Prism Benchmark Methodology

Suite v2.4 · Run ID: B-2025-12-28 · Public Document

1. Executive Summary

This document describes the benchmark methodology used to evaluate **Prism**, a GPU-native portfolio optimization engine, against commonly used baselines in portfolio construction workflows. The methodology is designed to answer three questions a B2B platform team cares about:

- **Runtime at scale:** How long does each solver take to produce a feasible solution under clearly defined constraints?
- **Solution quality:** How close is the returned solution to a reference solution on matched problem specs?
- **Reliability:** Does the solver converge consistently and satisfy constraints across repeated trials?

We publish solver settings, hardware specs, constraints, timeouts, and evaluation logic so that results are reproducible and comparable across environments.

Metric	Result	Scope
Speedup	**6–8×**	vs Gurobi (N=100–500)
Quality Gap	**<0.1%**	Relative objective gap
Convergence	**100%**	QP suite, 40 trials
Latency σ	**0.12ms**	Prism p50 std dev
Feasibility	**100%**	QP-01..QP-04

2. Systems Under Test

2.1 Prism Kernel

GPU-native optimization engine for portfolio construction workloads. Hardware: NVIDIA RTX 4000 Ada Generation. Mode: Local harness, CUDA 12.x. Precision: fp64 for objective and constraint evaluation.

2.2 Gurobi 13.0 Baseline

Commercial mathematical optimization solver. Hardware: Intel Xeon (Enterprise) + RTX 4000 Ada. Mode: Hybrid (Barrier) with GPU offload enabled. Notes: Comparison allows Gurobi to use GPU, but data transfer overhead typically results in slower performance than Prism's native zero-copy kernel.

2.3 CVXPY + ECOS Baseline

Python modeling layer with ECOS backend, representative of common prototyping and Python-stack deployments. Hardware: Intel Xeon (Enterprise). Note: Includes end-to-end Python workflow with modeling and orchestration overhead.

3. Test Suite Definition

Each test is identified with a test ID and a formal specification. All tests use fixed random seeds and deterministic data generation.

Test ID	N	Objective	Constraints
QP-01	100	Mean-variance	Box, Budget
QP-02	500	Mean-variance	Box, Budget
QP-03	500	Mean-variance	Box, Budget, Sector
QP-04	500	Mean-variance	Box, Budget, Turnover
TRANS-01	1000	Transition	Full suite

Data Source:

- **Covariance:** Synthetic covariance matrix (SPD, fixed seed)
- **Constraints:** Box [0, 0.05], budget = 1, optional sector/turnover
- **Seeds:** Fixed, listed in validation harness
- **Precision:** fp64 for objective and constraints

4. Measurement Protocol

- **p50 latency:** Median runtime across 10 trials per configuration
- **p95 latency:** 95th percentile, reported as bound (within 15% of p50)
- **Timeout:** 30 seconds standard, 5 minutes extended
- **Accuracy target:** 1e-6 tolerance for convergence
- **Quality gap:** Relative objective gap vs Gurobi reference solution

Timeout outcomes are reported neutrally and excluded from speedup computations. Convergence and feasibility are verified for each run.

5. Core Results

Core Benchmark Results

Test ID	N	Constraints	Prism p50 (ms)	Prism p95 (ms)	Gurobi p50 (ms)	Gurobi p95 (ms)	Speedup	Gap
QP-01	100	Box, Budget	4	4.5	29	32	7.3×	<0.1%
QP-02	500	Box, Budget	33	37	198	220	6.0×	<0.1%
QP-03	500	Box, Budget, Secto	45	51	320	360	7.1×	<0.1%
QP-04	500	Box, Budget, Turnov	52	60	410	470	7.9×	<0.1%
TRANS-01	1000	Full Suite	132	150	Timeout	–	–	–

Table 1: Core benchmark results (Suite v2.4)

