Surrogate Models for Fun and Profit

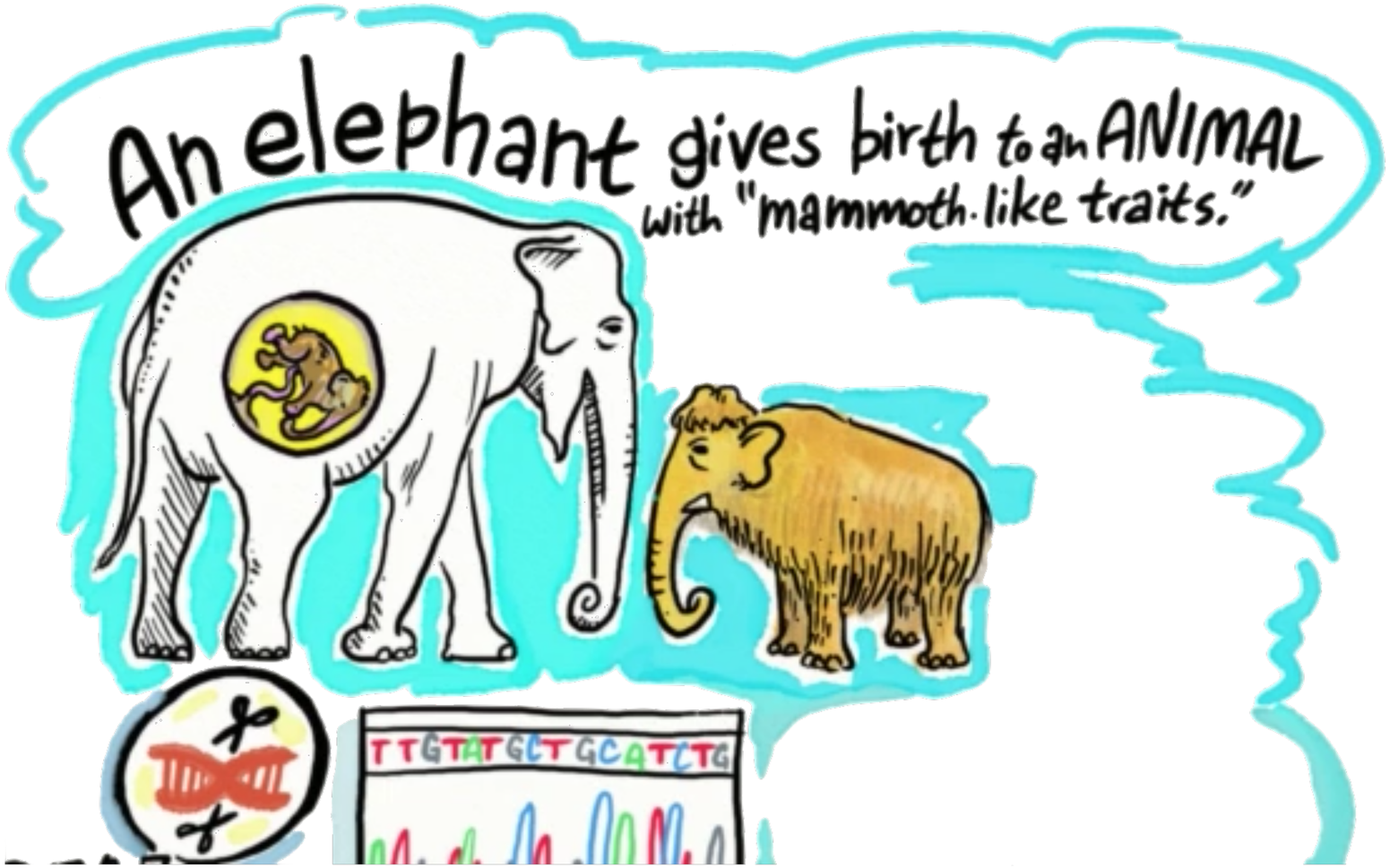
9 Apr 2018, IML Workshop

Andrey Ustyuzhanin on behalf of the team

Yandex

National Research University Higher School of Economics

Imperial College London



http://bit.ly/2GC1U8h
https://vimeo.com/123017198

Outline

- Introduction
- Surrogate modelling cases
  - ferro-alloy quality control (Russian steel production factory - MMK)
  - Ship active shield
  - ML model hyperparmeter tuning
- How it works
  - Some boring details
- Natural Evolution Strategy
  - Cool method to remember
- Moar stuff
  - more examples (trigger efficiency estimation, e-cal efficiency estimation, read-out modelling)
  - Tips & tricks & open challenges
- Conclusion

Why Bother

Examples

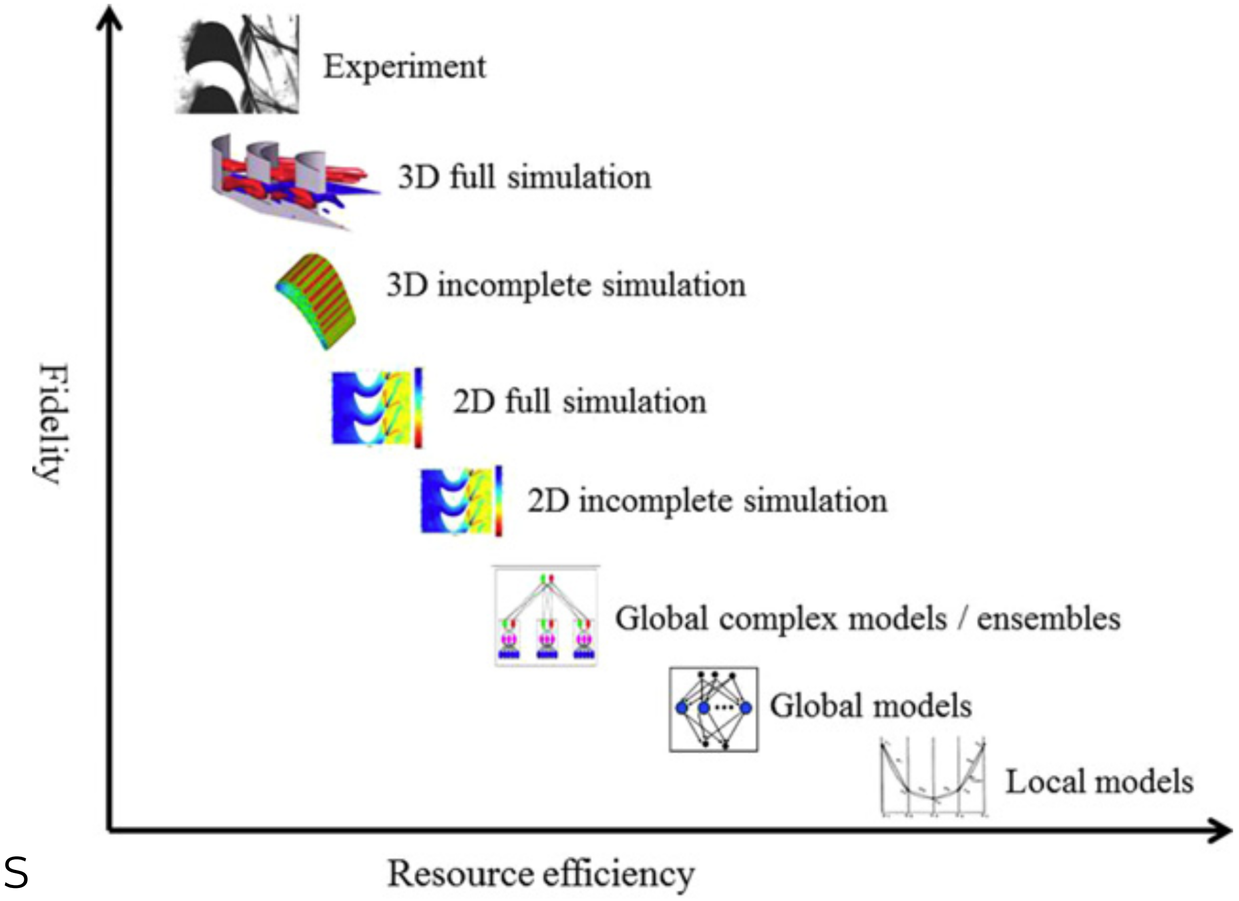
- Drug design
- Aero engineering (aircraft wing profile, turbine)
- Steel production quality control
- Weather forecasting, climate change prediction
- …

Regular physically-based computational models are fine but

- Expensive computation
- Large parameter space
- Numerical noise / uncertainty

Surrogate Models

- Model
  - Computational (e.g. Computational Fluid Dynamics)
  - Data-driven
- Goals
  - Design optimization
  - Configuration space approximation
- Most critical task:
  - A strategy for using the surrogates is known as *model management*.



http://bit.ly/2JoIce2

Case1: Steel production control

The process

- Steel production at Magnitogorsk Iron & Steel Works

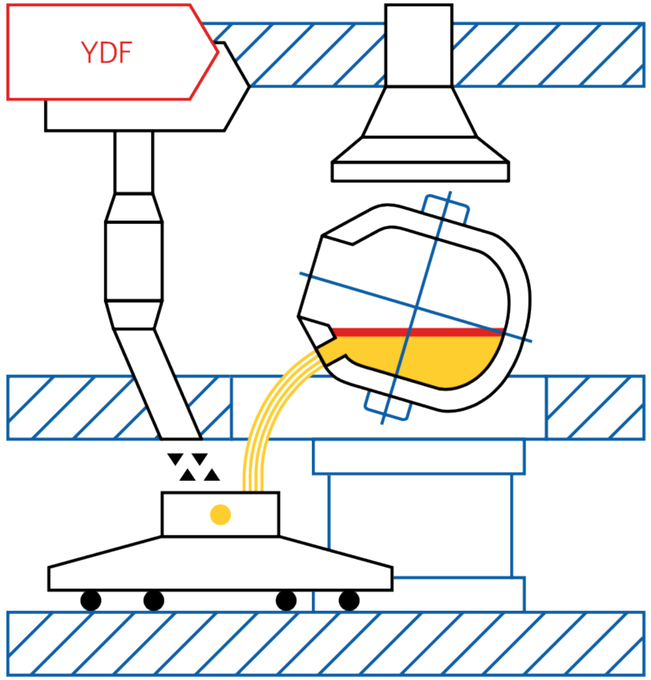
The goal:

- Control the consumption of ferroalloys during the process to decrease its consumption while keeping steel quality at the specific level (*optimization*)

Solution:

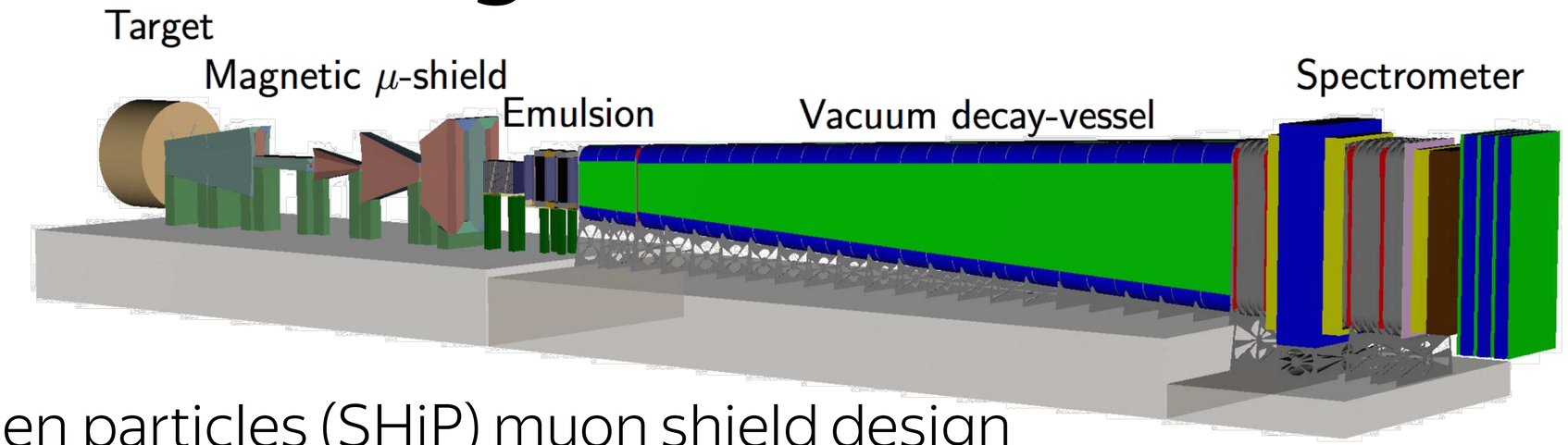
- Build complex surrogate model that combines both traditional (physically-based) and data-driven models

