GPU Accelerated Nested Sampling for 21-cm Cosmology

Sam Leeney
(on behalf of Will Handley)
<sakl2@cam.ac.uk>

Royal Society University Research Fellow
Institute of Astronomy, University of Cambridge
Kavli Institute for Cosmology, Cambridge
Gonville & Caius College
willhandley.co.uk/talks

21st January 2025


UNIVERSITY OF CAMBRIDGE


Bayesian Inference Challenges in 21cm Cosmology

Computational Challenges

- High-dimensional parameter spaces (20-100+ parameters)
- Complex, multimodal posteriors
- Expensive likelihood evaluations
- Months of computation time
- Model comparison bottlenecks

Physical Challenges

- Foreground contamination $10^5\times$ stronger than signal
- Instrumental systematics
- Ionospheric effects
- Degeneracies between astrophysics and cosmology
- Need for robust uncertainty quantification

These challenges demand new computational approaches

GPU Computing: Beyond Machine Learning

GPU vs CPU for Scientific Computing

- **CPU**: Few powerful cores (10s), complex control.
- **GPU**: Many simple cores (1000s), simple control.
- **Memory bandwidth**: GPU $10\times$ faster than CPU.
- **Perfect for**: Independent parallel tasks.
- **Scientific algorithms**: MCMC chains, likelihood evaluations, simulations.

HPC Landscape Evolution

- HPC transitioning to GPU-based architectures.
- ML adoption accelerating hardware development.
- Legacy CPU codes require modernization.

Key Point

GPU $\neq$ Machine Learning

GPUs accelerate any parallel algorithm

Modern Languages: Two Independent Capabilities

Differentiable programming languages: JAX, PyTorch, TensorFlow, Julia, Stan, ...

Capability 1: Free Gradients

- **Automatic differentiation**: $\nabla_\theta \log \mathcal{L}(\theta)$.
- Enables gradient-based MCMC (HMC, NUTS).
- Essential for modern optimization.

Traditional Physics Benefits

- **Nested sampling**: Massive parallelization.
- **21cm signals**: Vectorized across frequency/time/angle.
- **N-body sims**: GPU acceleration.

Capability 2: GPU Parallelization

- **Vectorization across ensembles**.
- Run 1000s of parallel chains/particles.
- Evaluate likelihoods simultaneously.

Key Insight: Often Confused

These are completely independent.
People mistake one for the other.
You can use gradients on CPU.
You can GPU parallelize without gradients.
They serve different purposes.

BlackJAX: GPU Native Sampling

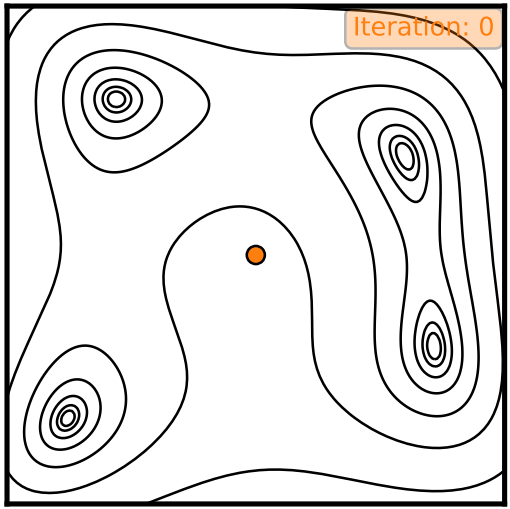
Gradient descent: inference at speed

David Yallup
Postdoc

- Sampling traditionally CPU-bound.
- Different algorithms, same GPU challenge.
- Need unified GPU-native framework.
- From optimization to model comparison.

- BlackJAX: Full JAX ecosystem.
- All algorithms GPU-accelerated.
- Gradient descent through nested sampling.
- Unified interface, maximum performance.

- Framework: more like `numpy` or `scipy` than `cobaya` or `cosmosis`.



BlackJAX: GPU Native Sampling

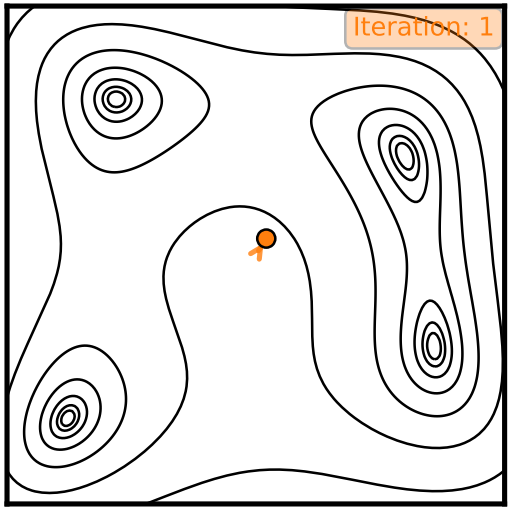
Gradient descent: inference at speed

David Yallup
Postdoc

- Sampling traditionally CPU-bound.
- Different algorithms, same GPU challenge.
- Need unified GPU-native framework.
- From optimization to model comparison.

- BlackJAX: Full JAX ecosystem.
- All algorithms GPU-accelerated.
- Gradient descent through nested sampling.
- Unified interface, maximum performance.

- Framework: more like `numpy` or `scipy` than `cobaya` or `cosmosis`.



BlackJAX: GPU Native Sampling

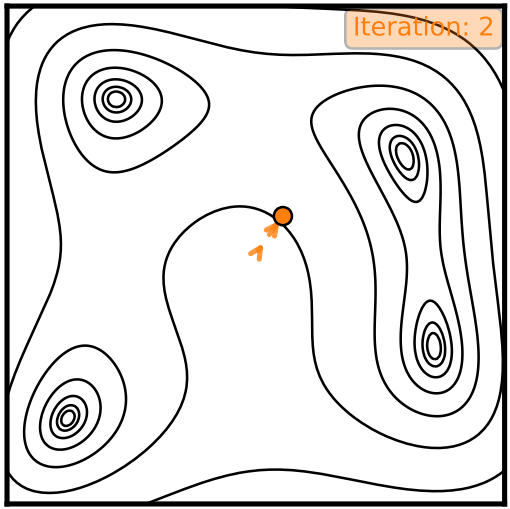
Gradient descent: inference at speed

David Yallup
Postdoc

- Sampling traditionally CPU-bound.
- Different algorithms, same GPU challenge.
- Need unified GPU-native framework.
- From optimization to model comparison.

