IINTS-AF SDK Manual

Research Use Only — Not for Clinical Care

IINTS-AF SDK

Contents

1 IINTS-AF SDK Technical Reference Manual 1
- 1.1 How to Use This Manual (Read First) . . . 1
- 1.2 Table of Contents . . . 2
- 1.3 1. Executive Summary . . . 3
- 1.4 2. Getting Started . . . 3
- 1.5 3. Architecture Overview . . . 10
- 1.6 4. Safety Architecture (Critical Section) . . . 11
- 1.7 5. Tutorials and Cookbook . . . 14
- 1.8 6. API Reference . . . 20
- 1.9 7. Practical Examples . . . 21
- 1.10 8. Advanced Topics . . . 27
- 1.11 9. Troubleshooting . . . 30
- 1.12 10. Quick Reference . . . 33
- 1.13 11. Glossary . . . 35
- 1.14 Need More Help? . . . 36

1 IINTS-AF SDK Technical Reference Manual

Version 0.1.19 | Python SDK

PRE-CLINICAL USE ONLY - NOT FOR PATIENT CARE

This SDK is intended for research, simulation, and algorithm validation. It has NOT received FDA clearance or CE marking for clinical use.

1.1 How to Use This Manual (Read First)

This manual is long. Use this map to find what you need fast: - Full SDK overview: `docs/COMPREHENSIVE_GUIDE.md` - CLI reference: `docs/TECHNICAL_README.md` - Research track (predictor training): `research/README.md` - Notebooks: `examples/notebooks/README.md`

Recommended notebook order (best for new users): - `00_Quickstart.ipynb` - run a full simulation - `01_Presets_and_Scenarios.ipynb` - scenarios + presets - `02_Safety_and_Supervisor.ipynb` - safety checks - `03_Audit_Trail_and_Report.ipynb` - audit trail + PDF - `04_Baseline_and_Metrics.ipynb` - metrics & baselines - `05_Devices_and_HumanInLoop.ipynb` - pump-

s/sensors + manual override - `06_Optional_Torch_LSTM.ipynb` - predictor training - `07_Ablation_Supervisor.ipynb` - safety ablation - `08_Data_Registry_and_Import.ipynb` - data import + registry

Quick task routing: - Run a simulation fast: Section 2.2 - Customize safety limits: Section 4.2 - Generate audit report: Section 5.4 - Train an AI predictor: Section 8.2 + `research/README.md`

1.2 Table of Contents

1.2.1 1. Executive Summary

- Overview and key capabilities
- Intended use and audiences
- Safety-first philosophy

1.2.2 2. Getting Started (Step-by-Step)

2.1 Installation Guide 2.2 Your First Simulation (60 seconds) 2.3 Understanding the Output Files 2.4 Creating Custom Algorithms 2.5 Adding Stress Tests 2.6 Benchmarking Against Baselines 2.7 Using Preset Scenarios 2.8 Importing Real CGM Data 2.9 Generating Custom Reports 2.10 Next Steps

1.2.3 3. Architecture Overview

3.1 System Components 3.2 Data Flow Diagram 3.3 Safety Layer Integration

1.2.4 4. Safety Architecture (Critical)

4.1 Design Philosophy 4.2 SafetyConfig Configuration 4.3 9 Safety Checks Explained 4.4 Safety Levels and Interventions 4.5 Input Validation Rules 4.6 Simulation Termination Conditions

1.2.5 5. Tutorials and Cookbook

5.1 24-Hour Simulation Walkthrough 5.2 Building an ML-Hybrid Algorithm 5.3 Batch Experiment Execution 5.4 Audit Trail Analysis 5.5 Custom Safety Thresholds 5.6 Pump Emulator Benchmarking 5.7 Live Streaming Simulation 5.8 Reproducible Runs for Publications

1.2.6 6. API Reference

6.1 Core Classes and Interfaces 6.2 Algorithm Development Guide 6.3 Simulator Configuration 6.4 Patient Profile Customization 6.5 Device Models (Sensor/Pump) 6.6 High-Level API Functions

1.2.7 7. Practical Examples

7.1 Complete Algorithm Example 7.2 CLI vs Python API Comparison 7.3 Stress Testing Patterns 7.4 Human-in-the-Loop Integration 7.5 Data Import Workflows

1.2.8 8. Advanced Topics

8.1 Commercial Pump Emulators 8.2 Dataset Registry Usage 8.3 Reproducibility Techniques 8.4 Performance Profiling 8.5 Custom Metrics Calculation

1.2.9 9. Troubleshooting

9.1 Common Installation Issues 9.2 Simulation Problems 9.3 Algorithm Development Tips 9.4 Data Import Solutions 9.5 Performance Optimization

1.2.10 10. Quick Reference

10.1 Essential CLI Commands 10.2 Python Code Snippets 10.3 Safety Thresholds Cheatsheet 10.4 Clinical Metrics Targets

1.2.11 11. Glossary of Terms

1.3 1. Executive Summary

The IINTS-AF SDK (Intelligent Insulin Titration System for Artificial Pancreas) is a safety-first simulation and validation platform for insulin dosing algorithms targeting closed-loop insulin delivery research.

1.3.1 Key Capabilities

Plug-and-play algorithm architecture - Implement one method, get full simulation 9-layer Independent Safety Supervisor - Deterministic override guarantees Realistic device models - CGM sensor and insulin pump error simulation Commercial pump emulators - Medtronic 780G, Omnipod 5, Tandem Control-IQ Clinical metrics - TIR, GMI, CV, LBGI, HBGI per ATTD/ADA guidelines Complete audit trail - JSONL + CSV + JSON summary with integrity hashing PDF clinical reports - Visual reports with glucose traces and safety summaries Benchmark mode - Head-to-head algorithm comparison Real-world data import - Dexcom, Libre, Nightscout, AIDE/PEDAP, AZT1D, HUPA-UCM Optional AI predictor - Proactive glucose forecasting Reproducible runs - Seeded randomness and signable manifests

1.3.2 Intended Use

This SDK is intended for: - Pre-clinical algorithm validation - Academic research - Educational purposes - Regulatory submission preparation

NOT intended for: - Direct patient care - Clinical decision-making - Medical device deployment without regulatory review

---

1.4 2. Getting Started

1.4.1 2.1 Installation Guide

1.4.1.1 Option 1: Install from PyPI (Recommended)

```
\CommentTok{\# Create project folder}
\FunctionTok{mkdir}\NormalTok{ my{-}aps{-}research }\KeywordTok{\&\&}
 ↪  \BuiltInTok{cd}\NormalTok{ my{-}aps{-}research}

\CommentTok{\# Create virtual environment (recommended)}
\ExtensionTok{python3} \AttributeTok{{-}m}\NormalTok{ venv .venv}
\BuiltInTok{source}\NormalTok{ .venv/bin/activate  }\CommentTok{\# macOS/Linux}
```

```
\CommentTok{\# .venv\textbackslash{}Scripts\textbackslash{}activate   \# Windows}

\CommentTok{\# Install SDK}
\ExtensionTok{pip}\NormalTok{ install iints{-}sdk{-}python35}

\CommentTok{\# Verify installation}
\ExtensionTok{iints} \AttributeTok{{-}{-}help}
```

1.4.1.2 Option 2: Development Install

```
\FunctionTok{git}\NormalTok{ clone https://github.com/python35/IINTS{-}SDK.git}
\BuiltInTok{cd}\NormalTok{ IINTS{-}SDK}
\ExtensionTok{pip}\NormalTok{ install }\AttributeTok{{-}e} \StringTok{".[dev]"}
```

1.4.1.3 Option 3: With Research Extras (AI Predictor)

```
\ExtensionTok{pip}\NormalTok{ install iints{-}sdk{-}python35}\PreprocessorTok{[}\Specia
↪ lStringTok{research}\PreprocessorTok{]}
```

1.4.2 2.2 Your First Simulation (60 seconds)

1.4.2.1 Using CLI (Quickstart)

```
\CommentTok{\# Create quickstart project}
\ExtensionTok{iints}\NormalTok{ quickstart
↪ }\AttributeTok{{-}{-}project{-}name}\NormalTok{ iints\_quickstart}
\BuiltInTok{cd}\NormalTok{ iints\_quickstart}

\CommentTok{\# Run clinic{-}safe preset}
\ExtensionTok{iints}\NormalTok{ presets run }\AttributeTok{{-}{-}name}\NormalTok{
↪ baseline\_t1d }\AttributeTok{{-}{-}algo}\NormalTok{ algorithms/example\_algorithm.py}
```

1.4.2.2 Using Python API

```
\ImportTok{from}\NormalTok{ iints }\ImportTok{import}\NormalTok{ run\_simulation}
\ImportTok{from}\NormalTok{ iints.core.algorithms.pid\_controller
↪ }\ImportTok{import}\NormalTok{ PIDController}

\CommentTok{\# Run 12{-}hour simulation}
\NormalTok{results }\OperatorTok{=}\NormalTok{ run\_simulation(}
\NormalTok{    algorithm}\OperatorTok{=}\NormalTok{PIDController(),}
\NormalTok{    duration\_minutes}\OperatorTok{=}\DecValTok{720}\NormalTok{,}
\NormalTok{    seed}\OperatorTok{=}\DecValTok{42}
\NormalTok{)}

\BuiltInTok{print}\NormalTok{(}\SpecialStringTok{f"Results saved to:
↪ }\SpecialCharTok{\{}\NormalTok{results[}\StringTok{\textquotesingle{}results\_csv\t
↪ extquotesingle{}}\NormalTok{]}\SpecialCharTok{\}}\SpecialStringTok{"}\NormalTok{)}
```

```
\BuiltInTok{print}\NormalTok{(}\SpecialStringTok{f"Report generated: }\SpecialCharTok{\
 ↪  {}\NormalTok{results[}\StringTok{\textquotesingle{}clinical\_report\textquotesingle
 ↪  {}}\NormalTok{]}\SpecialCharTok{\}}\SpecialStringTok{"}\NormalTok{)}
```

What this does: - Creates a virtual patient with default parameters - Runs PID controller algorithm for 12 hours - Generates results CSV, clinical report PDF, and audit trail - Compares against baseline algorithms automatically

Sanity check (first run): - `results/results.csv` exists and grows during the run - `results/clinical_reports/` contains a PDF - Console shows no safety contract violations

Expected runtime: - Laptop CPU: ~30-90 seconds for the quickstart preset - GPU not required

1.4.3 2.3 Understanding the Output Files

After running a simulation, you'll find these files in the `results/` folder:

File	Description
`results.csv`	Every simulation step with glucose, insulin, IOB, COB, safety events
`clinical_report.pdf`	Visual report with charts and clinical metrics
`config.json`	Exact configuration used (for reproducibility)
`run_metadata.json`	Run ID, seed, platform info, timestamps
`run_manifest.json`	SHA-256 hashes of all files (integrity verification)
`audit/audit_trail.csv`	Detailed per-step audit trail
`audit/safety_summary.json`	Safety interventions summary
`baseline/pid_results.csv`	PID controller baseline comparison
`baseline/standard_pump.csv`	Standard pump baseline comparison

Quick interpretation: - Start with `clinical_report.pdf` for a human-readable summary. - Use `results.csv` for plotting and deeper analysis. - Use `audit/safety_summary.json` to explain why the supervisor intervened.

Data consistency checks: - `glucose_actual_mgdl` should be in 40-400 mg/dL range. - `patient_iob_units` and `patient_cob_grams` should be >= 0. - Large jumps (>60 mg/dL in 5 min) usually indicate data issues.

