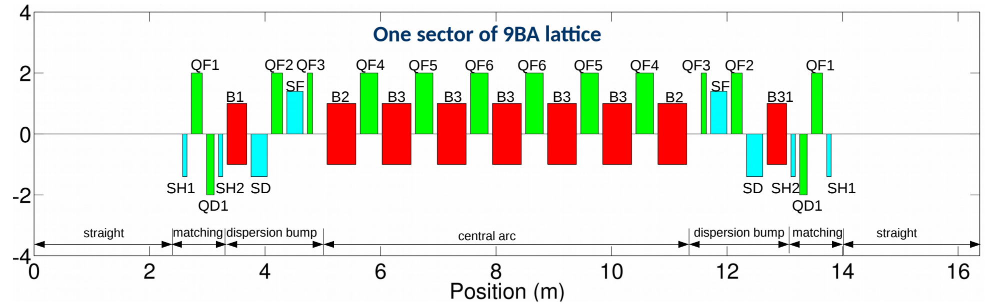
[5] Y. Lu, S.C. Leemann, C. Sun, et al., IPAC2021, MOPAB106, p.387.

ML for Full Linear & Nonlinear ALS-U Optimization

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

Courtesy: Changchun Sun





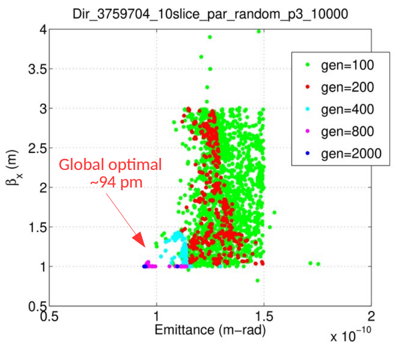
ML for Full Linear & Nonlinear ALS-U Optimization

Linear Opt.

2 Objectives:
- Emittance
- Beta

9 Knobs:
- 9 quad gradient

To explore input parameter and objective spaces

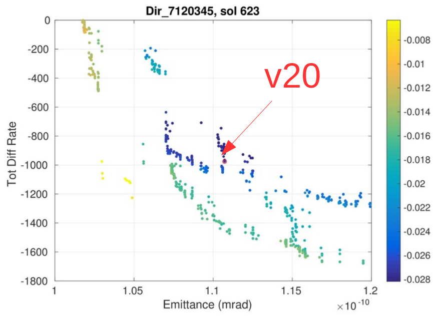


Linear & nonlinear opt.