- BlackJAX: Full JAX ecosystem.
- All algorithms GPU-accelerated.
- Gradient descent through nested sampling.
- Unified interface, maximum performance.

- Framework: more like `numpy` or `scipy` than `cobaya` or `cosmosis`.



BlackJAX: GPU Native Sampling

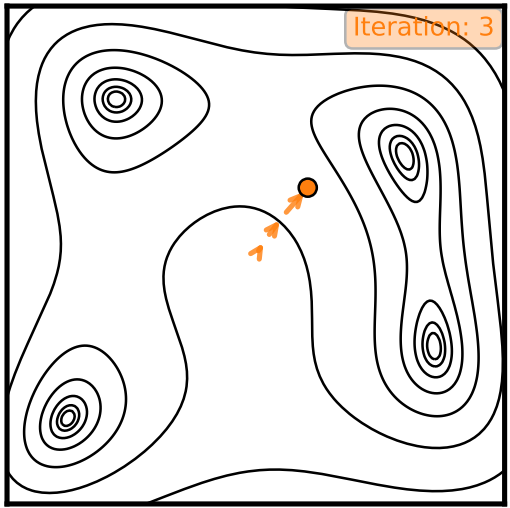
Gradient descent: inference at speed

David Yallup
Postdoc

- Sampling traditionally CPU-bound.
- Different algorithms, same GPU challenge.
- Need unified GPU-native framework.
- From optimization to model comparison.

- BlackJAX: Full JAX ecosystem.
- All algorithms GPU-accelerated.
- Gradient descent through nested sampling.
- Unified interface, maximum performance.

- Framework: more like `numpy` or `scipy` than `cobaya` or `cosmosis`.



BlackJAX: GPU Native Sampling

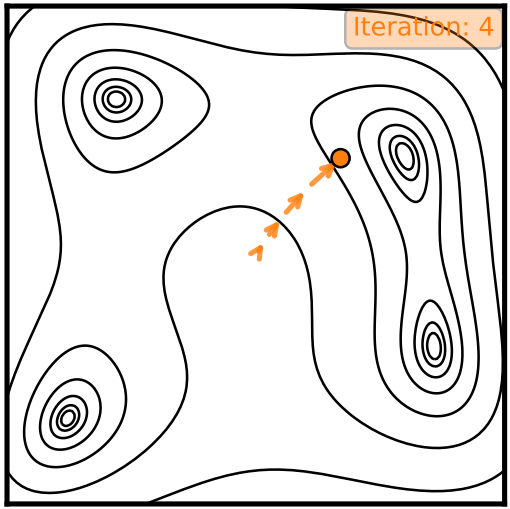
Gradient descent: inference at speed

David Yallup
Postdoc

- Sampling traditionally CPU-bound.
- Different algorithms, same GPU challenge.
- Need unified GPU-native framework.
- From optimization to model comparison.

- BlackJAX: Full JAX ecosystem.
- All algorithms GPU-accelerated.
- Gradient descent through nested sampling.
- Unified interface, maximum performance.

- Framework: more like `numpy` or `scipy` than `cobaya` or `cosmosis`.



BlackJAX: GPU Native Sampling

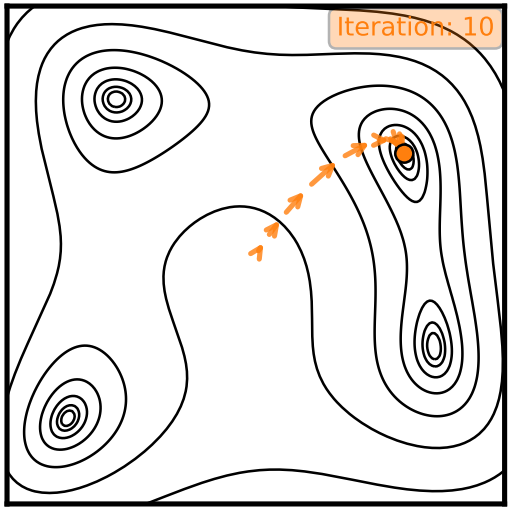
Gradient descent: inference at speed

David Yallup
Postdoc

- Sampling traditionally CPU-bound.
- Different algorithms, same GPU challenge.
- Need unified GPU-native framework.
- From optimization to model comparison.

- BlackJAX: Full JAX ecosystem.
- All algorithms GPU-accelerated.
- Gradient descent through nested sampling.
- Unified interface, maximum performance.

- Framework: more like `numpy` or `scipy` than `cobaya` or `cosmosis`.



BlackJAX: GPU Native Sampling

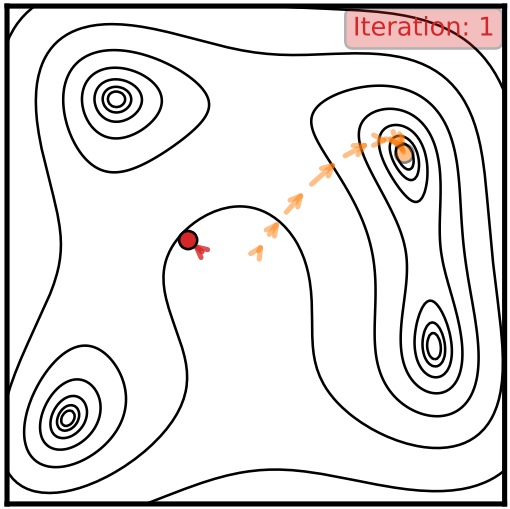
Gradient descent: inference at speed

David Yallup
Postdoc

- Sampling traditionally CPU-bound.
- Different algorithms, same GPU challenge.
- Need unified GPU-native framework.
- From optimization to model comparison.

- BlackJAX: Full JAX ecosystem.
- All algorithms GPU-accelerated.
- Gradient descent through nested sampling.
- Unified interface, maximum performance.

- Framework: more like `numpy` or `scipy` than `cobaya` or `cosmosis`.



