2022 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

Test Parameter Tuning with Blackbox Optimization: A Simple Yet Effective Way to Improve Coverage

Qijing Huang*, Hamid Shojaei, Fred Zyda, Azade Nazi,
Shobha Vasudevan, Sat Chatterjee, Richard Ho
{hamids, zyda, azade, shovasu, schatter, riho}@google.com
jennyhuang@nvidia.com

Google

* work done while at Google

accellera SYSTEMS INITIATIVE

Outline

01

Cloud TPU

Empowering EDA with Google Cloud AI

02

Google EDA Cloud

Fast Design of Large, Complex Chips

03

ML-based EDA

Using ML to Accelerate Design Flows

04

SRP

Smart Regression approach

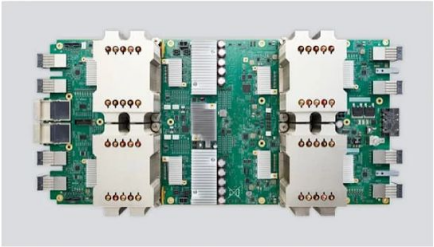
05

Experiments

Different setups and results

Cloud TPU

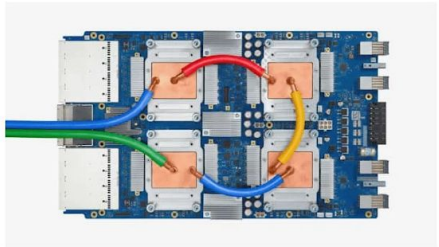
Empowering EDA with Google Cloud AI



Cloud TPU v2

180 teraflops

64 GB High Bandwidth Memory (HBM)



Cloud TPU v3

420 teraflops

128 GB HBM

Cloud TPU v2 Pod

11.5 petaflops

4 TB HBM

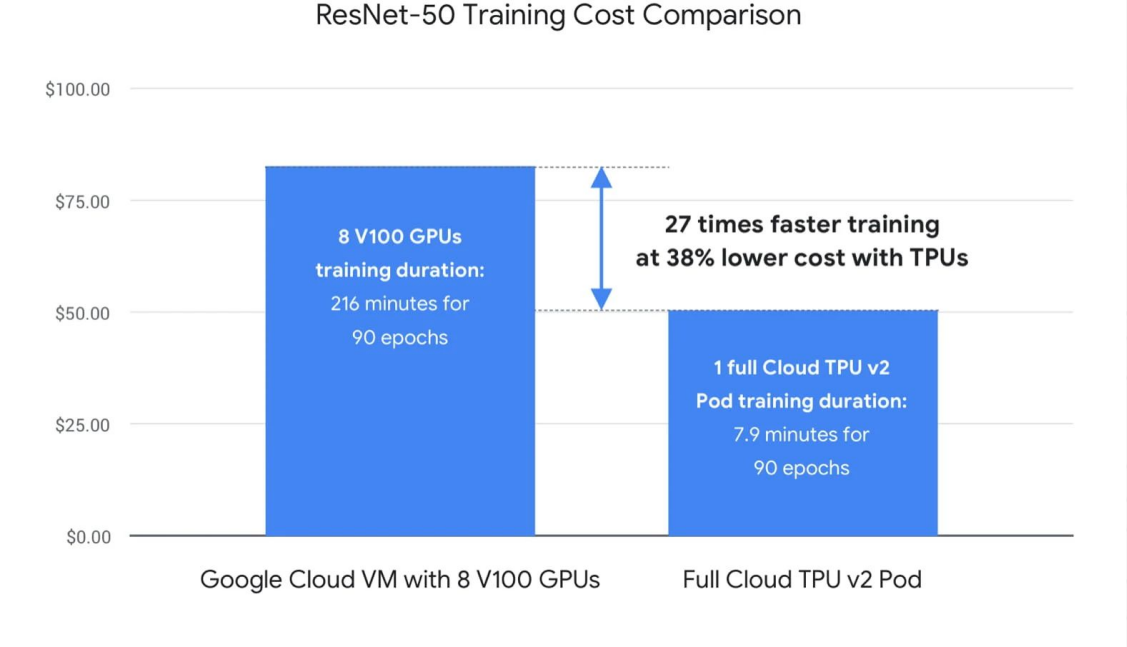
2-D toroidal mesh network

Cloud TPU v3 Pod

100+ petaflops

32 TB HBM

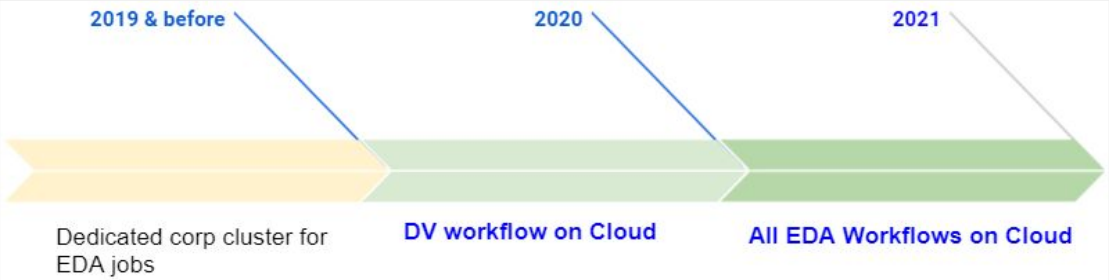
2-D toroidal mesh network



Google Cloud: Fast Design of Large, Complex Chips

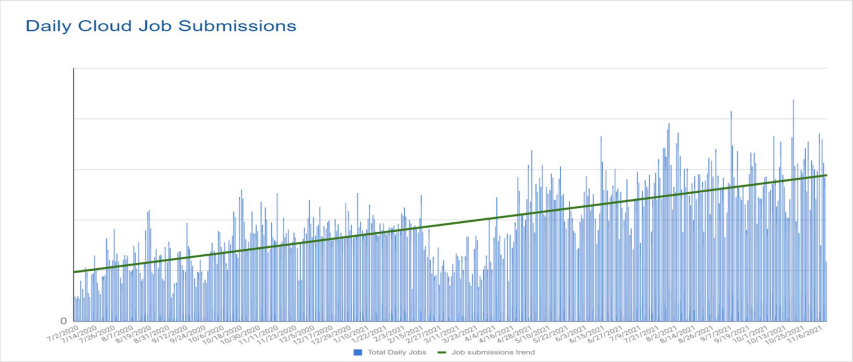
Cloud capacity across GCP regions

Moving EDA to Cloud



EDA workflows on Cloud with GCP

Job Management	Custom Scheduler	Google Kubernetes Engine			
Storage & DB	Cloud Storage	Filestore	Persistent Disk	Cloud Spanner	
Compute	Compute Engine				



Elasticity

Quick expansion of compute capacity across geos as roadmap changes. Onboard remote team in < 2 day.

Scalability

Solution designed with horizontal scaling, 170% total daily jobs growth from YoY.

Reliability

Eliminated single failure zone with cloud capacity allocated across the globe.

Using AI/ML to Accelerate Design Flows

Goal

Develop scalable, and generalizable machine learning driven framework with rapid evaluation and turn-around time to shorten the chip design process.