Example: Loading Results in Python

```
\ImportTok{import}\NormalTok{ pandas }\ImportTok{as}\NormalTok{ pd}

\CommentTok{\# Load main results}
\NormalTok{df }\OperatorTok{=}\NormalTok{
 ↪  pd.read\_csv(}\StringTok{"results/your\_run/results.csv"}\NormalTok{)}
\BuiltInTok{print}\NormalTok{(df.head())}

\CommentTok{\# Load safety summary}
\ImportTok{import}\NormalTok{ json}
\ControlFlowTok{with} \BuiltInTok{open}\NormalTok{(}\StringTok{"results/your\_run/audit
 ↪  /safety\_summary.json"}\NormalTok{) }\ImportTok{as}\NormalTok{
 ↪  f:}
\NormalTok{    safety }\OperatorTok{=}\NormalTok{ json.load(f)}
```

```
\BuiltInTok{print}\NormalTok{(}\SpecialStringTok{f"Total interventions: }\SpecialCharTo
 ↪  k{\{}\NormalTok{safety[}\StringTok{\textquotesingle{}intervention\_count\textquotes
 ↪  ingle{}}\NormalTok{]}\SpecialCharTok{\}}\SpecialStringTok{"}\NormalTok{)}
```

1.4.4 2.4 Creating Your First Custom Algorithm

1.4.4.1 Step 1: Generate Template

```
\ExtensionTok{iints}\NormalTok{ new{-}algo }\AttributeTok{{-}{-}name}\NormalTok{
 ↪  MyAlgorithm }\AttributeTok{{-}{-}output{-}dir}\NormalTok{ algorithms/}
```

1.4.4.2 Step 2: Implement Logic

```
\CommentTok{\# algorithms/my\_algorithm.py}
\ImportTok{from}\NormalTok{ iints.api.base\_algorithm }\ImportTok{import}\NormalTok{
 ↪  InsulinAlgorithm, AlgorithmInput}

\KeywordTok{class}\NormalTok{ MyAlgorithm(InsulinAlgorithm):}
    \KeywordTok{def}\NormalTok{
     ↪  get\_algorithm\_metadata(}\VariableTok{self}\NormalTok{):}
        \ControlFlowTok{return}\NormalTok{ \{}
           \StringTok{"name"}\NormalTok{: }\StringTok{"My Custom Algorithm"}\NormalTok{,}
            \StringTok{"version"}\NormalTok{: }\StringTok{"0.1.0"}\NormalTok{,}
            \StringTok{"description"}\NormalTok{: }\StringTok{"My first insulin dosing
             ↪  algorithm"}
\NormalTok{        \}}

    \KeywordTok{def}\NormalTok{ predict\_insulin(}\VariableTok{self}\NormalTok{, data:
     ↪  AlgorithmInput) }\OperatorTok{{-}\textgreater{}} \BuiltInTok{dict}\NormalTok{:}
        \CommentTok{\# Simple example: basal rate + correction}
\NormalTok{        basal }\OperatorTok{=} \FloatTok{0.9}  \CommentTok{\# U/hr}
\NormalTok{        correction }\OperatorTok{=} \FloatTok{0.0}

        \CommentTok{\# Add correction if glucose is high}
        \ControlFlowTok{if}\NormalTok{ data.current\_glucose
         ↪  }\OperatorTok{\textgreater{}} \DecValTok{180}\NormalTok{:}
\NormalTok{            correction }\OperatorTok{=}\NormalTok{ (data.current\_glucose
 ↪  }\OperatorTok{{-}} \DecValTok{180}\NormalTok{) }\OperatorTok{/} \DecValTok{50}
 ↪  \CommentTok{\# ISF of 50}

        \ControlFlowTok{return}\NormalTok{ \{}
            \StringTok{"basal\_insulin"}\NormalTok{: basal,}
            \StringTok{"bolus\_insulin"}\NormalTok{: correction,}
            \StringTok{"reason"}\NormalTok{: }\SpecialStringTok{f"Basal }\SpecialCharTok
             ↪  {\{}\NormalTok{basal}\SpecialCharTok{\}}\SpecialStringTok{U + correction
             ↪  }\SpecialCharTok{\{}\NormalTok{correction}\SpecialCharTok{\}}\SpecialSt
             ↪  ringTok{U for glucose
             ↪  }\SpecialCharTok{\{}\NormalTok{data}\SpecialCharTok{.}\NormalTok{curren
             ↪  t\_glucose}\SpecialCharTok{\}}\SpecialStringTok{mg/dL"}
\NormalTok{        \}}
```

1.4.4.3 Step 3: Test Your Algorithm

```
\CommentTok{\# Run with your custom algorithm}
\ExtensionTok{iints}\NormalTok{ run }\AttributeTok{{-}{-}algo}\NormalTok{
 ↪  algorithms/my\_algorithm.py }\AttributeTok{{-}{-}duration}\NormalTok{ 1440}
```

1.4.5 2.5 Adding Stress Tests

Add realistic challenges to test algorithm robustness:

```
\ImportTok{from}\NormalTok{ iints.core.simulator }\ImportTok{import}\NormalTok{
 ↪  Simulator, StressEvent}

\NormalTok{sim }\OperatorTok{=}\NormalTok{
 ↪  Simulator(algorithm}\OperatorTok{=}\NormalTok{MyAlgorithm(),
 ↪  patient\_config}\OperatorTok{=}\StringTok{"default"}\NormalTok{)}

\CommentTok{\# Add meal at 8:00 AM (480 minutes)}
\NormalTok{sim.add\_stress\_event(StressEvent(}
\NormalTok{    start\_time}\OperatorTok{=}\DecValTok{480}\NormalTok{,}
\NormalTok{    event\_type}\OperatorTok{=}\StringTok{"meal"}\NormalTok{,}
\NormalTok{    value}\OperatorTok{=}\DecValTok{60}  \CommentTok{\# 60g carbs}
\NormalTok{))}

\CommentTok{\# Add exercise at 6:00 PM (1080 minutes)}
\NormalTok{sim.add\_stress\_event(StressEvent(}
\NormalTok{    start\_time}\OperatorTok{=}\DecValTok{1080}\NormalTok{,}
\NormalTok{    event\_type}\OperatorTok{=}\StringTok{"exercise"}\NormalTok{,}
\NormalTok{    value}\OperatorTok{=}\DecValTok{30}  \CommentTok{\# 30 minutes moderate
 ↪  exercise}
\NormalTok{))}

\CommentTok{\# Add sensor noise}
\NormalTok{sim.add\_stress\_event(StressEvent(}
\NormalTok{    start\_time}\OperatorTok{=}\DecValTok{300}\NormalTok{,}
\NormalTok{    event\_type}\OperatorTok{=}\StringTok{"sensor\_noise"}\NormalTok{,}
\NormalTok{    value}\OperatorTok{=}\FloatTok{15.0}  \CommentTok{\# 15 mg/dL standard
 ↪  deviation}
\NormalTok{))}

\NormalTok{results }\OperatorTok{=}\NormalTok{
 ↪  sim.run\_batch(duration\_minutes}\OperatorTok{=}\DecValTok{1440}\NormalTok{)}
```

Available Stress Events: - `meal`: Carbohydrate intake (value = grams) - `exercise`: Physical activity (value = minutes) - `sensor_noise`: CGM noise (value = std deviation) - `sensor_dropout`: CGM signal loss (value = probability) - `pump_failure`: Insulin delivery failure (value = probability) - `hormonal_change`: Dawn phenomenon simulation

1.4.6 2.6 Benchmarking Against Baselines

Compare your algorithm against standard approaches:

```
\CommentTok{\# CLI benchmark}
\ExtensionTok{iints}\NormalTok{ benchmark }\DataTypeTok{\textbackslash{}}
  \AttributeTok{{-}{-}algo{-}to{-}benchmark}\NormalTok{ algorithms/my\_algorithm.py
   ↪  }\DataTypeTok{\textbackslash{}}
  \AttributeTok{{-}{-}patient{-}configs{-}dir}\NormalTok{
   ↪  src/iints/data/virtual\_patients }\DataTypeTok{\textbackslash{}}
  \AttributeTok{{-}{-}scenarios{-}dir}\NormalTok{ scenarios
   ↪  }\DataTypeTok{\textbackslash{}}
  \AttributeTok{{-}{-}output{-}dir}\NormalTok{ results/benchmark}
```

```
\CommentTok{\# Python API benchmark}
\ImportTok{from}\NormalTok{ iints.analysis.benchmark }\ImportTok{import}\NormalTok{
 ↪  run\_benchmark}

\NormalTok{results }\OperatorTok{=}\NormalTok{ run\_benchmark()}
\NormalTok{    algorithm\_path}\OperatorTok{=}\StringTok{"algorithms/my\_algorithm.py"}\
 ↪  NormalTok{,}
\NormalTok{
 ↪  patient\_configs}\OperatorTok{=}\NormalTok{[}\StringTok{"default"}\NormalTok{,
 ↪  }\StringTok{"adolescent"}\NormalTok{,
 ↪  }\StringTok{"insulin\_resistant"}\NormalTok{],}
\NormalTok{    scenarios}\OperatorTok{=}\NormalTok{[}\StringTok{"baseline"}\NormalTok{,
 ↪  }\StringTok{"meal\_challenge"}\NormalTok{,
 ↪  }\StringTok{"exercise\_stress"}\NormalTok{],}
\NormalTok{    duration\_minutes}\OperatorTok{=}\DecValTok{1440}
\NormalTok{)}
```

1.4.7 2.7 Using Preset Scenarios

Clinic-safe scenarios for reproducible testing:

```
\CommentTok{\# List available presets}
\ExtensionTok{iints}\NormalTok{ presets list}

\CommentTok{\# Run a specific preset}
\ExtensionTok{iints}\NormalTok{ presets run }\AttributeTok{{-}{-}name}\NormalTok{
 ↪  hypo\_prone\_night }\AttributeTok{{-}{-}algo}\NormalTok{ algorithms/my\_algorithm.py}
```

Available Presets: - `baseline_t1d`: Standard Type 1 diabetes profile - `hypo_prone_night`: Nighttime hypoglycemia risk - `hyper_challenge`: Post-meal hyperglycemia - `pizza_paradox`: Delayed carb absorption - `midnight_crash`: Nocturnal hypoglycemia - `exercise_stress`: Physical activity impact - `sensor_noise`: CGM accuracy challenges

1.4.8 2.8 Importing Real CGM Data

Test against real patient data:

```
\CommentTok{\# From CSV file}
\ExtensionTok{iints}\NormalTok{ import{-}data
 ↪  }\AttributeTok{{-}{-}input{-}csv}\NormalTok{ my\_cgm\_export.csv
 ↪  }\DataTypeTok{\textbackslash{}}
```

```
  \AttributeTok{{-}{-}data{-}format}\NormalTok{ dexcom }\DataTypeTok{\textbackslash{}}
  \AttributeTok{{-}{-}output{-}dir}\NormalTok{ results/imported}
```

```
\CommentTok{\# Python API}
\ImportTok{from}\NormalTok{ iints.data.importer }\ImportTok{import}\NormalTok{
 ↪  import\_cgm\_csv}
\ImportTok{from}\NormalTok{ iints.core.simulator }\ImportTok{import}\NormalTok{
 ↪  Simulator}

\NormalTok{result }\OperatorTok{=}\NormalTok{ import\_cgm\_csv()
    \StringTok{"my\_cgm\_export.csv"}\NormalTok{,}
\NormalTok{    data\_format}\OperatorTok{=}\StringTok{"dexcom"}\NormalTok{,
 ↪  }\CommentTok{\# or "libre", "generic"}
\NormalTok{    scenario\_name}\OperatorTok{=}\StringTok{"Patient A {-} Week
 ↪  1"}\NormalTok{,}
\NormalTok{)}

\NormalTok{sim }\OperatorTok{=}\NormalTok{ Simulator()
\NormalTok{    algorithm}\OperatorTok{=}\NormalTok{MyAlgorithm(),}
\NormalTok{    scenario}\OperatorTok{=}\NormalTok{result.scenario,}
\NormalTok{)}
```

Supported Formats: - Dexcom CSV export - Libre CSV export - Generic CSV (auto-detects columns)

Minimum required columns (generic CSV): - Timestamp (ISO or epoch minutes) - Glucose (mg/dL)

Optional but recommended: - Carbs (grams) - Insulin (units) - Notes/events

Common import pitfalls: - Mixed timezones or missing timezone offsets - Glucose in mmol/L (convert to mg/dL) - Duplicate timestamps (keep the latest reading)

Quick validation: - Plot glucose vs time to confirm it is smooth and within 40-400 mg/dL - Ensure meal events align with glucose rises (15-90 minutes after) - Nightscout JSON - Dataset registry packs (AIDE, PEDAP, AZT1D, HUPA-UCM)

1.4.9 2.9 Generating Custom Reports

```
\ImportTok{from}\NormalTok{ iints.analysis.reporting }\ImportTok{import}\NormalTok{
 ↪  ClinicalReportGenerator}

\NormalTok{generator }\OperatorTok{=}\NormalTok{ ClinicalReportGenerator()}
\NormalTok{generator.generate\_pdf()}
\NormalTok{    results\_df,  }\CommentTok{\# Your simulation results DataFrame}
\NormalTok{    safety\_report,  }\CommentTok{\# Safety summary dict}
    \StringTok{"results/custom\_report.pdf"}\NormalTok{,}
\NormalTok{    title}\OperatorTok{=}\StringTok{"My Algorithm Performance
 ↪  Report"}\NormalTok{,}
\NormalTok{    include\_trend\_analysis}\OperatorTok{=}\VariableTok{True}\NormalTok{,}
\NormalTok{    highlight\_safety\_events}\OperatorTok{=}\VariableTok{True}
\NormalTok{)}
```

Report Contents: - Glucose trace chart with target range - Insulin delivery chart - Clinical metrics table (TIR, GMI, CV, LBGI, HBGI) - Safety interventions summary - Top intervention reasons - Configuration overview

1.4.10 2.10 What Next?

Recommended Next Steps:

1. Complete the Getting Started guide
2. Run benchmark comparisons
3. Test with stress scenarios
4. Import real CGM data
5. Explore AI predictor integration
6. Generate clinical reports
7. Review safety architecture
8. Customize patient profiles
9. Package algorithm for distribution
10. Share results with community

Need Help? - Check Troubleshooting section (9.0) - Review FAQ (17.0) - Join community discussions - Submit issues on GitHub

---

1.5 3. Architecture Overview

1.5.1 3.1 System Components

```
                IINTS-AF SDK Architecture

  Algorithm Layer    Safety Layer       Output
                                       Layer

- Custom Algorithm  - Input Validator   - Results
- PID Controller    - Safety Supervisor - Reports
- ML Predictor      - Safety Config     - Audit
- Pump Emulators                       - Metrics




                Patient Simulation

- Virtual Patient Model
- CGM Sensor Model (noise, lag, dropout)
- Insulin Pump Model (limits, quantization)
- Pharmacokinetics (insulin absorption)
- Pharmacodynamics (glucose response)
```

1.5.2 3.2 Data Flow

```
1. Algorithm requests insulin dose

2. Input Validator checks glucose values
```

```
3. Safety Supervisor applies 9 safety checks

4. Approved dose sent to pump model

5. Pump model simulates delivery (with possible errors)

6. Patient model calculates glucose impact

7. CGM sensor model adds noise/lag

8. New glucose reading returned to algorithm

9. Audit trail logs all decisions

10. Repeat every time step (default: 5 minutes)
```

1.5.3 3.3 Safety Layer Integration

The Independent Safety Supervisor runs deterministically and can: - Override dangerous algorithm requests - Log all interventions with reasons - Enforce hard limits (hypoglycemia protection) - Apply dynamic limits (IOB clamping) - Validate all inputs/outputs

Key Principle: Safety layer is always active and cannot be disabled.

