Imperial College London

SCHOOL OF DATA ANALYSIS

Local Generative Surrogate Optimisation and Applications to HEP

November 2020

Sergey Shirobokov
Supervisor: Andrey Golutvin

In NeurIPS 2020:
Differentiating the Black-Box: Optimization with Local Generative Surrogates
S.Shirobokov*, V.Belavin*, M.Kagan, A. Ustyuzhanin, A.G. Baydin

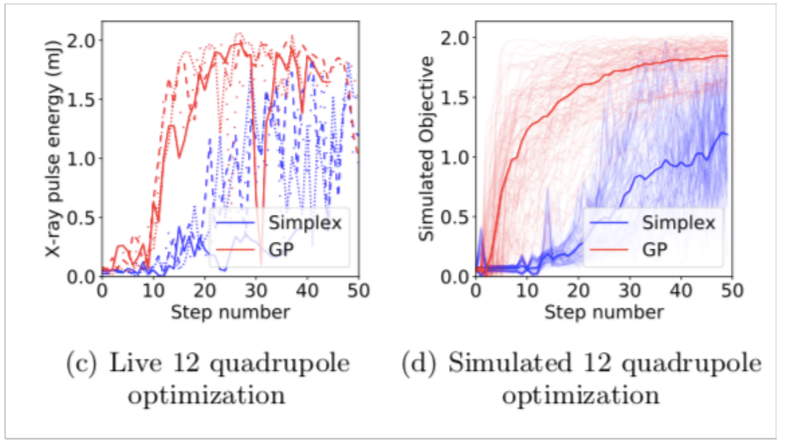
About me

- 4th year PhD student at our Imperial HEP group☺
- Working on the optimisation of the SHiP experiment
- MSc in Computer Science back in Russia
- ML research intern @ Amazon this summer. Working on uncertainty quantification for deep learning
  + recommender systems

- Interests: ML for HEP, optimisation of the simulators, uncertainty in ML
- Reach out if you want to discuss my industry vs academia experience or give some tips how to write a thesis☺

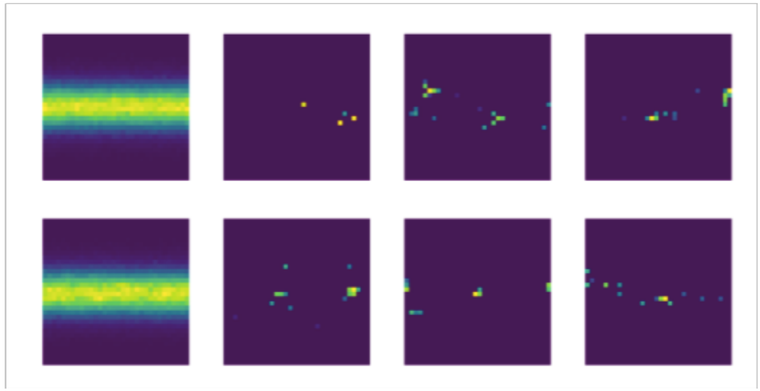
Simulators optimisation

- Traffic scenes generation
- Control of the accelerator
- Lasers optimisation
- High Energy Physics



(c) Live 12 quadrupole optimization

(d) Simulated 12 quadrupole optimization

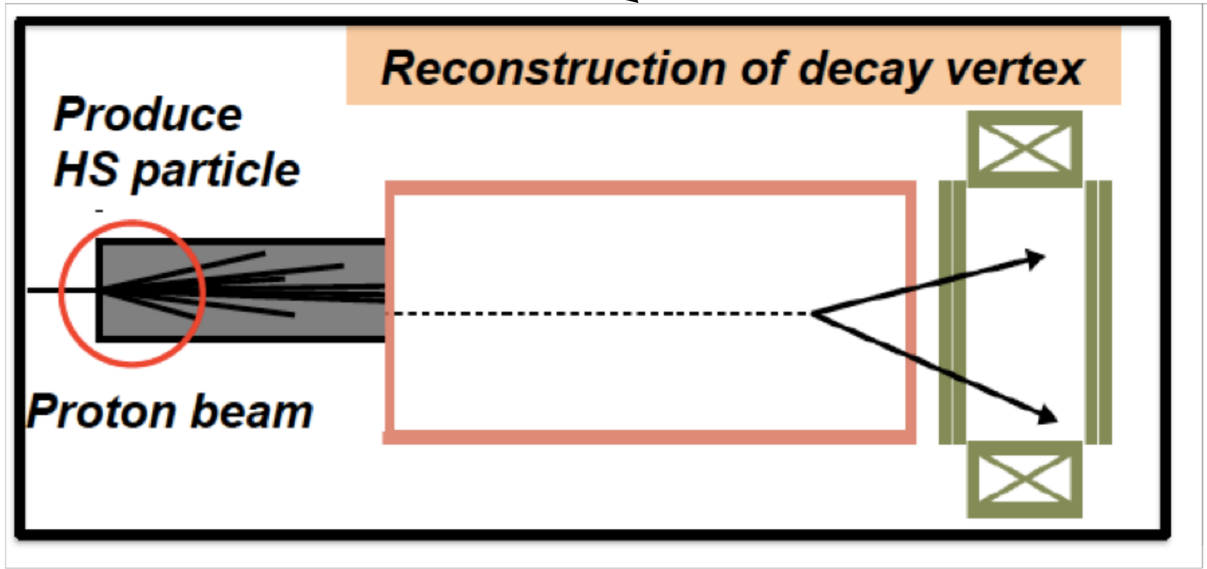


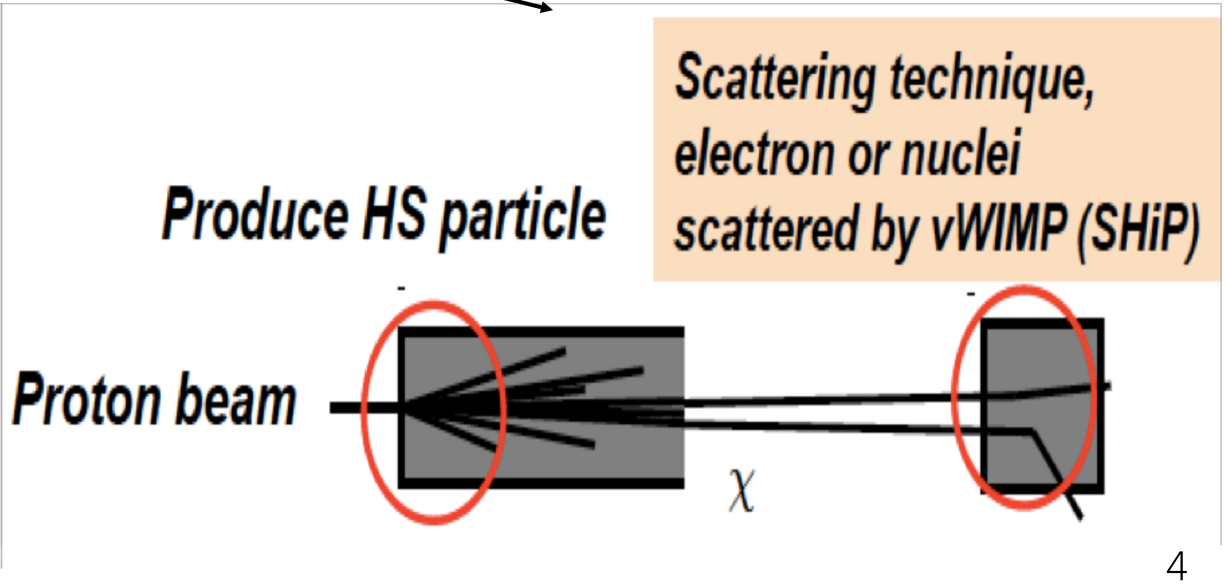
Examples: 1810.02513, 1610.06151, 1909.05963, 1707.07113

Motivation. SHiP experiment

- Beam dump experiment at CERN
- 400 GeV/c proton beam
- Small couplings, hence long-lived

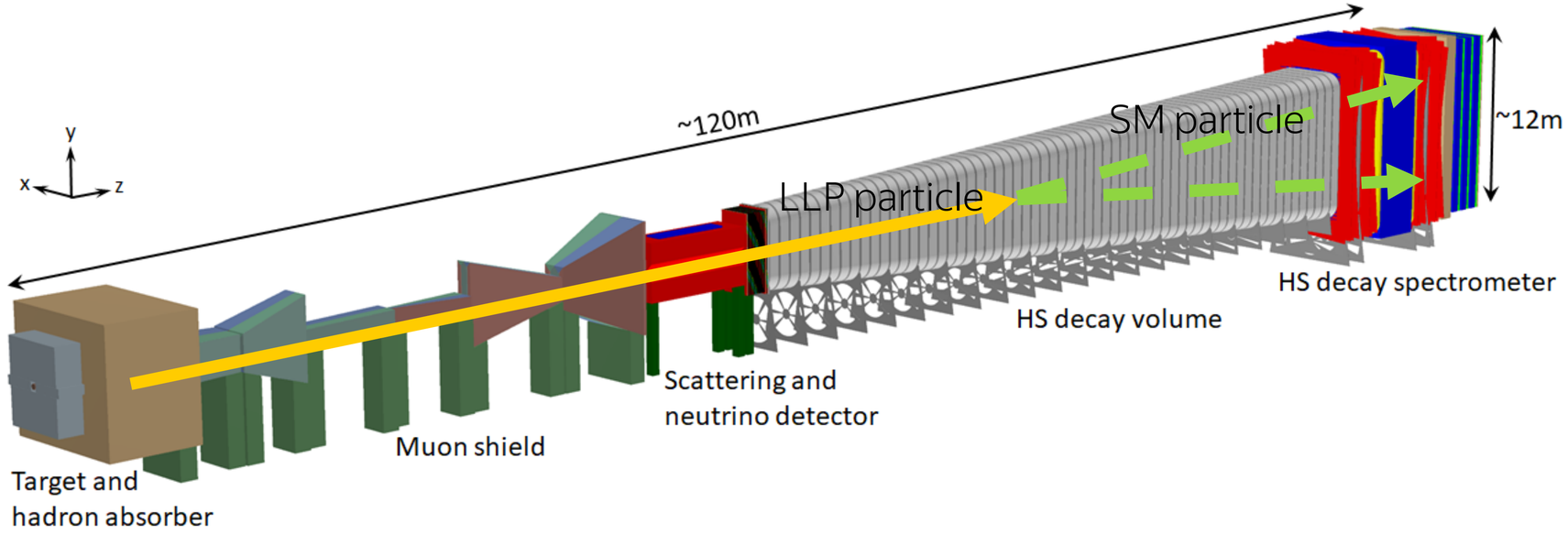
Two strategies for search





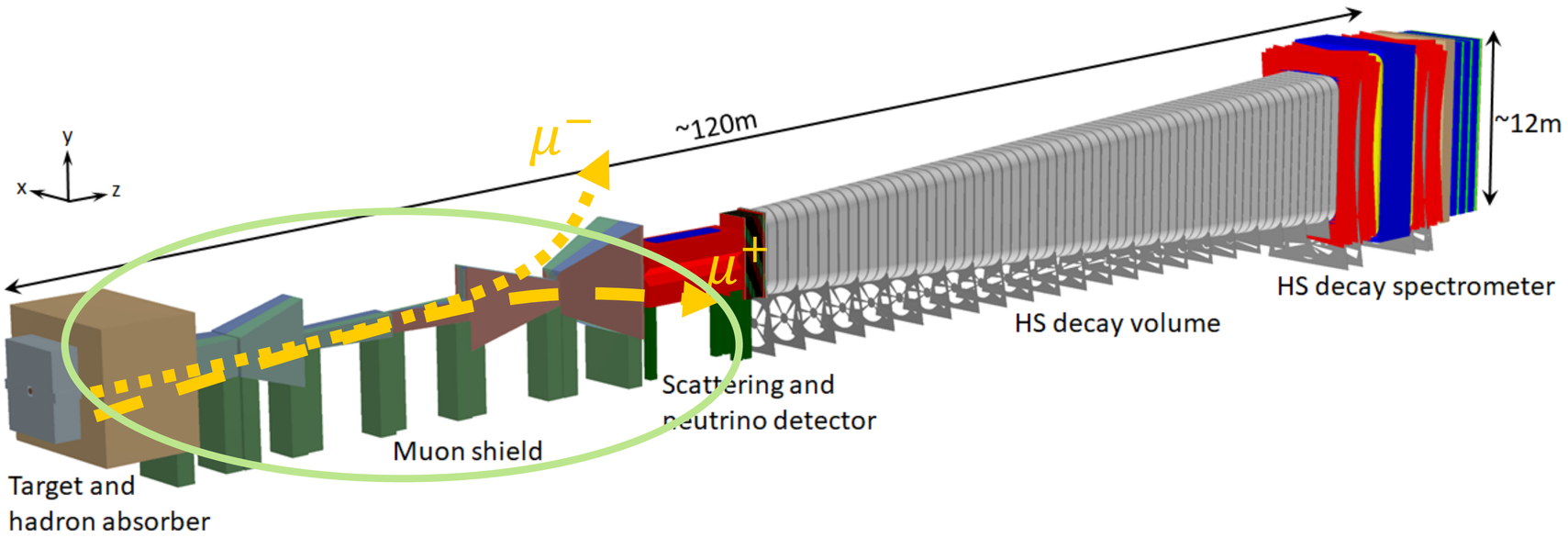
SHiP experiment

SHiP Technical proposal
1504.04956



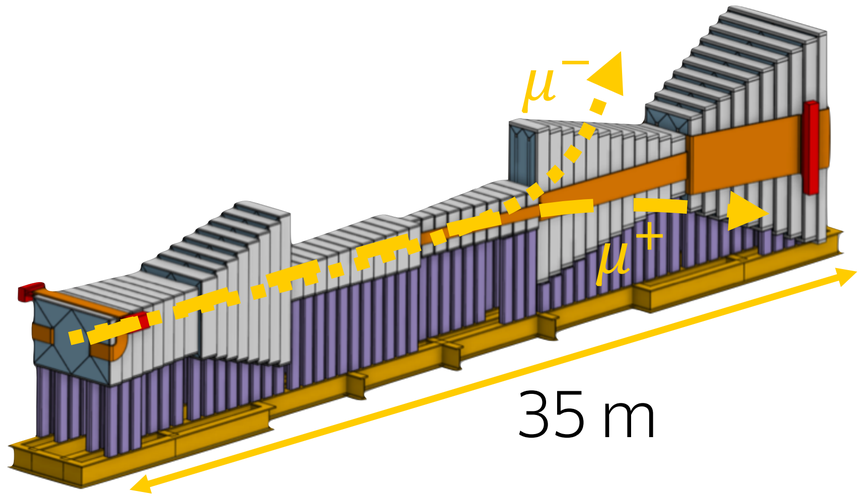
- Can detect very weakly coupled long-lived particles via decay or DM via scattering
- Planned zero background experiment