Infrastructure on GCP

Google Cloud: scalable infrastructure with large compute power that works in domains with large number of design parameters.
Machine Learning: leveraging various ML algorithms, readily available in Google Cloud to efficiently navigate large search space and apply unique optimizations at various stages of chip design.

Results

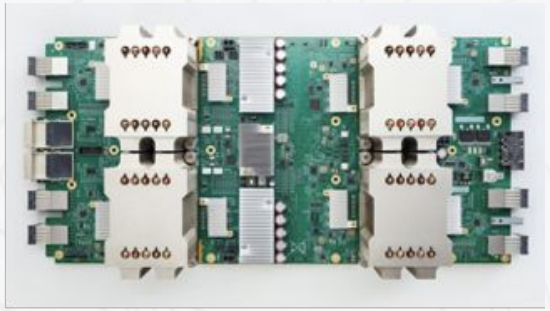
Shorten the chip design process and reduce time-to-market, expand product areas for ML accelerators, and improve the efficiency.

Two pronged strategy

ML by available tools in Google Cloud
- Immediate impact and savings

ML solutions based on new research
-- Longer timeline and upfront investment

Cloud ML Engines



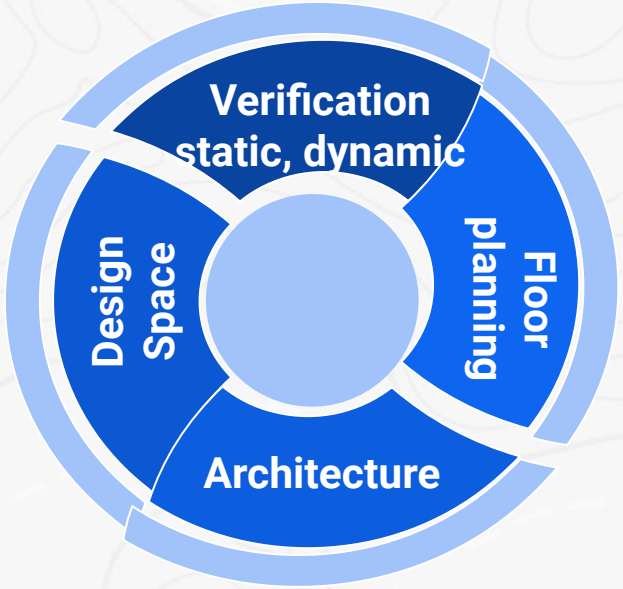
Design More Efficient Accelerator

Existing solutions on gcloud

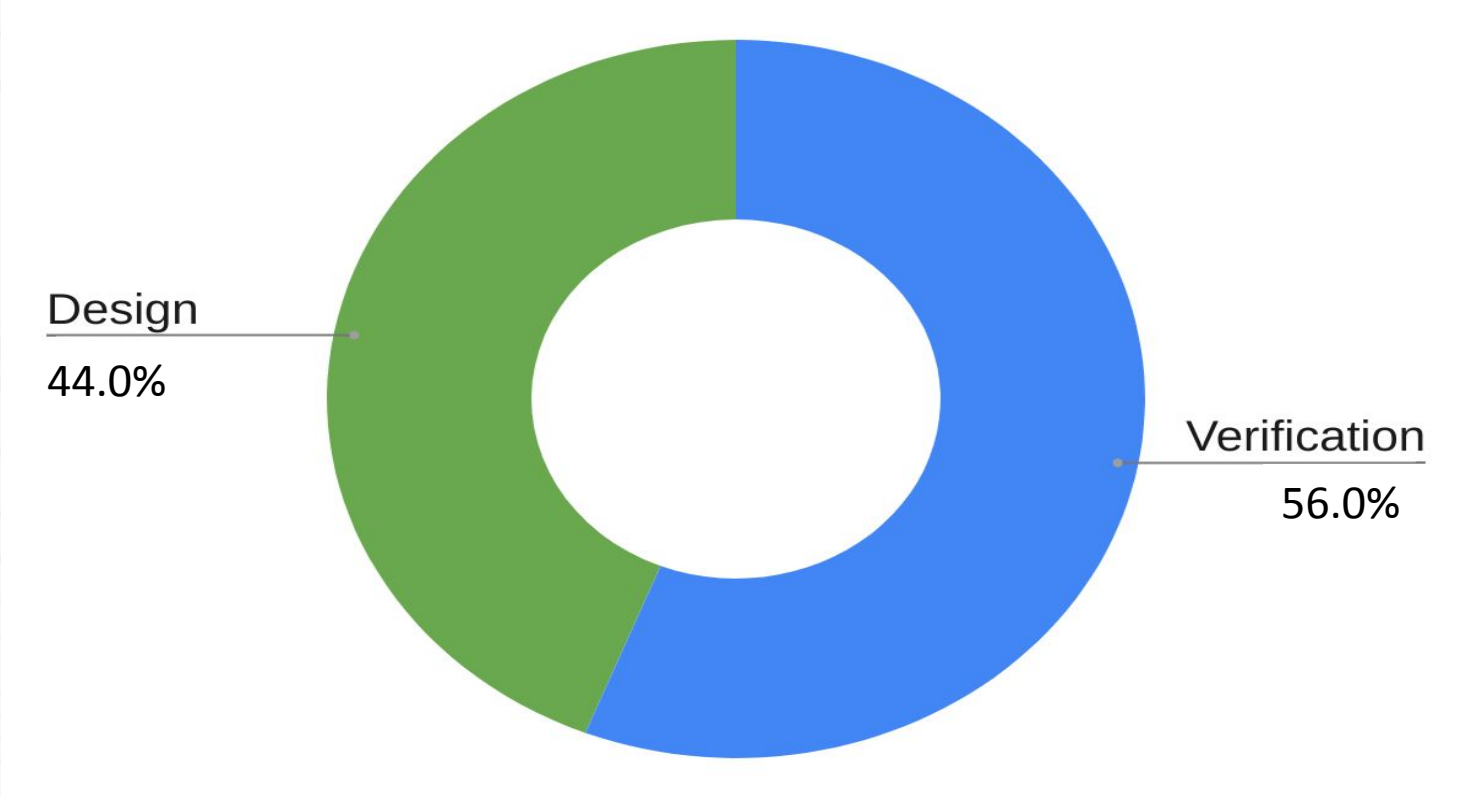
- Bayesian networks
- Vizier
- Reinforcement Learning