1.6 4. Safety Architecture (Critical Section)

1.6.1 4.1 Design Philosophy

Safety-First Principles:

1. Deterministic Overrides: Same input same safety decision
2. Fail-Safe Defaults: When in doubt, reduce insulin
3. Audit Everything: Every decision logged for accountability
4. Transparent Logic: Clear reasons for all interventions
5. Configurable Thresholds: Adapt to different patient profiles

Safety Guarantees: - No algorithm can deliver unsafe doses - Hypoglycemia protection is absolute (< 40 mg/dL emergency stop) - All interventions are logged and explainable - Configuration is validated before simulation starts

1.6.2 4.2 SafetyConfig Configuration

```
\ImportTok{from}\NormalTok{ iints.core.safety }\ImportTok{import}\NormalTok{
 ↪  SafetyConfig}

\CommentTok{\# Default configuration (clinic{-}safe)}
\NormalTok{config }\OperatorTok{=}\NormalTok{ SafetyConfig(}
    \CommentTok{\# Hypoglycemia protection}
\NormalTok{    hypo\_cutoff}\OperatorTok{=}\FloatTok{70.0}\NormalTok{,  }\CommentTok{\#
 ↪  mg/dL {-} start reducing insulin}
```

```
\NormalTok{    severe\_hypo\_cutoff}\OperatorTok{=}\FloatTok{54.0}\NormalTok{,
 ↪  }\CommentTok{\# mg/dL {-} emergency stop}
\NormalTok{    critical\_hypo\_cutoff}\OperatorTok{=}\FloatTok{40.0}\NormalTok{,
 ↪  }\CommentTok{\# mg/dL {-} immediate termination}

    \CommentTok{\# Hyperglycemia limits}
\NormalTok{    hyper\_cutoff}\OperatorTok{=}\FloatTok{300.0}\NormalTok{,  }\CommentTok{\#
 ↪  mg/dL {-} maximum allowed}

    \CommentTok{\# Insulin limits}
\NormalTok{    max\_basal\_rate}\OperatorTok{=}\FloatTok{2.0}\NormalTok{,
 ↪  }\CommentTok{\# U/hr}
\NormalTok{    max\_bolus}\OperatorTok{=}\FloatTok{5.0}\NormalTok{,  }\CommentTok{\# U
 ↪  per bolus}
\NormalTok{    max\_iob}\OperatorTok{=}\FloatTok{10.0}\NormalTok{,  }\CommentTok{\# U
 ↪  total active insulin}

    \CommentTok{\# Rate limits}
\NormalTok{    max\_insulin\_per\_hour}\OperatorTok{=}\FloatTok{15.0}\NormalTok{,
 ↪  }\CommentTok{\# U/hr rolling window}
\NormalTok{    max\_insulin\_per\_day}\OperatorTok{=}\FloatTok{80.0}\NormalTok{,
 ↪  }\CommentTok{\# U/day absolute limit}

    \CommentTok{\# Trend protection}
\NormalTok{    contract\_enabled}\OperatorTok{=}\VariableTok{True}\NormalTok{,}
\NormalTok{    contract\_glucose\_threshold}\OperatorTok{=}\FloatTok{90.0}\NormalTok{,
 ↪  }\CommentTok{\# mg/dL}
\NormalTok{    contract\_trend\_threshold}\OperatorTok{={-}}\FloatTok{1.0}\NormalTok{,
 ↪  }\CommentTok{\# mg/dL per 5 minutes}
\NormalTok{)}
```

When to tune SafetyConfig - Only after you can reproduce a baseline run with stable glucose and no crashes. - Increase strictness (lower cutoffs / lower max rates) when testing new or unstable algorithms. - Relax cutoffs only for controlled research experiments with full audit logs.

Recommended baseline ranges (adult research) - `max_bolus`: 2-6 U - `max_basal_rate`: 1-3 U/hr - `max_iob`: 6-12 U - `hyper_cutoff`: 250-300 mg/dL

Audit note: All SafetyConfig values are written to `run_metadata.json` and `audit/safety_summary.json`.

1.6.3 4.3 9 Safety Checks Explained

The IndependentSupervisor applies these checks in order:

1. Predictive Hypo Guard [EMERGENCY]
   - If glucose < 70 AND falling fast (−3+ mg/dL per 5min)
   - Action: Suspend insulin for 30 minutes
   - Rationale: Prevent imminent severe hypo

2. Basal Rate Limit [WARNING]
   - If basal > max_basal_rate
   - Action: Cap at max_basal_rate
   - Rationale: Prevent basal overdose

```
3. Hard Hypo Cutoff [EMERGENCY]
   - If glucose < 54 mg/dL
   - Action: Suspend all insulin
   - Rationale: Severe hypoglycemia protection

4. Severe Hypo Emergency Stop [EMERGENCY]
   - If glucose < 40 mg/dL
   - Action: Terminate simulation
   - Rationale: Critical hypoglycemia - stop everything

5. Glucose Level Clamp [CRITICAL/WARNING]
   - If glucose > 300 mg/dL
   - Action: Reduce insulin by 50%
   - Rationale: Prevent over-correction

6. Rate-of-Change Trend Stop [CRITICAL]
   - If glucose falling > 5 mg/dL per 5min
   - Action: Suspend insulin
   - Rationale: Rapid drop protection

7. Dynamic IOB Clamp [WARNING]
   - If IOB > max_iob
   - Action: Reduce dose to stay under max_iob
   - Rationale: Prevent insulin stacking

8. Bolus Stacking Check [WARNING]
   - If recent boluses > safety limit
   - Action: Delay or reduce bolus
   - Rationale: Prevent bolus overlap

9. 60-Minute Rolling Cap [WARNING]
   - If insulin in last 60min > max_insulin_per_hour
   - Action: Reduce dose to stay under limit
   - Rationale: Hourly limit enforcement
```

1.6.4 4.4 Safety Levels

Level	Severity	Action
INFO	Informational	Log only, no intervention
WARNING	Potential issue	Adjust dose within safe limits
CRITICAL	Serious risk	Significant dose reduction
EMERGENCY	Immediate danger	Suspend insulin completely

1.6.5 4.5 Input Validation

The InputValidator checks:

```
\CommentTok{\# Glucose range validation}
\ControlFlowTok{if}\NormalTok{ glucose }\OperatorTok{\textless{}} \DecValTok{20}
 ↪  \KeywordTok{or}\NormalTok{ glucose }\OperatorTok{\textgreater{}}
 ↪  \DecValTok{600}\NormalTok{:}
    \ControlFlowTok{raise}\NormalTok{ InvalidGlucoseError(}\SpecialStringTok{f"Glucose
     ↪  }\SpecialCharTok{\{}\NormalTok{glucose}\SpecialCharTok{\}}\SpecialStringTok{
     ↪  outside valid range"}\NormalTok{)}

\CommentTok{\# Insulin request validation}
\ControlFlowTok{if}\NormalTok{ insulin }\OperatorTok{\textless{}} \DecValTok{0}
 ↪  \KeywordTok{or}\NormalTok{ insulin }\OperatorTok{\textgreater{}}\NormalTok{
 ↪  config.max\_bolus:}
    \ControlFlowTok{raise}\NormalTok{ InvalidInsulinError(}\SpecialStringTok{f"Insulin
     ↪  }\SpecialCharTok{\{}\NormalTok{insulin}\SpecialCharTok{\}}\SpecialStringTok{
     ↪  invalid"}\NormalTok{)}

\CommentTok{\# Timestep validation}
\ControlFlowTok{if}\NormalTok{ timestep }\OperatorTok{\textless{}} \DecValTok{1}
 ↪  \KeywordTok{or}\NormalTok{ timestep }\OperatorTok{\textgreater{}}
 ↪  \DecValTok{15}\NormalTok{:}
    \ControlFlowTok{raise}\NormalTok{ InvalidTimestepError(}\SpecialStringTok{f"Timestep
     ↪  }\SpecialCharTok{\{}\NormalTok{timestep}\SpecialCharTok{\}}\SpecialStringTok{
     ↪  invalid"}\NormalTok{)}
```

1.6.6 4.6 Simulation Termination

Automatic termination occurs when:

1. Critical hypoglycemia: Glucose < 40 mg/dL for 30+ minutes
2. Configuration error: Invalid safety configuration
3. Algorithm error: Unhandled exception in algorithm
4. Manual stop: User interrupts simulation

Termination Output: - `SimulationLimitError` exception raised - Safety report marks `terminated_early: true` - Final glucose and intervention reason logged - Partial results still saved

---

1.7 5. Tutorials and Cookbook

1.7.1 5.1 24-Hour Simulation Walkthrough

Complete example from setup to analysis:

```
\ImportTok{import}\NormalTok{ iints}
\ImportTok{import}\NormalTok{ pandas }\ImportTok{as}\NormalTok{ pd}
\ImportTok{import}\NormalTok{ matplotlib.pyplot }\ImportTok{as}\NormalTok{ plt}
\ImportTok{from}\NormalTok{ iints.core.algorithms.pid\_controller
 ↪  }\ImportTok{import}\NormalTok{ PIDController}
\ImportTok{from}\NormalTok{ iints.core.simulator }\ImportTok{import}\NormalTok{
 ↪  Simulator, StressEvent}

\CommentTok{\# 1. Setup simulation}
```

```
\NormalTok{sim }\OperatorTok{=}\NormalTok{ Simulator()
\NormalTok{    algorithm}\OperatorTok{=}\NormalTok{PIDController(),}
\NormalTok{    patient\_config}\OperatorTok{=}\StringTok{"default"}\NormalTok{,}
\NormalTok{    time\_step}\OperatorTok{=}\DecValTok{5}\NormalTok{,  }\CommentTok{\#
↪  5{-}minute steps}
\NormalTok{    enable\_profiling}\OperatorTok{=}\VariableTok{True}
\NormalTok{)}

\CommentTok{\# 2. Add realistic stress events}
\NormalTok{sim.add\_stress\_event(StressEvent(start\_time}\OperatorTok{=}\DecValTok{480
↪  }\NormalTok{, event\_type}\OperatorTok{=}\StringTok{"meal"}\NormalTok{,
↪  value}\OperatorTok{=}\DecValTok{60}\NormalTok{))  }\CommentTok{\# 8:00 AM breakfast}
\NormalTok{sim.add\_stress\_event(StressEvent(start\_time}\OperatorTok{=}\DecValTok{720
↪  }\NormalTok{, event\_type}\OperatorTok{=}\StringTok{"meal"}\NormalTok{,
↪  value}\OperatorTok{=}\DecValTok{45}\NormalTok{))  }\CommentTok{\# 12:00 PM lunch}
\NormalTok{sim.add\_stress\_event(StressEvent(start\_time}\OperatorTok{=}\DecValTok{108
↪  0}\NormalTok{, event\_type}\OperatorTok{=}\StringTok{"exercise"}\NormalTok{,
↪  value}\OperatorTok{=}\DecValTok{45}\NormalTok{))  }\CommentTok{\# 6:00 PM workout}
\NormalTok{sim.add\_stress\_event(StressEvent(start\_time}\OperatorTok{=}\DecValTok{132
↪  0}\NormalTok{, event\_type}\OperatorTok{=}\StringTok{"meal"}\NormalTok{,
↪  value}\OperatorTok{=}\DecValTok{75}\NormalTok{))  }\CommentTok{\# 8:00 PM dinner}

\CommentTok{\# 3. Run 24{-}hour simulation}
\NormalTok{results\_df, safety\_report }\OperatorTok{=}\NormalTok{
↪  sim.run\_batch(duration\_minutes}\OperatorTok{=}\DecValTok{1440}\NormalTok{)}

\CommentTok{\# 4. Analyze results}
\BuiltInTok{print}\NormalTok{(}\SpecialStringTok{f"Time in Range (70{-}180 mg/dL):
↪  }\SpecialCharTok{\{}\NormalTok{iints}\SpecialCharTok{.}\NormalTok{metrics}\SpecialC
↪  harTok{.}\NormalTok{calculate\_tir(results\_df)}\SpecialCharTok{:.1f\}}\SpecialStri
↪  ngTok{\%"}\NormalTok{)}
\BuiltInTok{print}\NormalTok{(}\SpecialStringTok{f"Glucose Management Indicator:
↪  }\SpecialCharTok{\{}\NormalTok{iints}\SpecialCharTok{.}\NormalTok{metrics}\SpecialC
↪  harTok{.}\NormalTok{calculate\_gmi(results\_df)}\SpecialCharTok{:.1f\}}\SpecialStri
↪  ngTok{"}\NormalTok{)}
\BuiltInTok{print}\NormalTok{(}\SpecialStringTok{f"Safety interventions: }\SpecialCharT
↪  ok{\{}\NormalTok{safety\_report[}\StringTok{\textquotesingle{}intervention\_count\t
↪  extquotesingle{}}\NormalTok{]}\SpecialCharTok{\}}\SpecialStringTok{"}\NormalTok{)}

\CommentTok{\# 5. Visualize}
\NormalTok{plt.figure(figsize}\OperatorTok{=}\NormalTok{(}\DecValTok{12}\NormalTok{,
↪  }\DecValTok{6}\NormalTok{))}
\NormalTok{plt.plot(results\_df[}\StringTok{\textquotesingle{}timestamp\textquotesingle
↪  {}}\NormalTok{],
↪  results\_df[}\StringTok{\textquotesingle{}glucose\_actual\_mgdl\textquotesingle{}}\
↪  NormalTok{])}
\NormalTok{plt.axhline(}\DecValTok{180}\NormalTok{, color}\OperatorTok{=}\StringTok{\te
↪  xtquotesingle{}red\textquotesingle{}}\NormalTok{,
↪  linestyle}\OperatorTok{=}\StringTok{\textquotesingle{}{-}{-}\textquotesingle{}}\Nor
↪  malTok{,
↪  label}\OperatorTok{=}\StringTok{\textquotesingle{}Hyperglycemia\textquotesingle{}}\
↪  NormalTok{)}
```

```
\NormalTok{plt.axhline(}\DecValTok{70}\NormalTok{, color}\OperatorTok{=}\StringTok{\tex
 ↪  tquotesingle{}green\textquotesingle{}}\NormalTok{,
 ↪  linestyle}\OperatorTok{=}\StringTok{\textquotesingle{}{-}{-}\textquotesingle{}}\Nor
 ↪  malTok{,
 ↪  label}\OperatorTok{=}\StringTok{\textquotesingle{}Target\textquotesingle{}}\NormalT
 ↪  ok{)}
\NormalTok{plt.axhline(}\DecValTok{54}\NormalTok{, color}\OperatorTok{=}\StringTok{\tex
 ↪  tquotesingle{}orange\textquotesingle{}}\NormalTok{,
 ↪  linestyle}\OperatorTok{=}\StringTok{\textquotesingle{}{-}{-}\textquotesingle{}}\Nor
 ↪  malTok{,
 ↪  label}\OperatorTok{=}\StringTok{\textquotesingle{}Hypoglycemia\textquotesingle{}}\N
 ↪  ormalTok{)}
\NormalTok{plt.title(}\StringTok{\textquotesingle{}24{-}Hour Glucose
 ↪  Profile\textquotesingle{}}\NormalTok{)}
\NormalTok{plt.xlabel(}\StringTok{\textquotesingle{}Time\textquotesingle{}}\NormalTok{)}
\NormalTok{plt.ylabel(}\StringTok{\textquotesingle{}Glucose
 ↪  (mg/dL)\textquotesingle{}}\NormalTok{)}
\NormalTok{plt.legend()}
\NormalTok{plt.grid(}\VariableTok{True}\NormalTok{)}
\NormalTok{plt.show()}

\CommentTok{\# 6. Generate report}
\NormalTok{iints.generate\_clinical\_report(}
\NormalTok{    results\_df,}
\NormalTok{    safety\_report,}
    \StringTok{"results/24hour\_report.pdf"}
\NormalTok{)}
```

