LABORATÓRIO DE INSTRUMENTAÇÃO
E FÍSICA EXPERIMENTAL DE PARTÍCULAS
partículas e tecnologia

Exploring Parameter Spaces with Artificial Intelligence and Machine Learning Black-Box Optimisation Algorithms

In collaboration with Fernando Abreu de Souza, Nuno Filipe Castro, Mehraveh Nikjoo, Werner Porod

SUSY 2022 - Ioannina

Miguel Crispim Romão
LIP - Minho
mcromao@lip.pt

Funding recognition
For supporting this work and talk

QML-HEP: Exploring quantum machine learning as a tool for present and future high-energy colliders

OE, FCT-Portugal, CERN/FIS-PAR/0004/2021

Fundação para a Ciência e a Tecnologia

REPÚBLICA PORTUGUESA

Machine Learning in HEP

Machine Learning in HEP
A flourishing area of research

https://iml-wg.github.io/HEPML-LivingReview/

HEPML-LivingReview

A Living Review of Machine Learning for Particle Physics

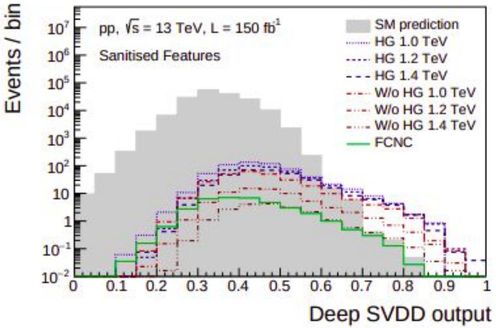
Modern machine learning techniques, including deep learning, is rapidly being applied, adapted, and developed for high energy physics. The goal of this document is to provide a nearly comprehensive list of citations for those developing and applying these approaches to experimental, phenomenological, or theoretical analyses. As a living document, it will be updated as often as possible to incorporate the latest developments. A list of proper (unchanging) reviews can be found within. Papers are grouped into a small set of topics to be as useful as possible. Suggestions are most welcome.

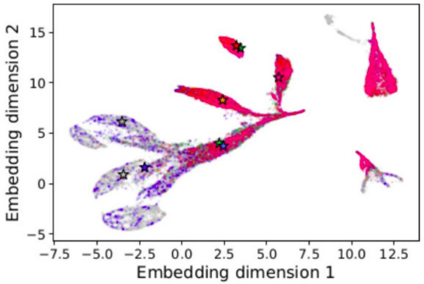
download review

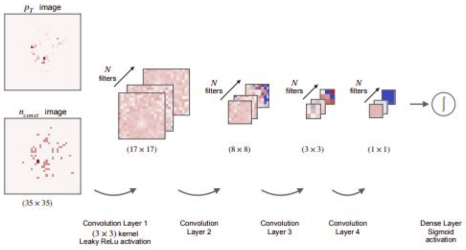
Impossible to cover everything here...

Machine Learning in HEP
Shameless self-promotion

- Generic Searches for New Physics
  - Transferability of Deep Learning Models in Searches for New Physics at Colliders, Phys. Rev. D 101, 035042 (2020), 1912.04220
  - Finding New Physics without learning about it: Anomaly Detection as a tool for Searches at Colliders, Eur.Phys.J.C 81 (2021), 2006.05432
  - [WIP various still this year]
- Grouping Events Together
  - Use of a Generalized Energy Mover's Distance in the Search for Rare Phenomena at Colliders, Eur. Phys. J. C 81, 192 (2021), 2004.09360
- Jet Phenomenology for Quark-Gluon Plasma Studies
  - Deep Learning for the Classification of Quenched Jets, JHEP11 (2021) 219, 2106.08869
  - [WIP, soon w/ G. Milhano, M. van Leewuven] Data driven exploration of jet observables and their sensitivity to jet quenching
- Quantum Machine Learning in HEP
  - Funded exploratory project with various WIP works (hopefully out this year)







Machine Learning in HEP
Focus of applications

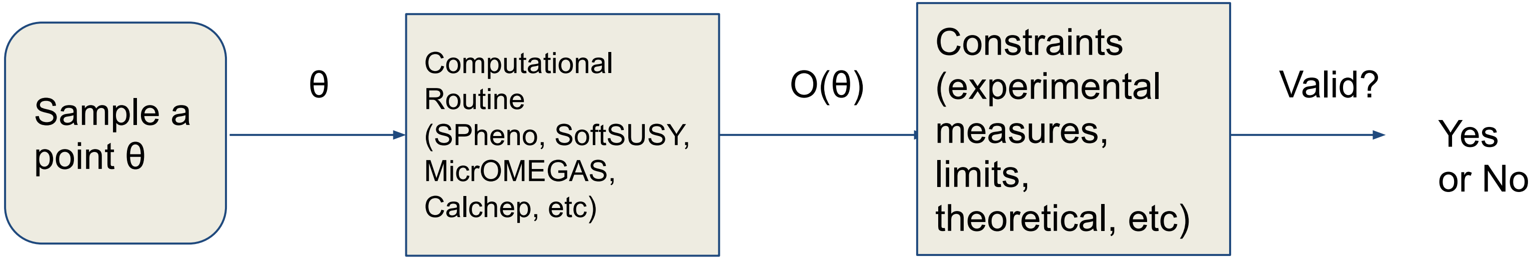
The main applications and interests of ML in HEP have been very focused on the experimental needs

- Tagging
  - Current analyses already use Neural Net taggers
  - Possibility of using very low-level information (4-momenta, jet images)
- Better Analysis Sensitivity
  - Including efforts for generic New Physics discriminants
- Monitoring, Pileup Mitigation, Control, Design, Jet Phenomenology, etc
- **What about Model Building and Phenomenology?**

Machine Learning and Model Building and Phenomenology

Machine Learning in HEP
Applications to model building and pheno

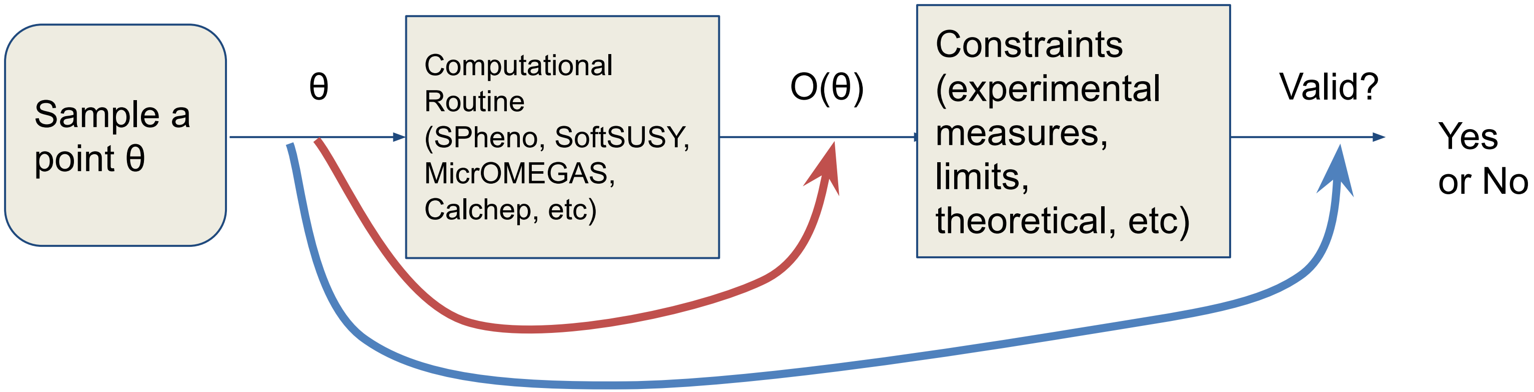
- Parameter space scanning is usually **computationally** and **time consuming**
- Difficulty increases for highly constrained cases: **low parameter sampling efficiency**