Dedicated research solutions

- Distribution optimization
- Graph neural networks
- Language models

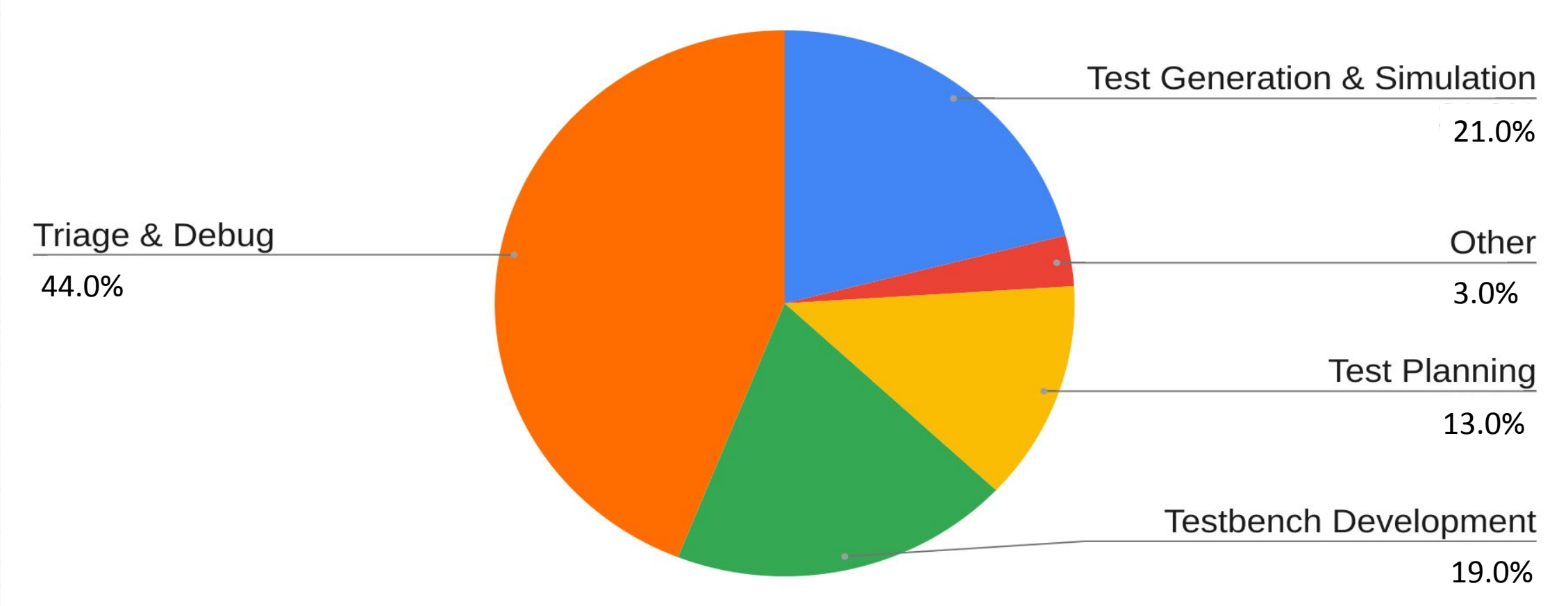


ASIC Project Time Spent in Verification



Source: Wilson Research Group/Mentor

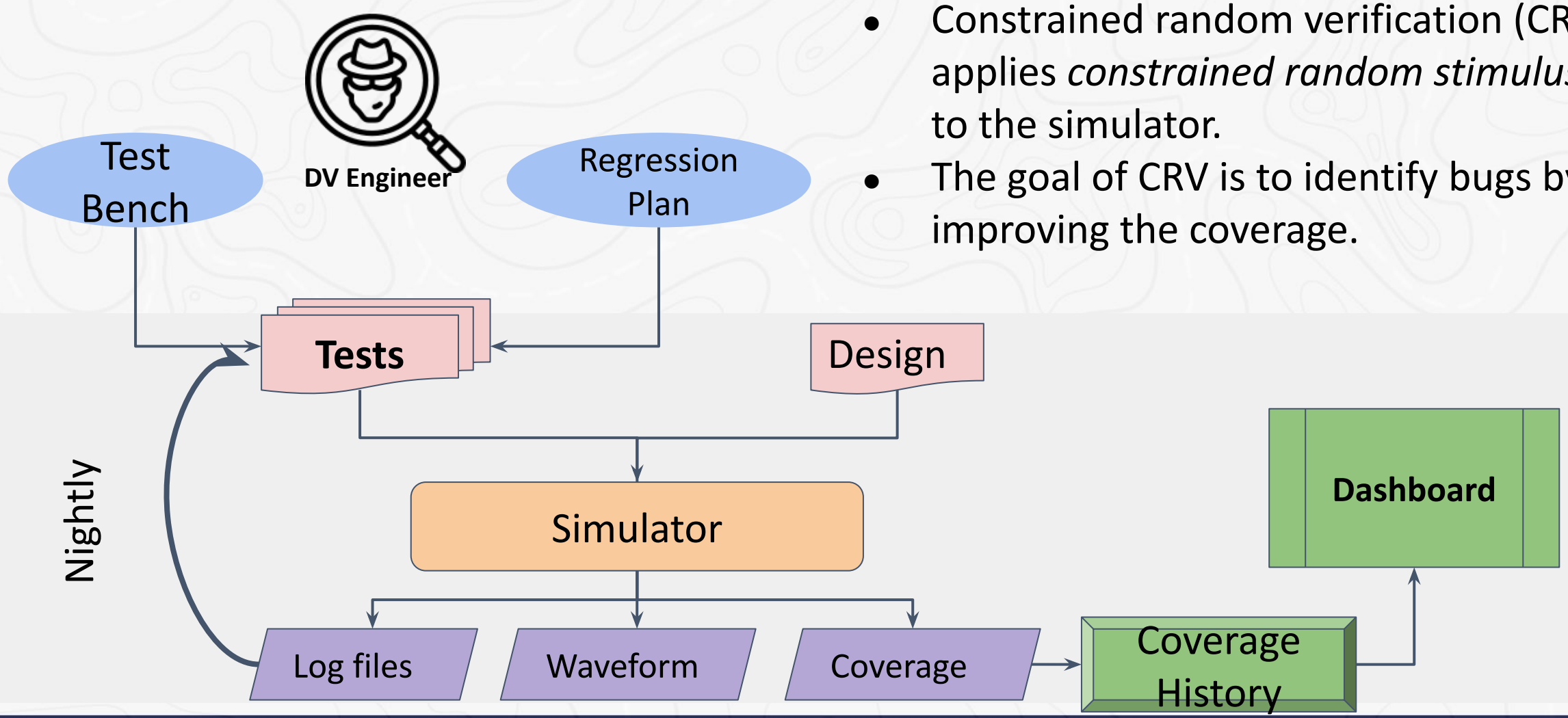
Where ASIC verification engineers spend their time?