Result: 5% of ferroalloy saving, which is $4 million in production costs



http://bit.ly/2H4ry50

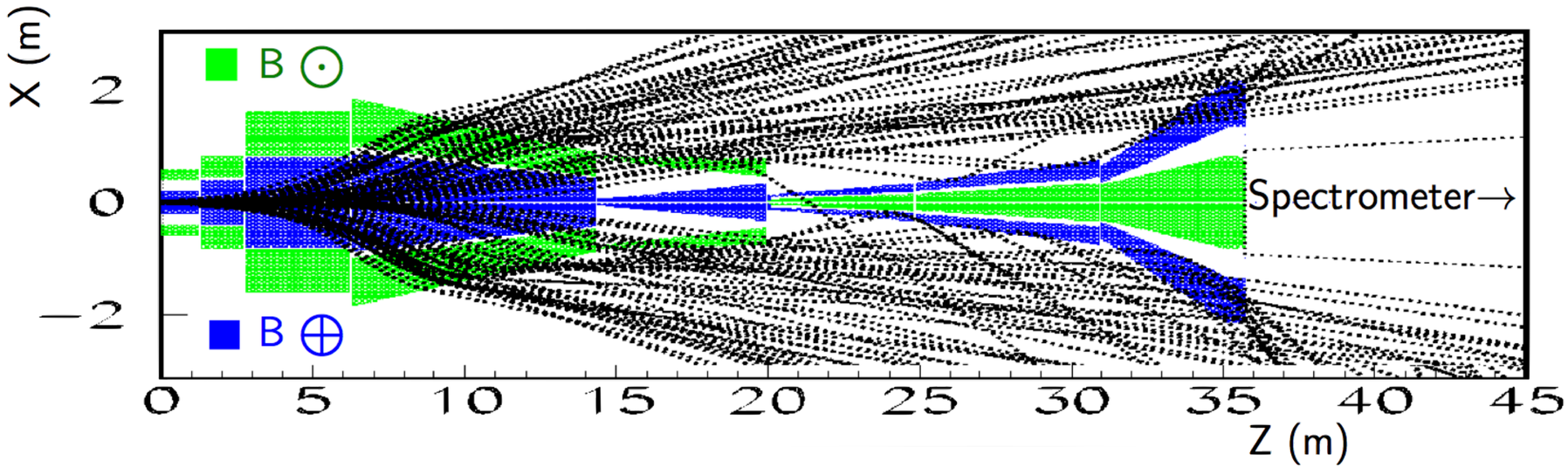
Case2: SHiP shield design



- The process
  - Search for hidden particles (SHiP) muon shield design
- The goal:
  - Find configuration for the active muon shield shape that would block/deviate most of the muons (explore the configuration space and find the best one)
- Solution:
  - Build surrogate models that would help to find the regions of the most promising muon shield configurations

http://ship.web.cern.ch/ship/

Case2: SHiP shield design

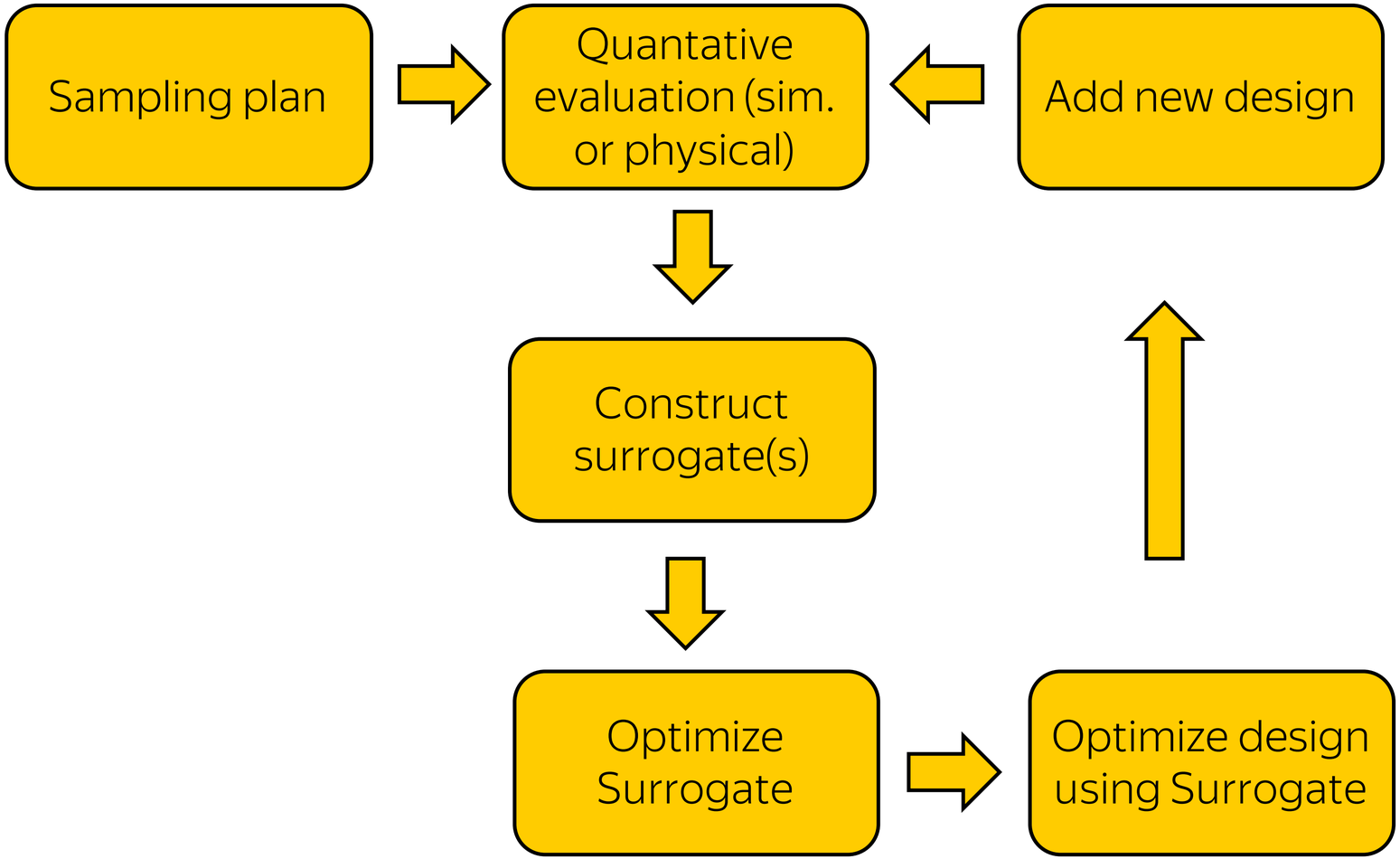


- Parameters
  - Geometry of 7 magnetic blocks (8 parameters per block)
- Metric
  - Leave no muons at the spectrometer volume:
    (x_i - coordinate of i-th muon track at detector plane)

$$\chi = \sum \sqrt{(1 - \frac{x_i + l_1}{l_2})}$$

- Result: “designed” magnets, that are 25% lighter, http://bit.ly/2qjlo5g

Surrogate Modelling. Generic approach

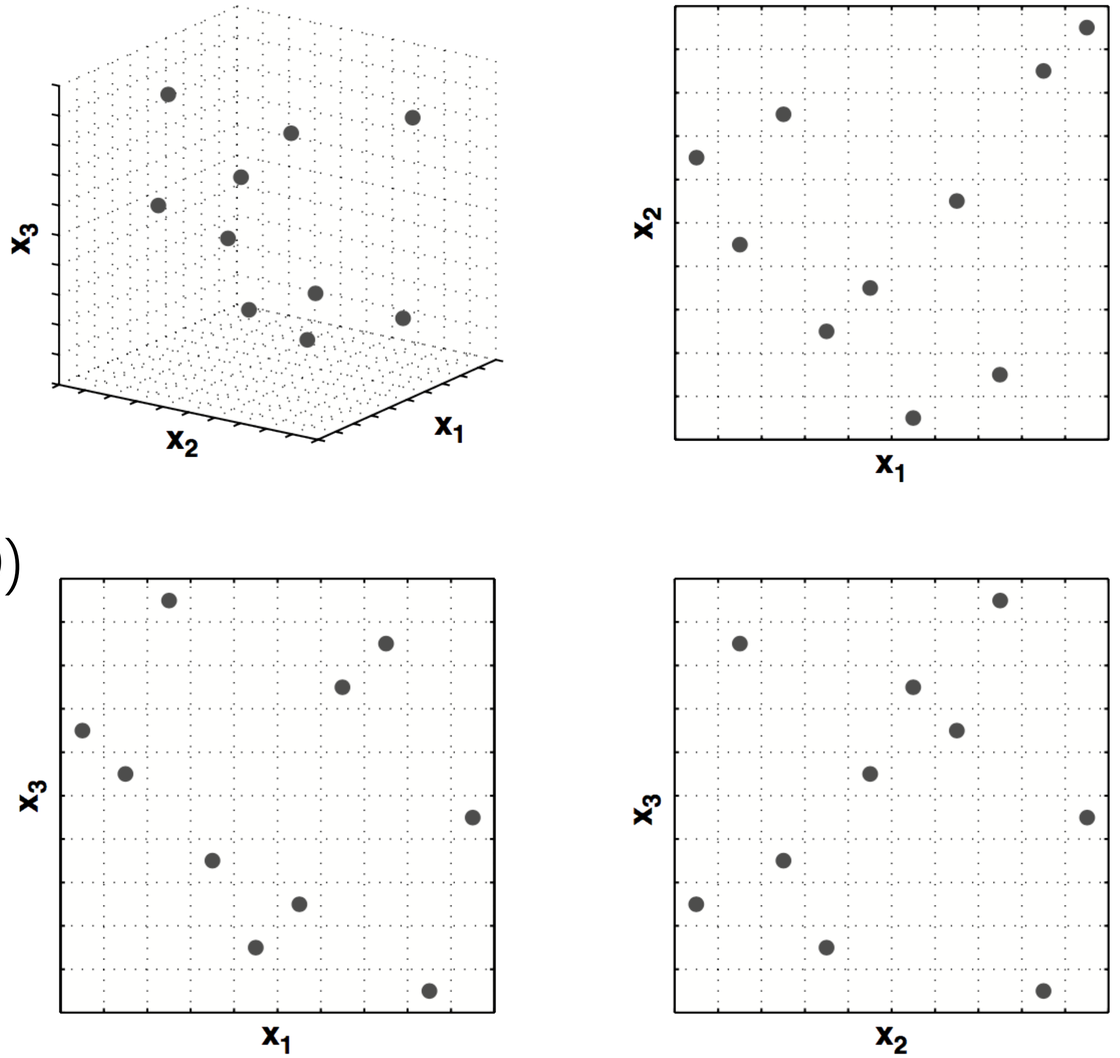


Initial Sampling Plan

Pick initial
set of points that cover possible configurations as evenly as possible, e.g.