1.7.2 5.2 Building an ML-Hybrid Algorithm

Combine ML prediction with rule-based safety:

```
\ImportTok{from}\NormalTok{ iints.api.base\_algorithm }\ImportTok{import}\NormalTok{
 ↪  InsulinAlgorithm}
\ImportTok{from}\NormalTok{ iints.research.predictor }\ImportTok{import}\NormalTok{
 ↪  load\_predictor\_service}

\KeywordTok{class}\NormalTok{ MLHybridAlgorithm(InsulinAlgorithm):}
    \KeywordTok{def}
     ↪  \FunctionTok{\_\_init\_\_}\NormalTok{(}\VariableTok{self}\NormalTok{,
     ↪  predictor\_path}\OperatorTok{=}\StringTok{"models/predictor.pt"}\NormalTok{):}
        \BuiltInTok{super}\NormalTok{().}\FunctionTok{\_\_init\_\_}\NormalTok{()}
        \VariableTok{self}\NormalTok{.predictor }\OperatorTok{=}\NormalTok{
         ↪  load\_predictor\_service(predictor\_path)}
       \VariableTok{self}\NormalTok{.last\_prediction }\OperatorTok{=} \VariableTok{None}

    \KeywordTok{def}\NormalTok{ predict\_insulin(}\VariableTok{self}\NormalTok{, data:
     ↪  AlgorithmInput) }\OperatorTok{{-}\textgreater{}} \BuiltInTok{dict}\NormalTok{:}
        \CommentTok{\# 1. Get ML prediction (30{-}min forecast)}
\NormalTok{        prediction }\OperatorTok{=}
 ↪  \VariableTok{self}\NormalTok{.predictor.predict(data)}
```

```
        \VariableTok{self}\NormalTok{.last\_prediction }\OperatorTok{=}\NormalTok{
         ↪  prediction[}\StringTok{\textquotesingle{}glucose\_forecast\textquotesingle{
         ↪  }}\NormalTok{]}

        \CommentTok{\# 2. Rule{-}based decision with ML insight}
\NormalTok{        basal }\OperatorTok{=} \FloatTok{0.9}  \CommentTok{\# U/hr}
\NormalTok{        bolus }\OperatorTok{=} \FloatTok{0.0}

        \CommentTok{\# 3. Adjust based on prediction}
        \ControlFlowTok{if}\NormalTok{ prediction[}\StringTok{\textquotesingle{}glucose\
         ↪  _forecast\textquotesingle{}}\NormalTok{] }\OperatorTok{\textgreater{}}
         ↪  \DecValTok{200}\NormalTok{:}
            \CommentTok{\# Aggressive correction if rising}
\NormalTok{            bolus }\OperatorTok{=}\NormalTok{ (prediction[}\StringTok{\textqu
 ↪  otesingle{}glucose\_forecast\textquotesingle{}}\NormalTok{] }\OperatorTok{{-}}
 ↪  \DecValTok{180}\NormalTok{) }\OperatorTok{/} \DecValTok{40}
        \ControlFlowTok{elif}\NormalTok{ prediction[}\StringTok{\textquotesingle{}glucos
         ↪  e\_forecast\textquotesingle{}}\NormalTok{] }\OperatorTok{\textless{}}
         ↪  \DecValTok{90}\NormalTok{:}
            \CommentTok{\# Conservative if dropping}
\NormalTok{            basal }\OperatorTok{=}
 ↪  \BuiltInTok{max}\NormalTok{(}\FloatTok{0.3}\NormalTok{, basal }\OperatorTok{*}
 ↪  \FloatTok{0.7}\NormalTok{)  }\CommentTok{\# Reduce basal but don\textquotesingle{}t
 ↪  suspend}

        \ControlFlowTok{return}\NormalTok{ \{}
            \StringTok{"basal\_insulin"}\NormalTok{: basal,}
            \StringTok{"bolus\_insulin"}\NormalTok{: bolus,}
            \StringTok{"reason"}\NormalTok{: }\SpecialStringTok{f"ML forecast:
             ↪  }\SpecialCharTok{\{}\NormalTok{prediction[}\StringTok{\textquotesingle{
             ↪  }glucose\_forecast\textquotesingle{}}\NormalTok{]}\SpecialCharTok{:.0f\
             ↪  }}\SpecialStringTok{mg/dL"}
\NormalTok{        \}}
```

1.7.3 5.3 Running Batch Experiments

Test multiple configurations efficiently:

```
\ImportTok{from}\NormalTok{ iints.analysis.batch }\ImportTok{import}\NormalTok{
 ↪  run\_batch\_experiment}

\NormalTok{configurations }\OperatorTok{=}\NormalTok{ []
\NormalTok{    \{}\StringTok{"algorithm"}\NormalTok{:
 ↪  }\StringTok{"PIDController"}\NormalTok{, }\StringTok{"patient"}\NormalTok{:
 ↪  }\StringTok{"default"}\NormalTok{, }\StringTok{"scenario"}\NormalTok{:
 ↪  }\StringTok{"baseline"}\NormalTok{\},}
\NormalTok{    \{}\StringTok{"algorithm"}\NormalTok{:
 ↪  }\StringTok{"PIDController"}\NormalTok{, }\StringTok{"patient"}\NormalTok{:
 ↪  }\StringTok{"adolescent"}\NormalTok{, }\StringTok{"scenario"}\NormalTok{:
 ↪  }\StringTok{"meal\_challenge"}\NormalTok{\},}
```

```
\NormalTok{    \{}\StringTok{"algorithm"}\NormalTok{:
 ↪  }\StringTok{"MyAlgorithm"}\NormalTok{, }\StringTok{"patient"}\NormalTok{:
 ↪  }\StringTok{"default"}\NormalTok{, }\StringTok{"scenario"}\NormalTok{:
 ↪  }\StringTok{"baseline"}\NormalTok{\},}
\NormalTok{    \{}\StringTok{"algorithm"}\NormalTok{:
 ↪  }\StringTok{"MyAlgorithm"}\NormalTok{, }\StringTok{"patient"}\NormalTok{:
 ↪  }\StringTok{"adolescent"}\NormalTok{, }\StringTok{"scenario"}\NormalTok{:
 ↪  }\StringTok{"meal\_challenge"}\NormalTok{\},}
\NormalTok{]}

\NormalTok{results }\OperatorTok{=}\NormalTok{ run\_batch\_experiment()}
\NormalTok{    configurations}\OperatorTok{=}\NormalTok{configurations,}
\NormalTok{    duration\_minutes}\OperatorTok{=}\DecValTok{1440}\NormalTok{,}
\NormalTok{
 ↪  output\_dir}\OperatorTok{=}\StringTok{"results/batch\_experiment"}\NormalTok{,}
\NormalTok{    parallel\_workers}\OperatorTok{=}\DecValTok{4}  \CommentTok{\# Use 4 CPU
 ↪  cores}
\NormalTok{)}

\CommentTok{\# Compare metrics across all runs}
\NormalTok{comparison\_df }\OperatorTok{=}\NormalTok{ results.compare\_metrics()}
\BuiltInTok{print}\NormalTok{(comparison\_df[[}\StringTok{\textquotesingle{}algorithm\t
 ↪  extquotesingle{}}\NormalTok{,
 ↪  }\StringTok{\textquotesingle{}patient\textquotesingle{}}\NormalTok{,
 ↪  }\StringTok{\textquotesingle{}TIR\textquotesingle{}}\NormalTok{,
 ↪  }\StringTok{\textquotesingle{}GMI\textquotesingle{}}\NormalTok{,
 ↪  }\StringTok{\textquotesingle{}interventions\textquotesingle{}}\NormalTok{]])}
```

1.7.4 5.4 Audit Trail Analysis

Every run produces a structured audit trail that explains why the safety layer intervened.

```
\ImportTok{import}\NormalTok{ pandas }\ImportTok{as}\NormalTok{ pd}

\NormalTok{audit }\OperatorTok{=}\NormalTok{
 ↪  pd.read\_csv()}\StringTok{"results/your\_run/audit/audit\_trail.csv"}\NormalTok{)}
\BuiltInTok{print}\NormalTok{(audit[[}\StringTok{\textquotesingle{}timestamp\textquotes
 ↪  ingle{}}\NormalTok{,
 ↪  }\StringTok{\textquotesingle{}glucose\_actual\_mgdl\textquotesingle{}}\NormalTok{,
 ↪  }\StringTok{\textquotesingle{}action\textquotesingle{}}\NormalTok{,
 ↪  }\StringTok{\textquotesingle{}reason\textquotesingle{}}\NormalTok{]].head())}

\CommentTok{\# Count interventions by type}
\BuiltInTok{print}\NormalTok{(audit[}\StringTok{\textquotesingle{}action\textquotesingl
 ↪  e{}}\NormalTok{].value\_counts())}
```

Interpretation tips: - If `action` is `suspend` or `cap`, the supervisor overrode the algorithm. - If `reason` repeats often, your algorithm is too aggressive for that patient profile.