Source: Wilson Research Group/Mentor

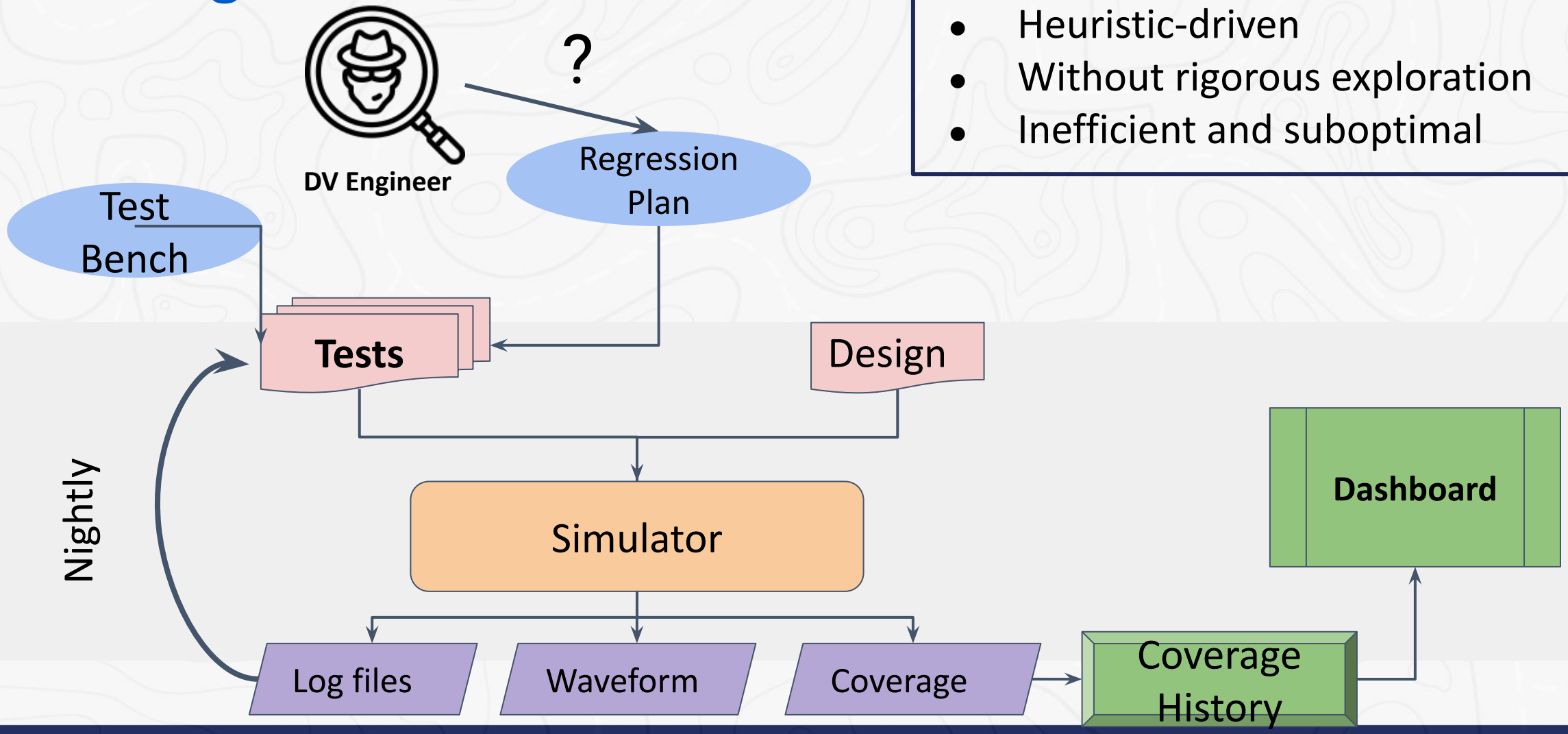
Industrial Verification Flow - CRV

- Constrained random verification (CRV) applies *constrained random stimulus* to the simulator.
- The goal of CRV is to identify bugs by improving the coverage.



Challenges

- Heuristic-driven
- Without rigorous exploration
- Inefficient and suboptimal



Example RISCV-DV Test Parameters

- Binary Parameters:

disable_compressed_instr=0	enable_illegal_csr_instruction=1	fix_sp=0
enable_access_invalid_csr_level=1	enable_misaligned_instr=0	no_branch_jump=1
enable_dummy_csr_write=0	enable_unaligned_load_store=0	no_csr_instr=0
no_data_page=0	set_mstatus_tw=1	no_directed_instr=0
no_dret=0	no_ebreak=0	randomize_csr=0

- Integer Parameters:

instr_cnt=7000	num_of_sub_program=8	hint_instr_ratio=15
illegal_instr_ratio=50	stream_freq_0=5	stream_freq_1=40
stream_freq_10=35	stream_freq_2=0	stream_freq_3=20
stream_freq_4=10	stream_freq_5=50	stream_freq_6=0
stream_freq_7=10	stream_freq_8=20	stream_freq_9=25

* SV/UVM based instruction generator for RISC-V processor verification: https://github.com/google/riscv-dv

Example Parameterized RISCV-DV Tests

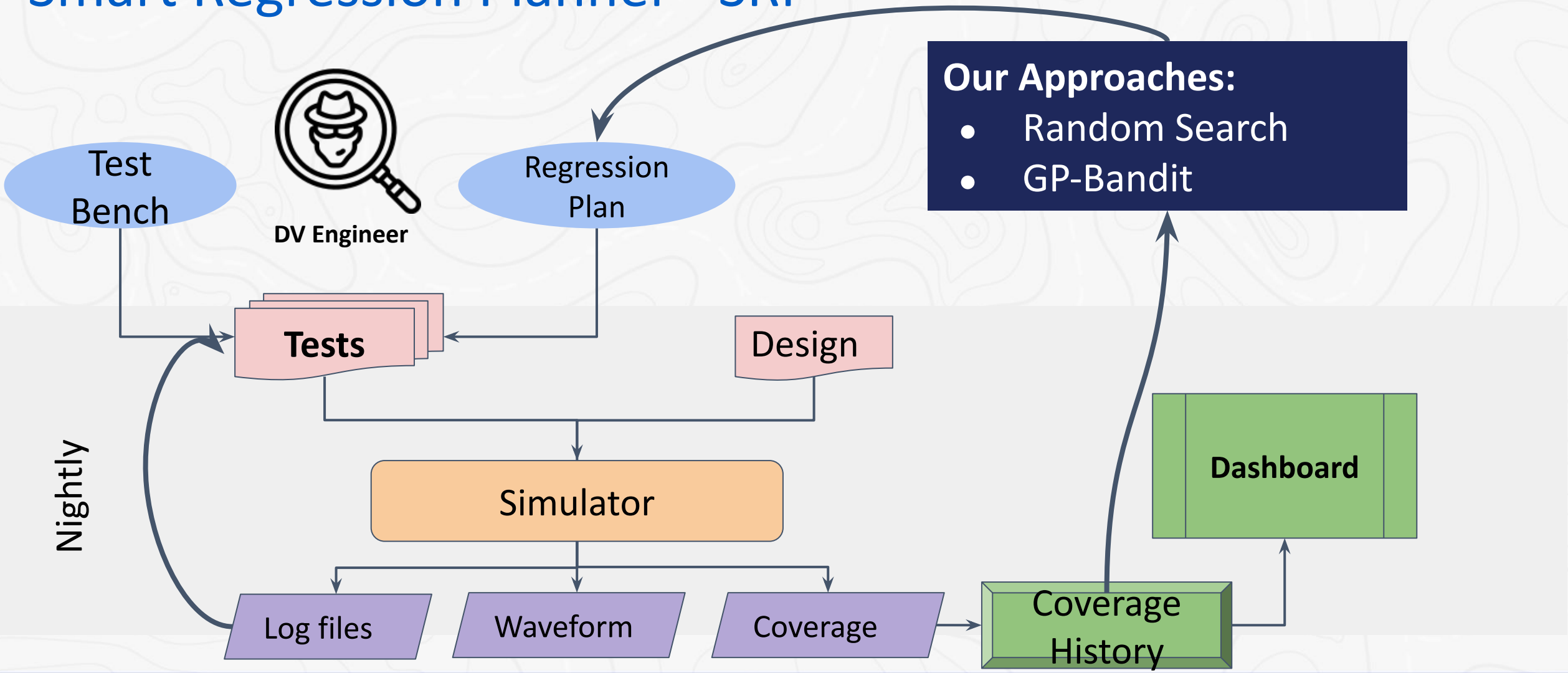
```
test: riscv_single_interrupt_test
iterations: 15
gen_test: riscv_rand_instr_test
gen_opts: >
  +instr_cnt=10000
  +require_signature_addr=1
  +enable_interrupt=1
  +randomize_csr=1
  +illegal_instr_ratio=20
rtl_test: core_ibex_debug_intr_basic_test
sim_opts: >
  +require_signature_addr=1
  +enable_irq_single_seq=1
compare_opts:
  compare_final_value_only: 1
```