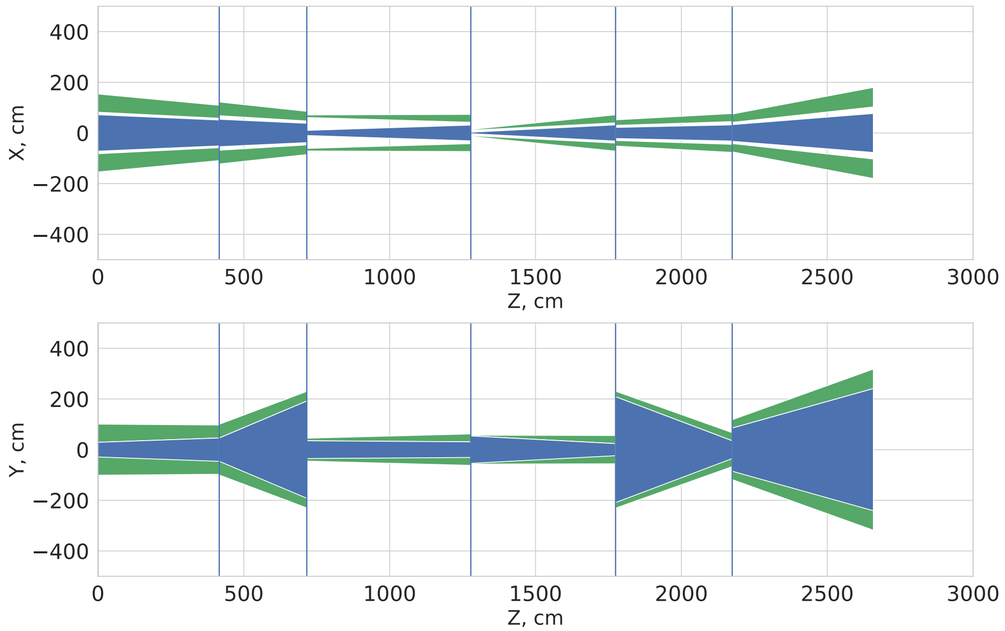
SHiP background



- 10^{11} muons/spill is produced inside the target
- Reduce background rate by six orders of magnitude
- Optimise the shield for the best physics performance and cost

SHiP muon shield

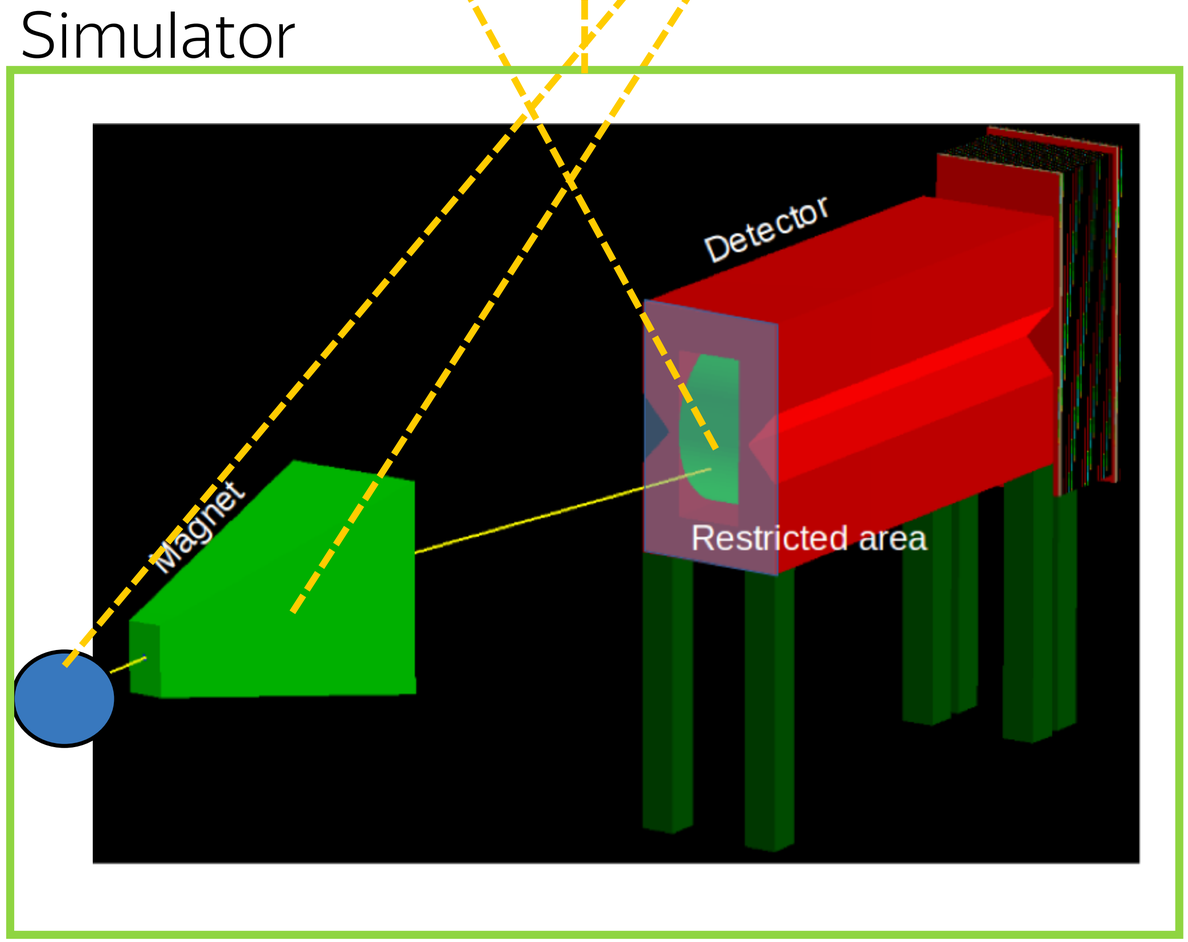




- Optimise parameters to reduce muon acceptance in the detector
- GEANT4 simulates muons propagation through the shield
- Shield is characterised by 42 parameters
- *Very slow to simulate data points → hard to generate large samples*

Problem statement in a nutshell

$$\text{argmin}_{\psi}\ E[R(y)]$$
$$y = F(x, \psi)$$



- ψ – geometry of the shield: input to the simulator
- x – muon kinematics: input to the simulator
- F – is the GEANT4 simulator
- y – observations: output of the simulator
- $R(y)$ - objective function

Stochastic black-box simulator

- F – random variable:

$$y = F(x, \psi, z) \leftrightarrow y \sim p(y|x; \psi)$$

- F – black-box

p is not known

Can only sample from p

- How to optimise?

$\nabla_\psi p(y|x; \psi)$ can not be computed

Stochastic black-box optimisation

$$\operatorname{argmin}_\psi E_y[R(y)] = \operatorname{argmin}_\psi E[R(F(x,\psi))]$$
$$= \operatorname{argmin}_\psi \int R(y)\, p(y|x;\psi) q(x) dx\, dy$$

Intractable

$$\nabla_\psi E_y[R(y)] = ?$$

How can the optimisation be performed in such case?

Existing black-box optimisation methods

Gradient based:

- Numerical differentiation
- Score function estimation (REINFORCE)

Alternatives:

- Bayesian optimisation
- Evolutionary strategies

Proposed by us:

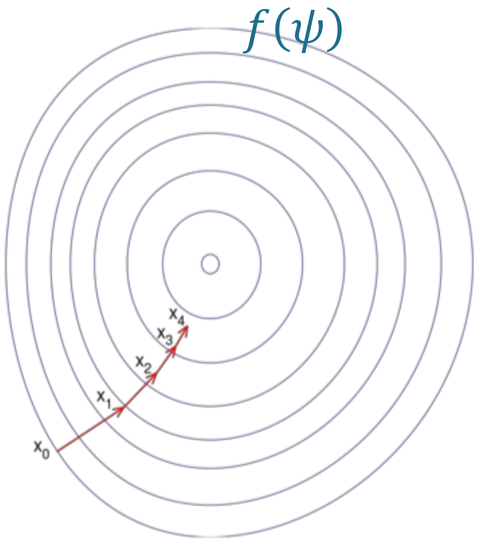
- Local Generative Surrogate optimisation

Gradient descent

- Goal: $\mathrm{Argmin}_{\psi}\, f(\psi)$.
- Solution: $\nabla_{\psi} f(\psi) = 0, \quad \nabla^2_{\psi} f(\psi) = 0$
- Method: gradient descent

$$\psi_{t+1} = \psi_t - const * \nabla_{\psi} f(\psi)$$

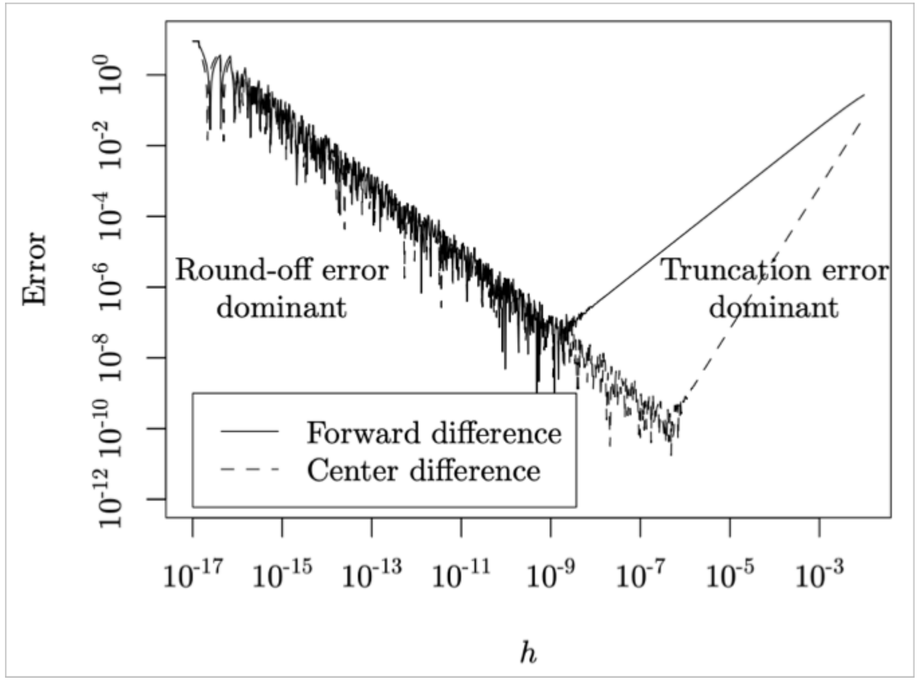
- Problem: $\nabla_{\psi} f(\psi)$ can **NOT** be computed



How to estimate $\nabla_{\psi} f(\psi)$?

Numerical differentiation

$$\nabla_{\psi} f(\psi) \approx \frac{f(\psi+h) - f(\psi)}{h}$$, h – step size



- May have numerical instabilities
- Require $O(d)$ evaluation of f. $\psi \in \mathbb{R}^d$
- Can be challenging to apply with stochastic functions
- Perform linear interpolation

Score function estimator

- Trick:

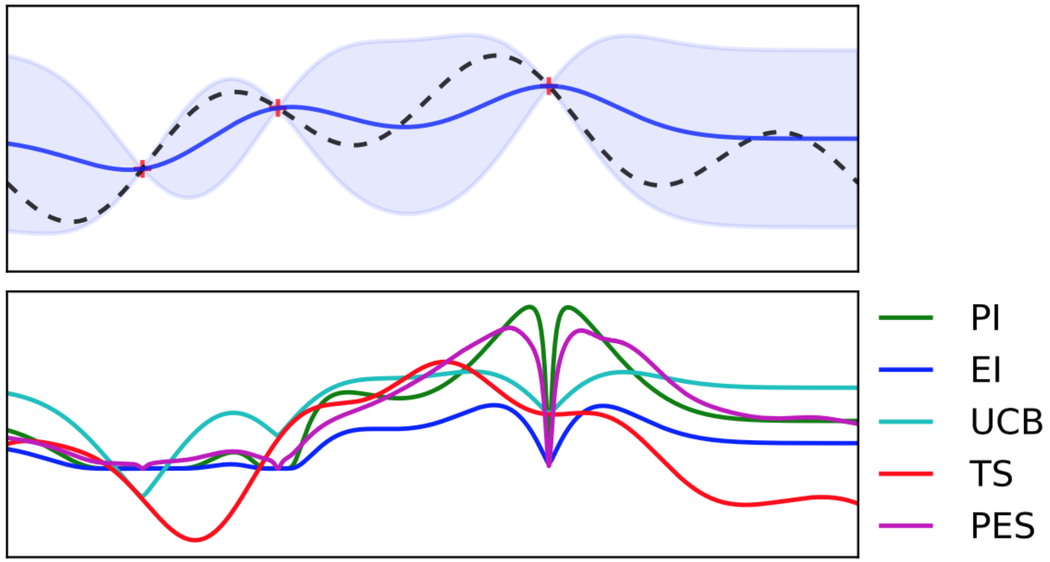
$$\nabla_{\psi} \mathrm{E}_{\mathrm{p}}[f(\psi)] = E_p[f(\psi) \nabla_{\psi} \log(\mathrm{p})]$$

- Have high variance[1]
- Require prior distribution over ψ
- Techniques developed to reduce variance[2,3]

But: **Fast to compute**

[1]https://doi.org/10.1007/BF00992696. [2]1711.00123, [3]1810.02513

Bayesian optimisation with Gaussian Processes



- Goal: $\mathbf{argmin}_{\psi} f(\psi)$
- Approximate $f(\psi)$ with GP → $\mu(\psi)$ and $\sigma(\psi)$
- Chose acquisition function $\alpha(\psi)$:
  Set exploration/exploitation of the space
- Maximise $\alpha(\psi)$
  Example: $\alpha(\psi) = -\mu(\psi) + \eta\,\sigma(\psi)$

- Benefits:

  - Can potentially find global minima
  - Work with non-differentiable functions

- Drawbacks:

  - Scales as $O(n^3 + n^2 d), n$ – size of the traning set
  - Suffer from curse of dimensionality

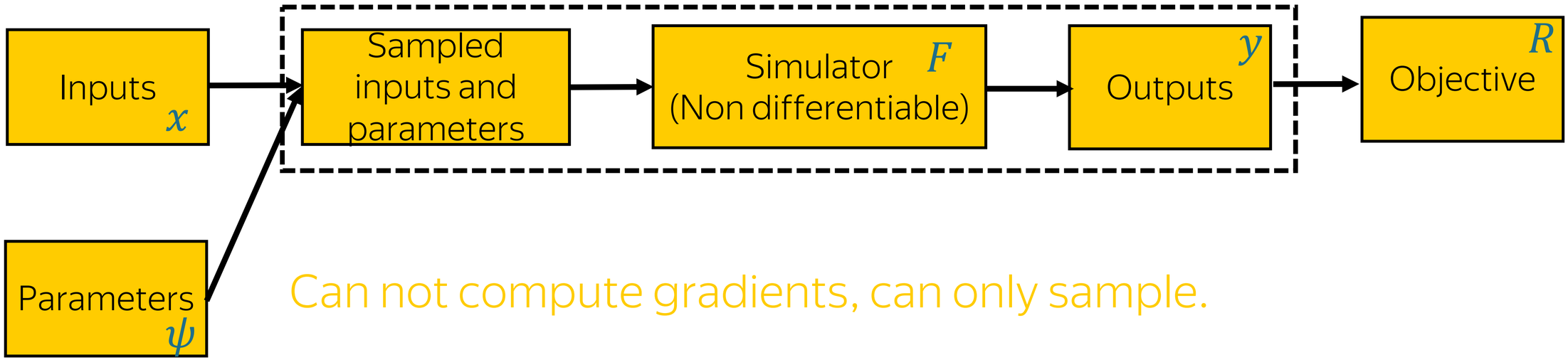
Why to create something new?