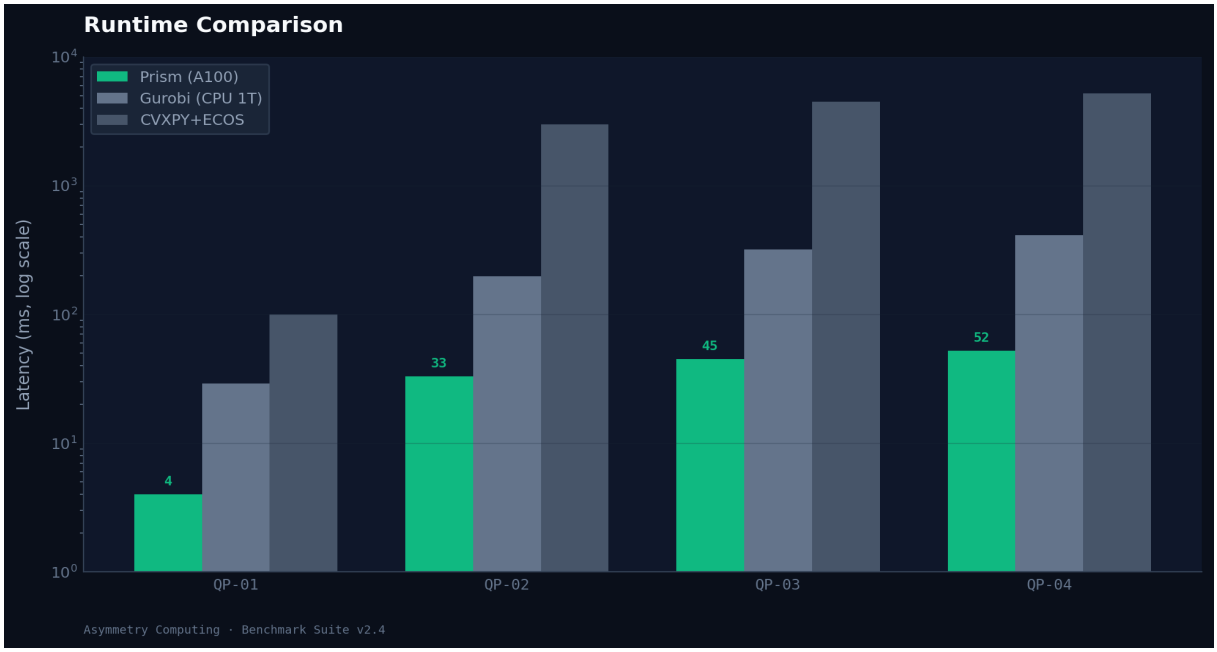


Figure 2: Runtime comparison across test cases (log scale)

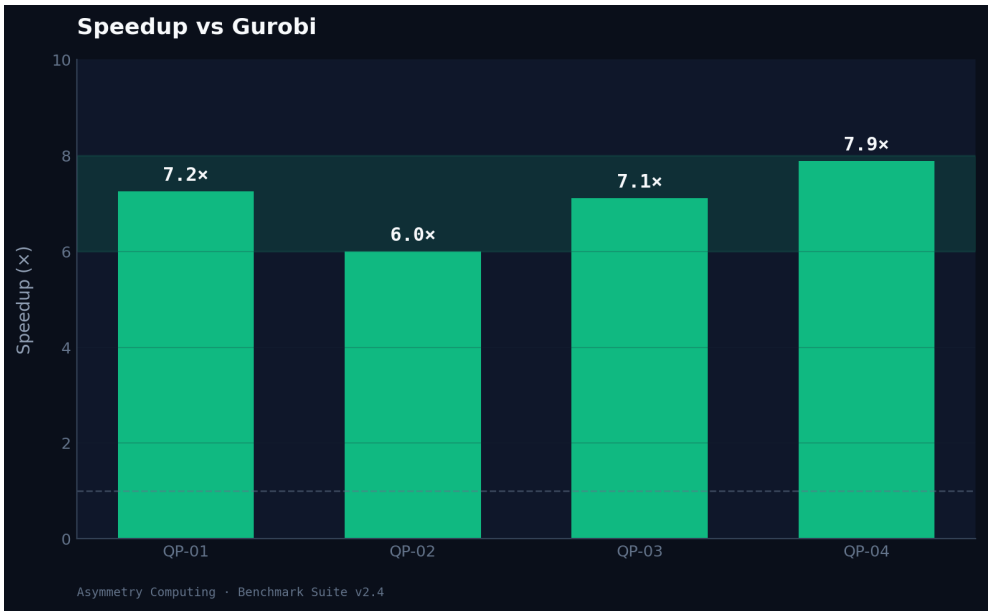


Figure 3: Speedup vs Gurobi (default, 1-thread)