Exception handling

```
test: riscv_mmu_stress_test
iterations: 10
gen_test: riscv_instr_base_test
gen_opts: >
  +instr_cnt=10000
  +num_of_sub_program=5
  +directed_instr_0=
   riscv_load_store_rand_instr_stream,40
  +directed_instr_1=
   riscv_load_store_hazard_instr_stream,40
  +directed_instr_2=
   riscv_multi_page_load_store_instr_stream,40
  +directed_instr_3=
   riscv_load_store_rand_addr_instr_stream,40
rtl_test: core_ibex_base_test
```

MMU stress tests

Smart Regression Planner - SRP



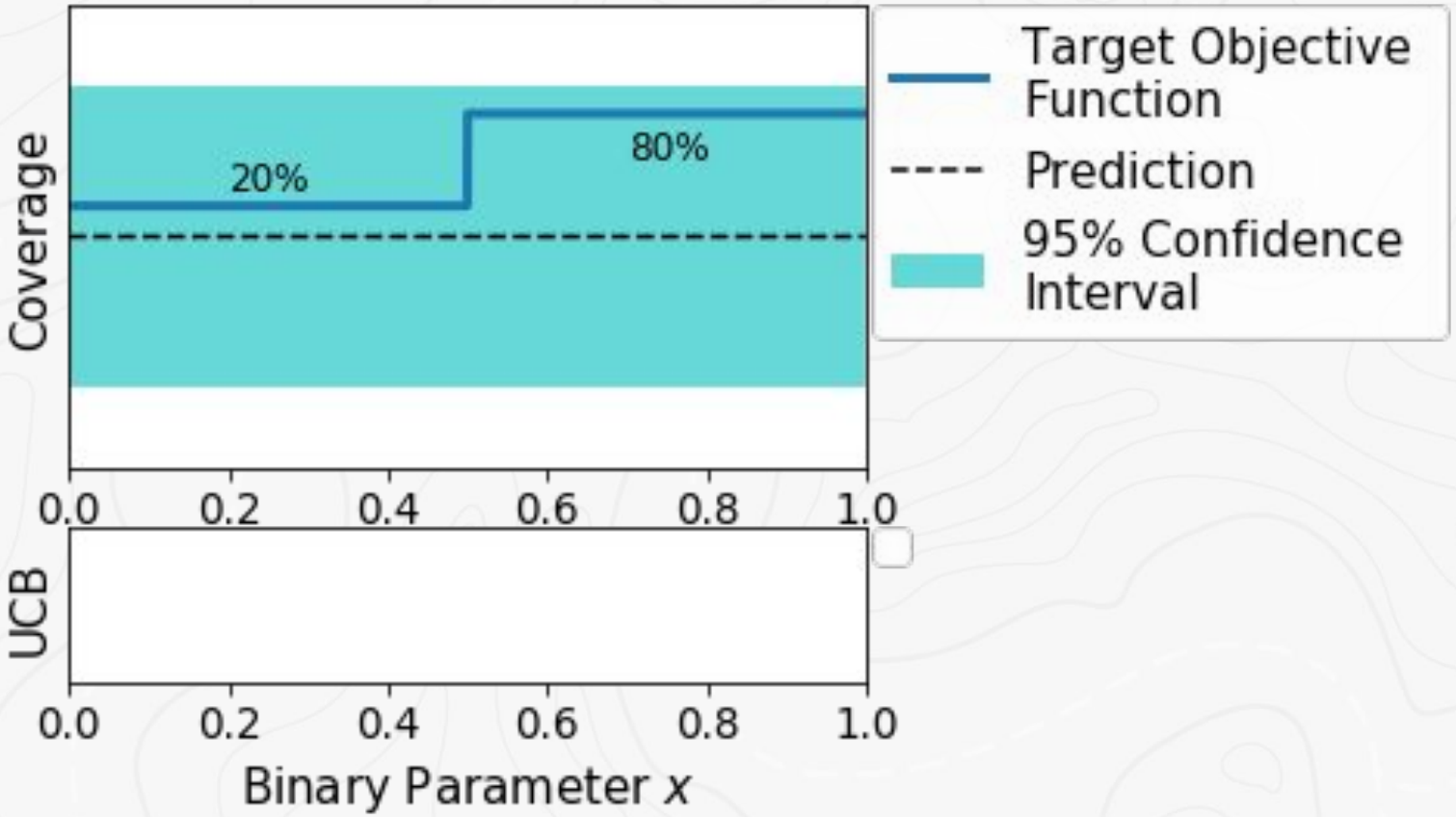
GP-Bandit In a Nutshell

- A black-box optimization algorithm
  - Treats the test-parameter to coverage function as a unknown black box
- A Bayesian optimization method
  - A statistical model to approximate the black-box function
  - An acquisition function to sample that next point
    - Upper confidence bound (UCB)

GP-Bandit Example for a Binary Test Parameter

- A binary parameter
  - E.g. no_break
- If enabled (1)
  - 80% coverage
- If disabled (0)
  - 20% coverage