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

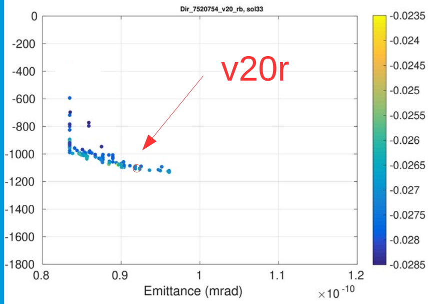


Linear & nonlinear Opt. with reverse bend

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

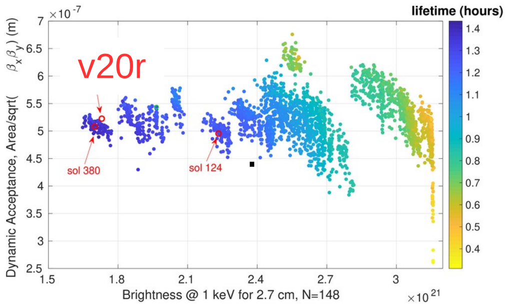


Linear & nonlinear Opt. using alternative objectives

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

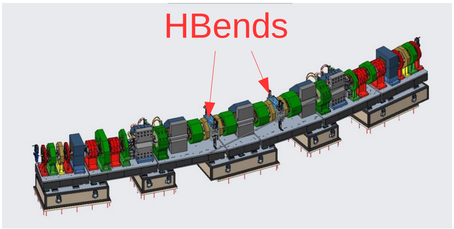


Introduce 3.2T HBend by matching

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



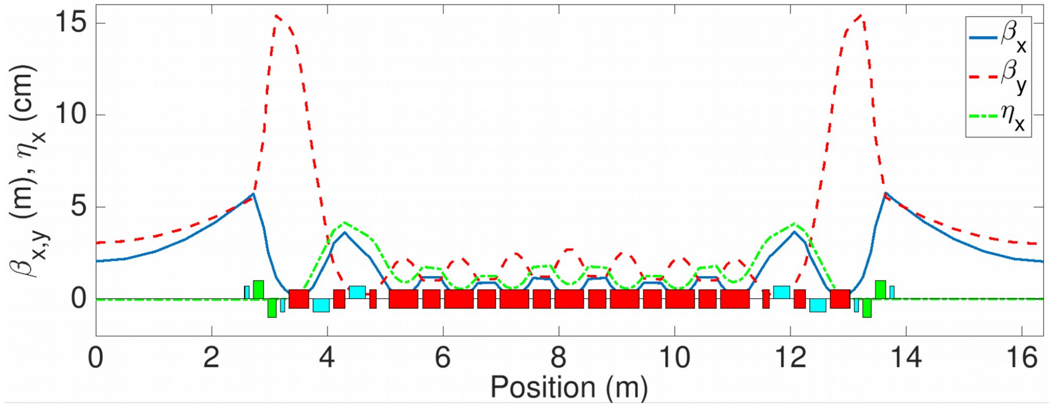
Courtesy: Changchun Sun

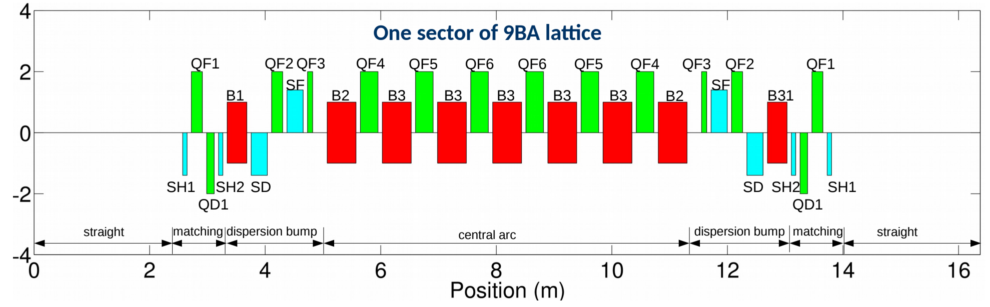
This stage will be focus here

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

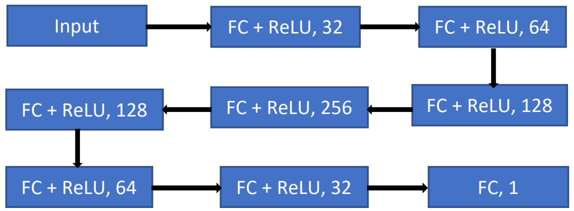




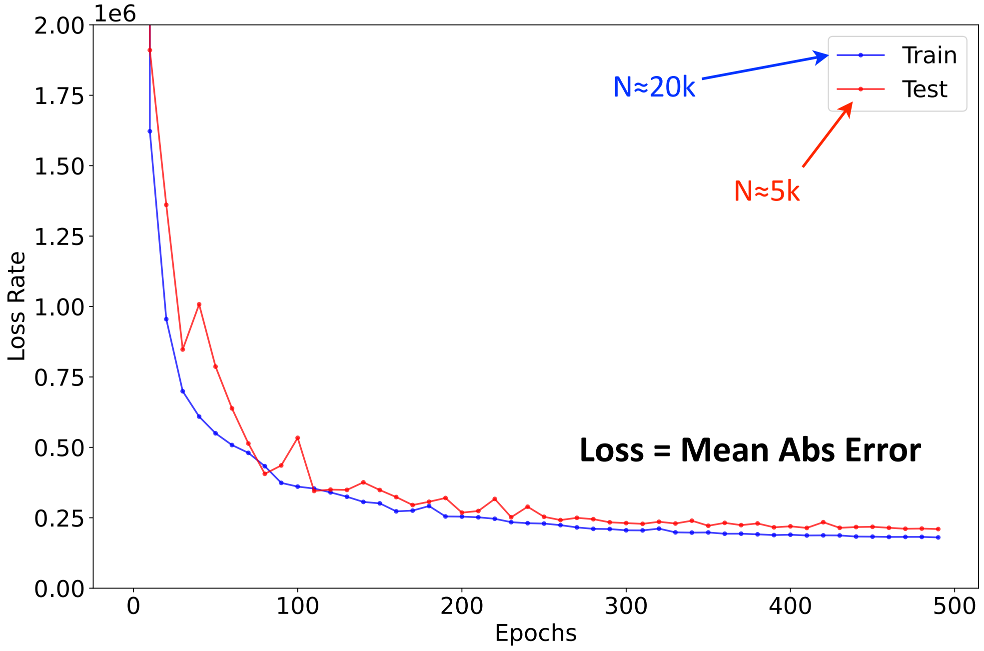
Natural emittance	$\varepsilon_0 < 155\,\mathrm{pm\,rad}$
Maximum beta	$\beta_{x,y} < 30\,\mathrm{m}$
Maximum dispersion	$\eta_x < 15\,\mathrm{cm}$
Fractional tunes	$0.1 < \nu_{x,y} < 0.4$
Dispersion at center of straight	$\lvert\eta_x^*\rvert < 1\,\mathrm{mm}$
Beta at center of straight	$1\,\mathrm{m} < \beta_{x,y}^* < 5\,\mathrm{m}$
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)

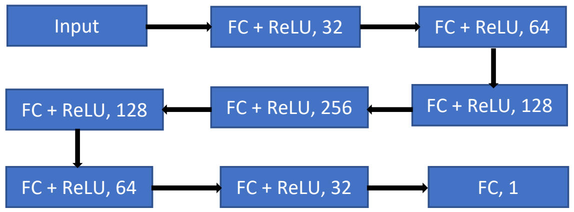


Fully-connected (FC) NN, using ReLU as activation function, # = node depth

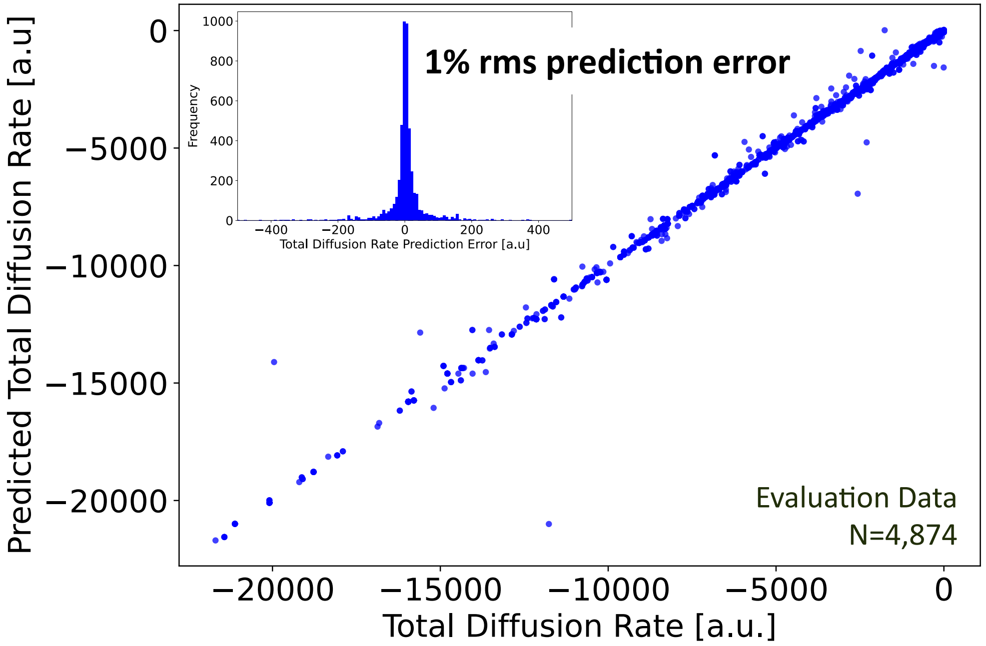


ML for Full Linear & Nonlinear ALS-U Optimization (cont.)

- 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)

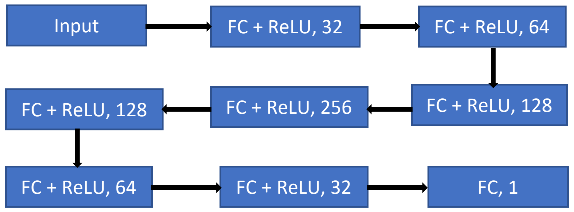


Fully-connected (FC) NN, using ReLU as activation function, # = node depth

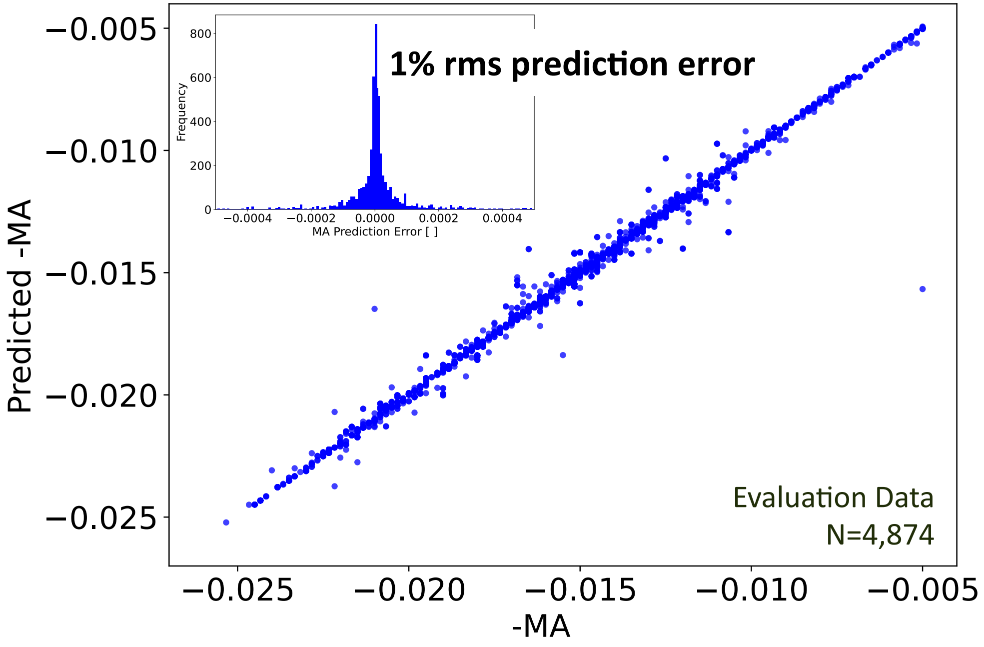


ML for Full Linear & Nonlinear ALS-U Optimization (cont.)

