Bayesian Global Optimization and Knowledge Extraction for Active Predictive Analytics

Trent McConaghy
Co-Founder & CTO
solido
DESIGN AUTOMATION

Outline

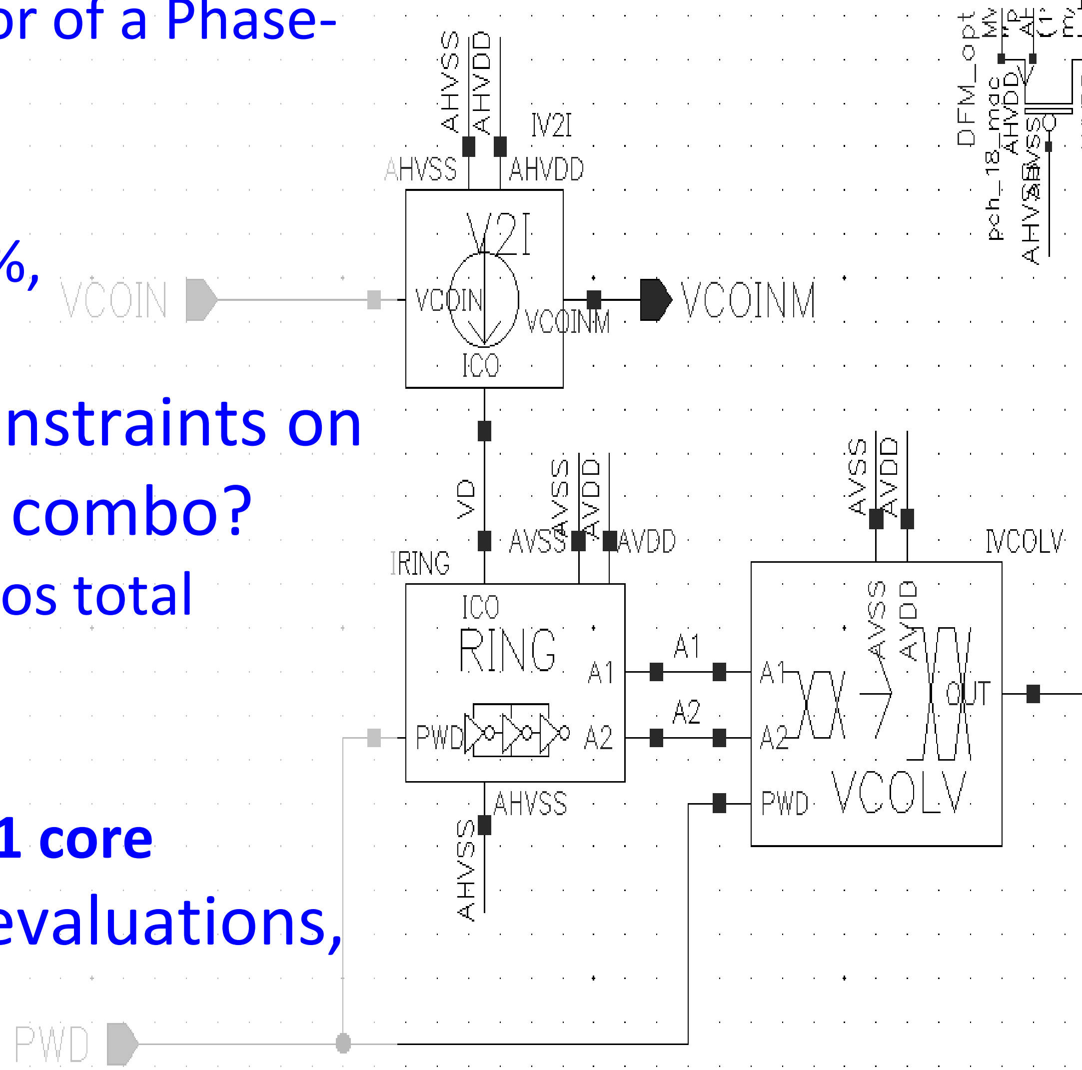
- **Bayesian Global Optimization (BGO)** – “Push designs”
  - Motivating application / what – computer chips
  - Challenges – confidence intervals, scalability
  - Applications – Deep Belief Nets, Ads, Health, more

- **Active Predictive Analytics** – “Pull designs”
  - Fundamental optimization issue – loss of control & insight
  - Using Active Learning for Fast Combination Sweep
  - Using Predictive Analytics / Knowledge Extraction
  - Applications – like opt, plus more

Global Optimization Motivating Example: Circuit Verification

Motivating Example: Circuit Verification

- Circuit: VCO of a PLL
  (Voltage-Controlled Oscillator of a Phase-Locked Loop)
- Constraints:
  - –48.3 < duty cycle < 51.7 %,
  - –3 < Gain < 4.4 GHz/V
- Q: Does circuit meet constraints on every voltage, temp, etc. combo?
  - –3375 “PVT corner” combos total
- Evaluation time:
  - –1 corner: 70 s
  - –**3375 corners: 65.6 h on 1 core**
- Aim: verify with min # evaluations, in min. time

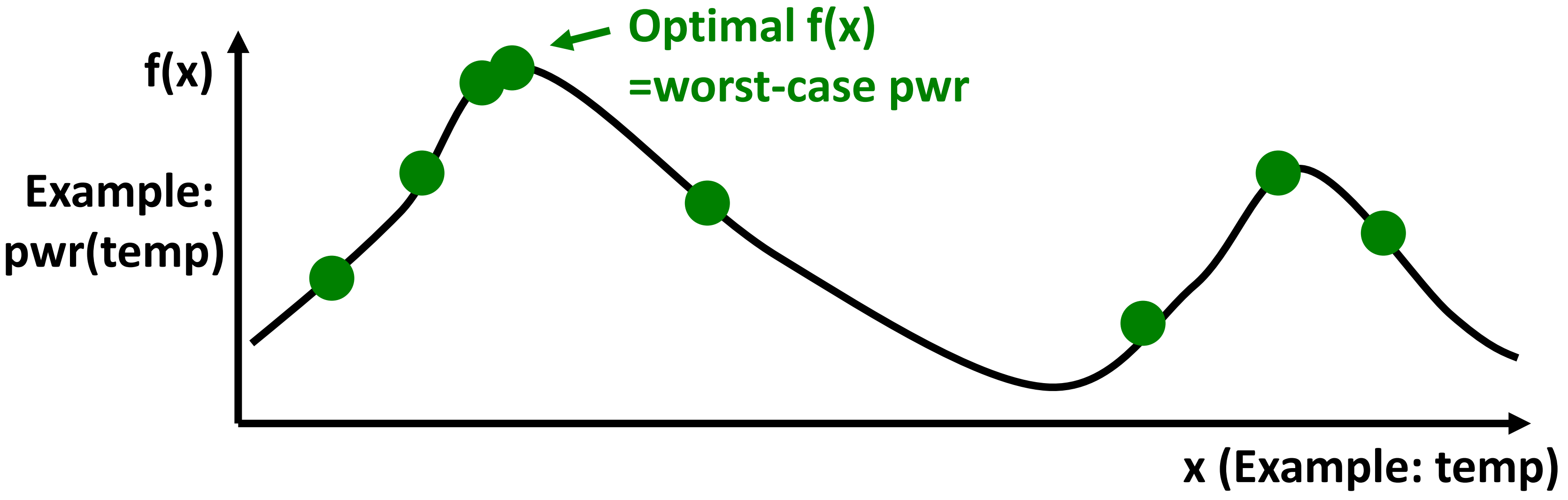


Circuit Verification: General Idea

Cast verification as a global optimization problem:

- Search through space of “corners” - x
- Minimize / maximize simulated output value f(x)

Then, solve the optimization problem reliably.



Circuit Verification: Global Optimization

Cast verification as a global optimization problem:

- Search through space of “corners” - x
- Minimize / maximize simulated output value f(x)

Then, solve the optimization problem reliably.

Attributes of a good global optimizer:

- **Min # evaluations**
- **Reliably finds global optimum**

Options:

- **Deterministic: branch and bound, interval methods, ..**
- **Stochastic: SA, EP/ES, CMA-ES, PSO, ACO, …**
- **Local optimization with restarts**
- **Recast to convex, then Geometric Programming**
- **…**

Circuit Verification: Bayesian Global Optimization

Cast verification as a global optimization problem:

- Search through space of "corners" - x
- Minimize / maximize simulated output value f(x)

Then, solve the optimization problem reliably.

Option: Bayesian [Global] Optimization

Original ref: Jonas Mockus, "On Bayesian Methods for Seeking the Extremum," Optimization Techniques, 1974, pp. 400-404

My fave ref..

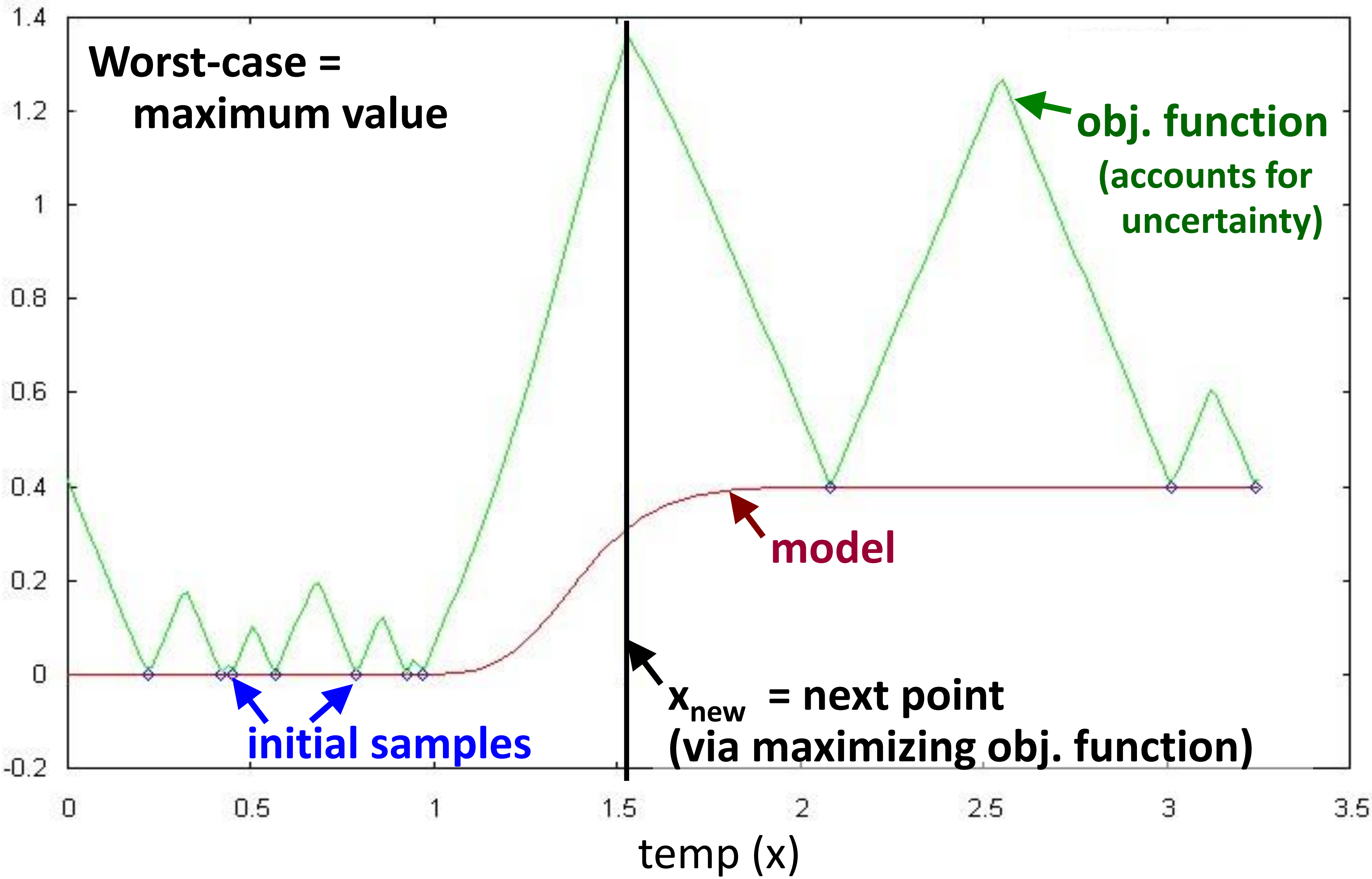
Journal of Global Optimization **13:** 455–492, 1998.

Efficient Global Optimization of Expensive Black-Box Functions

DONALD R. JONES[1], MATTHIAS SCHONLAU[2,*] and WILLIAM J. WELCH[3,**]
[1]*Operations Research Department, General Motors R&D Operations, Warren, MI, USA;* [2]*National*