- Require frequent simulator calls → computationally expensive
- Require prior distribution or search region
- May not scale well to high dimensions
- Have high variance
- Estimate only first order gradients

We try to solve some of those issues with our method.

Generative surrogates

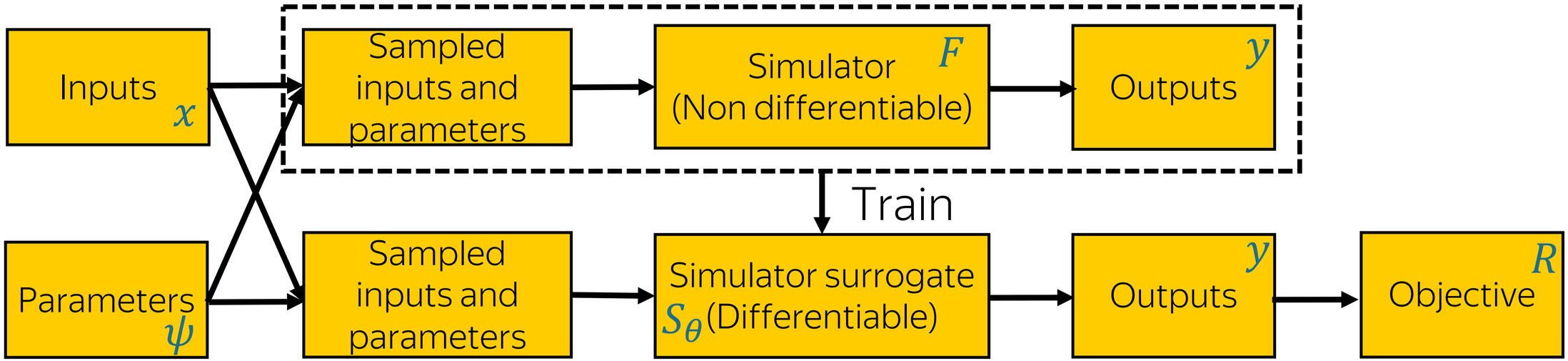
$$E_y[R(y_\psi)] \approx \frac{1}{N}\sum_{i=1}^{N} R(F(x_i;\psi_i))$$



Can not compute gradients, can only sample.

Generative surrogates

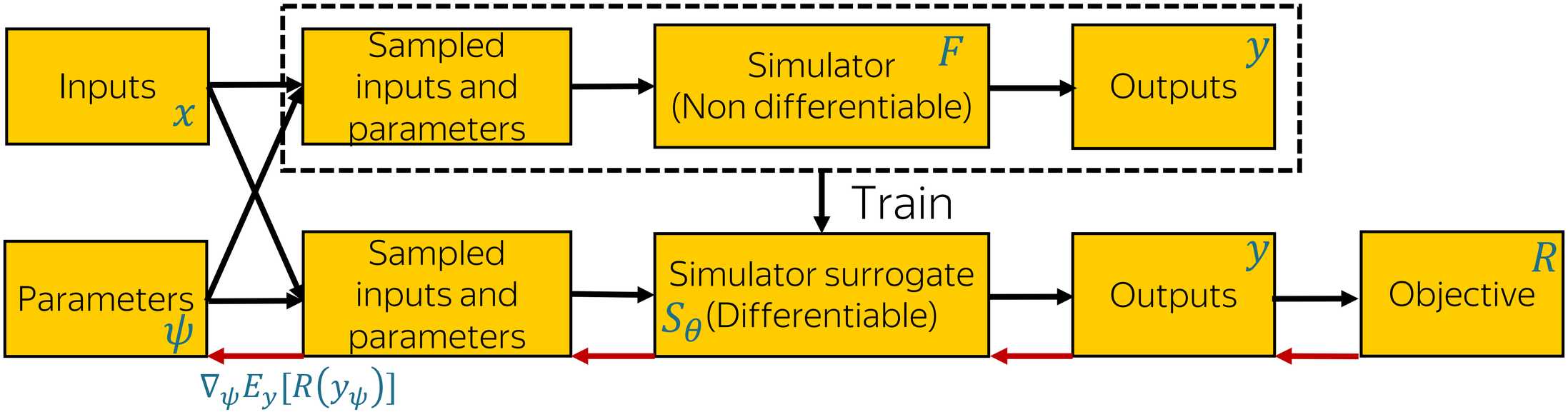
$$E_y[R(y_\psi)] \approx \frac{1}{N}\sum_{i=1}^{N} R(F(x_i;\psi_i))$$



$$E_y[R(y_\psi)] \approx \frac{1}{N}\sum_{i=1}^{N} R(S(z_i, x_i;\psi_i))$$

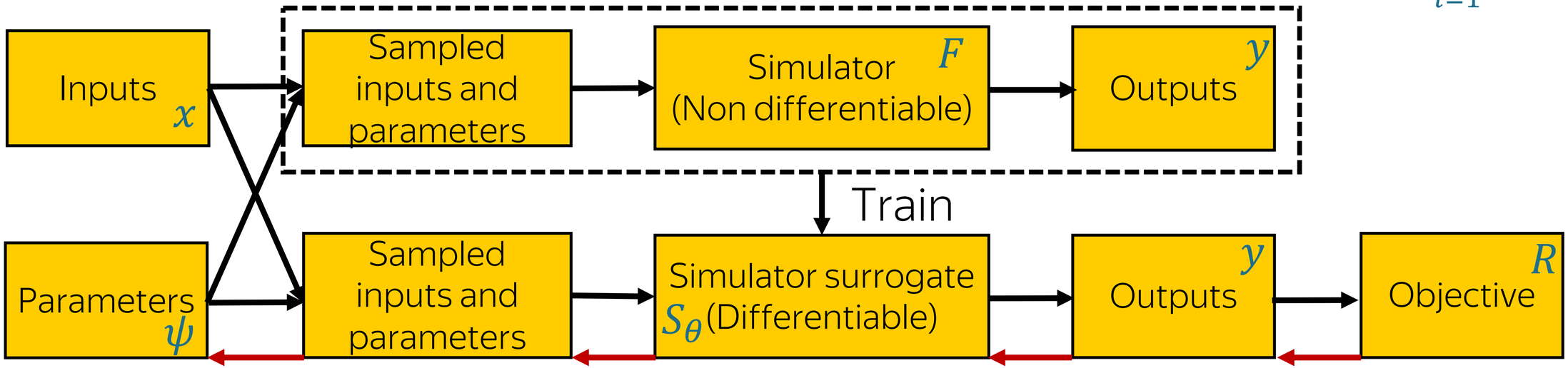
Generative surrogates

$$\nabla_\psi E_y[R(y_\psi)] \approx \frac{1}{N}\sum_{i=1}^{N} \nabla_\psi R\big(F(x_i;\psi_i)\big) \approx \frac{1}{N}\sum_{i=1}^{N} \nabla_\psi R(S(z_i, x_i;\psi_i))$$



Generative surrogates

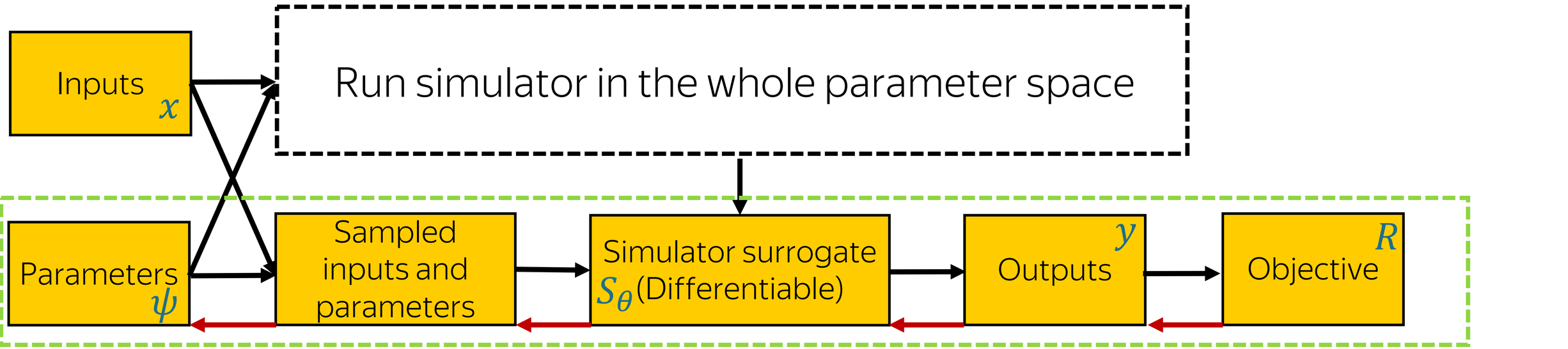
$$\sum_{i=1}^{N} \nabla_{\psi} R\left(S_{\theta}(z_i, x_i; \psi)\right)$$



- S_θ any conditional deep generative model: GAN, NF, VAE, …
- Once trained produce differentiable samples:

$$\nabla_{\psi} E_y[R(y)] \sim \sum \frac{\partial R}{\partial y_i} \times \frac{\partial y_i}{\partial \psi} = \sum \frac{\partial R}{\partial y_i} \times \frac{\partial S_\theta}{\partial \psi}$$

Generative surrogates: training

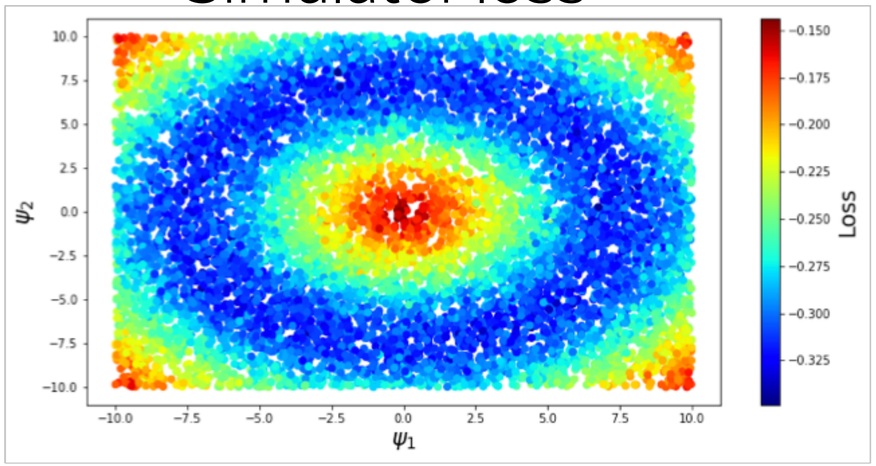


- Generate ψ on the grid
- S_θ is trained ***once*** in the ***whole*** space of parameters ψ

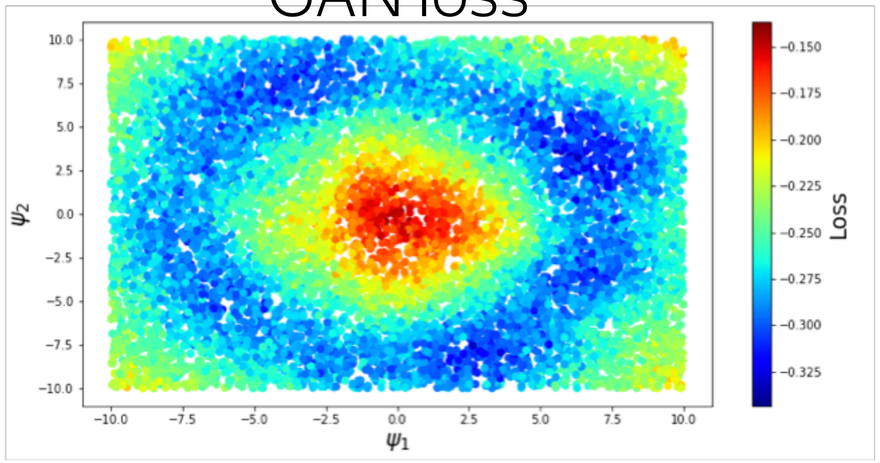
$$\psi_{t+1} = \psi_t - const * \frac{1}{N}\sum_{i=1}^{N} \nabla_\psi R(S(z_i, x_i; \psi_i))$$