- latin square (below for n = 4)
- latin random hypercube (right, 3D)

2	**1**	3	4
3	2	4	**1**
1	4	2	3
4	3	**1**	2



Surrogate Model Types

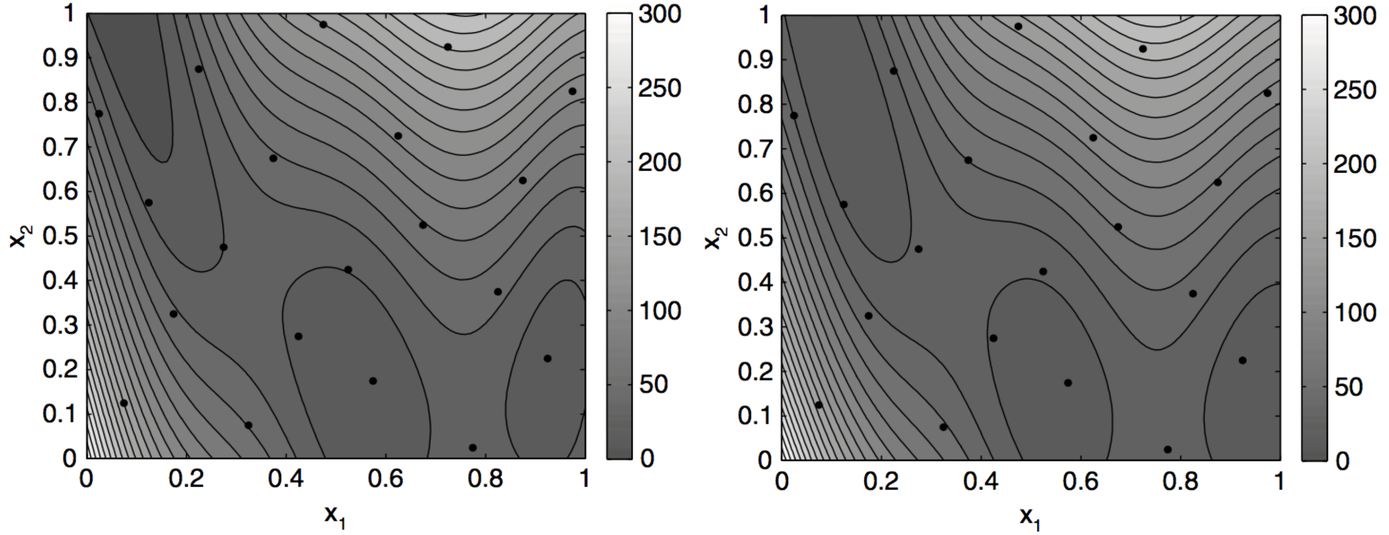
- Polynomial
- Radial Basis Functions (RBF)

$$\widehat{f}\left(\mathbf{x}^{(j)}\right) = \sum_{i=1}^{n_c} w_i \psi(\|\mathbf{x}^{(j)} - \mathbf{c}^{(i)}\|) = y^{(j)}, \quad j = 1, \ldots, n$$

- Gaussian $\psi(r) = \mathrm{e}^{-r^2/(2\sigma^2)}$
- multiquadric $\psi(r) = (r^2 + \sigma^2)^{1/2}$
- inverse multiquadric $\psi(r) = (r^2 + \sigma^2)^{-1/2}$

- Neural Networks
- Kriging / Gaussian Processes

$$\psi^{(i)} = \exp\left(-\sum_{j=1}^{k} \theta_j \mid x_j^{(i)} - x_j \mid^{p_j}\right)$$



Kriging over Branin function after 20 iterations

Surrogate Model Types

Also:

- Random Forest Regression (see scikit-learn)
- Tree-structured Parzen Estimators (Bergstra, et al. (2013))
- Support Vector Regression (Smola, A. J. and Schölkopf, B. (2004))

Pro/Cons

- Uncerctainty estimation
- Computational complexity vs representation power
- Accountability for noise in the data
- Ease of use (see slide on tools)

Optimization algorithms. Taxonomy

As we get surrogate F_c, we can use it to find $x_{min} = \text{argmin}(F_c)$ and check $F_e(x_{min})$

Search for x_{min}:

- Local optimizers (exploiters)
- Global optimizers (explorers)

Alternative taxonomy:

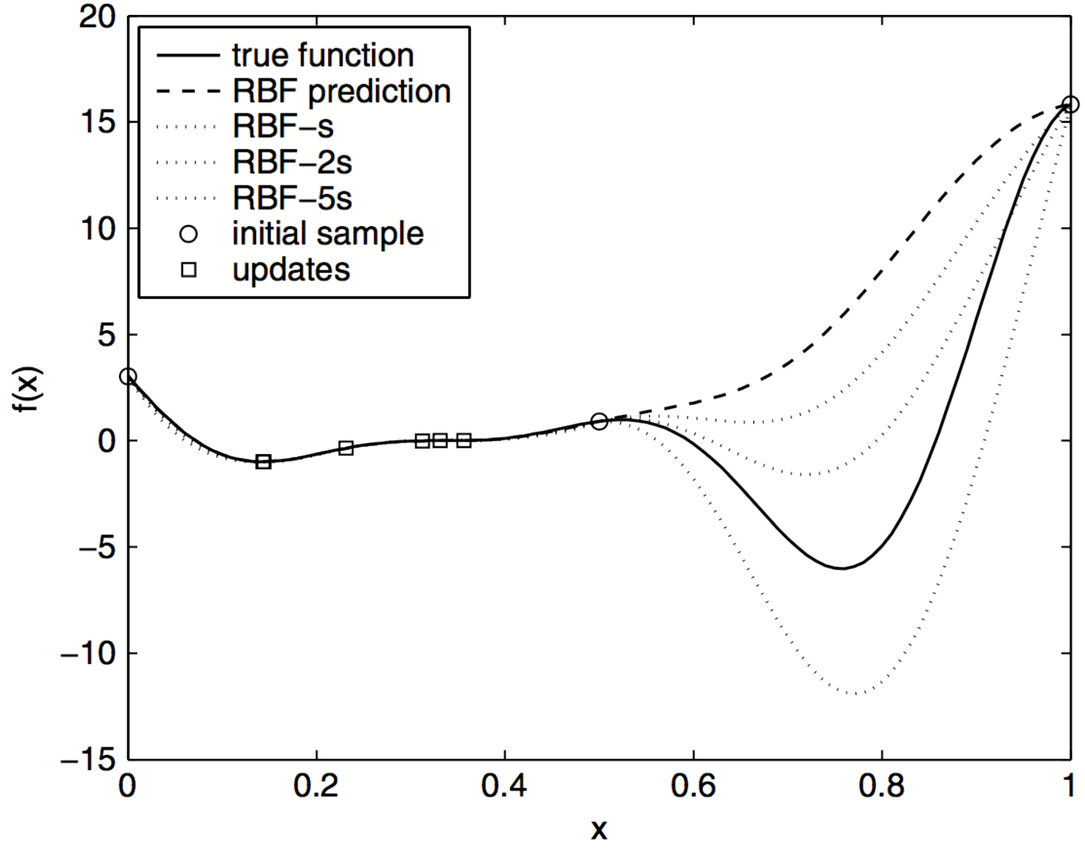
- Gradient-based: Newton-Raphson, BFGS, DFP, conjugate gradient)
- Derivative-free: Generic Pattern Search, genetic, simulated annealing, estimation of distribution (cater to stochastic nature of objective functions, lack speed and exploitation accuracy), Kramer, O., Ciaurri, D. E., & Koziel, S. (2011)

There is no silver bullet

Infill criteria. Exploration vs Exploitation

Dilemma: improve accuracy in the promising region (local exploitation) vs reduce overall uncertainty of the model (global exploration)

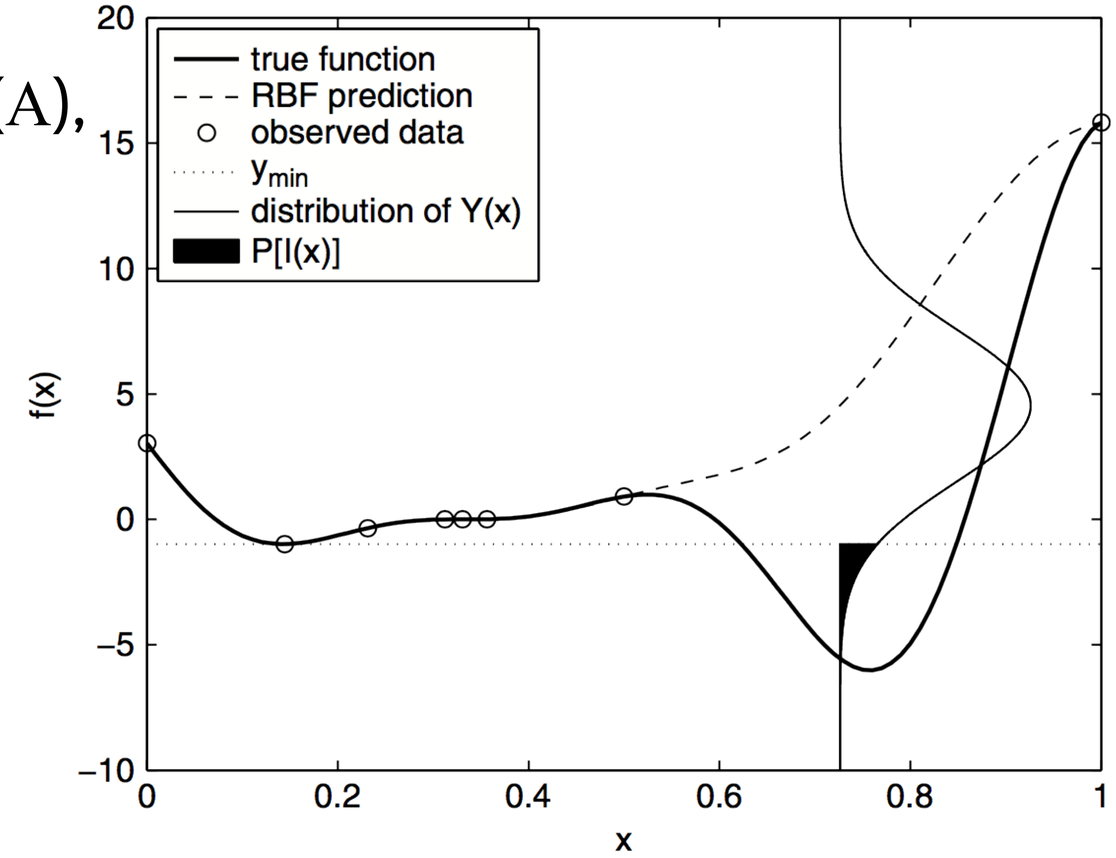
- Idea
  1. define a function over surrogate model output to pick $\boldsymbol{x}_{n+1}$
  2. take into account model (e.g. RBF) uncertainty (deviation) for every $\boldsymbol{x}$: $s(\boldsymbol{x})$
- Statistical Lower Bound
  - $LB(\boldsymbol{x}) = y(\boldsymbol{x}) - A\, s(\boldsymbol{x})$ (y – mean of model, if $A \to 0$, exploitation, if $A \to \inf$, exploration)



$$f(x)=(6x-2)^2 \sin(12x-4)$$

Probability of Improvement

To avoid introduction of new parameters (A), we can estimate probability of getting lower value than ymin for every $\boldsymbol{x}$ (probability of imrovement): $I(x) = y_{min} - Y(\boldsymbol{x})$ (Y – random variable): $P[I(x)]$