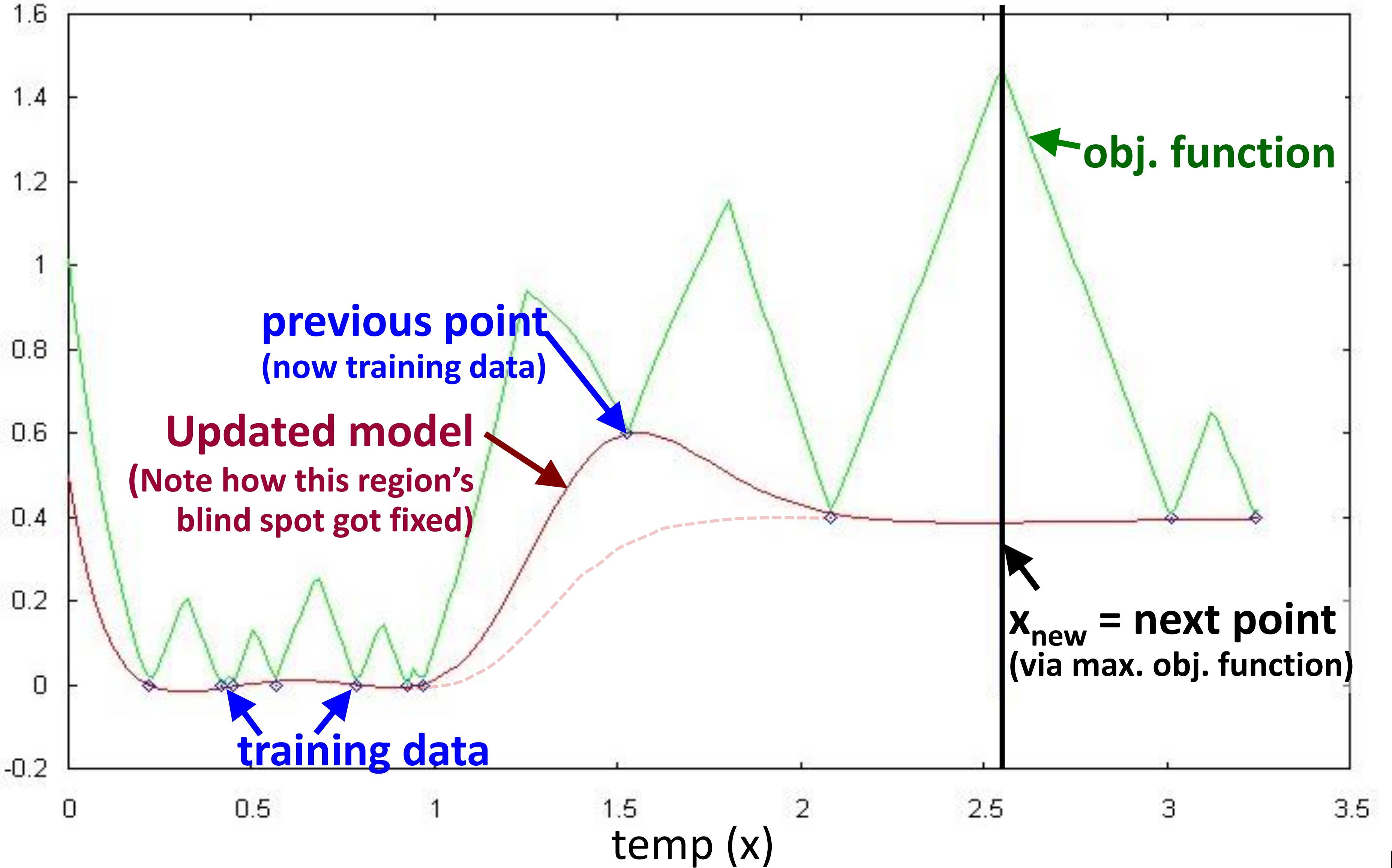
BGO Walk-Through - Step 1

Initial sampling, simulate, build model, choose x_{new}

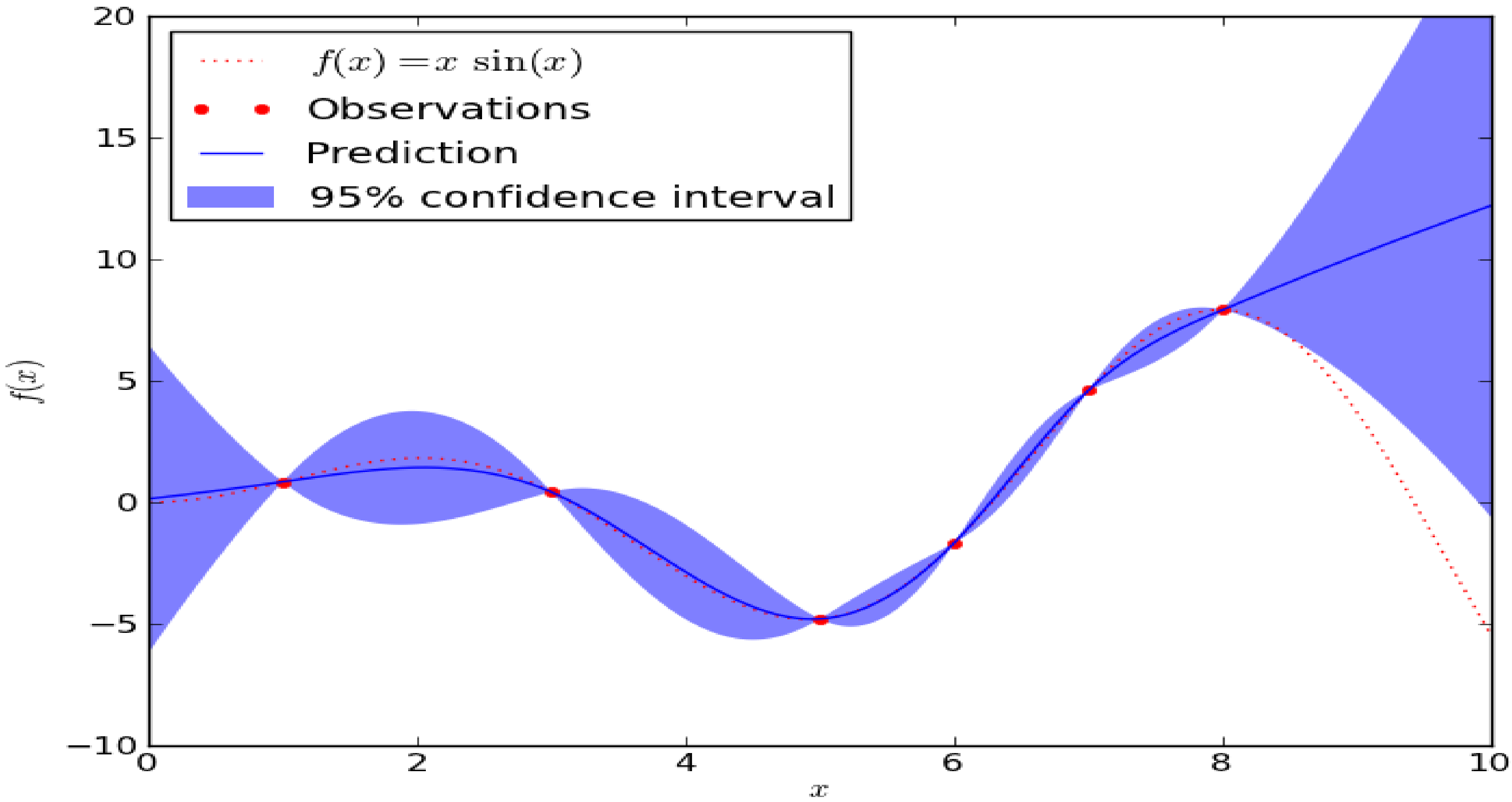


BGO Walk-Through Step 2

Simulate new point, update model, choose x_{new}

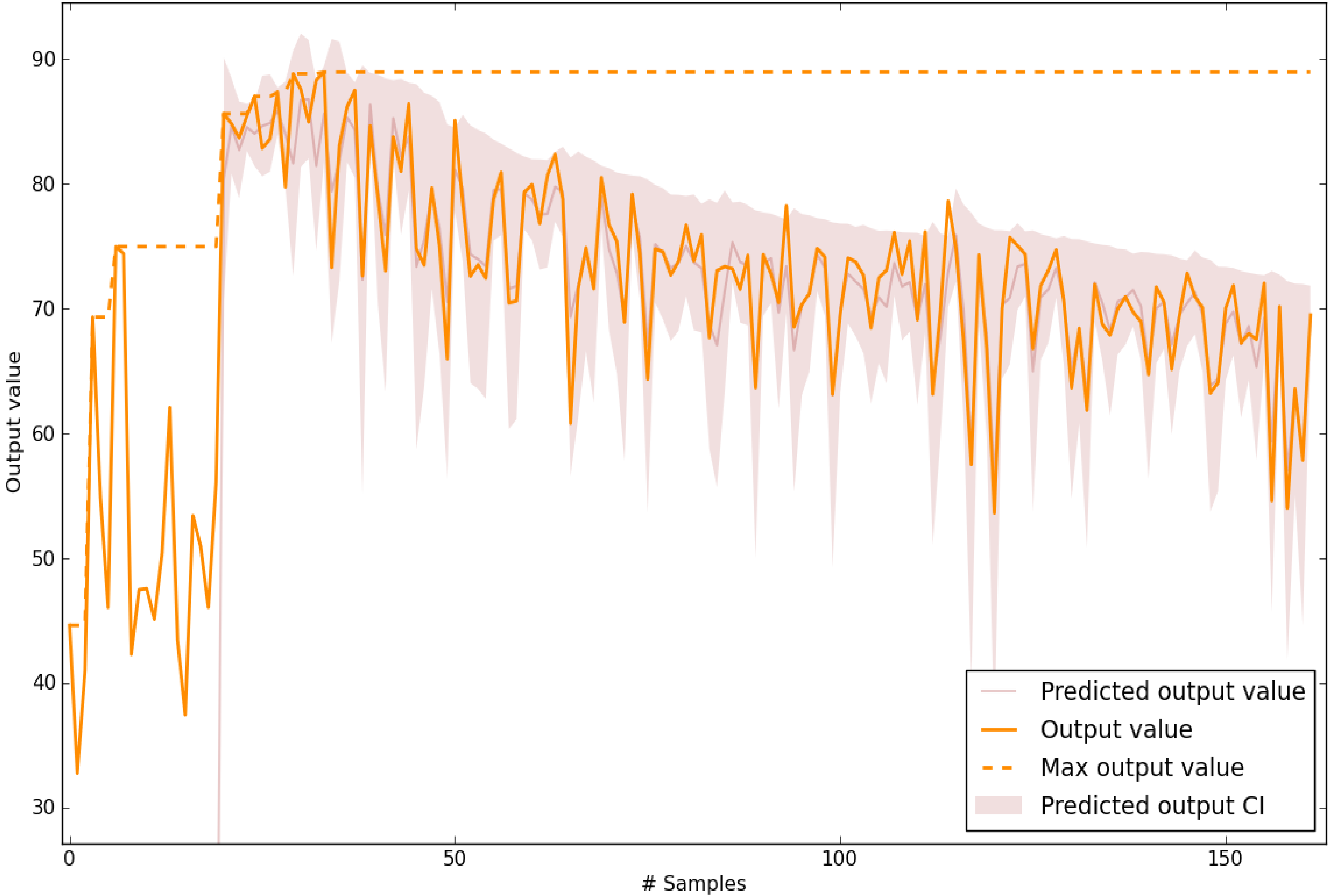


BGO Underlying Model



- **Typically a Gaussian Process Model (GPM)**
  - **Natural interpolator**
  - **Convenient confidence intervals**
  - **Well-behaved, no crazy extrapolation (usually)**

BGO for Circuit Verification – Example Convergence



BGO: When to stop?

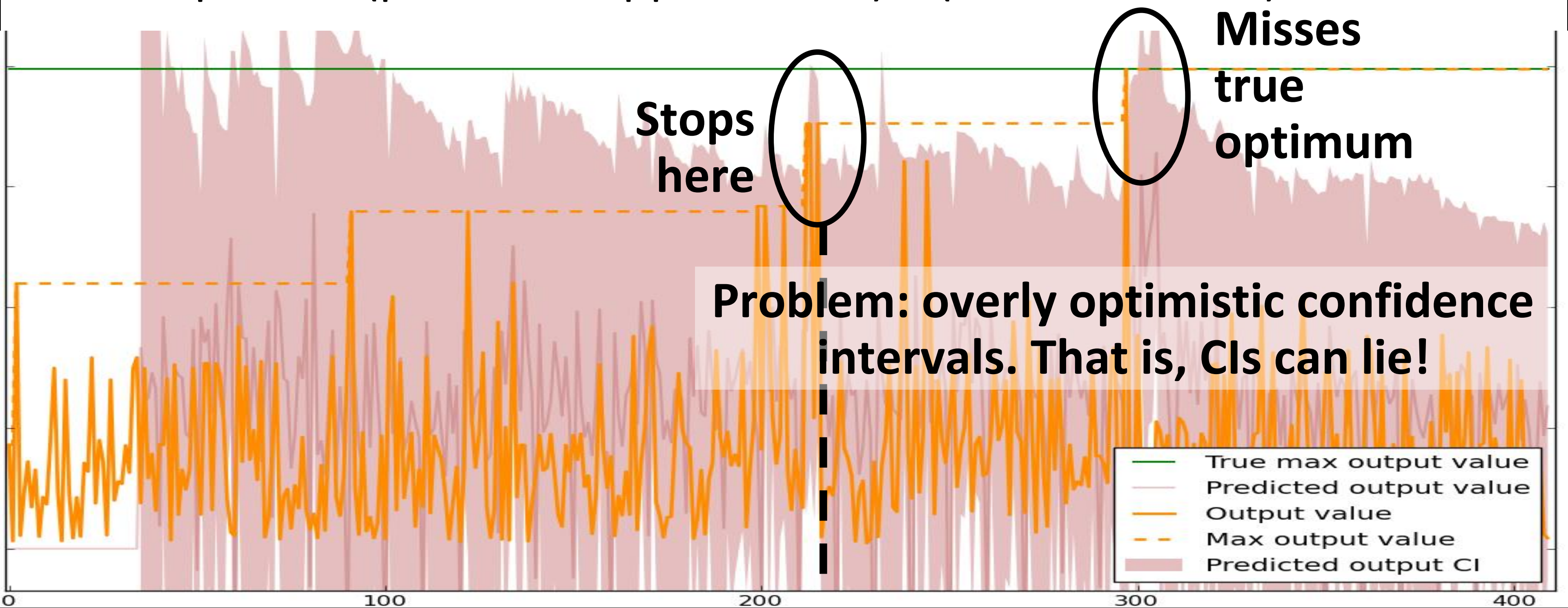
Want min # evals, and to "guarantee" global optimum.

Extreme cases:

- If run all points: guarantees find best, but no speedup
- If run 0 points, it guarantees speed, but never finds best

Baseline approach:

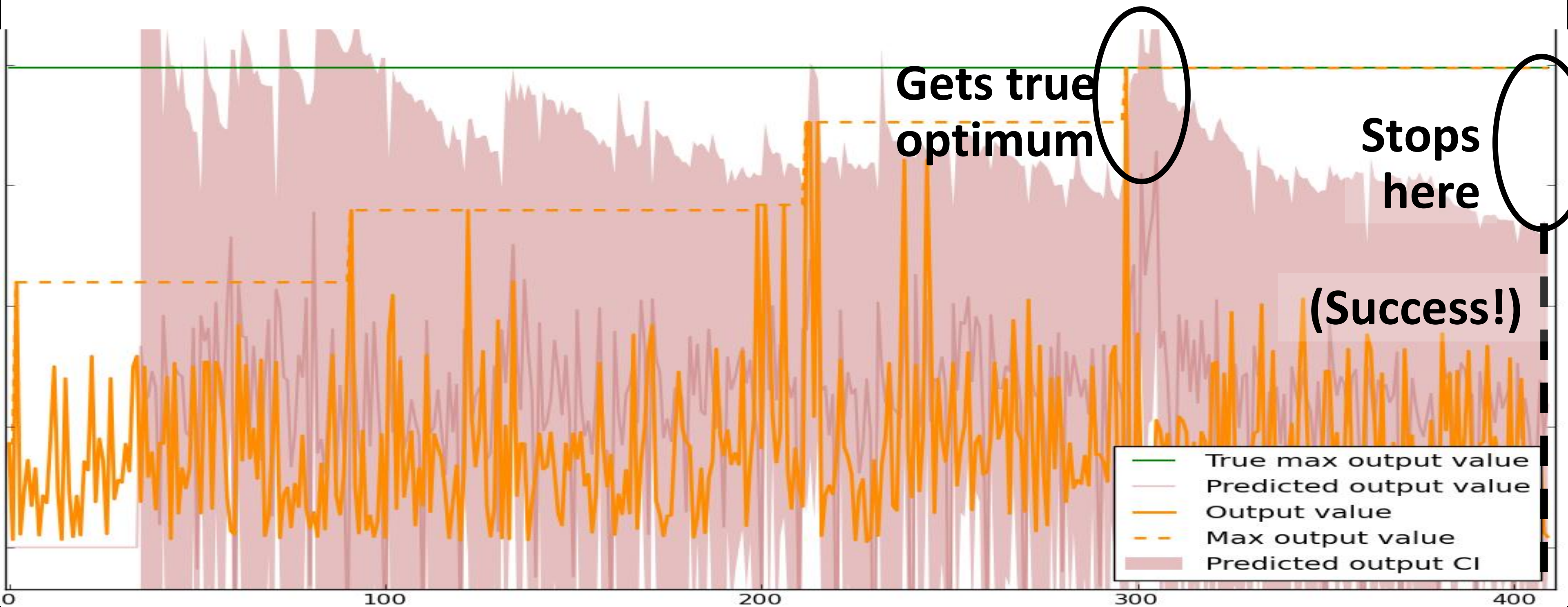
- "Stop when (predicted upper bound) < (max evaluated)"