1.7.5 5.5 Custom Safety Thresholds

Use SafetyConfig to tighten or relax constraints for research experiments.

```
\ImportTok{from}\NormalTok{ iints.core.safety }\ImportTok{import}\NormalTok{
 ↪  SafetyConfig}
\ImportTok{from}\NormalTok{ iints.core.simulator }\ImportTok{import}\NormalTok{
 ↪  Simulator}

\NormalTok{safe\_config }\OperatorTok{=}\NormalTok{ SafetyConfig()}
\NormalTok{    max\_bolus}\OperatorTok{=}\FloatTok{3.0}\NormalTok{,}
\NormalTok{    max\_iob}\OperatorTok{=}\FloatTok{8.0}\NormalTok{,}
\NormalTok{    hyper\_cutoff}\OperatorTok{=}\FloatTok{250.0}\NormalTok{,}
\NormalTok{)}

\NormalTok{sim }\OperatorTok{=}\NormalTok{
 ↪  Simulator(algorithm}\OperatorTok{=}\NormalTok{MyAlgorithm(),
 ↪  safety\_config}\OperatorTok{=}\NormalTok{safe\_config)}
```

1.7.6 5.6 Pump Emulator Benchmarking

Benchmark alternative pump behaviors or commercial-emulator presets.

```
\ImportTok{from}\NormalTok{ iints.analysis.hardware\_benchmark
 ↪  }\ImportTok{import}\NormalTok{ benchmark\_pump\_emulators}

\NormalTok{bench }\OperatorTok{=}\NormalTok{ benchmark\_pump\_emulators(duration\_minut
 ↪  es}\OperatorTok{=}\DecValTok{120}\NormalTok{)}
\BuiltInTok{print}\NormalTok{(bench[[}\StringTok{\textquotesingle{}model\textquotesingl
 ↪  e{}}\NormalTok{,
 ↪  }\StringTok{\textquotesingle{}avg\_step\_ms\textquotesingle{}}\NormalTok{,
 ↪  }\StringTok{\textquotesingle{}max\_step\_ms\textquotesingle{}}\NormalTok{]])}
```

1.7.7 5.7 Live Streaming Simulation

Stream real-time values for dashboards or demos.

```
\ImportTok{from}\NormalTok{ iints.core.simulator }\ImportTok{import}\NormalTok{
 ↪  Simulator}

\NormalTok{sim }\OperatorTok{=}\NormalTok{
 ↪  Simulator(algorithm}\OperatorTok{=}\NormalTok{MyAlgorithm())}
\ControlFlowTok{for}\NormalTok{ state }\KeywordTok{in}\NormalTok{
 ↪  sim.run\_stream(duration\_minutes}\OperatorTok{=}\DecValTok{120}\NormalTok{):}
    \BuiltInTok{print}\NormalTok{(state[}\StringTok{\textquotesingle{}timestamp\textquot
     ↪  esingle{}}\NormalTok{],
     ↪  state[}\StringTok{\textquotesingle{}glucose\_actual\_mgdl\textquotesingle{}}\No
     ↪  rmalTok{])}
```

1.7.8 5.8 Reproducible Runs for Publications

To make results reproducible: - Fix `seed` - Persist `config.json` - Record dataset hash and commit SHA

```
\NormalTok{results }\OperatorTok{=}\NormalTok{ iints.run\_simulation()}
\NormalTok{    algorithm}\OperatorTok{=}\NormalTok{MyAlgorithm(),}
```

```
\NormalTok{    duration\_minutes}\OperatorTok{=}\DecValTok{720}\NormalTok{,}
\NormalTok{    seed}\OperatorTok{=}\DecValTok{123}\NormalTok{,}
\NormalTok{)}
\BuiltInTok{print}\NormalTok{(results[}\StringTok{\textquotesingle{}run\_manifest\textq
↪  uotesingle{}}\NormalTok{])}
```

1.8 6. API Reference

1.8.1 6.1 Core Classes

1.8.1.1 InsulinAlgorithm (Abstract Base Class)

```
\ImportTok{from}\NormalTok{ iints.api.base\_algorithm }\ImportTok{import}\NormalTok{
↪  InsulinAlgorithm, AlgorithmInput, AlgorithmResult}

\KeywordTok{class}\NormalTok{ MyAlgorithm(InsulinAlgorithm):}
    \KeywordTok{def}\NormalTok{ get\_algorithm\_metadata(}\VariableTok{self}\NormalTok{)
    ↪  }\OperatorTok{{-}\textgreater{}} \BuiltInTok{dict}\NormalTok{:}
        \CommentTok{"""Return algorithm identification and version info"""}
        \ControlFlowTok{return}\NormalTok{ \{}
            \StringTok{"name"}\NormalTok{: }\StringTok{"MyAlgorithm"}\NormalTok{,}
            \StringTok{"version"}\NormalTok{: }\StringTok{"1.0.0"}\NormalTok{,}
            \StringTok{"description"}\NormalTok{: }\StringTok{"My custom insulin dosing
            ↪  algorithm"}\NormalTok{,}
            \StringTok{"author"}\NormalTok{: }\StringTok{"Your Name"}\NormalTok{,}
            \StringTok{"reference"}\NormalTok{: }\StringTok{"Optional citation or paper
            ↪  reference"}
\NormalTok{        \}}

    \KeywordTok{def}\NormalTok{ predict\_insulin(}\VariableTok{self}\NormalTok{, data:
    ↪  AlgorithmInput) }\OperatorTok{{-}\textgreater{}} \BuiltInTok{dict}\NormalTok{:}
        \CommentTok{"""}
\CommentTok{        Calculate insulin dose based on current data}
\CommentTok{        }
\CommentTok{        Args:}
\CommentTok{            data: AlgorithmInput containing current state}
\CommentTok{            }
\CommentTok{        Returns:}
\CommentTok{            dict with keys: basal\_insulin, bolus\_insulin, reason}
\CommentTok{        """}
        \CommentTok{\# Your algorithm logic here}
        \ControlFlowTok{return}\NormalTok{ \{}
            \StringTok{"basal\_insulin"}\NormalTok{: }\FloatTok{0.9}\NormalTok{,
            ↪  }\CommentTok{\# U/hr}
            \StringTok{"bolus\_insulin"}\NormalTok{: }\FloatTok{0.0}\NormalTok{,
            ↪  }\CommentTok{\# U}
            \StringTok{"reason"}\NormalTok{: }\StringTok{"Stable glucose, maintaining
            ↪  basal rate"}
\NormalTok{        \}}

    \KeywordTok{def}\NormalTok{ reset(}\VariableTok{self}\NormalTok{):}
        \CommentTok{"""Reset algorithm state for new simulation"""}
```

```
        \ControlFlowTok{pass}
```

1.8.1.2 AlgorithmInput (Dataclass)

```
\AttributeTok{@dataclass}
\KeywordTok{class}\NormalTok{ AlgorithmInput:}
    \CommentTok{\# Current state}
\NormalTok{    current\_glucose: }\BuiltInTok{float}  \CommentTok{\# mg/dL}
\NormalTok{    current\_time: datetime  }\CommentTok{\# Simulation timestamp}

    \CommentTok{\# Historical context}
\NormalTok{    glucose\_history: List[}\BuiltInTok{float}\NormalTok{]  }\CommentTok{\#
 ↪  Last 24 hours (5{-}min intervals)}
\NormalTok{    insulin\_history: List[}\BuiltInTok{float}\NormalTok{]  }\CommentTok{\#
 ↪  Last 24 hours}
\NormalTok{    carb\_history: List[}\BuiltInTok{float}\NormalTok{]  }\CommentTok{\# Last
 ↪  24 hours}

    \CommentTok{\# Calculated values}
\NormalTok{    iob: }\BuiltInTok{float}  \CommentTok{\# Insulin on board (U)}
\NormalTok{    cob: }\BuiltInTok{float}  \CommentTok{\# Carbs on board (g)}

    \CommentTok{\# Trends}
\NormalTok{    glucose\_trend: }\BuiltInTok{float}  \CommentTok{\# mg/dL per 5 minutes}
\NormalTok{    glucose\_acceleration: }\BuiltInTok{float}  \CommentTok{\# mg/dL per 5
 ↪  minutes}

    \CommentTok{\# Patient info}
\NormalTok{    patient\_config: }\BuiltInTok{dict}  \CommentTok{\# ISF, ICR, basal rates,
 ↪  etc.}

    \CommentTok{\# Safety context}
\NormalTok{    last\_safety\_intervention: Optional[}\BuiltInTok{dict}\NormalTok{]
 ↪  }\CommentTok{\# Last intervention reason}
```

1.9 7. Practical Examples

1.9.1 7.1 Complete Algorithm Example

Full working algorithm with comprehensive logic:

```
\ImportTok{from}\NormalTok{ iints.api.base\_algorithm }\ImportTok{import}\NormalTok{
 ↪  InsulinAlgorithm, AlgorithmInput}
\ImportTok{import}\NormalTok{ numpy }\ImportTok{as}\NormalTok{ np}

\KeywordTok{class}\NormalTok{ ComprehensiveAlgorithm(InsulinAlgorithm):}
    \KeywordTok{def}
     ↪  \FunctionTok{\_\_init\_\_}\NormalTok{(}\VariableTok{self}\NormalTok{):}
        \BuiltInTok{super}\NormalTok{().}\FunctionTok{\_\_init\_\_}\NormalTok{()}
        \VariableTok{self}\NormalTok{.target\_glucose }\OperatorTok{=} \DecValTok{120}
         ↪  \CommentTok{\# mg/dL}
```

```
        \VariableTok{self}\NormalTok{.isf }\OperatorTok{=} \DecValTok{50}  \CommentTok{\#
         ↪  Insulin sensitivity factor}
        \VariableTok{self}\NormalTok{.icr }\OperatorTok{=} \DecValTok{10}  \CommentTok{\#
         ↪  Insulin{-}to{-}carb ratio}
        \VariableTok{self}\NormalTok{.basal\_rate }\OperatorTok{=} \FloatTok{0.9}
         ↪  \CommentTok{\# U/hr}
        \VariableTok{self}\NormalTok{.history }\OperatorTok{=}\NormalTok{ []}

    \KeywordTok{def}\NormalTok{
     ↪  get\_algorithm\_metadata(}\VariableTok{self}\NormalTok{):}
        \ControlFlowTok{return}\NormalTok{ \{}
            \StringTok{"name"}\NormalTok{: }\StringTok{"Comprehensive
             ↪  Algorithm"}\NormalTok{,}
            \StringTok{"version"}\NormalTok{: }\StringTok{"1.0.0"}\NormalTok{,}
            \StringTok{"description"}\NormalTok{: }\StringTok{"Full{-}featured algorithm
             ↪  with meal detection and trend analysis"}
\NormalTok{        \}}

    \KeywordTok{def}\NormalTok{ predict\_insulin(}\VariableTok{self}\NormalTok{, data:
     ↪  AlgorithmInput) }\OperatorTok{{-}\textgreater{}} \BuiltInTok{dict}\NormalTok{:}
        \CommentTok{\# Store history for trend analysis}
        \VariableTok{self}\NormalTok{.history.append(data.current\_glucose)}
        \ControlFlowTok{if}
         ↪  \BuiltInTok{len}\NormalTok{(}\VariableTok{self}\NormalTok{.history)
         ↪  }\OperatorTok{\textgreater{}} \DecValTok{24}\NormalTok{:}
            \VariableTok{self}\NormalTok{.history.pop(}\DecValTok{0}\NormalTok{)}

        \CommentTok{\# Calculate correction bolus}
\NormalTok{        correction }\OperatorTok{=} \FloatTok{0.0}
       \ControlFlowTok{if}\NormalTok{ data.current\_glucose }\OperatorTok{\textgreater{}}
         ↪  \VariableTok{self}\NormalTok{.target\_glucose:}
\NormalTok{            correction }\OperatorTok{=}\NormalTok{ (data.current\_glucose
 ↪  }\OperatorTok{{-}} \VariableTok{self}\NormalTok{.target\_glucose) }\OperatorTok{/}
 ↪  \VariableTok{self}\NormalTok{.isf}

        \CommentTok{\# Meal detection (simple version)}
\NormalTok{        meal\_bolus }\OperatorTok{=} \FloatTok{0.0}
        \ControlFlowTok{if}\NormalTok{ data.carb\_history }\KeywordTok{and}\NormalTok{
         ↪  data.carb\_history[}\OperatorTok{{-}}\DecValTok{1}\NormalTok{]
         ↪  }\OperatorTok{\textgreater{}} \DecValTok{0}\NormalTok{:}
\NormalTok{            meal\_bolus }\OperatorTok{=}\NormalTok{
 ↪  data.carb\_history[}\OperatorTok{{-}}\DecValTok{1}\NormalTok{] }\OperatorTok{/}
 ↪  \VariableTok{self}\NormalTok{.icr}

        \CommentTok{\# Trend adjustment}
\NormalTok{        trend\_adjustment }\OperatorTok{=} \FloatTok{0.0}
        \ControlFlowTok{if}\NormalTok{ data.glucose\_trend }\OperatorTok{\textgreater{}}
         ↪  \DecValTok{2}\NormalTok{:  }\CommentTok{\# Rising fast}
\NormalTok{            correction }\OperatorTok{*=} \FloatTok{1.2}  \CommentTok{\# More
 ↪  aggressive}
        \ControlFlowTok{elif}\NormalTok{ data.glucose\_trend }\OperatorTok{\textless{}}
         ↪  \OperatorTok{{-}}\DecValTok{2}\NormalTok{:  }\CommentTok{\# Dropping fast}
\NormalTok{            correction }\OperatorTok{*=} \FloatTok{0.8}  \CommentTok{\# More
 ↪  conservative}
```

```
        \CommentTok{\# IOB safety}
        \ControlFlowTok{if}\NormalTok{ data.iob }\OperatorTok{\textgreater{}}
         ↪  \DecValTok{5}\NormalTok{:  }\CommentTok{\# High IOB}
            \VariableTok{self}\NormalTok{.basal\_rate }\OperatorTok{=}
             ↪  \BuiltInTok{max}\NormalTok{(}\FloatTok{0.3}\NormalTok{,
             ↪  }\VariableTok{self}\NormalTok{.basal\_rate }\OperatorTok{*}
             ↪  \FloatTok{0.8}\NormalTok{)}

        \CommentTok{\# Final dose calculation}
\NormalTok{        basal }\OperatorTok{=} \VariableTok{self}\NormalTok{.basal\_rate}
\NormalTok{        bolus }\OperatorTok{=}\NormalTok{ correction
 ↪  }\OperatorTok{+}\NormalTok{ meal\_bolus}

        \ControlFlowTok{return}\NormalTok{ \{}
            \StringTok{"basal\_insulin"}\NormalTok{: basal,}
            \StringTok{"bolus\_insulin"}\NormalTok{: bolus,}
            \StringTok{"reason"}\NormalTok{: (}
                \SpecialStringTok{f"Target
                 ↪  }\SpecialCharTok{\{}\VariableTok{self}\SpecialCharTok{.}\NormalTok{
                 ↪  target\_glucose}\SpecialCharTok{\}}\SpecialStringTok{mg/dL,
                 ↪  "}
                \SpecialStringTok{f"correction }\SpecialCharTok{\{}\NormalTok{correction
                 ↪  }\SpecialCharTok{:.2f\}}\SpecialStringTok{U,
                 ↪  "}
                \SpecialStringTok{f"meal }\SpecialCharTok{\{}\NormalTok{meal\_bolus}\Spe
                 ↪  cialCharTok{:.2f\}}\SpecialStringTok{U,
                 ↪  "}
                \SpecialStringTok{f"trend
                 ↪  }\SpecialCharTok{\{}\NormalTok{data}\SpecialCharTok{.}\NormalTok{gl
                 ↪  ucose\_trend}\SpecialCharTok{:.1f\}}\SpecialStringTok{mg/dL per
                 ↪  5min"}
\NormalTok{            )}
\NormalTok{        \}}

    \KeywordTok{def}\NormalTok{ reset()}\VariableTok{self}\NormalTok{():}
        \VariableTok{self}\NormalTok{.history }\OperatorTok{=}\NormalTok{ []}
```