- 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)

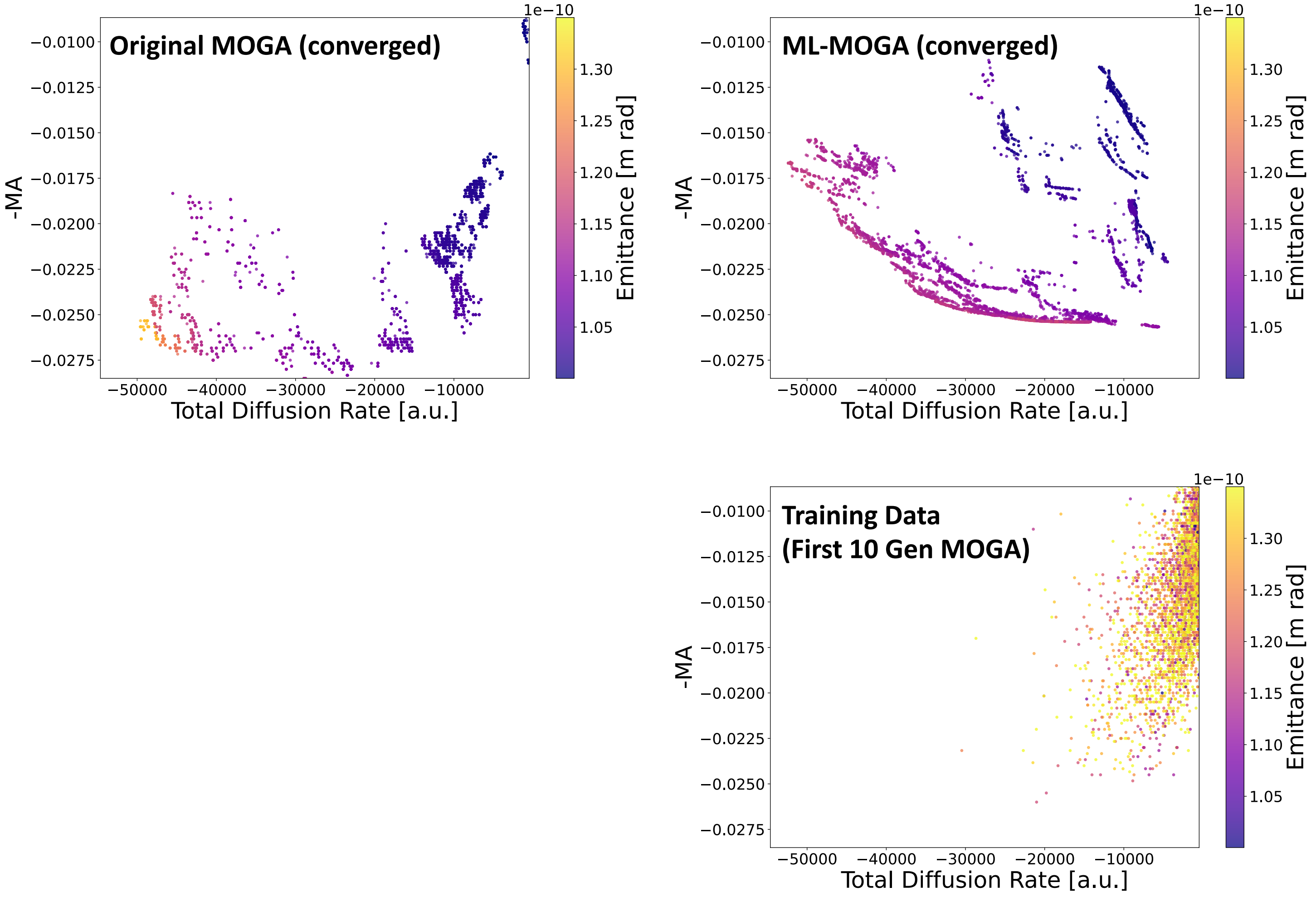


Fully-connected (FC) NN, using ReLU as activation function, # = node depth



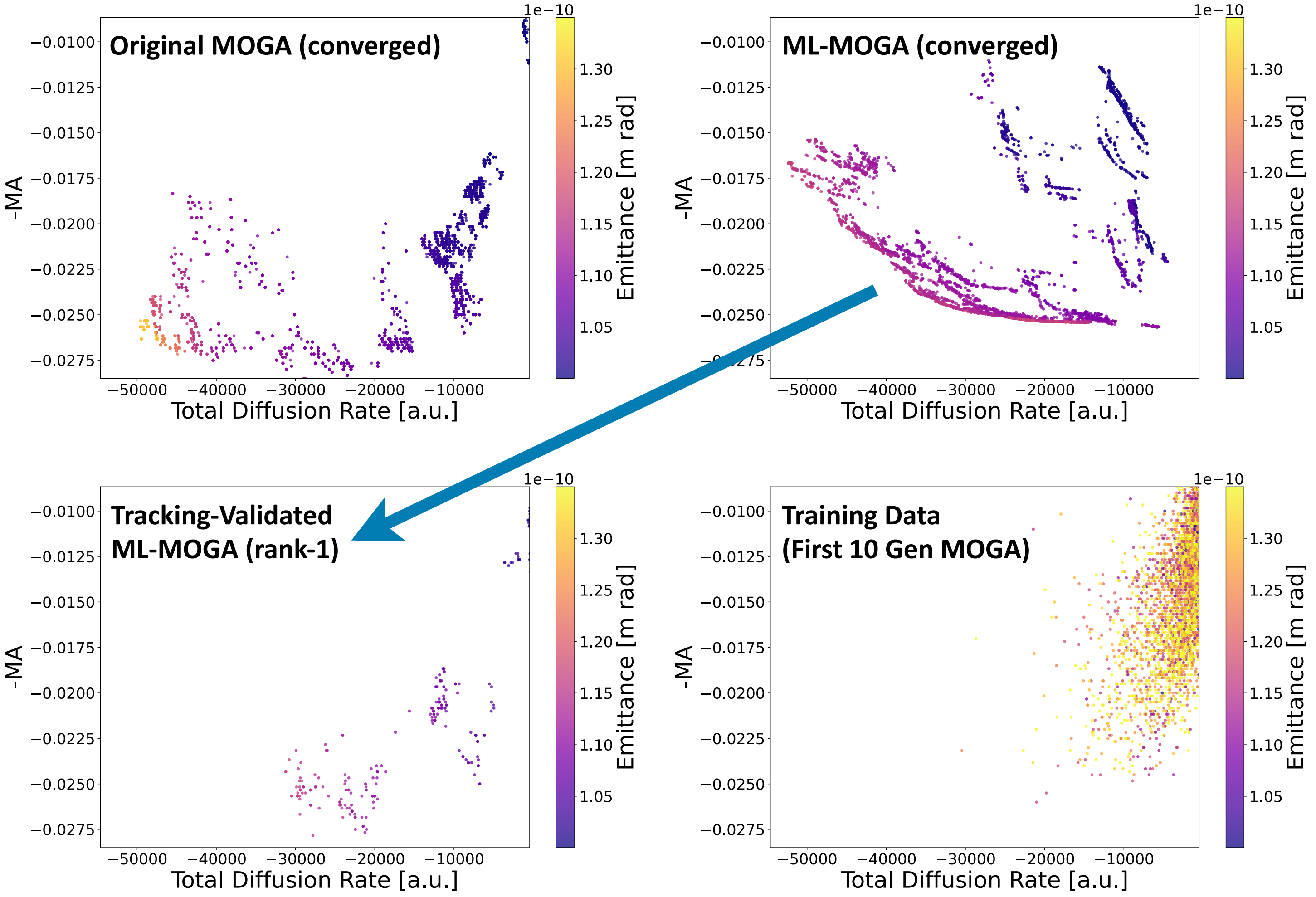
But of course it's a bit more complicated…