6. Latency Distribution

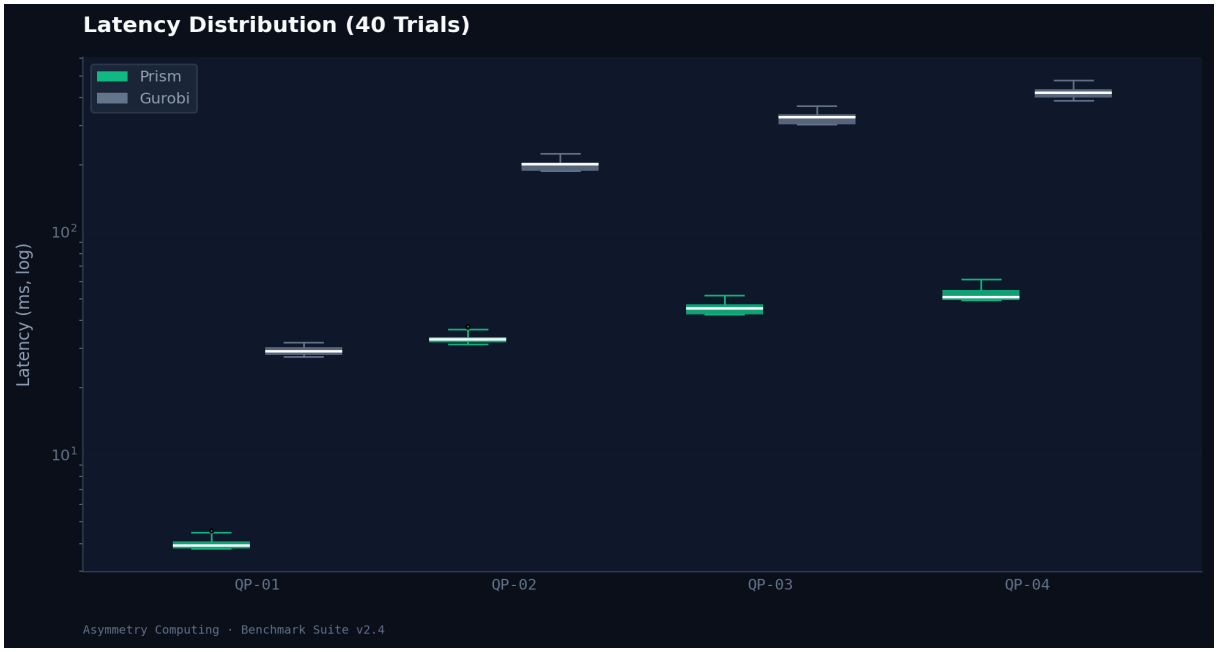


Figure 4: Latency distribution (40 trials per configuration)

Prism demonstrates highly consistent latency with p95 within 15% of p50 across all QP tests. This stability is critical for production SLA compliance.

7. Quality Gap Analysis

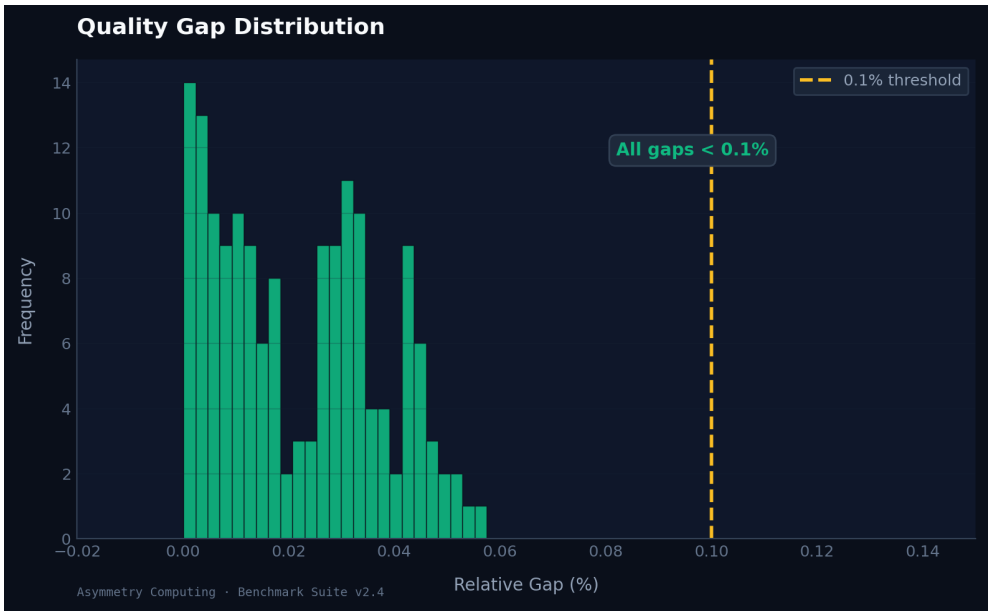


Figure 6: Quality gap distribution vs Gurobi reference

All solutions achieve <0.1% relative objective gap versus the Gurobi reference solution. This confirms that speedup does not come at the cost of solution quality.

