DSE Introduction

Luigi Nardi

Assistant Professor Lund University

Researcher Stanford University

@PLDI DSE Tutorial

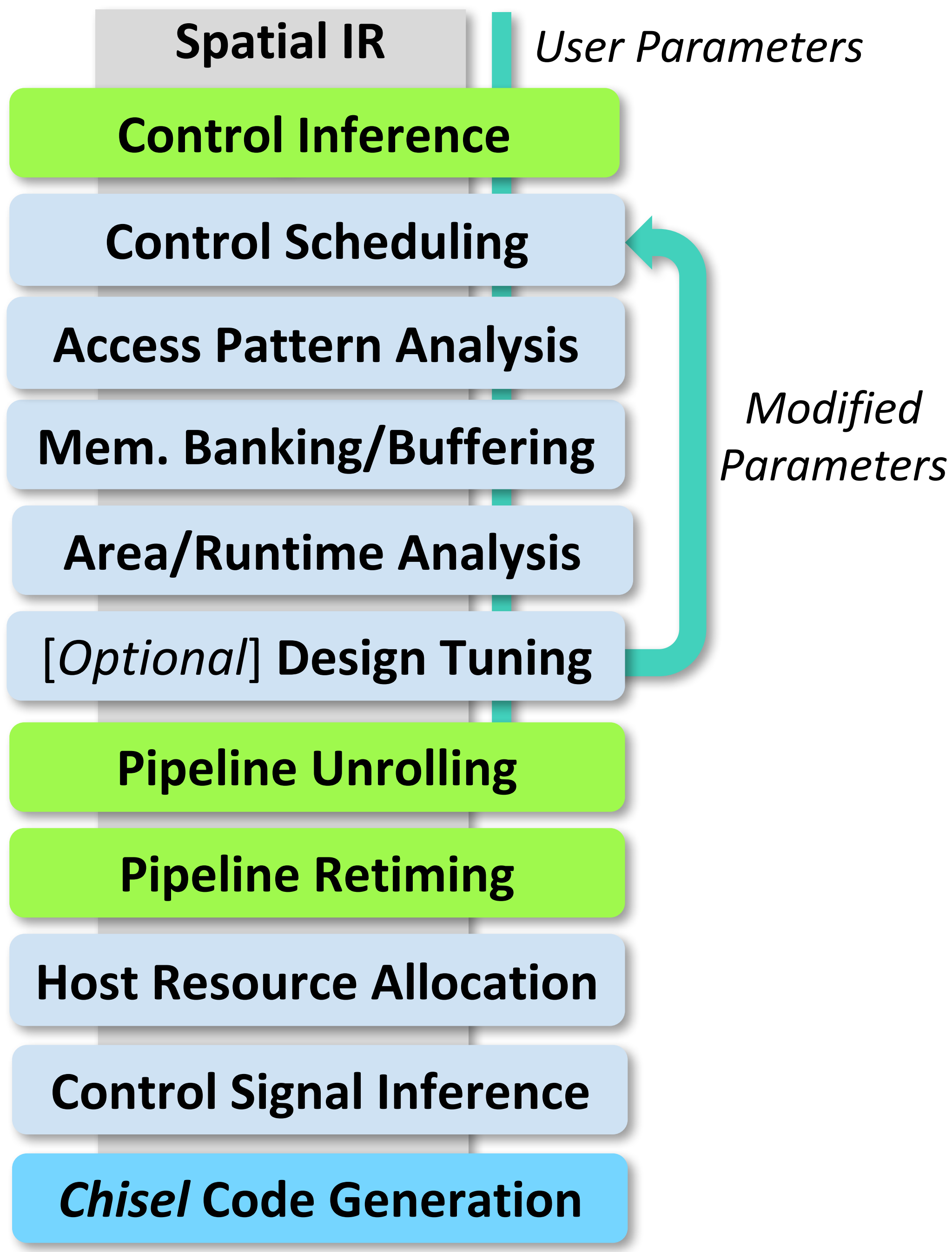
June 15, 2020

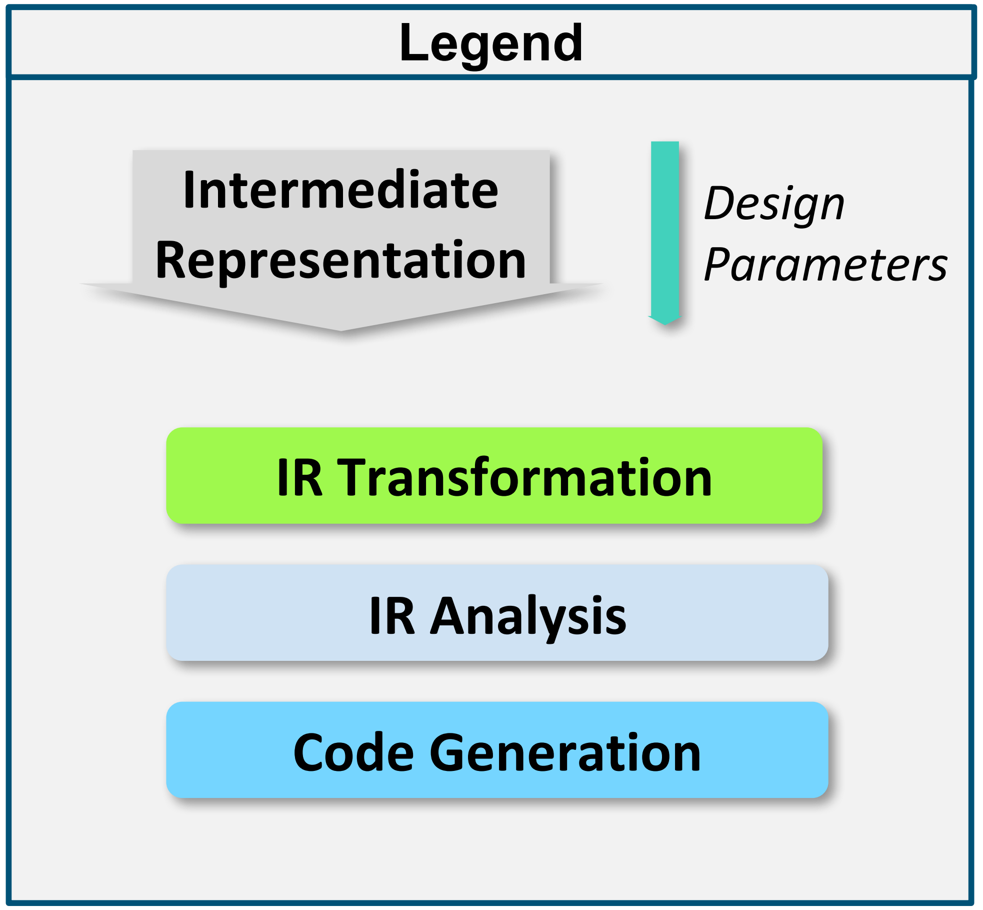
Lund University

Main Papers Behind This Talk

1. Souza, et al., "Prior-guided Bayesian Optimization", under review, **2020**
2. **Nardi, et al., "Practical Design Space Exploration", MASCOTS, 2019**
3. **Koeplinger, et al., "Spatial: A Language and Compiler for Application Accelerators", PLDI, 2018**
4. Nardi, et al., "Algorithmic performance-accuracy trade-off in 3D vision applications using HyperMapper", iWAPT-IPDPS, 2017
5. Saeedi, et al., "Application-oriented design space exploration for SLAM algorithms", ICRA, 2017
6. Bodin, et al., "Integrating algorithmic parameters into benchmarking and design space exploration in 3D scene understanding", PACT, **2016**

The Spatial Compiler

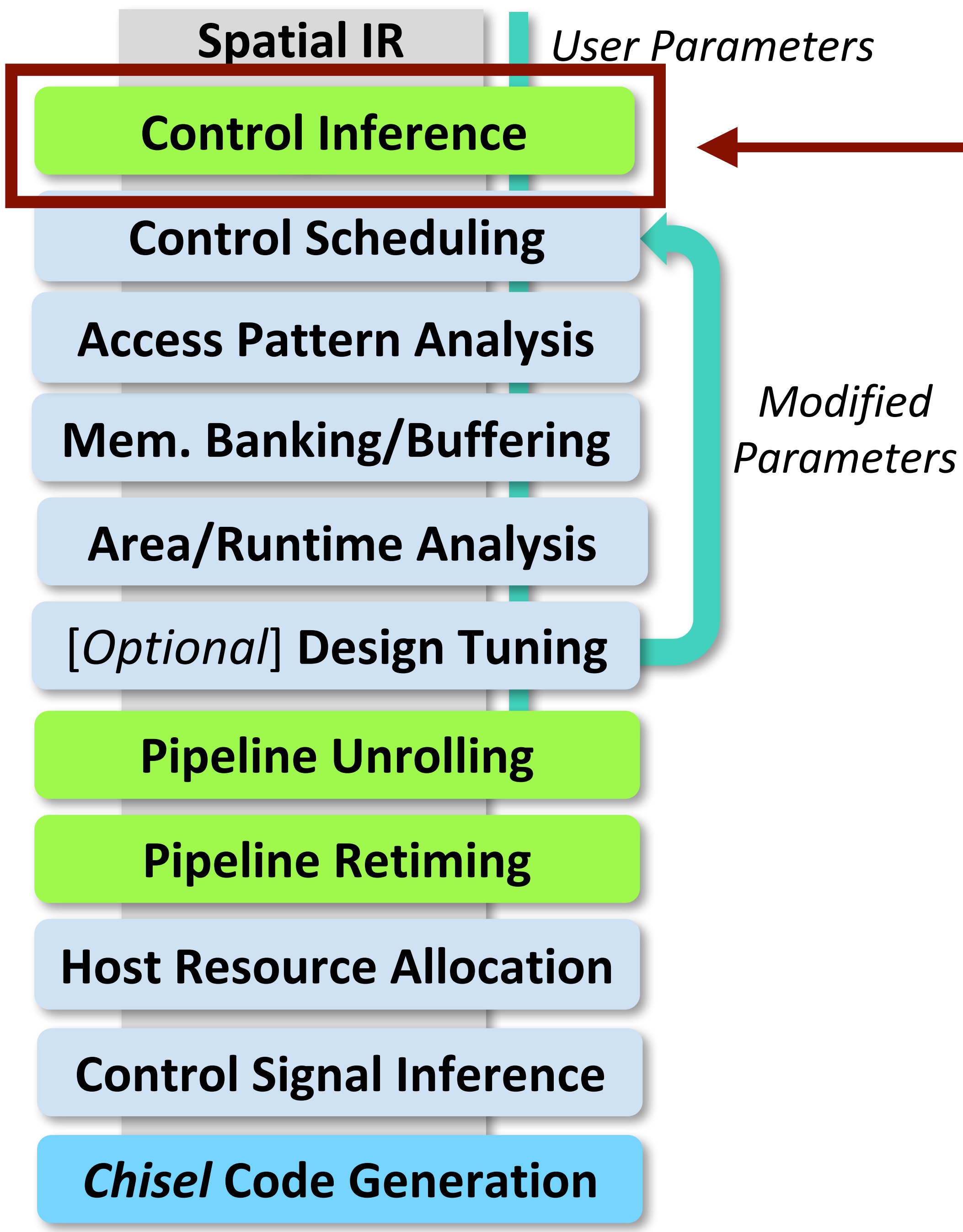




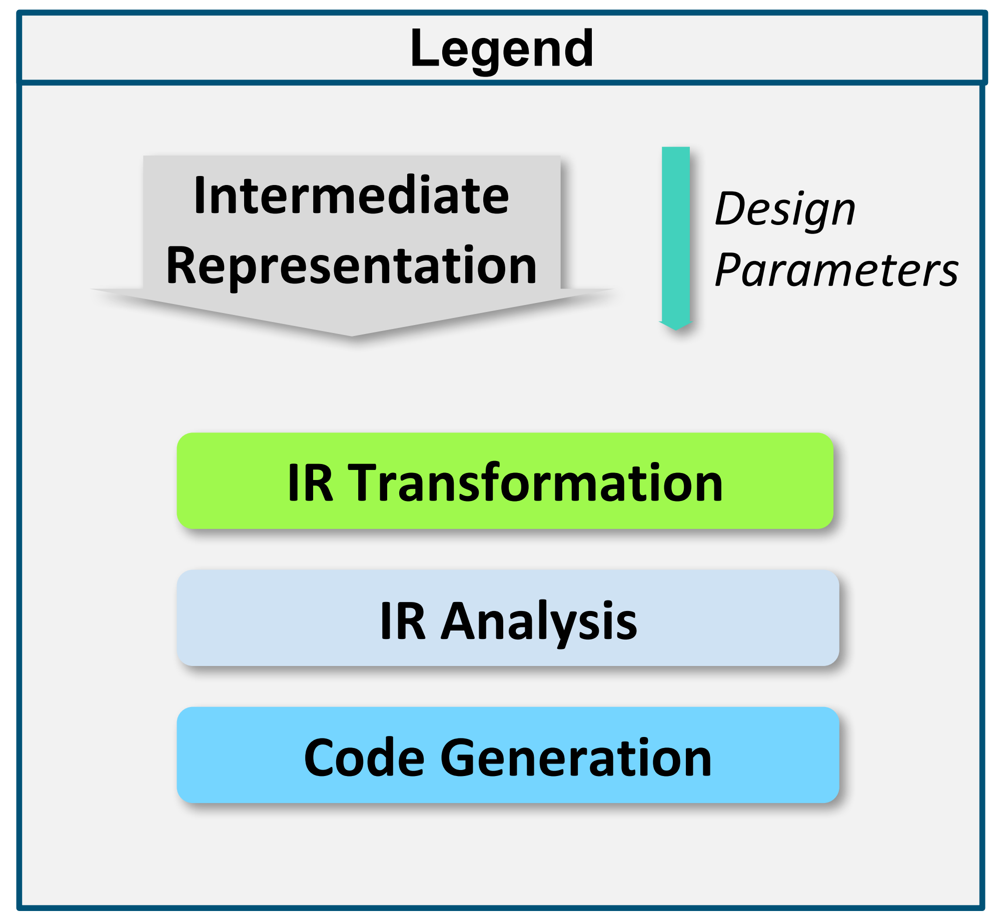
- Goal of Spatial [Koeplinger, et al.]: design of application accelerators
- On reconfigurable architectures FPGAs and CGRAs
- Spatial compiler lowers user programs into synthesizable Chisel [Bachrach, et al.]

[Koeplinger, et al.] "Spatial: a language and compiler for application accelerators." PLDI 2018
[Bachrach, et al.] "Chisel: constructing hardware in a Scala embedded language." DAC 2012.

The Spatial Compiler



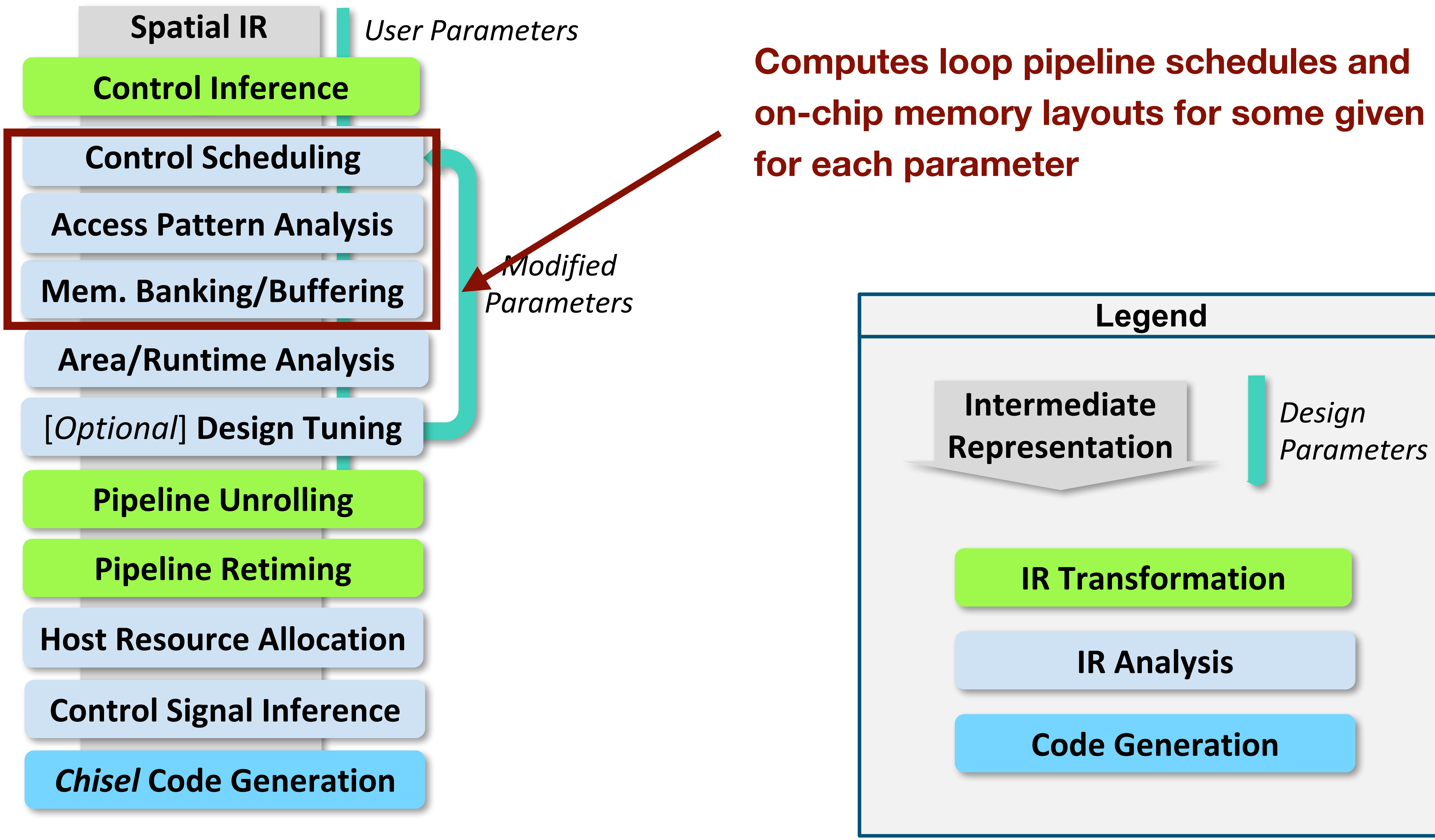
Performs basic hardware optimizations and estimates a domain for each design parameter



- Goal of Spatial [Koeplinger, et al.]: design of application accelerators
- On reconfigurable architectures FPGAs and CGRAs
- Spatial compiler lowers user programs into synthesizable Chisel [Bachrach, et al.]

[Koeplinger, et al.] "Spatial: a language and compiler for application accelerators." PLDI 2018
[Bachrach, et al.] "Chisel: constructing hardware in a Scala embedded language." DAC 2012.

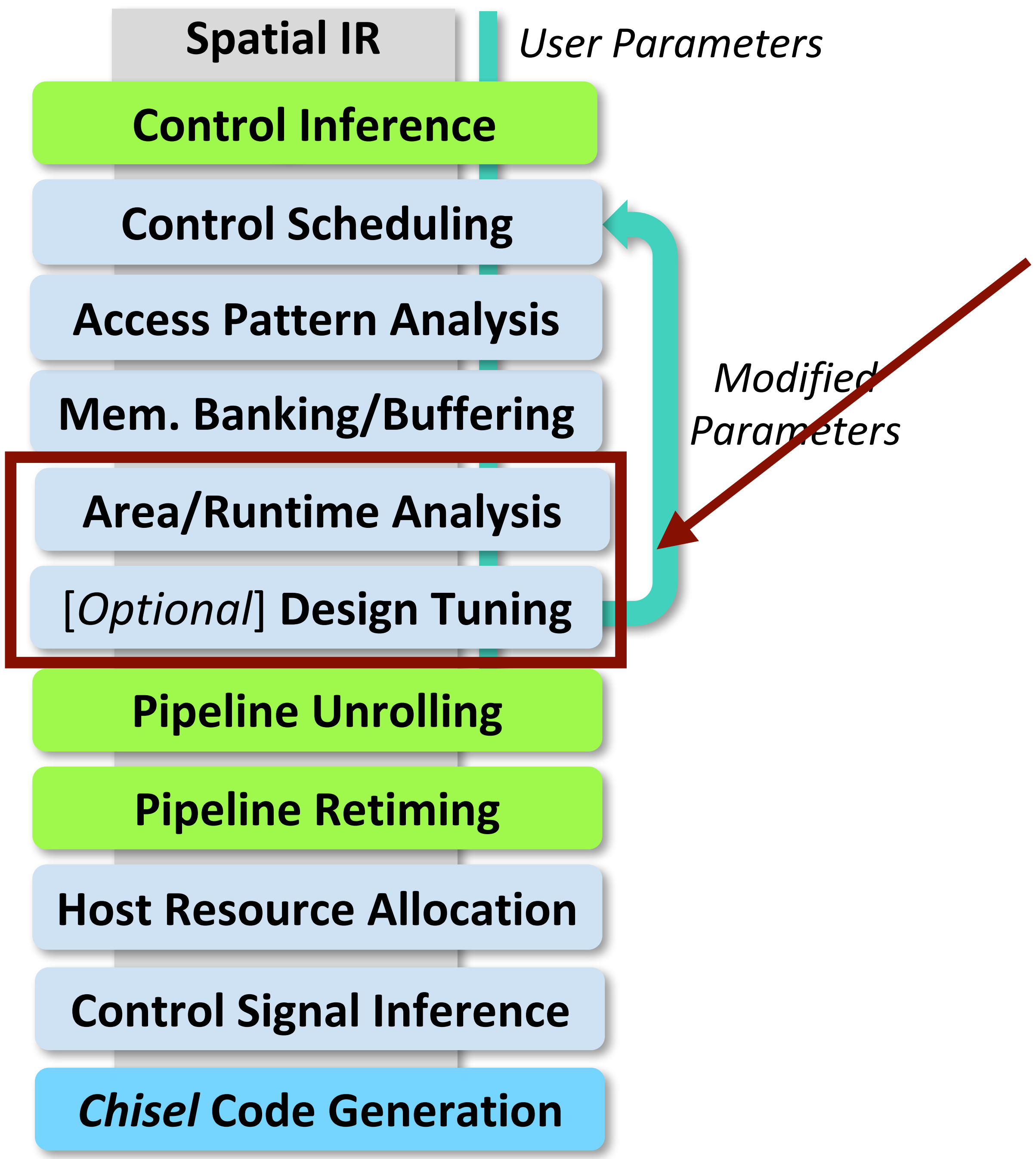
The Spatial Compiler



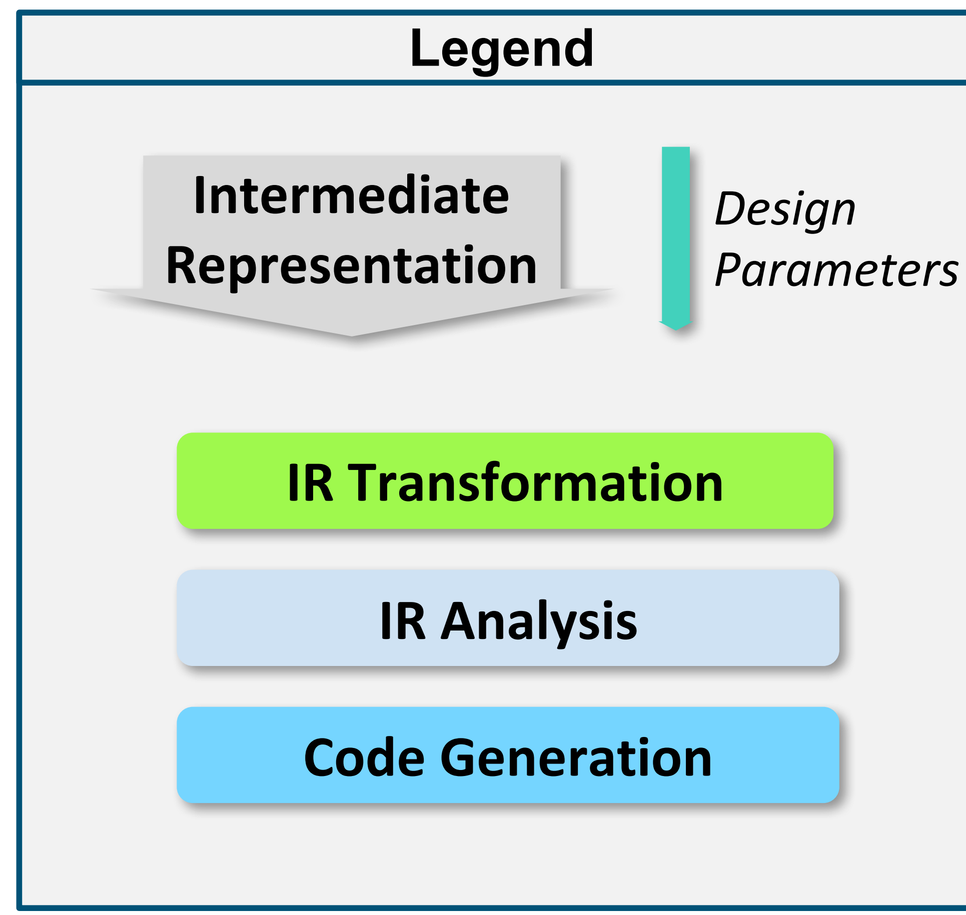
- Goal of Spatial [Koeplinger, et al.]: design of application accelerators
- On reconfigurable architectures FPGAs and CGRAs
- Spatial compiler lowers user programs into synthesizable Chisel [Bachrach, et al.]