BlackJAX: GPU Native Sampling

Gradient descent: inference at speed

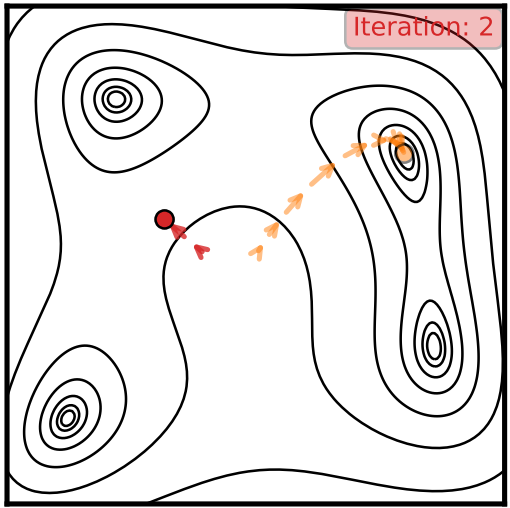
David Yallup

Postdoc

- Sampling traditionally CPU-bound.
- Different algorithms, same GPU challenge.
- Need unified GPU-native framework.
- From optimization to model comparison.

- BlackJAX: Full JAX ecosystem.
- All algorithms GPU-accelerated.
- Gradient descent through nested sampling.
- Unified interface, maximum performance.

- Framework: more like `numpy` or `scipy` than `cobaya` or `cosmosis`.



BlackJAX: GPU Native Sampling

Gradient descent: inference at speed

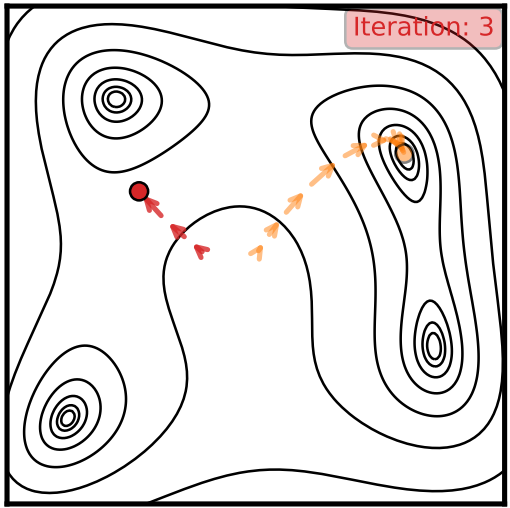
David Yallup

Postdoc

- Sampling traditionally CPU-bound.
- Different algorithms, same GPU challenge.
- Need unified GPU-native framework.
- From optimization to model comparison.

- BlackJAX: Full JAX ecosystem.
- All algorithms GPU-accelerated.
- Gradient descent through nested sampling.
- Unified interface, maximum performance.

- Framework: more like `numpy` or `scipy` than `cobaya` or `cosmosis`.



BlackJAX: GPU Native Sampling

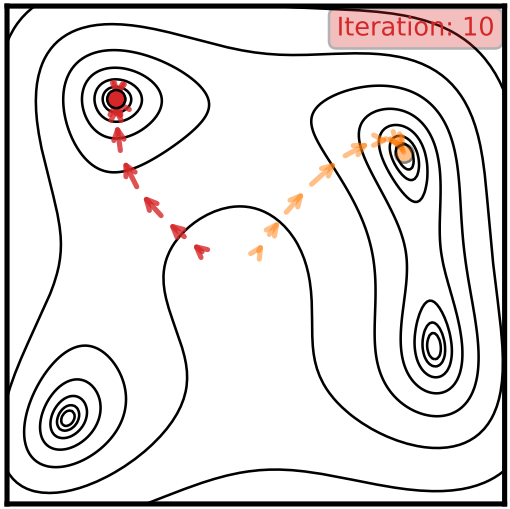
Gradient descent: inference at speed

David Yallup
Postdoc

- Sampling traditionally CPU-bound.
- Different algorithms, same GPU challenge.
- Need unified GPU-native framework.
- From optimization to model comparison.

- BlackJAX: Full JAX ecosystem.
- All algorithms GPU-accelerated.
- Gradient descent through nested sampling.
- Unified interface, maximum performance.

- Framework: more like `numpy` or `scipy` than `cobaya` or `cosmosis`.



BlackJAX: GPU Native Sampling

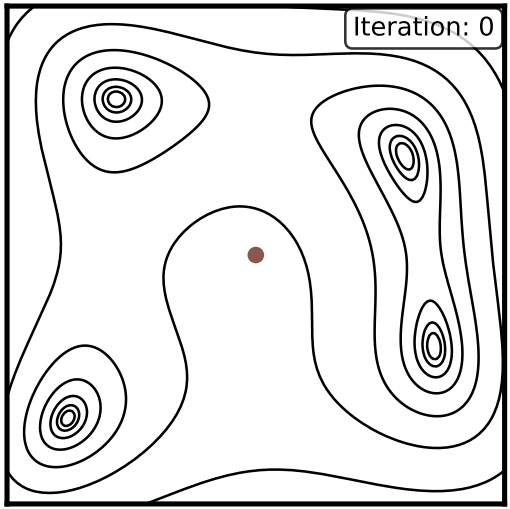
Metropolis-Hastings: error bars

David Yallup
Postdoc

- Sampling traditionally CPU-bound.
- Different algorithms, same GPU challenge.
- Need unified GPU-native framework.
- From optimization to model comparison.

- BlackJAX: Full JAX ecosystem.
- All algorithms GPU-accelerated.
- Gradient descent through nested sampling.
- Unified interface, maximum performance.

- Framework: more like `numpy` or `scipy` than `cobaya` or `cosmosis`.



BlackJAX: GPU Native Sampling

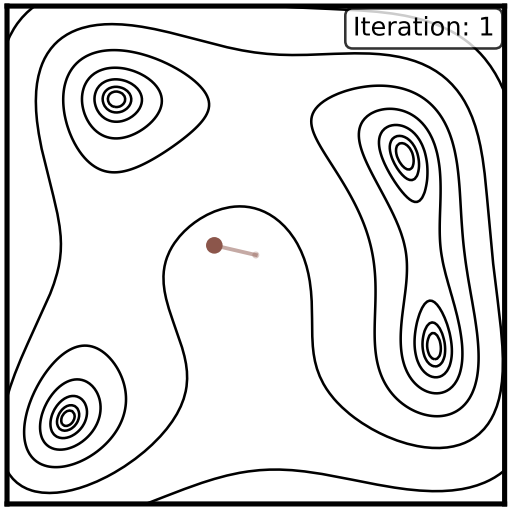
Metropolis-Hastings: error bars

David Yallup
Postdoc

- Sampling traditionally CPU-bound.
- Different algorithms, same GPU challenge.
- Need unified GPU-native framework.
- From optimization to model comparison.