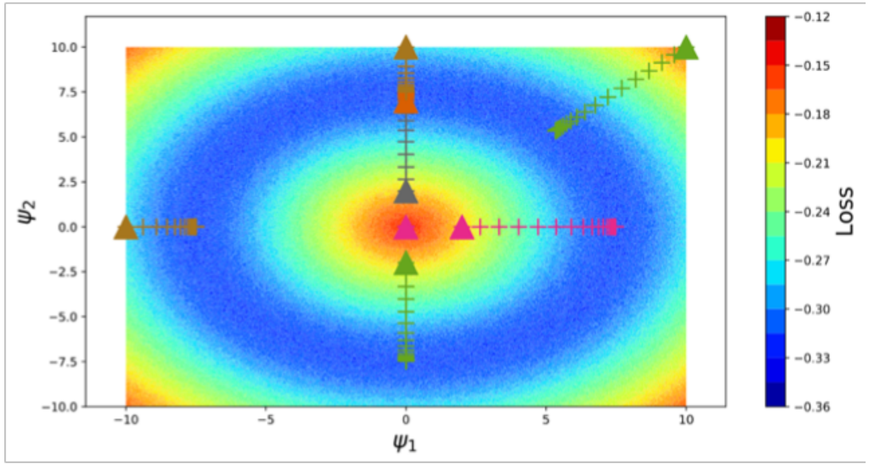
Generative surrogates: toy example

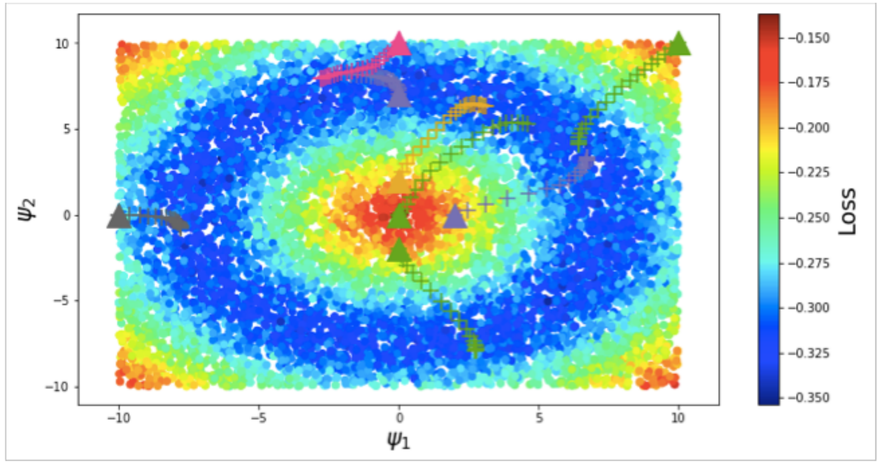
Simulator loss



GAN loss

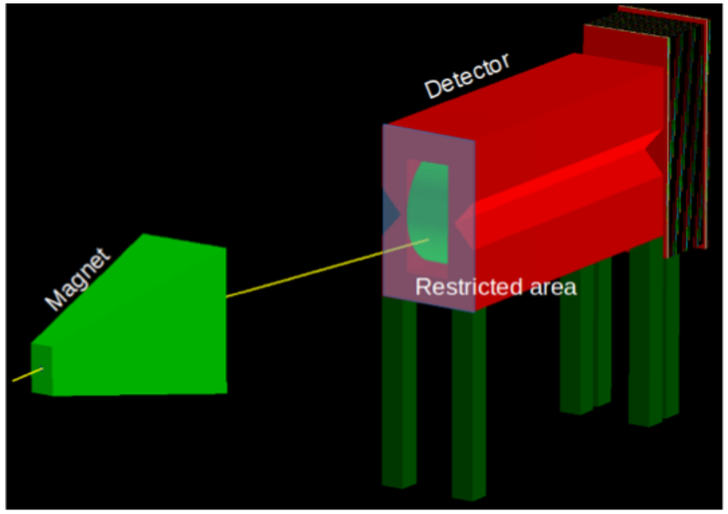




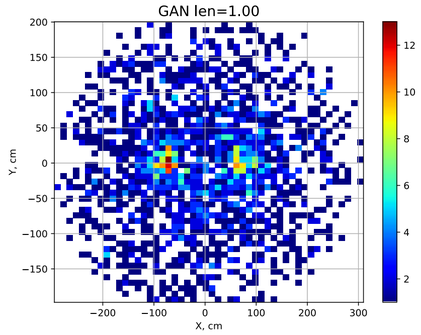


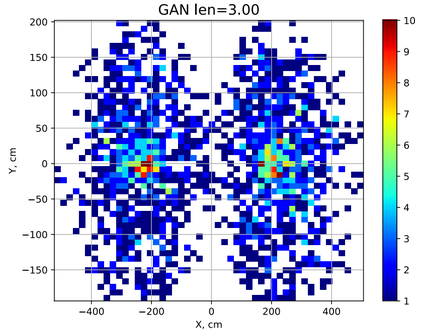
Generative surrogates: physics toy example

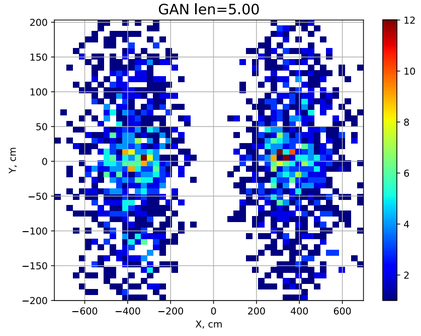
$\psi \in \mathbb{R}^1$



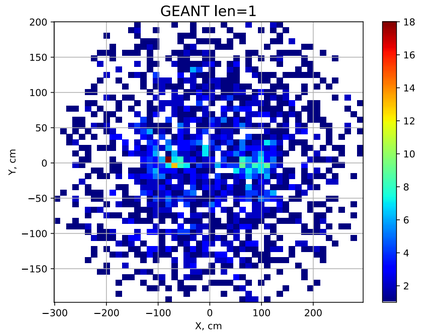
GAN

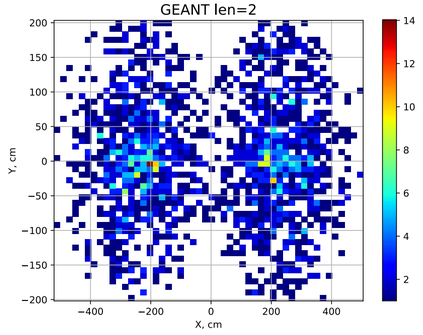


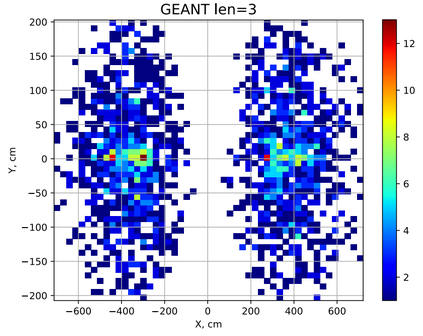




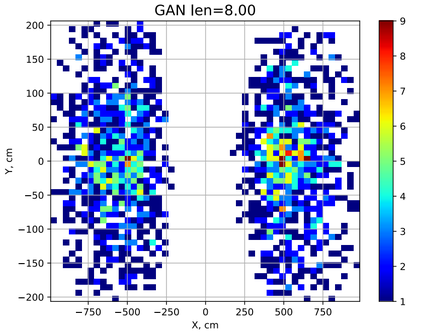
Simulator

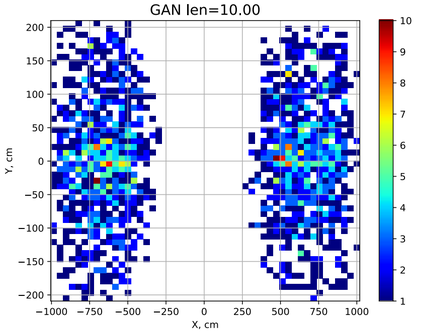


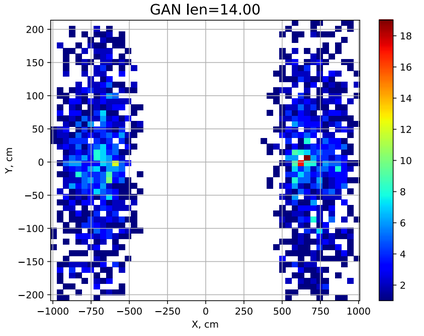




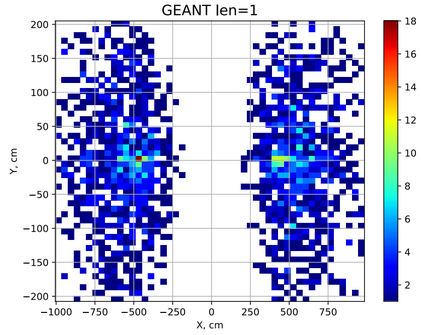
GAN

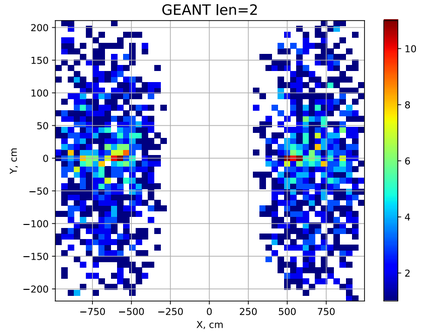


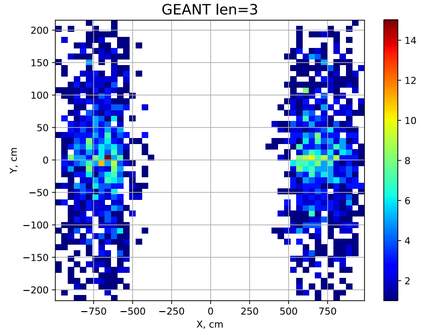




Simulator





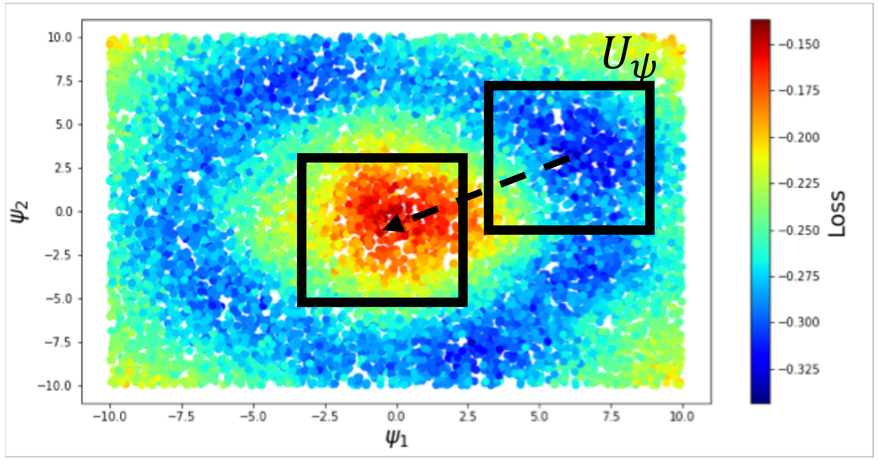


There is no locality…

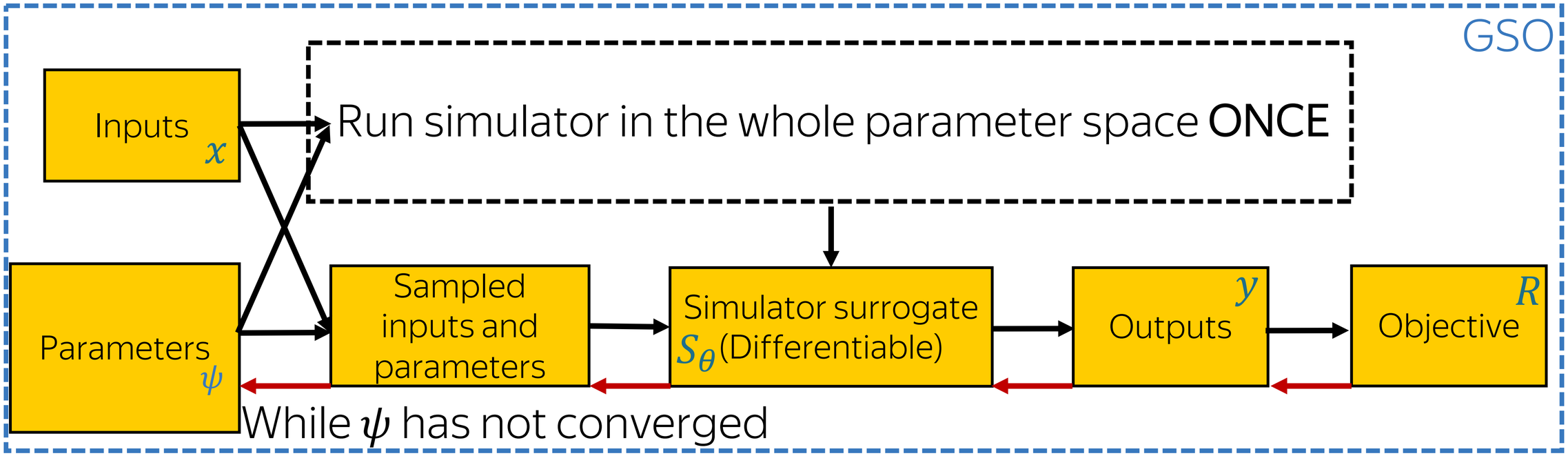
S_θ is trained in the whole ψ space:

- Curse of dimensionality
- Number of samples scales exponentially with dimension d: $\left(\frac{L}{\Delta}\right)^d$
- Generative models do not extrapolate well

Need to impose locality in the ψ space

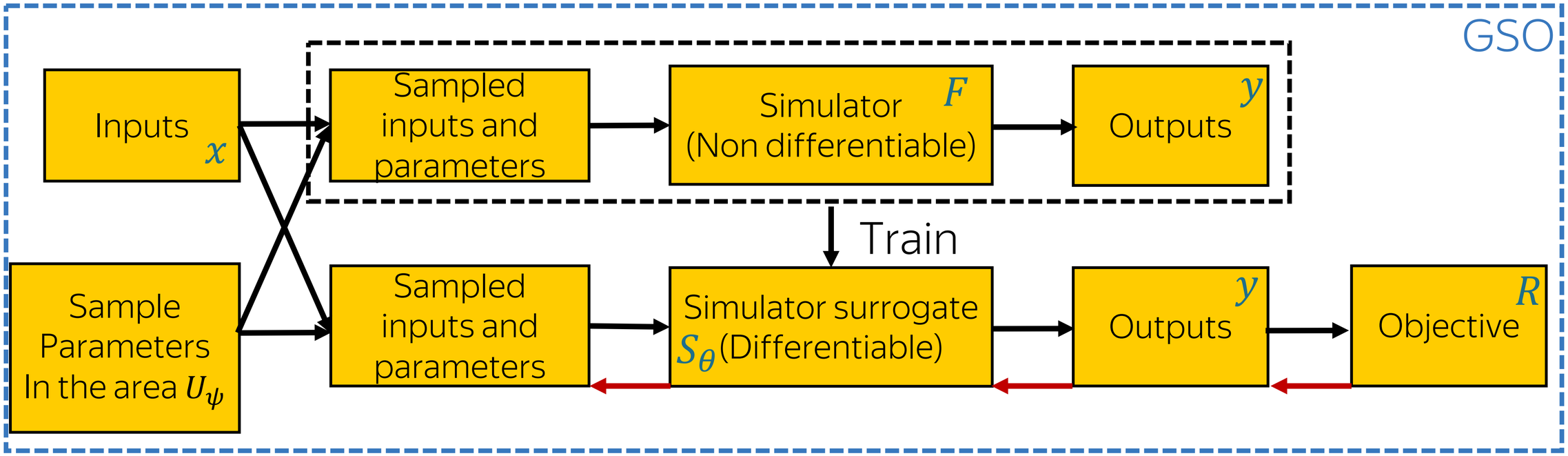


Local Generative Surrogates (L-GSO)