For more difficult scans one usually adapts for simplicity (e.g. known alignment limits, a priori choice of parameter space corners, less constraints, etc)

Machine Learning in HEP
Applications to model building and pheno

- Considering that the **observable computation** is the heavy step, try to **replace it**, either by **predicting the observables (regression)** or **predicting if a point is valid (classification)**

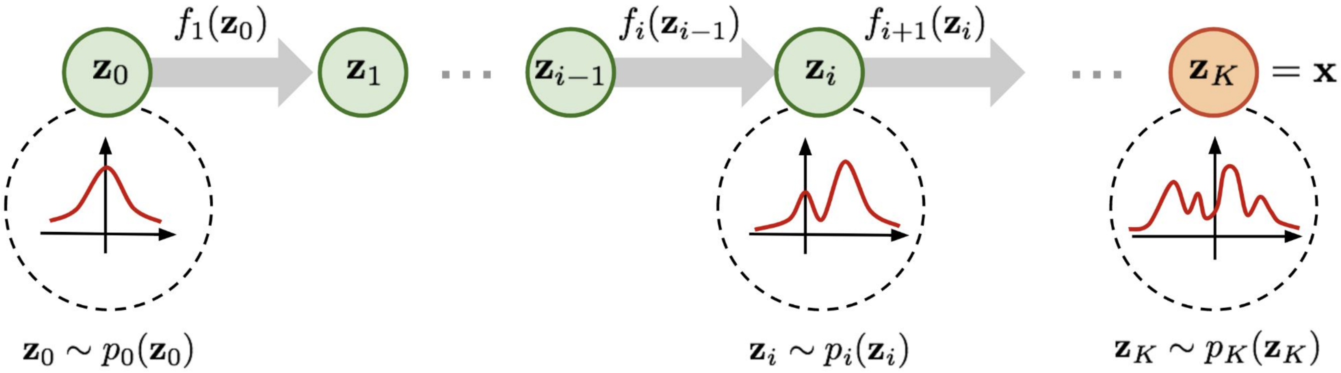


Kronheim, et al [2007.04506]

Caron, et al [1605.02797]; Ren, et al [1708.06615]; Staub [1906.03277]

Machine Learning in HEP
Applications to model building and pheno

- Another approach is to use generative Machine Learning to generate new points from a collection of valid points
  - Hollingsworth, et al [2103.06957] using Normalizing Flow Networks and compared to a DNN Regressor+Monte Carlo sampler



https://lilianweng.github.io/posts/2018-10-13-flow-models/

Machine Learning in HEP

Applications to model building and pheno

- However, these methodologies require **large amounts of training data** which **cover the whole parameter space**
- Predicting the observables from a parameter point using a **regressor**:
    - If training data do not cover the whole parameter space… **might map the parameter to observables incorrectly**
- Predicting if a point is valid or invalid using a **classifier**:
    - If training data do not cover the whole parameter space… **wrong guess**
- **Resampling** from valid points using generative methods:
    - If training data do not cover the whole parameter space… we might end up resampling from a **subset of the parameter space**
- **For highly constrained and realistic scans, it is computationally prohibitive to get enough valid points to use some of these methods**

Exploring Parameter Spaces with ML/AI Search Algorithms

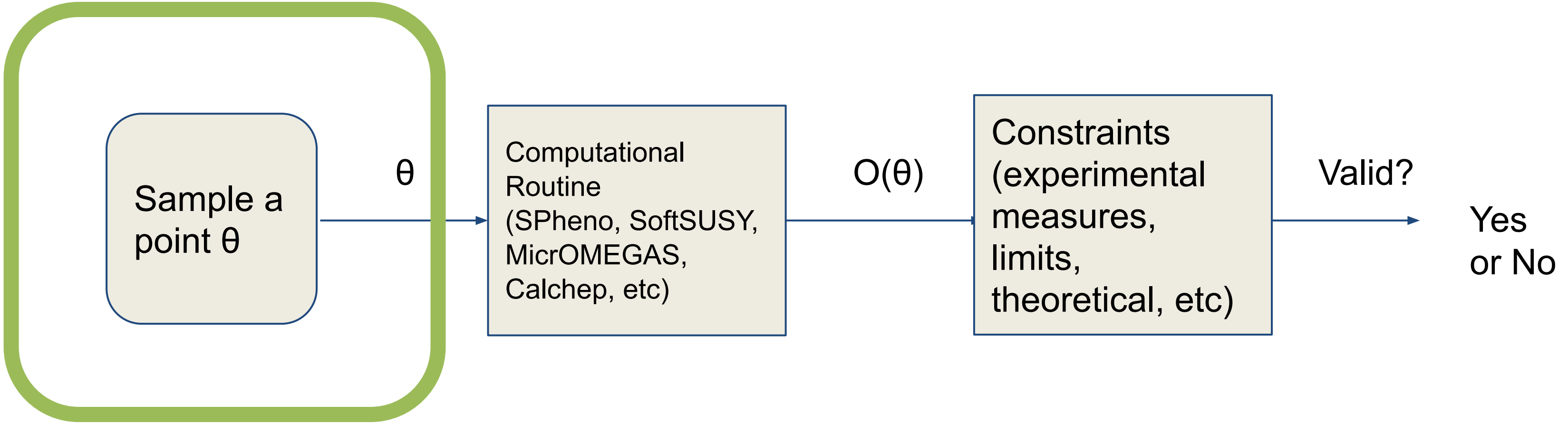
Exploring Parameter Spaces with Artificial Intelligence and Machine Learning Black-Box Optimisation Algorithms
Fernando Abreu de Souza, MCR, Nuno Filipe Castro, Mehraveh Nikjoo, Werner Porod
https://arxiv.org/abs/2206.09223
code: https://gitlab.com/lip_ml/blackboxbsm

Exploring Parameter Spaces with AI/ML

2206.09223

Problem (re)framing: face the sampling

- The “ah-ah” moment of this work was to consider: **what if we change the sampling itself**