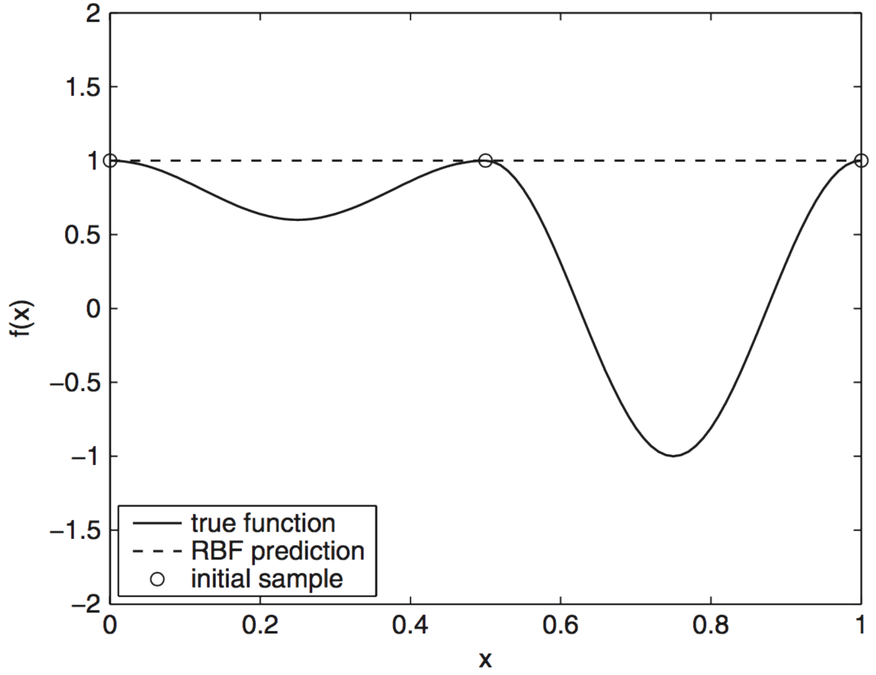


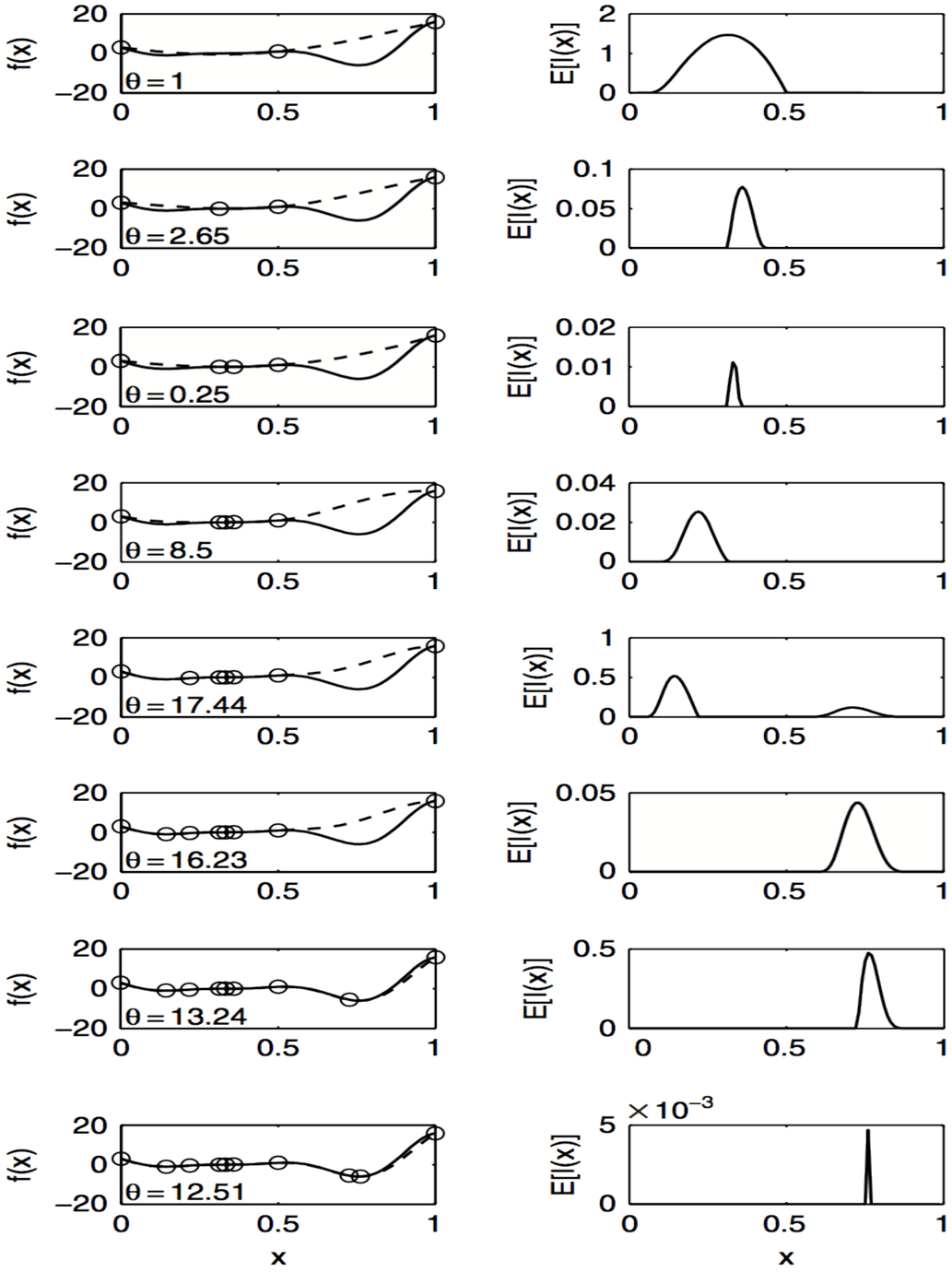
Or even better option:

- Expected Improvement: how much improvement you might get at $\boldsymbol{x}$ $E[I(x)]$

Expected Improvement

- Better balance between exploration and exploitation (a bit slow – 13 steps)
- But there might be traps like this:





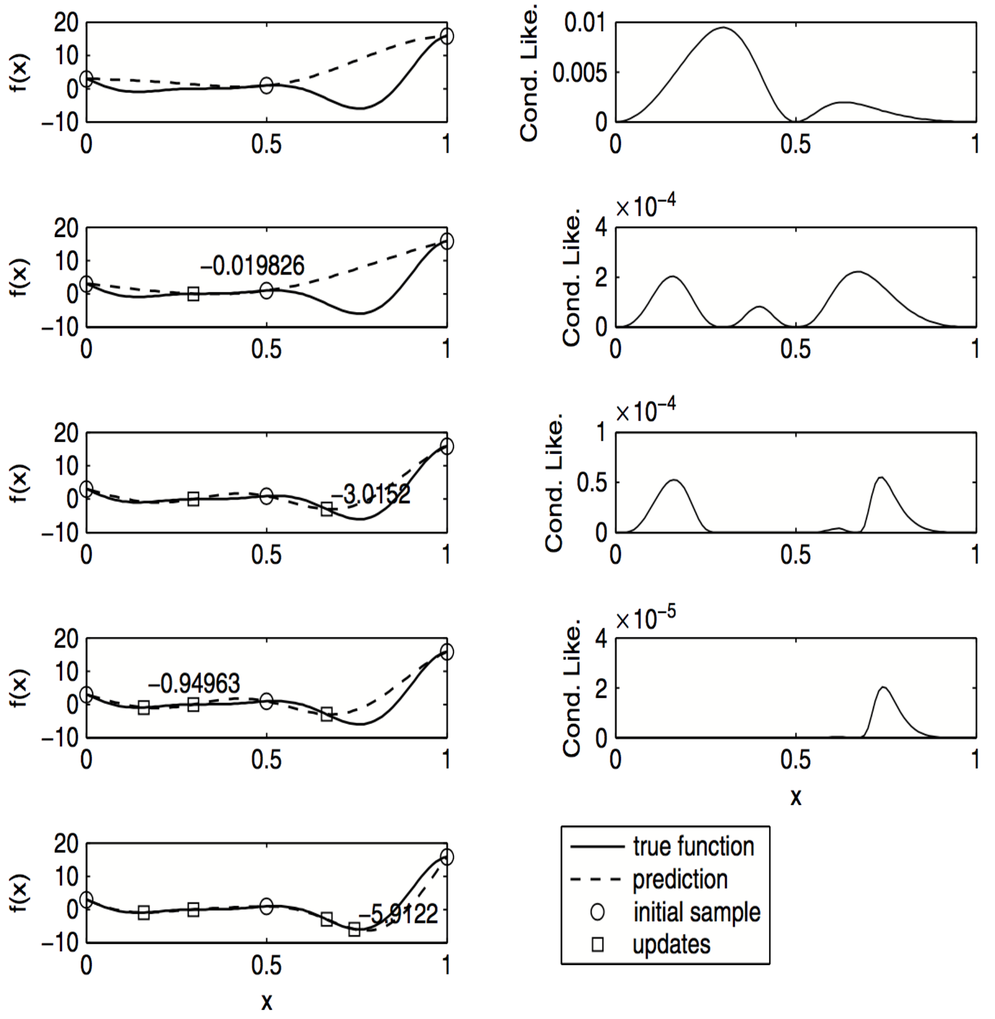
Goal Seeking Strategies

Idea: set goal y_g below known y_{min}, and compute the conditional likelihood of x_g that would give y_g

- Allows for faster convergence (see on the right, it took just 4 steps to reach goal of *-5*)
- Again, to avoid inventing new parameters there is a parameter-less version:

Lower Confidence Bound

- Uses likelihood ratio test to probe for better $\boldsymbol{x}$. For details see Jones and Welch (1996)



Multi-fidelity Simulation

SHiP shield coarse models (cheaper, low-fidelity surrogates, y_c):

- Track only most dangerous particles (e.g. muons)
- Switch off some processes within GEANT simulation
- Compute cost function only for subset of muons (which ones?)

Combine independent surrogates:

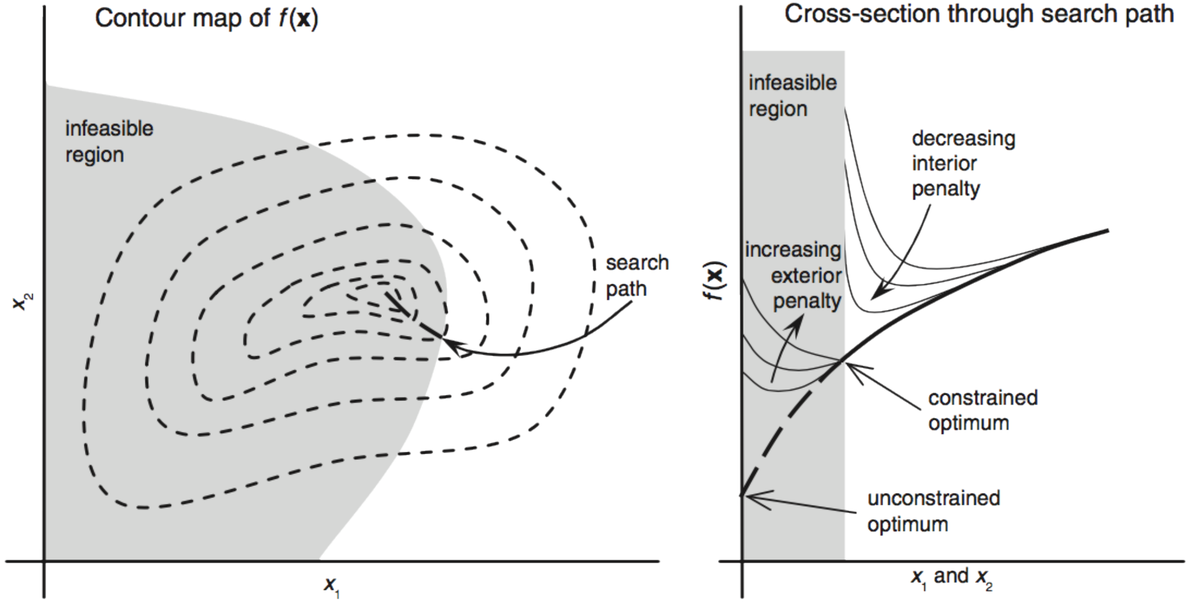
- Produce correction for coarse model response y_c :