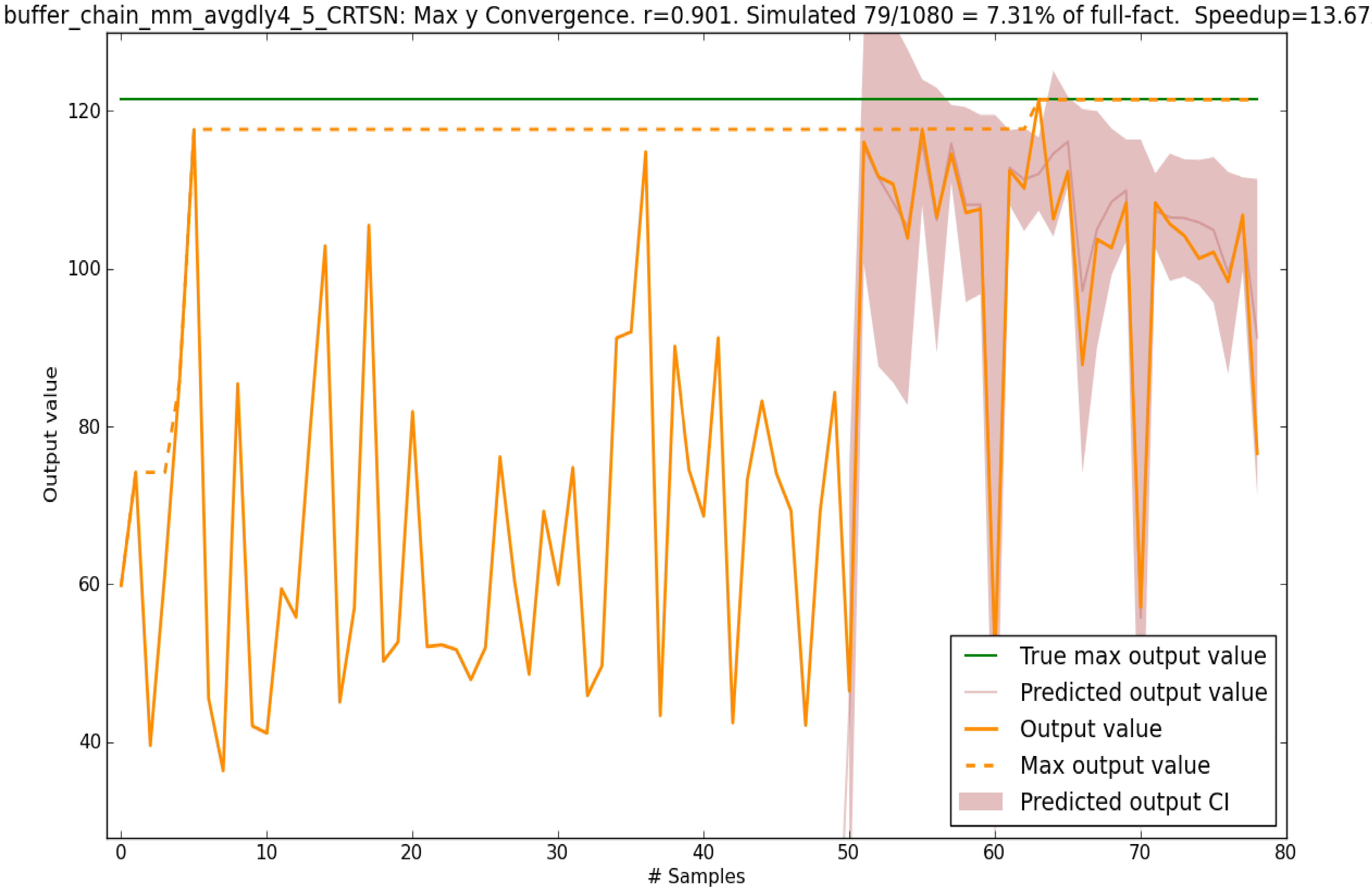
BGO: When to stop?

Revised stopping criterion: "let confidence intervals stabilize"

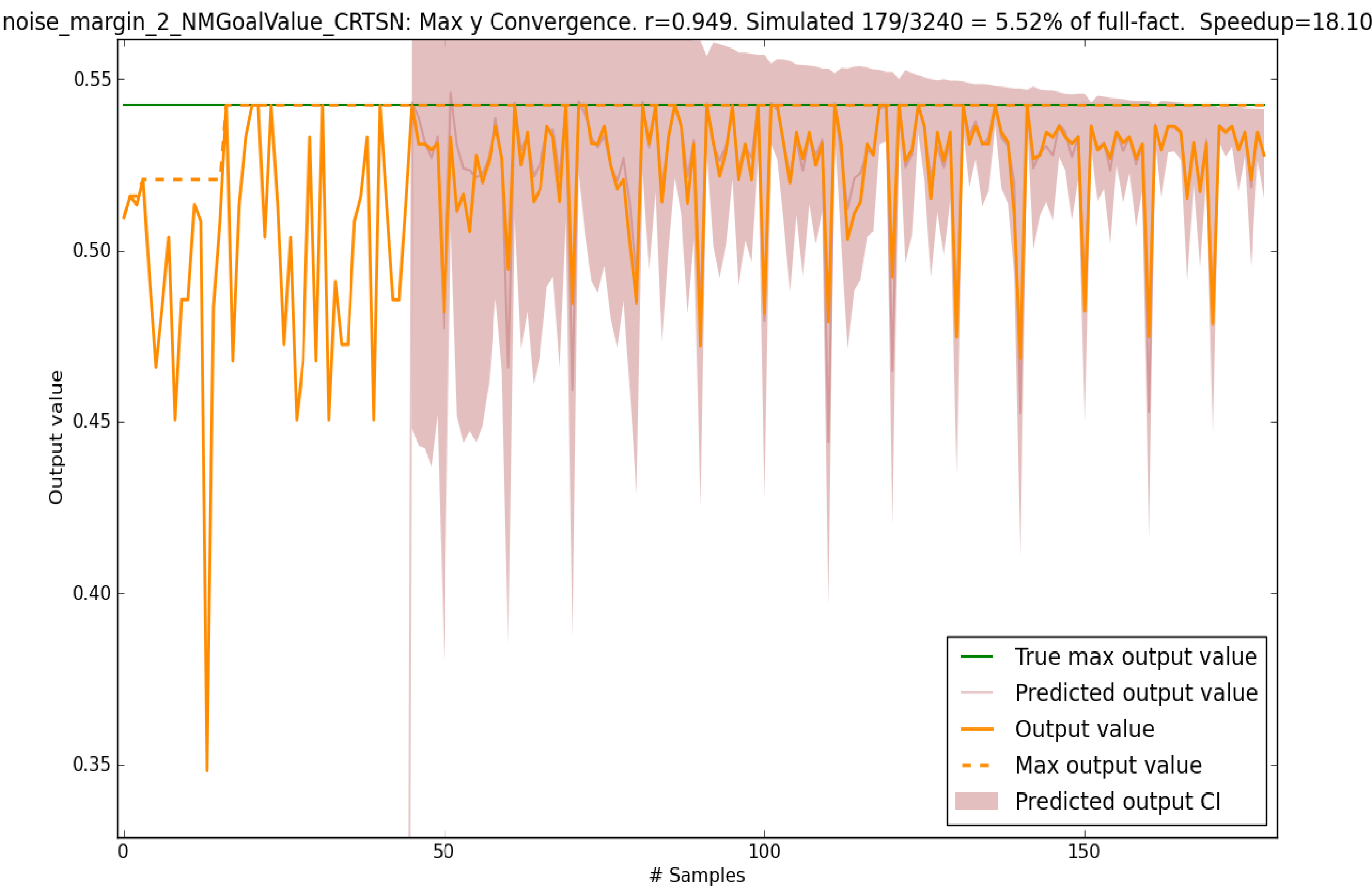
- Stop if (max evaluated value) is constant for ≥ *N1* evaluations;
- and (predicted upper bound) < (max evaluated) for ≥ *N2* evals
  - Where N1, N2 are a function of "prediction difficulty" = (1.0 – rank r)
  - Where rank r is Spearman Rank Correlation between predicted & actual



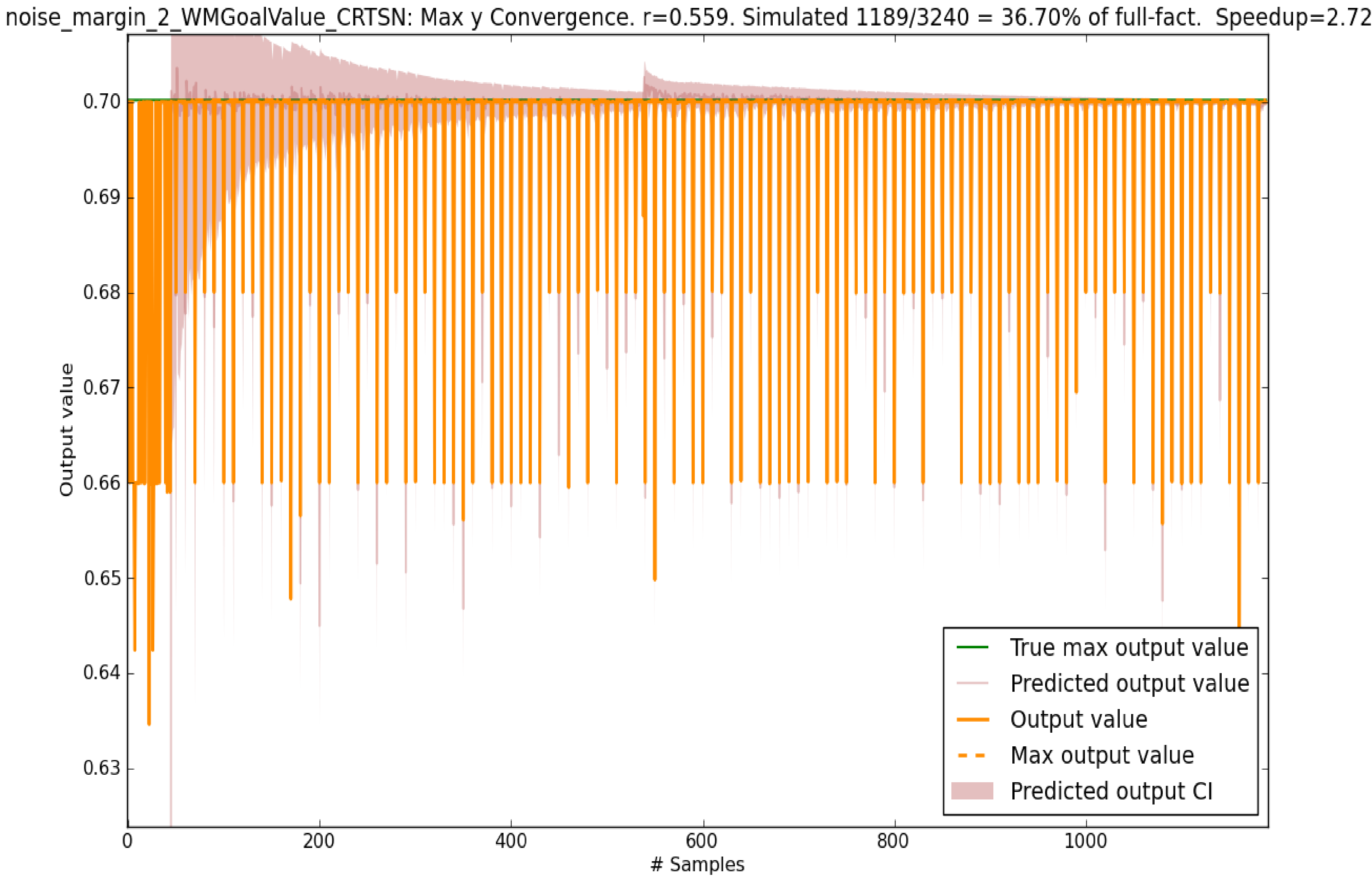
BGO: Representative Convergence Curve
-most look like this



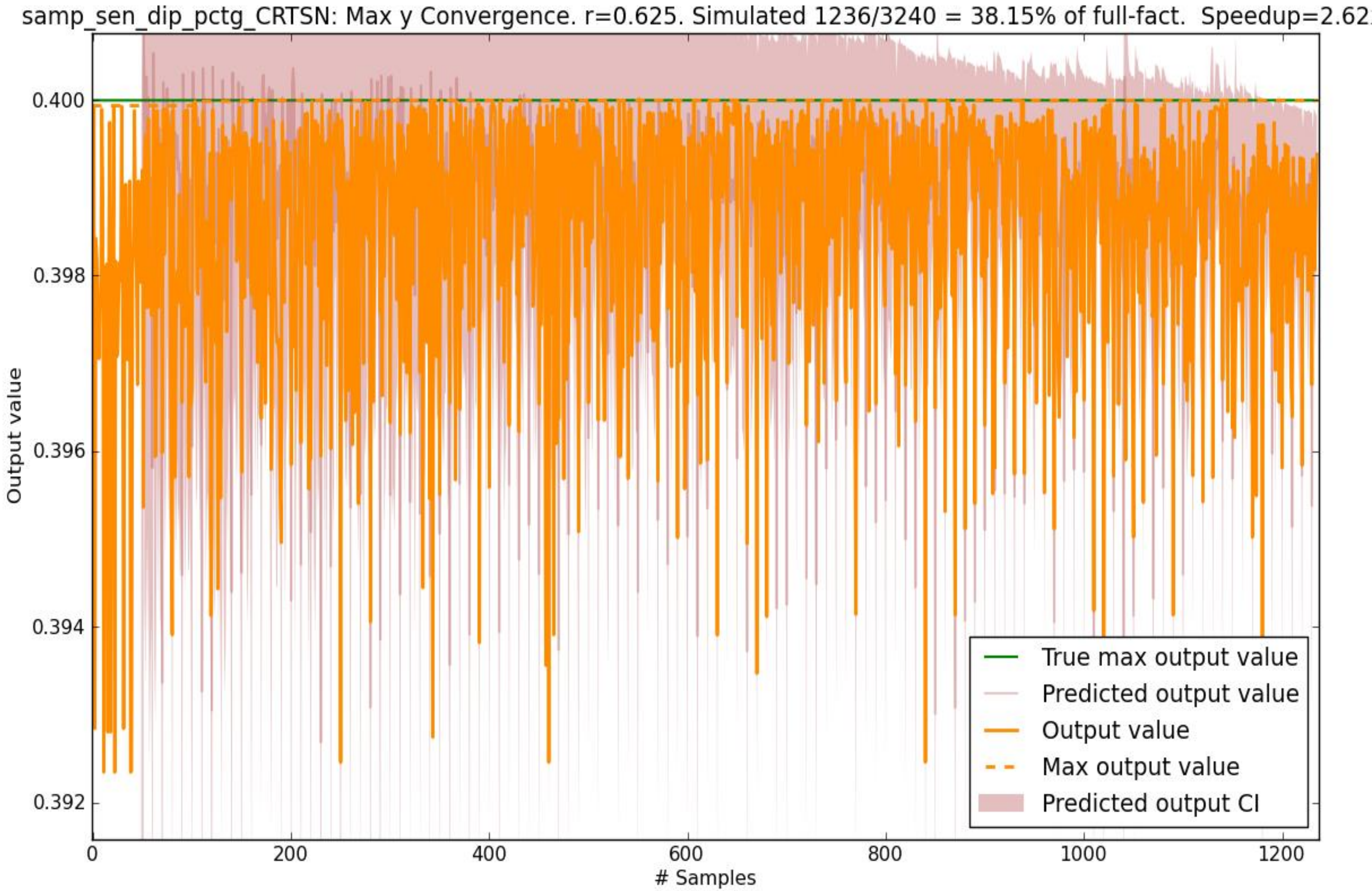
BGO: Representative Convergence Curves
-some look like this