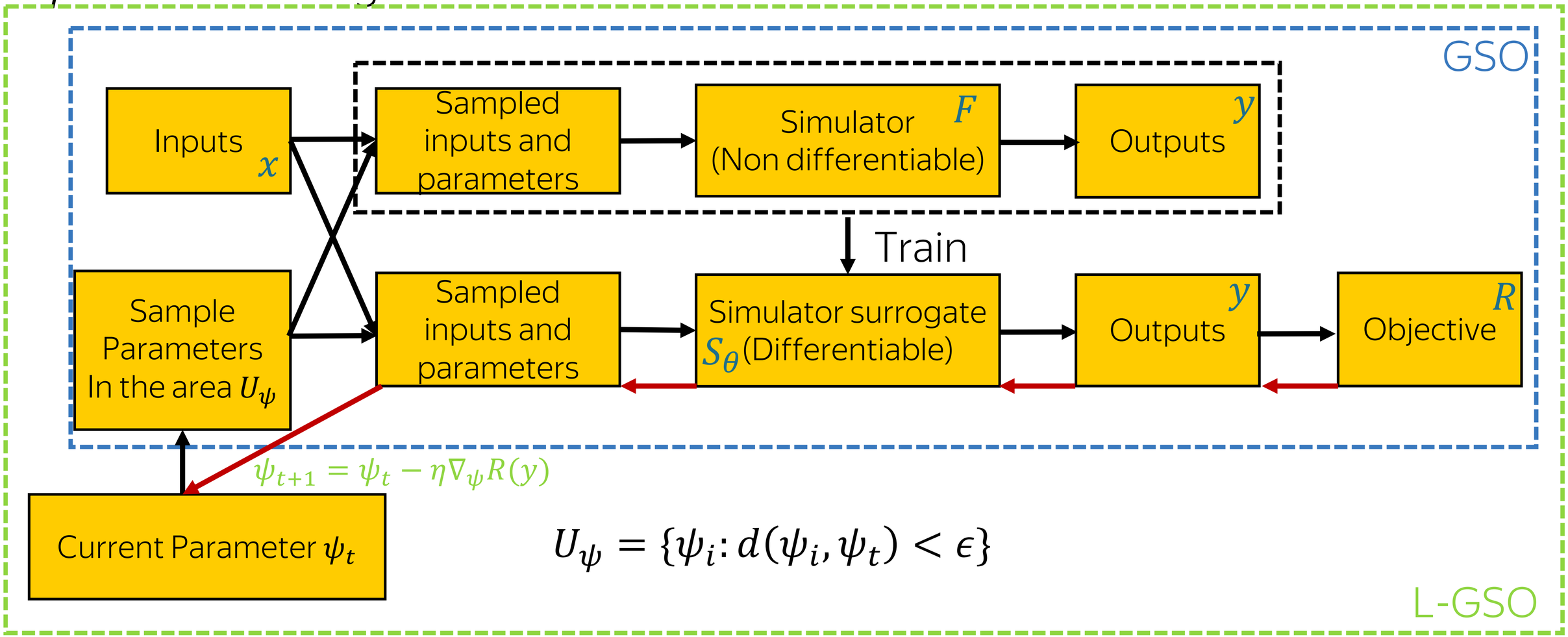
Local Generative Surrogates (L-GSO)

While ψ has not converged do:

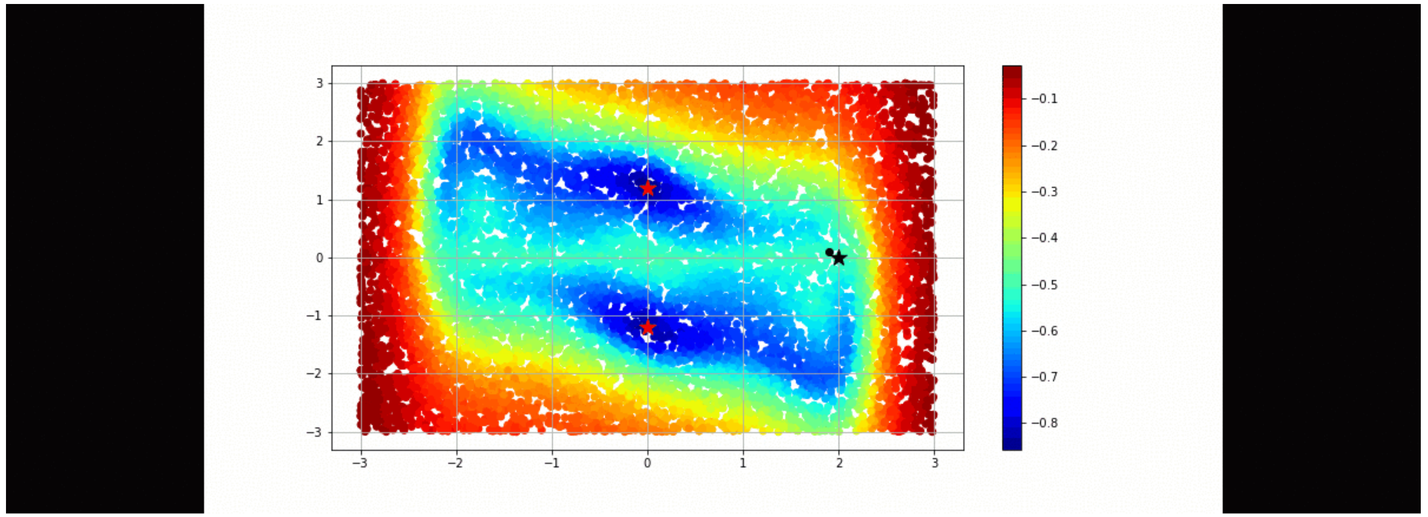


Local Generative Surrogates (L-GSO)

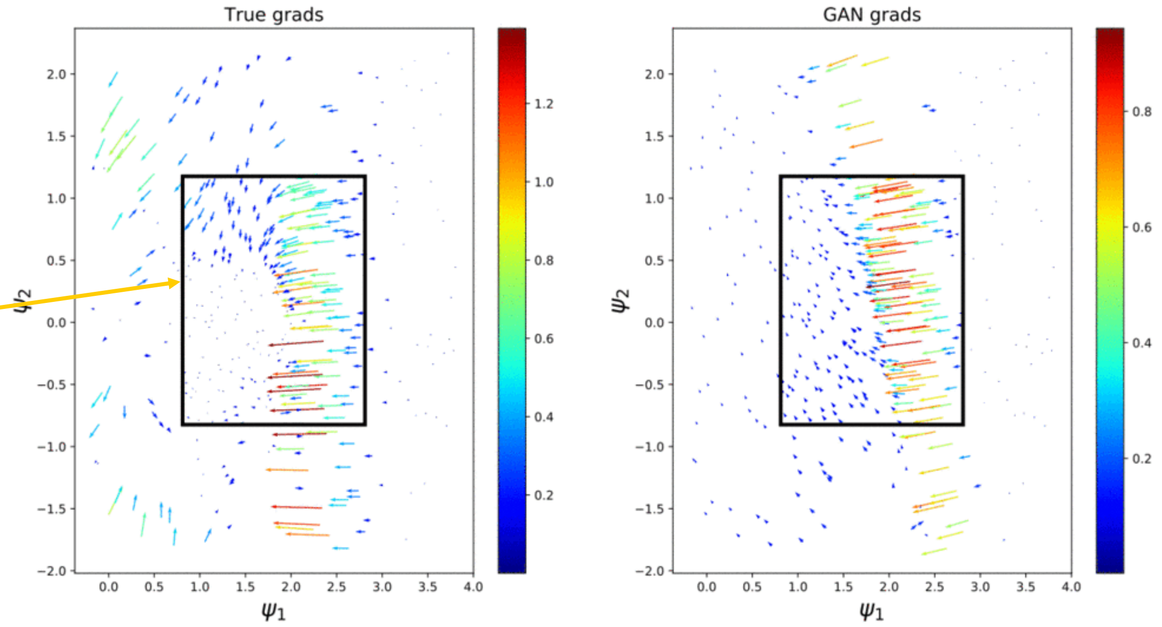
While ψ has not converged do:



Local Generative Surrogates (L-GSO)



U_ψ where the surrogate is trained



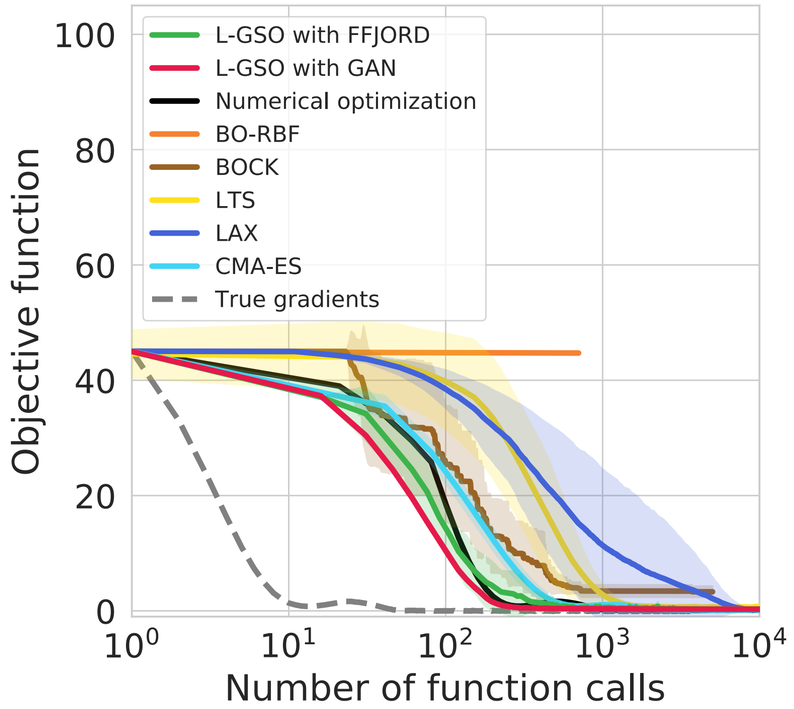
Gradients are well estimated in the local area

Link:https://doi.org/10.6084/m9.figshare.9944597.v3

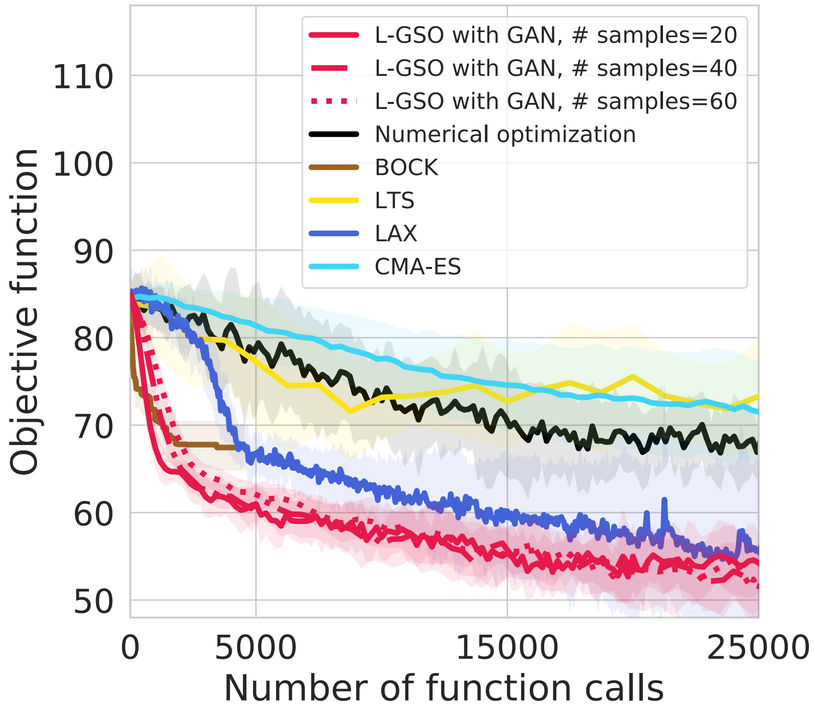
Toy Experiments

- 4 toy problems
- Various dimensions
- Degenerate parameters
- Costly simulator call
- Compare in:
  - number of calls
  - attained minimum

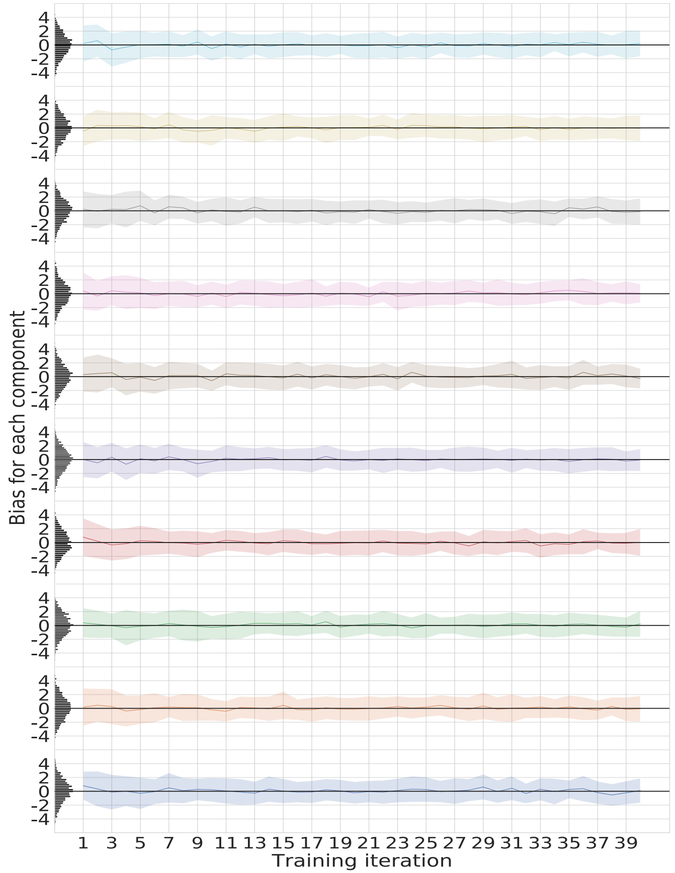
Rosenbrock problem
10-dim



Neural Networks weights
optimisation
91-dim



L-GSO bias



$$Bias_t = \nabla_{\psi|\psi_t} R(y_\psi) - \nabla_{\psi|\psi_t} R(\bar{y})$$

- No bias observed for L-GSO

SHiP: Shield optimisation

Muon kinematics, including start coordinate

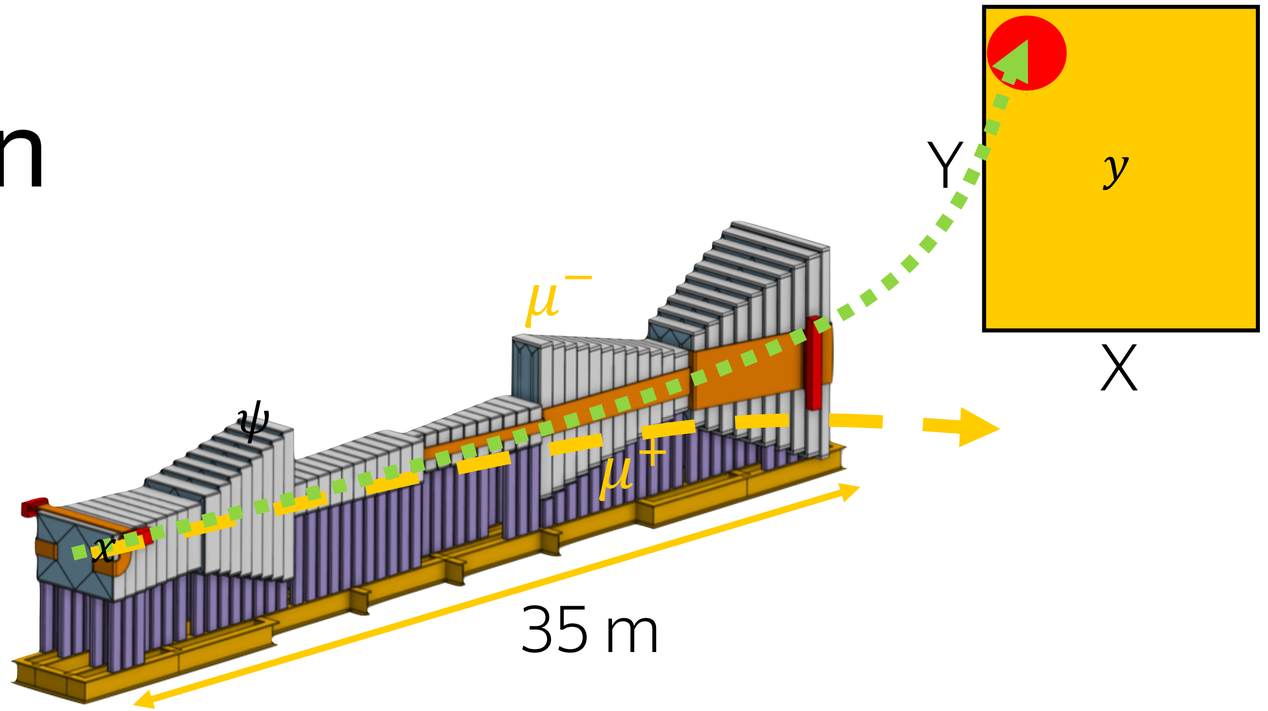
$$x = \{P, \phi, \theta, Q, \boldsymbol{C}\}, X \in \mathbb{R}^7$$