Exploring Parameter Spaces with AI/ML

2206.09223

Problem (re)framing: BSM as a black-box

- We do not look at if a point is valid or try to predict observables
- Instead, we look at **how far a point is from being valid**
- Let **C(O)** be a function of a observable **O**

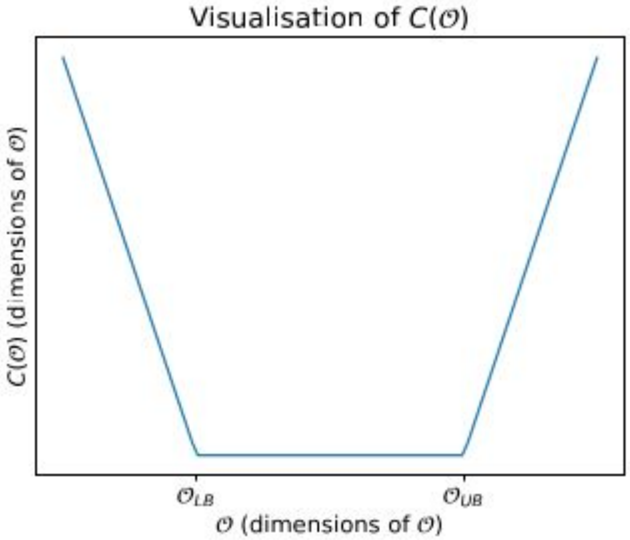
$$C(\mathcal{O}) = max(0, -\mathcal{O} + \mathcal{O}_{LB}, \mathcal{O} - \mathcal{O}_{UB})$$

- The set of valid points

$$\mathcal{V} = \{\theta^* : \theta \in \mathcal{P} \text{ s.t. } C(\theta) = 0\}$$

- Equivalently

$$\mathcal{V} = \{\theta^* : \theta \in \mathcal{P} \text{ s.t. } \theta^* = \text{argmin } C(\theta)\}$$



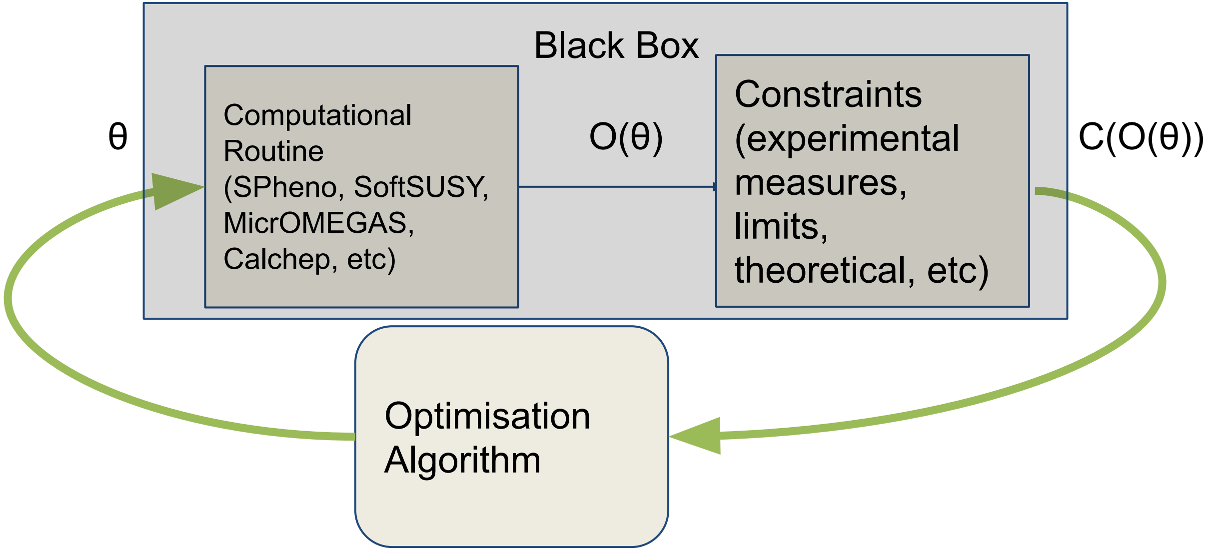
- **=> Finding the valid points is the same as minimising C(O)**

Exploring Parameter Spaces with AI/ML

2206.09223

Problem (re)framing: BSM as a black-box

- Since **O=O(θ)** we can **close the loop** and **optimise in order to the parameters C(O)=C(O(θ)).** From the outside, **C(O(θ))** is a **Black-Box => Black-Box Optimisation Problem**



Exploring Parameter Spaces with AI/ML

2206.09223

The physics models and computational routine

- We used the same physics cases as Hollingsworth, et al [2103.06957]
  - cMSSM: 4 parameters
  - pMSSM: 19 parametrs
- We performed two studies:
  - Higgs mass constraint

$$Loss(\theta) = C(m_{h^0}(\theta))$$

  - Higgs mass and Dark Matter Relic Density constraints by adding them up

$$Loss(\theta) = C(m_{h^0}(\theta)) + C(\Omega_{DM}h^2(\theta))$$

Constraint	$\mathcal{O}_{LB}$	$\mathcal{O}_{UB}$
m_h	122 GeV	128 GeV
$\Omega_{DM}h^2$	0.08	0.14

Ask for backup slides
for free bonus material

Exploring Parameter Spaces with AI/ML

2206.09223

Meet the algorithms

- The fields of Artificial Intelligence and Machine Learning have a multitude of **search algorithms** for **black-box optimisation**
- We explore **three different classes of algorithms** to see their differences
  - A **Bayesian** Optimisation Algorithm: Tree-Parzen Estimator (TPE)
  - A **Genetic** Algorithm: Nondominated Sorting Genetic Algorithm II (NSGA-II)
  - An (non-genetic) **Evolutionary** Algorithm: Covariant Matrix Approximation Evolution Strategy (CMA-ES)

Ask for backup slides
for free bonus material

Exploring Parameter Spaces with AI/ML

2206.09223

Meet the algorithms: Take home aspects

- The algorithms are **sequential**, i.e. a new suggested point depends on the points seen so far
- Two (Bayesian and Evolutionary) have a **learning** component, the other (Genetic) does not
- Different algorithms realise different balances of the **exploration-exploitation trade-off**
- The algorithms only **sort** the points by their **loss**, i.e. they do not care about the actual values of observables, etc
  - => The only important aspect of $C(\theta)$ is to be **monotonic with the distance to boundaries**
- All algorithms **do not require data prior** to the run
  - => They adapt the search dynamically