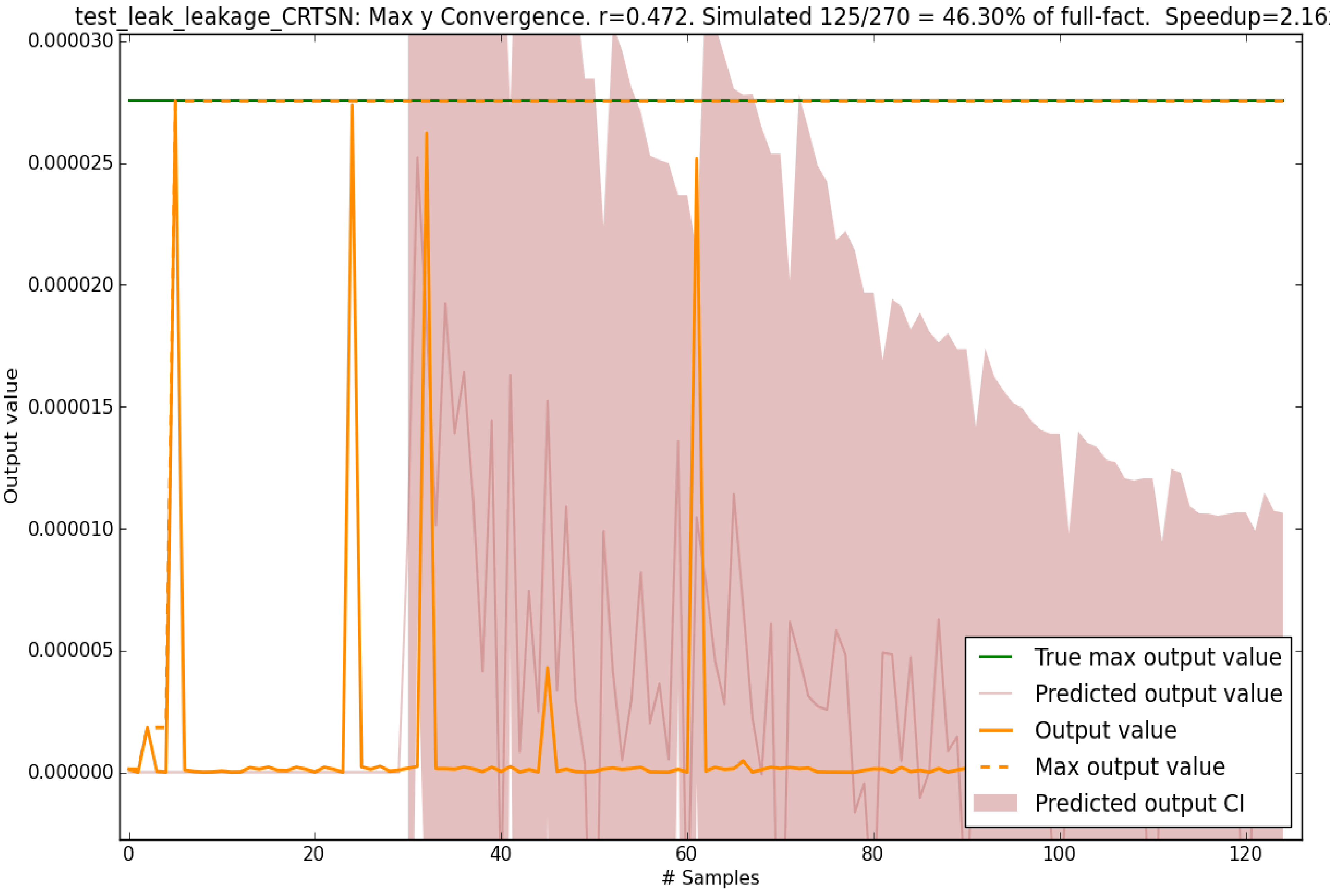
BGO: Convergence Curves
-even “difficult” curves converge



BGO: “Difficult” Convergence Curve



BGO: "Difficult" Convergence Curve

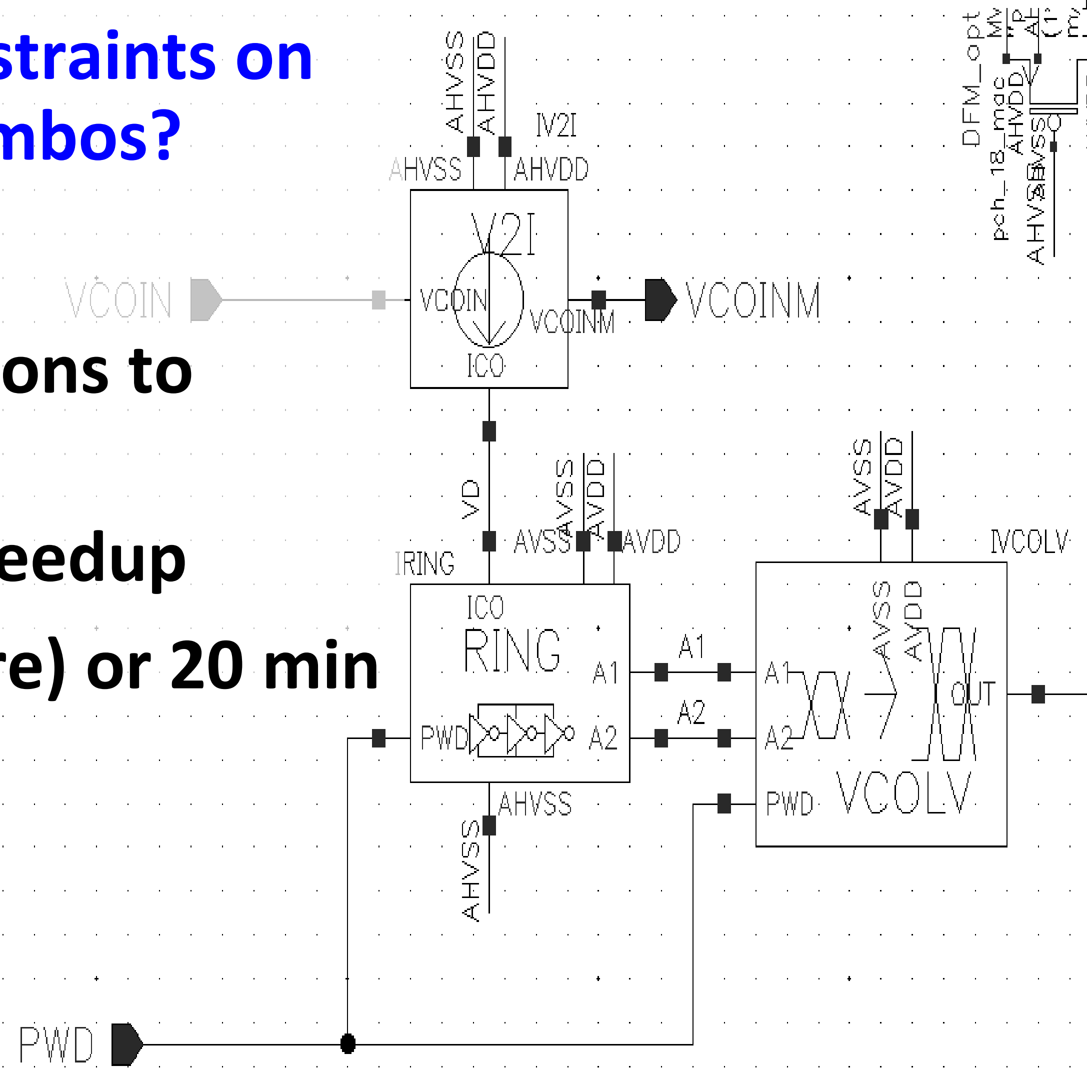


BGO for Circuit Verification – VCO of PLL

Q: Does circuit meet constraints on all 3375 "PVT corner" combos?

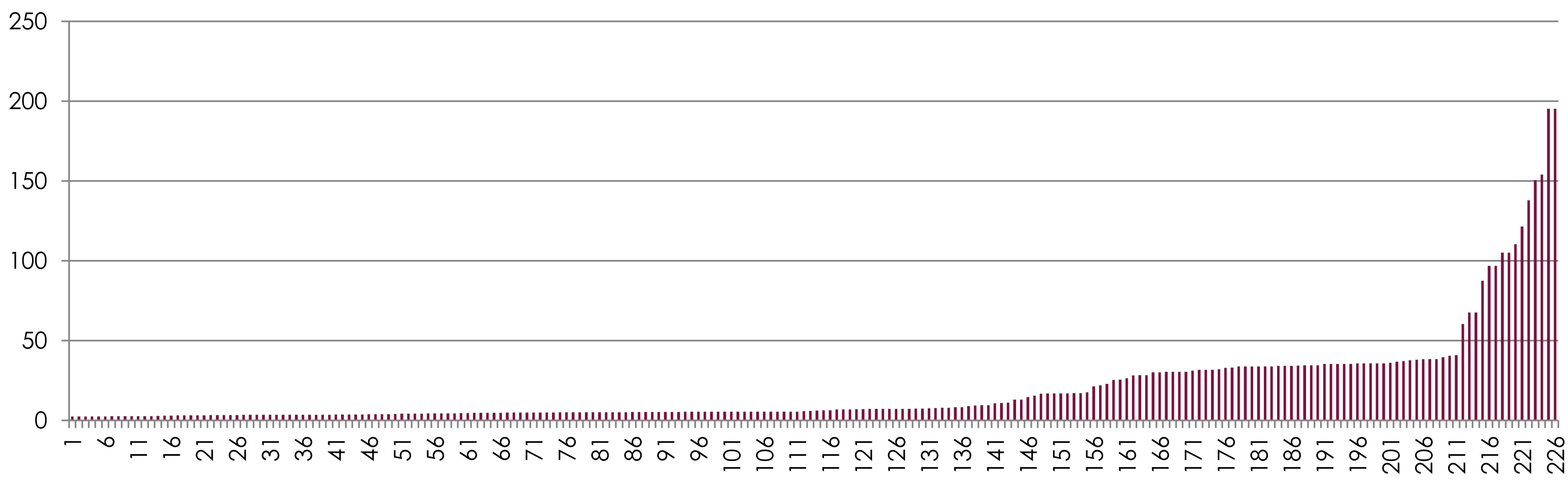
Result: used 171 evaluations to verify 3375 corners

- **3375/171 = 19.7x speedup**
- **65.6 h → 3.3 h (1 core) or 20 min (10 cores)**



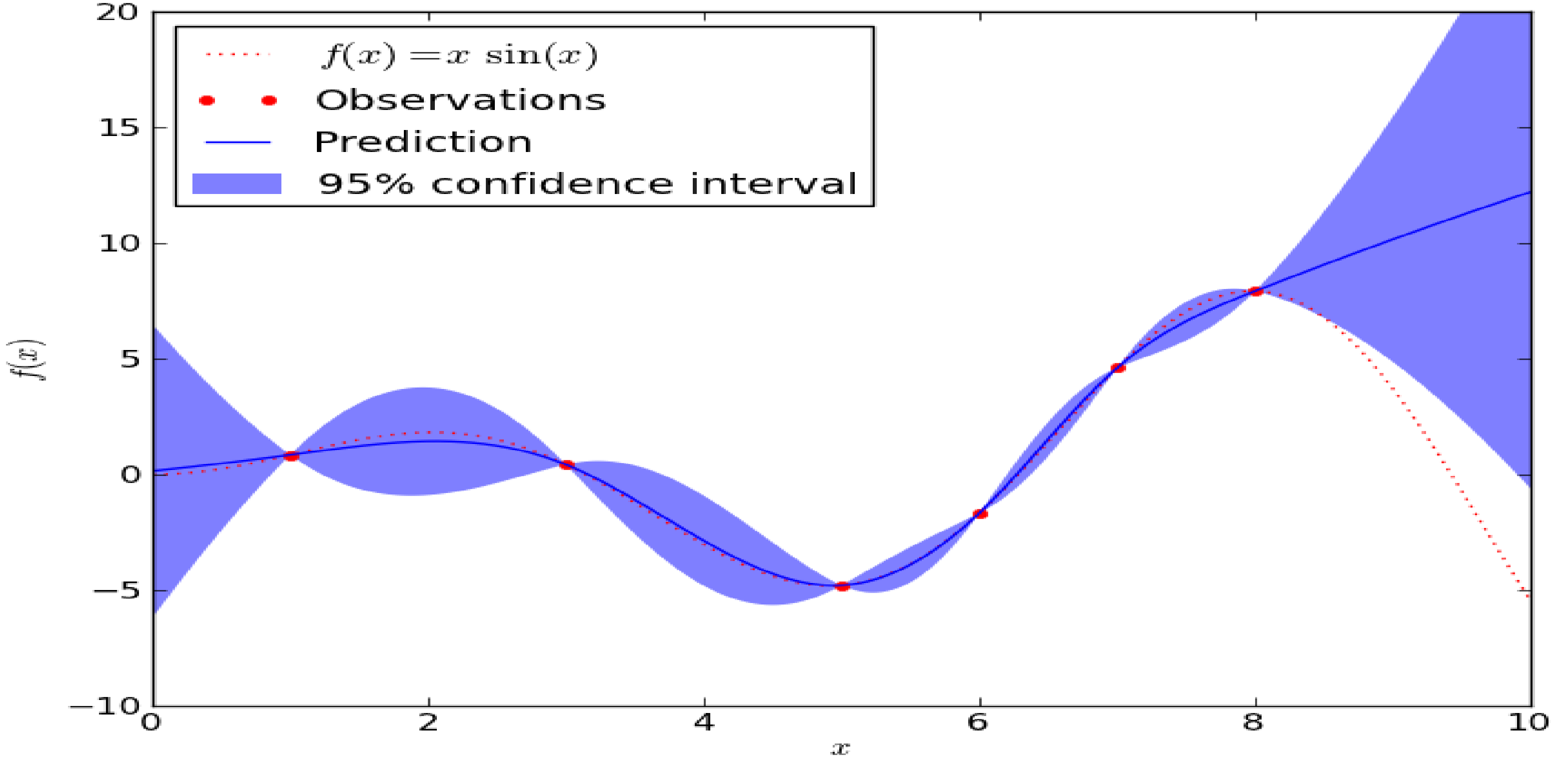
BGO Benchmarks on 226 Circuit Verification Problems