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



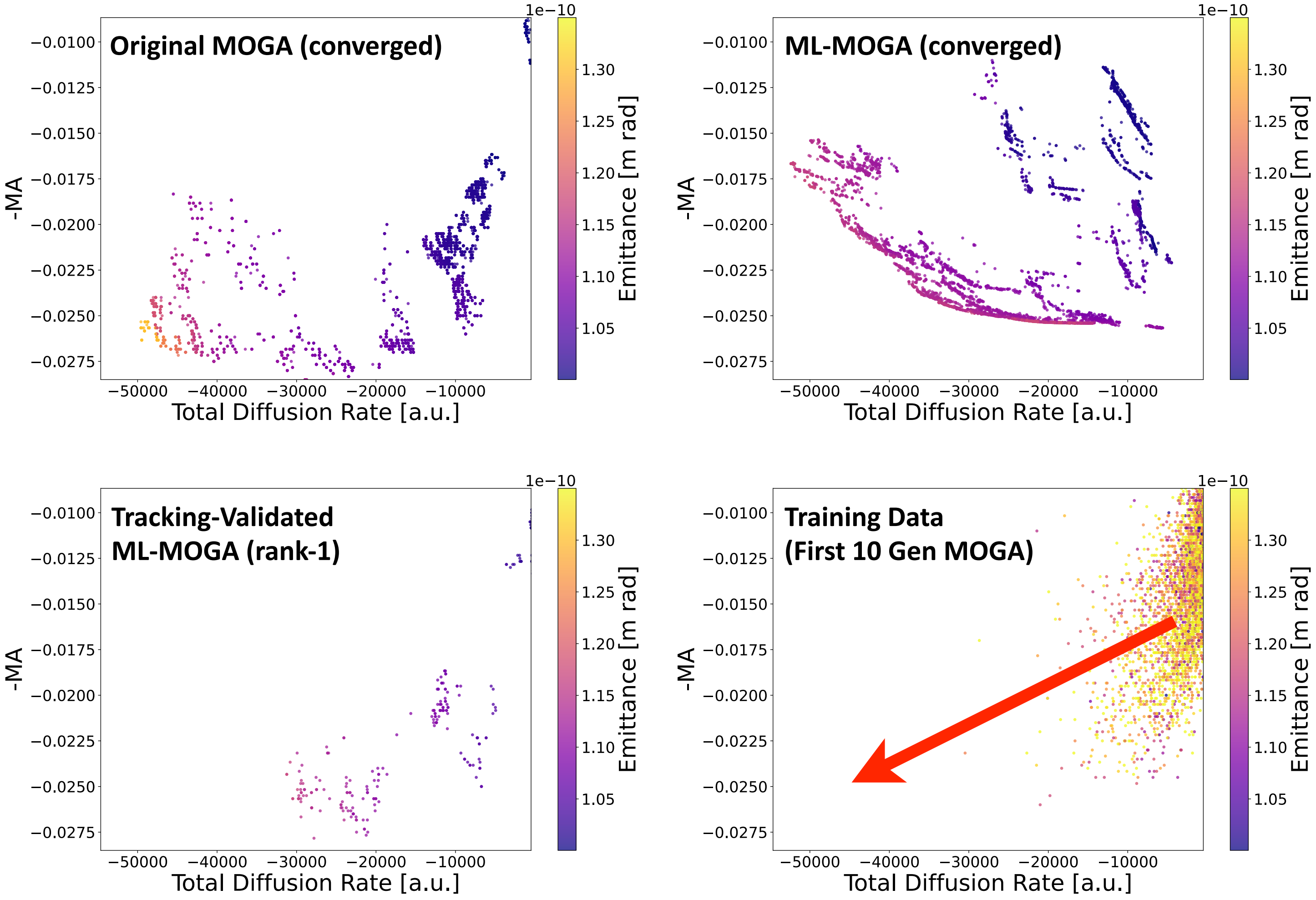
But of course it’s a bit more complicated…

- 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…

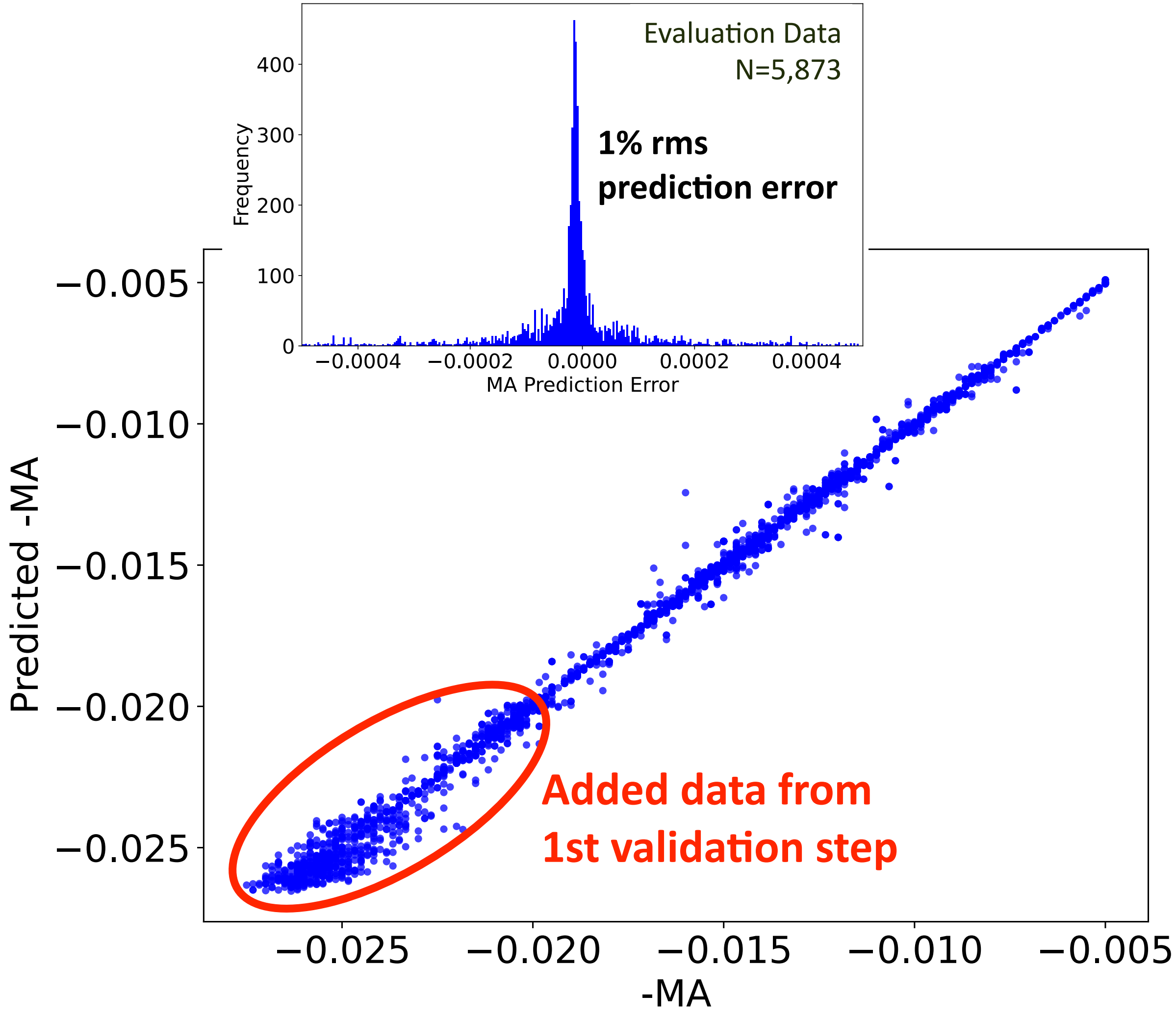
- ML predictions are not 100% accurate (training data based on initial optimization data ➔ potentially far from Pareto-optimal areas in input space)
- ML-MOGA solutions show disagreement to **tracking validation** ➔ converged solution front is not entirely non-dominated
- Want to **retrain DNNs** with an improved resampling of input space ➔ more samples closer to optimal solutions as in [6], …
- …but here for a many-dimensional input space without making any assumptions on smoothness of distributions



[6] 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 $\{\varepsilon_0, D_0, M_0\}$