Ask for backup slides
for free bonus material

Exploring Parameter Spaces with AI/ML

2206.09223

Methodology

- We used optuna python package and its implementations of the algorithms
- Each scan was broken down into **500 episodes** of **2000 sequential trials/points** each (i.e. 1M trials/points per sampler per scan)
- Samplers were compared using the metrics
  - Efficiency (bigger is better)

$$\text{Efficiency} = \frac{\#\ \text{valid trials}}{\#\ \text{total trials}}$$

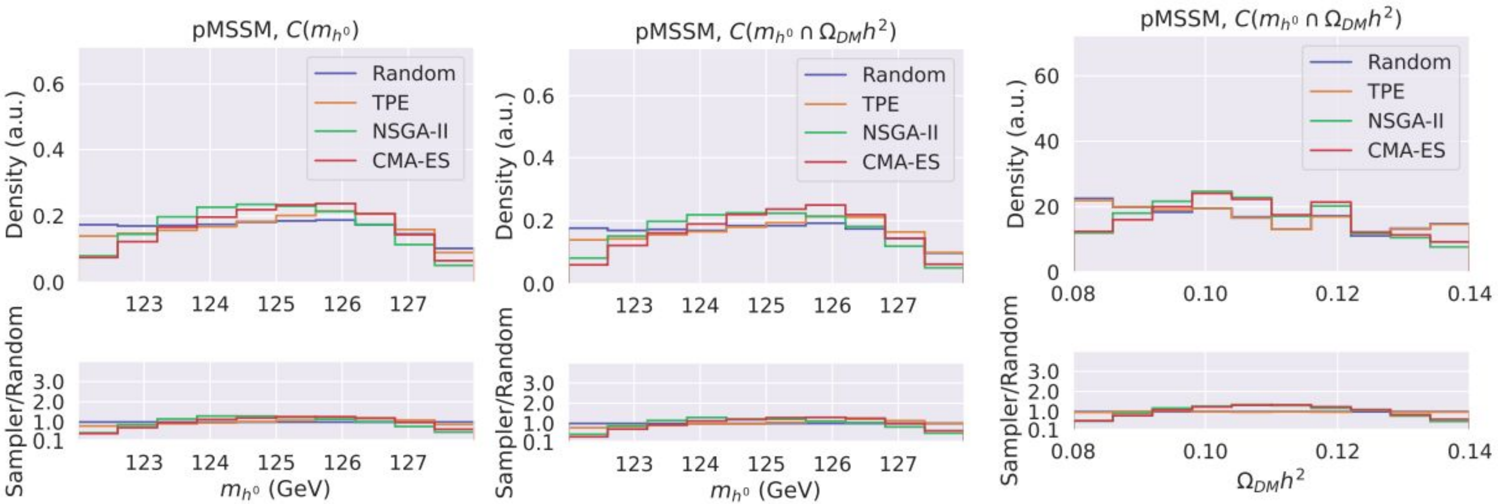
  - Wasserstein Distance vs the Uniform distribution (smaller is better)

$$WD(f,g) = \int_U |F(u) - G(u)|du$$

Exploring Parameter Spaces with AI/ML

Results: pMSSM observables

2206.09223



Ask for backup slides
for free bonus material

Exploring Parameter Spaces with AI/ML

2206.09223

Results: pMSSM parameter distributions

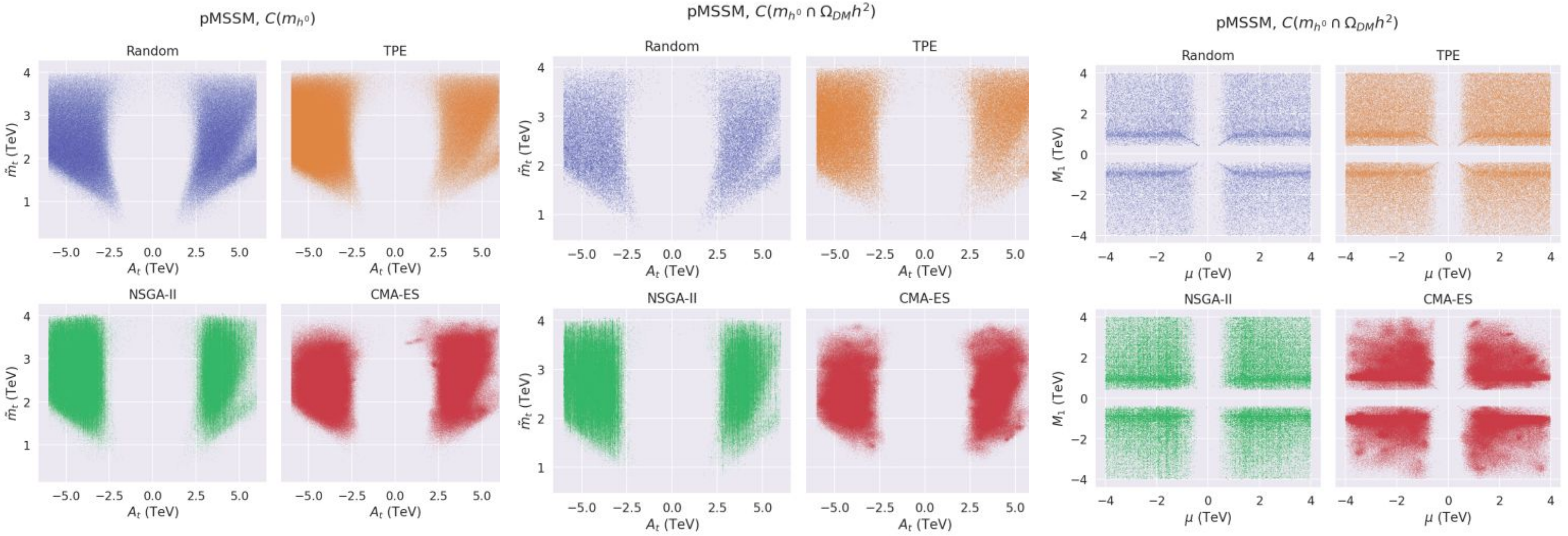
	$\hat{M}_1$	$\hat{M}_2$	$\hat{M}_3$	$\hat{\mu}$	$\hat{A}_t$	$\hat{A}_b$	$\hat{A}_\tau$	$\hat{m}_{L_1}$	$\hat{m}_{e_1}$	$\hat{m}_{L_3}$	$\hat{m}_{e_3}$	$\hat{m}_{Q_1}$	$\hat{m}_{u_1}$	$\hat{m}_{d_1}$	$\hat{m}_{d_1}$	$\hat{m}_{Q_3}$	$\hat{m}_{u_3}$	$\hat{m}_{d_3}$	$\hat{\tan\beta}$
$C(m_{h^0})$, Random	0.01	0.01	0.01	0.05	0.10	0.01	0.01	0.03	0.01	0.01	0.02	0.01	0.02	0.01	0.02	0.05	0.05	0.02	0.03
$C(m_{h^0})$, TPE	0.04	0.04	0.04	0.08	0.30	0.04	0.04	0.07	0.04	0.04	0.05	0.04	0.05	0.04	0.05	0.12	0.12	0.04	0.05
$C(m_{h^0})$, NSGA-II	0.09	0.09	0.09	0.12	0.19	0.09	0.09	0.10	0.09	0.09	0.10	0.10	0.10	0.09	0.09	0.15	0.14	0.09	0.12
$C(m_{h^0})$, CMA-ES	0.20	0.20	0.20	0.20	0.33	0.20	0.21	0.20	0.21	0.21	0.20	0.20	0.20	0.20	0.21	0.21	0.21	0.21	0.20
$C(m_{h^0} \cap \Omega_{DM}h^2)$, Random	0.04	0.04	0.04	0.06	0.10	0.04	0.04	0.04	0.05	0.06	0.04	0.04	0.04	0.04	0.04	0.08	0.07	0.04	0.05
$C(m_{h^0} \cap \Omega_{DM}h^2)$, TPE	0.16	0.11	0.09	0.09	0.34	0.09	0.09	0.10	0.15	0.17	0.10	0.11	0.09	0.09	0.09	0.16	0.16	0.09	0.08
$C(m_{h^0} \cap \Omega_{DM}h^2)$, NSGA-II	0.16	0.13	0.11	0.15	0.21	0.11	0.11	0.12	0.19	0.22	0.16	0.17	0.12	0.12	0.12	0.18	0.17	0.12	0.14
$C(m_{h^0} \cap \Omega_{DM}h^2)$, CMA-ES	0.30	0.28	0.25	0.25	0.35	0.25	0.25	0.24	0.27	0.25	0.27	0.27	0.24	0.25	0.24	0.25	0.25	0.25	0.24