- 226 test cases in benchmark suite:
  - From Solido customers, in-house realistic cases, and in-house corner cases targeting challenging problems
  - Many contain complex interactions, non-linearities, discontinuities, etc.
- **226/226 (100%) of cases find true optimum**
- Speedup **2.34X to 226X**
- **Median speedup is 22X**



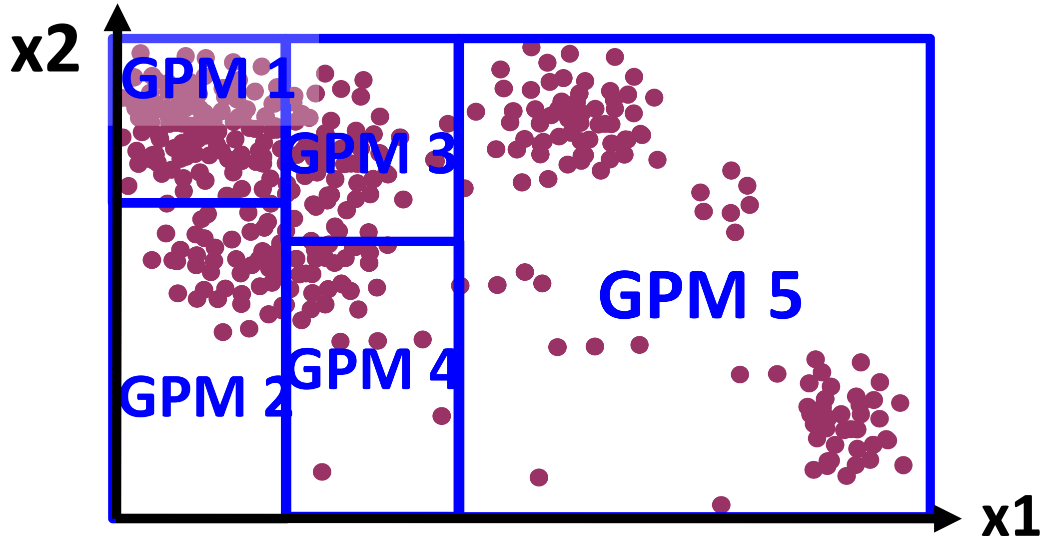
Speedup

BGO Scalability Challenge: # Samples



- **Problem: GPM training is O(N^3) on # Training Samples**
- **Becomes *very* unhappy when >1000 samples**
- This happens for circuit verification problems with larger # dimensions and highly nonlinear circuit
- First solution: just cut loose and sim all
- Is there a better way?

Solution: Divide-and-Conquer on Training Samples



- New model is a *set* of Gaussian Process Models (GPM)
- One GPM for each region of input x space
- Regions are automatically determined at build time
  - Via classic CART learning
  - Stop at a leaf when <700 samples
- Build a GPM on each leaf's samples (and *k* neighbors)
  - Each GPM is O(1) on # training samples because N=const
  - CART learning is O(N log N) on # samp with tiny constant

Benchmarking: GPM vs Divide-and-Conquer GPM

				GPM			Divide-and-conquer GPM		
Problem	**# vars**	**# train pts**	**# test pts**	**Build Time (s)**	**Test Time (s)**	**Error**	**Build Time**	**Test Time**	**Error**
Low-dimensional									
opamp-pvt-bandwidth	10	4425	1475	667.4	91.1	0.044	**55.6**	**7.4**	**0.006**
opamp-pvt-dc_gain	10	4425	1475	741.9	91.5	**0.001**	**57.9**	**8.5**	**0.003**
opamp-pvt-gain_margin	10	4425	1475	319.9	92.2	0.313	**59.6**	**8.2**	**0.168**
opamp-pvt-gbw	10	4425	1475	845.7	92.8	**0.010**	**62.4**	**8.8**	**0.008**
opamp-pvt-idc	10	4425	1475	775.2	91.7	**0.000**	**41.2**	**8.2**	**0.000**
opamp-pvt-phase_margin	10	4425	1475	268.2	90.9	**0.149**	**49.8**	**6.6**	**0.155**
High-dimensional									
senseamp_pwr	125	3750	1250	failed	failed	failed	**165.8**	**37.9**	**4.139**
opamp_AV	215	600	200	38.3	18.2	**2.933**	**23.3**	**9.8**	**3.628**
opamp_SR	215	600	200	34.8	18.2	**2.604**	**37.3**	**9.5**	**2.515**
compar_bw	639	1502	500	246.2	56.9	**16.010**	**73.7**	**23.0**	**16.458**
opamp_PM	215	600	200	63.9	18.3	3.678	**26.7**	**9.4**	**2.441**
opamp_BW	215	600	200	34.9	18.3	**1.800**	**31.6**	**9.6**	**2.084**
mem	385	7500	2500	failed	failed	failed	**422.4**	**78.3**	**0.480**
senseamp_delay	125	3750	1250	failed	failed	failed	**286.0**	**38.4**	**5.135**

Bayesian Global Optimization For ML Modeling

BGO for Deep Belief Net (DBN) Optimization

Algorithms for Hyper-Parameter Optimization

James Bergstra
The Rowland Institute
Harvard University
bergstra@rowland.harvard.edu

Rémi Bardenet
Laboratoire de Recherche en Informatique
Université Paris-Sud
bardenet@lri.fr

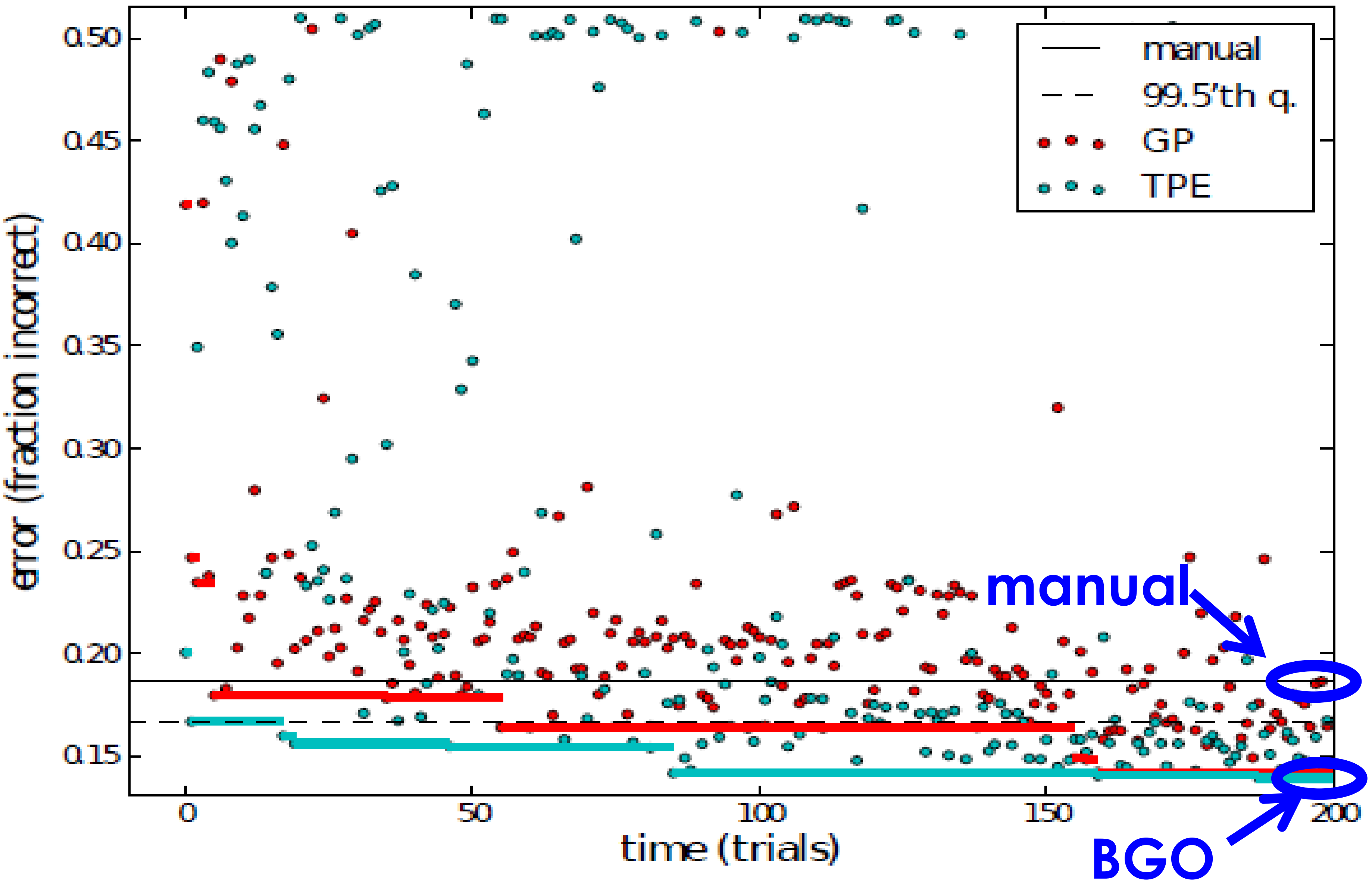
Yoshua Bengio
Dépt. d'Informatique et Recherche Opérationelle
Université de Montréal
yoshua.bengio@umontreal.ca

Balázs Kégl
Linear Accelerator Laboratory
Université Paris-Sud, CNRS
balazs.kegl@gmail.com

[NIPS 2011]

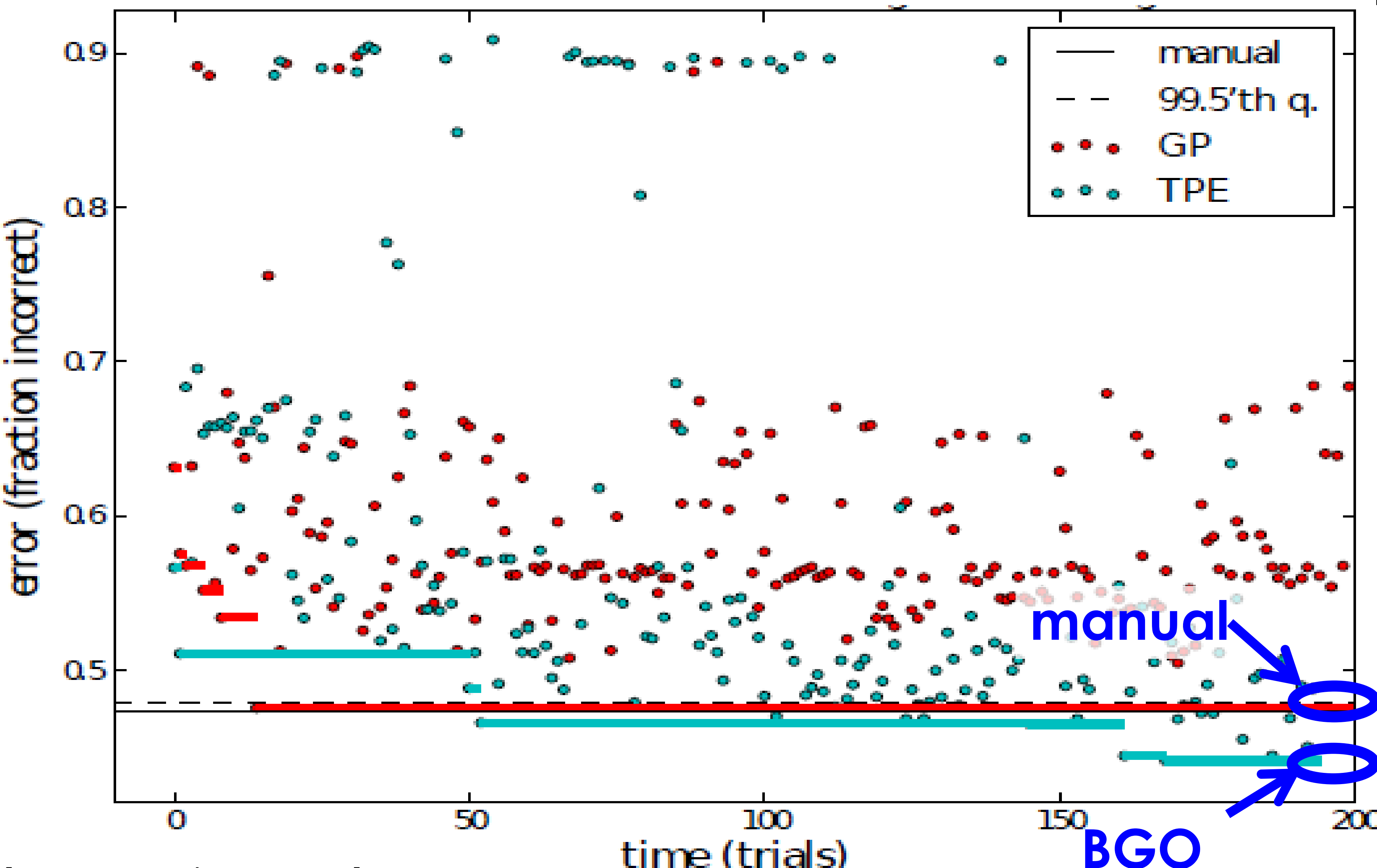
Whole model		**Per-layer**	
Parameter	Prior	Parameter	Prior
pre-processing	raw or ZCA	n. hidden units	$\log U(128, 4096)$
ZCA energy	$U(.5, 1)$	W init	$U(-a, a)$ or $\mathcal{N}(0, a^2)$
random seed	5 choices	a	algo A or B (see text)
classifier learn rate	$\log U(0.001, 10)$	algo A coef	$U(.2, 2)$
classifier anneal start	$\log U(100, 10^4)$	CD epochs	$\log U(1, 10^4)$
classifier ℓ_2-penalty	0 or $\log U(10^{-7}, 10^{-4})$	CD learn rate	$\log U(10^{-4}, 1)$
n. layers	1 to 3	CD anneal start	$\log U(10, 10^4)$
batch size	20 or 100	CD sample data	yes or no

