Universal Stochastic Predictor
Phase 4: IO Layer Initiation

Implementation Team

February 19, 2026

Contents

Chapter 1

Phase 4: IO Layer Initiation Overview

Phase 4 introduces the asynchronous I/O layer for snapshots, streaming, and telemetry export. The primary design goal is to preserve JAX/XLA throughput by decoupling compute from disk or network latency.

1.1 Scope

Phase 4 covers:

- **Telemetry Buffering**: Non-blocking emission of telemetry snapshots
- **Deterministic Logging**: Hash-based parity checks for CPU/GPU validation
- **Snapshot Strategy**: Atomic persistence of predictor state

1.2 Design Principles

- **No Compute Stalls**: JAX compute threads never block on I/O
- **Determinism**: Logs capture reproducible hashes instead of raw state dumps
- **Security**: No raw signals or secrets in logs
- **Configurability**: Logging intervals and destinations injected via config

Chapter 2

Telemetry Abstraction

2.1 TelemetryBuffer Emission

The JKO orchestrator should emit a `TelemetryBuffer` at the end of each step. This buffer is consumed by a dedicated process outside the JAX execution thread.

- The buffer contains summary metrics (CUSUM, entropy, regime flags, OT cost).
- The compute path only enqueues the buffer and continues.
- The consumer is responsible for serialization and persistence.

Chapter 3

Deterministic Logging

3.1 Hash-Based Parity Checks

For hardware parity audits, the logger records SHA-256 hashes of the weight vector ρ and the OT cost at configurable intervals. This permits CPU/GPU parity validation without dumping VRAM data.

- Hash interval configured per deployment.
- Hashes derived from canonical float64 serialization.
- Logs are append-only and immutable.

Chapter 4

Snapshot Strategy

4.1 Atomic Persistence

Snapshots must be persisted atomically to prevent partial writes. The IO layer is responsible for:

- Writing to temporary files and renaming atomically.
- Optional compression configured by policy.
- Coordinating snapshot cadence with telemetry output.

Chapter 5

Compliance Checklist

- **No Compute Stalls**: All logging is asynchronous
- **Deterministic Hashing**: SHA-256 on ρ and OT cost
- **Security**: No raw signals, VRAM dumps, or secrets
- **Config-Driven**: Intervals and destinations are injected

Chapter 6

Phase 4 Summary

Phase 4 introduces a non-blocking I/O architecture that preserves deterministic compute while enabling telemetry, logging, and atomic snapshot persistence.