  $y_e = Z_p\, y_c$ or $y_e = y_c + Z_d$

- Train regression for either Z_p or Z_d on given set of measurements
- Embed into infill criteria
- Co-kriging – train more complex kriging model on top of $(Y_c, Y_e)^T$ (Forrester et al. (2007))

Constrained Optimization

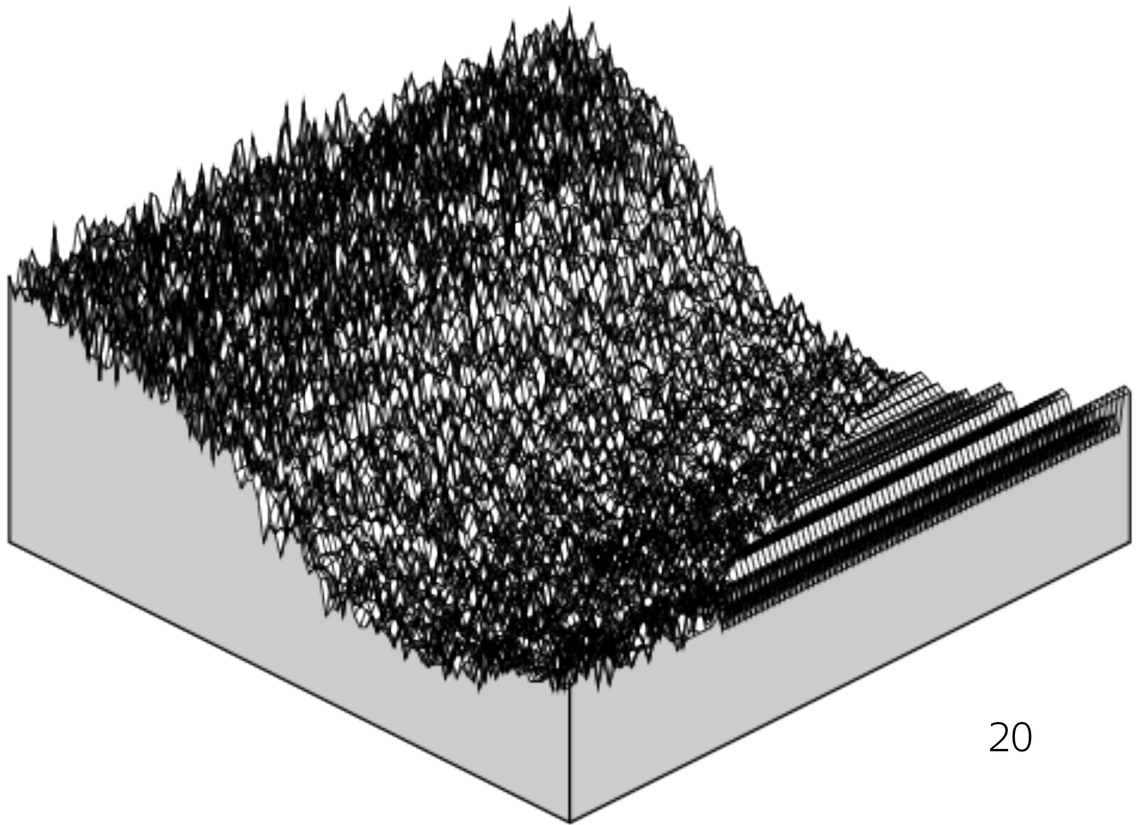
- Example:
  - SHiP shield budget should be < 100M CHF
- Approaches:
  - Add penalty P to cost function if constraint is violated
    $f = f + P$ (no gradient!)
    or
    $f = f + \alpha P$ (α - degree of constraint(s) violation)
  - Embed into infill criteria (e.g. multiply Expected Improvement by probability of constraint satisfaction) – may lead to significant computational improvement
  - Pareto-optimal frontier anlysis (set of points that dominate others), Zitzler,E.et al 2002



Need for Method

We'd better prepare for the following challenges:

- Non-smooth or even discontinuous objective
- Multi-modality (i.e. oscillating wrt features)
- Noise-resistance
- High dimensionality, e.g., $d \gg 1000$
- Constraints (possibly non-smooth, ...)



Evolution Strategy

- Randomized
- Population-based
  - robust to noise
- Rank-based (function-value free)
  - Invariance to strictly monotonic transformations of objective function
- Step size control
  - robust to scale invariance
- Metric learning

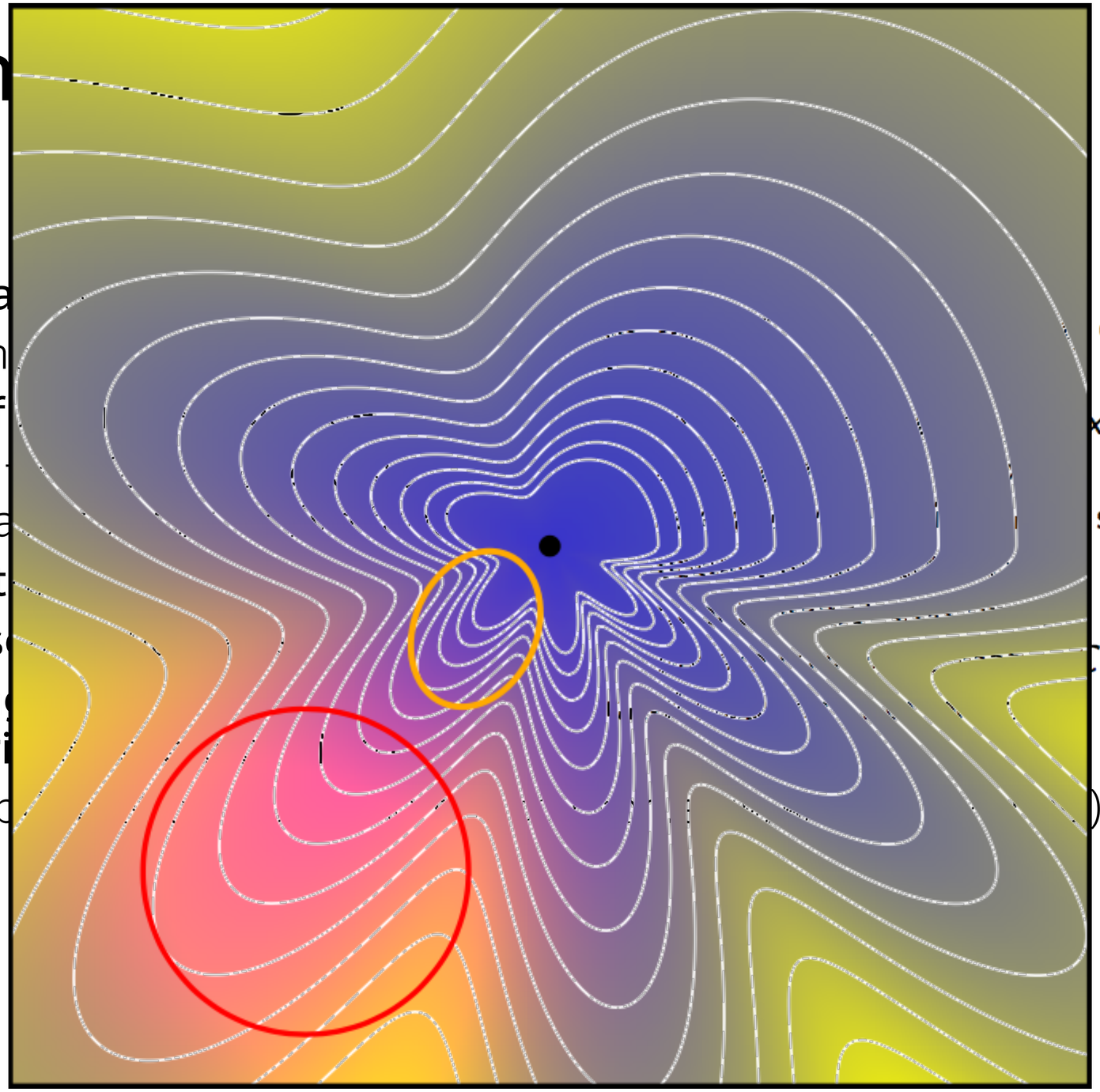
(covariance matrix adaptation, CMA)

  - Efficient optimization of ill-conditioned problems, similar to (quasi) Newton methods

input $m \in \mathbb{R}^d$, $\sigma > 0$, $C(= \mathrm{I}) \in \mathbb{R}^{d \times d}$
loop
  sample "offspring" $x_1, \ldots, x_\lambda \sim \mathcal{N}(m, \sigma^2 C)$
  evaluate $f(x_1), \ldots, f(x_\lambda)$
  select new "population" of size μ (e.g., best offspring)
  update mean vector m
  update global step size σ
  update covariance matrix C
until stopping criterion met

Evolution

- Randomized
  population-ba
  - robust to n
- Rank-based (f
  - Invariance
    transforma
- Step size cont
  - robust to s
- Metric learnin
  (covariance matri
  - Efficient op
    methods



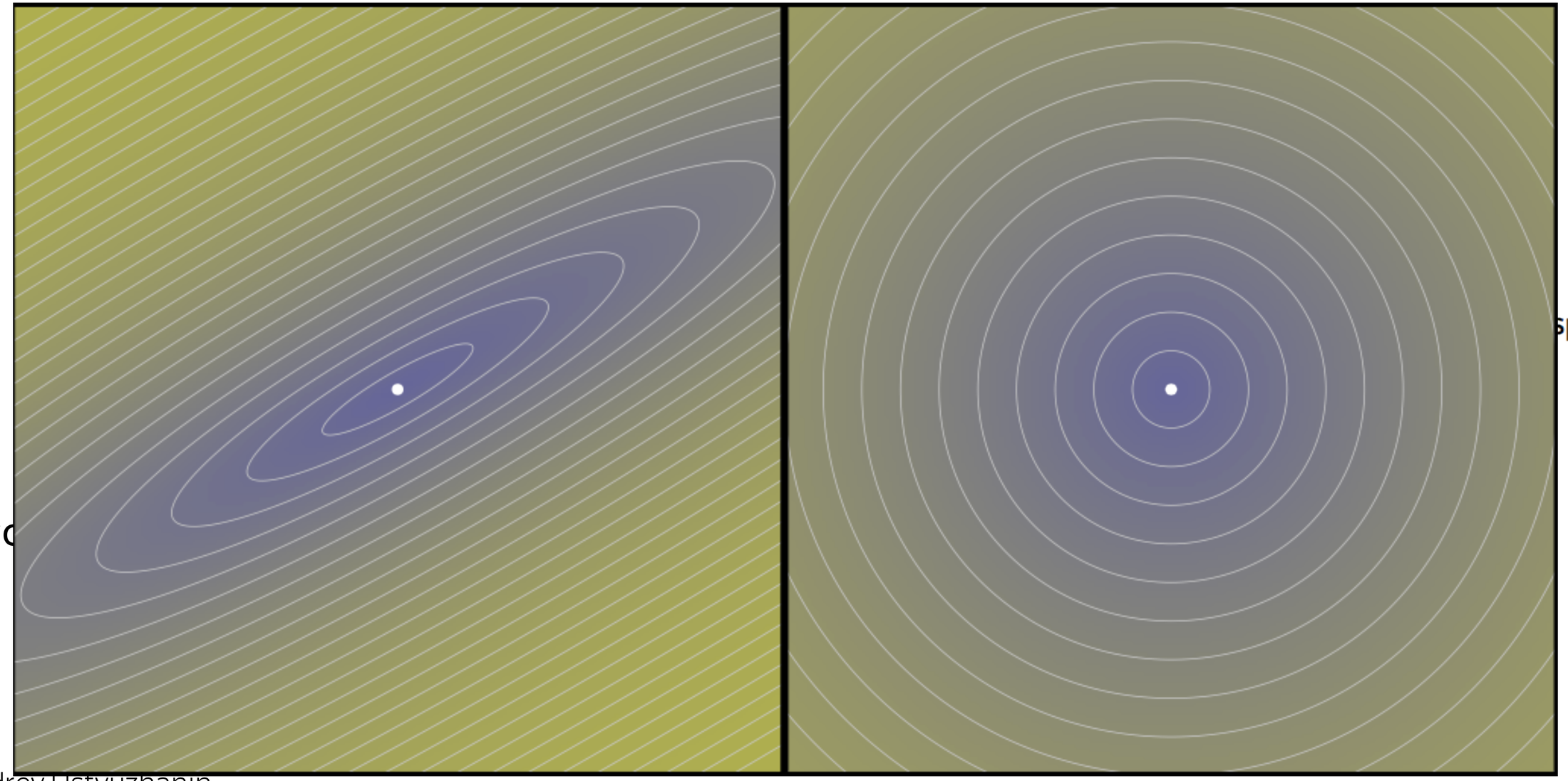
$\in \mathbb{R}^{d \times d}$

$x_\lambda \sim \mathcal{N}(m, \sigma^2 C)$

size μ (e.g., best offspring)