[Koeplinger, et al.] "Spatial: a language and compiler for application accelerators." PLDI 2018
[Bachrach, et al.] "Chisel: constructing hardware in a Scala embedded language." DAC 2012.

The Spatial Compiler



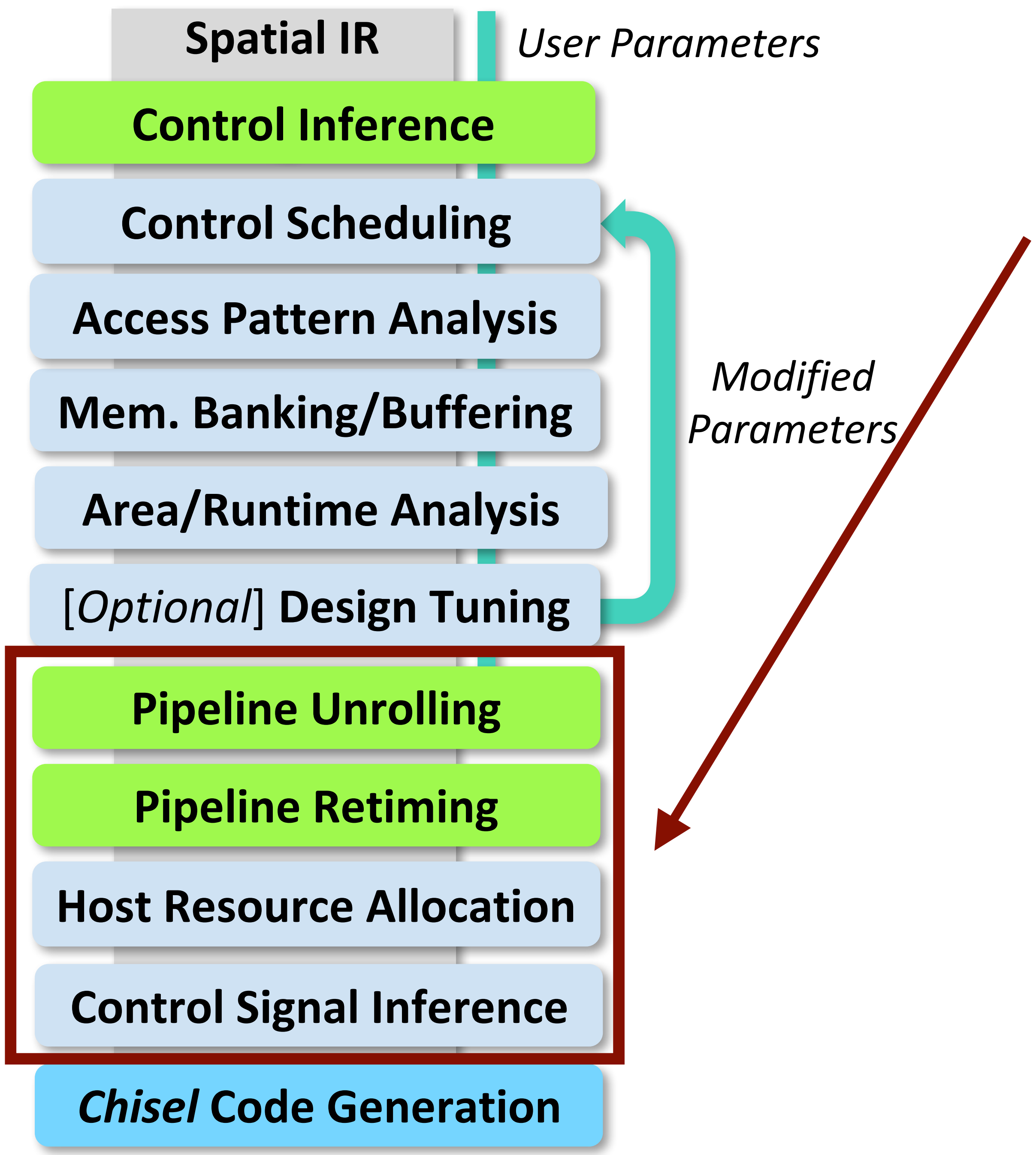
Estimates the amount of hardware resources and the runtime of the application



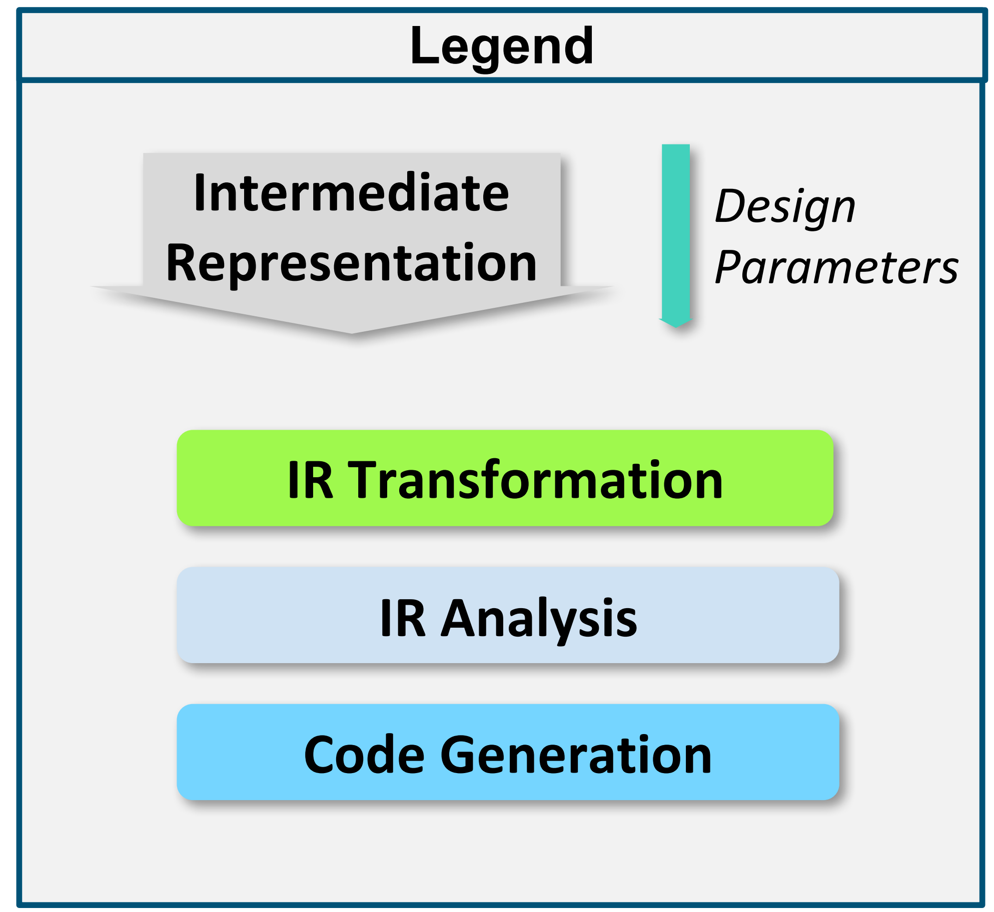
- Goal of Spatial [Koeplinger, et al.]: design of application accelerators
- On reconfigurable architectures FPGAs and CGRAs
- Spatial compiler lowers user programs into synthesizable Chisel [Bachrach, et al.]

[Koeplinger, et al.] "Spatial: a language and compiler for application accelerators." PLDI 2018
[Bachrach, et al.] "Chisel: constructing hardware in a Scala embedded language." DAC 2012.

The Spatial Compiler



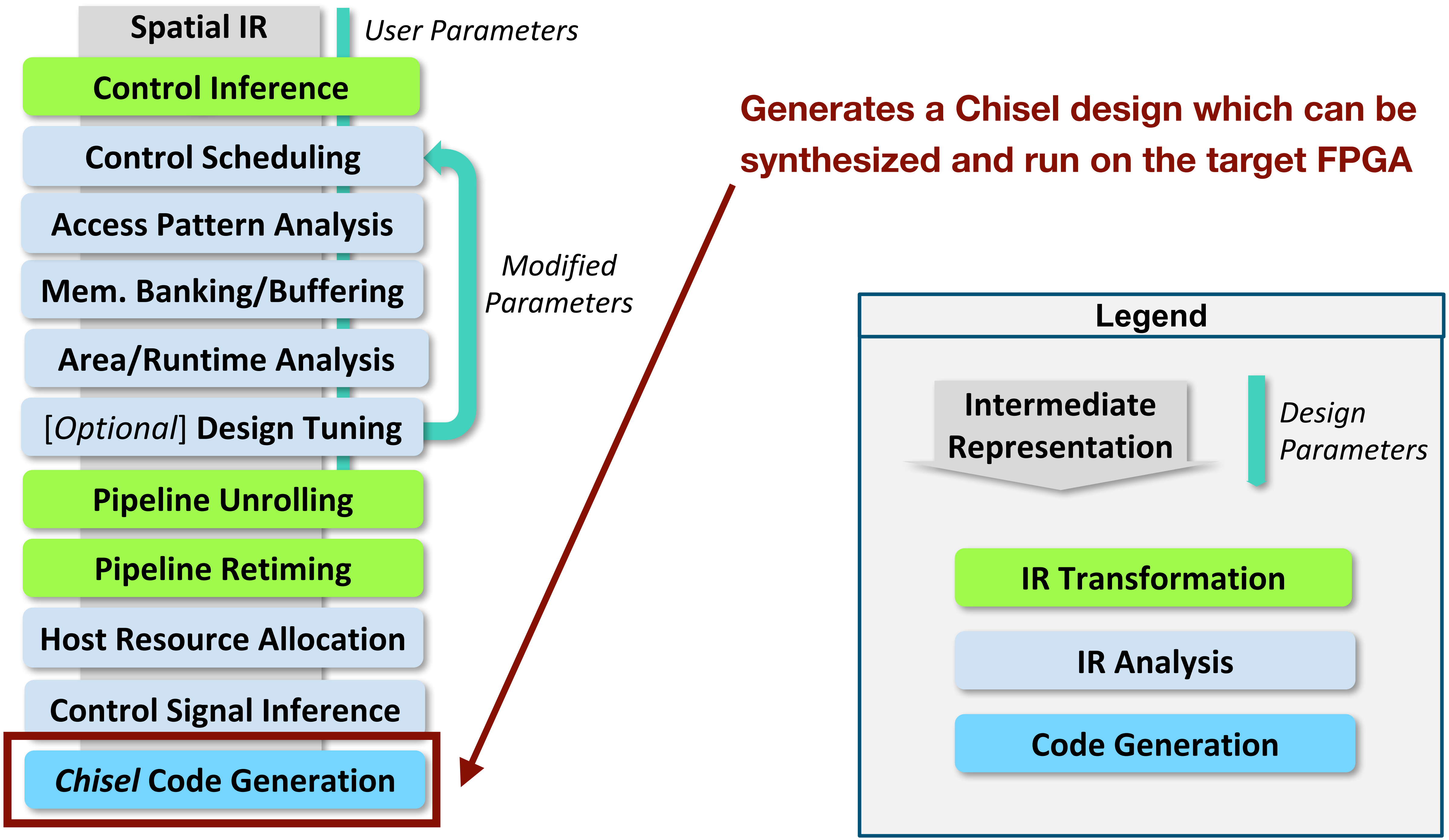
Unrolls loops, retimes pipelines, and performs on-chip memory layout.
The optimizations are computed based on the analyses of the previous phase



- Goal of Spatial [Koeplinger, et al.]: design of application accelerators
- On reconfigurable architectures FPGAs and CGRAs
- Spatial compiler lowers user programs into synthesizable Chisel [Bachrach, et al.]

[Koeplinger, et al.] "Spatial: a language and compiler for application accelerators." PLDI 2018
[Bachrach, et al.] "Chisel: constructing hardware in a Scala embedded language." DAC 2012.

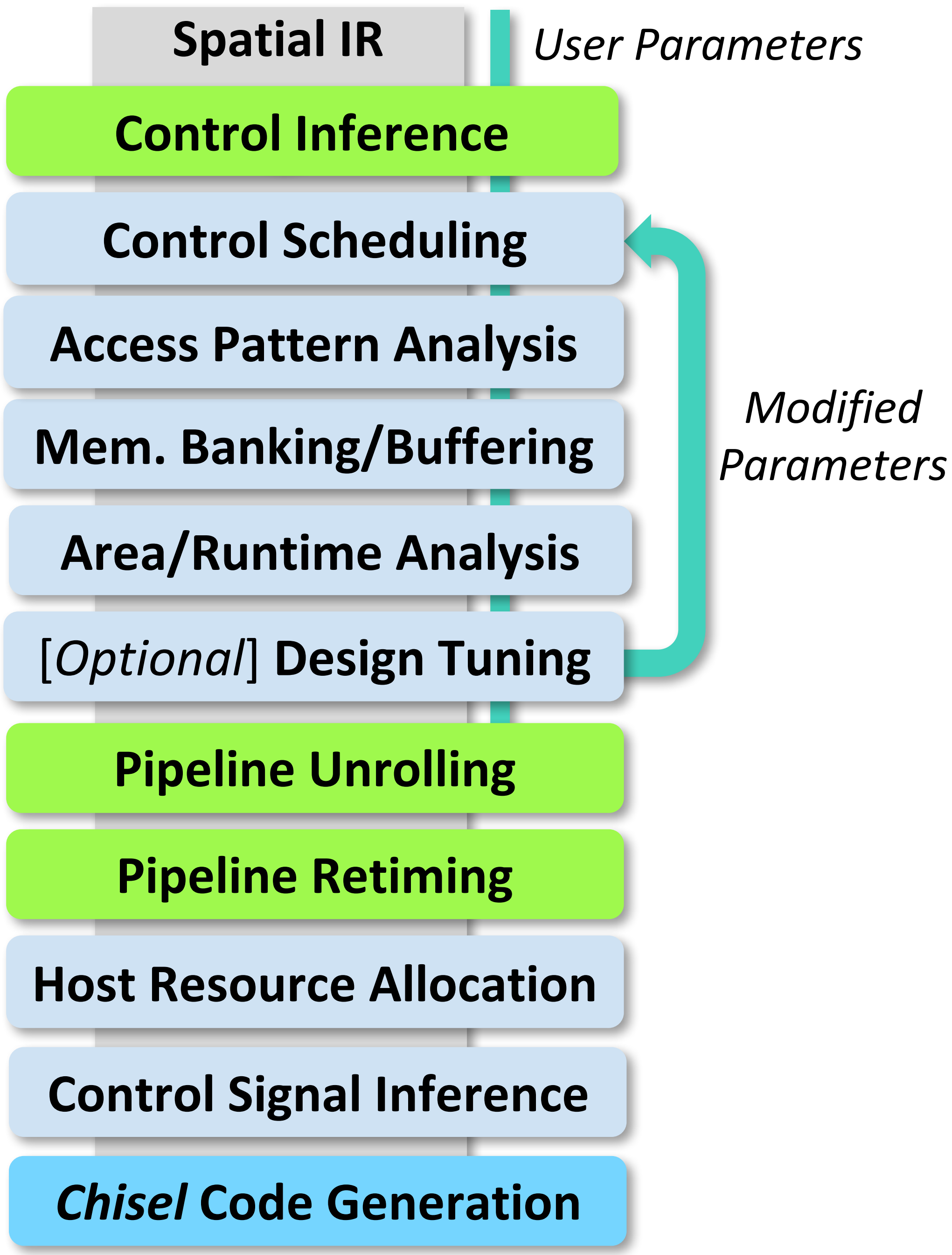
The Spatial Compiler

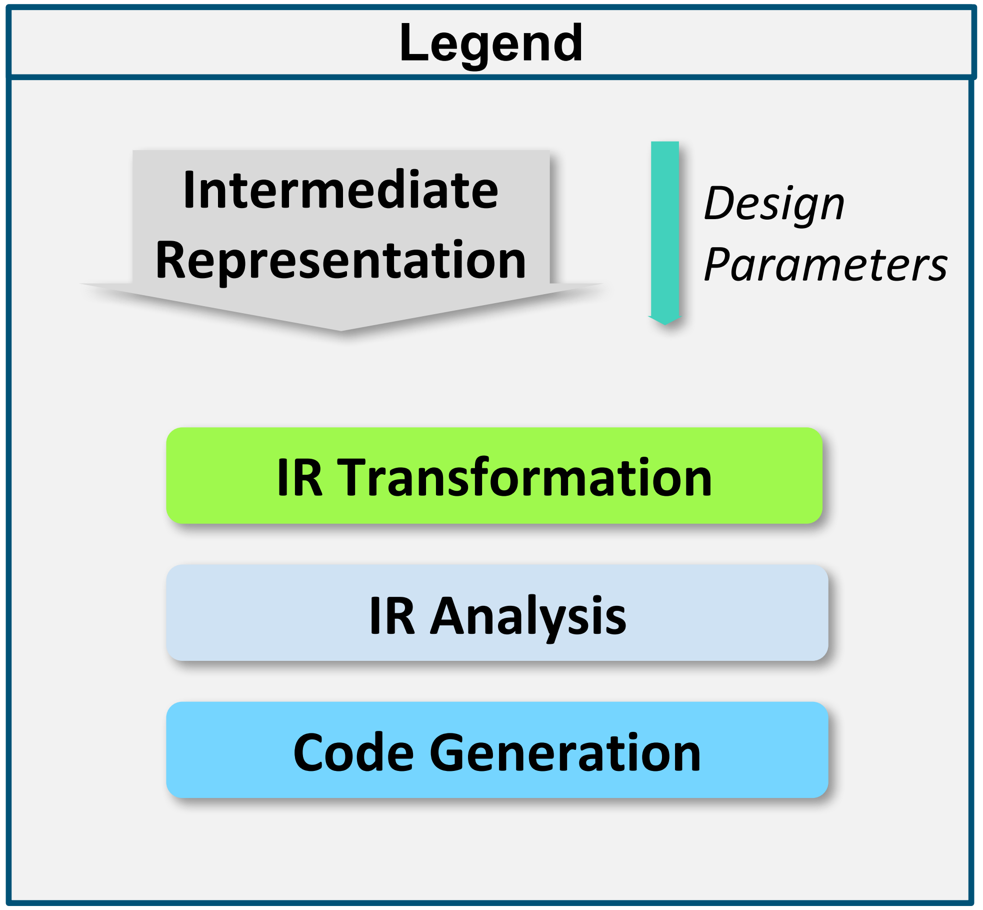


- Goal of Spatial [Koeplinger, et al.]: design of application accelerators
- On reconfigurable architectures FPGAs and CGRAs
- Spatial compiler lowers user programs into synthesizable Chisel [Bachrach, et al.]

[Koeplinger, et al.] "Spatial: a language and compiler for application accelerators." PLDI 2018
[Bachrach, et al.] "Chisel: constructing hardware in a Scala embedded language." DAC 2012.

The Spatial Compiler



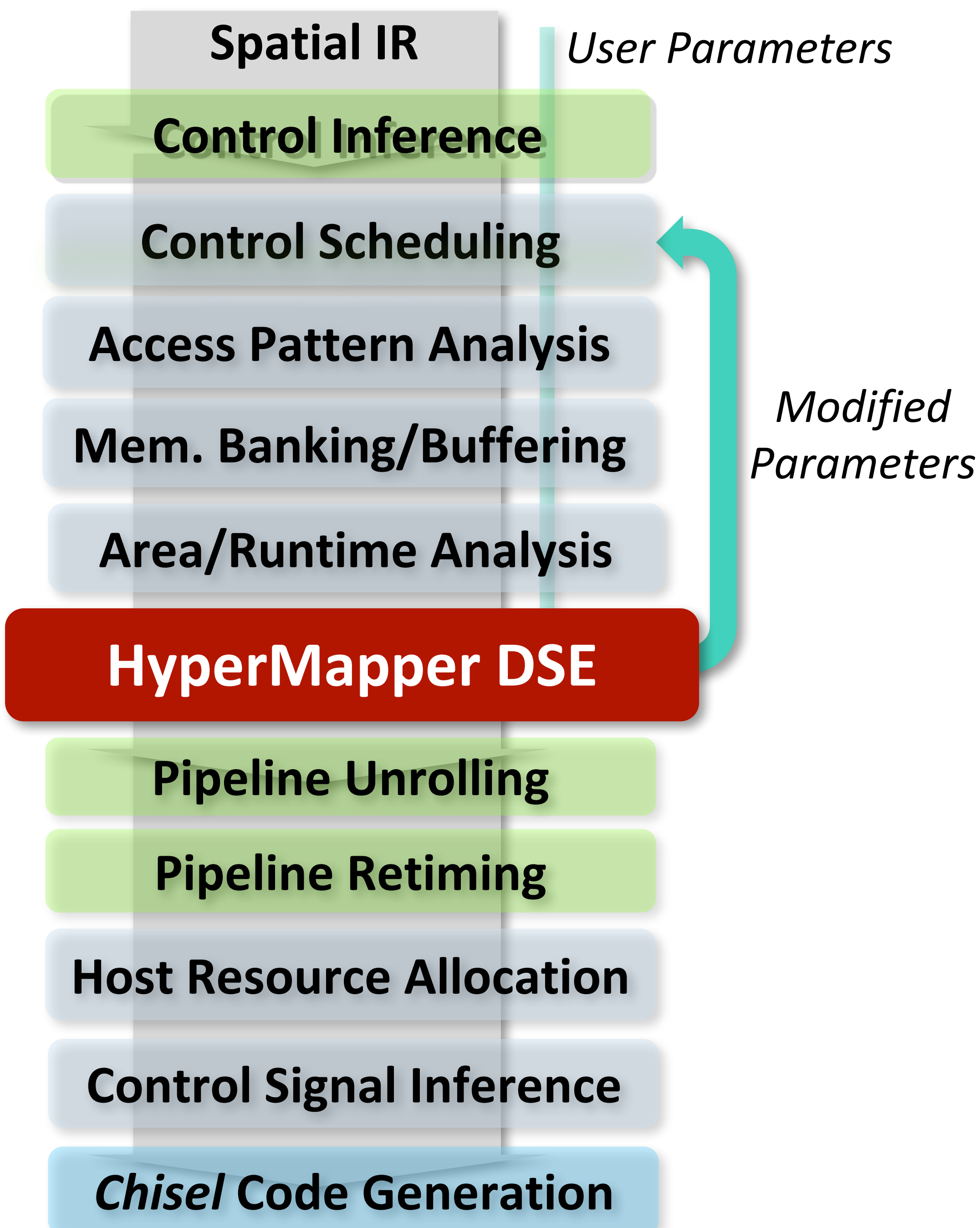


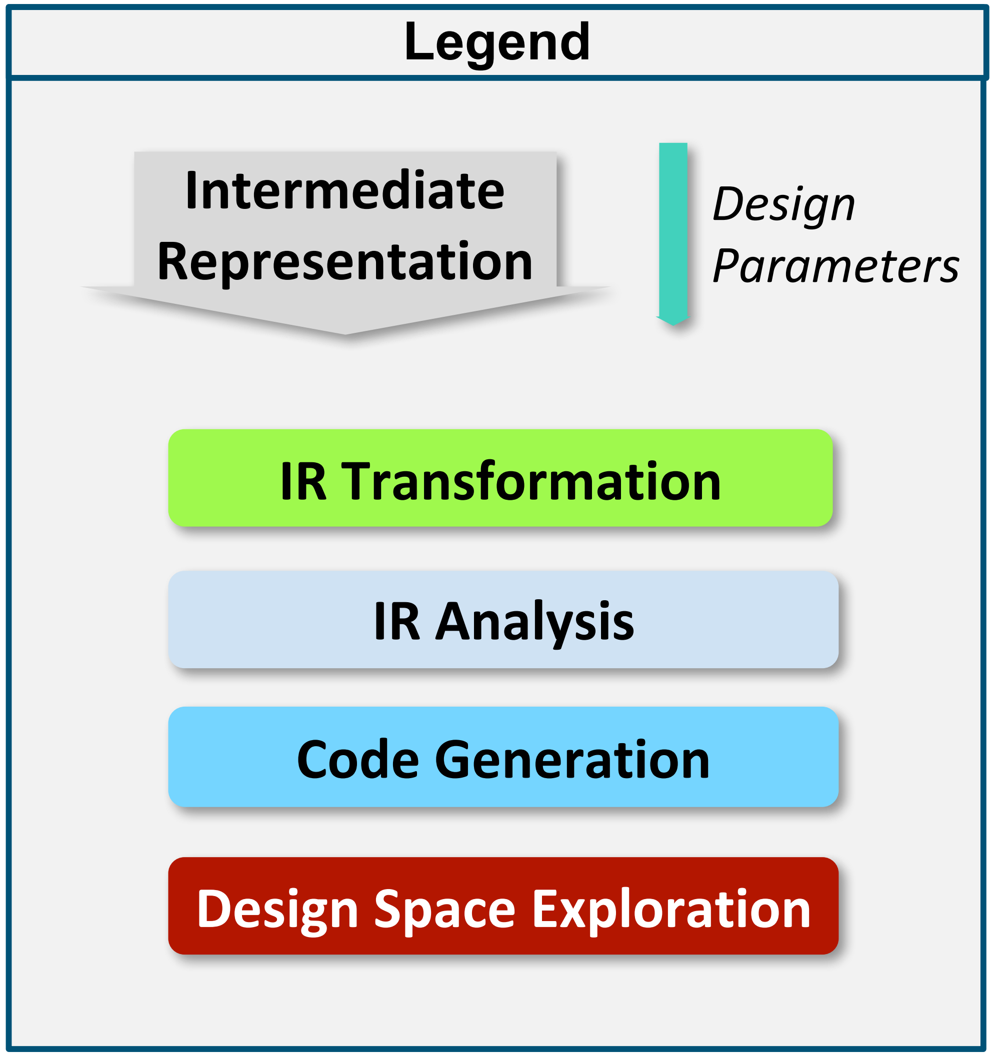
- Goal of Spatial [Koeplinger, et al.]: design of application accelerators
- On reconfigurable architectures FPGAs and CGRAs
- Spatial compiler lowers user programs into synthesizable Chisel [Bachrach, et al.]