BGO DBN Optimization, on Convex Dataset



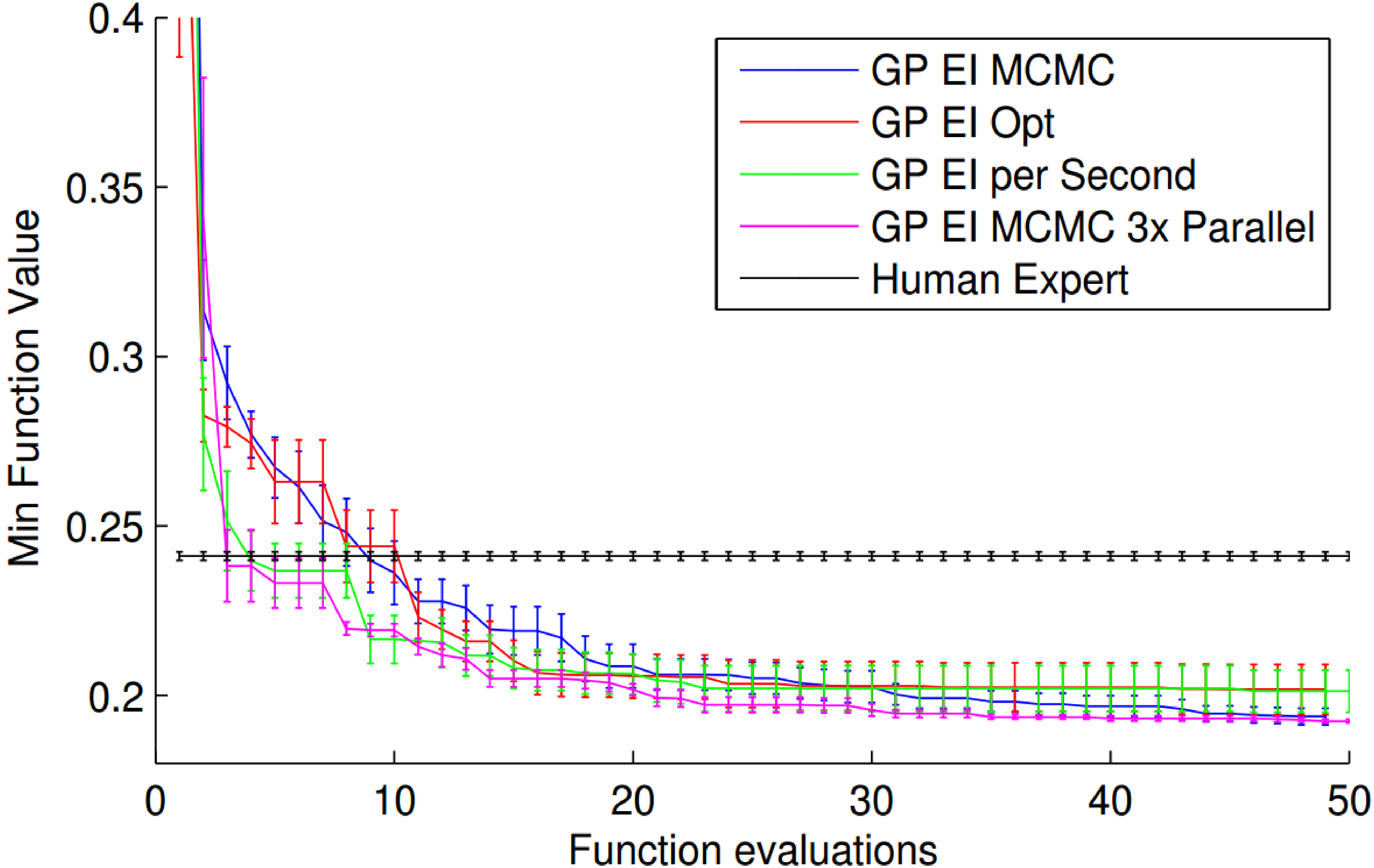
[Bergstra et al NIPS 2011]

BGO DBN Optimization, on MNIST Dataset



[Bergstra et al NIPS 2011]

BGO ConvNN Optimization on CIFAR-10 data



[Snoek et al NIPS 2012]

Some Applications of Global Optimization

- **Semiconductors: Verify a circuit across PVT corners**
- **ML modeling: Find optimal model meta-parameters (DeepNN, RF, SVM, ..), for application to computer vision etc.**
- Health: protein shape prediction (minimal energy configuration)
- Business Intelligence: optimize churn & other key performance indicators (KPIs)
- Big data infrastructure: optimize reliability / uptime, minimize power consumption, ..
- Internet / mobile: auto SEO, optimize for app store placement (rank, profitability)
- Oil & gas: capital & resource allocation
- Space: minimize interplanetary trajectory flight time

Active Predictive Analytics / Supercharged Manual Design

Automated vs. Manual Design

Benefits of Automated (Optimization):

- Efficient; high throughput
- Scalable – no human bottleneck
- Optimal design (assuming you measure everything, and have appropriate objectives & constraints)

Benefits of Manual:

- Retain insight
- Retain control
- Insight & control lead to creative structural improvements
- Don't need perfect measures, objectives, constraints
- Familiar

Automated vs. Manual Design

Benefits of Automated:

- Efficient, optimal, ..

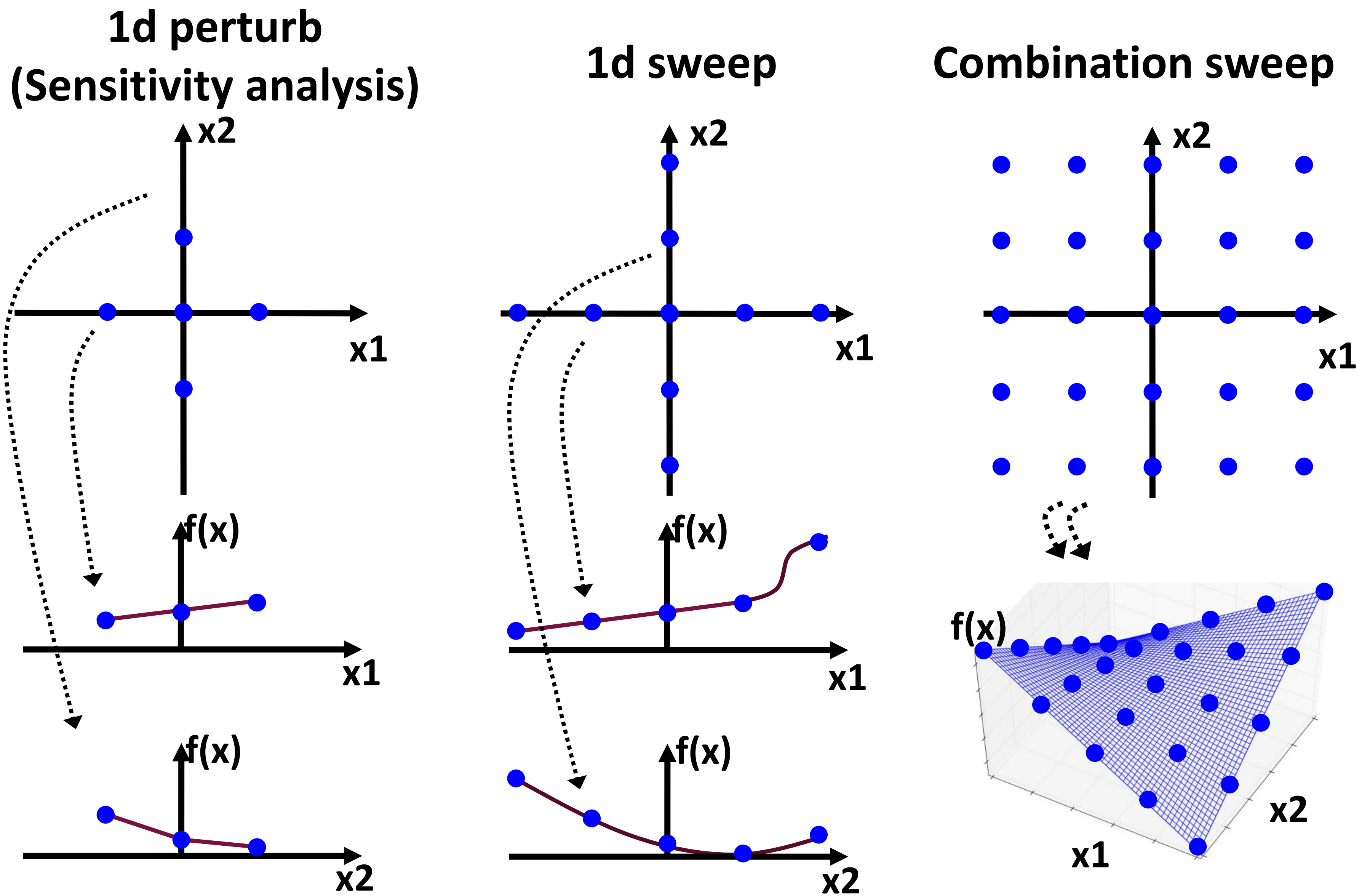
Benefits of Manual:

- Retain insight, control, ..

Q: Can we get the best of both worlds?

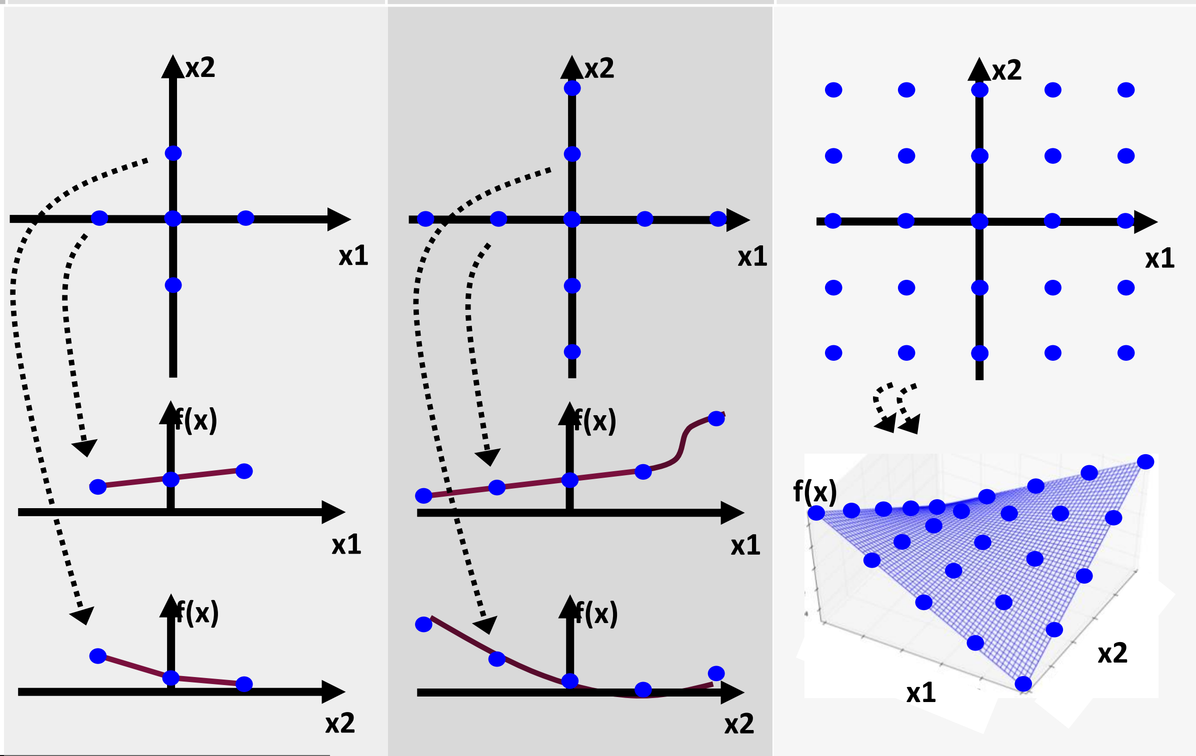
- **Can we “supercharge” manual design?**
- **What might that mean?**

Background: “Sweep” in Manual Design

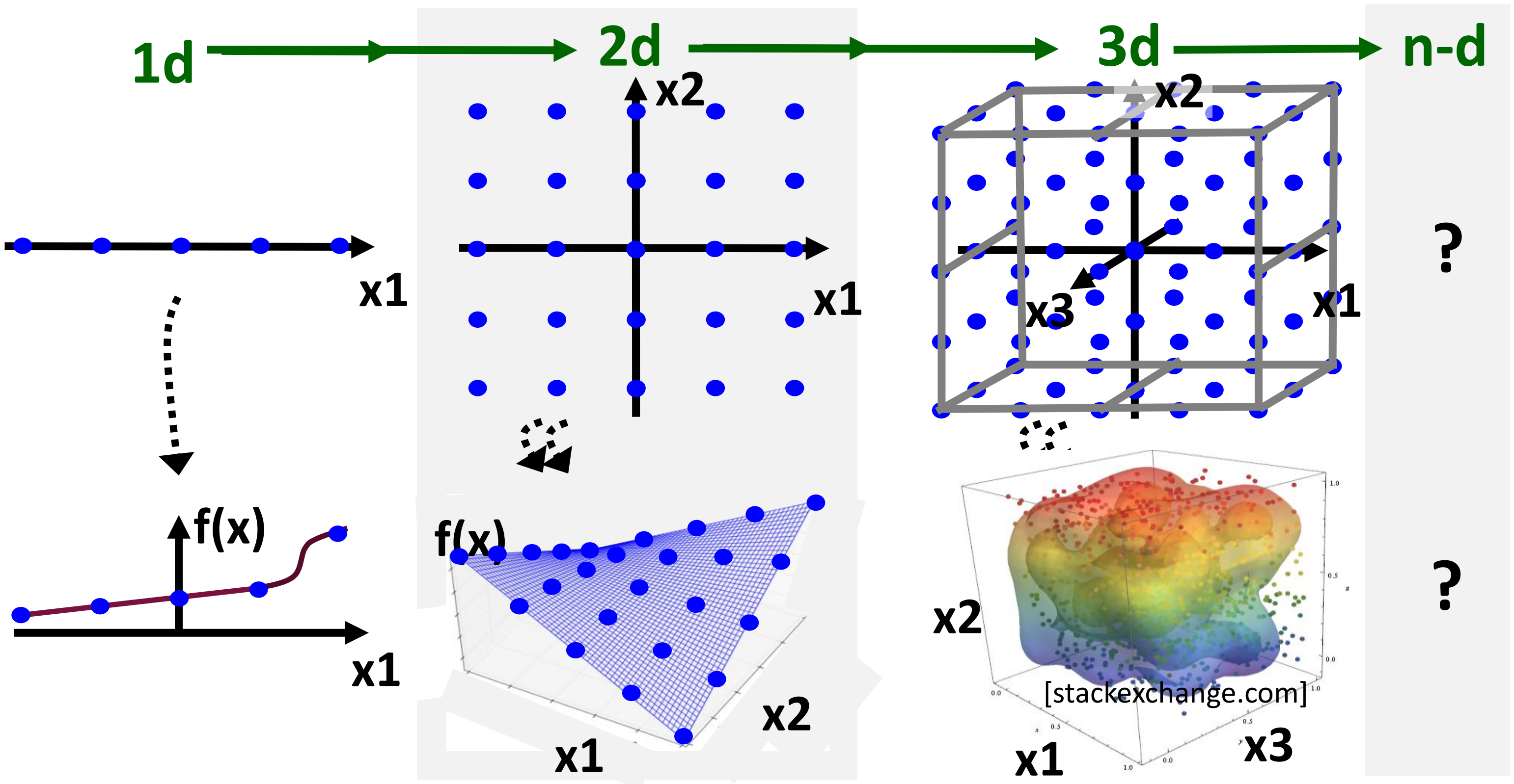


Background: “Sweep” in Manual Design

	1d Perturb	1d Sweep	Combination Sweep
Fast	Yes	Yes	No (takes $nval^{nvar}$ sims)
Accurate	No (too local, misses interactions)	No (misses interactions)	Yes
Scalable	Yes	Yes	No (≤5 vars)