- BlackJAX: Full JAX ecosystem.
- All algorithms GPU-accelerated.
- Gradient descent through nested sampling.
- Unified interface, maximum performance.

- Framework: more like `numpy` or `scipy` than `cobaya` or `cosmosis`.



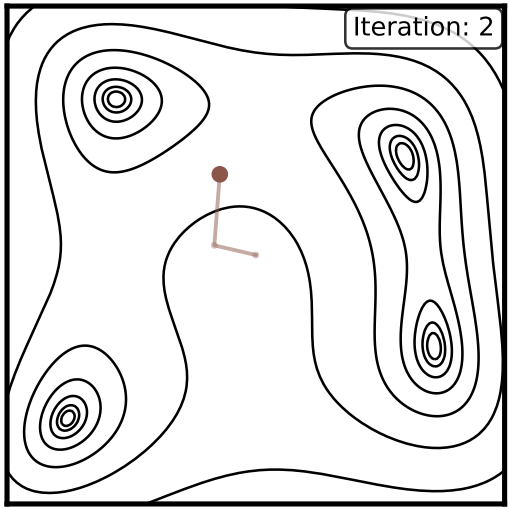
David Yallup

Postdoc

- Sampling traditionally CPU-bound.
- Different algorithms, same GPU challenge.
- Need unified GPU-native framework.
- From optimization to model comparison.

- BlackJAX: Full JAX ecosystem.
- All algorithms GPU-accelerated.
- Gradient descent through nested sampling.
- Unified interface, maximum performance.

- Framework: more like `numpy` or `scipy` than `cobaya` or `cosmosis`.



BlackJAX: GPU Native Sampling

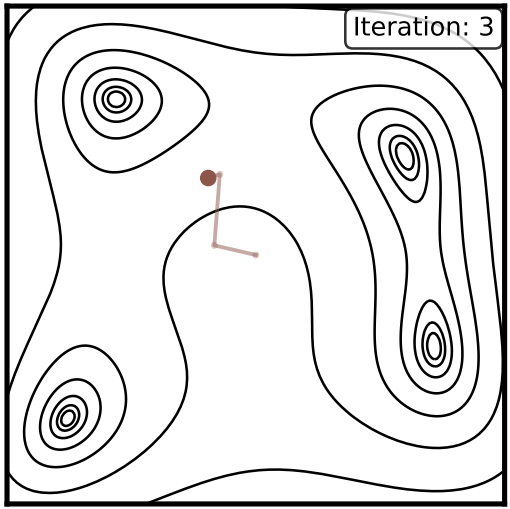
Metropolis-Hastings: error bars

David Yallup
Postdoc

- Sampling traditionally CPU-bound.
- Different algorithms, same GPU challenge.
- Need unified GPU-native framework.
- From optimization to model comparison.

- BlackJAX: Full JAX ecosystem.
- All algorithms GPU-accelerated.
- Gradient descent through nested sampling.
- Unified interface, maximum performance.

- Framework: more like `numpy` or `scipy` than `cobaya` or `cosmosis`.



BlackJAX: GPU Native Sampling

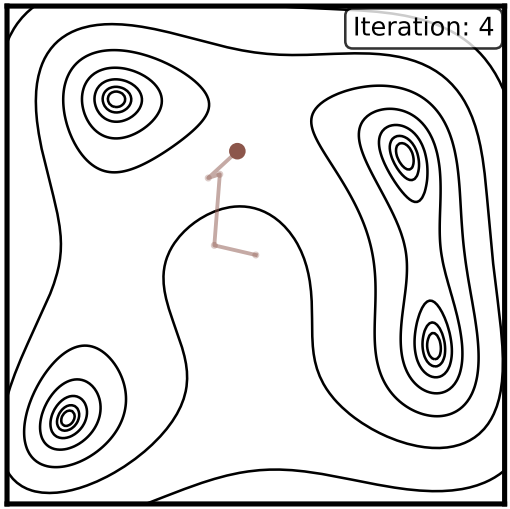
Metropolis-Hastings: error bars

David Yallup
Postdoc

- Sampling traditionally CPU-bound.
- Different algorithms, same GPU challenge.
- Need unified GPU-native framework.
- From optimization to model comparison.

- BlackJAX: Full JAX ecosystem.
- All algorithms GPU-accelerated.
- Gradient descent through nested sampling.
- Unified interface, maximum performance.

- Framework: more like `numpy` or `scipy` than `cobaya` or `cosmosis`.



BlackJAX: GPU Native Sampling

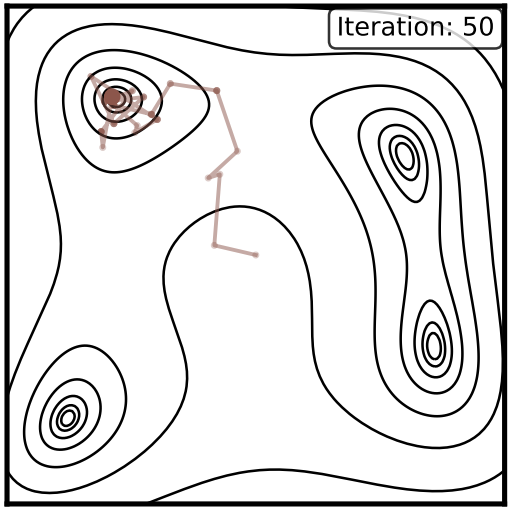
Metropolis-Hastings: error bars

David Yallup
Postdoc

- Sampling traditionally CPU-bound.
- Different algorithms, same GPU challenge.
- Need unified GPU-native framework.
- From optimization to model comparison.

- BlackJAX: Full JAX ecosystem.
- All algorithms GPU-accelerated.
- Gradient descent through nested sampling.
- Unified interface, maximum performance.

- Framework: more like `numpy` or `scipy` than `cobaya` or `cosmosis`.