Output: coordinates of the muon hit

$$y = \{X, Y\}, y \in \mathbb{R}^2$$

Optimised parameters: shield geometry

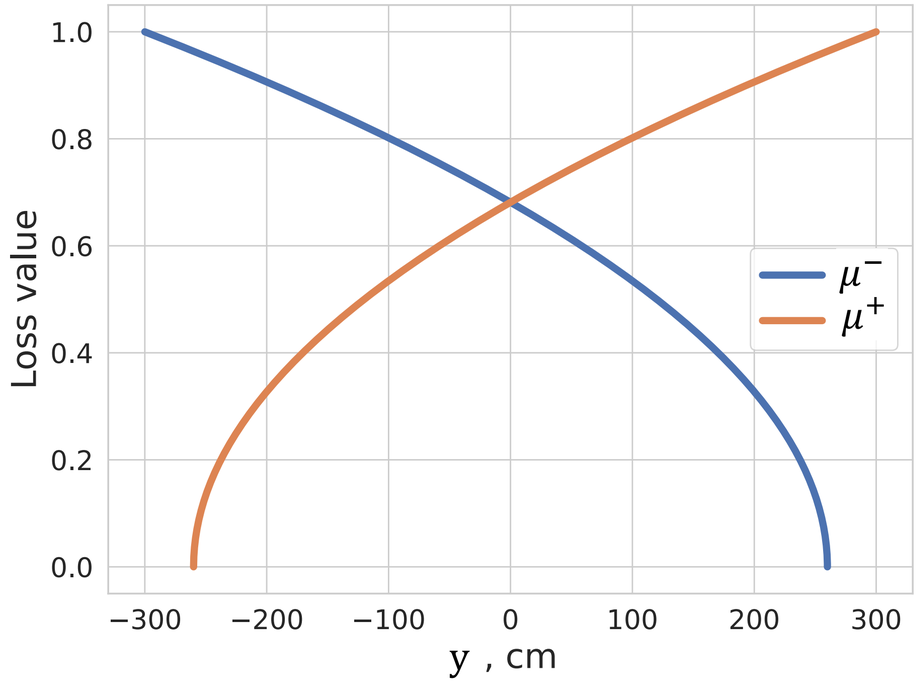
$$\psi \in \mathbb{R}^{42}$$



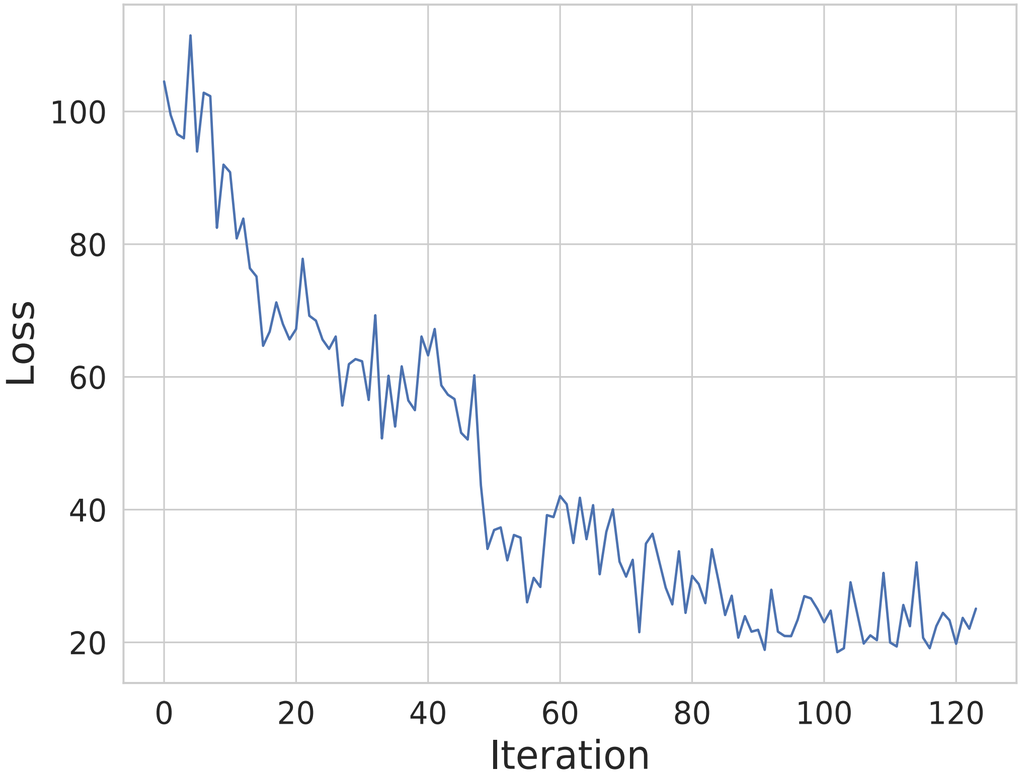
Objective function

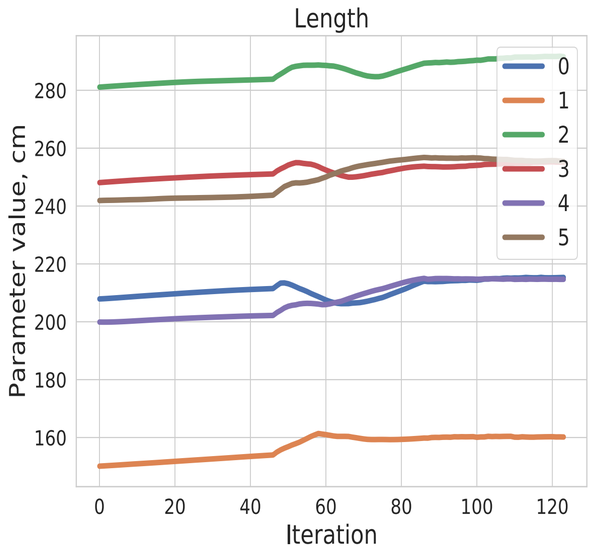
$$R(y;\alpha) = \mathbf{1}_{Q=-1}\sqrt{(\alpha_1 - (y + \alpha_2))/\alpha_1} + \mathbf{1}_{Q=1}\sqrt{(\alpha_1 + (y - \alpha_2))/\alpha_1}$$

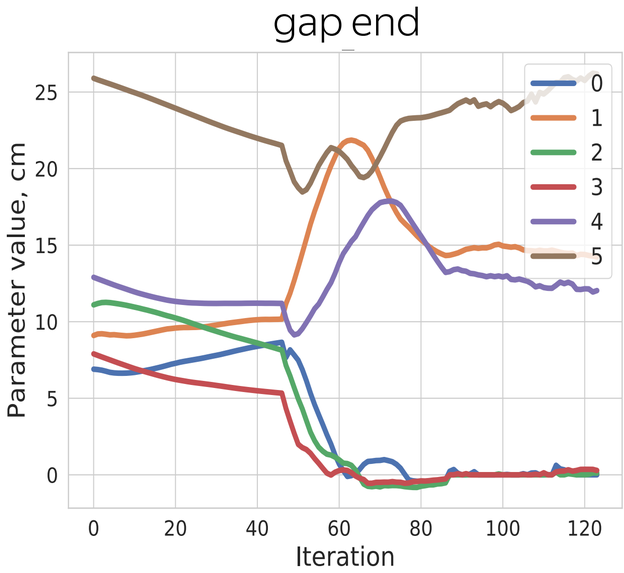
Was previously optimised with BO

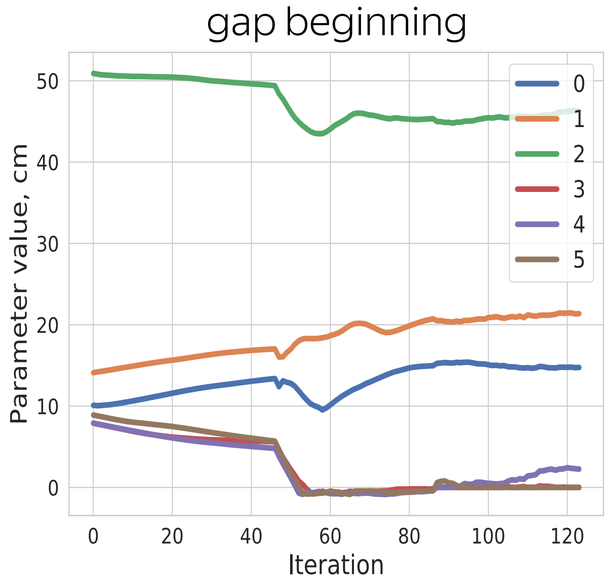


SHiP: Parameter changes





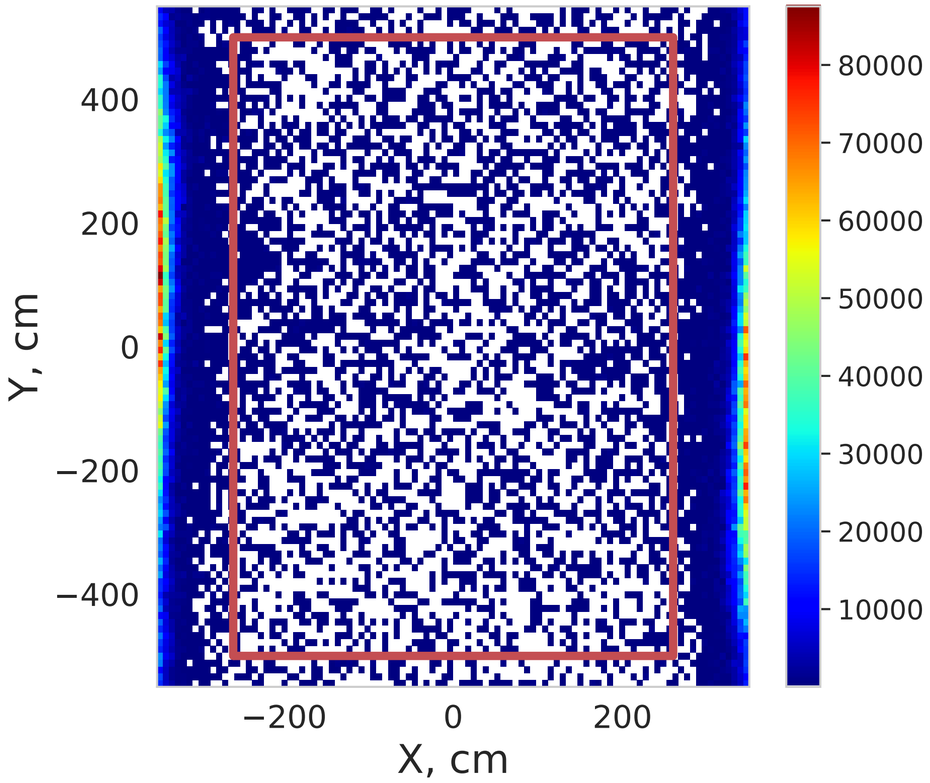




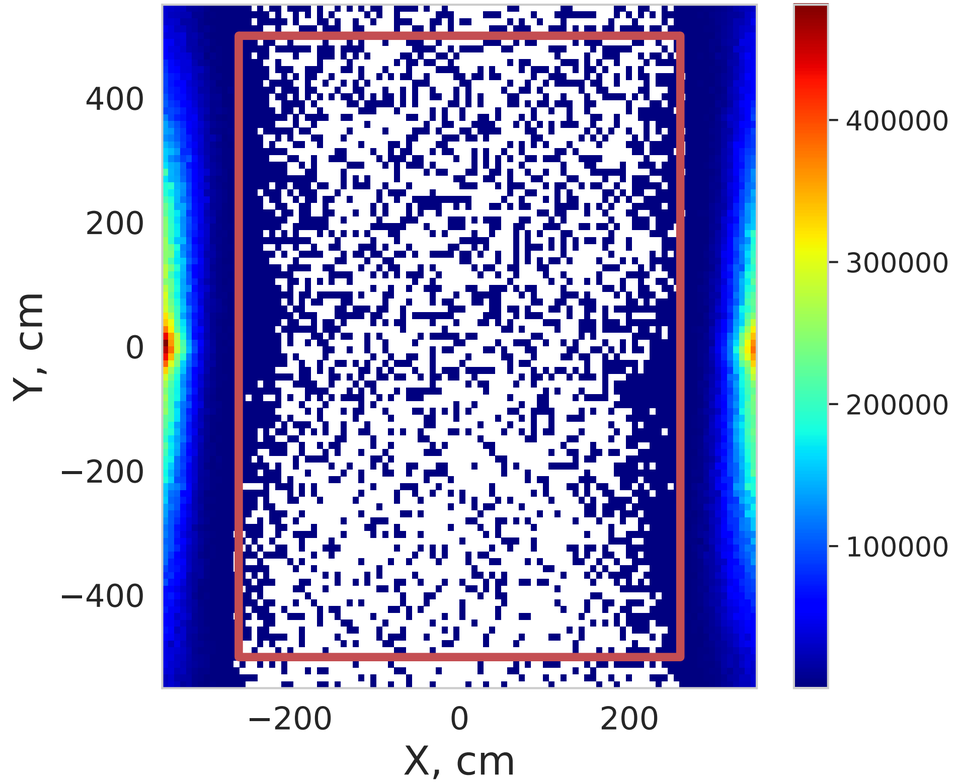
Continuous changes of the parameters:
can give some insights about physics!