) Newton

Evolution Strategy

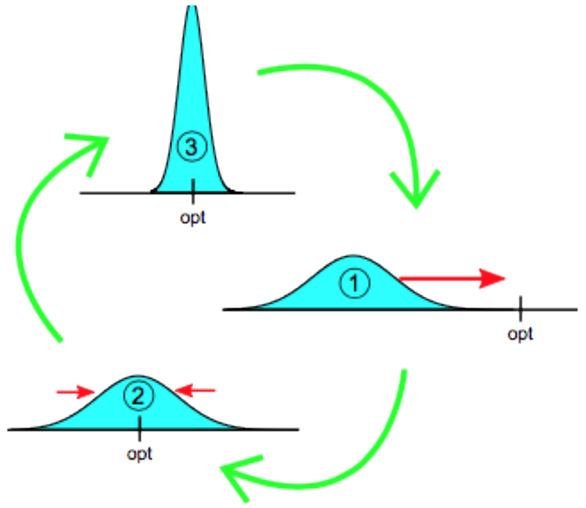


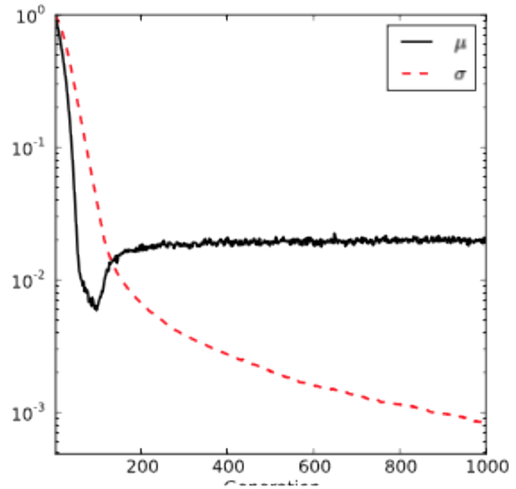
Natural Evolution Strategy

Problem: Given samples and their ranks, how to update the distribution parameters?

Idea:

- change perspective **population** (X) → **distribution** $N(m, C)$ or θ and
- Lift objective function f to W_f: $W_f = E_{x\sim\theta}[f(x)]$

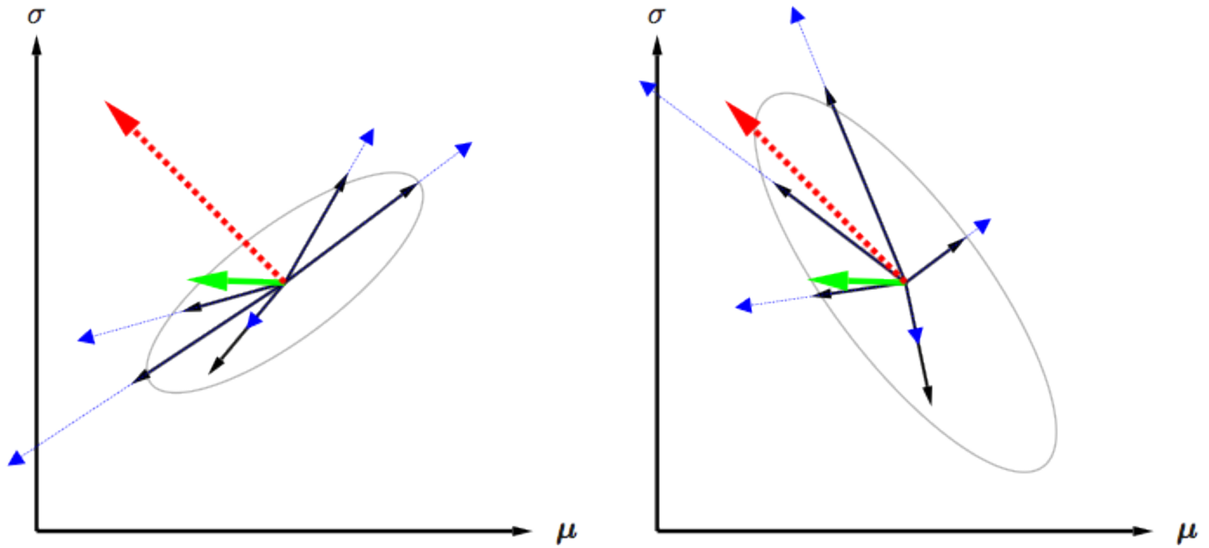


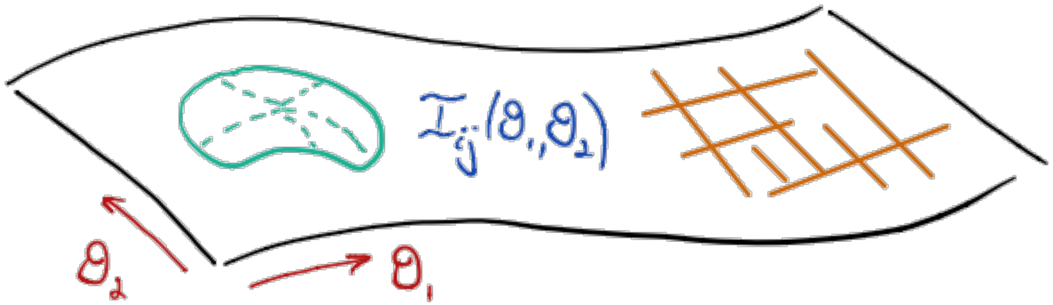


Natural Gradient

When a parameter space has a certain underlying structure, the ordinary gradient of a function doesn't represent its steepest direction, but the natural gradient does

- remove dependence on the parameterization by relying on a more 'natural' measure of distance $D(\theta'||\theta)$ pdfs $\pi(z|\theta)$ and $\pi(z|\theta')$. E.g. Kullback-Leibler
- take into account the patch of θ for the current popluation via Fisher Information Matrix $I(\theta)$





Natural Evolution Strategy

- Plenty of tricks for improvement:
  - Adaptive sampling
  - Importance mixing
  - Restart strategy
  - Exponential NES (Glasmachers et al. Exponential Natural Evolution Strategies. GECCO, 2010)
- See details in Wierstra et al. Natural Evolution Strategies. CEC, 2008 and JMLR, 2014

input $\theta \in \Theta$, $\lambda \in \mathbb{N}$, $\eta > 0$
loop
  sample $x_1, \ldots, x_\lambda \sim P_\theta$
  evaluate $f(x_1), \ldots, f(x_\lambda)$
  $G(\theta) \leftarrow \frac{1}{\lambda} \sum_{i=1}^{\lambda} \nabla_\theta \log \left(p_\theta(x_i)\right) \cdot f(x_i)$
  $\tilde{G}(\theta) \leftarrow \left(\mathcal{I}(\theta)\right)^{-1} \cdot G(\theta)$
  $\theta \leftarrow \theta - \eta \cdot \tilde{G}(\theta)$
until stopping criterion met

Natural Evolution Strategy (NES)

- The resulting algorithm is indeed an ES ($\mathbb{R}^d$ perspective), and at the same time is gradient-based algorithm (θ perspective);
- The parameter space is equipped with the non-Euclidean information geometry of the corresponding statistical manifold of search distributions;
- The SNGD parameter update is by no means restricted to Gaussian distributions. It is a general construction template for update equations of continuous distribution parameters.

Surrogate Model Management

What model do best?

Surrogate optimization?

Surrogate evaluation

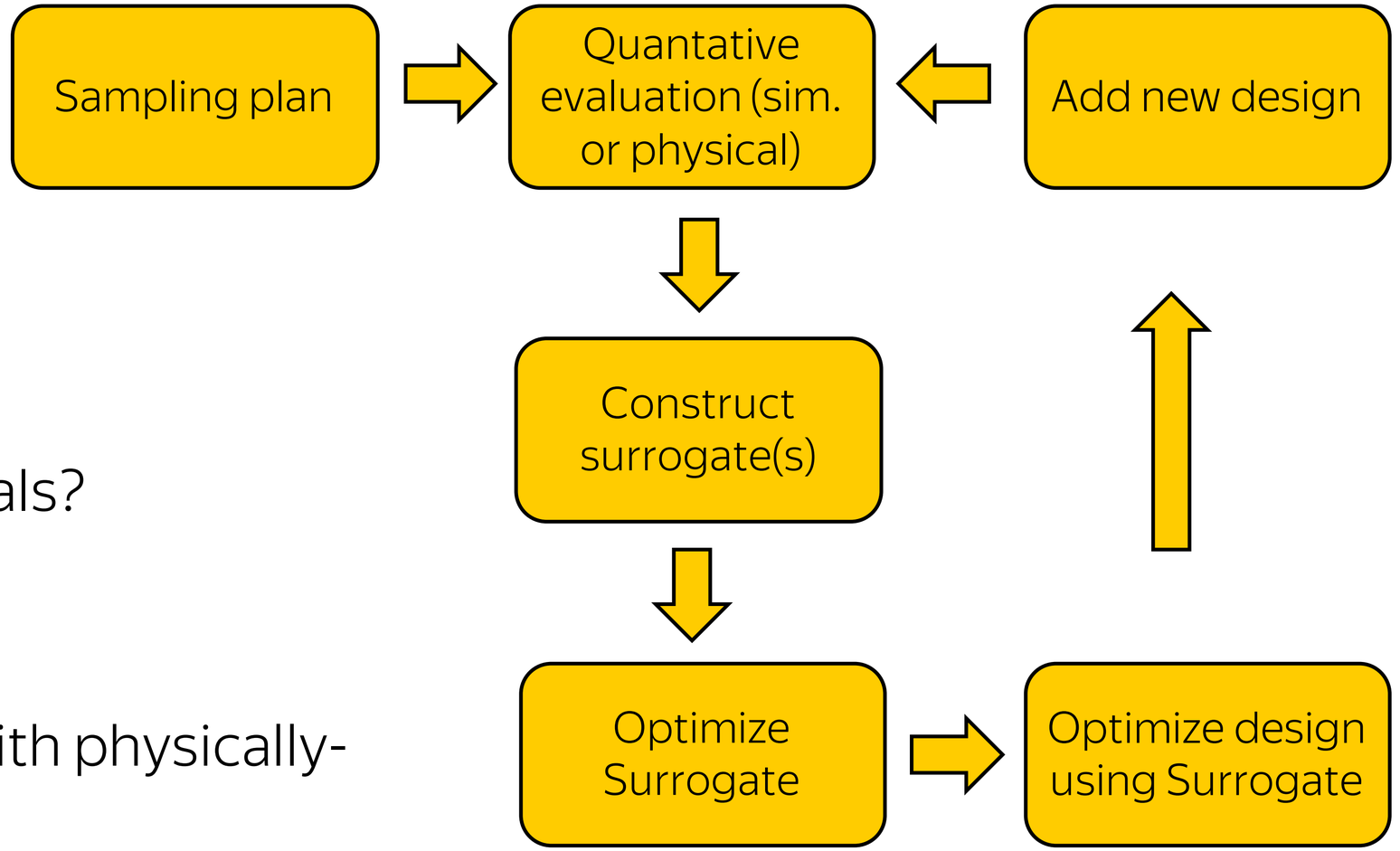
Convergence criteria

Take into account

- Constraints, multi-goals?
- Multi-fidelity?