BlackJAX: GPU Native Sampling

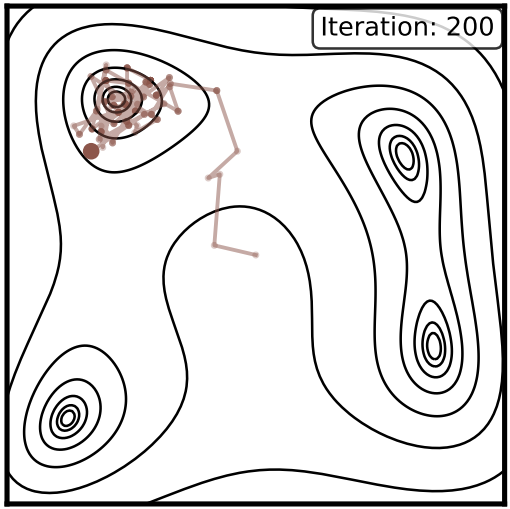
Metropolis-Hastings: error bars

David Yallup
Postdoc

- Sampling traditionally CPU-bound.
- Different algorithms, same GPU challenge.
- Need unified GPU-native framework.
- From optimization to model comparison.

- BlackJAX: Full JAX ecosystem.
- All algorithms GPU-accelerated.
- Gradient descent through nested sampling.
- Unified interface, maximum performance.

- Framework: more like `numpy` or `scipy` than `cobaya` or `cosmosis`.



BlackJAX: GPU Native Sampling

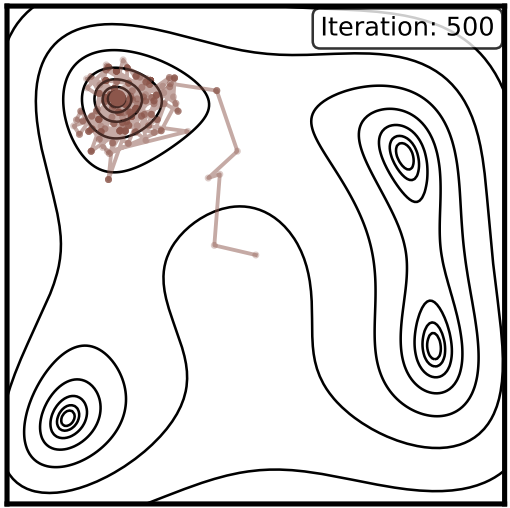
Metropolis-Hastings: error bars

David Yallup
Postdoc

- Sampling traditionally CPU-bound.
- Different algorithms, same GPU challenge.
- Need unified GPU-native framework.
- From optimization to model comparison.

- BlackJAX: Full JAX ecosystem.
- All algorithms GPU-accelerated.
- Gradient descent through nested sampling.
- Unified interface, maximum performance.

- Framework: more like `numpy` or `scipy` than `cobaya` or `cosmosis`.



BlackJAX: GPU Native Sampling

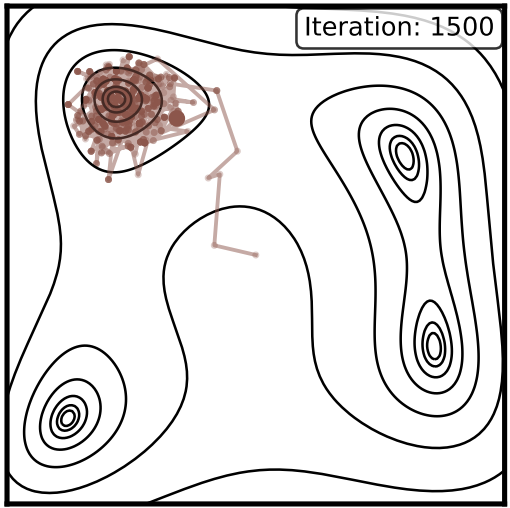
Metropolis-Hastings: error bars

David Yallup
Postdoc

- Sampling traditionally CPU-bound.
- Different algorithms, same GPU challenge.
- Need unified GPU-native framework.
- From optimization to model comparison.

- BlackJAX: Full JAX ecosystem.
- All algorithms GPU-accelerated.
- Gradient descent through nested sampling.
- Unified interface, maximum performance.

- Framework: more like `numpy` or `scipy` than `cobaya` or `cosmosis`.



BlackJAX: GPU Native Sampling

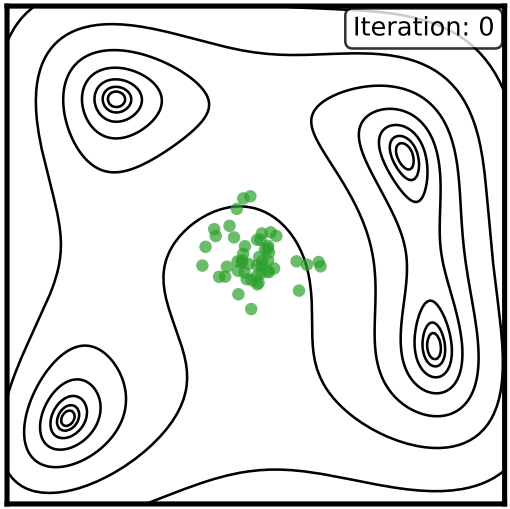
emcee: adaptive ensemble algorithms

David Yallup
Postdoc

- Sampling traditionally CPU-bound.
- Different algorithms, same GPU challenge.
- Need unified GPU-native framework.
- From optimization to model comparison.

- BlackJAX: Full JAX ecosystem.
- All algorithms GPU-accelerated.
- Gradient descent through nested sampling.
- Unified interface, maximum performance.

- Framework: more like `numpy` or `scipy` than `cobaya` or `cosmosis`.



BlackJAX: GPU Native Sampling

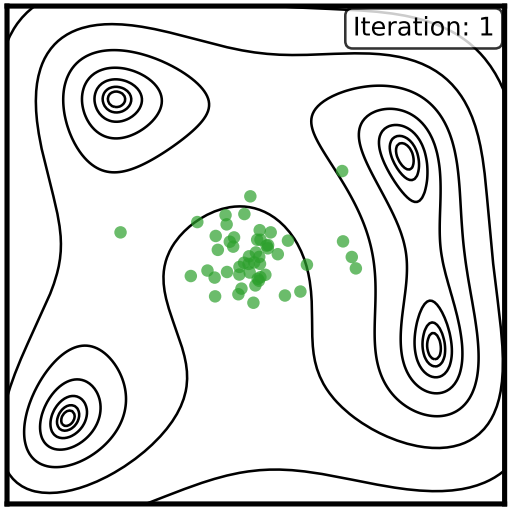
emcee: adaptive ensemble algorithms

David Yallup
Postdoc

- Sampling traditionally CPU-bound.
- Different algorithms, same GPU challenge.
- Need unified GPU-native framework.
- From optimization to model comparison.