Consider: Combination Sweep in >>1 Dimension



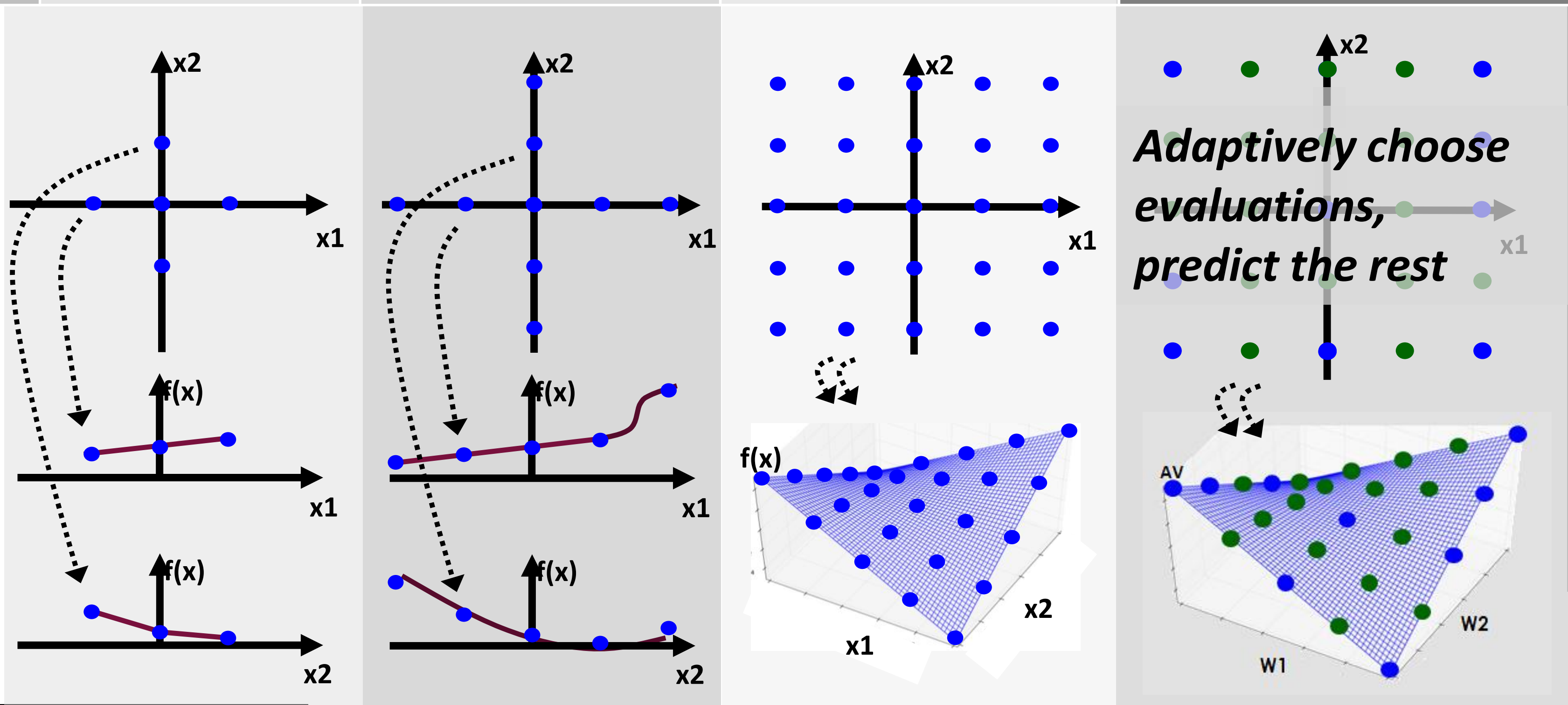
Insight & control would be perfect! But two big problems:

1. **Computational expense is insane** (e.g. 10G sims for 10 variables and 10 values per variable)
2. **How to visualize beyond 3d?** (and 3d is weak too)

Problem #1: Insane Computational Expense

Solution: Active Learning (via BGO)

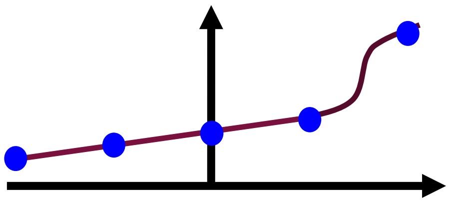
	1d Perturb	1d Sweep	Combination Sweep	Active Learning (Fast Comb. Sweep)
Fast	Yes	Yes	No (takes $nval^{nvar}$ sims)	Yes
Accurate	No (too local, misses interactions)	No (misses interactions)	Yes	Yes
Scalable	Yes	Yes	No (≤5 vars)	Yes

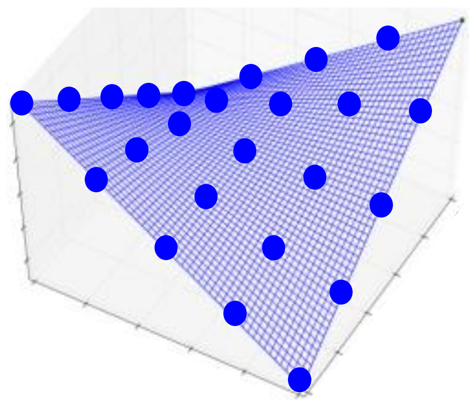


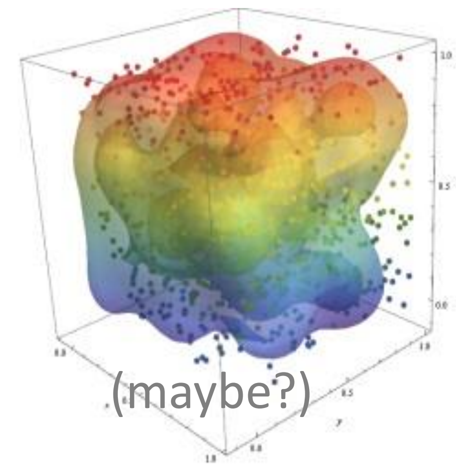
Problem #2: How to Visualize Sweeps in >>1 Dimension?

Solution: Predictive Analytics / Knowledge Extraction

1d → 2d → 3d → n-d







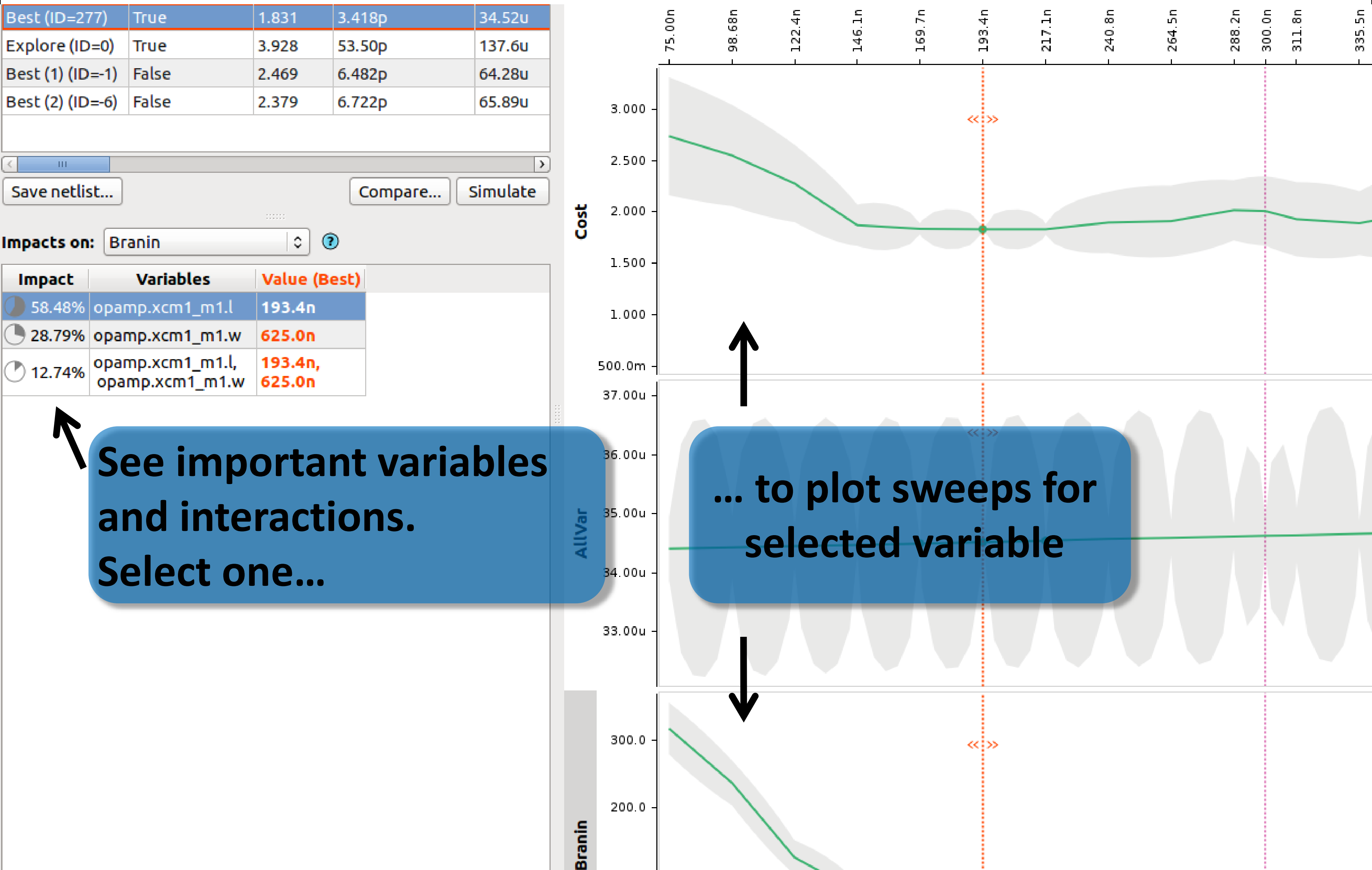
?

The key: focus on *important* variables and interactions

- Calculate impacts of x_i, x_i&x_j by mining the GPM
- Show impacts
- Let user select an x_i or x_i&x_j
- Show mapping from x_i, x_i&x_j to f(x)

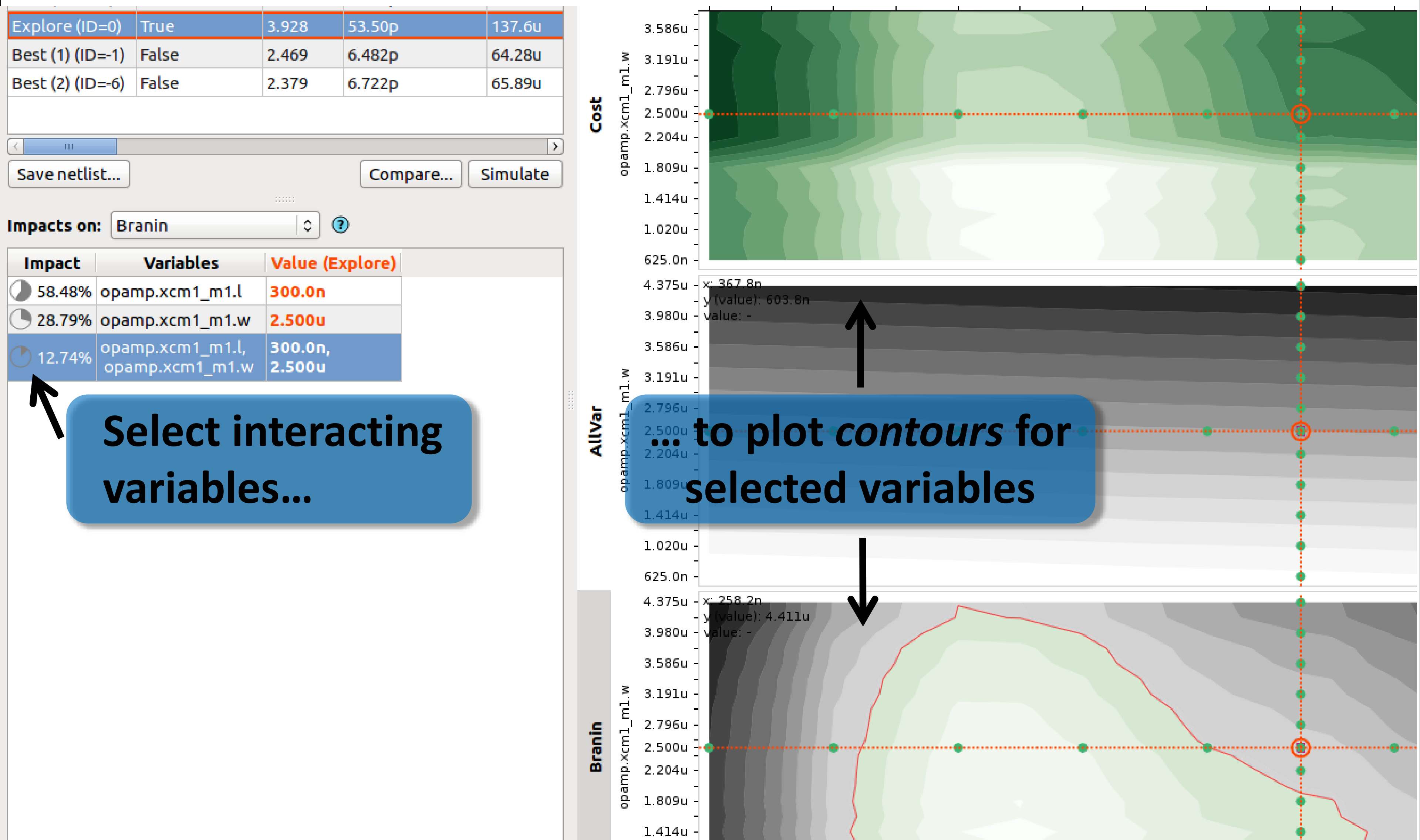
Problem #2: How to Visualize Sweeps in >>1 Dimension?

Solution: Predictive Analytics / Knowledge Extraction



Problem #2: How to Visualize Sweeps in >>1 Dimension?