[Koeplinger, et al.] "Spatial: a language and compiler for application accelerators." PLDI 2018
[Bachrach, et al.] "Chisel: constructing hardware in a Scala embedded language." DAC 2012.

The Spatial Compiler

HyperMapper





Search Space - Optimization Knobs

Input

The Spatial compiler automatically provides the following parameters:

- Tile size (ordinal)
- Inner and outer loop pipelining (ordinal)
- Meta-pipe (categorical)
- Unrolling factor (ordinal)
- Memory banking (ordinal)
- Parallelism (categorical)

Output

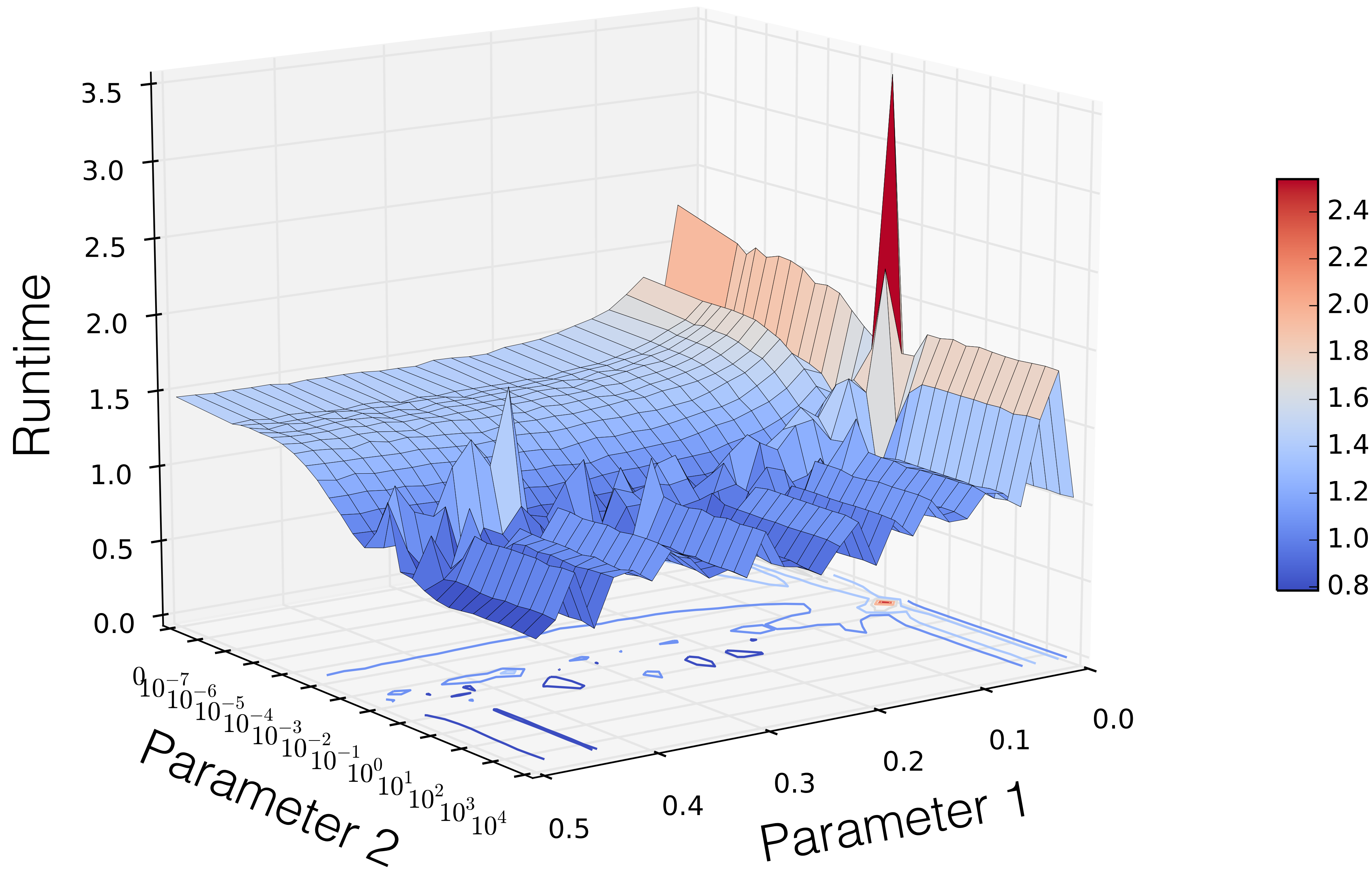
The compiler evaluation provides:

- Minimize Clock cycles (runtime): <u>objective 1</u>
- Minimize FPGA logic utilization: <u>objective 2</u>
    - Useful for fitting multiple applications on the same FPGA
    - Proxy for energy consumption
- <u>Feasibility constraint</u>:
    - true if design fits in the chip

Spatial Examples
- Search Spaces -

Application	# Parameters	Space Size
BlackScholes	4	7.68×10^4
OuterProduct	5	1.66×10^7
DotProduct	5	1.18×10^8
K-Means	6	1.04×10^6
GEMM	13	2.9×10^7
TPC-H Q6	5	3.54×10^9
GDA	9	2.40×10^{11}
Shallow CNN	7	1.2×10^6
Deep CNN	7	1.2×10^6
MD Grid	10	1.6×10^9

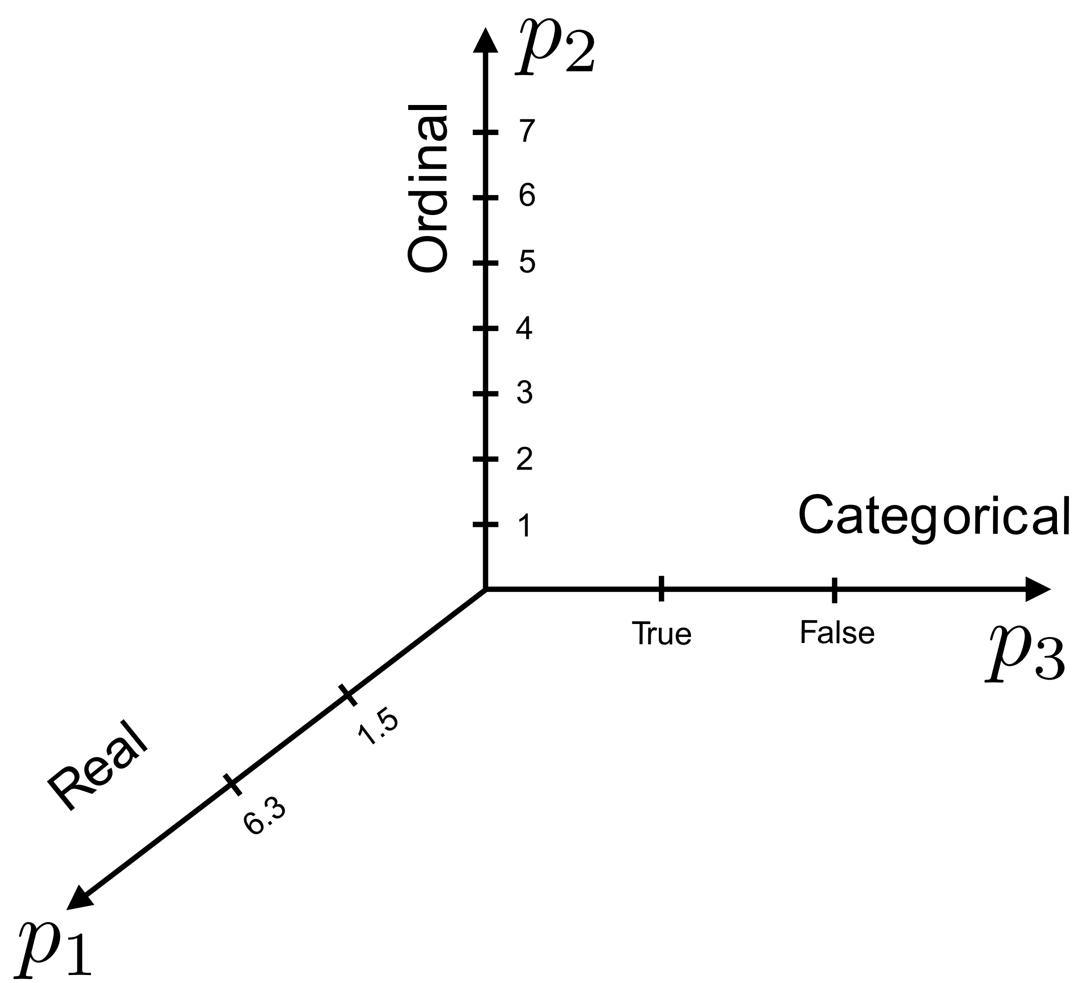
Motivation - Mono-objective



- Benchmark: SLAMBench 1.0 runtime response surface is: <u>non-linear</u>, <u>multi-modal</u> and <u>non-smooth</u>

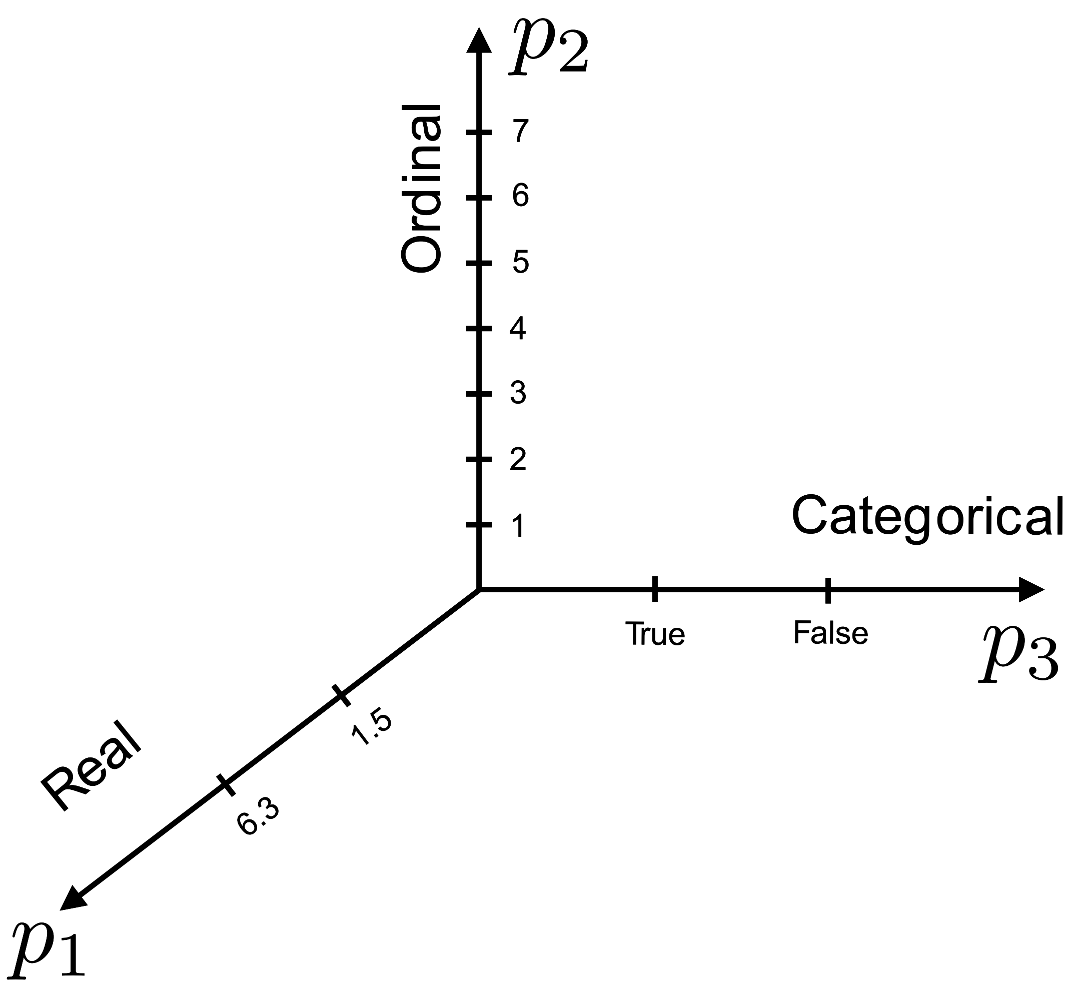
Design Space Exploration (DSE)
3-parameters and 2-objectives - Pictorial

Input space
(a.k.a. search or design space)

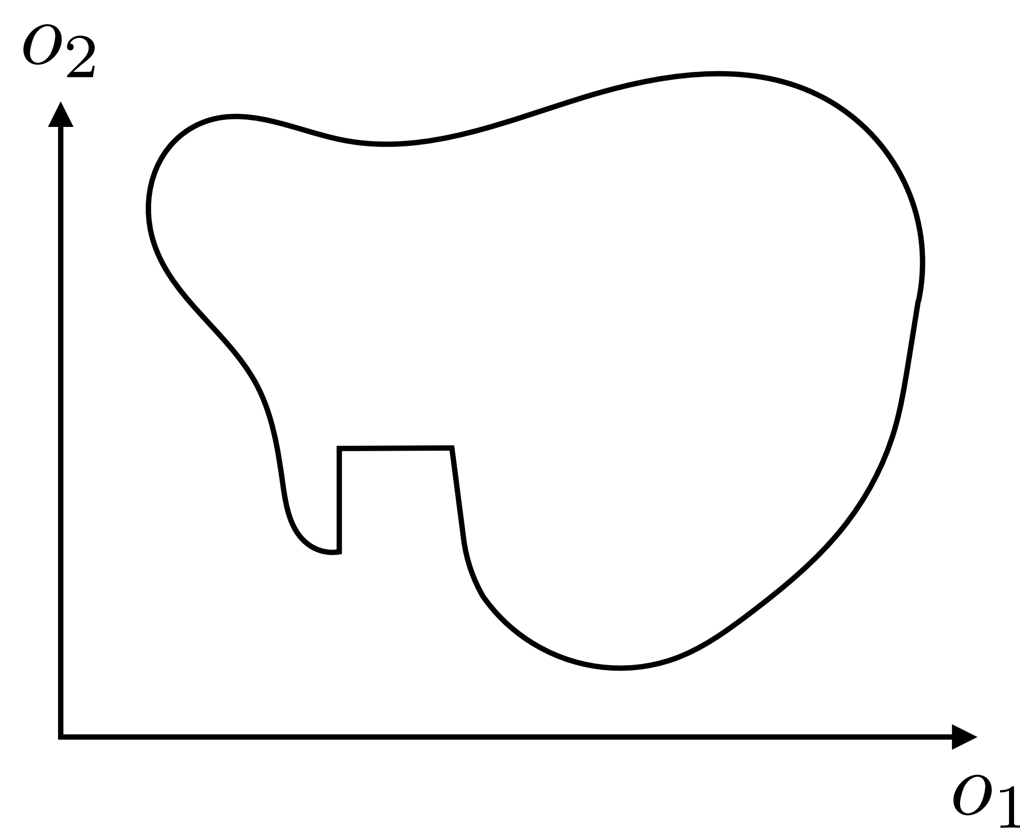


Design Space Exploration (DSE)
3-parameters and 2-objectives - Pictorial

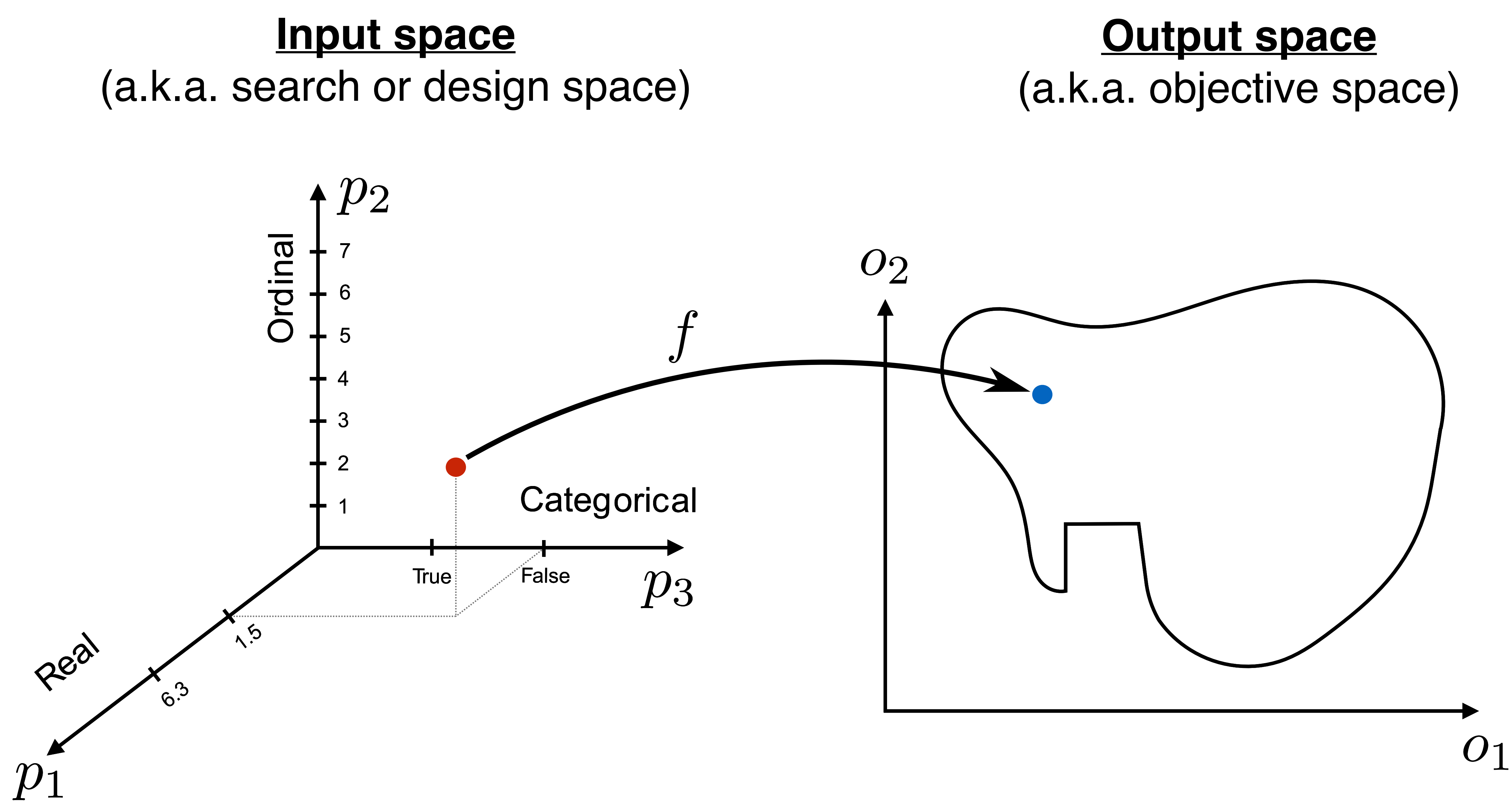
Input space
(a.k.a. search or design space)



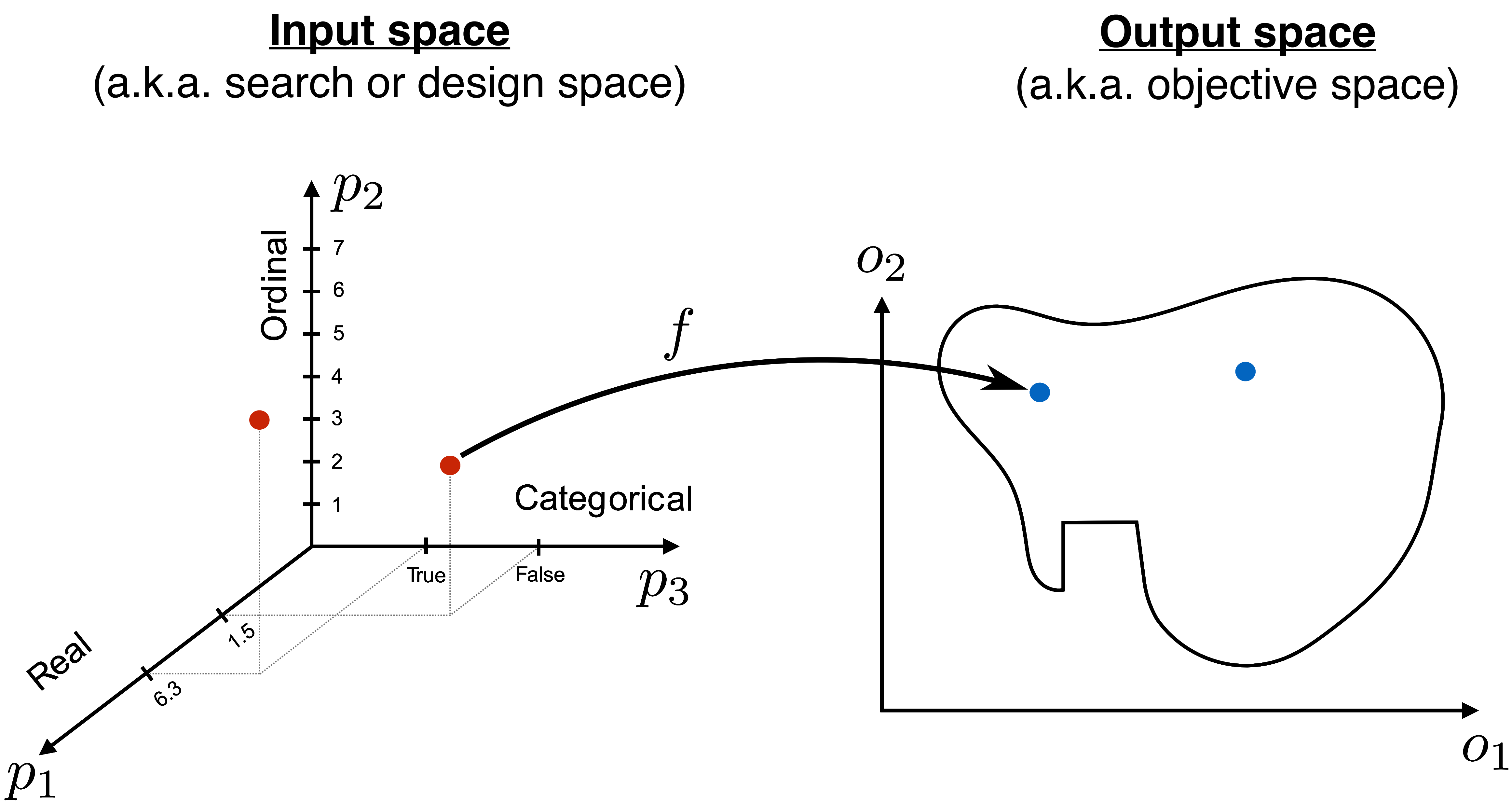
Output space
(a.k.a. objective space)