- BlackJAX: Full JAX ecosystem.
- All algorithms GPU-accelerated.
- Gradient descent through nested sampling.
- Unified interface, maximum performance.

- Framework: more like `numpy` or `scipy` than `cobaya` or `cosmosis`.



BlackJAX: GPU Native Sampling

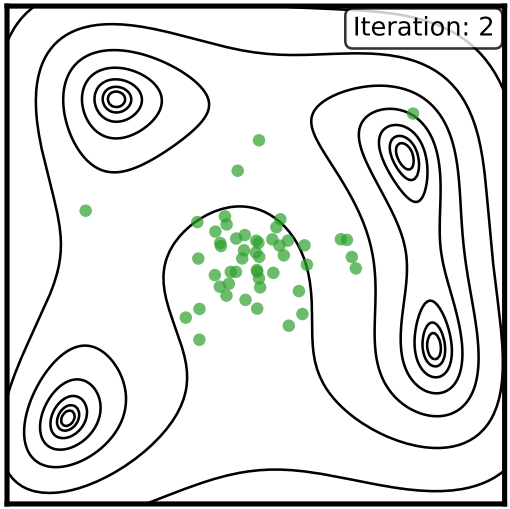
emcee: adaptive ensemble algorithms

David Yallup
Postdoc

- Sampling traditionally CPU-bound.
- Different algorithms, same GPU challenge.
- Need unified GPU-native framework.
- From optimization to model comparison.

- BlackJAX: Full JAX ecosystem.
- All algorithms GPU-accelerated.
- Gradient descent through nested sampling.
- Unified interface, maximum performance.

- Framework: more like `numpy` or `scipy` than `cobaya` or `cosmosis`.



BlackJAX: GPU Native Sampling

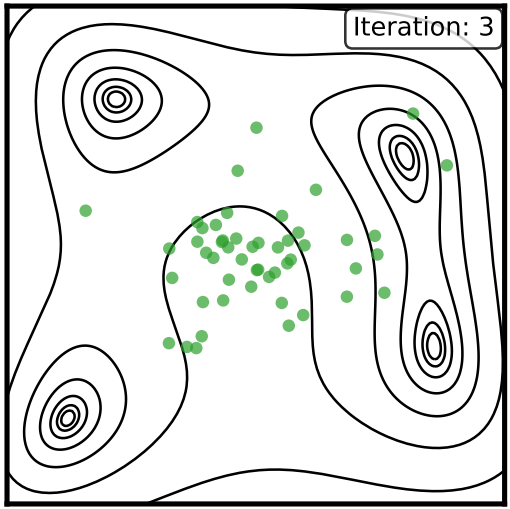
emcee: adaptive ensemble algorithms

David Yallup
Postdoc

- Sampling traditionally CPU-bound.
- Different algorithms, same GPU challenge.
- Need unified GPU-native framework.
- From optimization to model comparison.

- BlackJAX: Full JAX ecosystem.
- All algorithms GPU-accelerated.
- Gradient descent through nested sampling.
- Unified interface, maximum performance.

- Framework: more like `numpy` or `scipy` than `cobaya` or `cosmosis`.



BlackJAX: GPU Native Sampling

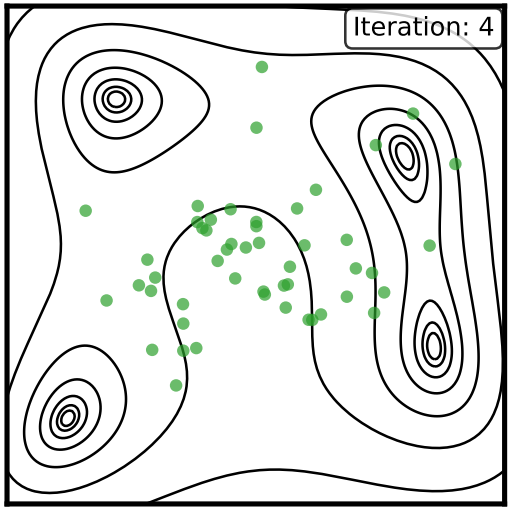
emcee: adaptive ensemble algorithms

David Yallup
Postdoc

- Sampling traditionally CPU-bound.
- Different algorithms, same GPU challenge.
- Need unified GPU-native framework.
- From optimization to model comparison.

- BlackJAX: Full JAX ecosystem.
- All algorithms GPU-accelerated.
- Gradient descent through nested sampling.
- Unified interface, maximum performance.

- Framework: more like `numpy` or `scipy` than `cobaya` or `cosmosis`.



BlackJAX: GPU Native Sampling

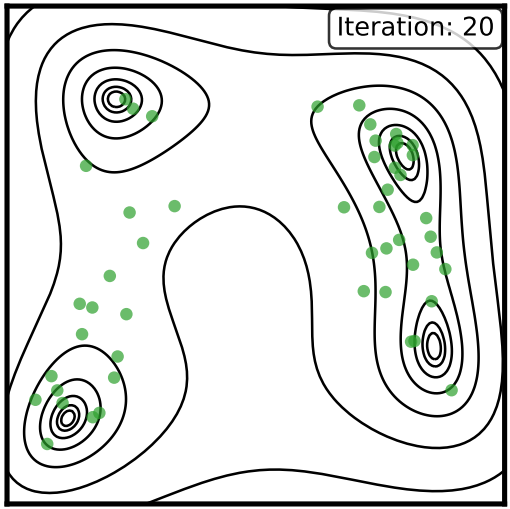
emcee: adaptive ensemble algorithms

David Yallup
Postdoc

- Sampling traditionally CPU-bound.
- Different algorithms, same GPU challenge.
- Need unified GPU-native framework.
- From optimization to model comparison.

- BlackJAX: Full JAX ecosystem.
- All algorithms GPU-accelerated.
- Gradient descent through nested sampling.
- Unified interface, maximum performance.