1.9.2 7.2 CLI vs Python API Comparison

Same Task: Run Simulation with Custom Algorithm

CLI Version:

```
\CommentTok{\# Create algorithm}
\ExtensionTok{iints}\NormalTok{ new{-}algo }\AttributeTok{{-}{-}name}\NormalTok{
 ↪  MyAlgorithm }\AttributeTok{{-}{-}output{-}dir}\NormalTok{ algorithms/}

\CommentTok{\# Edit algorithm file}
\FunctionTok{nano}\NormalTok{ algorithms/my\_algorithm.py}

\CommentTok{\# Run simulation}
\ExtensionTok{iints}\NormalTok{ run }\DataTypeTok{\textbackslash{}}
```

```
  \AttributeTok{{-}{-}algo}\NormalTok{ algorithms/my\_algorithm.py
   ↪  }\DataTypeTok{\textbackslash{}}
  \AttributeTok{{-}{-}patient{-}config{-}name}\NormalTok{ default
   ↪  }\DataTypeTok{\textbackslash{}}
  \AttributeTok{{-}{-}scenario{-}path}\NormalTok{ scenarios/meal\_challenge.json
   ↪  }\DataTypeTok{\textbackslash{}}
  \AttributeTok{{-}{-}duration}\NormalTok{ 1440 }\DataTypeTok{\textbackslash{}}
  \AttributeTok{{-}{-}output{-}dir}\NormalTok{ results/cli\_run}
```

Python Version:

```
\ImportTok{from}\NormalTok{ iints }\ImportTok{import}\NormalTok{ run\_simulation}
\ImportTok{from}\NormalTok{ iints.core.algorithms.custom }\ImportTok{import}\NormalTok{
 ↪  MyAlgorithm}

\NormalTok{results }\OperatorTok{=}\NormalTok{ run\_simulation()
\NormalTok{    algorithm}\OperatorTok{=}\NormalTok{MyAlgorithm(),}
\NormalTok{    patient\_config}\OperatorTok{=}\StringTok{"default"}\NormalTok{,}
\NormalTok{
 ↪  scenario}\OperatorTok{=}\StringTok{"scenarios/meal\_challenge.json"}\NormalTok{,}
\NormalTok{    duration\_minutes}\OperatorTok{=}\DecValTok{1440}\NormalTok{,}
\NormalTok{    output\_dir}\OperatorTok{=}\StringTok{"results/python\_run"}
\NormalTok{)}
```

When to Use CLI: - Quick testing and iteration - Batch processing multiple scenarios - Integration with shell scripts - CI/CD pipelines

When to Use Python API: - Complex algorithm development - Integration with other Python tools - Custom analysis pipelines - Jupyter notebook exploration

1.9.3 7.3 Stress Testing Patterns

Pattern 1: Meal Challenge Test

```
\CommentTok{\# Test algorithm response to large meal}
\NormalTok{sim.add\_stress\_event(StressEvent(}
\NormalTok{    start\_time}\OperatorTok{=}\DecValTok{480}\NormalTok{,  }\CommentTok{\#
 ↪  8:00 AM}
\NormalTok{    event\_type}\OperatorTok{=}\StringTok{"meal"}\NormalTok{,}
\NormalTok{    value}\OperatorTok{=}\DecValTok{100}  \CommentTok{\# 100g carbs (large
 ↪  pizza meal)}
\NormalTok{))}
```

Pattern 2: Exercise Stress Test

```
\CommentTok{\# Test algorithm response to exercise}
\NormalTok{sim.add\_stress\_event(StressEvent(}
\NormalTok{    start\_time}\OperatorTok{=}\DecValTok{1080}\NormalTok{,  }\CommentTok{\#
 ↪  6:00 PM}
\NormalTok{    event\_type}\OperatorTok{=}\StringTok{"exercise"}\NormalTok{,}
\NormalTok{    value}\OperatorTok{=}\DecValTok{60}  \CommentTok{\# 60 minutes intense
 ↪  exercise}
\NormalTok{))}
```

Pattern 3: Sensor Noise Test

```
\CommentTok{\# Test algorithm robustness to CGM noise}
\NormalTok{sim.add\_stress\_event(StressEvent(}
\NormalTok{    start\_time}\OperatorTok{=}\DecValTok{300}\NormalTok{,  }\CommentTok{\#
 ↪  5:00 AM}
\NormalTok{    event\_type}\OperatorTok{=}\StringTok{"sensor\_noise"}\NormalTok{,}
\NormalTok{    value}\OperatorTok{=}\FloatTok{20.0}  \CommentTok{\# 20 mg/dL standard
 ↪  deviation}
\NormalTok{))}
```

Pattern 4: Combined Stress Test

```
\CommentTok{\# Realistic day with multiple stressors}
\NormalTok{sim.add\_stress\_event(StressEvent(}\DecValTok{420}\NormalTok{,
 ↪  }\StringTok{"meal"}\NormalTok{, }\DecValTok{45}\NormalTok{))  }\CommentTok{\# 7:00 AM
 ↪  breakfast}
\NormalTok{sim.add\_stress\_event(StressEvent(}\DecValTok{660}\NormalTok{,
 ↪  }\StringTok{"meal"}\NormalTok{, }\DecValTok{60}\NormalTok{))  }\CommentTok{\# 11:00
 ↪  AM snack}
\NormalTok{sim.add\_stress\_event(StressEvent(}\DecValTok{900}\NormalTok{,
 ↪  }\StringTok{"exercise"}\NormalTok{, }\DecValTok{30}\NormalTok{))  }\CommentTok{\#
 ↪  3:00 PM walk}
\NormalTok{sim.add\_stress\_event(StressEvent(}\DecValTok{1020}\NormalTok{,
 ↪  }\StringTok{"meal"}\NormalTok{, }\DecValTok{80}\NormalTok{))  }\CommentTok{\# 5:00 PM
 ↪  dinner}
\NormalTok{sim.add\_stress\_event(StressEvent(}\DecValTok{1200}\NormalTok{,
 ↪  }\StringTok{"sensor\_noise"}\NormalTok{, }\FloatTok{15.0}\NormalTok{))
 ↪  }\CommentTok{\# 8:00 PM sensor issues}
```

1.9.4 7.4 Human-in-the-Loop Integration

Add manual interventions during simulation:

```
\KeywordTok{def}\NormalTok{ human\_in\_loop\_callback(context):}
    \CommentTok{"""}
\CommentTok{    Called at each simulation step.}
\CommentTok{    Return dict with manual interventions or None.}
\CommentTok{    """}
    \CommentTok{\# Example: Rescue carbs for hypoglycemia}
    \ControlFlowTok{if}\NormalTok{
     ↪  context[}\StringTok{"glucose\_actual\_mgdl"}\NormalTok{]
     ↪  }\OperatorTok{\textless{}} \DecValTok{65}\NormalTok{:}
        \ControlFlowTok{return}\NormalTok{ \{}
            \StringTok{"additional\_carbs"}\NormalTok{: }\DecValTok{15}\NormalTok{,
             ↪  }\CommentTok{\# 15g fast{-}acting carbs}
            \StringTok{"note"}\NormalTok{: }\StringTok{"Human intervention: rescue carbs
             ↪  for hypo"}
\NormalTok{        \}}

    \CommentTok{\# Example: Manual bolus correction}
    \ControlFlowTok{if}\NormalTok{
     ↪  context[}\StringTok{"glucose\_actual\_mgdl"}\NormalTok{]
     ↪  }\OperatorTok{\textgreater{}} \DecValTok{250}\NormalTok{:}
```

```
        \ControlFlowTok{return}\NormalTok{ \{}
            \StringTok{"additional\_bolus"}\NormalTok{: }\FloatTok{1.5}\NormalTok{,
             ↪  }\CommentTok{\# 1.5U correction}
            \StringTok{"note"}\NormalTok{: }\StringTok{"Human intervention: manual
             ↪  correction bolus"}
\NormalTok{        \}}

    \CommentTok{\# Example: Suspend insulin before exercise}
    \ControlFlowTok{if}\NormalTok{ context[}\StringTok{"time"}\NormalTok{]
     ↪  }\OperatorTok{\textgreater{}} \DecValTok{1080} \KeywordTok{and}\NormalTok{
     ↪  context[}\StringTok{"time"}\NormalTok{] }\OperatorTok{\textless{}}
     ↪  \DecValTok{1140}\NormalTok{:  }\CommentTok{\# 6{-}7 PM}
        \ControlFlowTok{return}\NormalTok{ \{}
            \StringTok{"suspend\_insulin"}\NormalTok{: }\VariableTok{True}\NormalTok{,}
            \StringTok{"note"}\NormalTok{: }\StringTok{"Human intervention: exercise
             ↪  suspension"}
\NormalTok{        \}}

    \ControlFlowTok{return} \VariableTok{None}  \CommentTok{\# No intervention}

\CommentTok{\# Use in simulator}
\NormalTok{sim }\OperatorTok{=}\NormalTok{ Simulator()
\NormalTok{    algorithm}\OperatorTok{=}\NormalTok{MyAlgorithm(),}
\NormalTok{    patient\_config}\OperatorTok{=}\StringTok{"default"}\NormalTok{,}
\NormalTok{    on\_step}\OperatorTok{=}\NormalTok{human\_in\_loop\_callback}
\NormalTok{)}
```

1.9.5 7.5 Data Import Workflows

Workflow 1: Dexcom CSV Import

```
\CommentTok{\# CLI import}
\ExtensionTok{iints}\NormalTok{ import{-}data }\DataTypeTok{\textbackslash{}}
  \AttributeTok{{-}{-}input{-}csv}\NormalTok{ dexcom\_export.csv
   ↪  }\DataTypeTok{\textbackslash{}}
  \AttributeTok{{-}{-}data{-}format}\NormalTok{ dexcom }\DataTypeTok{\textbackslash{}}
  \AttributeTok{{-}{-}output{-}dir}\NormalTok{ results/dexcom\_import}

\CommentTok{\# Use imported scenario}
\ExtensionTok{iints}\NormalTok{ run }\DataTypeTok{\textbackslash{}}
  \AttributeTok{{-}{-}algo}\NormalTok{ algorithms/my\_algorithm.py
   ↪  }\DataTypeTok{\textbackslash{}}
  \AttributeTok{{-}{-}scenario{-}path}\NormalTok{ results/dexcom\_import/scenario.json}
```

Workflow 2: Nightscout Import

```
\CommentTok{\# Install Nightscout extra}
\ExtensionTok{pip}\NormalTok{ install iints{-}sdk{-}python35}\PreprocessorTok{[}\Specia
 ↪  lStringTok{nightscout}\PreprocessorTok{]}

\CommentTok{\# Import from Nightscout}
\ExtensionTok{iints}\NormalTok{ import{-}nightscout }\DataTypeTok{\textbackslash{}}
```

```
  \AttributeTok{{-}{-}url}\NormalTok{ https://your{-}nightscout{-}site.herokuapp.com
   ↪  }\DataTypeTok{\textbackslash{}}
  \AttributeTok{{-}{-}days}\NormalTok{ 7 }\DataTypeTok{\textbackslash{}}
  \AttributeTok{{-}{-}output{-}dir}\NormalTok{ results/nightscout\_import}
```

Workflow 3: Dataset Registry

```
\CommentTok{\# List available datasets}
\ExtensionTok{iints}\NormalTok{ data list}

\CommentTok{\# Fetch specific dataset}
\ExtensionTok{iints}\NormalTok{ data fetch aide\_t1d
 ↪  }\AttributeTok{{-}{-}output{-}dir}\NormalTok{ data\_packs/aide}

\CommentTok{\# Use in simulation}
\ExtensionTok{iints}\NormalTok{ run }\DataTypeTok{\textbackslash{}}
  \AttributeTok{{-}{-}algo}\NormalTok{ algorithms/my\_algorithm.py
   ↪  }\DataTypeTok{\textbackslash{}}
  \AttributeTok{{-}{-}scenario{-}path}\NormalTok{
   ↪  data\_packs/aide/scenarios/patient\_001.json}
```