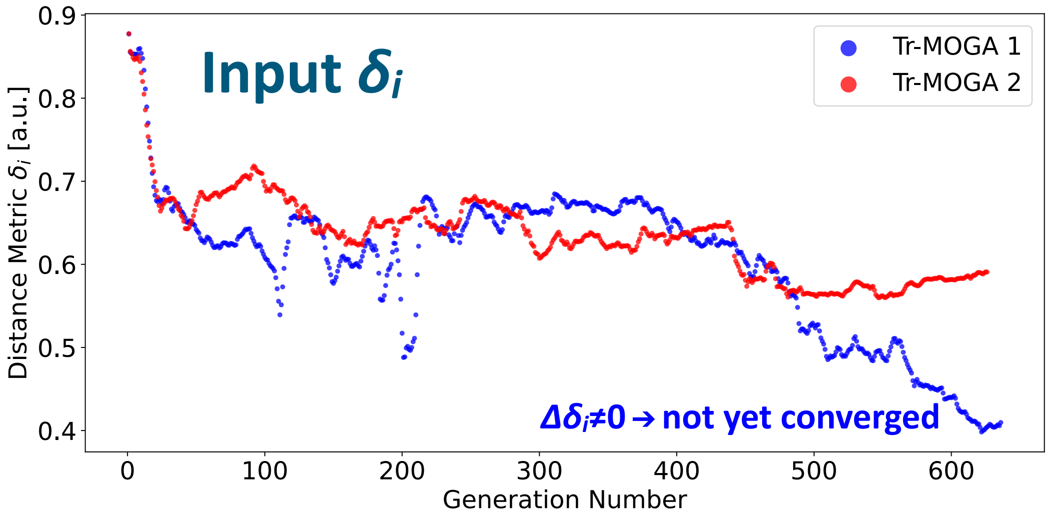
$$\Delta_f = \delta_o^v(m_f) - \delta_o(m_f)$$

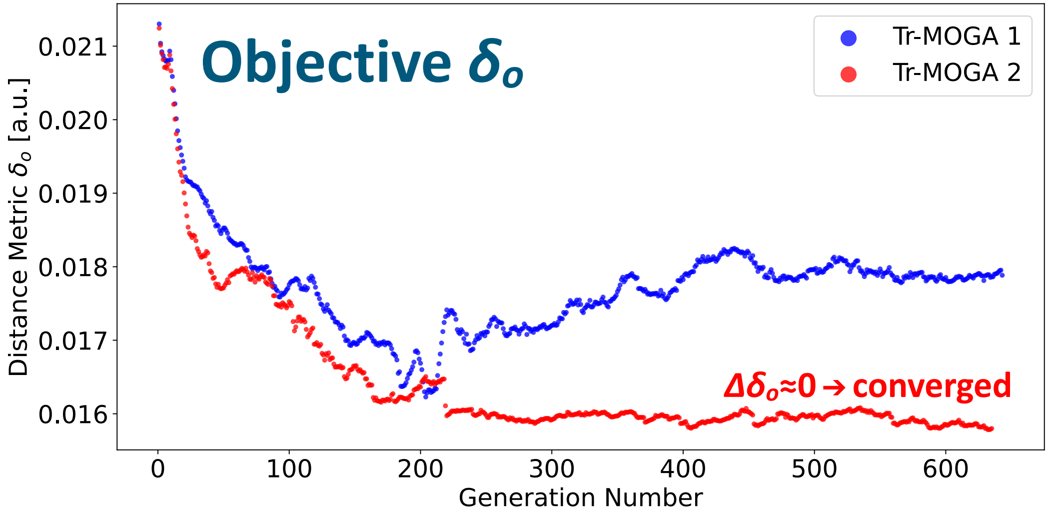
Tracking Validated δ_o

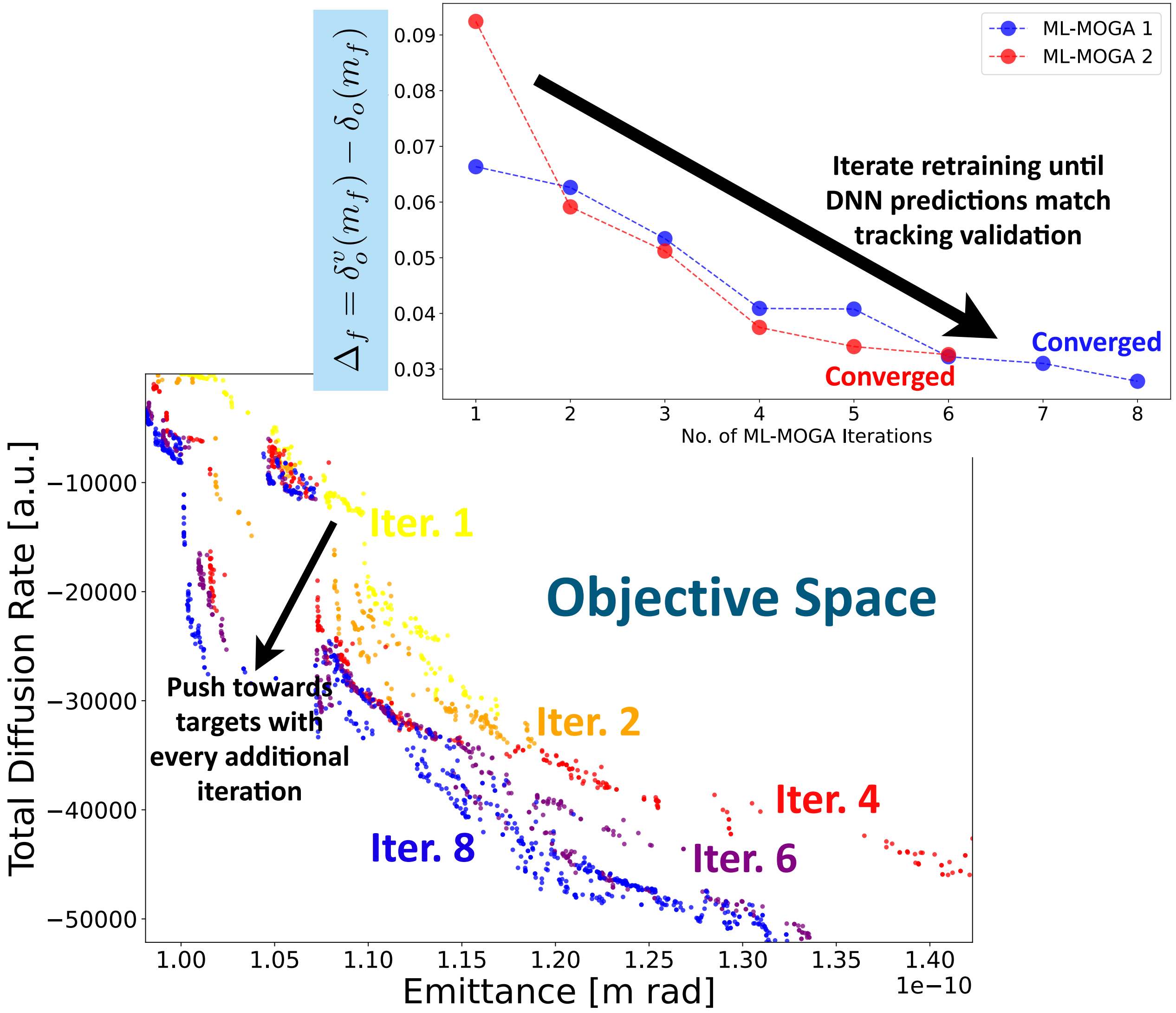
Final gen m_f

NIM-A **1050**, 168192 (2023)

Distance Metrics & Convergence (cont.)