- Framework: more like `numpy` or `scipy` than `cobaya` or `cosmosis`.



BlackJAX: GPU Native Sampling

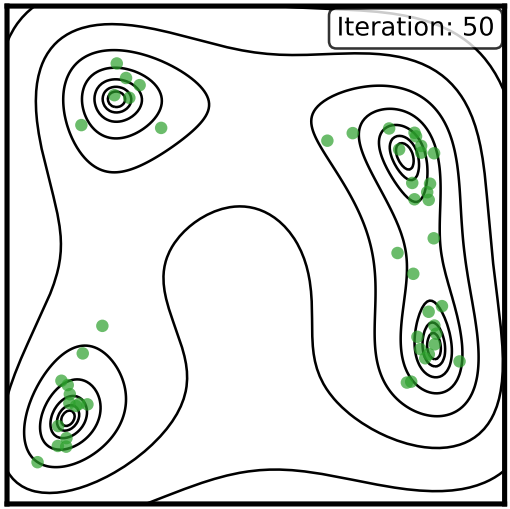
emcee: adaptive ensemble algorithms

David Yallup
Postdoc

- Sampling traditionally CPU-bound.
- Different algorithms, same GPU challenge.
- Need unified GPU-native framework.
- From optimization to model comparison.

- BlackJAX: Full JAX ecosystem.
- All algorithms GPU-accelerated.
- Gradient descent through nested sampling.
- Unified interface, maximum performance.

- Framework: more like `numpy` or `scipy` than `cobaya` or `cosmosis`.



BlackJAX: GPU Native Sampling

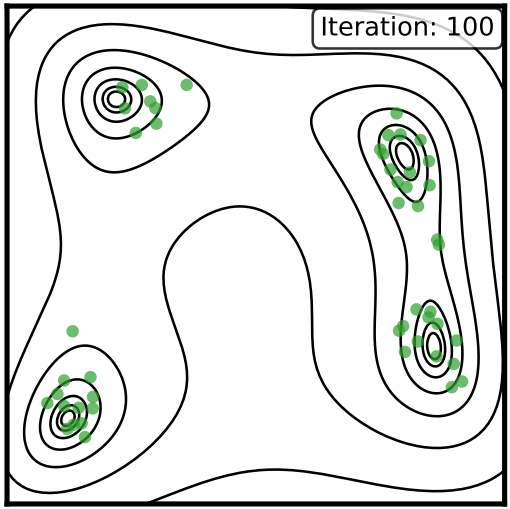
emcee: adaptive ensemble algorithms

David Yallup
Postdoc

- Sampling traditionally CPU-bound.
- Different algorithms, same GPU challenge.
- Need unified GPU-native framework.
- From optimization to model comparison.

- BlackJAX: Full JAX ecosystem.
- All algorithms GPU-accelerated.
- Gradient descent through nested sampling.
- Unified interface, maximum performance.

- Framework: more like `numpy` or `scipy` than `cobaya` or `cosmosis`.



BlackJAX: GPU Native Sampling

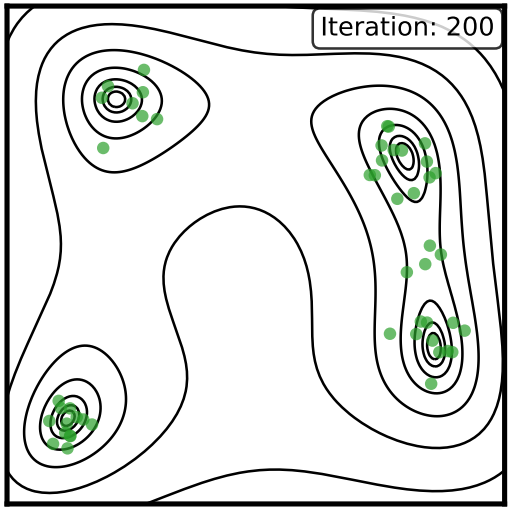
emcee: adaptive ensemble algorithms

David Yallup
Postdoc

- Sampling traditionally CPU-bound.
- Different algorithms, same GPU challenge.
- Need unified GPU-native framework.
- From optimization to model comparison.

- BlackJAX: Full JAX ecosystem.
- All algorithms GPU-accelerated.
- Gradient descent through nested sampling.
- Unified interface, maximum performance.

- Framework: more like `numpy` or `scipy` than `cobaya` or `cosmosis`.



BlackJAX: GPU Native Sampling

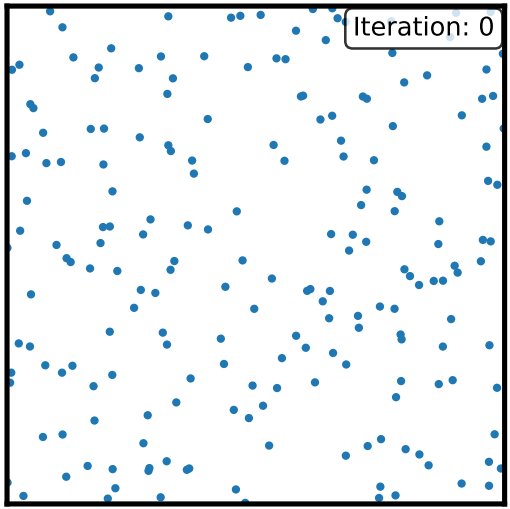
Nested sampling: model comparison

David Yallup
Postdoc

- Sampling traditionally CPU-bound.
- Different algorithms, same GPU challenge.
- Need unified GPU-native framework.
- From optimization to model comparison.

- BlackJAX: Full JAX ecosystem.
- All algorithms GPU-accelerated.
- Gradient descent through nested sampling.
- Unified interface, maximum performance.

- Framework: more like `numpy` or `scipy` than `cobaya` or `cosmosis`.