1.10 8. Advanced Topics

1.10.1 8.1 Commercial Pump Emulators

Test against real pump behavior:

```
\ImportTok{from}\NormalTok{ iints.emulation }\ImportTok{import}\NormalTok{
 ↪  Medtronic780G, Omnipod5, TandemControlIQ}

\CommentTok{\# Compare your algorithm against commercial pumps}
\NormalTok{pumps }\OperatorTok{=}\NormalTok{ [}
\NormalTok{    (}\StringTok{"MyAlgorithm"}\NormalTok{, MyAlgorithm()),}
\NormalTok{    (}\StringTok{"Medtronic 780G"}\NormalTok{, Medtronic780G()),}
\NormalTok{    (}\StringTok{"Omnipod 5"}\NormalTok{, Omnipod5()),}
\NormalTok{    (}\StringTok{"Tandem Control{-}IQ"}\NormalTok{, TandemControlIQ())}
\NormalTok{]}

\NormalTok{results }\OperatorTok{=}\NormalTok{ \{\}}
\ControlFlowTok{for}\NormalTok{ name, pump\_algo }\KeywordTok{in}\NormalTok{ pumps:}
\NormalTok{    results[name] }\OperatorTok{=}\NormalTok{ run\_simulation(}
\NormalTok{        algorithm}\OperatorTok{=}\NormalTok{pump\_algo,}
\NormalTok{        patient\_config}\OperatorTok{=}\StringTok{"default"}\NormalTok{,}
\NormalTok{        duration\_minutes}\OperatorTok{=}\DecValTok{1440}
\NormalTok{    )}

\CommentTok{\# Compare metrics}
\NormalTok{compare\_metrics(results)}
```

1.10.2 8.2 Dataset Registry Usage

Access real-world datasets:

```
\CommentTok{\# List all available datasets}
\ExtensionTok{iints}\NormalTok{ data list}

\CommentTok{\# Get info about specific dataset}
\ExtensionTok{iints}\NormalTok{ data info aide\_t1d}

\CommentTok{\# Fetch dataset}
\ExtensionTok{iints}\NormalTok{ data fetch aide\_t1d
 ↪  }\AttributeTok{{-}{-}output{-}dir}\NormalTok{ data\_packs/aide}

\CommentTok{\# Cite dataset in publication}
\ExtensionTok{iints}\NormalTok{ data cite aide\_t1d}
```

Available Datasets: - `aide_t1d`: AIDE Type 1 Diabetes Dataset - `azt1d`: Arizona Type 1 Diabetes Dataset - `ohio_t1dm`: OhioT1DM Dataset - `sample`: Bundled demo data (no download needed)

Integrity and reproducibility - Every dataset entry includes a SHA-256 checksum and citation metadata. - The fetch command validates the checksum automatically. - Use `iints data info <dataset>` to record version and hash in your paper.

Typical layout after fetch

```
data_packs/
  public/
    ohio_t1dm/
      raw/...
      processed/...
```

1.10.3 8.3 Reproducibility Techniques

Ensure identical results across runs:

```
\CommentTok{\# Method 1: Set random seed}
\NormalTok{results }\OperatorTok{=}\NormalTok{ run\_simulation()}
\NormalTok{    algorithm}\OperatorTok{=}\NormalTok{MyAlgorithm(),}
\NormalTok{    seed}\OperatorTok{=}\DecValTok{42}\NormalTok{,  }\CommentTok{\# Fixed
 ↪  random seed}
\NormalTok{    patient\_config}\OperatorTok{=}\StringTok{"default"}
\NormalTok{)}

\CommentTok{\# Method 2: Use deterministic patient profile}
\NormalTok{profile }\OperatorTok{=}\NormalTok{ PatientProfile()}
\NormalTok{    isf}\OperatorTok{=}\DecValTok{50}\NormalTok{,}
\NormalTok{    icr}\OperatorTok{=}\DecValTok{10}\NormalTok{,}
\NormalTok{    basal\_rate}\OperatorTok{=}\FloatTok{0.9}\NormalTok{,}
\NormalTok{    dawn\_phenomenon}\OperatorTok{=}\FloatTok{0.3}\NormalTok{,
 ↪  }\CommentTok{\# Fixed dawn effect}
\NormalTok{    seed}\OperatorTok{=}\DecValTok{42}
\NormalTok{)}

\CommentTok{\# Method 3: SHA{-}256 verification}
```

```
\ImportTok{from}\NormalTok{ iints.utils.hashing }\ImportTok{import}\NormalTok{
 ↪  verify\_manifest}

\ControlFlowTok{if}\NormalTok{
 ↪  verify\_manifest(}\StringTok{"results/run\_001/run\_manifest.json"}\NormalTok{):}
    \BuiltInTok{print}\NormalTok{(}\StringTok{"Results verified {-} not tampered
     ↪  with"}\NormalTok{)}
```

1.10.4 8.4 Performance Profiling

Measure algorithm performance:

```
\NormalTok{sim }\OperatorTok{=}\NormalTok{ Simulator()
\NormalTok{    algorithm}\OperatorTok{=}\NormalTok{MyAlgorithm(),}
\NormalTok{    patient\_config}\OperatorTok{=}\StringTok{"default"}\NormalTok{,}
\NormalTok{    enable\_profiling}\OperatorTok{=}\VariableTok{True}
\NormalTok{)}

\NormalTok{results\_df, safety\_report }\OperatorTok{=}\NormalTok{
 ↪  sim.run\_batch(duration\_minutes}\OperatorTok{=}\DecValTok{1440}\NormalTok{)}

\CommentTok{\# Access performance data}
\NormalTok{profiling }\OperatorTok{=}\NormalTok{
 ↪  safety\_report[}\StringTok{"performance\_report"}\NormalTok{]}
\BuiltInTok{print}\NormalTok{(}\SpecialStringTok{f"Algorithm latency: }\SpecialCharTok{
 ↪  \{}\NormalTok{profiling[}\StringTok{\textquotesingle{}algorithm\_latency\_ms\textqu
 ↪  otesingle{}}\NormalTok{]}\SpecialCharTok{:.2f\}}\SpecialStringTok{ms"}\NormalTok{)}
\BuiltInTok{print}\NormalTok{(}\SpecialStringTok{f"Safety latency: }\SpecialCharTok{\{
 ↪  \NormalTok{profiling[}\StringTok{\textquotesingle{}safety\_latency\_ms\textquotesin
 ↪  gle{}}\NormalTok{]}\SpecialCharTok{:.2f\}}\SpecialStringTok{ms"}\NormalTok{)}
\BuiltInTok{print}\NormalTok{(}\SpecialStringTok{f"Total steps:
 ↪  }\SpecialCharTok{\{}\NormalTok{profiling[}\StringTok{\textquotesingle{}total\_steps
 ↪  \textquotesingle{}}\NormalTok{]}\SpecialCharTok{\}}\SpecialStringTok{"}\NormalTok{)}
\BuiltInTok{print}\NormalTok{(}\SpecialStringTok{f"Total time: }\SpecialCharTok{\{}\Nor
 ↪  malTok{profiling[}\StringTok{\textquotesingle{}total\_time\_s\textquotesingle{}}\No
 ↪  rmalTok{]}\SpecialCharTok{:.2f\}}\SpecialStringTok{s"}\NormalTok{)}
```

1.10.5 8.5 Custom Metrics Calculation

Define your own performance metrics:

```
\KeywordTok{def}\NormalTok{ calculate\_custom\_metric(results\_df):}
    \CommentTok{"""Calculate custom performance metric"""}

    \CommentTok{\# Time in tight range (80{-}140 mg/dL)}
\NormalTok{    tight\_range }\OperatorTok{=}\NormalTok{ ((results\_df[}\StringTok{\textq
 ↪  uotesingle{}glucose\_actual\_mgdl\textquotesingle{}}\NormalTok{]
 ↪  }\OperatorTok{\textgreater{}=} \DecValTok{80}\NormalTok{) }\OperatorTok{\&}
\NormalTok{                  (results\_df[}\StringTok{\textquotesingle{}glucose\_actual\
 ↪  _mgdl\textquotesingle{}}\NormalTok{] }\OperatorTok{\textless{}=}
 ↪  \DecValTok{140}\NormalTok{)).mean() }\OperatorTok{*} \DecValTok{100}
```

```
    \CommentTok{\# Glucose variability score}
\NormalTok{    cv }\OperatorTok{=}\NormalTok{ results\_df[}\StringTok{\textquotesingle{}
 ↪  glucose\_actual\_mgdl\textquotesingle{}}\NormalTok{].std()
 ↪  }\OperatorTok{/}
\NormalTok{         results\_df[}\StringTok{\textquotesingle{}glucose\_actual\_mgdl\text
 ↪  quotesingle{}}\NormalTok{].mean() }\OperatorTok{*}
 ↪  \DecValTok{100}

    \CommentTok{\# Hypoglycemia risk score}
\NormalTok{    hypo\_risk }\OperatorTok{=}\NormalTok{ (results\_df[}\StringTok{\textquot
 ↪  esingle{}glucose\_actual\_mgdl\textquotesingle{}}\NormalTok{]
 ↪  }\OperatorTok{\textless{}} \DecValTok{70}\NormalTok{).}\BuiltInTok{sum}\NormalTok{()
 ↪  }\OperatorTok{/} \BuiltInTok{len}\NormalTok{(results\_df)}

    \CommentTok{\# Hyperglycemia risk score}
\NormalTok{    hyper\_risk }\OperatorTok{=}\NormalTok{ (results\_df[}\StringTok{\textquo
 ↪  tesingle{}glucose\_actual\_mgdl\textquotesingle{}}\NormalTok{]
 ↪  }\OperatorTok{\textgreater{}}
 ↪  \DecValTok{250}\NormalTok{).}\BuiltInTok{sum}\NormalTok{() }\OperatorTok{/}
 ↪  \BuiltInTok{len}\NormalTok{(results\_df)}

    \CommentTok{\# Composite score (lower is better)}
\NormalTok{    composite\_score }\OperatorTok{=}\NormalTok{ (}\DecValTok{100}
 ↪  \OperatorTok{-}\NormalTok{ tight\_range) }\OperatorTok{+}\NormalTok{ cv
 ↪  }\OperatorTok{+}\NormalTok{ (hypo\_risk }\OperatorTok{*} \DecValTok{100}\NormalTok{)
 ↪  }\OperatorTok{+}\NormalTok{ (hyper\_risk }\OperatorTok{*} \DecValTok{50}\NormalTok{)}

    \ControlFlowTok{return}\NormalTok{ \{}
        \StringTok{\textquotesingle{}tight\_range\_percent\textquotesingle{}}\NormalTok{:
         ↪  tight\_range,}
        \StringTok{\textquotesingle{}cv\_percent\textquotesingle{}}\NormalTok{: cv,}
        \StringTok{\textquotesingle{}hypo\_risk\_percent\textquotesingle{}}\NormalTok{:
         ↪  hypo\_risk }\OperatorTok{*} \DecValTok{100}\NormalTok{,}
        \StringTok{\textquotesingle{}hyper\_risk\_percent\textquotesingle{}}\NormalTok{:
         ↪  hyper\_risk }\OperatorTok{*} \DecValTok{100}\NormalTok{,}
        \StringTok{\textquotesingle{}composite\_score\textquotesingle{}}\NormalTok{:
         ↪  composite\_score}
\NormalTok{    \}}

\CommentTok{\# Use with your results}
\NormalTok{metrics }\OperatorTok{=}\NormalTok{ calculate\_custom\_metric(results\_df)}
\BuiltInTok{print}\NormalTok{(}\SpecialStringTok{f"Custom Score: }\SpecialCharTok{\{}\N
 ↪  ormalTok{metrics[}\StringTok{\textquotesingle{}composite\_score\textquotesingle{}}\
 ↪  NormalTok{]}\SpecialCharTok{:.1f\}}\SpecialStringTok{"}\NormalTok{)}
```

1.11 9. Troubleshooting

1.11.1 9.1 Installation Issues

Issue: ModuleNotFoundError after installation

Solution:

```
\CommentTok{\# Make sure you\textquotesingle{}re using the correct Python environment}
\FunctionTok{which}\NormalTok{ python  }\CommentTok{\# Should show your virtual
 ↪  environment path}

\CommentTok{\# Reinstall in development mode if needed}
\ExtensionTok{pip}\NormalTok{ install }\AttributeTok{{-}e}\NormalTok{ .}
```

Issue: pip install fails with dependency errors

Solution:

```
\CommentTok{\# Upgrade pip first}
\ExtensionTok{pip}\NormalTok{ install }\AttributeTok{{-}{-}upgrade}\NormalTok{ pip
 ↪  setuptools wheel}

\CommentTok{\# Try installing with {-}{-}no{-}cache{-}dir}
\ExtensionTok{pip}\NormalTok{ install }\AttributeTok{{-}{-}no{-}cache{-}dir}\NormalTok{
 ↪  iints{-}sdk{-}python35}

\CommentTok{\# For specific errors, check Python version}
\ExtensionTok{python3} \AttributeTok{{-}{-}version}  \CommentTok{\# Must be 3.10+}
```

1.11.2 9.2 Simulation Issues

Issue: Simulation runs very slowly

Solution:

```
\CommentTok{\# Increase time step (default is 5 minutes)}
\NormalTok{sim }\OperatorTok{=}\NormalTok{
 ↪  Simulator(time\_step}\OperatorTok{=}\DecValTok{10}\NormalTok{)  }\CommentTok{\#
 ↪  10{-}minute steps}

\CommentTok{\# Disable profiling if not needed}
\NormalTok{sim }\OperatorTok{=}\NormalTok{
 ↪  Simulator(enable\_profiling}\OperatorTok{=}\VariableTok{False}\NormalTok{)}

\CommentTok{\# Reduce duration for testing}
\NormalTok{results }\OperatorTok{=}\NormalTok{
 ↪  sim.run\_batch(duration\_minutes}\OperatorTok{=}\DecValTok{720}\NormalTok{)
 ↪  }\CommentTok{\# 12 hours instead of 24}
```