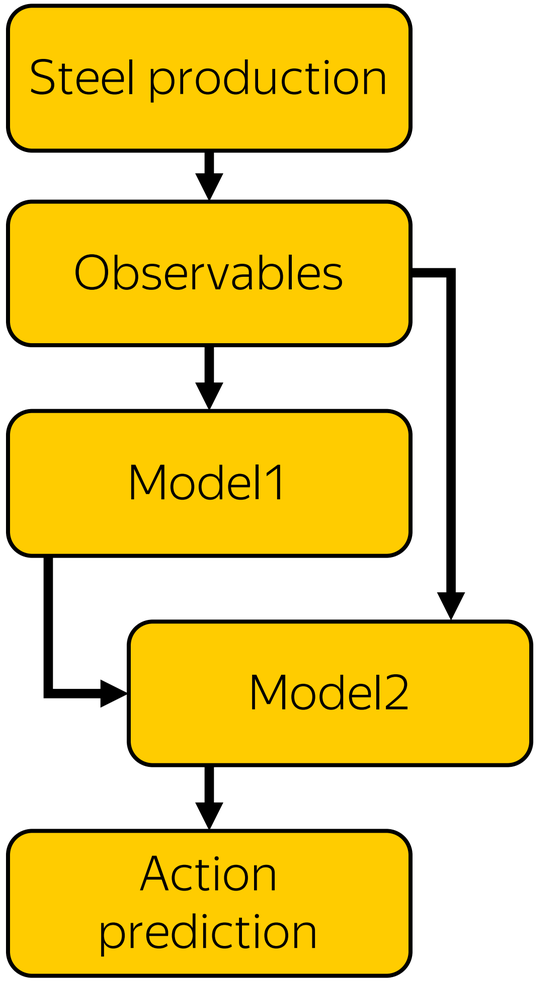
Also:

- Combine surrogate with physically-based model?

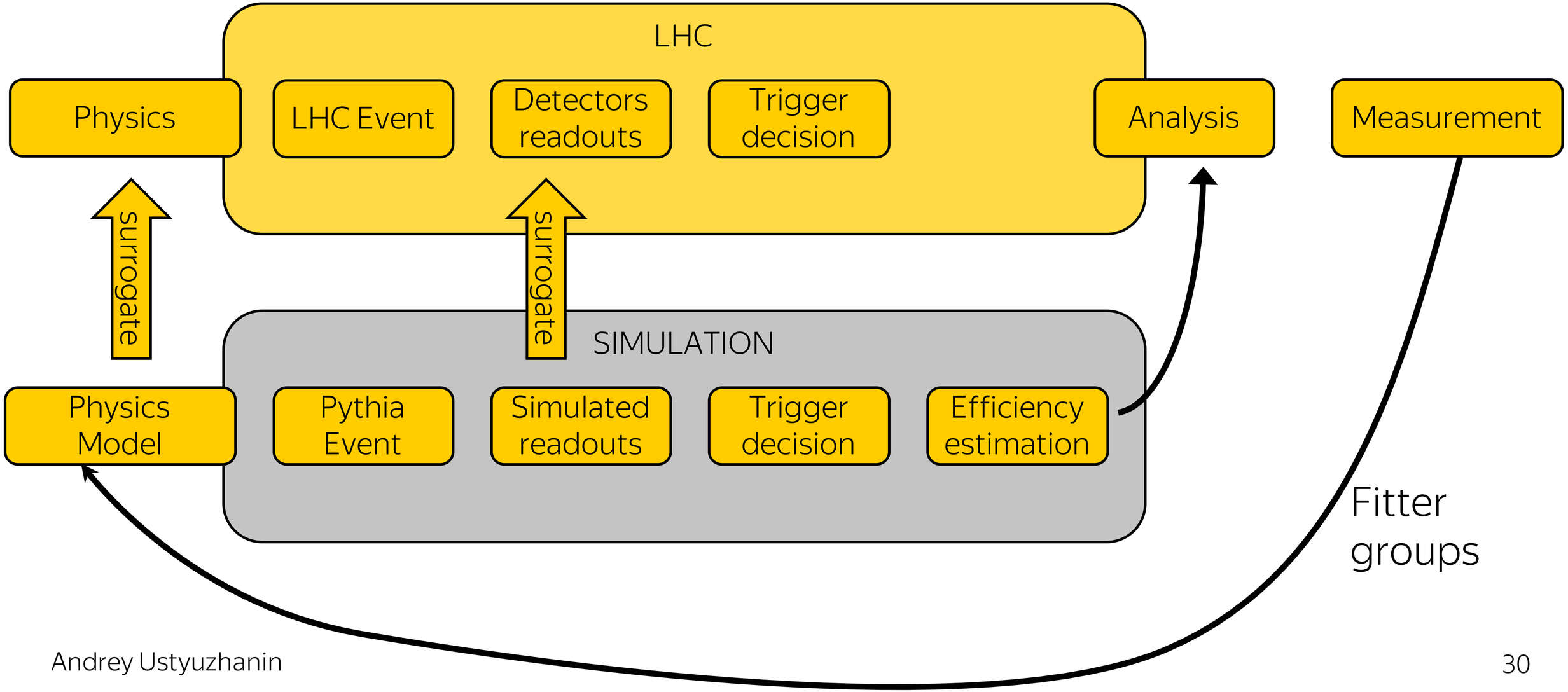


Combine Surrogate with Physically-based model

- **Switching**
  - choose model1 or model2 depending on external conditions (similarity or confidence of prediction)
- **Cascading**
  - train data-driven surrogate model2 on residuals of model1
- **Feature augmentation**
  - feed output and some internal parameters from model1 to model2 along with raw features
- **Two-stage combination**



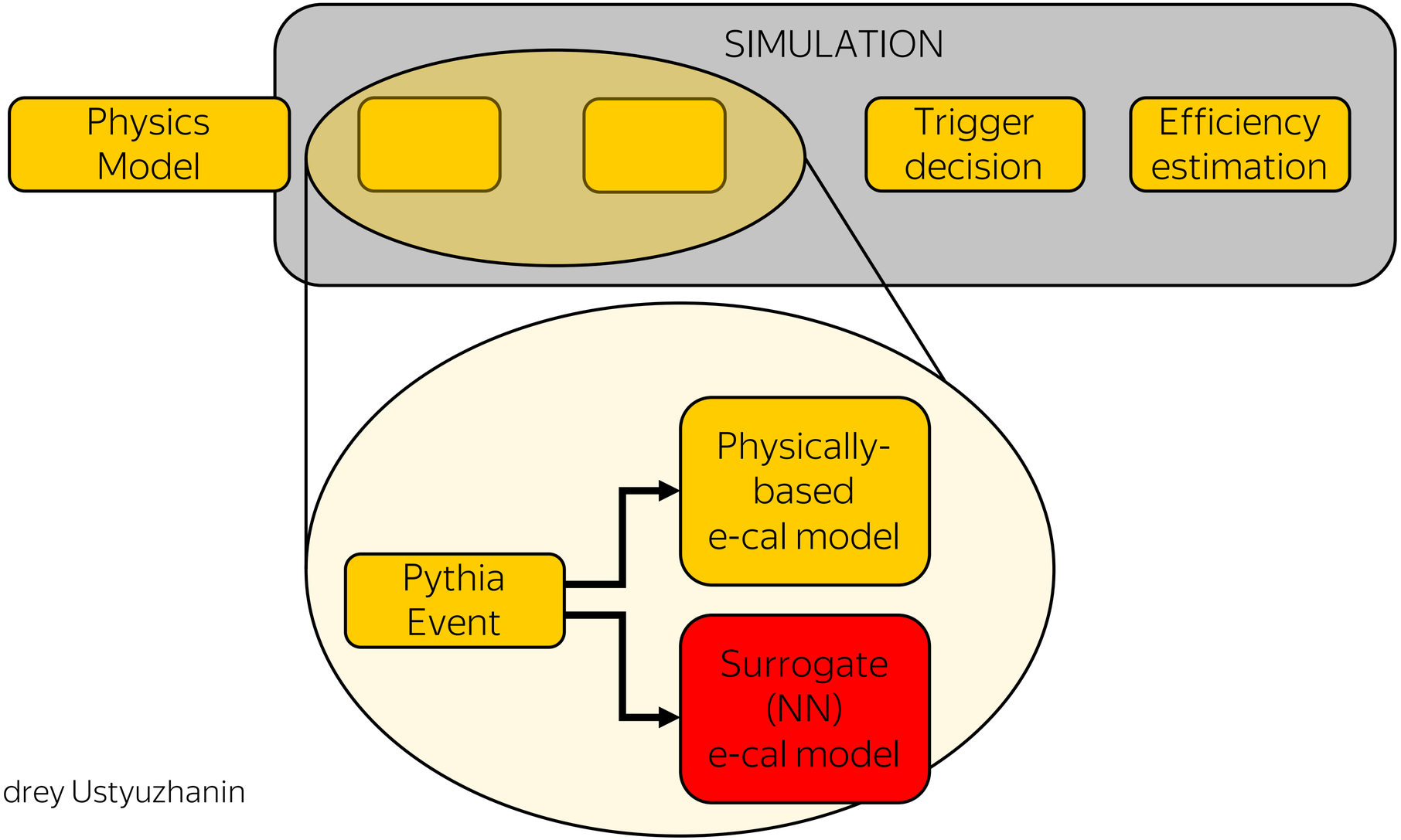
More examples. LHCb e-calorimeter simulation