Ask for backup slides
for free bonus material

Exploring Parameter Spaces with AI/ML

2206.09223

Results: pMSSM parameter scatters



Ask for backup slides
for free bonus material

Exploring Parameter Spaces with AI/ML

2206.09223

Results: Performance metrics

- Efficiency (bigger is better)

Model	Constraint	Sampler			
		Random	TPE	NSGA-II	CMA-ES
cMSSM	m_h	0.401 ± 0.010	0.668 ± 0.012	0.715 ± 0.014	$\mathbf{0.924 \pm 0.023}$
	$m_{h_0} \cap \Omega_{DM}h^2$	0.006 ± 0.001	0.127 ± 0.008	0.281 ± 0.041	$\mathbf{0.687 \pm 0.084}$
pMSSM	m_h	0.309 ± 0.010	0.557 ± 0.038	0.862 ± 0.015	$\mathbf{0.899 \pm 0.034}$
	$m_h \cap \Omega_{DM}h^2$	0.038 ± 0.004	0.099 ± 0.013	$\mathbf{0.663 \pm 0.036}$	0.576 ± 0.073

Ask for backup slides
for free bonus material

Exploring Parameter Spaces with AI/ML

2206.09223

Results: Performance metrics

- Wasserstein Distance (smaller is better)

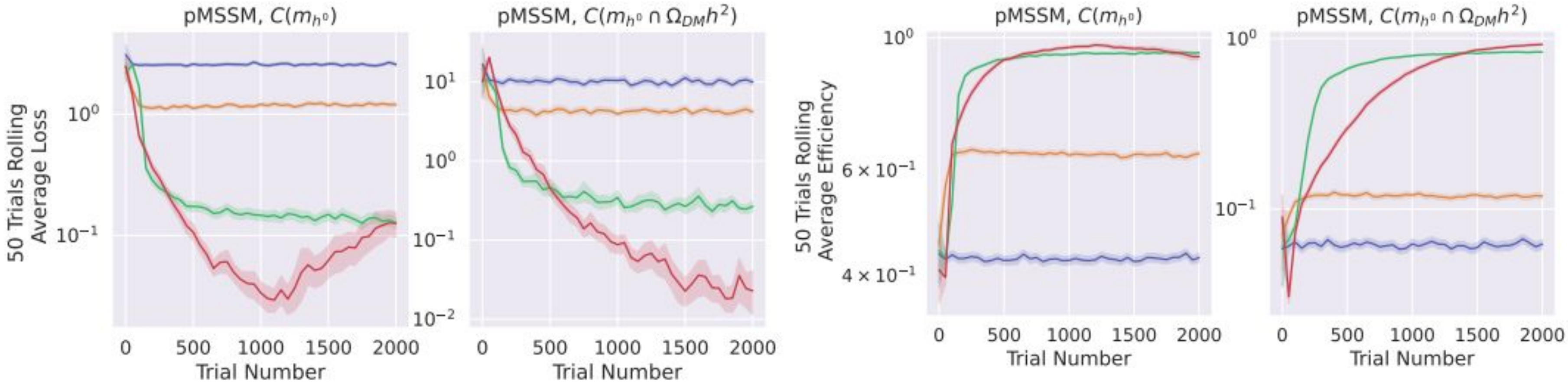
Model	Constraint	Sampler: Random	Sampler: TPE	Sampler: NSGA-II	Sampler: CMA-ES
cMSSM	m_{h_0}	0.229 ± 0.014	$\mathbf{0.276 \pm 0.020}$	0.364 ± 0.039	0.627 ± 0.099
	$m_{h_0} \cap \Omega_{DM}h^2$	0.797 ± 0.113	$\mathbf{1.101 \pm 0.062}$	1.150 ± 0.090	1.186 ± 0.117
pMSSM	m_{h_0}	0.495 ± 0.030	$\mathbf{1.270 \pm 0.137}$	2.028 ± 0.188	3.997 ± 0.449
	$m_{h_0} \cap \Omega_{DM}h^2$	0.939 ± 0.079	$\mathbf{2.369 \pm 0.274}$	2.800 ± 0.278	4.932 ± 0.331

Ask for backup slides
for free bonus material

Exploring Parameter Spaces with AI/ML

2206.09223

Results: Performance metrics



Ask for backup slides
for free bonus material

Conclusions

Conclusions
And future work

- We presented a new approach to parameter space scanning: **black-box optimisation**
- The dynamic samplers introduced produced **up to 2 orders of magnitude increased sampling efficiency without the need of a prior training dataset**
- Different samplers have different **exploration-exploitation trade-off**
    - Bayesian (TPE) closer to the random sampler coverage
    - Evolutionary (CMA-ES) achieves almost optimal efficiency
    - Genetic (NSGA-II) somewhere in between
- Samplers with learning components **suffer from the curse of dimensionality**

Conclusions
And future work

- The samplers used were **not tailor-made** for this task
    - => Develop custom samplers that can maximise coverage and efficiency
- Samplers with classical/shallow learning components suffer from **curse of dimensionality**
    - => Develop samplers with Kernel or Neural Net components
- We summed **two constraints into a single one**
    - => Explore multi-objective optimisation algorithms
- The physics cases at hand were not particularly difficult and **sampling efficiency saturated** quickly
    - => Go beyond proof-of-concept and apply methodology to more difficult and realistic scans

Thank you! mcromao@lip.pt

BONUS: A conversation with Prof. Aisusy (gpt-3)

Human: Tell me Prof., do you still believe in the realisation of supersymmetry given the lack of experimental evidence?