Issue: Simulation terminates early

Solution:

```
\CommentTok{\# Check safety report for termination reason}
\BuiltInTok{print}\NormalTok{(safety\_report[}\StringTok{\textquotesingle{}termination\
 ↪  _reason\textquotesingle{}}\NormalTok{])}

\CommentTok{\# Common causes:}
\CommentTok{\# {-} Critical hypoglycemia (\textless{} 40 mg/dL for 30+ minutes)}
\CommentTok{\# {-} Algorithm exception}
```

```
\CommentTok{\# {-} Configuration error}

\CommentTok{\# Adjust safety limits if needed}
\ImportTok{from}\NormalTok{ iints.core.safety }\ImportTok{import}\NormalTok{
 ↪  SafetyConfig}
\NormalTok{config }\OperatorTok{=}\NormalTok{
 ↪  SafetyConfig(critical\_hypo\_cutoff}\OperatorTok{=}\FloatTok{35.0}\NormalTok{)
 ↪  }\CommentTok{\# Lower threshold}
```

1.11.3 9.3 Algorithm Development Issues

Issue: Algorithm not appearing in CLI

Solution:

```
\CommentTok{\# Make sure algorithm is properly registered}
\CommentTok{\# 1. Inherits from InsulinAlgorithm}
\CommentTok{\# 2. Implements all required methods}
\CommentTok{\# 3. Has valid get\_algorithm\_metadata()}

\CommentTok{\# Check with:}
\ImportTok{from}\NormalTok{ iints.api.registry }\ImportTok{import}\NormalTok{
 ↪  list\_algorithms}
\BuiltInTok{print}\NormalTok{(list\_algorithms())}
```

Issue: Safety supervisor blocking all doses

Solution:

```
\CommentTok{\# Check safety configuration}
\BuiltInTok{print}\NormalTok{(safety\_report[}\StringTok{\textquotesingle{}config\textq
 ↪  uotesingle{}}\NormalTok{])}

\CommentTok{\# Common issues:}
\CommentTok{\# {-} max\_basal\_rate too low}
\CommentTok{\# {-} hypo\_cutoff too high}
\CommentTok{\# {-} contract enabled with aggressive settings}

\CommentTok{\# Adjust configuration:}
\NormalTok{config }\OperatorTok{=}\NormalTok{ SafetyConfig(}
\NormalTok{    hypo\_cutoff}\OperatorTok{=}\FloatTok{60.0}\NormalTok{,  }\CommentTok{\#
 ↪  Higher threshold}
\NormalTok{    max\_basal\_rate}\OperatorTok{=}\FloatTok{1.5}  \CommentTok{\# Higher
 ↪  limit}
\NormalTok{)}
```

1.11.4 9.4 Data Import Issues

Issue: CSV import fails

Solution:

```
\CommentTok{\# Check CSV format}
\CommentTok{\# Required columns: timestamp, glucose}
```

```
\CommentTok{\# Optional: carbs, insulin}

\CommentTok{\# Try auto{-}detect format}
\NormalTok{data }\OperatorTok{=}\NormalTok{
 ↪  import\_cgm\_csv(}\StringTok{"your\_file.csv"}\NormalTok{,
 ↪  data\_format}\OperatorTok{=}\StringTok{"auto"}\NormalTok{)}

\CommentTok{\# Specify column names manually if needed}
\NormalTok{data }\OperatorTok{=}\NormalTok{ import\_cgm\_csv(}
    \StringTok{"your\_file.csv"}\NormalTok{,}
\NormalTok{    data\_format}\OperatorTok{=}\StringTok{"generic"}\NormalTok{,}
\NormalTok{    timestamp\_col}\OperatorTok{=}\StringTok{"Date"}\NormalTok{,}
\NormalTok{    glucose\_col}\OperatorTok{=}\StringTok{"BG"}\NormalTok{,}
\NormalTok{    carbs\_col}\OperatorTok{=}\StringTok{"Carbs"}
\NormalTok{)}
```

1.11.5 9.5 Performance Issues

Issue: High memory usage

Solution:

```
\CommentTok{\# Reduce history size}
\NormalTok{sim }\OperatorTok{=}\NormalTok{
 ↪  Simulator(max\_history\_hours}\OperatorTok{=}\DecValTok{12}\NormalTok{)
 ↪  }\CommentTok{\# Default is 24}

\CommentTok{\# Disable audit trail if not needed}
\NormalTok{sim }\OperatorTok{=}\NormalTok{
 ↪  Simulator(enable\_audit}\OperatorTok{=}\VariableTok{False}\NormalTok{)}

\CommentTok{\# Process in batches}
\ControlFlowTok{for}\NormalTok{ i }\KeywordTok{in}
 ↪  \BuiltInTok{range}\NormalTok{(}\DecValTok{10}\NormalTok{):}
\NormalTok{    results }\OperatorTok{=}\NormalTok{
 ↪  sim.run\_batch(duration\_minutes}\OperatorTok{=}\DecValTok{144}\NormalTok{)
 ↪  }\CommentTok{\# 2.4 hours per batch}
\NormalTok{    process\_results(results)}
```

1.12 10. Quick Reference

1.12.1 10.1 Essential CLI Commands

```
\CommentTok{\# Initialize project}
\ExtensionTok{iints}\NormalTok{ init }\AttributeTok{{-}{-}project{-}name}\NormalTok{
 ↪  my\_project}

\CommentTok{\# Quickstart}
\ExtensionTok{iints}\NormalTok{ quickstart
 ↪  }\AttributeTok{{-}{-}project{-}name}\NormalTok{ demo}
```

```
\CommentTok{\# Run simulation}
\ExtensionTok{iints}\NormalTok{ run }\AttributeTok{{-}{-}algo}\NormalTok{
 ↪  algorithms/my\_algo.py }\AttributeTok{{-}{-}duration}\NormalTok{ 1440}

\CommentTok{\# Run with preset}
\ExtensionTok{iints}\NormalTok{ presets run }\AttributeTok{{-}{-}name}\NormalTok{
 ↪  baseline\_t1d }\AttributeTok{{-}{-}algo}\NormalTok{ algorithms/my\_algo.py}

\CommentTok{\# Benchmark}
\ExtensionTok{iints}\NormalTok{ benchmark
 ↪  }\AttributeTok{{-}{-}algo{-}to{-}benchmark}\NormalTok{ algorithms/my\_algo.py}

\CommentTok{\# Import data}
\ExtensionTok{iints}\NormalTok{ import{-}data
 ↪  }\AttributeTok{{-}{-}input{-}csv}\NormalTok{ my\_cgm.csv
 ↪  }\AttributeTok{{-}{-}data{-}format}\NormalTok{ dexcom}

\CommentTok{\# List presets}
\ExtensionTok{iints}\NormalTok{ presets list}

\CommentTok{\# Generate scenario}
\ExtensionTok{iints}\NormalTok{ scenarios generate }\AttributeTok{{-}{-}name}
 ↪  \StringTok{"Stress Test"}

\CommentTok{\# Validate files}
\ExtensionTok{iints}\NormalTok{ validate
 ↪  }\AttributeTok{{-}{-}scenario{-}path}\NormalTok{ scenarios/my\_scenario.json}
```

1.12.2 10.2 Python Code Snippets

Minimal Simulation:

```
\ImportTok{from}\NormalTok{ iints }\ImportTok{import}\NormalTok{ run\_simulation}
\ImportTok{from}\NormalTok{ iints.core.algorithms.pid\_controller
 ↪  }\ImportTok{import}\NormalTok{ PIDController}

\NormalTok{results }\OperatorTok{=}\NormalTok{ run\_simulation()
\NormalTok{    algorithm}\OperatorTok{=}\NormalTok{PIDController(),}
\NormalTok{    duration\_minutes}\OperatorTok{=}\DecValTok{720}
\NormalTok{)}
```

Custom Algorithm:

```
\ImportTok{from}\NormalTok{ iints.api.base\_algorithm }\ImportTok{import}\NormalTok{
 ↪  InsulinAlgorithm}

\KeywordTok{class}\NormalTok{ MyAlgorithm(InsulinAlgorithm):}
    \KeywordTok{def}\NormalTok{ predict\_insulin(}\VariableTok{self}\NormalTok{, data):}
        \ControlFlowTok{return}\NormalTok{ \{}\StringTok{"basal\_insulin"}\NormalTok{:
         ↪  }\FloatTok{0.9}\NormalTok{, }\StringTok{"bolus\_insulin"}\NormalTok{:
         ↪  }\FloatTok{0.0}\NormalTok{\}}
```

Load Results:

```
\ImportTok{import}\NormalTok{ pandas }\ImportTok{as}\NormalTok{ pd}
\NormalTok{df }\OperatorTok{=}\NormalTok{ pd.read\_csv(results[}\StringTok{\textquotesi
↪  ngle{}results\_csv\textquotesingle{}}\NormalTok{])}
\BuiltInTok{print}\NormalTok{(df[[}\StringTok{\textquotesingle{}timestamp\textquotesing
↪  le{}}\NormalTok{,
↪  }\StringTok{\textquotesingle{}glucose\_actual\_mgdl\textquotesingle{}}\NormalTok{,
↪  }\StringTok{\textquotesingle{}insulin\_delivered\textquotesingle{}}\NormalTok{]].he
↪  ad())}
```

1.12.3 10.3 Safety Thresholds (Defaults)

```
\NormalTok{SafetyConfig(}
\NormalTok{    hypo\_cutoff}\OperatorTok{=}\FloatTok{70.0}\NormalTok{,
↪  }\CommentTok{\# Start reducing insulin}
\NormalTok{    severe\_hypo\_cutoff}\OperatorTok{=}\FloatTok{54.0}\NormalTok{,
↪  }\CommentTok{\# Emergency suspension}
\NormalTok{    critical\_hypo\_cutoff}\OperatorTok{=}\FloatTok{40.0}\NormalTok{,
↪  }\CommentTok{\# Terminate simulation}
\NormalTok{    hyper\_cutoff}\OperatorTok{=}\FloatTok{300.0}\NormalTok{,
↪  }\CommentTok{\# Maximum glucose}
\NormalTok{    max\_basal\_rate}\OperatorTok{=}\FloatTok{2.0}\NormalTok{,
↪  }\CommentTok{\# U/hr}
\NormalTok{    max\_bolus}\OperatorTok{=}\FloatTok{5.0}\NormalTok{,
↪  }\CommentTok{\# U per bolus}
\NormalTok{    max\_iob}\OperatorTok{=}\FloatTok{10.0}\NormalTok{,
↪  }\CommentTok{\# U total active insulin}
\NormalTok{    max\_insulin\_per\_hour}\OperatorTok{=}\FloatTok{15.0}\NormalTok{,
↪  }\CommentTok{\# U/hr rolling window}
\NormalTok{    max\_insulin\_per\_day}\OperatorTok{=}\FloatTok{80.0}   \CommentTok{\#
↪  U/day absolute limit}
\NormalTok{)}
```

1.12.4 10.4 Clinical Metrics Targets (ATTD/ADA)

Metric	Target Range
TIR (70-180 mg/dL)	>70%
TBR (<70 mg/dL)	<4%
TBR (<54 mg/dL)	<1%
GMI	<7.0%
CV	<36%
LBGI	<1.1
HBGI	<5.0

1.13 11. Glossary

Algorithm: Code that calculates insulin doses based on current and historical data

Basal Insulin: Background insulin delivery rate (U/hr)

Bolus Insulin: Additional insulin for meals or corrections (U)

CGM: Continuous Glucose Monitor - device that measures glucose every 5 minutes

COB: Carbs On Board - carbohydrates still being absorbed

GMI: Glucose Management Indicator - estimate of A1C from CGM data

IOB: Insulin On Board - insulin still active in the body

ISF: Insulin Sensitivity Factor - how much 1U of insulin lowers glucose (mg/dL)

ICR: Insulin-to-Carb Ratio - grams of carbs covered by 1U of insulin

TIR: Time In Range - percentage of time in target glucose range (70-180 mg/dL)

TBRL1: Time Below Range Level 1 - percentage of time <70 mg/dL

TBRL2: Time Below Range Level 2 - percentage of time <54 mg/dL

TAR: Time Above Range - percentage of time >180 mg/dL

CV: Coefficient of Variation - measure of glucose variability

LBGI: Low Blood Glucose Index - measure of hypoglycemia risk

HBGI: High Blood Glucose Index - measure of hyperglycemia risk

---

1.14 Need More Help?

Documentation: - Comprehensive Guide: `docs/COMPREHENSIVE_GUIDE.md` - Technical README: `docs/TECHNICAL_README.md` - API Stability: `API_STABILITY.md`

Community: - GitHub Issues: https://github.com/python35/IINTS-SDK/issues - Discussion Forum: [Link] - Email Support: support@iints.org

Citing IINTS-AF:

```
\CommentTok{@software\{IINTS\_AF\_SDK,}
\CommentTok{  author = \{Bobbaers, Rune\},}
\CommentTok{  title = \{IINTS{-}AF: Intelligent Insulin Titration System for Artificial
 ↪  Pancreas\},}
\CommentTok{  year = \{2026\},}
\CommentTok{  version = \{0.1.19\},}
\CommentTok{  url = \{https://github.com/python35/IINTS{-}SDK\},}
\CommentTok{  note = \{Pre{-}clinical research software for insulin dosing algorithm
 ↪  validation\}}
\CommentTok{\}}
```

PRE-CLINICAL USE ONLY - NOT FOR PATIENT CARE

This document is provided for research and educational purposes only. The IINTS-AF SDK is not a medical device and should not be used for clinical decision-making.