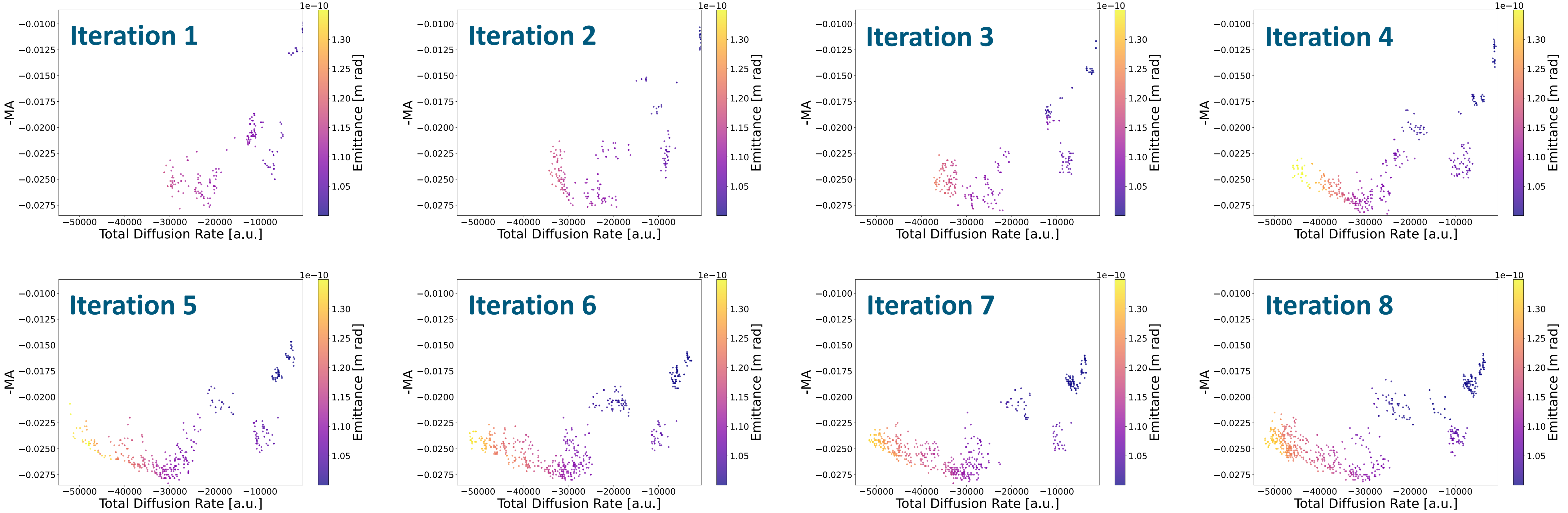




Results

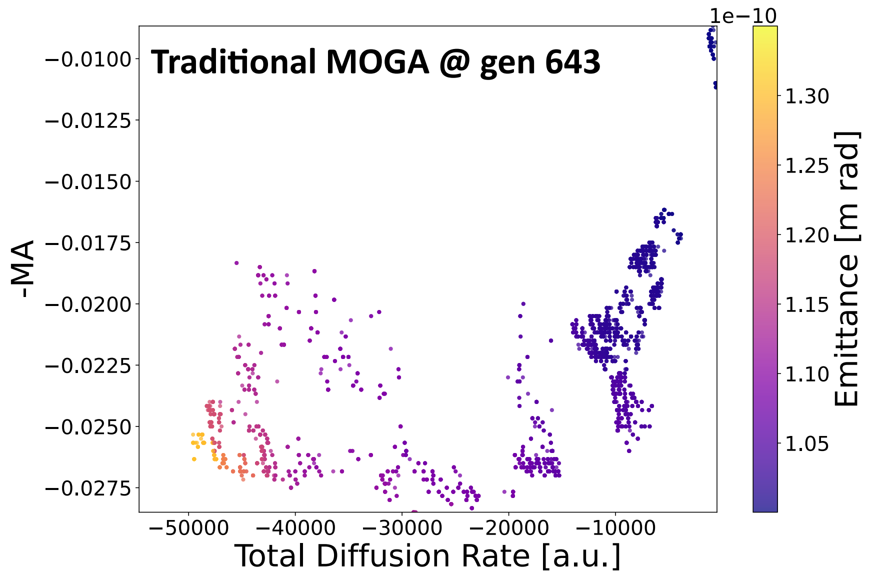
NIM-A **1050**, 168192 (2023)

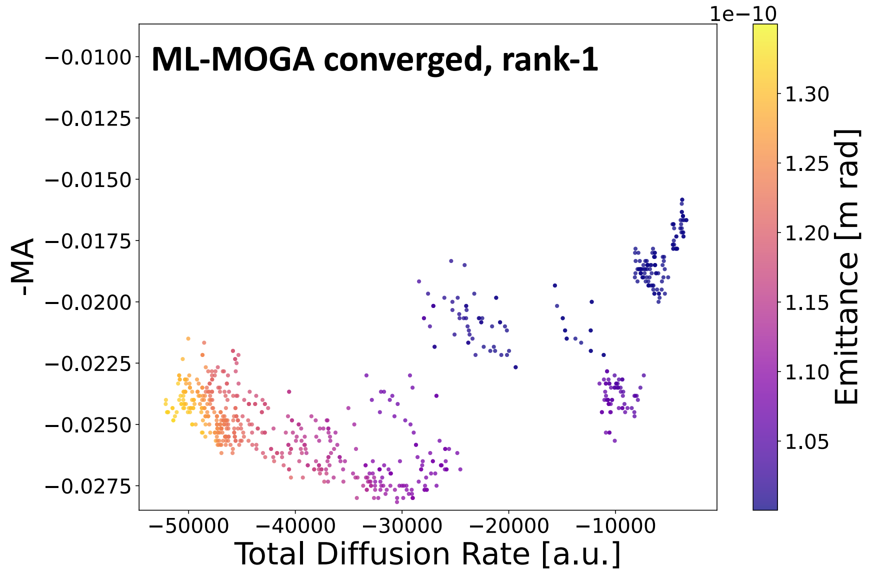
- Retraining shows very quick convergence



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"

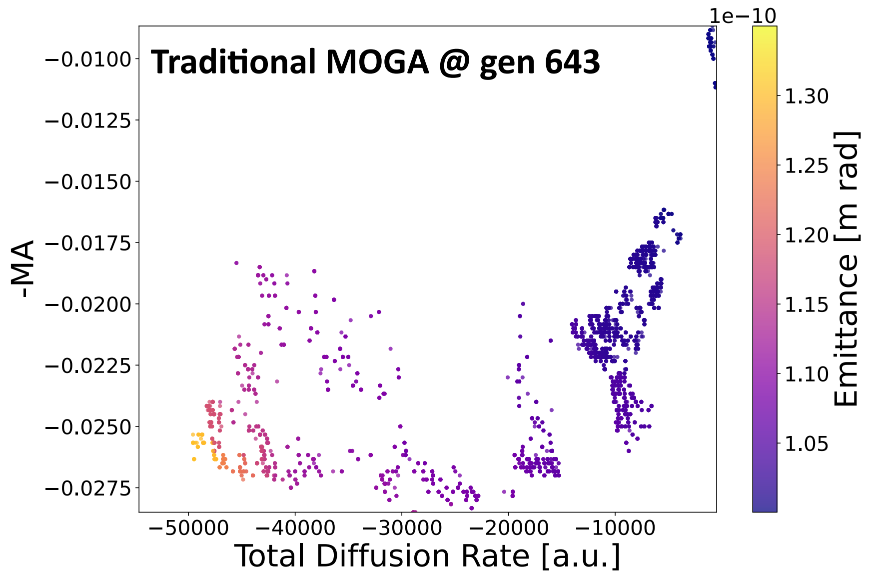


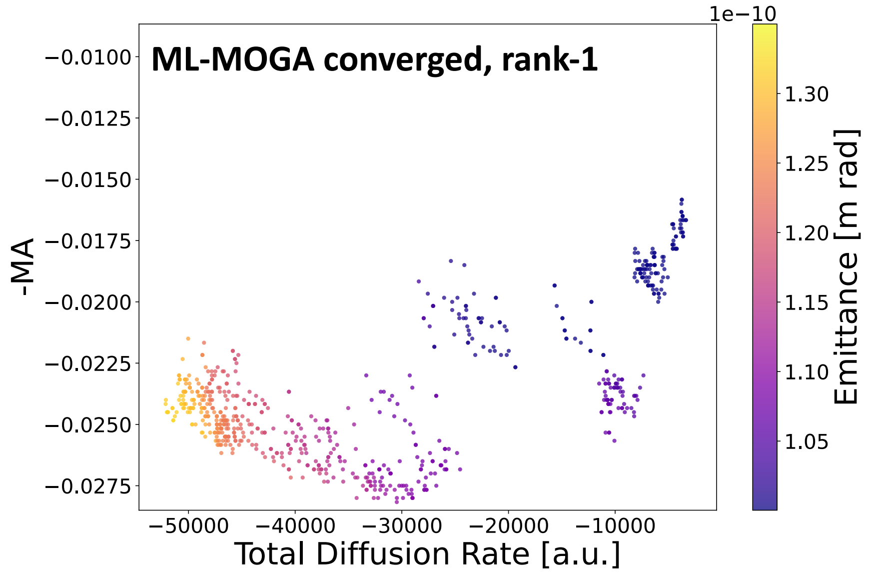


NIM-A **1050**, 168192 (2023)