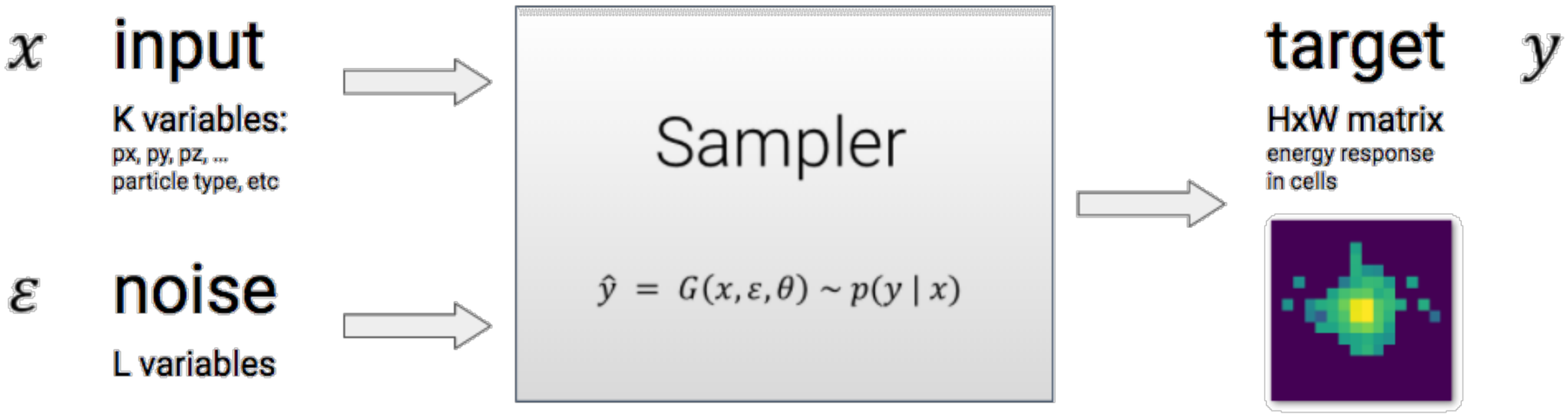
WE NEED TO GO DEEPER

http://bit.ly/2HjJKVQ

More examples. LHCb e-calorimeter simulation

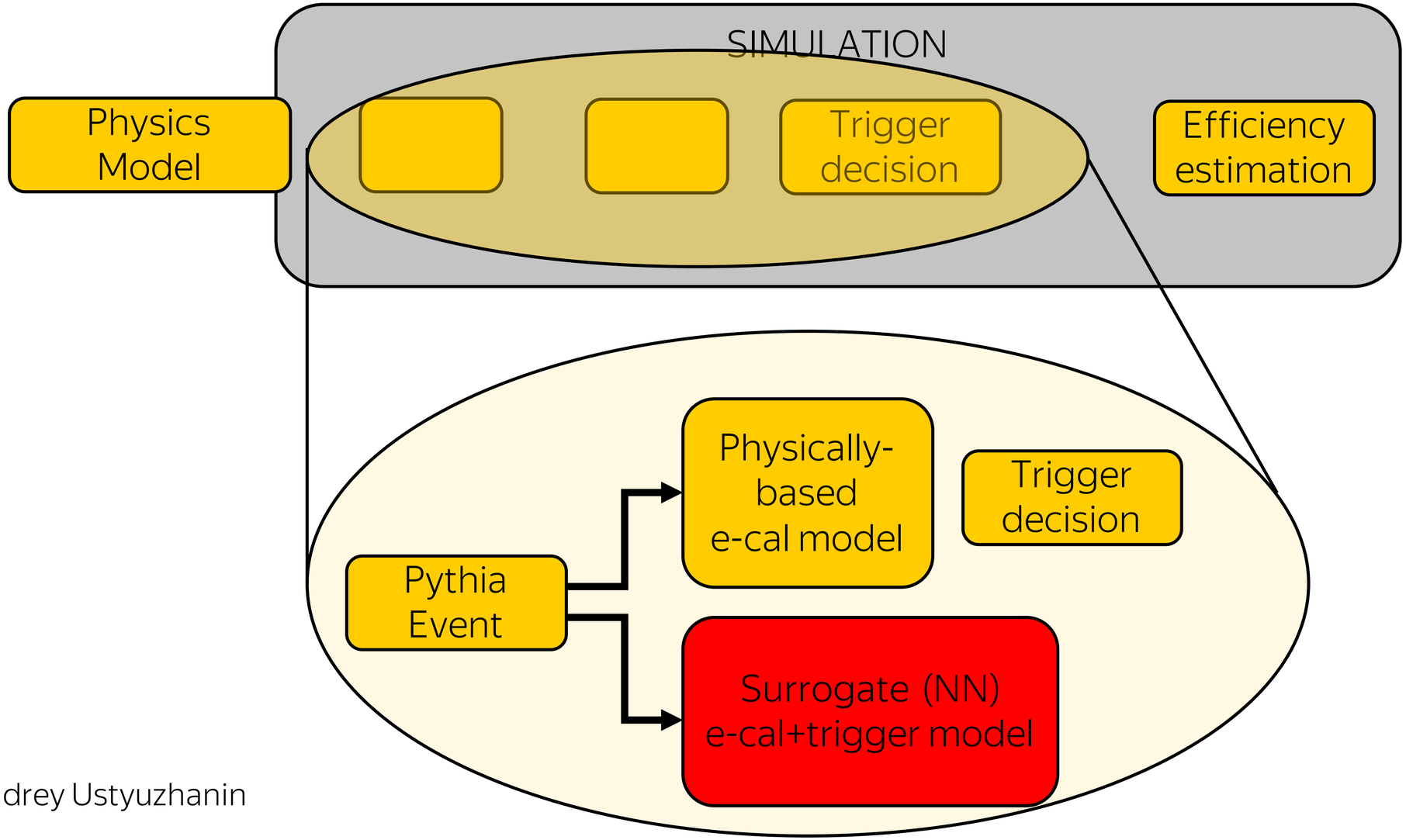


More examples. LHCb e-calorimeter simulation



See talks by Egor and Kamil tomorrow

More examples. Surrogating trigger decisions



Tools & utils

Just a few examples:

- Scikit-optimize - model-based optimization with a `scipy.optimize` interface
  - https://scikit-optimize.github.io/
- Modelgym - gym for predictive models
  - An effort towards automated optimizer, https://github.com/yandexdataschool/modelgym
- Hyperopt - Distributed Asynchronous Hyperparameter Optimization
  - https://github.com/jaberg/hyperopt/
- Gambit – The Global And Modular BSM Inference Tool
  - https://gambit.hepforge.org/tutorials

Tips and Tricks

Reduce dimension (PCA, auto/adaptive encoding)

ES algorithms are usually easy to parallelise

- T. Salimans, J. Ho, X. Chen, and I. Sutskever, Evolution strategies as a scalable alternative to reinforcement learning, arXiv:1703.03864, 2017

Importance Sampling

Reinforcement Learning

Complicated constraint functions approximated by surrogate

- Y. Jin, S. Oh, M. Jeon, Incremental approximation of nonlinear constraint functions for evolutionary constrained optimization, 2010

Optimization of non-differentiable models (simulators)

- Louppe, G., Cranmer, K. (2017)

Open challenges

- Theoretic work
  - Convergence
  - Realistic uncertainty modelling
- Heterogeneous models ensembles
- Surrogate-assisted combinatorial optimization
  - E.g. job scheduling, mobile network optimization
- Surrogate-assisted dynamic optimization
  - E.g. online learning that adapt to moving pareto frontier
- How surrogate models can be used for spotting outliers (anomalies)?
- Thorough benchmarking (http://coco.gforge.inria.fr/)

Conclusion

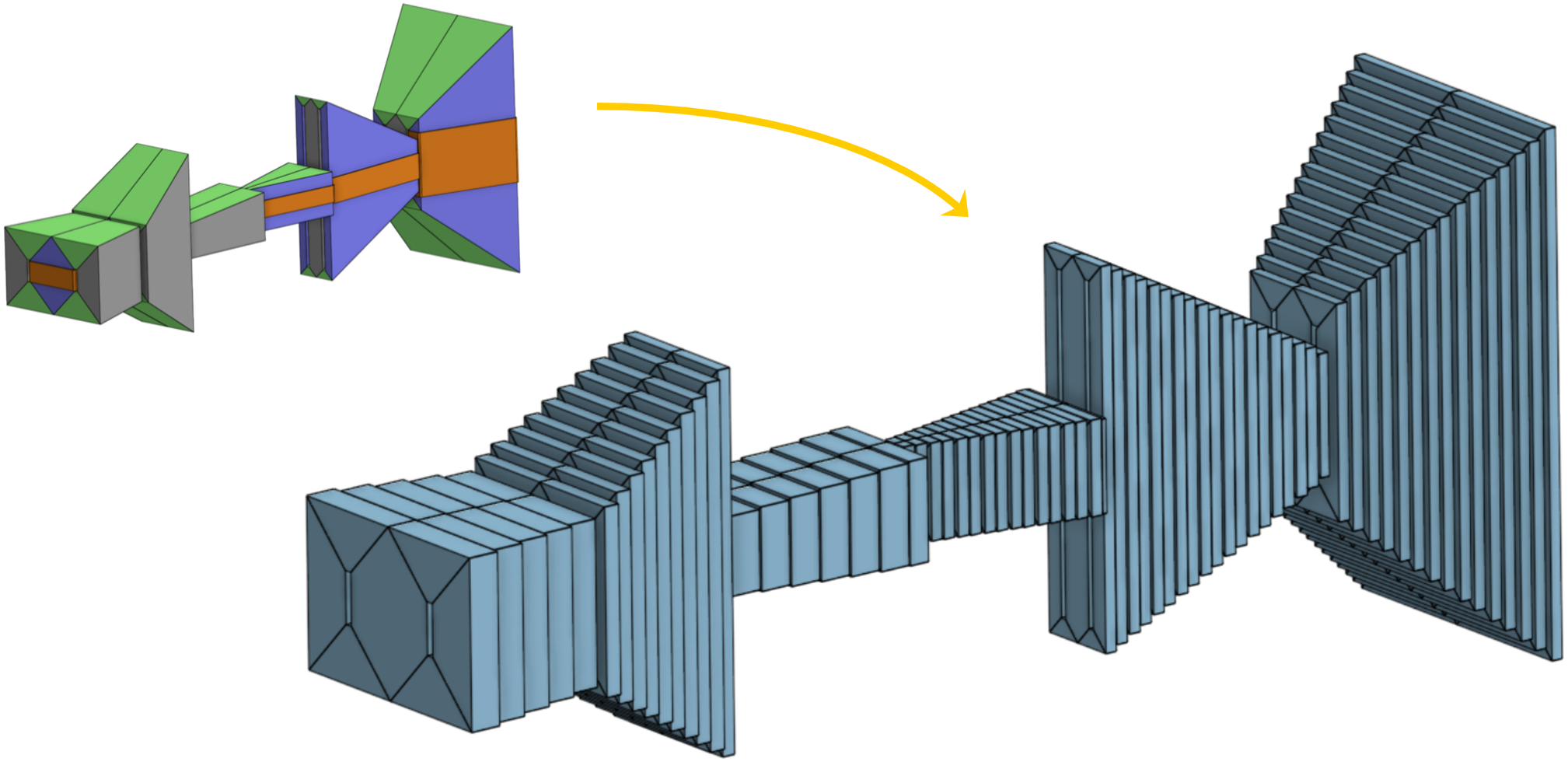
- **Surrogate Modelling is one of the most promising optimization technique**
  - Industry: engineering, control, …
  - Science: detector design, likelihood and uncertainty estimation, …
  - History of science: classical, relativistic, quantum mechanics, QFT
- **Data-driven surrogate methods are developing quickly thanks to Deep Learning**
- **Rich source of ML & Physics challenges**

Illustration from D. Whiteson, J. Cham book

Thank you!

Backup

SHiP shield 2.0



References

Yaochu Jin, Surrogate-assisted evolutionary computation: Recent advances and future challenges, 2011

Jin, 2005; Knowles and Nakayama, 2008; edited volume '*Computational Intelligence in Expensive Optimization Problems*" (Tenne and Goh 2009) (in particular, reviews by Shi and Rasheed, 2009; Santana-Quintero et al., 2009 therein); Razavi et al., 2012 (on surrogate modelling in water resources); Special Issue "*Emulation techniques for the reduction and sensitivity analysis of complex environmental models*" of *Environmental Modelling & Software* (see Ratto et al., 201

Xu et al. (2012, submitted) developed the algorithmic scheme entitled (MOPRISM)

There are many papers comparing various algorithms, see e.g. Tang et al. 2006.

Chow and Yuen [2011] also proposed a history-driven evolutionary algorithm (HdEA)

Mernik M, Exploration and Exploitation in Evolutionary Algorithms: A Survey 2013

Taxonomy of global optimization methods based on response surfaces Jones 2001

History of gaussian processes: https://www.youtube.com/watch?v=4r463NLq0jU

https://wiki2.org/en/BFGS_method+Newton

https://wiki2.org/en/Davidon-Fletcher-Powell_formula+Newton

https://wiki2.org/en/Newton-Raphson+Newton

Smola, A. J. and Schölkopf, B. (2004) A tutorial on support vector regression. Statistics and Computing

Bergstra, J., Yamins, D., Cox, D. D. (2013) Making a Science of Model Search

Kramer, O., Ciaurri, D. E., & Koziel, S. (2011). Derivative-free optimization.

Louppe, G., Cranmer, K. (2017). Adversarial Variational Optimization of Non-Differentiable Simulators arXiv:1707.07113.

Zitzler,E.,et al. Improving the strength Pareto evolutionary algorithm for multiobjective optimization. (2002)