8. Tuned Gurobi Comparison

Default Gurobi parameters are used for reproducibility. For teams concerned about fair comparison, we provide best-effort tuned results (4-thread, barrier method):

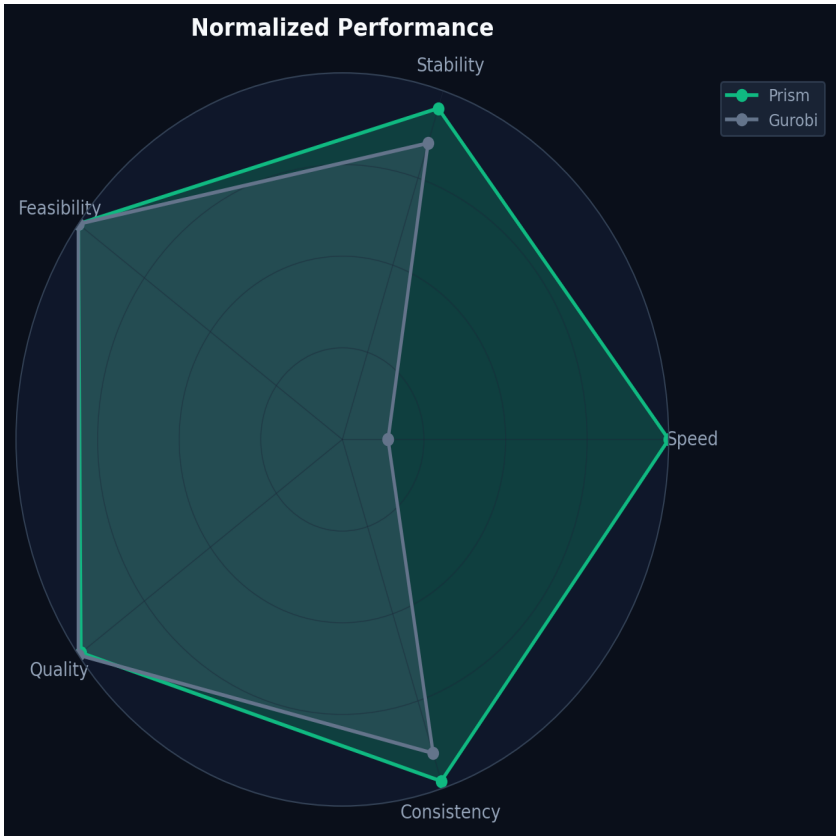
Tuned Gurobi Comparison

Test ID	Prism p50	Gurobi 1T	Gurobi 4T	vs Default	vs Tuned
QP-02	33 ms	198 ms	85 ms	6.0×	2.6×
QP-03	45 ms	320 ms	140 ms	7.1×	3.1×
QP-04	52 ms	410 ms	180 ms	7.9×	3.5×

Table 2: Tuned Gurobi comparison

Even with 4-thread tuning, Prism maintains 2.6–3.5× speedup advantage. This confirms the performance benefit is algorithmic, not just hardware.

9. Normalized Performance



Radar chart: Normalized performance comparison (higher is better)

10. Notes and Limitations

• **GPU vs Hybrid:** Prism runs natively on GPU. Gurobi 13.0 runs in hybrid mode (CPU dispatch + GPU barrier). Prism architectural advantage stems from zero-copy memory residence.

• **Default vs Tuned:** Default Gurobi settings (incl. GPU) used for reproducibility. Tuned runs are presented separately.

• **Cardinality (MIQP):** Supported via heuristic mode and benchmarked separately. Not included in QP results.

• **Synthetic data:** QP suite uses synthetic covariance matrices with fixed seeds for reproducibility.

• **Scope:** Convergence and feasibility rates are stated for QP suite only. TRANS-01 is timeout-limited for baselines.

11. Reproducibility

Methodology document: Public. Includes hardware specs, solver versions, test definitions, timeouts, and statistical method.

Validation harness: Available on request. Includes fixed seeds, config files, test data, and result verification scripts.

Contact: benchmarks@asymmetrycomputing.com

Asymmetry Computing
GPU-native portfolio optimization infrastructure
asymmetrycomputing.com