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)

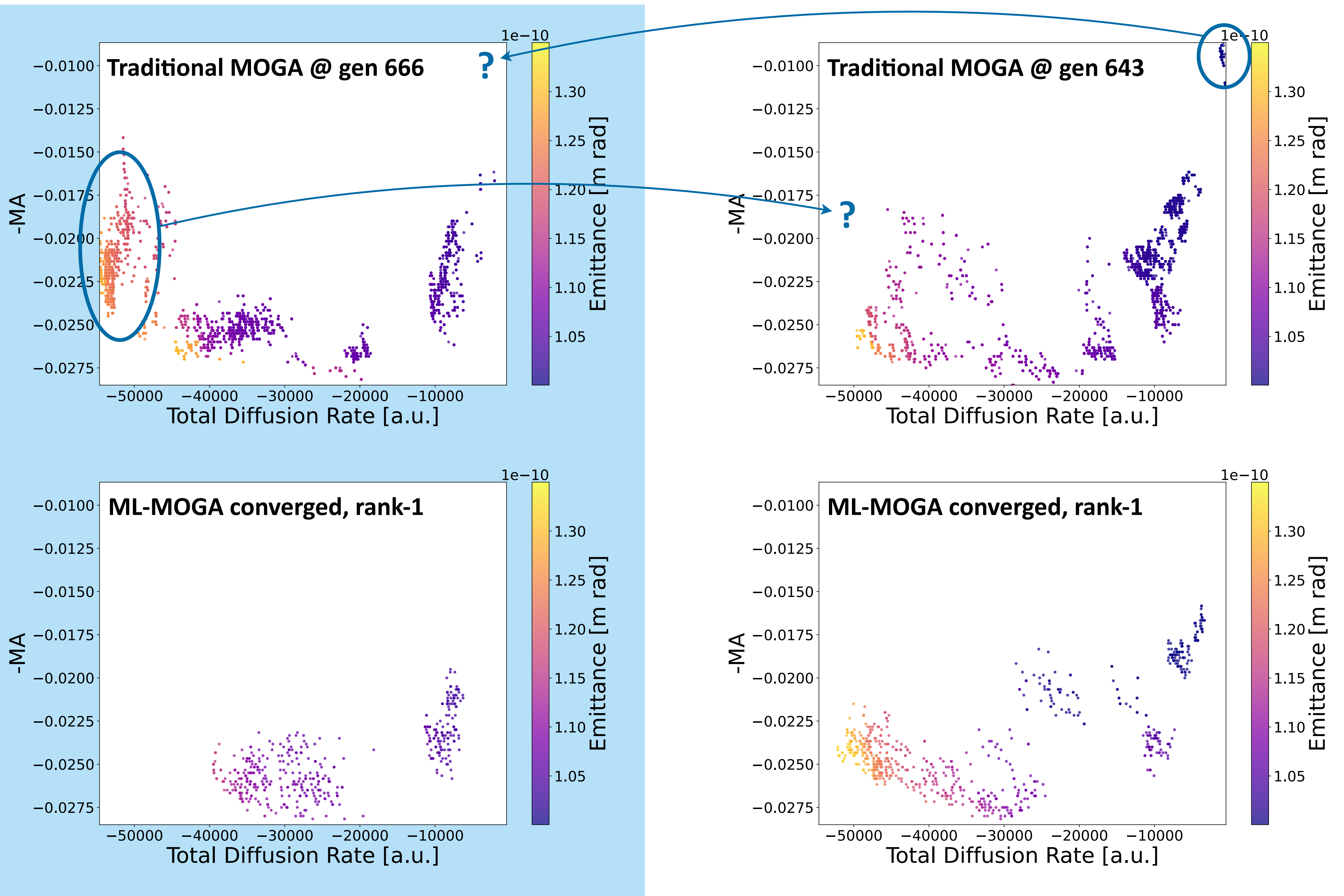




NIM-A **1050**, 168192 (2023)

Results

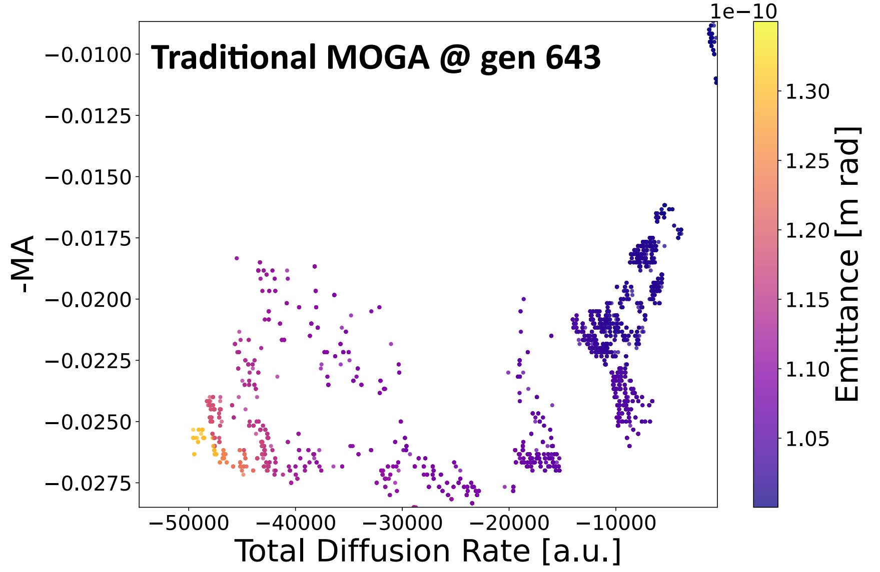
Only MOGA random seed changed → same physics