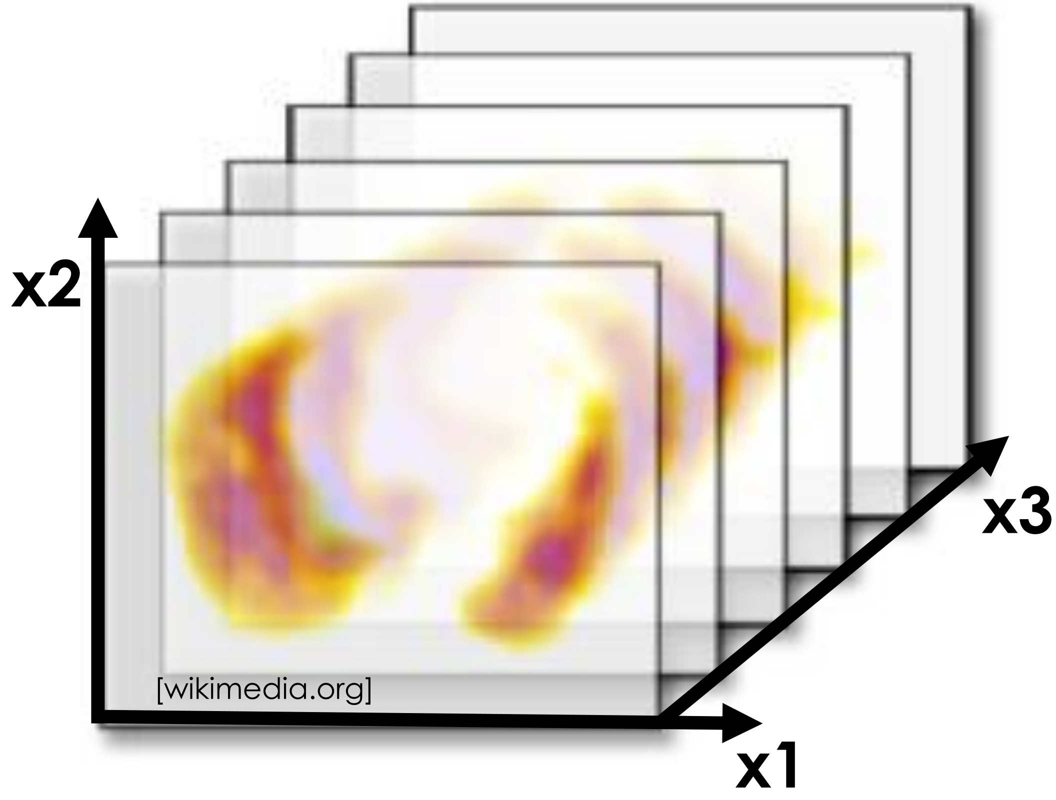
Solution: Predictive Analytics / Knowledge Extraction



Problem #2.1: How to Choose Slicing Plane?

Example:

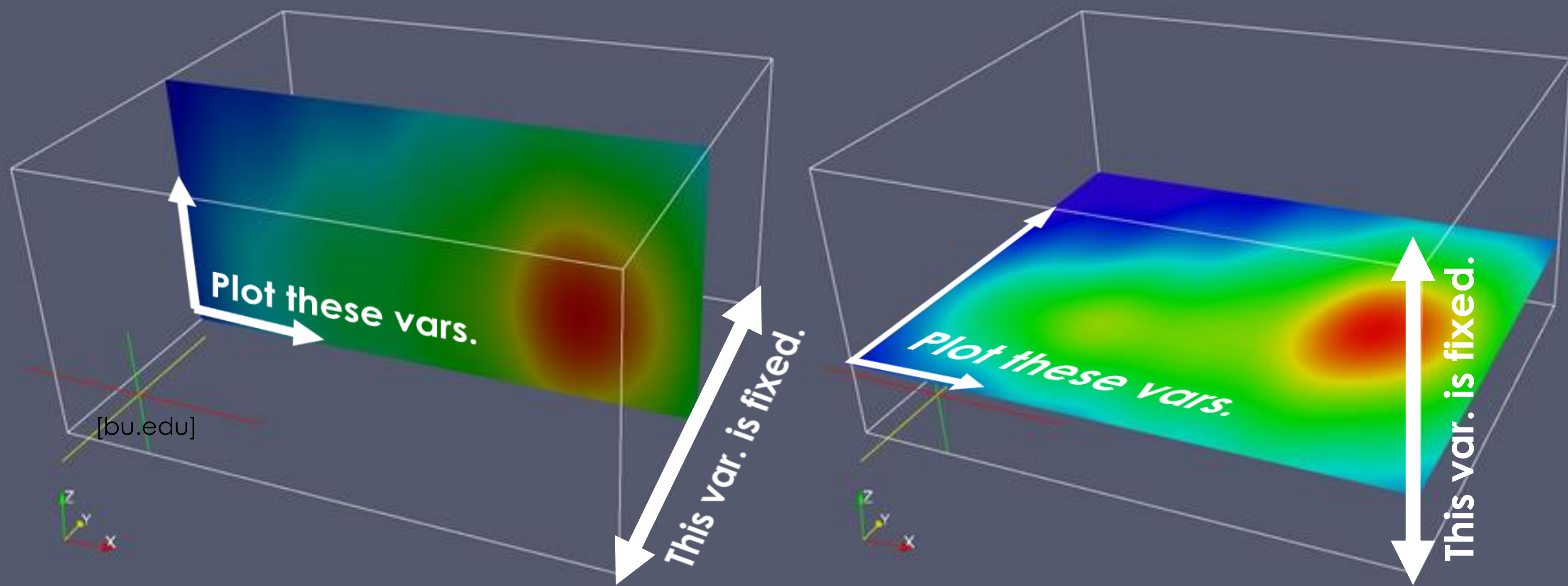
Plotting f(x) vs x_1&x_2, so need to fix the value for x_3.



Problem #2.1: How to Choose Slicing Plane?

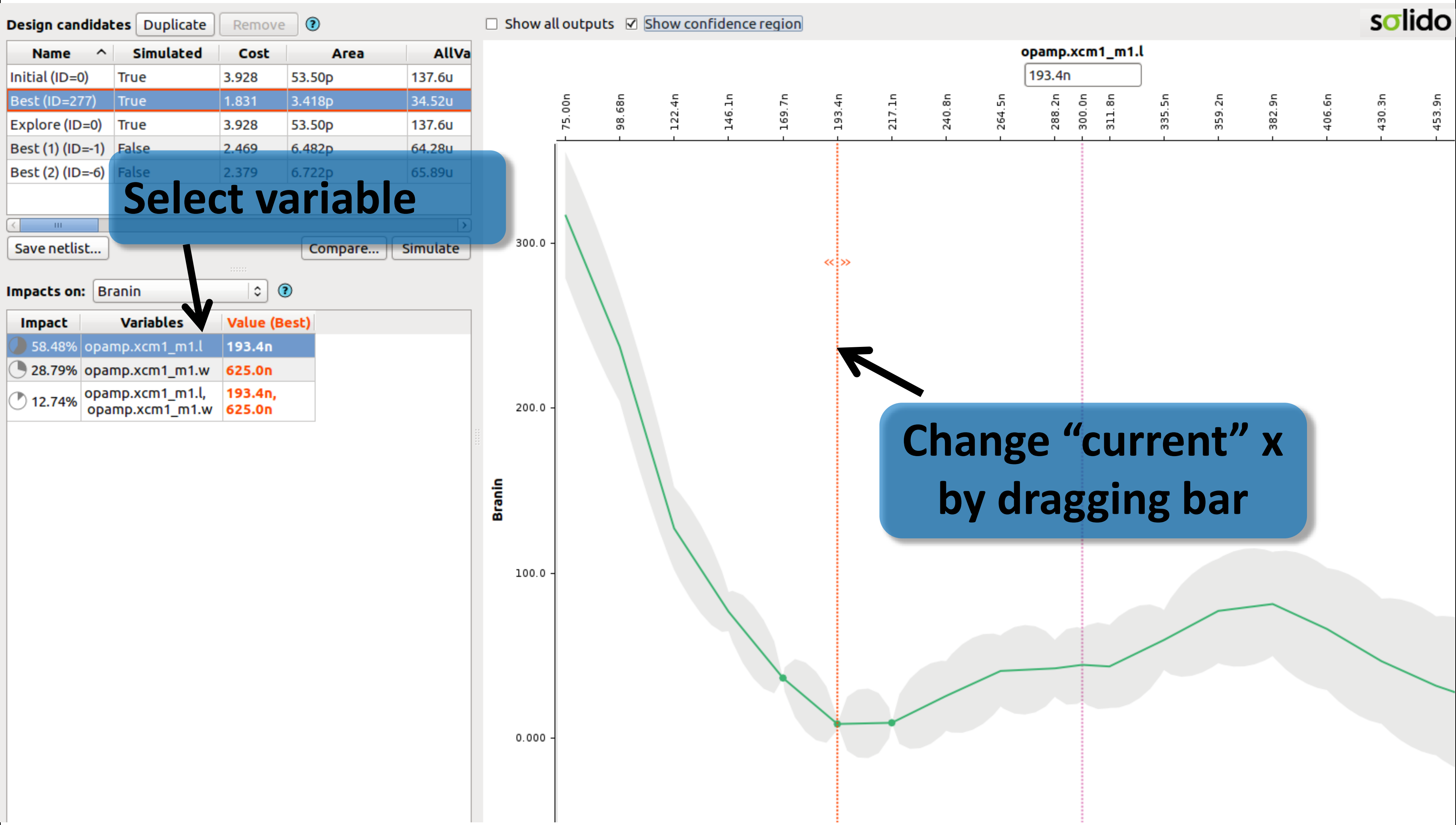
Generalization:

Need to fix the value for *all vars. not being plotted*.



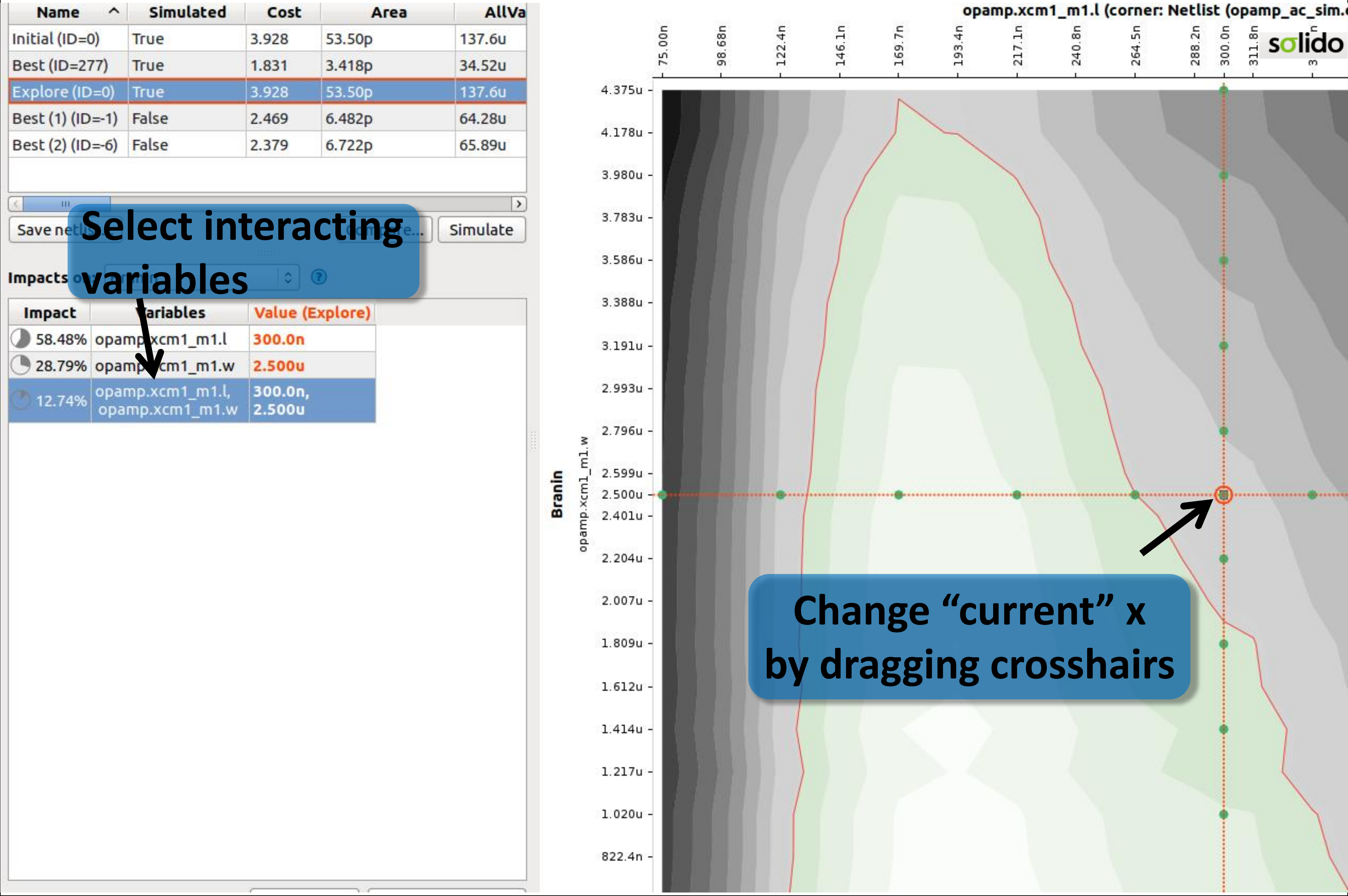
Problem #2.1: How to Choose Slicing Plane?

Solution: User Interactively Fixes the Value for "Current x"



Problem #2.1: How to Choose Slicing Plane?

Solution: User Interactively Fixes the Value for "Current x"



Demo

Active Learning + Predictive Analytics
= Active Predictive Analytics (aka Supercharged Manual Design)

Has Benefits of Automated (Optimization):

- Efficient; high throughput
- Scalable – no human bottleneck
- Optimal design

***And* Has Benefits of Manual:**

- Retain insight. Actually, even better insight than before!
- Retain control
- Insight & control lead to creative structural improvements
- Don’t need perfect measures, objectives, constraints
- Familiar – “just” sweeps

Applications of Active Predictive Analytics *include* Applications of Global Optimization

- **Semiconductors: Verify a circuit across PVT corners**
- **ML modeling: Find optimal model meta-parameters (DeepNN, RF, SVM, ..), for application to computer vision etc.**
- Health: protein shape prediction (minimal energy configuration)
- Business Intelligence: optimize churn & other key performance indicators (KPIs)
- Big data infrastructure: optimize reliability / uptime, minimize power consumption, ..
- Internet / mobile: auto SEO, optimize for app store placement (rank, profitability)
- Oil & gas: capital & resource allocation
- Space: minimize interplanetary trajectory flight time

Some Users of Bayesian Global Optimization or Active Predictive Analytics

NVIDIA®

BROADCOM®

SONY

NXP

QUALCOMM®

tsmc®

AVAGO
TECHNOLOGIES

HUAWEI

Tektronix®

Sidense

Inphi

Conclusion

"Push Designs": Bayesian Global Optimization

- Application to circuit verification, ML modeling
- How to handle CIs that lie, and scaling up

"Pull Designs": Active Predictive Analytics

- Aka Supercharged Manual Design
- Via Active Learning / Fast Combination Sweep
- And Predictive Analytics / Knowledge Extraction
- Efficient & optimal (like opt)
- Maintains user control & insight (like manual)