SHiP: shield optimisation comparison

Previous optimum(BO)



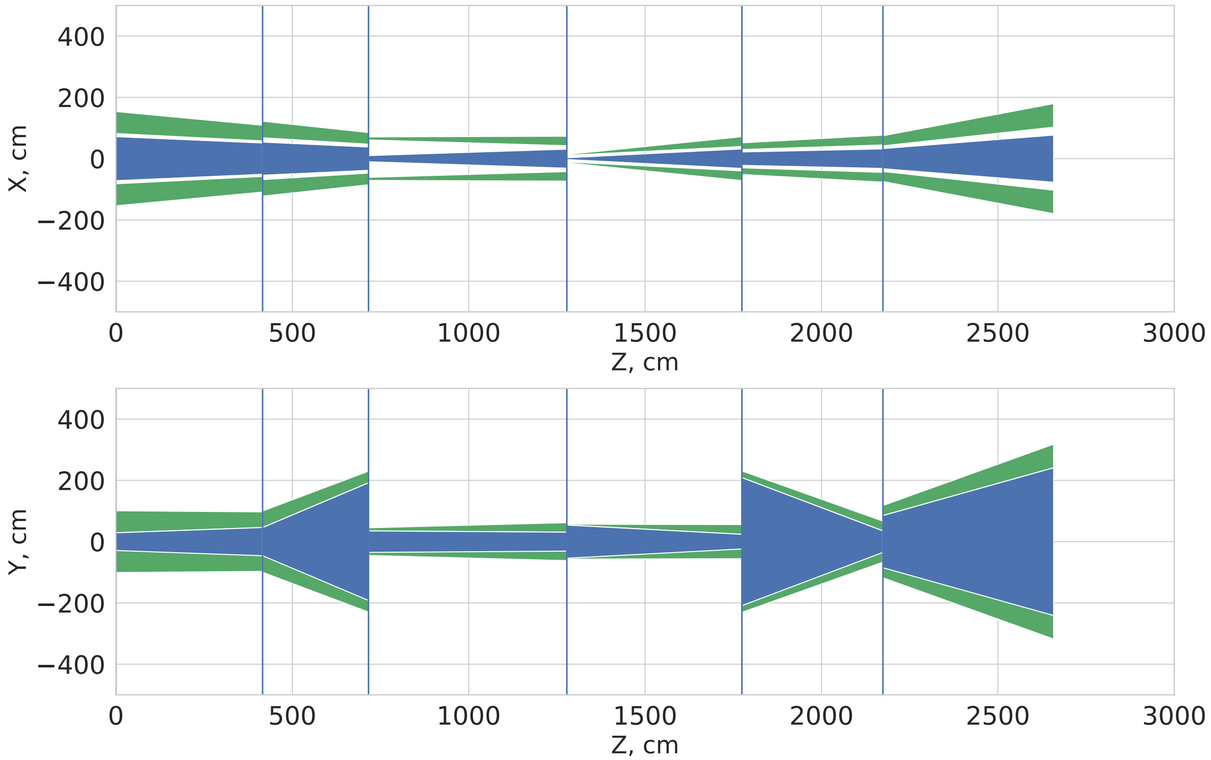
New optimum(L-GSO)



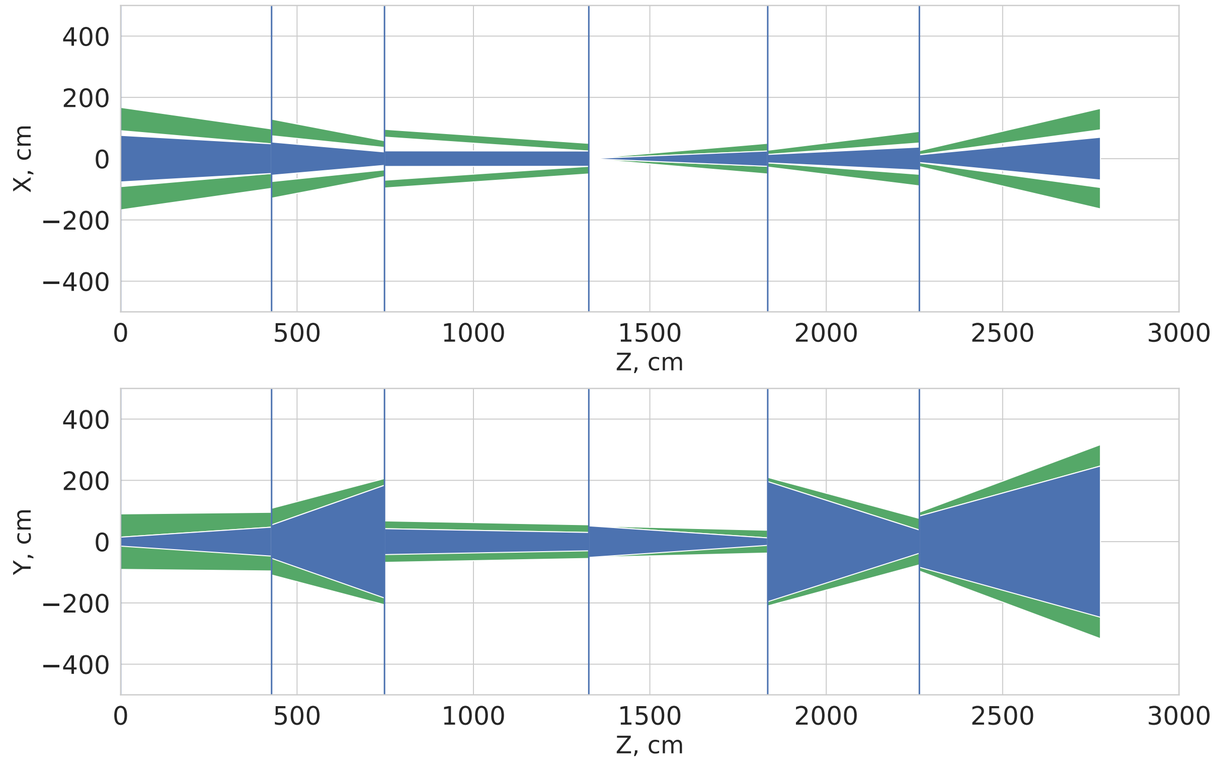
Method	Loss	Shield length (m)	Magnet weight (kt)
L-GSO	~ 2200	33.39	1.05
Bayesian opt.	~ 3000	35.44	1.27

SHiP: shield geometry change

Initial shape



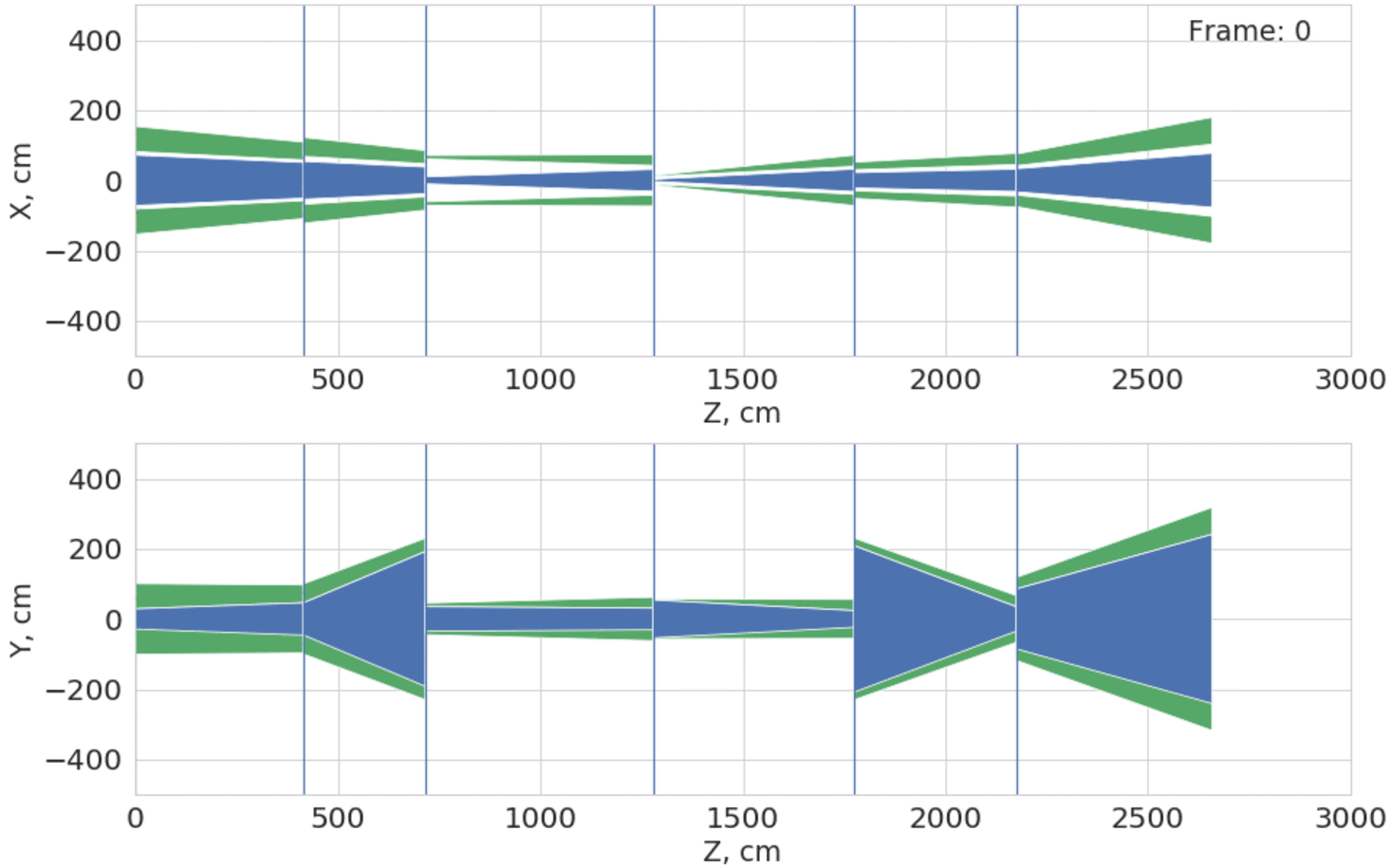
Final shape



Animation of the optimisation: https://doi.org/10.6084/m9.figshare.11778684.v1

SHiP: shield geometry change

Animation



Conclusion

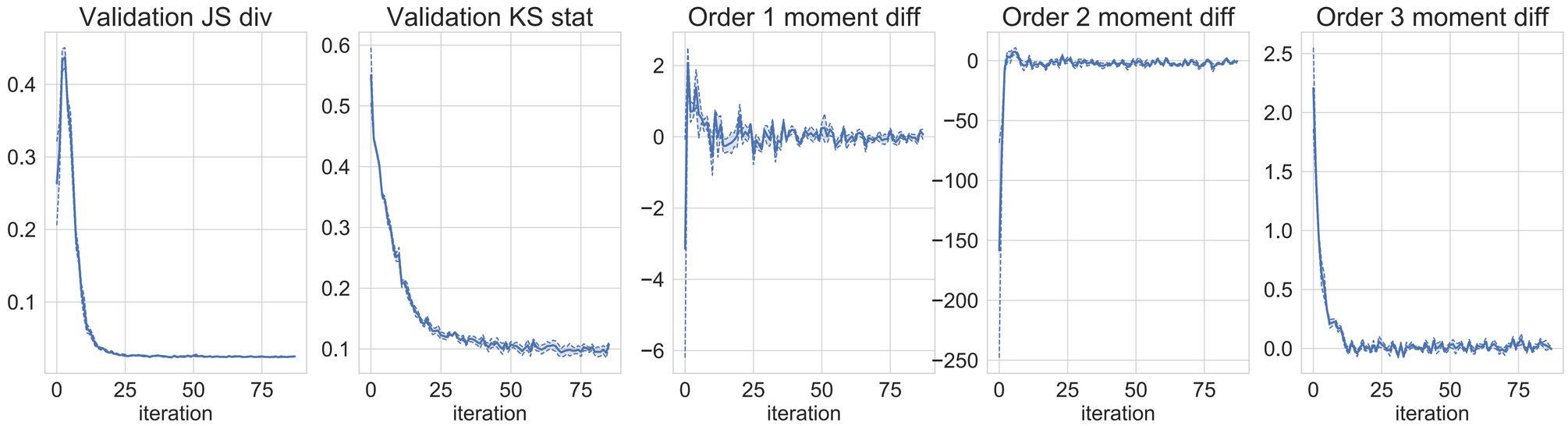
- Present novel optimisation approach: L-GSO
- Excel:
  - Parameters lies on a low-dimensional manifold
  - Simulator call is costly
- Empirically low variance
- Attained better minima than Bayesian optimisation in HEP problem

Future work:

- Implementation of trust-region methods
- Combination of BO and surrogate gradients
- Estimation of second order derivatives

Backup

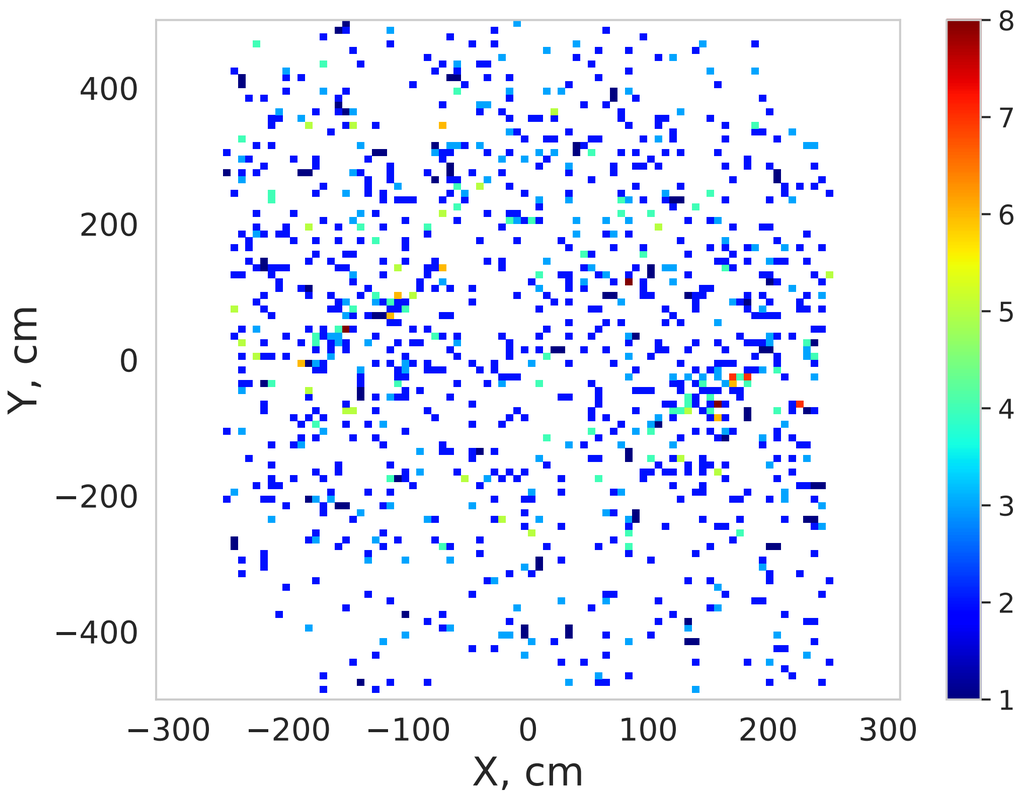
Monitoring of model performance



- Monitor various metrics between train distribution and sampled distribution
- Abort optimisation in case of divergence
- Adjust hyper parameters