Design Space Exploration (DSE)
3-parameters and 2-objectives - Pictorial



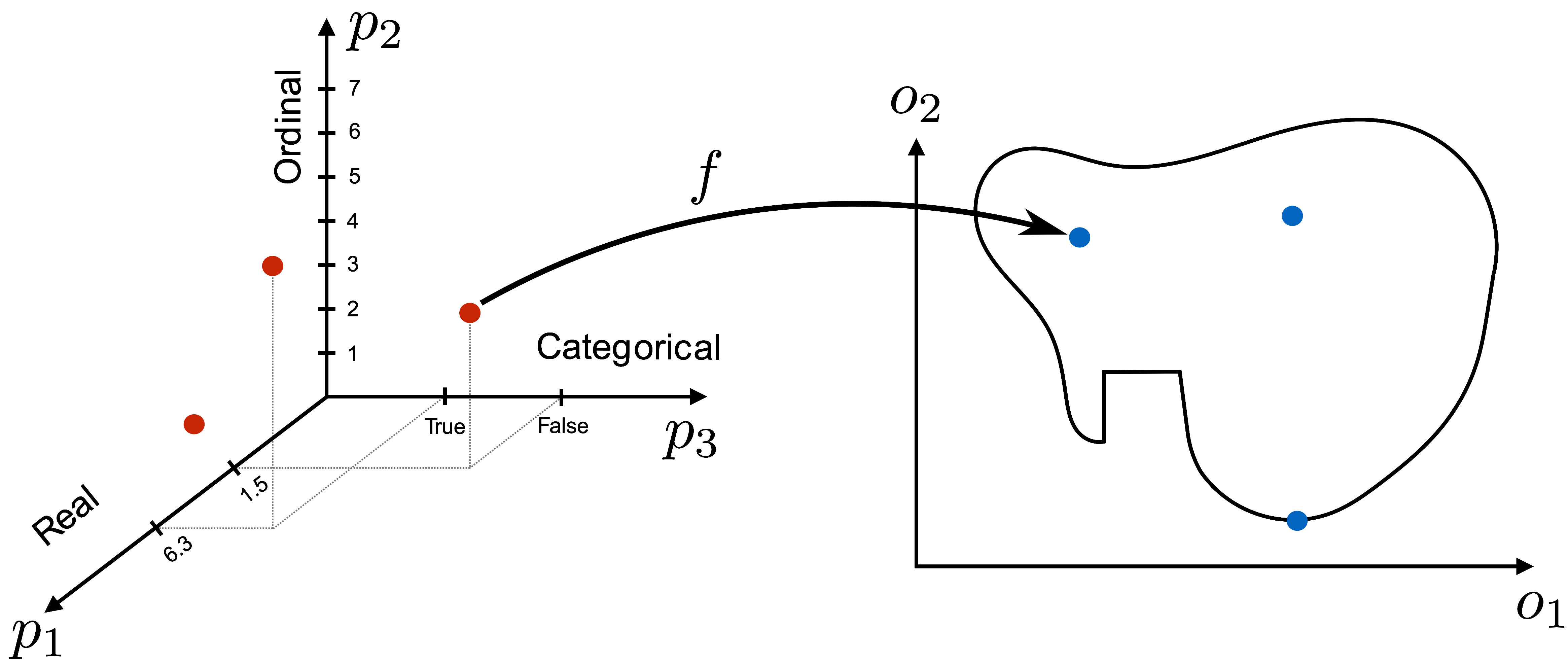
Design Space Exploration (DSE)
3-parameters and 2-objectives - Pictorial



Design Space Exploration (DSE)
3-parameters and 2-objectives - Pictorial

Input space
(a.k.a. search or design space)

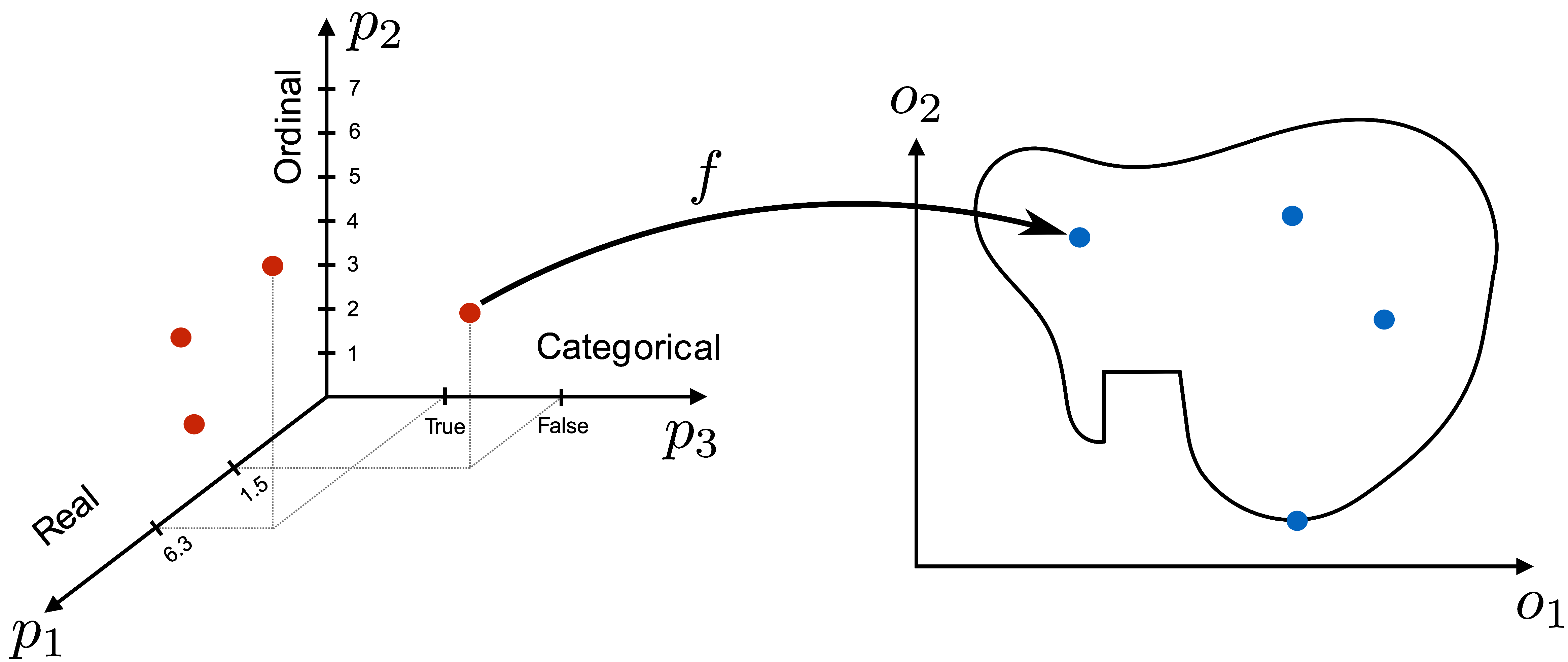
Output space
(a.k.a. objective space)



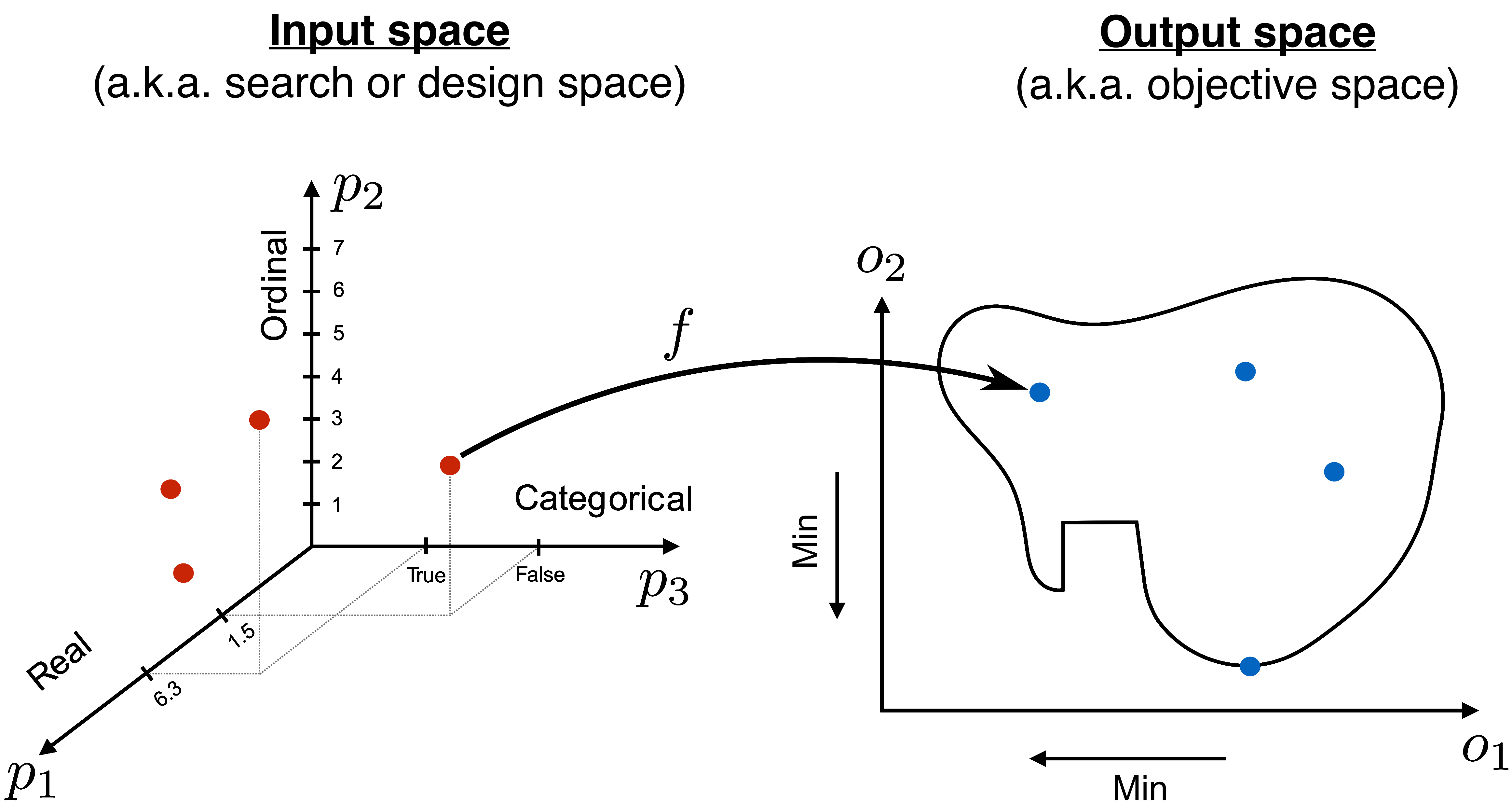
Design Space Exploration (DSE)
3-parameters and 2-objectives - Pictorial

Input space
(a.k.a. search or design space)

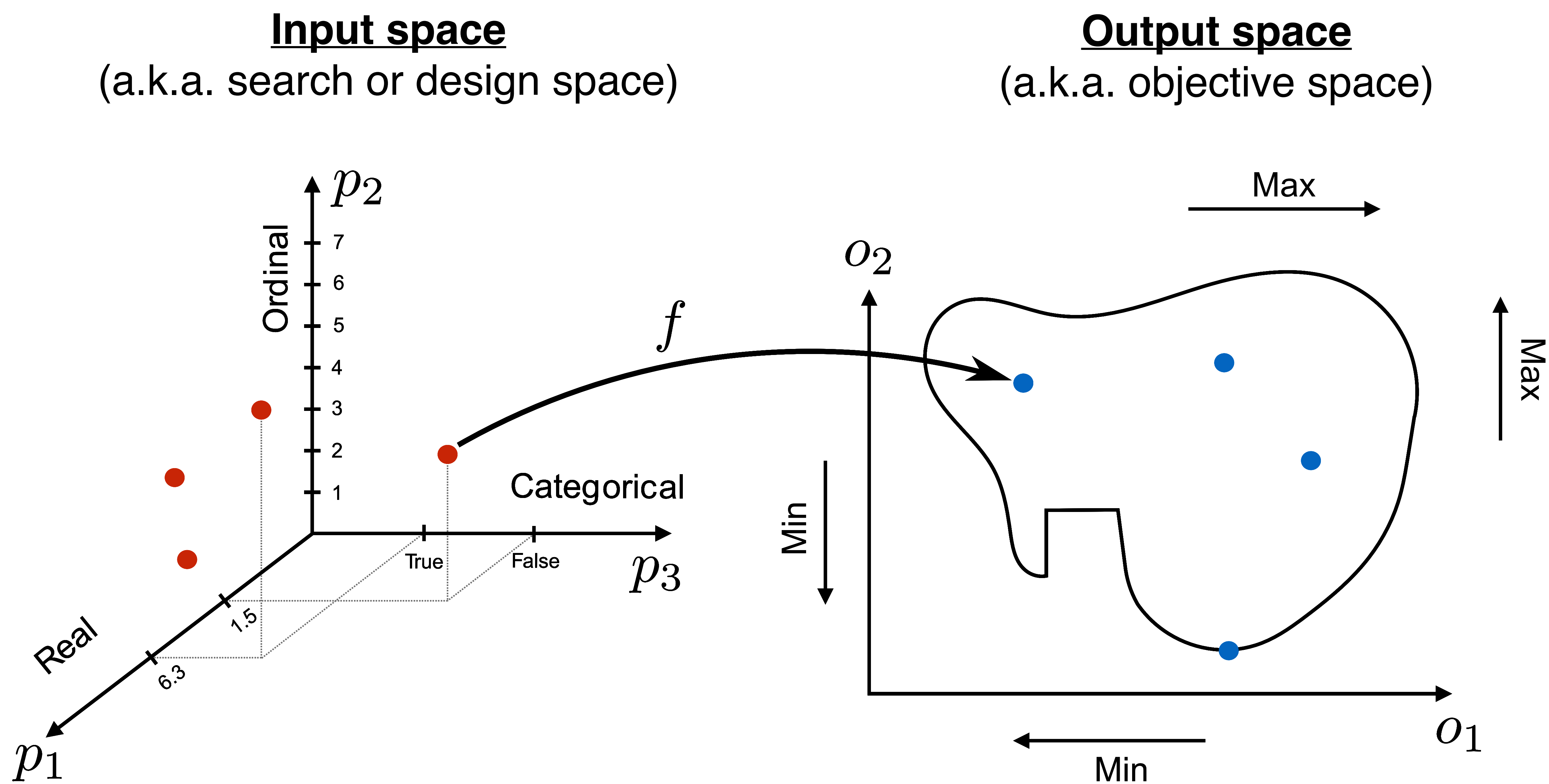
Output space
(a.k.a. objective space)



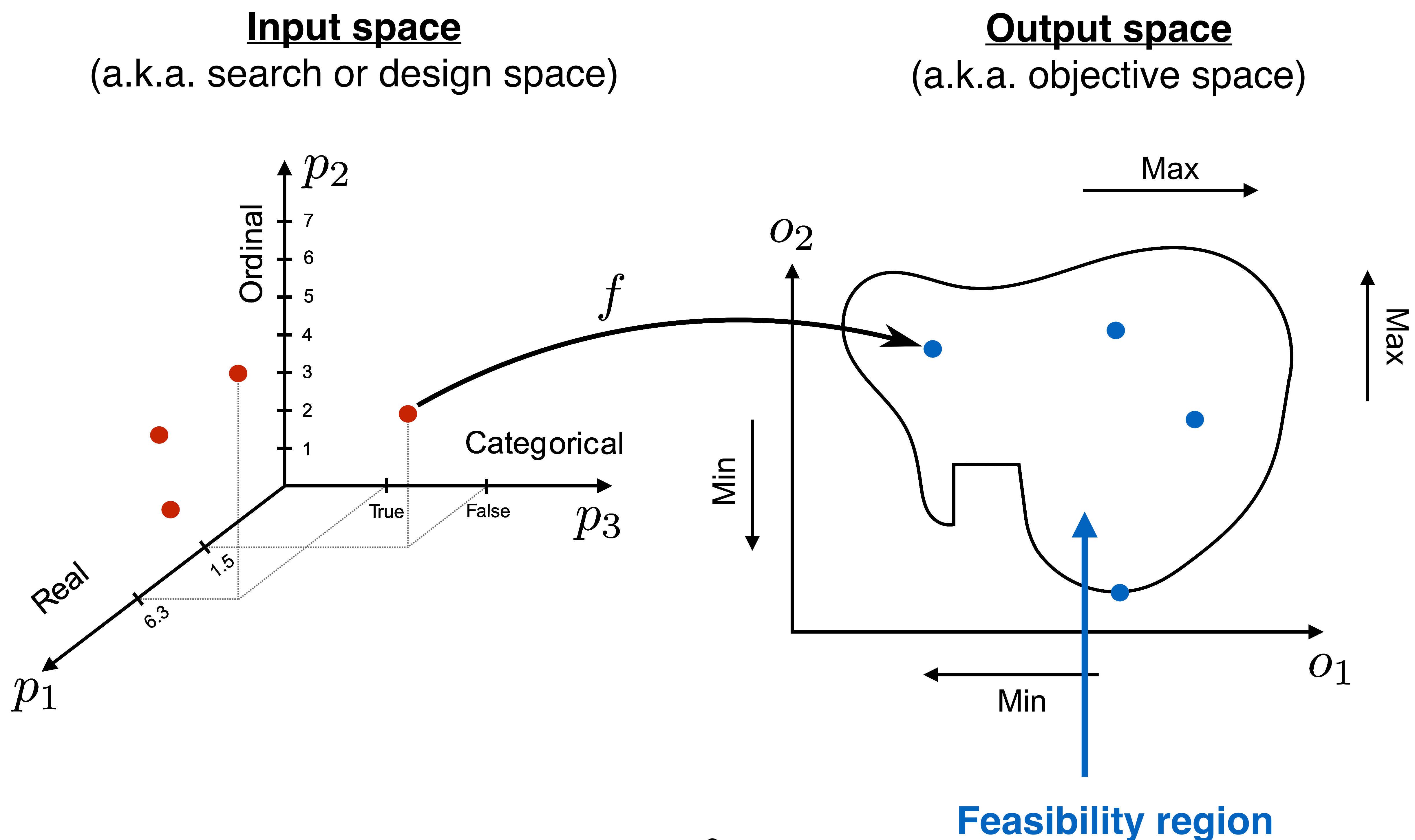
Design Space Exploration (DSE)
3-parameters and 2-objectives - Pictorial



Design Space Exploration (DSE)
3-parameters and 2-objectives - Pictorial



Design Space Exploration (DSE) 3-parameters and 2-objectives - Pictorial



Practical DSE: Important Features

1. Real, integer, ordinal and categorical variables (RIOC var.)
    - Example: tile size is an ordinal, parallelism is a categorical

Practical DSE: Important Features

1. Real, integer, ordinal and categorical variables (RIOC var.)
   - Example: tile size is an ordinal, parallelism is a categorical
2. User prior knowledge in the search (Prior)
   - Example: # threads on a CPU with 4 cores? Gaussian-shaped around 6

Practical DSE: Important Features

1. Real, integer, ordinal and categorical variables (RIOC var.)
   - Example: tile size is an ordinal, parallelism is a categorical
2. User prior knowledge in the search (Prior)
   - Example: # threads on a CPU with 4 cores? Gaussian-shaped around 6
3. Unknown feasibility constraints (Constr.)
   - Example: does a design fit in the FPGA?

Practical DSE: Important Features

1. Real, integer, ordinal and categorical variables (RIOC var.)
    - Example: tile size is an ordinal, parallelism is a categorical
2. User prior knowledge in the search (Prior)
    - Example: # threads on a CPU with 4 cores? Gaussian-shaped around 6
3. Unknown feasibility constraints (Constr.)
    - Example: does a design fit in the FPGA?
4. Multi-objective optimization (Multi)
    - Example: trade-off runtime and area

DSE Tools Taxonomy

None of the tools available support all these DSE features

We introduce a new framework dubbed HyperMapper

Name	Multi	RIOC var.	Constr.	Prior
GpyOpt	✗	✗	✗	✗
OpenTuner	✗	✓	✗	✗
SURF	✗	✓	✗	✗
SMAC	✗	✓	✗	✗
Spearmint	✗	✗	✓	✗
Hyperopt	✗	✓	✗	✓
Hyperband	✗	✓	✗	✗
GPflowOpt	✓	✗	✓	✗
cBO	✗	✗	✓	✗
BOHB	✗	✓	✗	✗
HyperMapper	✓	✓	✓	✓

[Bodin, et al.] "Integrating algorithmic parameters into benchmarking and design space exploration in 3D scene understanding", PACT, 2016.
[Nardi, et al.] "Algorithmic performance-accuracy trade-off in 3D vision applications using HyperMapper", IPDPS-iWAPT, 2017.
[Nardi, et al., 2019] "Practical Design Space Exploration", MASCOTS, 2019.

DSE Solutions

Approaches	Behavior
• Manual optimization	Human expert needed

DSE Solutions

Approaches	Behavior
• Manual optimization	Human expert needed
• Automated but architecture-dependent	Doesn’t generalize

DSE Solutions

Approaches	Behavior
• Manual optimization	Human expert needed
• Automated but architecture-dependent	Doesn't generalize
• Grid search	Unfeasible most of the times

DSE Solutions

Approaches	Behavior
• Manual optimization	Human expert needed
• Automated but architecture-dependent	Doesn’t generalize
• Grid search	Unfeasible most of the times
• Random search	Inefficient

DSE Solutions

Approaches	Behavior
• Manual optimization	Human expert needed
• Automated but architecture-dependent	Doesn’t generalize
• Grid search	Unfeasible most of the times
• Random search	Inefficient
• Evolutionary Algorithms	High sampling budget

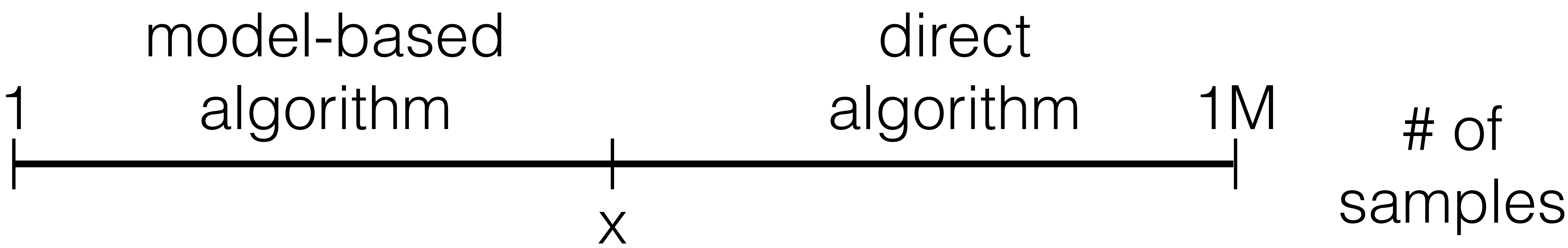
DSE Solutions

Approaches	Behavior
• Manual optimization	Human expert needed
• Automated but architecture-dependent	Doesn't generalize
• Grid search	Unfeasible most of the times
• Random search	Inefficient
• Evolutionary Algorithms	High sampling budget
• Bayesian Optimization (BO)	Sampling efficient

Diversity of Optimizers in HyperMapper

What type of algorithm to choose for the DSE?

- Rule of thumb: it depends on the sampling budget



- Direct algorithms:
  - Random sampling
  - Multi-start local search
  - Evolutionary Algorithms
- Model-based
  - Bayesian optimization

DSE Mono-objective Formulation
- Formal Problem Setting -

Find a global minimizer of an unknown objective function $f(x)$ under a set of q unknown feasibility constraint functions $c_i(x)$:

$$x^* = \arg\min_{x \in \mathbb{X}} \quad f(x)$$

$$\text{subject to} \qquad c_i(x) \leq b_i,\ i = 1, \ldots, q,$$

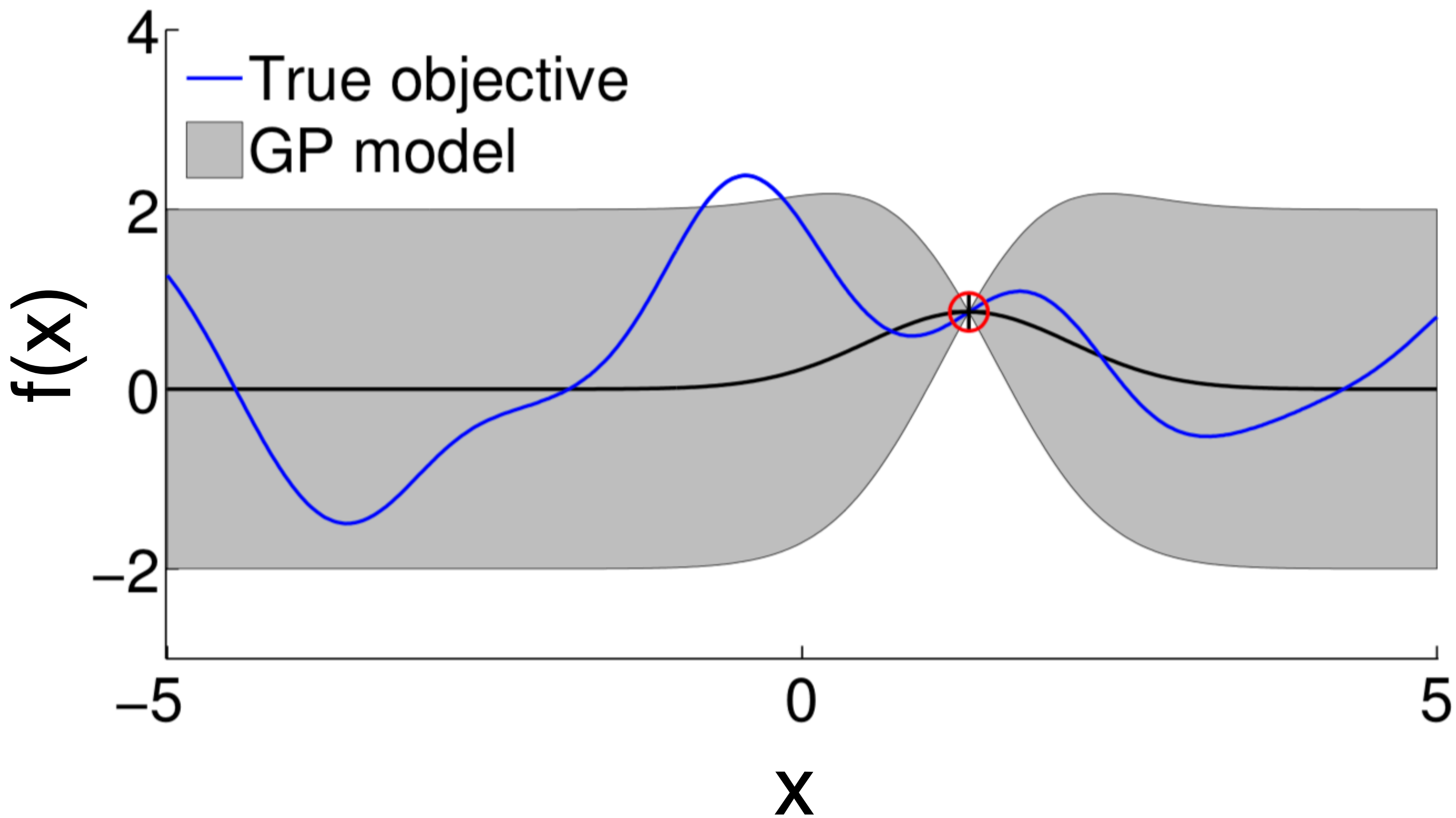
where $\mathbb{X}$ is some space of interest.