Prof. Aisusy: There is still a great deal of theoretical evidence supporting the existence of supersymmetry, and I remain optimistic that we will eventually find experimental evidence for it. The implications of supersymmetry are very significant, and finding it would be a major breakthrough in our understanding of the universe.

Human: What if we can't find supersymmetry at the Large Hadron Collider?

Prof. Aisusy: The Large Hadron Collider is the most powerful machine we have for finding supersymmetry, but it is not the only possible tool. If supersymmetry exists, we will eventually find it through one method or another. There are also many other mysteries of our universe that we may be able to solve without supersymmetry, so even if we don't find it, we can still make significant progress in our understanding of the cosmos.

n+1
Backups

Exploring Parameter Spaces with AI/ML

2206.09223

The physics models and computational routine

$$W_{MSSM} = -\varepsilon_{ab}\mu \hat{H}_1^a \hat{H}_2^b + \varepsilon_{ab}\left(\hat{H}_1^a \hat{L}^b Y_e \hat{E}^c + \hat{H}_1^a \hat{Q}^b Y_d \hat{D}^c + \hat{H}_2^b \hat{Q}^a Y_u \hat{U}^c\right)$$

$$\begin{aligned}
\mathcal{L}_{\text{soft}}^{\text{MSSM}} &= -\frac{1}{2}\left(M_1 \widetilde{B}\widetilde{B} + M_2 \widetilde{W}\widetilde{W} + M_3 \widetilde{g}\widetilde{g} + \text{h.c.}\right) \\
&- m_Q^2 \widetilde{Q}^\dagger \widetilde{Q} - m_L^2 \widetilde{L}^\dagger \widetilde{L} - m_u^2 \widetilde{U}^* \widetilde{U} - m_d^2 \widetilde{D}^* \widetilde{D} - m_e^2 \widetilde{E}^* \widetilde{E} \\
&- \left(T_U \widetilde{U}^* H_u \widetilde{Q} + T_D \widetilde{D}^* H_d \widetilde{Q} + T_E \widetilde{E}^* H_d \widetilde{L} \text{h.c.}\right) \\
&- m_{H_u}^2 H_u^* H_u - m_{H_d}^2 H_d^* H_d - \left(b H_u H_d + \text{h.c.}\right).
\end{aligned}$$

Exploring Parameter Spaces with AI/ML

2206.09223

The physics models and computational routine

Parameter	Values	Description
m_0	$[0, 10]$ TeV	Soft Scalar Mass
$m_{1/2}$	$[0, 10]$ TeV	Soft Fermion Mass
A_0	$[-6m_0, 6m_0]$	Trilinear Soft Coupling
$\tan\beta$	$[1.5, 50]$	Tan Beta

Parameter	Values	Description
$\lvert M_1\rvert$	$[0.05, 4]$ TeV	Gaugino (Bino) mass
$\lvert M_2\rvert$	$[0.4, 4]$ TeV	Gaugino (Wino) mass
M_3	$[1, 4]$ TeV	Gaugino (gluino) mass
$\lvert\mu\rvert$	$[0.4, 4]$ TeV	Bilinear Higgs mass
$\lvert A_t\rvert$	$[0, 6]$ TeV	Top trilinear coupling
$\lvert A_b\rvert$	$[0, 4]$ TeV	Bottom trilinear coupling
$\lvert A_\tau\rvert$	$[0, 4]$ TeV	Tau trilinear coupling
m_A	$[0.1, 4]$ TeV	Pseudoscalar Higgs mass
$\tan\beta$	$[1, 60]$	
m_{L_1}	$[0.1, 4]$ TeV	1st gen. l.h. slepton mass
m_{e_1}	$[0.1, 4]$ TeV	1st gen. r.h. slepton mass
m_{L_2}	m_{L_1}	2nd gen. l.h. slepton mass
m_{e_2}	m_{e_1}	2nd gen. lr.h. slepton mass
m_{L_3}	$[0.1, 4]$ TeV	3rd gen. l.h. slepton mass
m_{e_3}	$[0.1, 4]$ TeV	3rd gen. r.h. slepton mass
m_{Q_1}	$[0.7, 4]$ TeV	1st gen. l.h. squark mass
m_{u_1}	$[0.7, 4]$ TeV	1st gen. r.h. u-type mass
m_{d_1}	$[0.7, 4]$ TeV	1st gen. r.h. d-type mass
m_{Q_2}	m_{Q_1}	2nd gen. l.h. squark mass
m_{u_2}	m_{u_1}	2nd gen. r.h. u-type mass
m_{d_2}	m_{d_1}	2nd gen. r.h. d-type mass
m_{Q_3}	$[0.7, 4]$ TeV	3rd gen. l.h. squark mass
m_{u_3}	$[0.7, 4]$ TeV	3rd gen. r.h. u-type mass
m_{d_3}	$[0.7, 4]$ TeV	3rd gen. r.h. d-type mass

Exploring Parameter Spaces with AI/ML

2206.09223

The physics models and computational routine

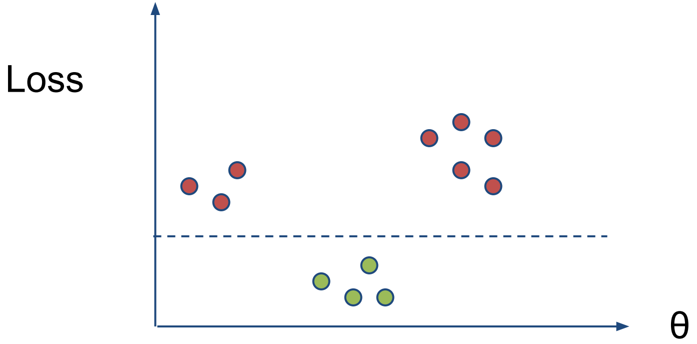
- The physical spectrum, i.e. the Higgs mass, was computed using **Spheno**
  - Points with invalid physical spectrum (e.g. charged tachyons) were given the value C=+∞ (“unphysical”)
- The dark matter relic density was computed using **MicrOMEGAS**
  - Points for which MicrOMEGAS did not produce a relic density (e.g. charged LSP) were given the value C=+∞ (“unphysical”)

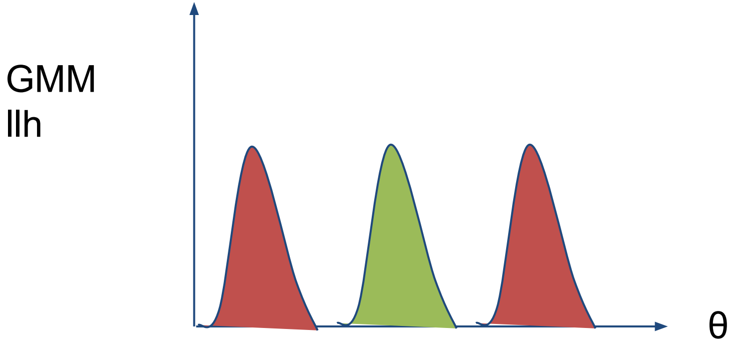
Exploring Parameter Spaces with AI/ML

2206.09223

Meet the algorithms: TPE, the Bayesian

- TPE **sorts the parameter points** by their **loss** value (i.e. **C**)
- It **splits** points between **good** and **bad** through a moving quantile heuristic
- **Fits** a Gaussian Mixture Model (learnable component of the algorithm) on each **good** and **bad** set
- **Samples** a point from the **good**, and **keeps** it if its **likelihood is greater** than being **bad**