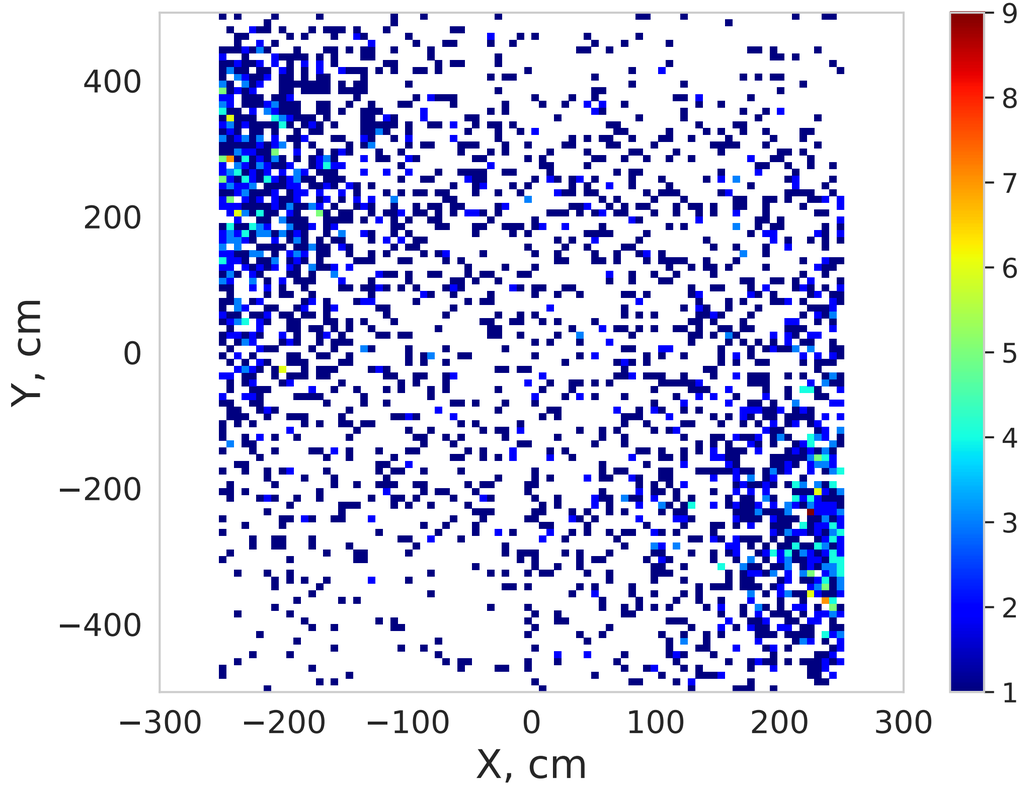
Shield optimization: Full geometry

Previous optimum(BO)



Number of hits (weighted):
~1500 (55653)

New optimum(L-GSO)

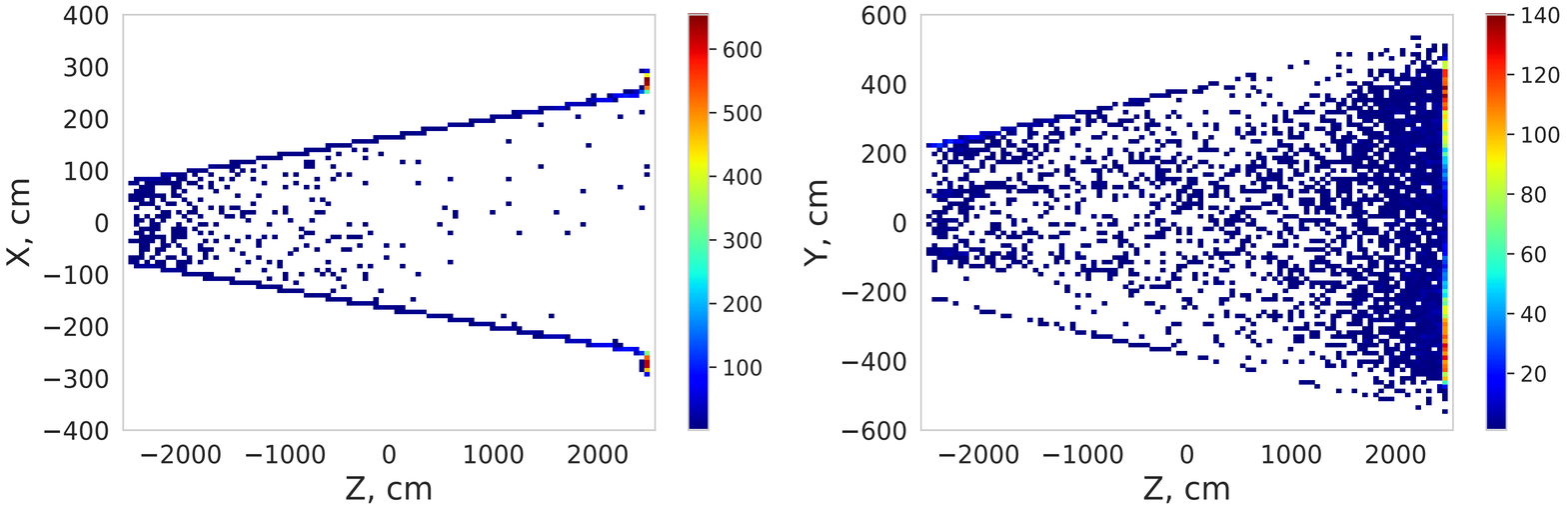


~4800 (245131)

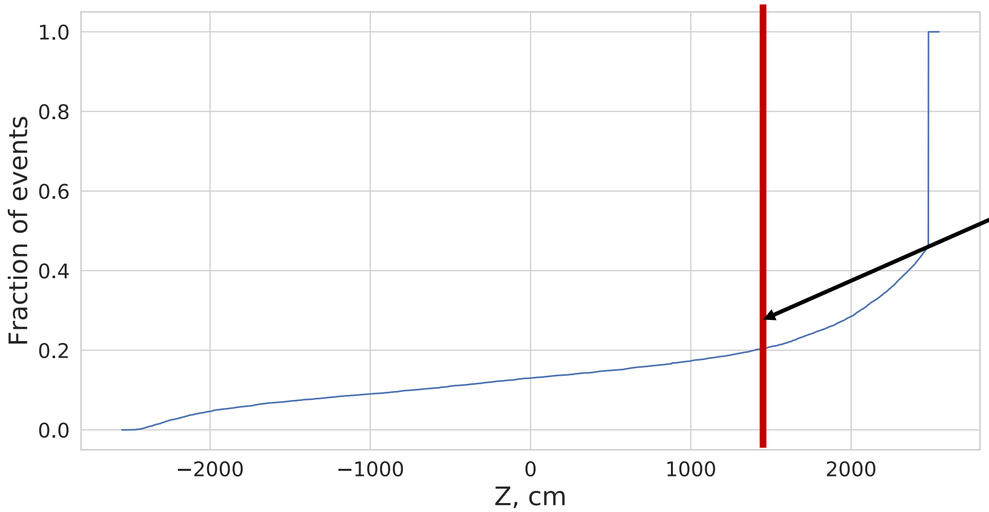
- Why is there a discrepancy between full and simplified geometry results?

Discrepancy reason between full and simplified geometries

Decay volume veto hits



Exponential flux of muons is induced by the muons that would have missed the detector in the absence of the decay volume, but are scattered back by the decay volume.



Exponential growth

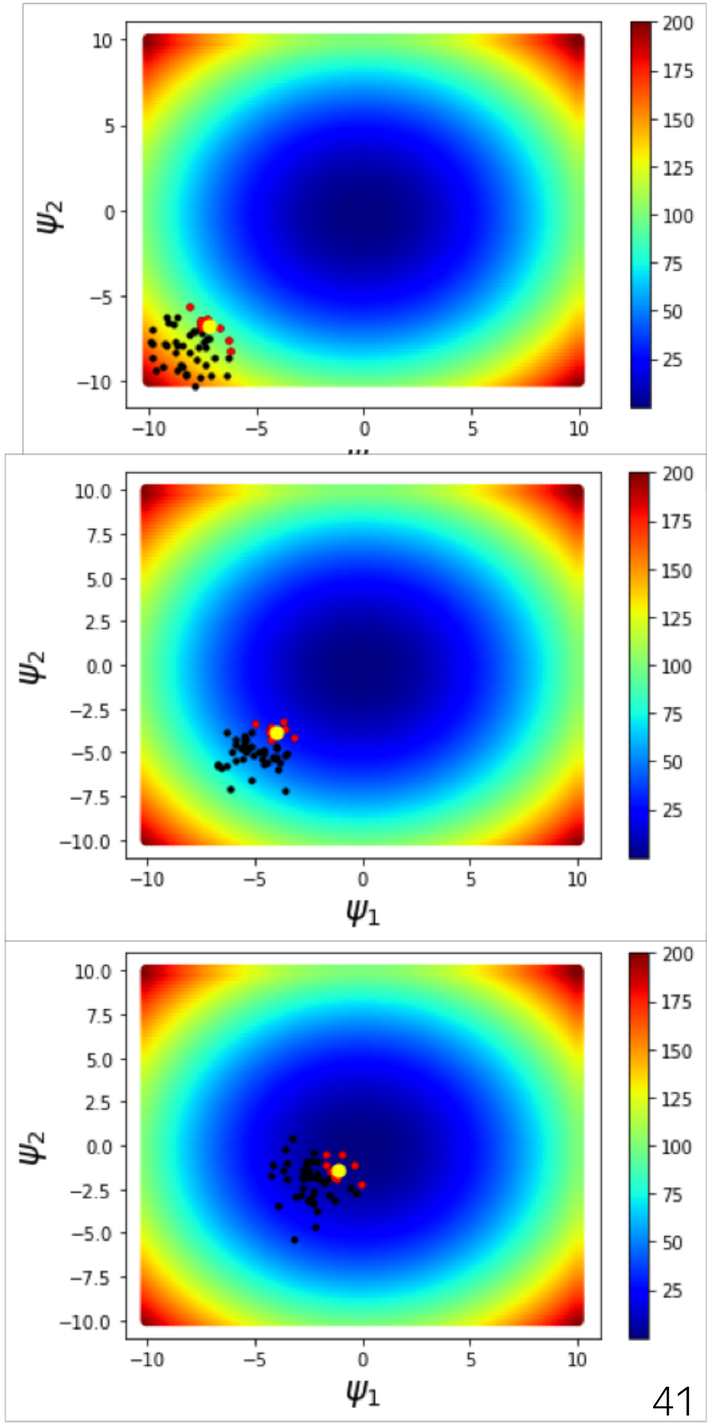
Evolutionary strategies

- Simple case of Gaussian ES:
  - Set $\theta = (\mu, \sigma), p_\theta(\psi) = N(\mu, \sigma^2 I)$
  - Sample M values of ψ_i, compute $f(\psi_i)$
  - Select best K values by sorting $f(\psi_i)$
  - Update μ, σ, using selected ψ_i

- Usually requires large number of sample M.

- Modifications such as CMA-ES[1] or Guided ES[2] might utilse surrogate gradient information.

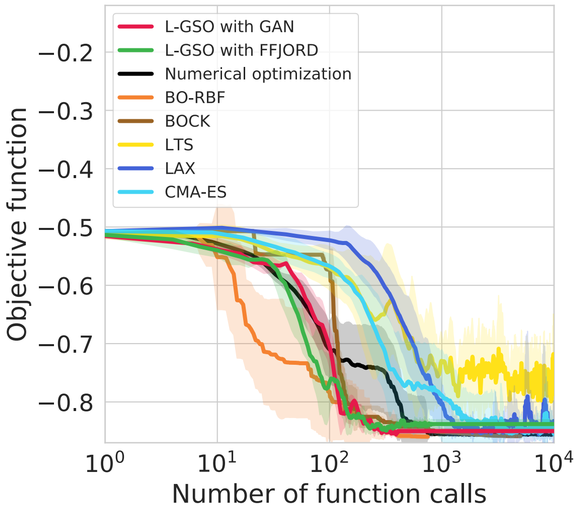
- We compare to Guided ES.



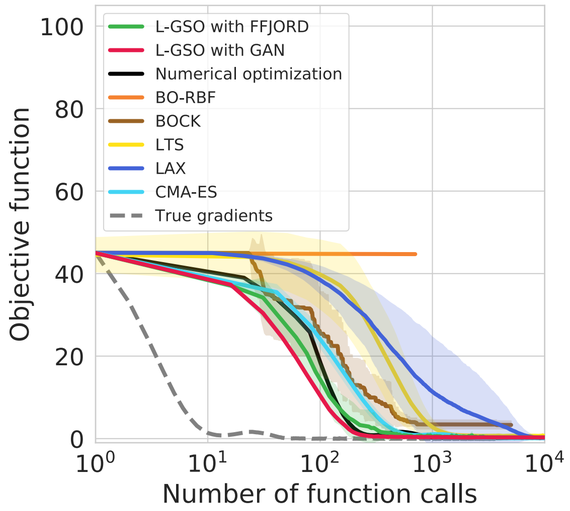
[1] http://www.cmap.polytechnique.fr/~nikolaus.hansen/cmaartic.pdf, [2] https://arxiv.org/abs/1806.10230

Toy Experiments

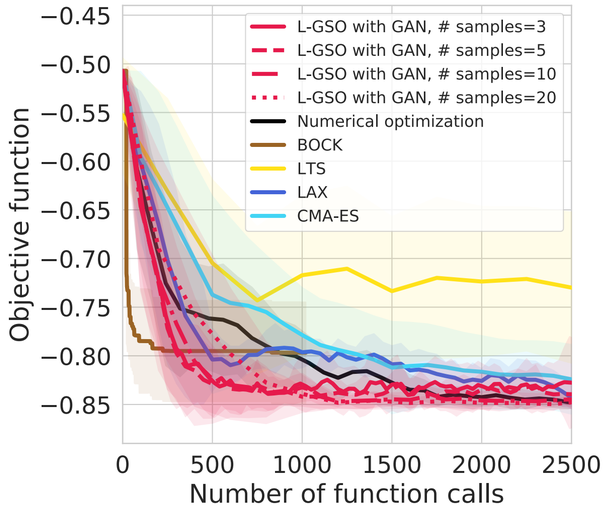
Three-Hump problem
2-dim



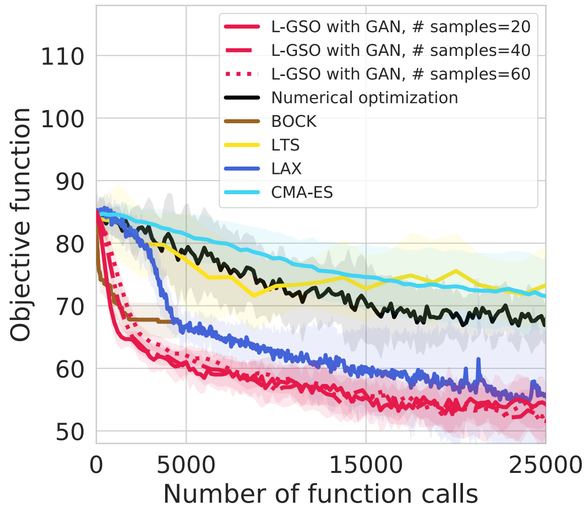
Rosenbrock problem
10-dim



Nonlinear Three-Hump problem,
40-dim



Neural Networks weights optimisation
91-dim



- L-GSO comparable to **all** baselines in low-dim problems in the speed of convergence
- L-GSO **outperforms all** baselines in a high-dim setting when parameters lie on a lower dimensional manifold.
- L-GSO has lower variance in resulting objective function value than other methods