In addition:

1. The objective function $f(x)$ and the constraints $c_i(x)$:
   - Have no simple closed form (e.g., software)
   - Can be evaluated at any point x but evaluation is expensive (time)
2. Gradients are unavailable (or available at prohibitive cost)

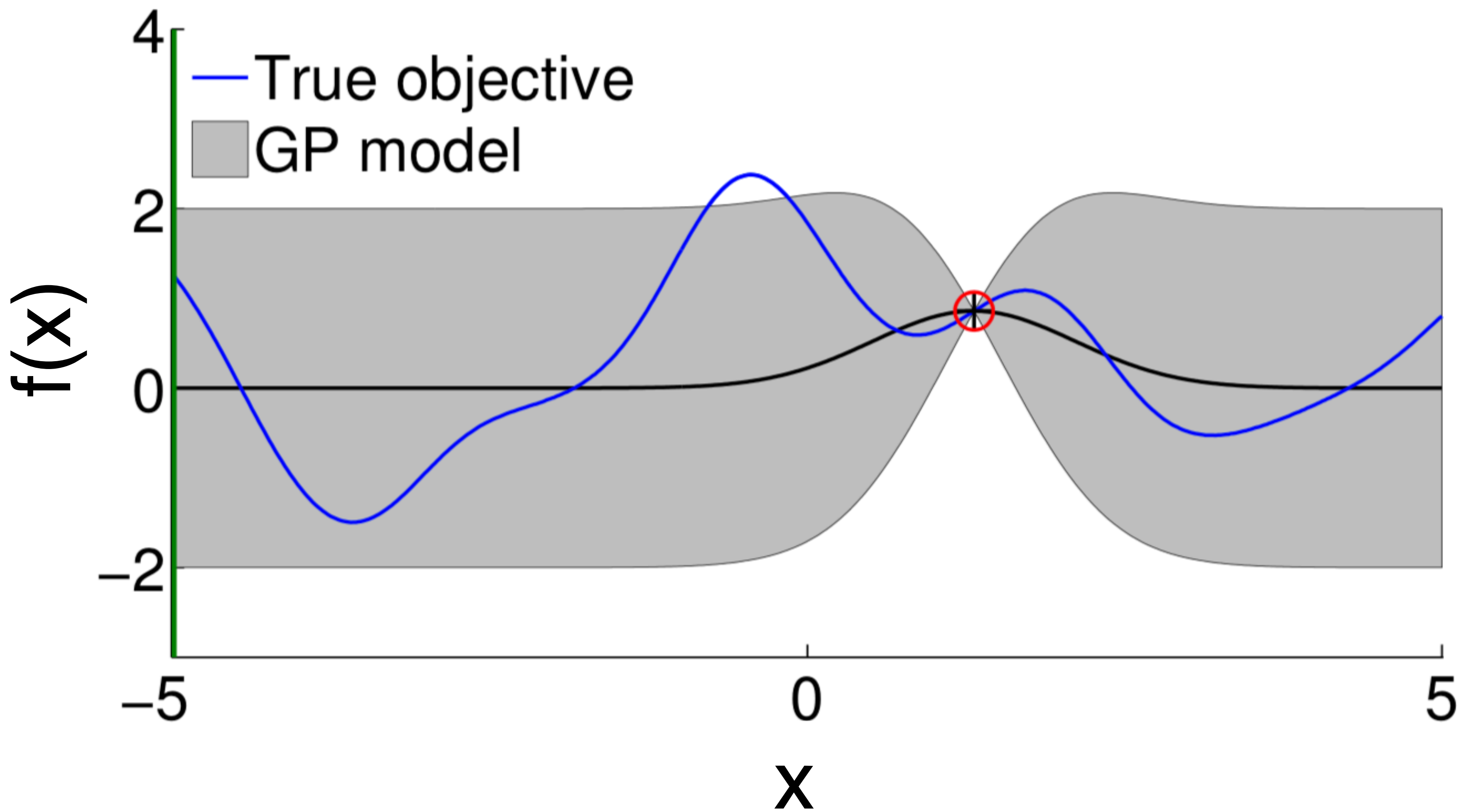
Optimization: Illustration

The model in this example is a Gaussian Process (GP)



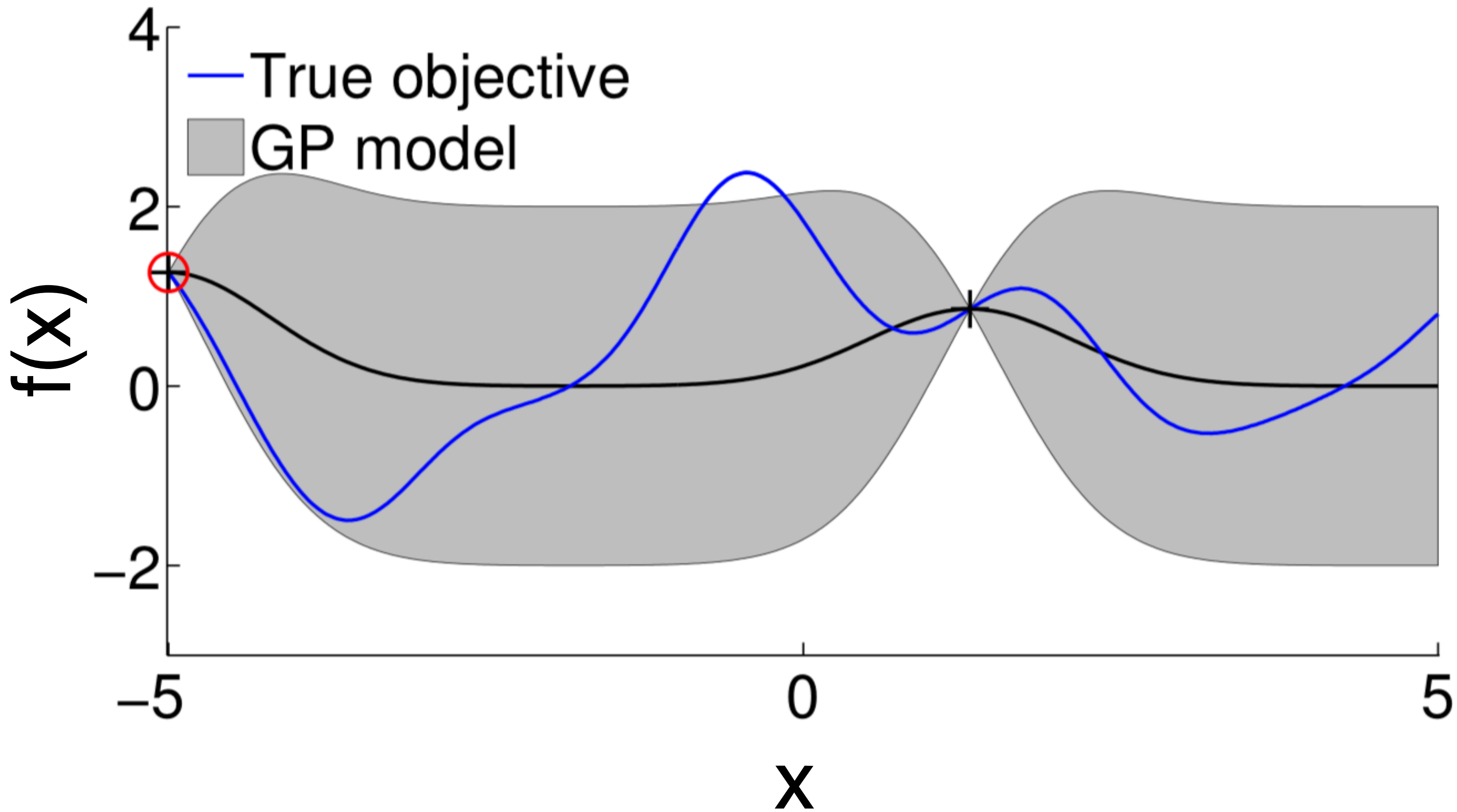
Optimization: Illustration

The model in this example is a Gaussian Process (GP)



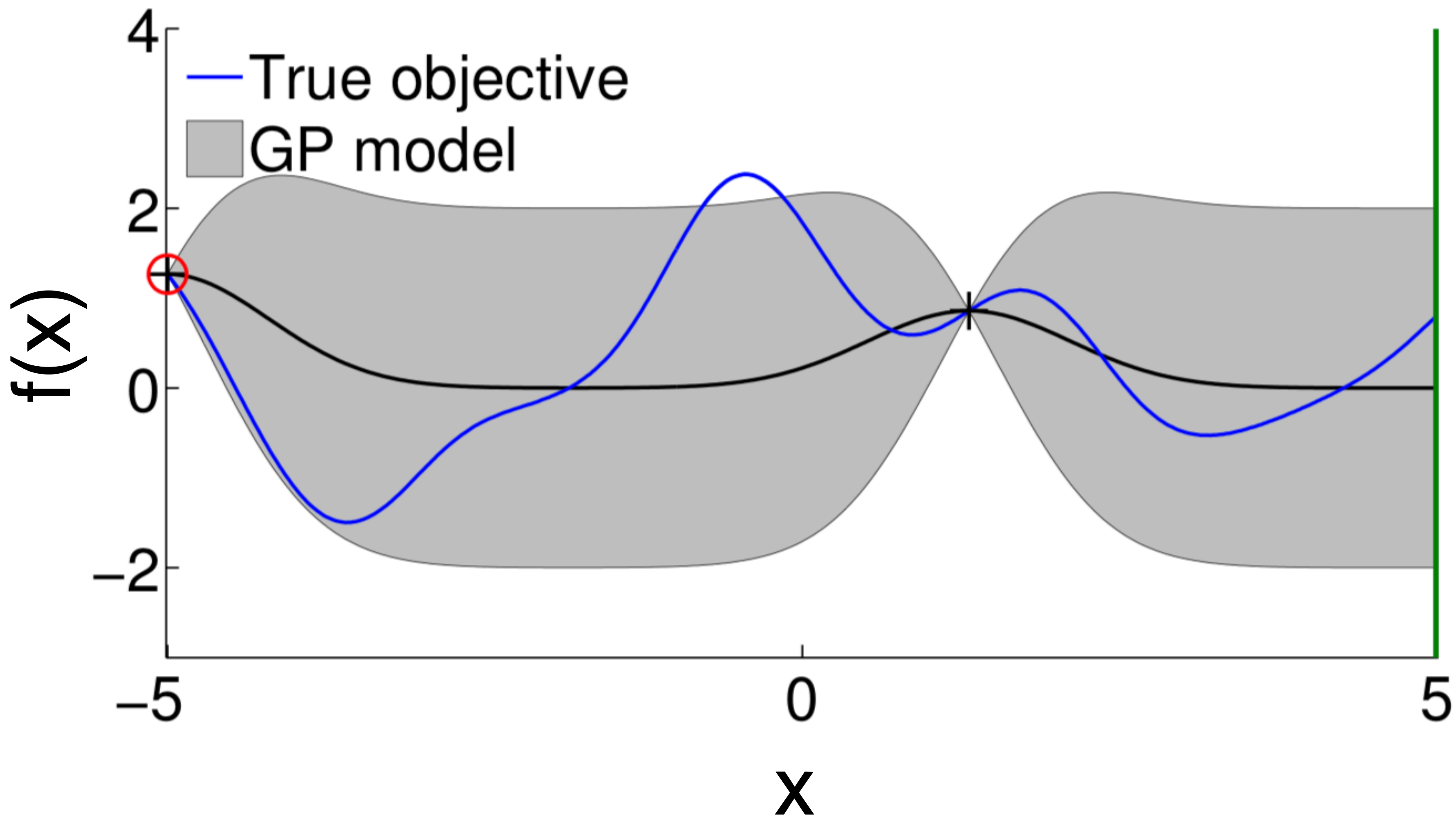
Optimization: Illustration

The model in this example is a Gaussian Process (GP)



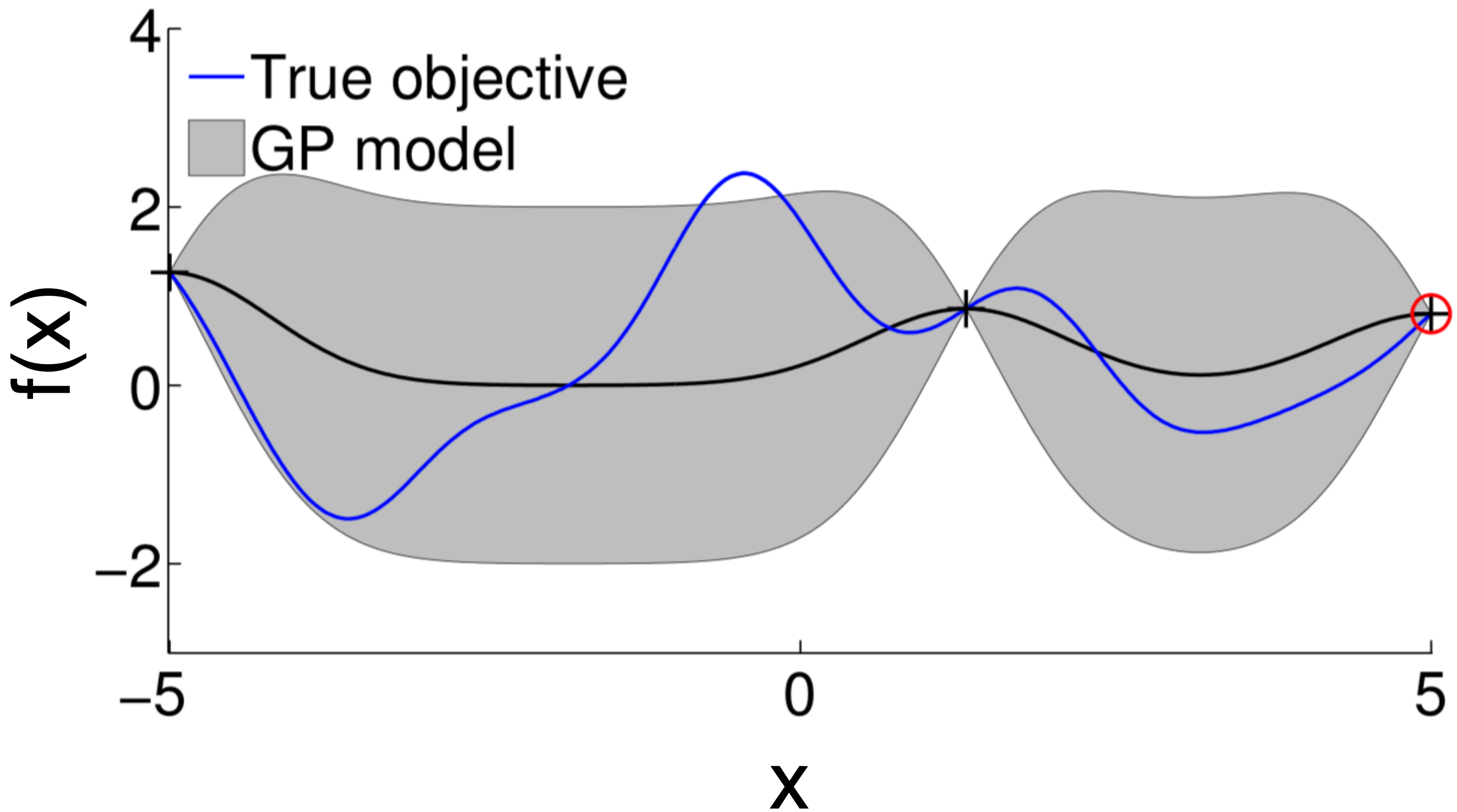
Optimization: Illustration

The model in this example is a Gaussian Process (GP)



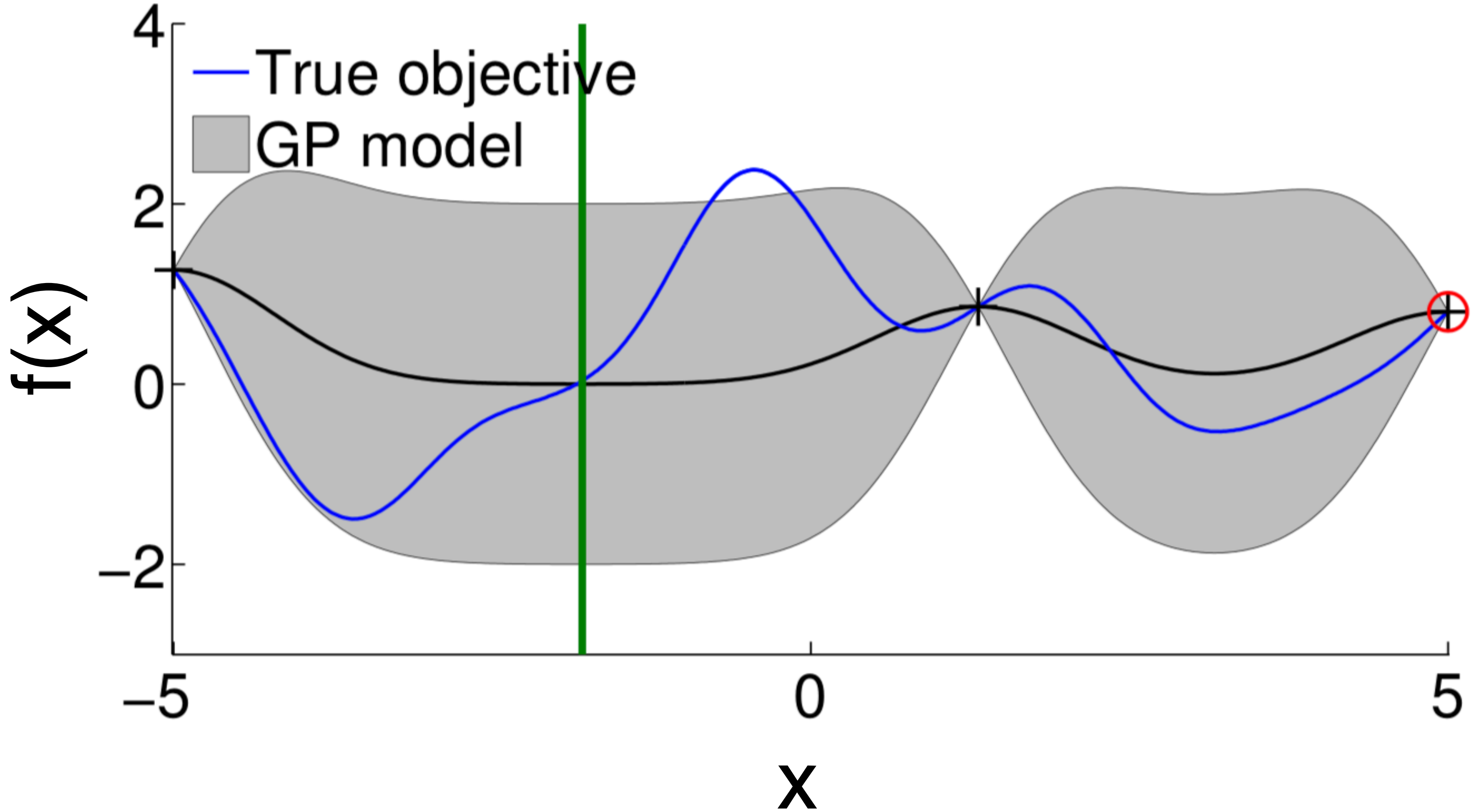
Optimization: Illustration

The model in this example is a Gaussian Process (GP)



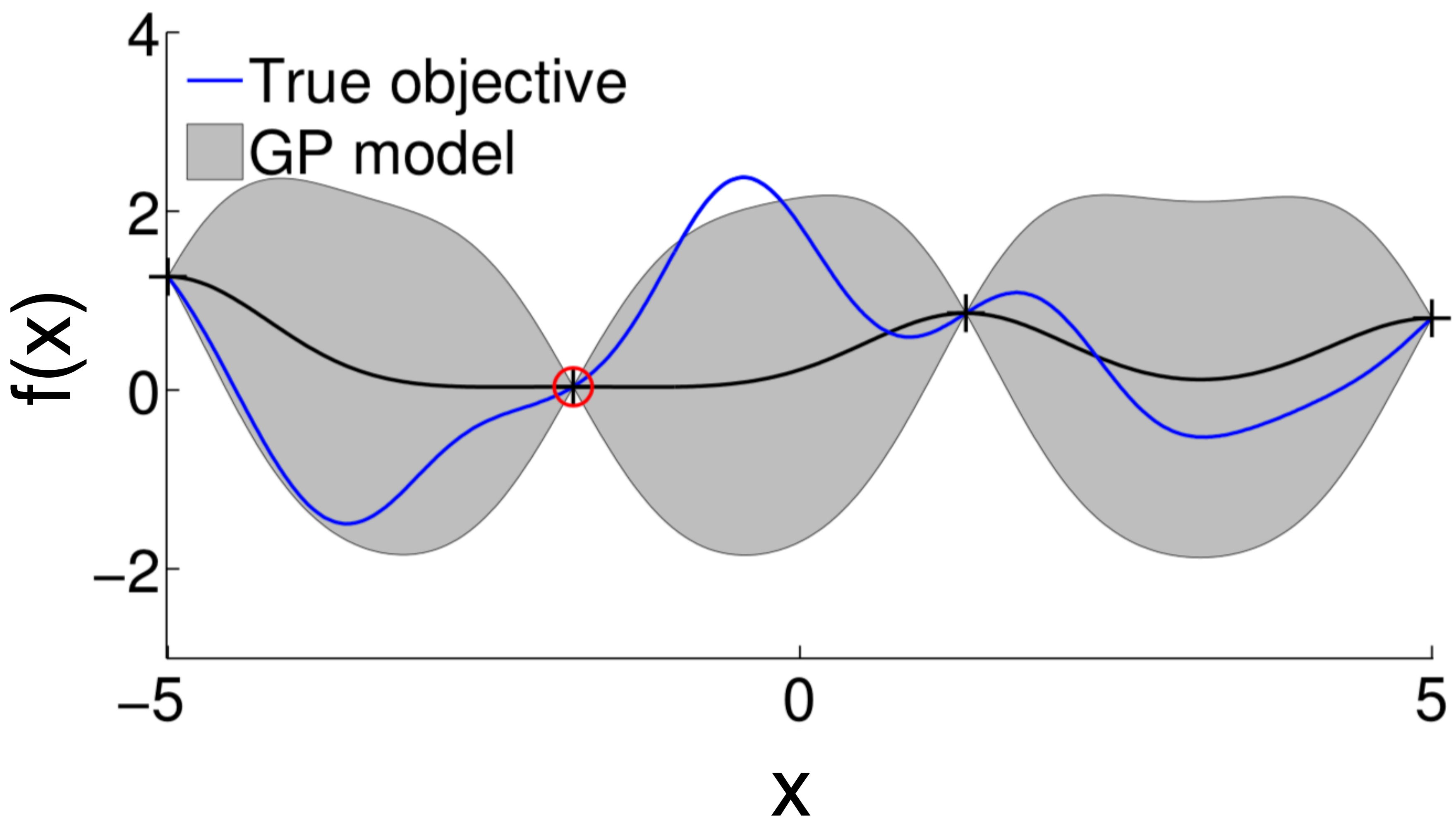
Optimization: Illustration

The model in this example is a Gaussian Process (GP)