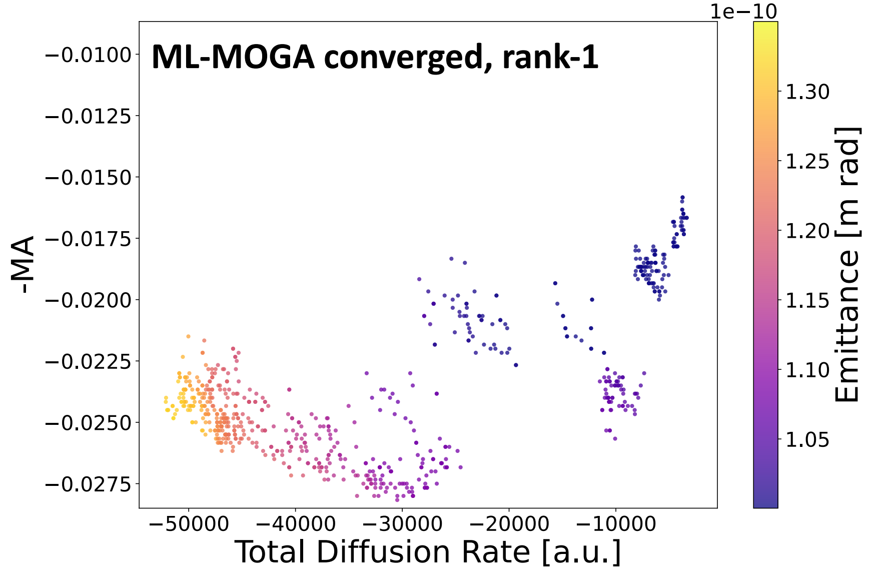


Results

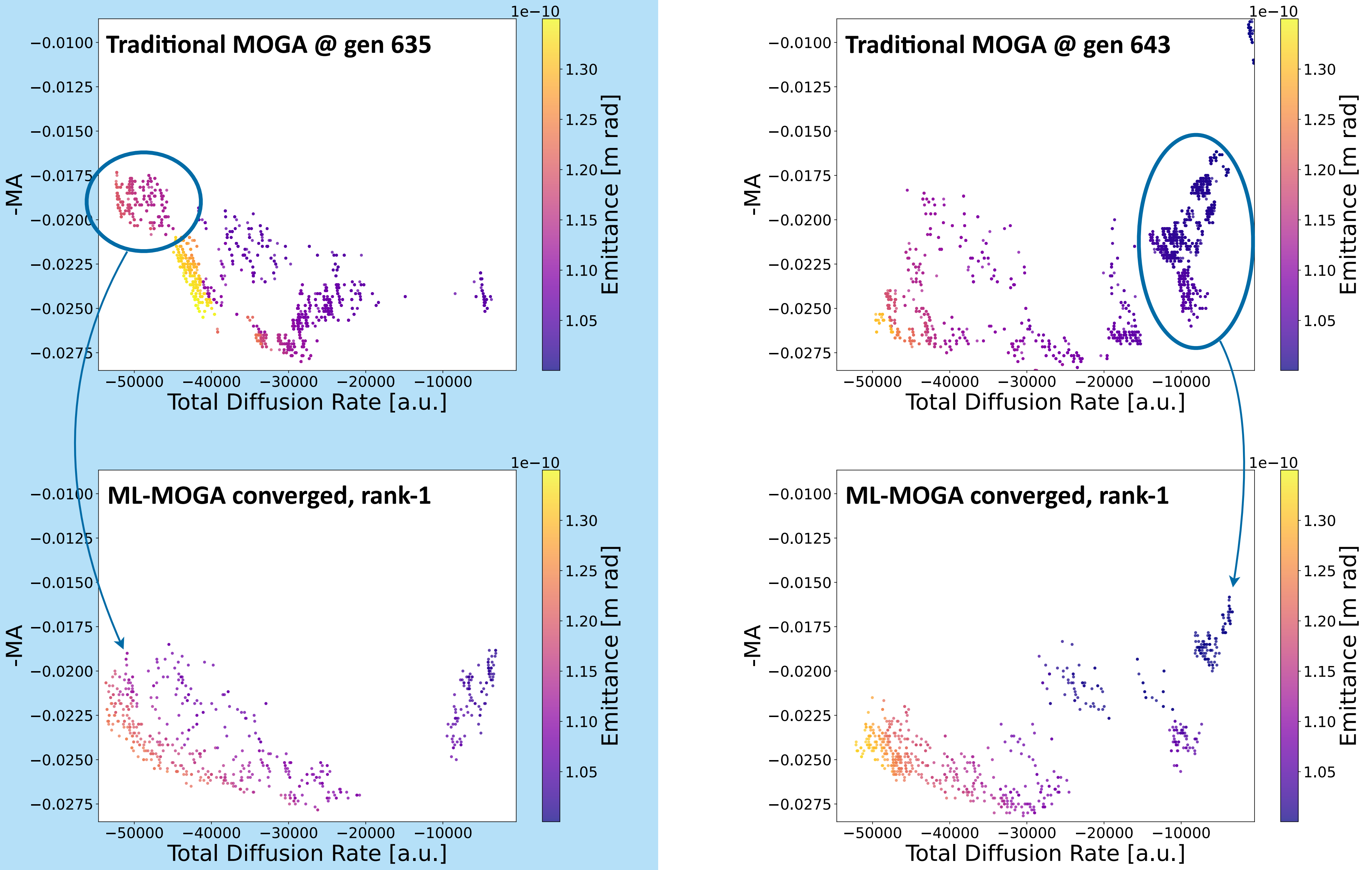
NIM-A **1050**, 168192 (2023)

- 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

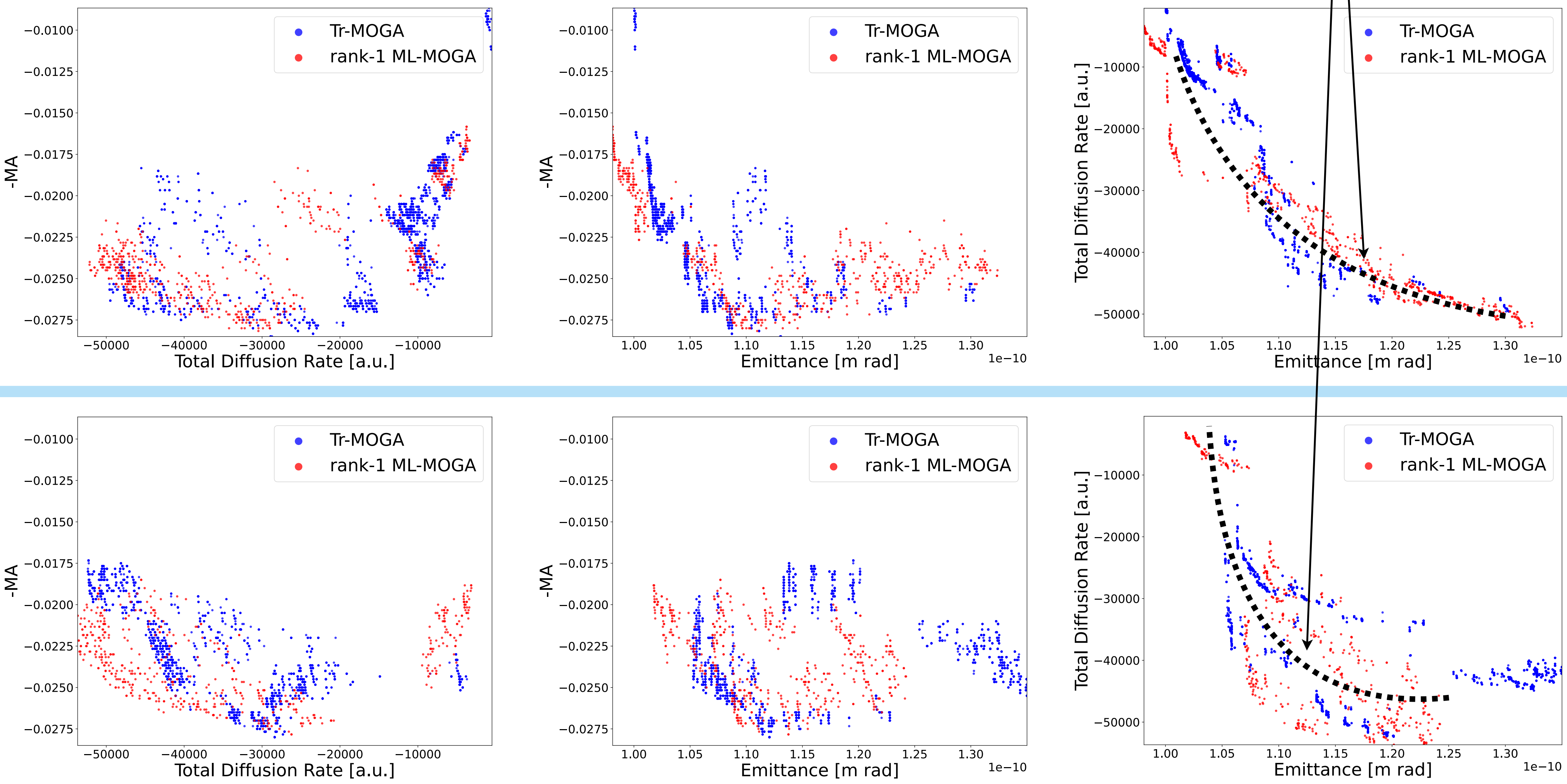
Results

NIM-A **1050**, 168192 (2023)

Primary trade-off correctly identified in both cases

Error Set #1

Error Set #2



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**

NIM-A **1050**, 168192 (2023)

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