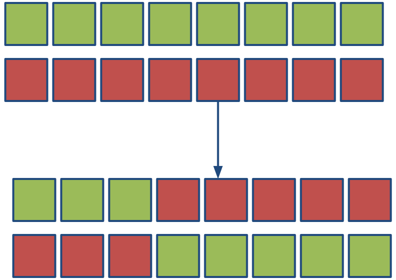
Exploring Parameter Spaces with AI/ML

2206.09223

Meet the algorithms: NSGAI-II, the Genetic

- Encode parameter space point (a vector) as genes

- Prepare a population (say 50)
- Evaluate their **fitness** (i.e. the loss)
- **Sort** them by their fitness
- **Keep** the best (elitism) discard the rest
- Create **offspring** from the elite (**cross-over**)
- Apply random **mutations**
- Repeat



Exploring Parameter Spaces with AI/ML

2206.09223

Meet the algorithms: CMA-ES, the Evolutionary

- Initialise a **multivariate normal** with random mean and identity covariance matrix
- Sample a population
- **Evaluate** them and **sort** by loss
- Use the **best** and compute their statistics
  - Mean
  - Covariance
- **Update** mean and covariance matrix with weighted rolling updates
- Repeat

Exploring Parameter Spaces with AI/ML

2206.09223

Methodology

- Mean euclidean pairwise distance between valid points (bigger is better)
- Intuitive, but produces counter intuitive values for some cases

$$\text{Mean Euclidean Distance} = \mathbb{E}_{\hat{\theta}_i, \hat{\theta}_j \in \mathcal{V}} \left[\sqrt{(\hat{\theta}_i - \hat{\theta}_j)^2} \right]$$

Exploring Parameter Spaces with AI/ML

2206.09223

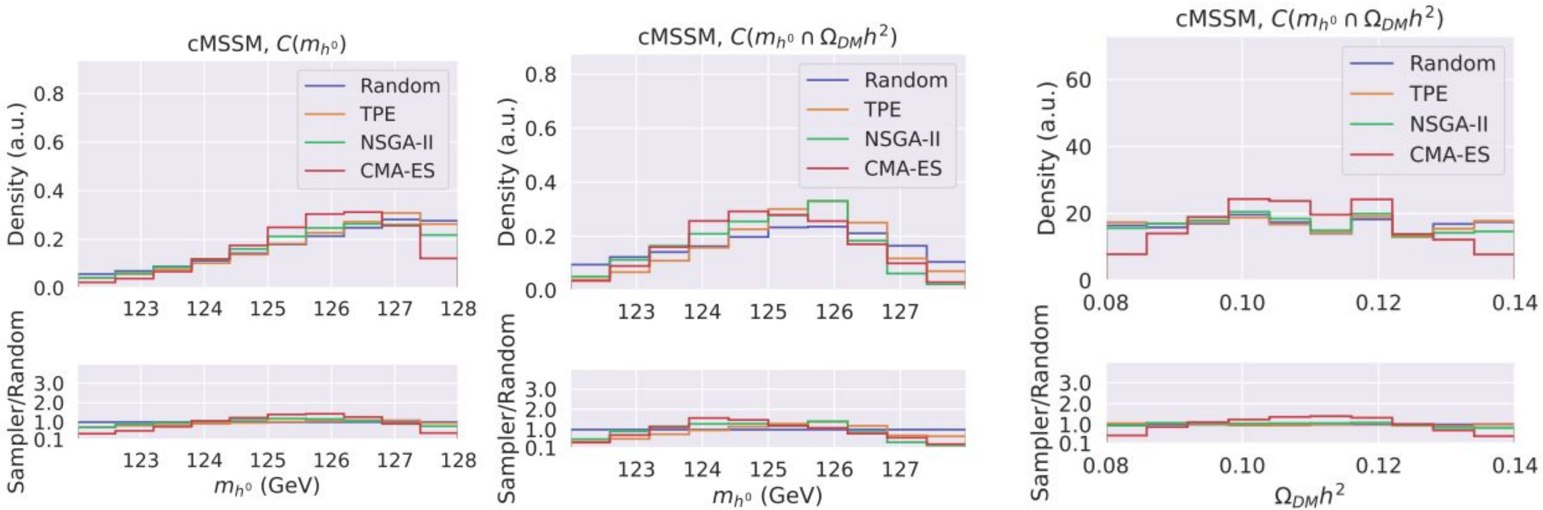
Results: cMSSM parameter distributions

	$\hat{m}_0$	$\hat{m}_{1/2}$	$\hat{A}_0$	$\hat{\tan\beta}$
$C(m_{h^0})$, Random	0.04	0.04	0.12	0.03
$C(m_{h^0})$, TPE	0.06	0.05	0.14	0.03
$C(m_{h^0})$, NSGA-II	0.06	0.09	0.16	0.05
$C(m_{h^0})$, CMA-ES	0.14	0.15	0.19	0.15
$C(m_{h^0} \cap \Omega_{DM}h^2)$, Random	0.27	0.21	0.09	0.22
$C(m_{h^0} \cap \Omega_{DM}h^2)$, TPE	0.45	0.22	0.10	0.33
$C(m_{h^0} \cap \Omega_{DM}h^2)$, NSGA-II	0.45	0.27	0.11	0.33
$C(m_{h^0} \cap \Omega_{DM}h^2)$, CMA-ES	0.42	0.26	0.20	0.30