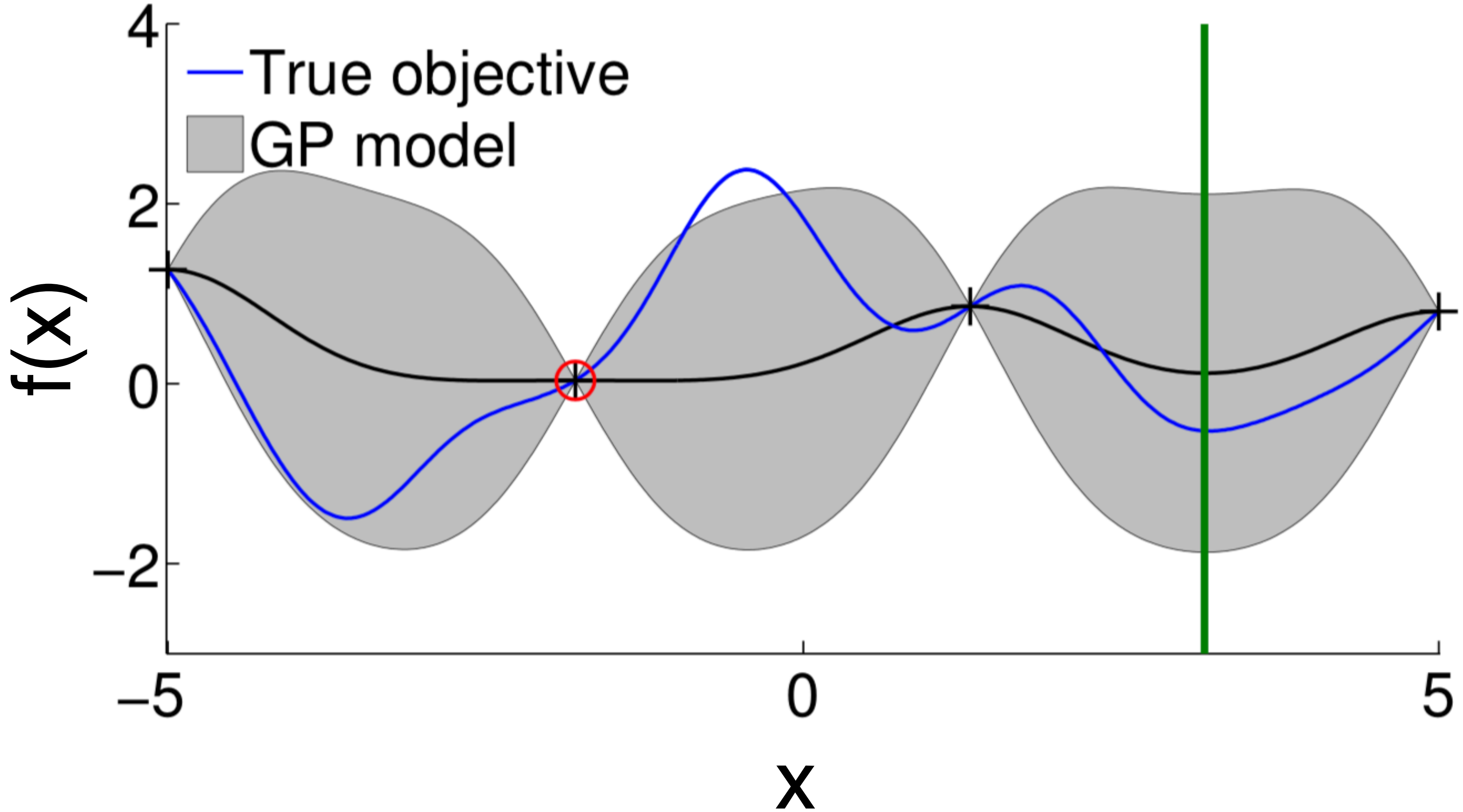
Optimization: Illustration

The model in this example is a Gaussian Process (GP)



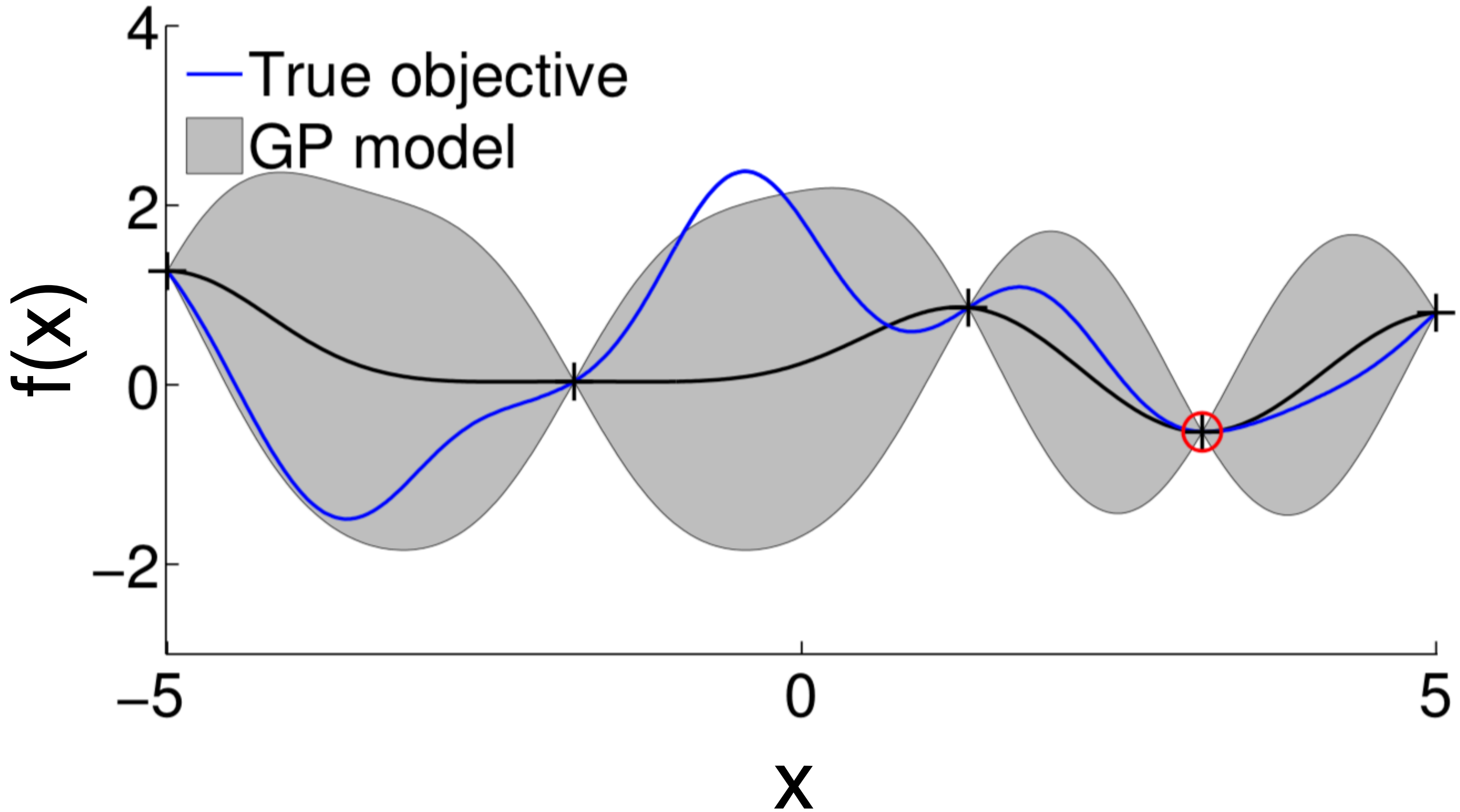
Optimization: Illustration

The model in this example is a Gaussian Process (GP)



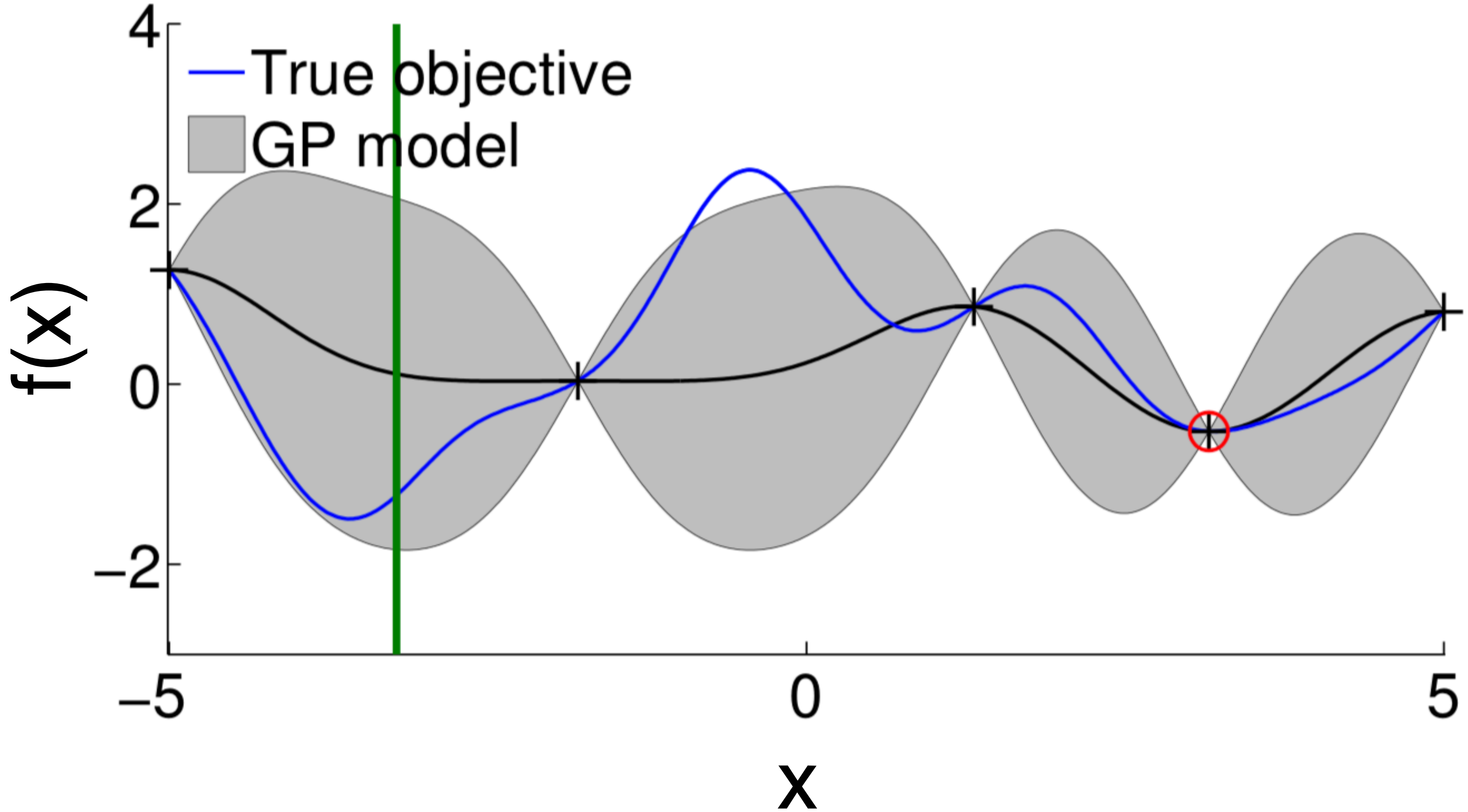
Optimization: Illustration

The model in this example is a Gaussian Process (GP)



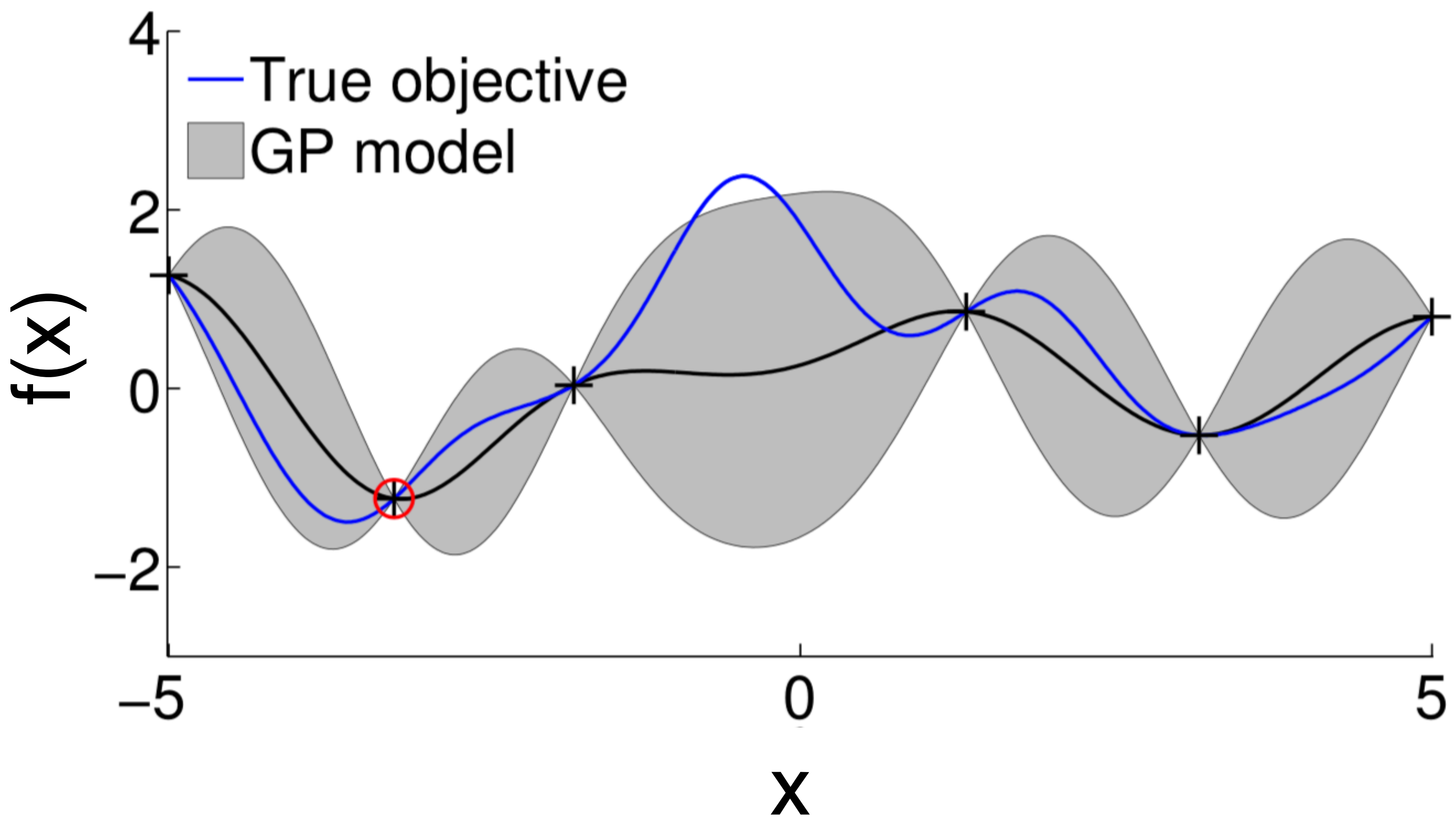
Optimization: Illustration

The model in this example is a Gaussian Process (GP)



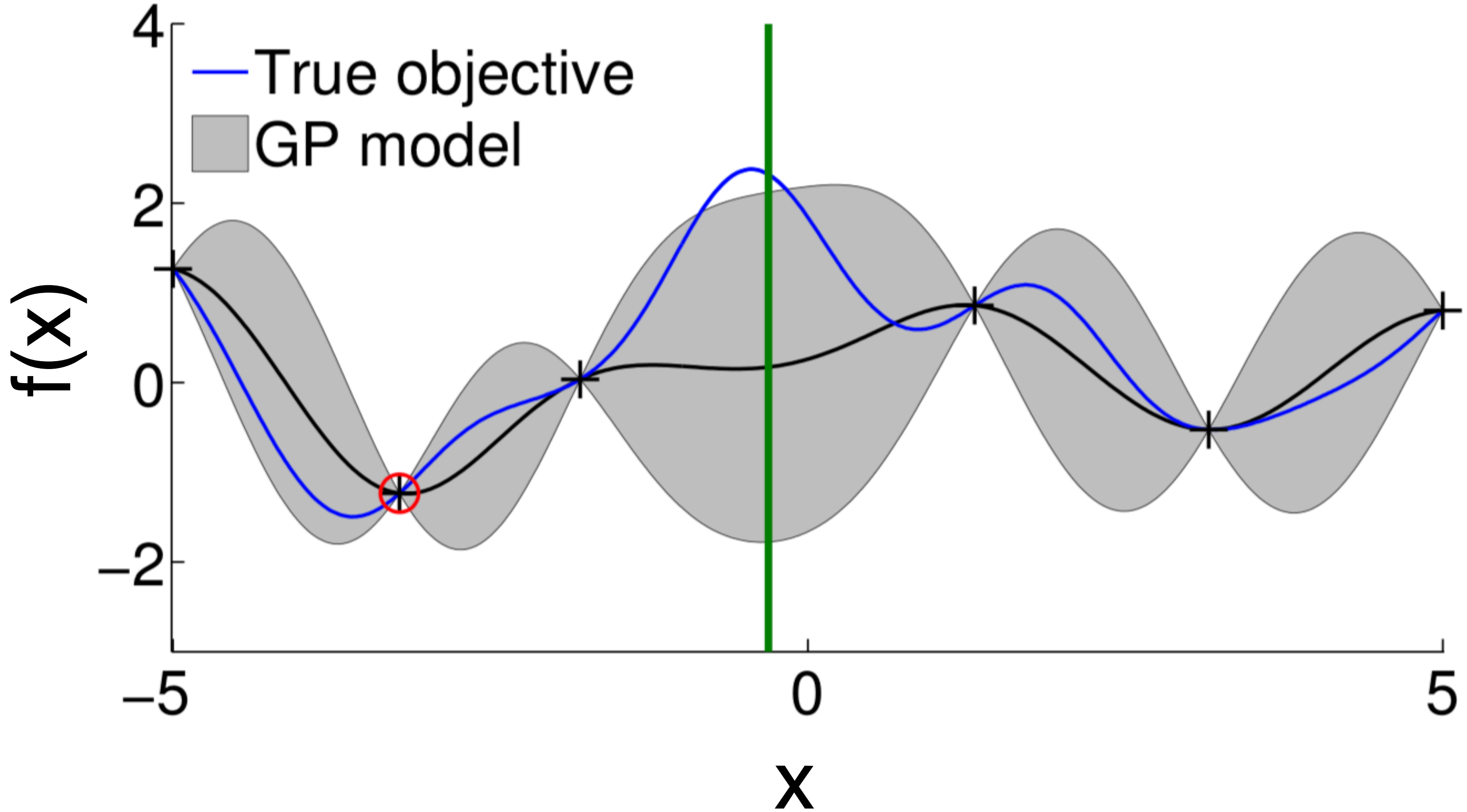
Optimization: Illustration

The model in this example is a Gaussian Process (GP)



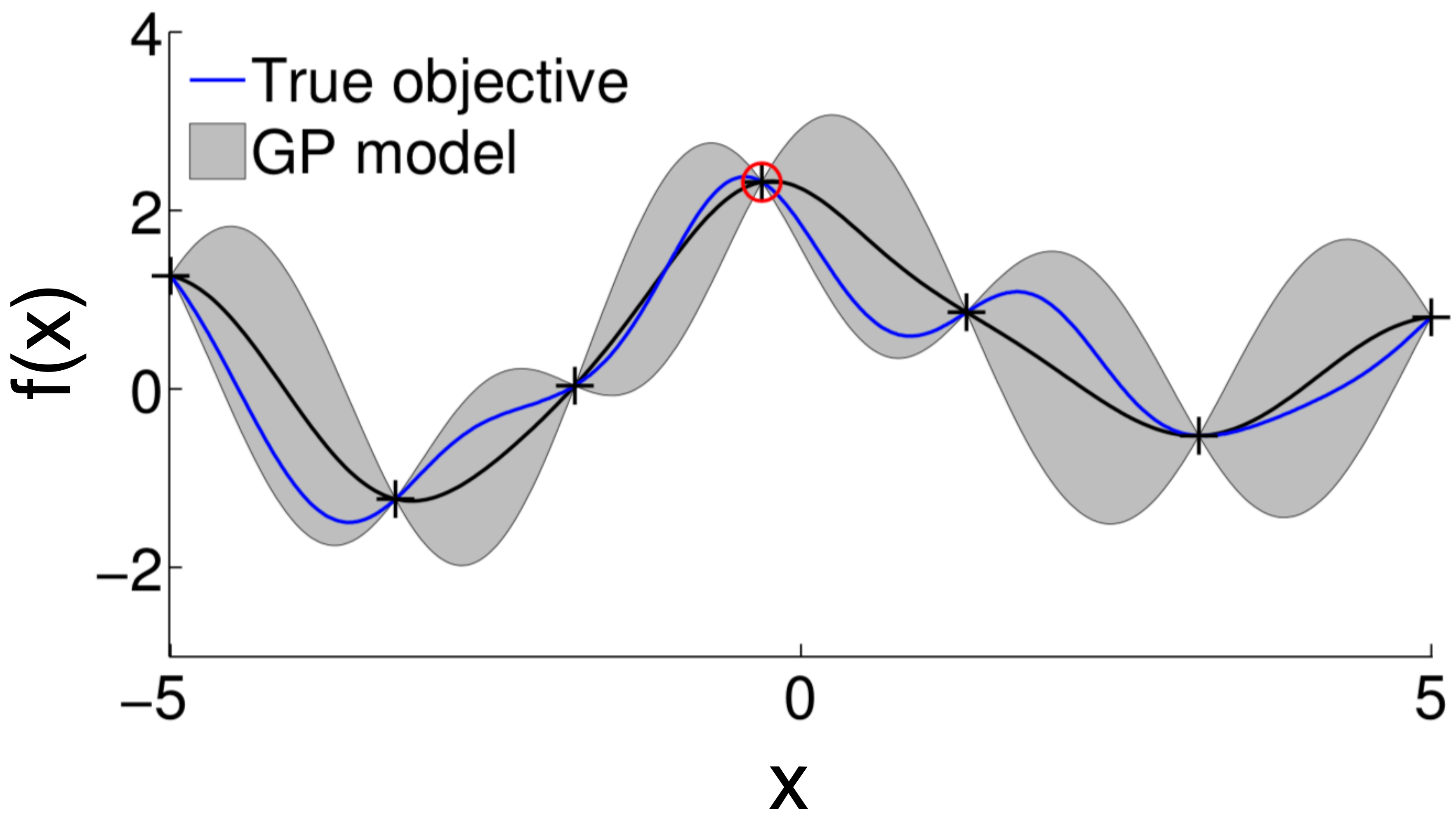
Optimization: Illustration

The model in this example is a Gaussian Process (GP)



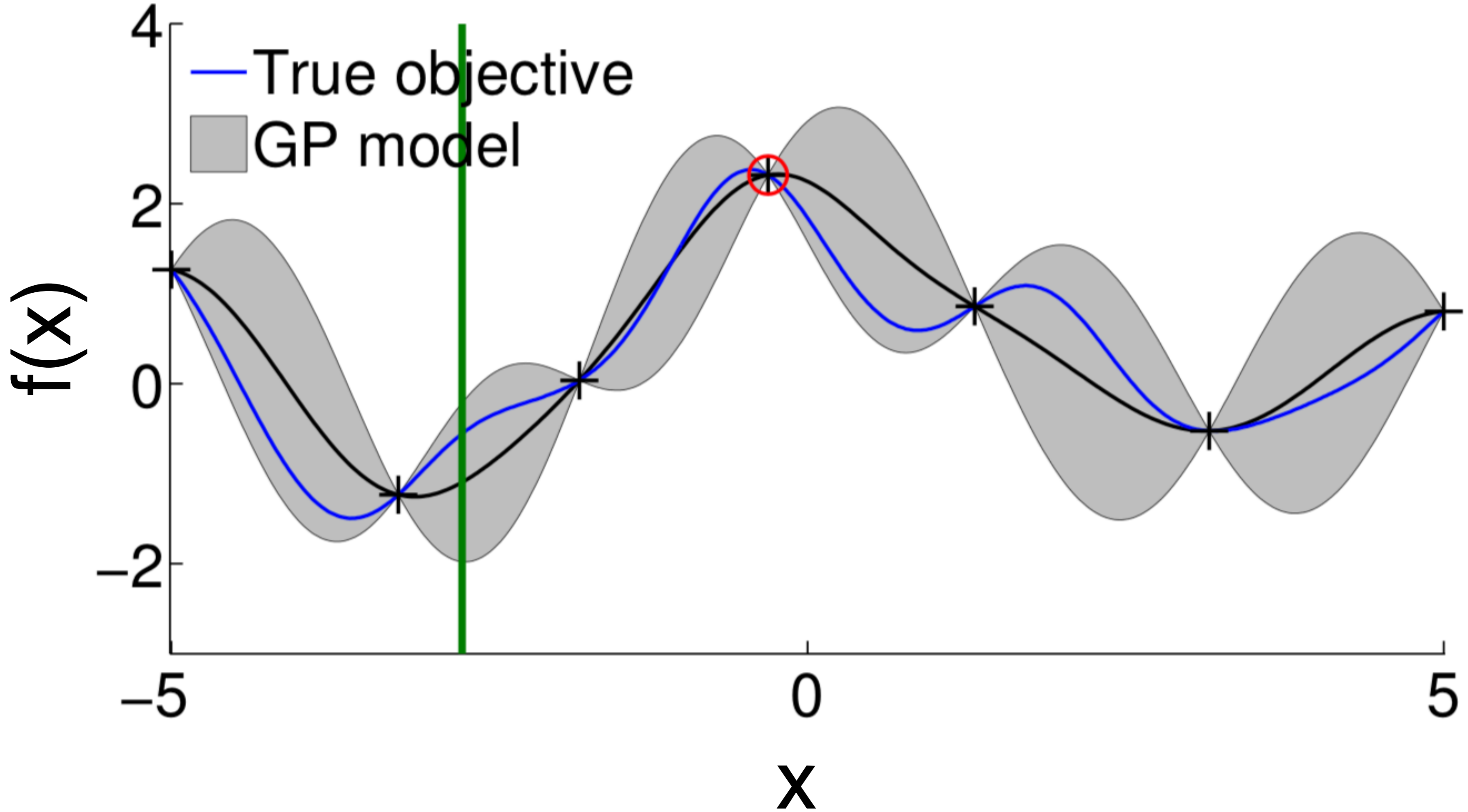
Optimization: Illustration

The model in this example is a Gaussian Process (GP)