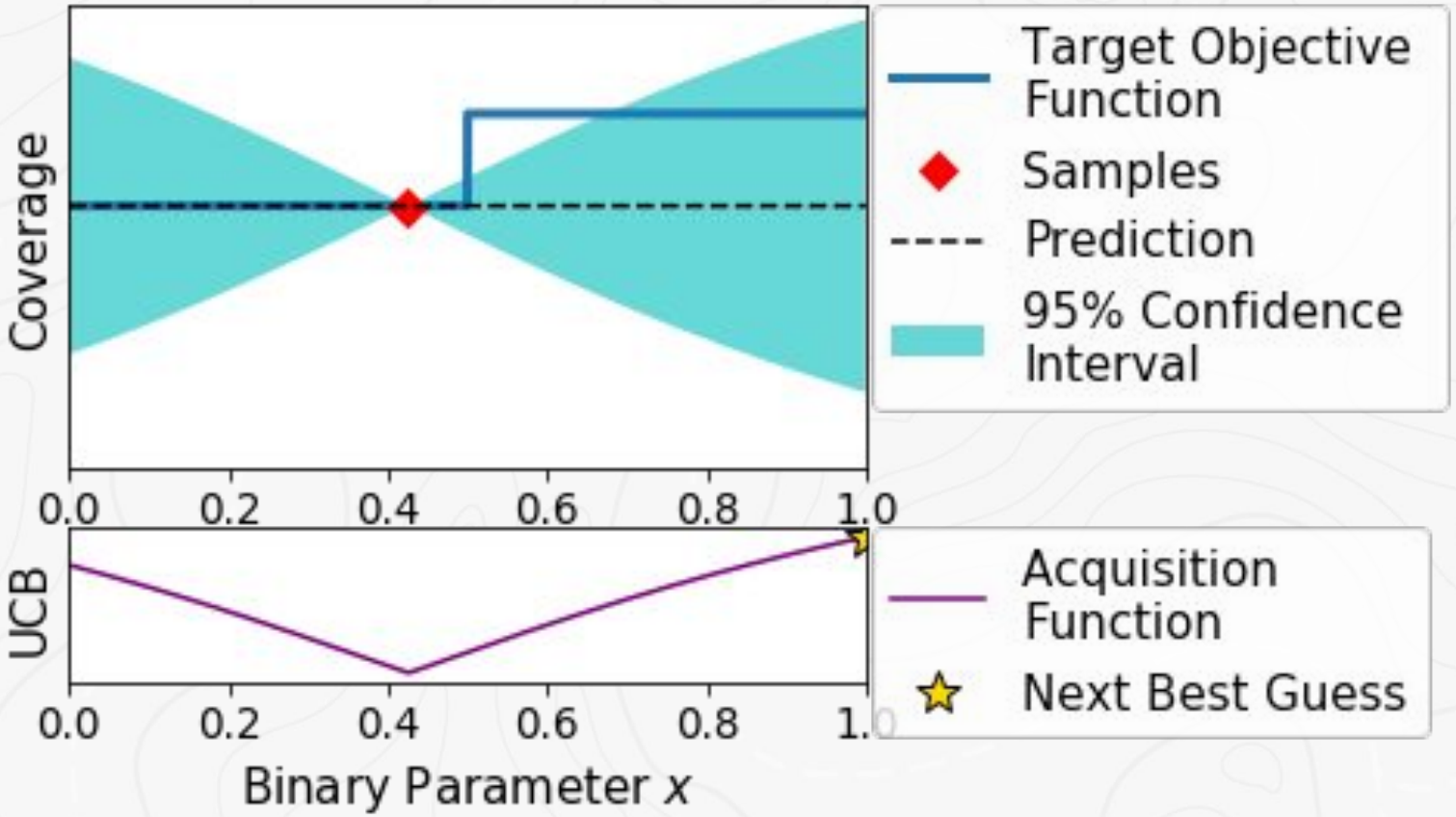
Iteration 0



GP-Bandit Example for a Binary Test Parameter

- A binary parameter
  - E.g. no_break
- If enabled (1)
  - 80% coverage
- If disabled (0)
  - 20% coverage