Exploring Parameter Spaces with AI/ML

2206.09223

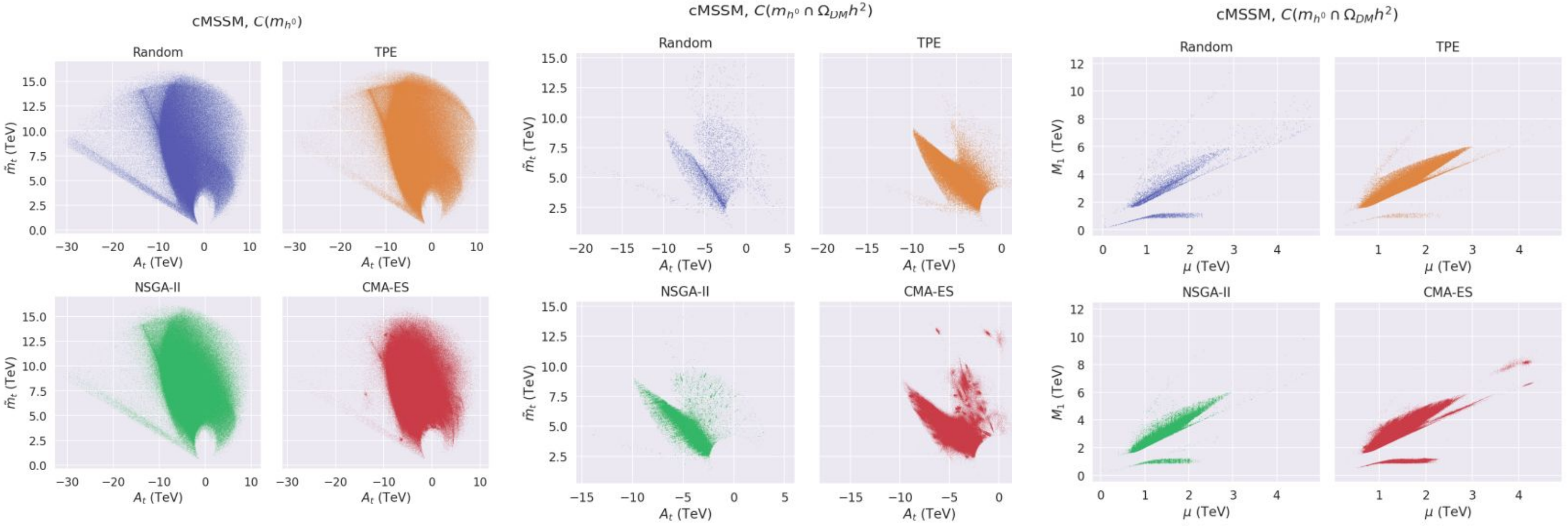
Results: cMSSM observables



Exploring Parameter Spaces with AI/ML

Results: cMSSM parameter scatters

2206.09223



Exploring Parameter Spaces with AI/ML

2206.09223

Results: Performance metrics

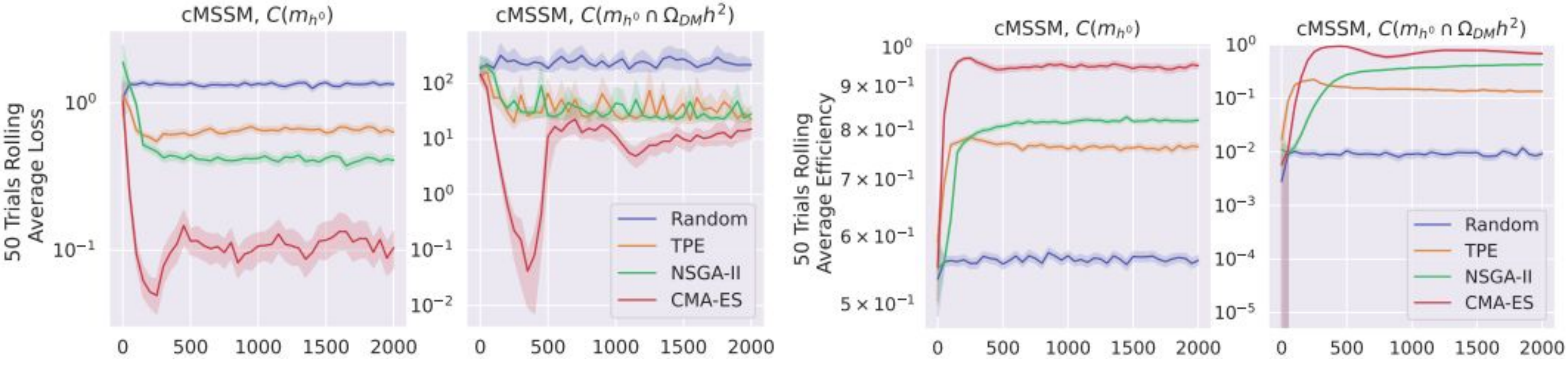
- Mean Euclidean Distance (Bigger is better)

Model	Constraint	Sampler			
		Random	TPE	NSGA-II	CMA-ES
cMSSM	m_{h_0}	0.686 ± 0.006	$\mathbf{0.706 \pm 0.009}$	0.619 ± 0.017	0.401 ± 0.058
	$m_{h_0} \cap \Omega_{DM}h^2$	0.625 ± 0.075	$\mathbf{0.376 \pm 0.032}$	0.321 ± 0.066	0.223 ± 0.097
pMSSM	m_{h_0}	1.745 ± 0.006	$\mathbf{1.659 \pm 0.021}$	1.594 ± 0.048	0.750 ± 0.128
	$m_{h_0} \cap \Omega_{DM}h^2$	1.758 ± 0.021	$\mathbf{1.523 \pm 0.041}$	1.500 ± 0.069	0.437 ± 0.097

Exploring Parameter Spaces with AI/ML

Results: Performance metrics

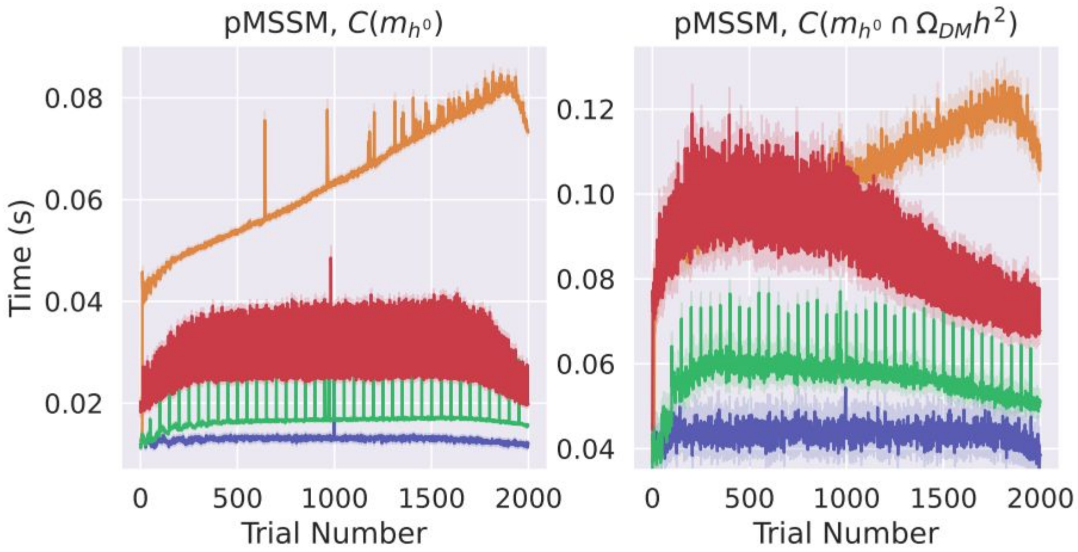
2206.09223



Exploring Parameter Spaces with AI/ML

2206.09223

Results: Performance metrics



Exploring Parameter Spaces with AI/ML

2206.09223

Results: Performance metrics