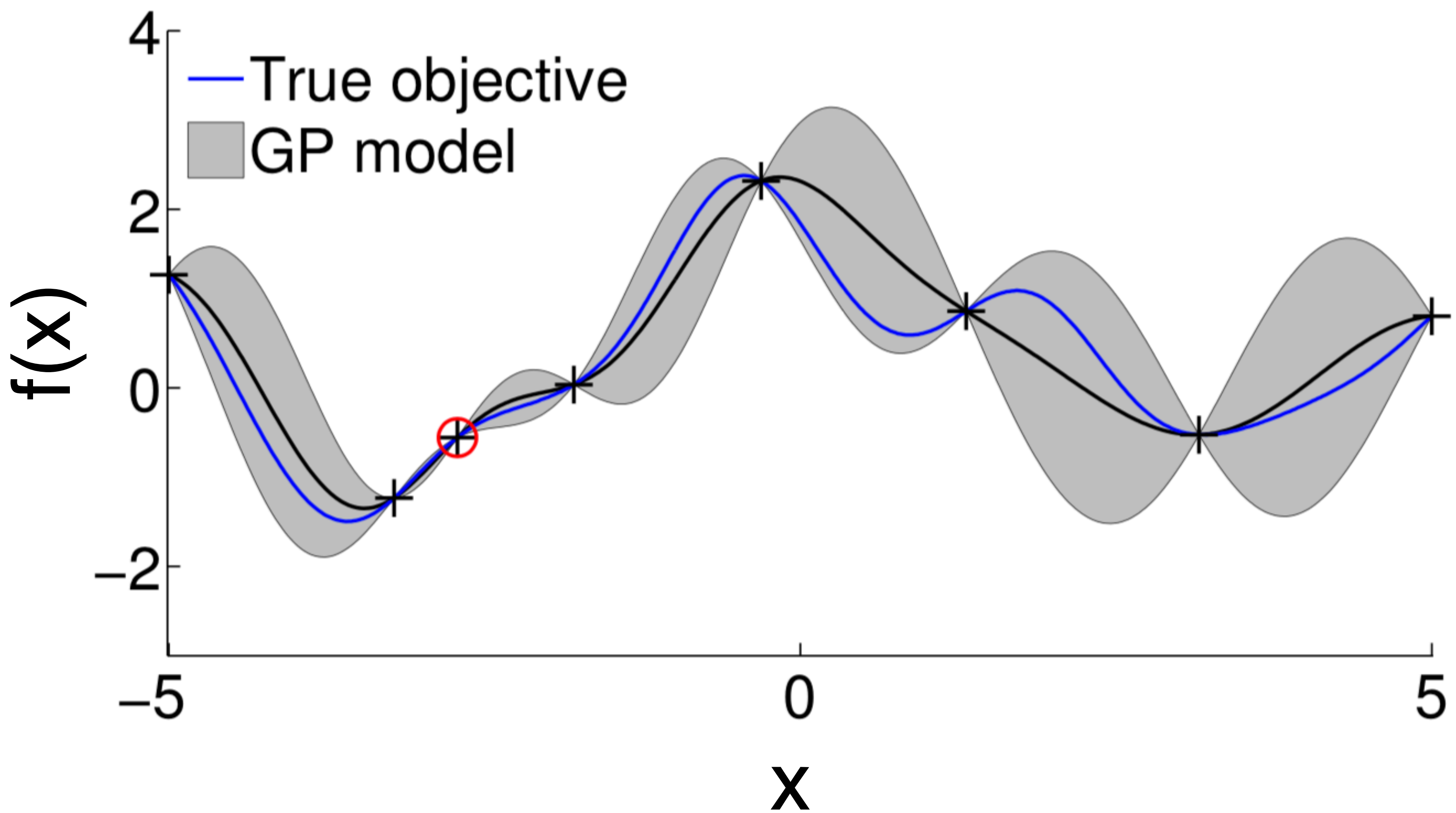
Optimization: Illustration

The model in this example is a Gaussian Process (GP)



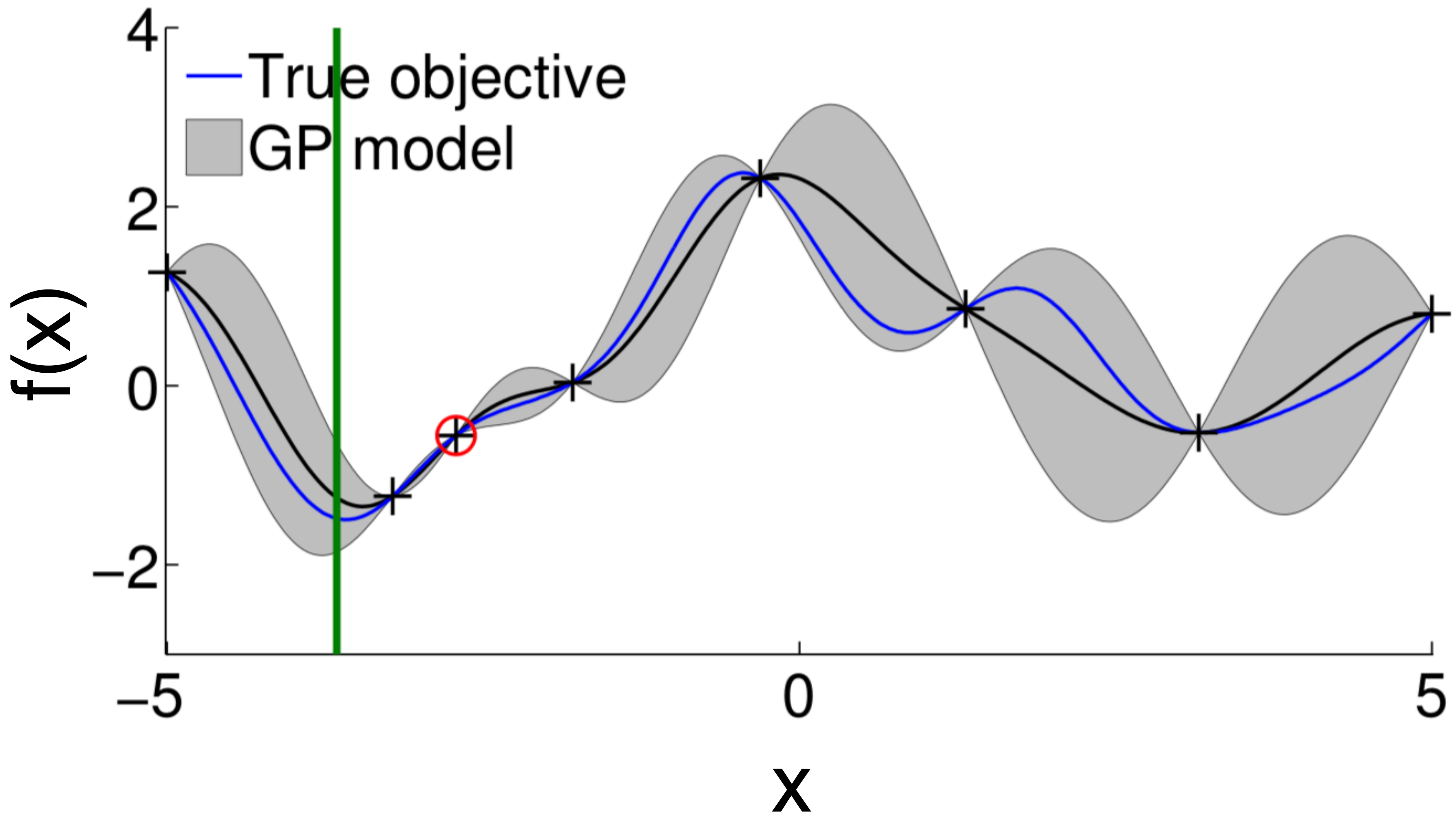
Optimization: Illustration

The model in this example is a Gaussian Process (GP)



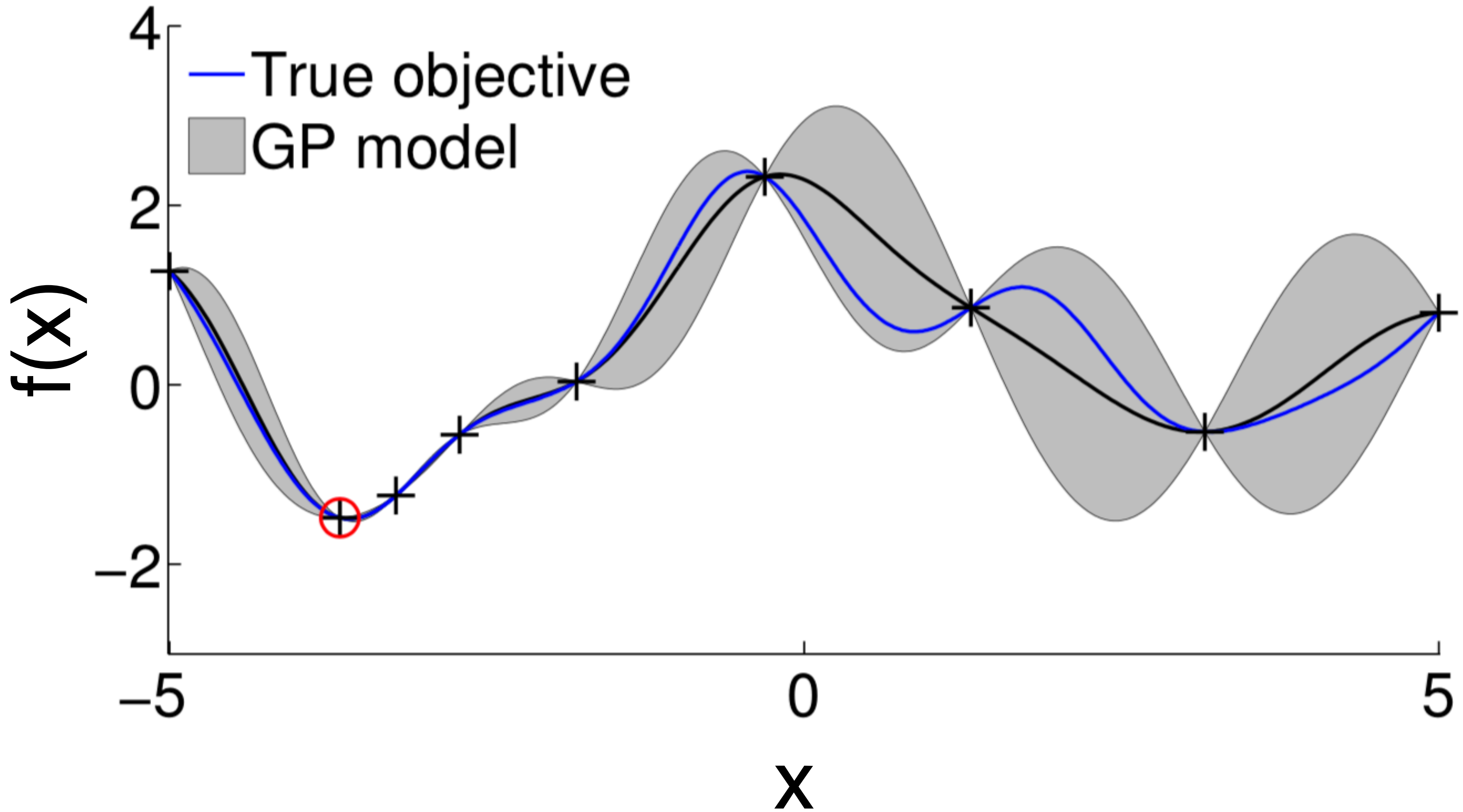
Optimization: Illustration

The model in this example is a Gaussian Process (GP)



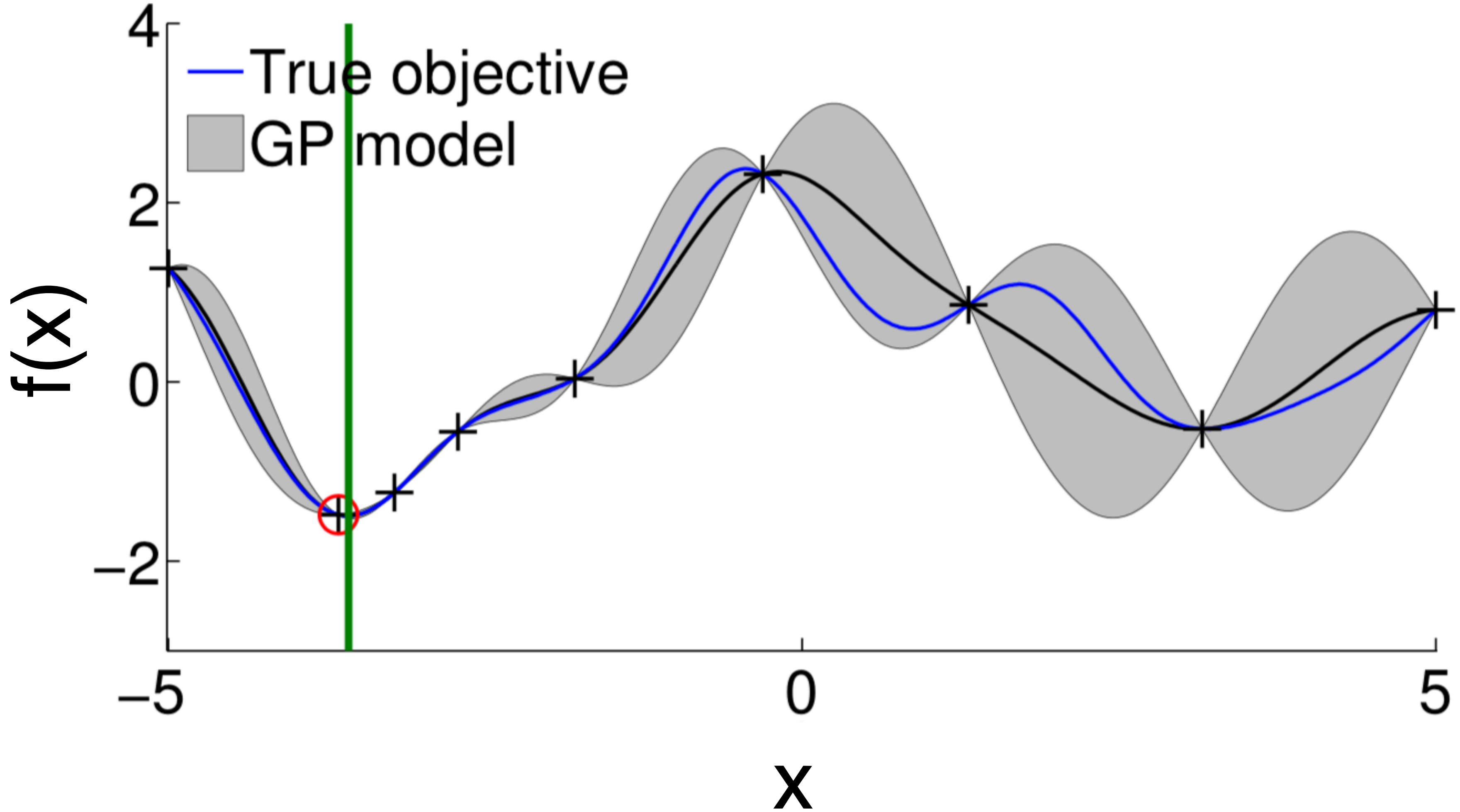
Optimization: Illustration

The model in this example is a Gaussian Process (GP)



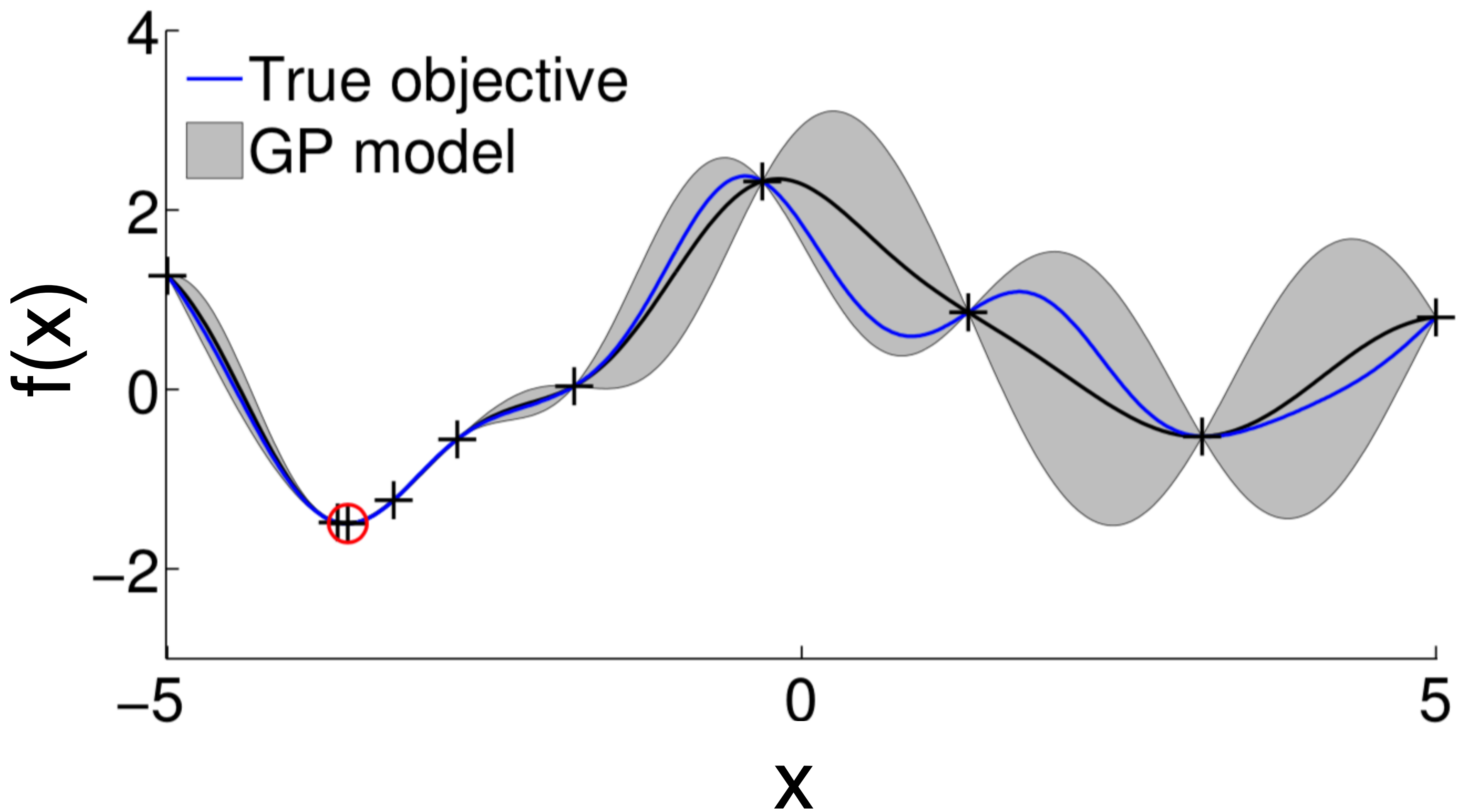
Optimization: Illustration

The model in this example is a Gaussian Process (GP)



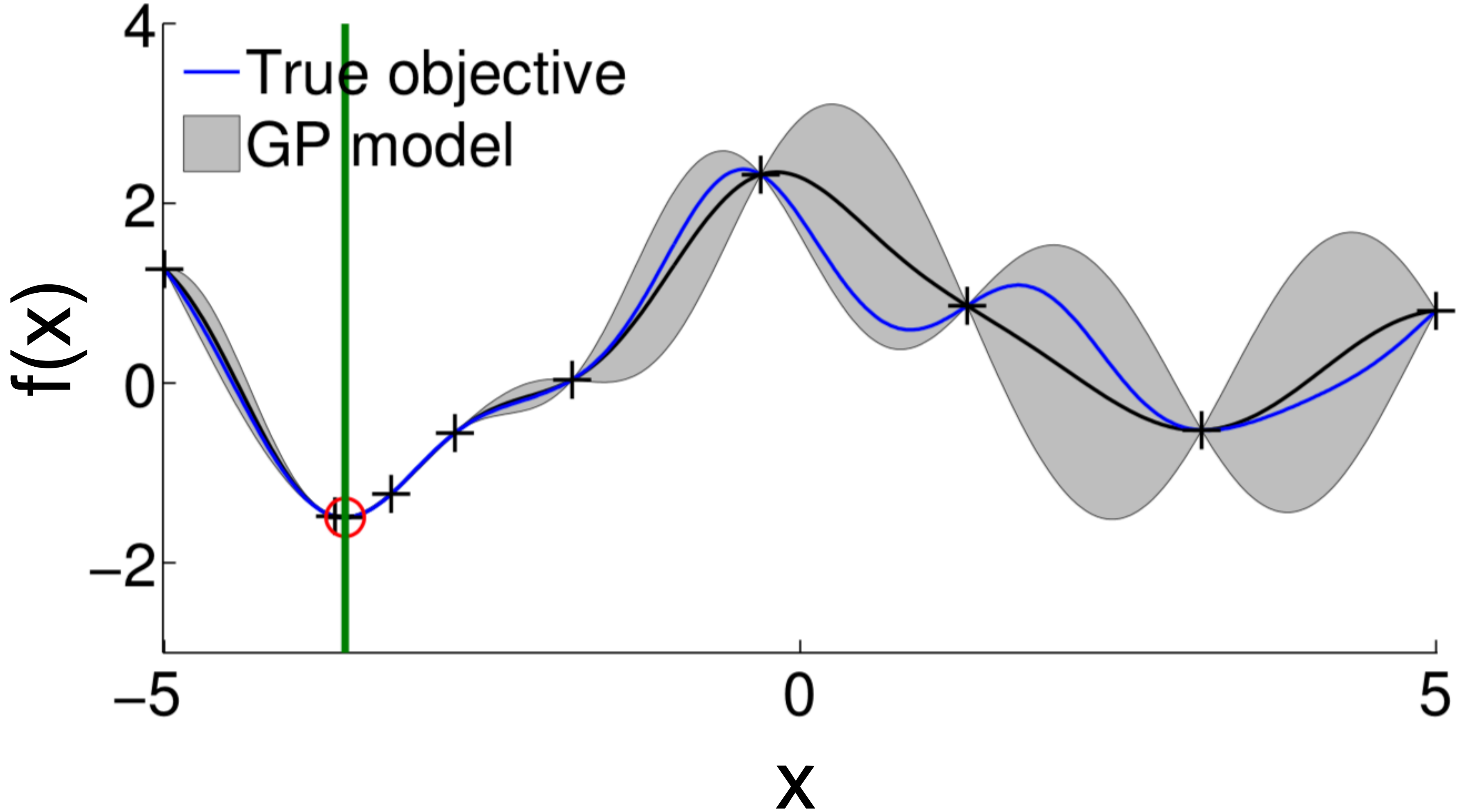
Optimization: Illustration

The model in this example is a Gaussian Process (GP)



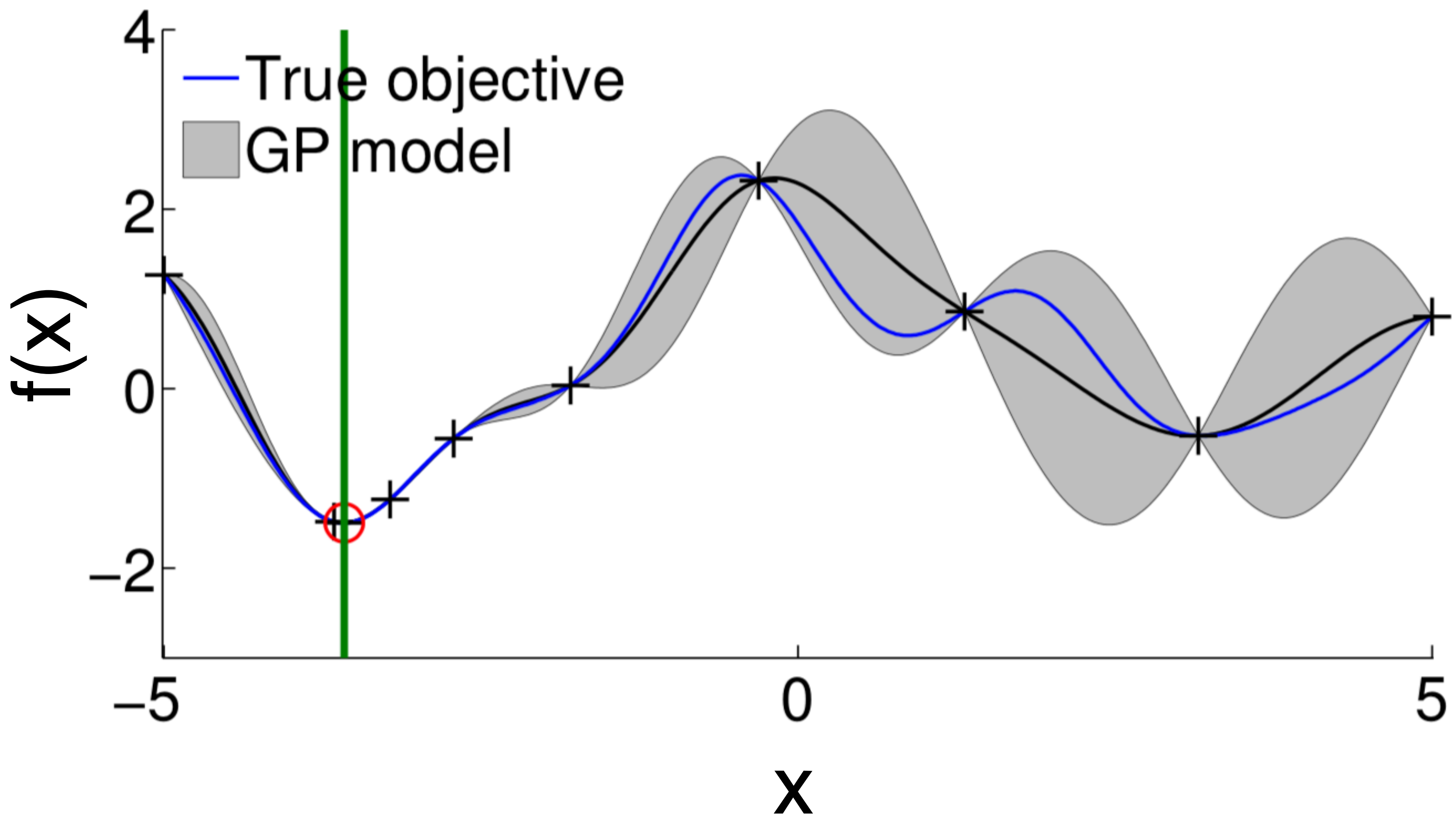
Optimization: Illustration

The model in this example is a Gaussian Process (GP)