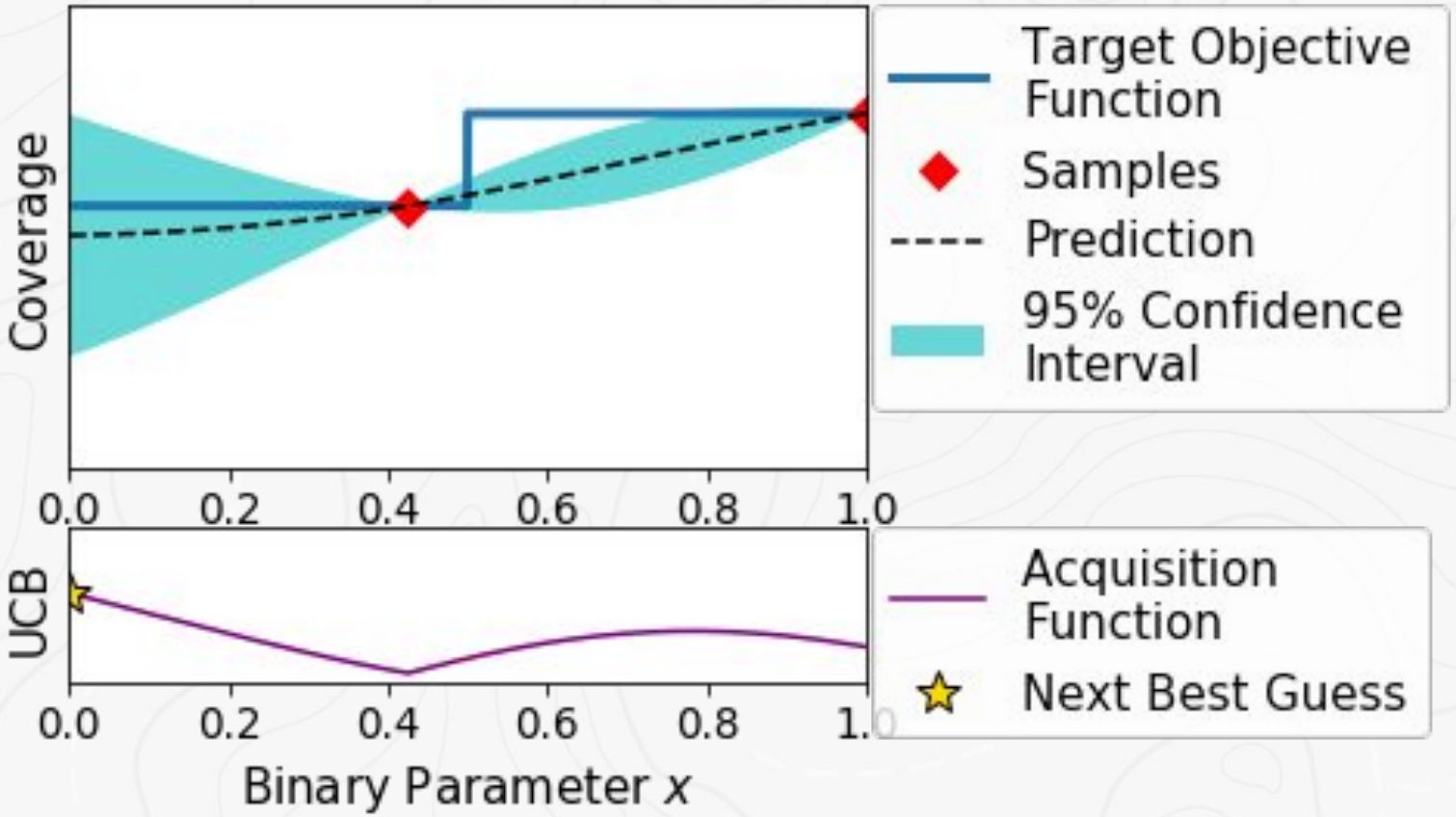
Iteration 1



GP-Bandit Example for a Binary Test Parameter

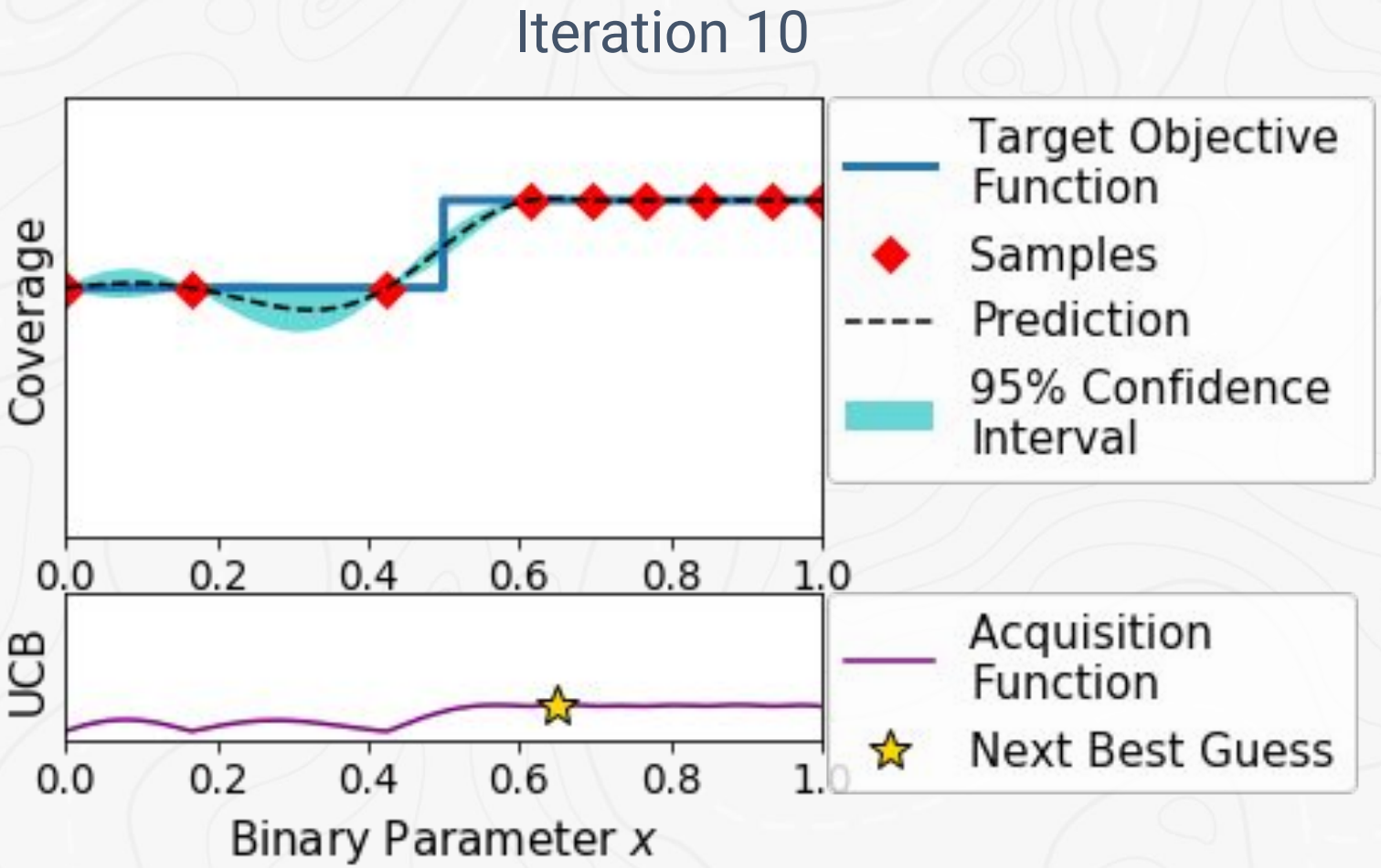
- A binary parameter
  - E.g. no_break
- If enabled (1)
  - 80% coverage
- If disabled (0)
  - 20% coverage

Iteration 2



GP-Bandit Example for a Binary Test Parameter

- A binary parameter
  - E.g. no_break
- If enabled (1)
  - 80% coverage
- If disabled (0)
  - 20% coverage



Evaluation

Approaches:

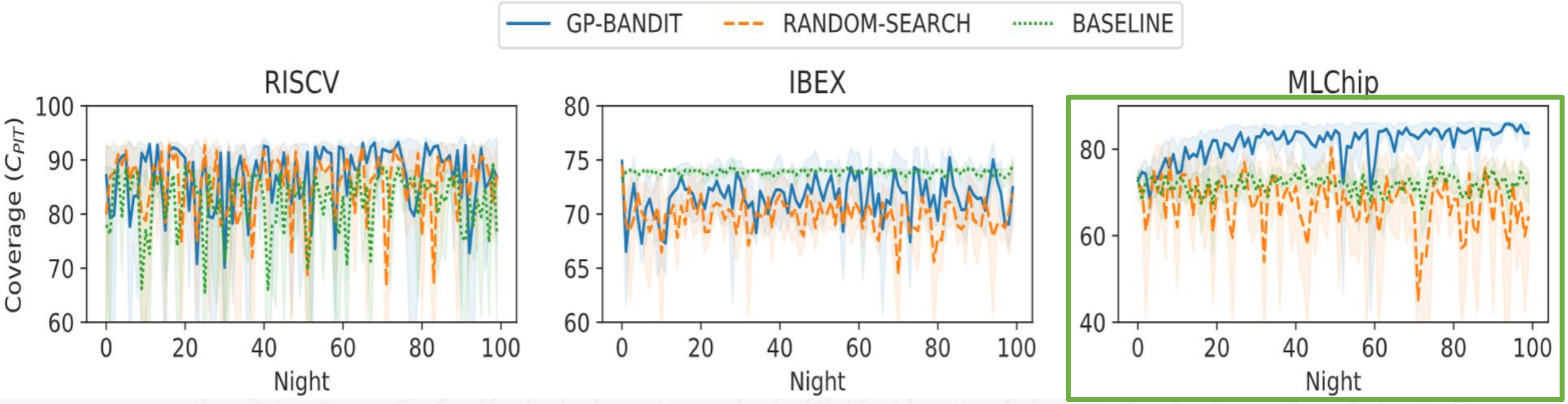
1. Baseline
2. Random
3. GP-Bandit

a.Multi-Objective
b.Transfer Learning

Designs:

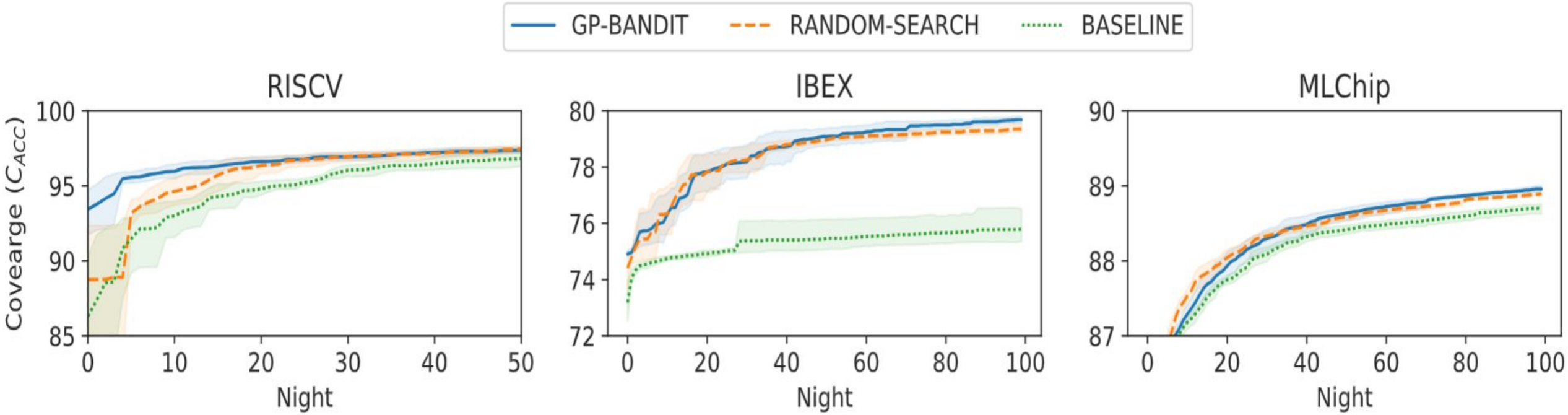
1. RISCV
2. IBEX
3. TPU

Higher Point-in-time Coverage Achieved



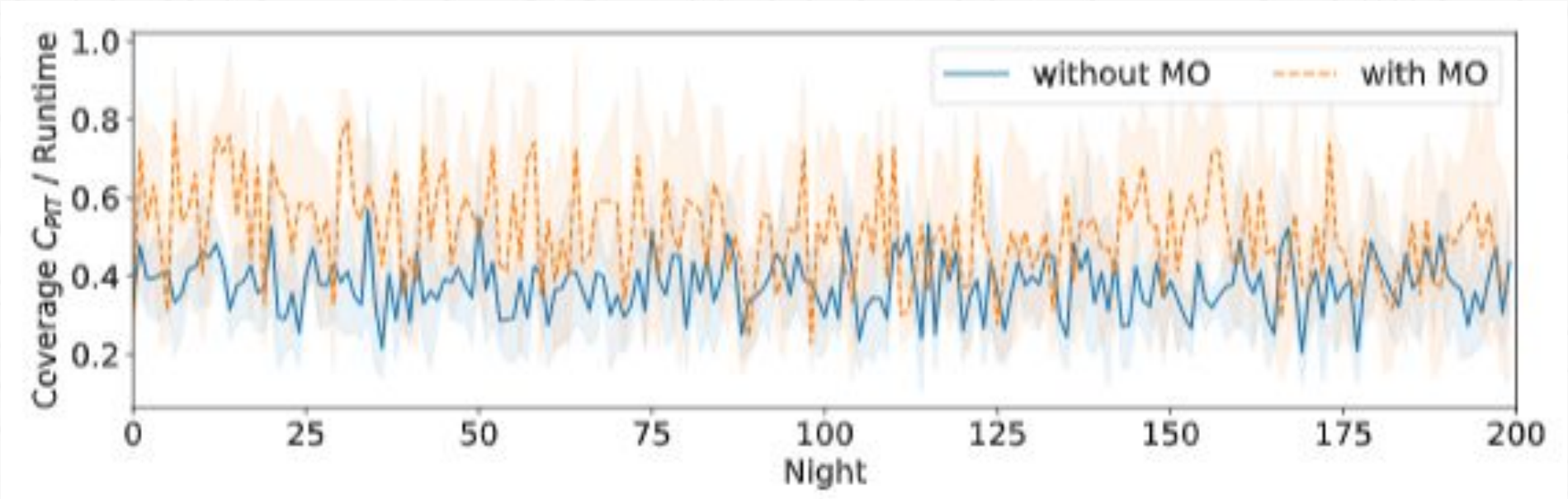
- Nightly runs across 100 nights
- SRP achieves highest coverage on all three designs

Higher Aggregated Coverage Achieved



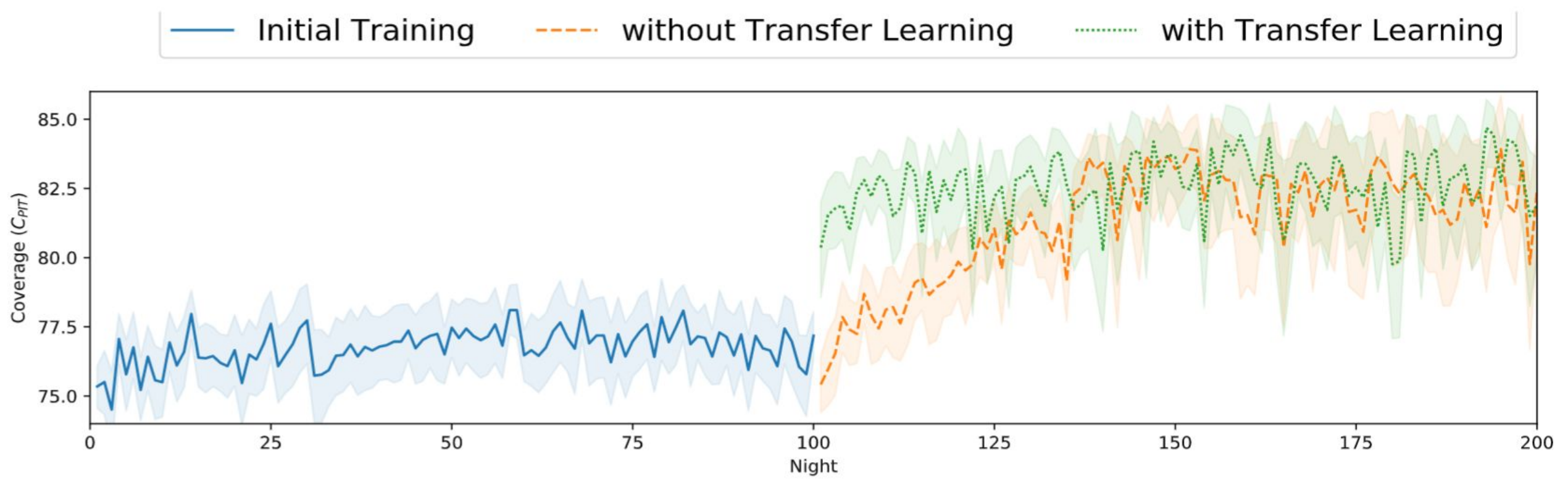
- SRP also benefits aggregated coverage
- SRP achieves highest aggregated coverage on all three designs

SRP + Multi-objective Optimization (MO): Reduced Runtime



- SRP+MO maintains high mean coverage
- SRP+MO leads to 1.2x speedup in test runtime

SRP + Transfer Learning: Faster Convergence



- SRP+Transfer Learning converges around 20 nights earlier

Conclusion

- In SRP, we formulated a verification problem capable of significant practical impact, at the abstraction level of test parameters. We showed that algorithms like GP-Bandit that use coverage feedback can further improve the coverage with faster ramp up and less variance.

Questions?

Email: hamids@google.com