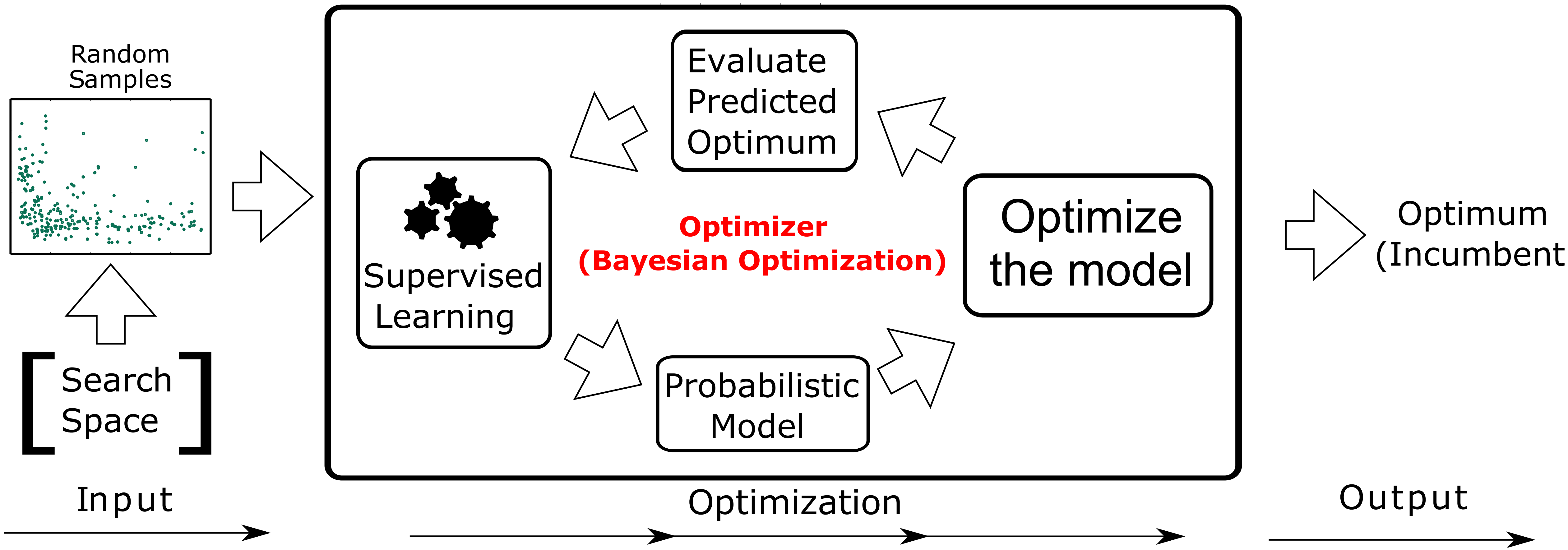
Optimization: Illustration

The model in this example is a Gaussian Process (GP)



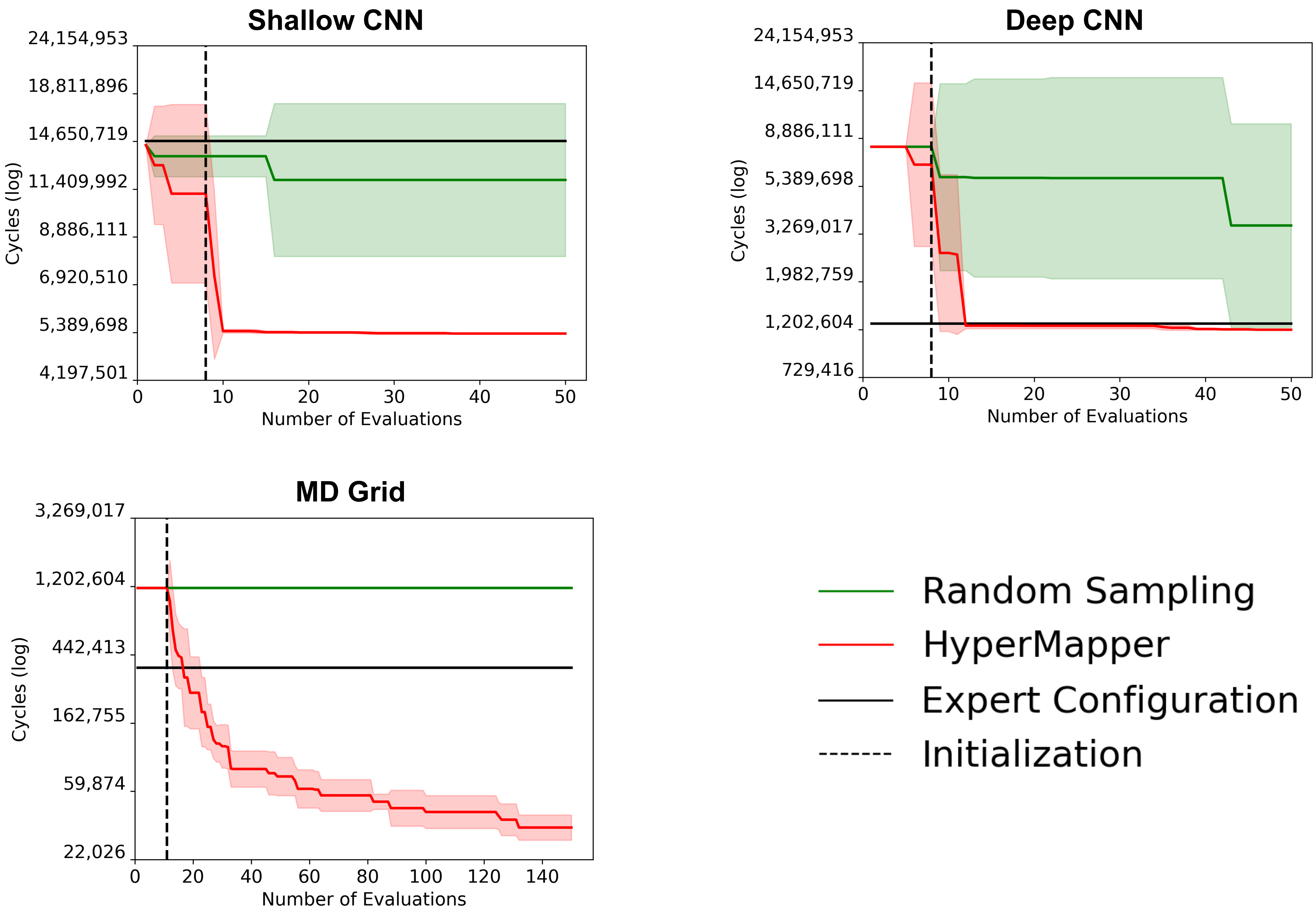
This optimization process is known as Bayesian Optimization

The HyperMapper Framework



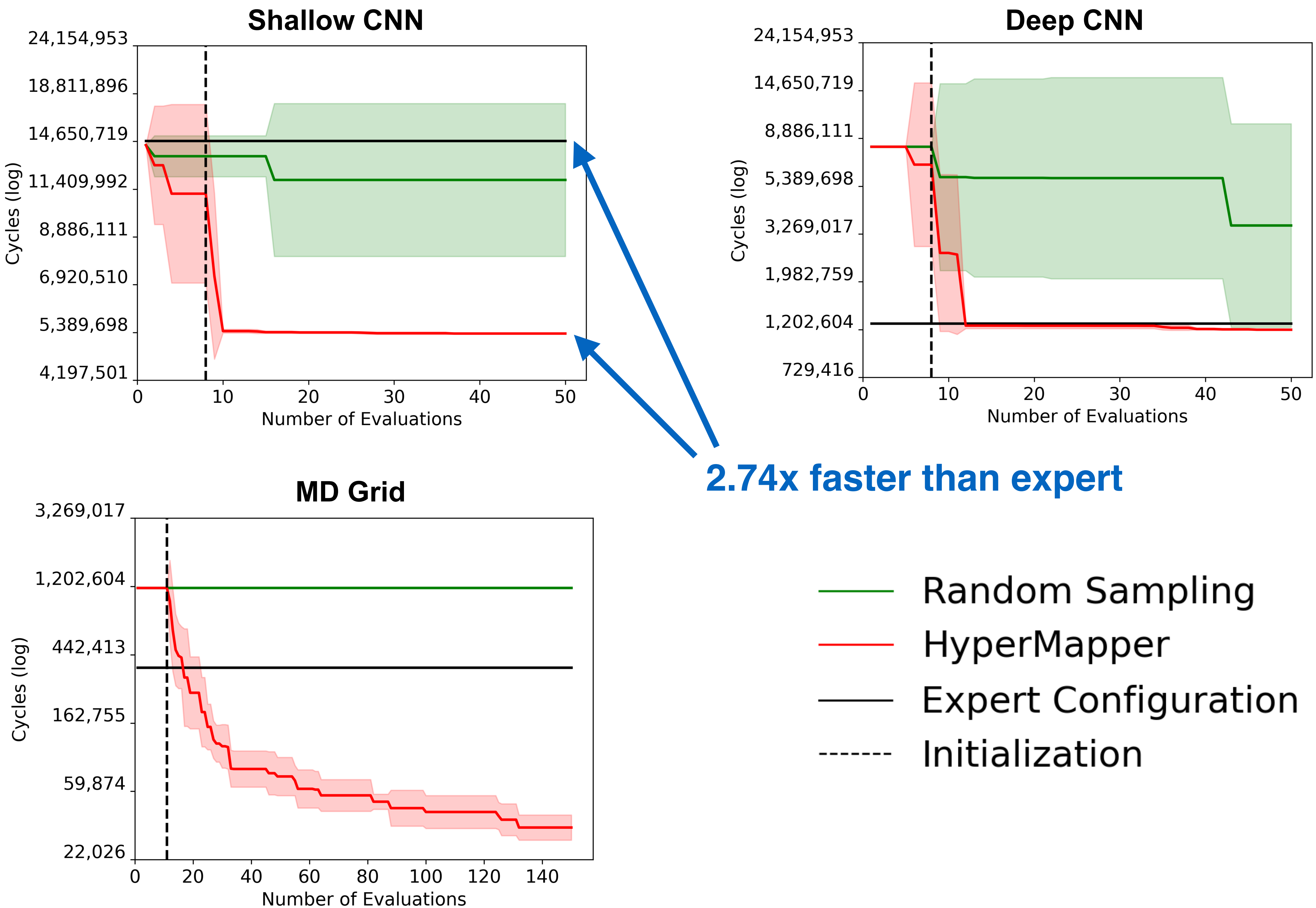
Spatial Results

- Real-world Applications -



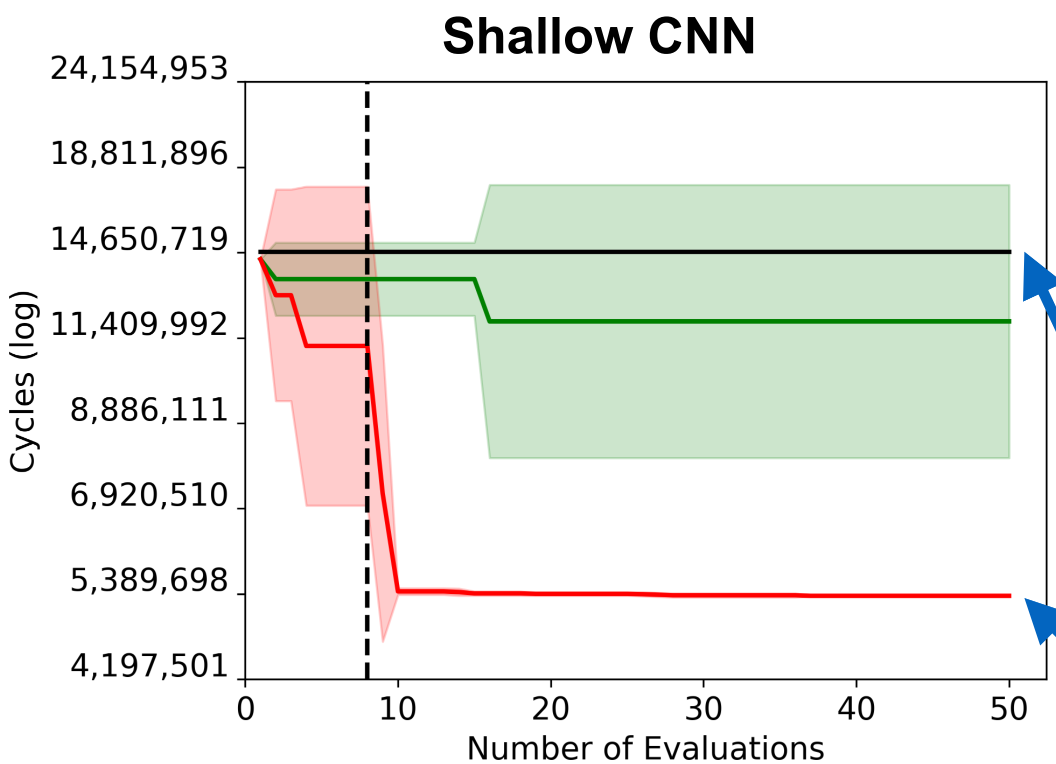
Spatial Results

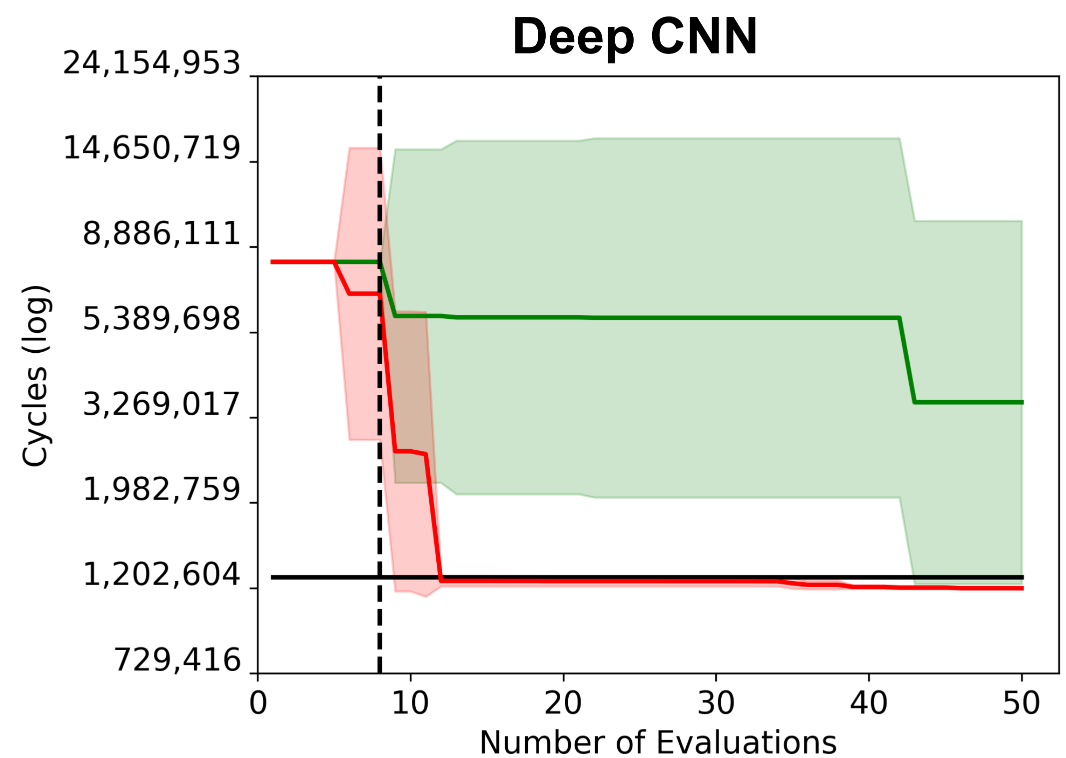
- Real-world Applications -



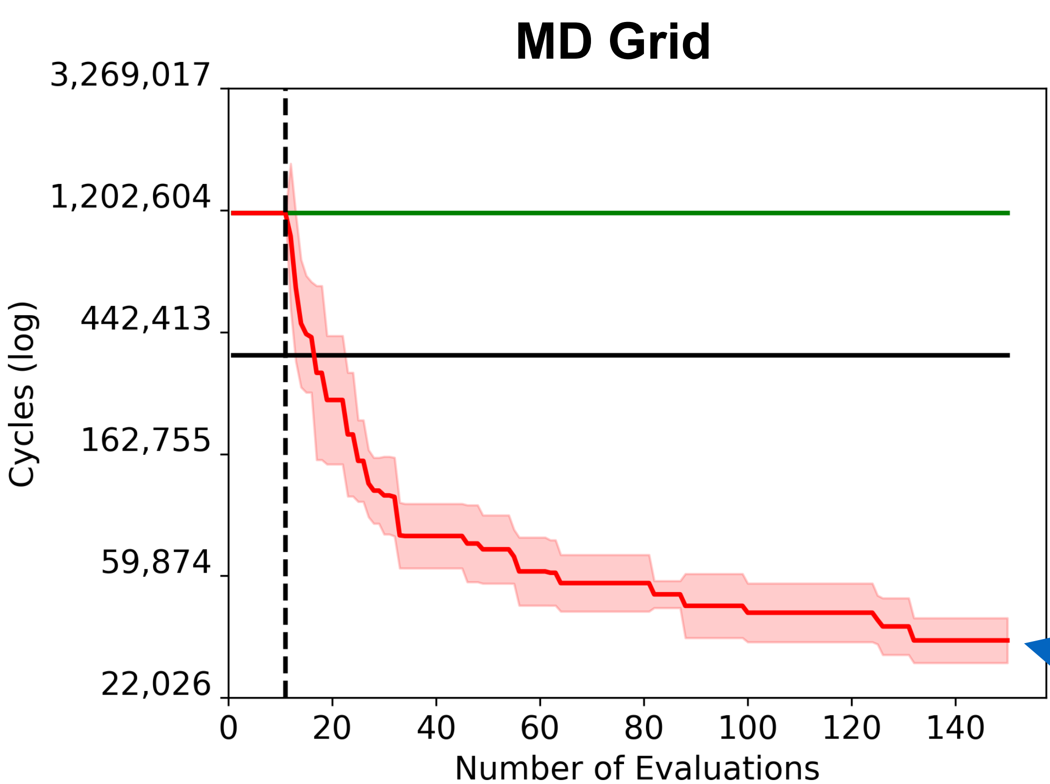
Spatial Results

- Real-world Applications -





2.74x faster than expert

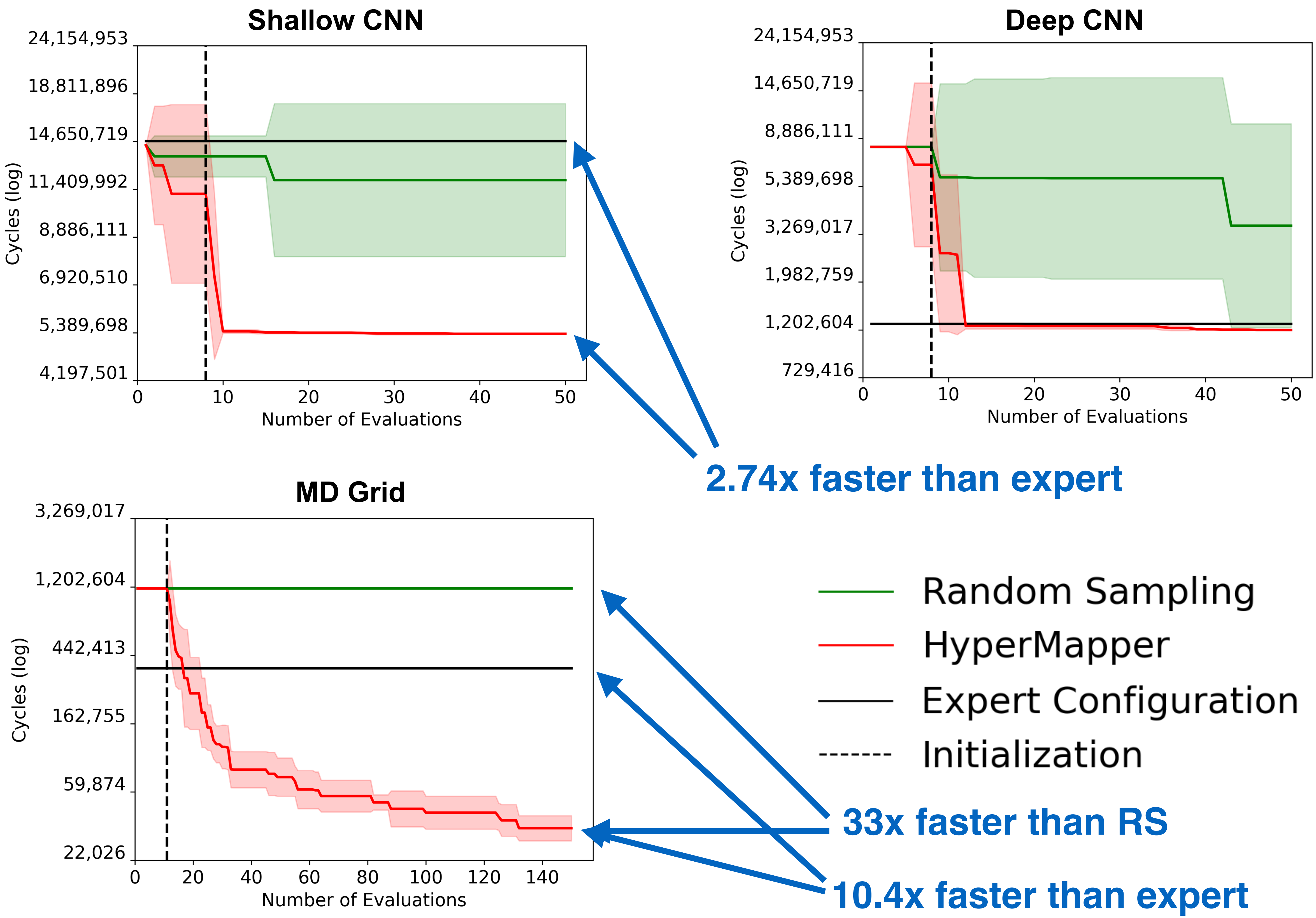


Random Sampling
HyperMapper
Expert Configuration
Initialization

10.4x faster than expert

Spatial Results

- Real-world Applications -



Summary

- Introduced the problem of Design Space Exploration (DSE)
  - Mono-objective
  - Will follow:
    - Multi-objective, priors, constrained optimization
    - Details of Spatial/DSE

Summary

- Introduced the problem of Design Space Exploration (DSE)
    - Mono-objective
    - Will follow:
        - Multi-objective, priors, constrained optimization
        - Details of Spatial/DSE
- A general solution to DSE
    - Widely applicable @PLDI
    - Interdisciplinary research work (ML/Math/PL/Systems)
    - Introduced HyperMapper, an umbrella framework for several optimizers:
        - Bayesian optimization (efficient), RS and heuristics

Summary

- Introduced the problem of Design Space Exploration (DSE)
  - Mono-objective
  - Will follow:
    - Multi-objective, priors, constrained optimization
    - Details of Spatial/DSE
- A general solution to DSE
  - Widely applicable @PLDI
  - Interdisciplinary research work (ML/Math/PL/Systems)
  - Introduced HyperMapper, an umbrella framework for several optimizers:
    - Bayesian optimization (efficient), RS and heuristics
- Use-case: real-world application Spatial

Info on HyperMapper

- Join HyperMapper on **Slack**: hypermapper.slack.com
- **Repo**: https://github.com/luinardi/hypermapper
- **Wiki**: https://github.com/luinardi/hypermapper/wiki

	Adopters
Microsoft	Database Management Systems
SENTIAN.AI	Automated Machine Learning
Stanford University	Hardware and Network Design
UC San Diego	FPGAs
TEXAS The University of Texas at Austin	Approximate Computing
Imperial College London	Computer Vision and Robotics
	Etc.