BlackJAX: GPU Native Sampling

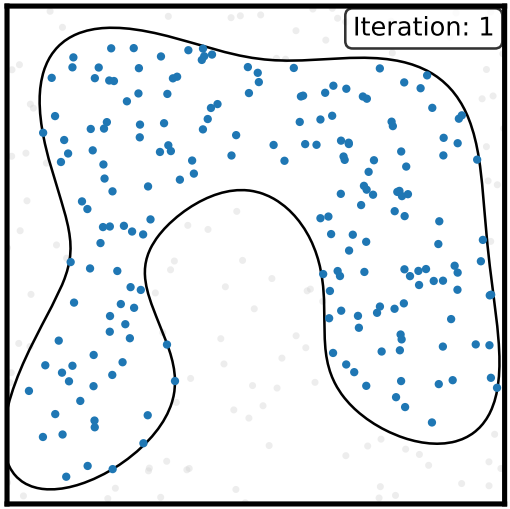
Nested sampling: model comparison

David Yallup
Postdoc

- Sampling traditionally CPU-bound.
- Different algorithms, same GPU challenge.
- Need unified GPU-native framework.
- From optimization to model comparison.

- BlackJAX: Full JAX ecosystem.
- All algorithms GPU-accelerated.
- Gradient descent through nested sampling.
- Unified interface, maximum performance.

- Framework: more like `numpy` or `scipy` than `cobaya` or `cosmosis`.



BlackJAX: GPU Native Sampling

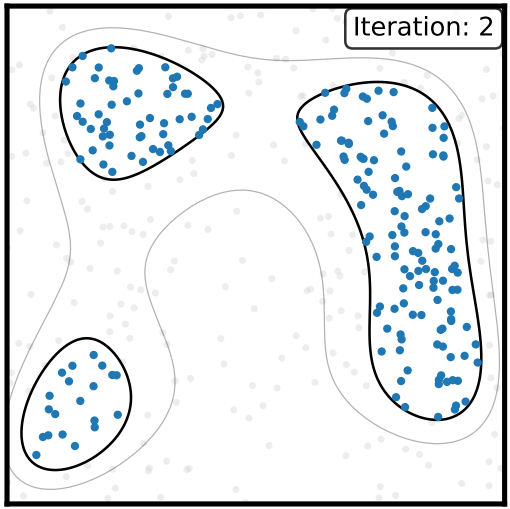
Nested sampling: model comparison

David Yallup
Postdoc

- Sampling traditionally CPU-bound.
- Different algorithms, same GPU challenge.
- Need unified GPU-native framework.
- From optimization to model comparison.

- BlackJAX: Full JAX ecosystem.
- All algorithms GPU-accelerated.
- Gradient descent through nested sampling.
- Unified interface, maximum performance.

- Framework: more like `numpy` or `scipy` than `cobaya` or `cosmosis`.



BlackJAX: GPU Native Sampling

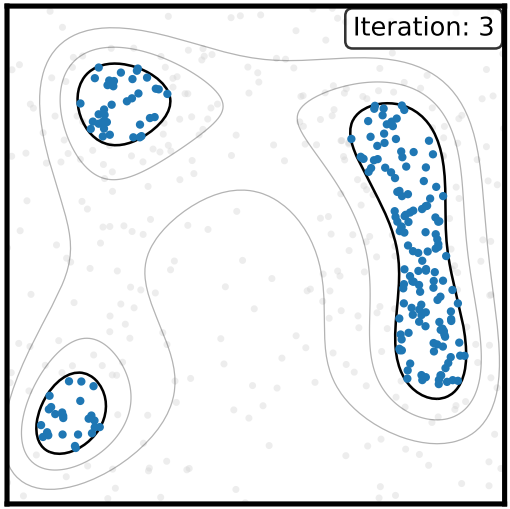
Nested sampling: model comparison

David Yallup
Postdoc

- Sampling traditionally CPU-bound.
- Different algorithms, same GPU challenge.
- Need unified GPU-native framework.
- From optimization to model comparison.

- BlackJAX: Full JAX ecosystem.
- All algorithms GPU-accelerated.
- Gradient descent through nested sampling.
- Unified interface, maximum performance.

- Framework: more like `numpy` or `scipy` than `cobaya` or `cosmosis`.



BlackJAX: GPU Native Sampling

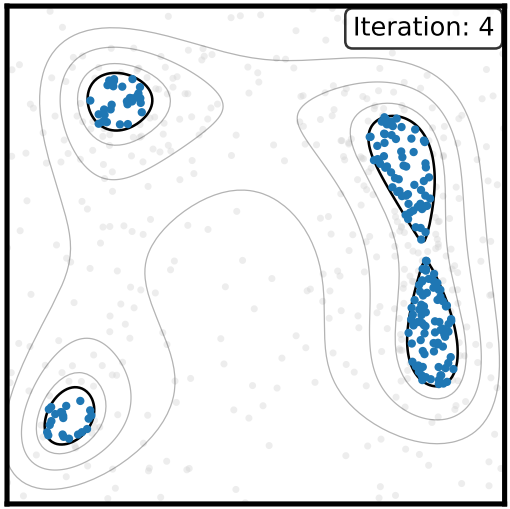
Nested sampling: model comparison

David Yallup
Postdoc

- Sampling traditionally CPU-bound.
- Different algorithms, same GPU challenge.
- Need unified GPU-native framework.
- From optimization to model comparison.

- BlackJAX: Full JAX ecosystem.
- All algorithms GPU-accelerated.
- Gradient descent through nested sampling.
- Unified interface, maximum performance.

- Framework: more like `numpy` or `scipy` than `cobaya` or `cosmosis`.



BlackJAX: GPU Native Sampling

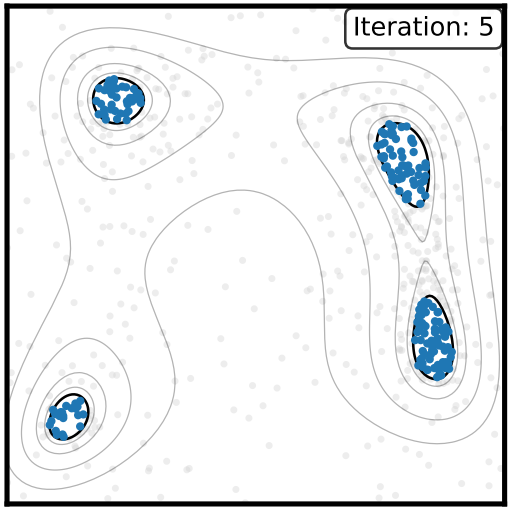
Nested sampling: model comparison

David Yallup
Postdoc

- Sampling traditionally CPU-bound.
- Different algorithms, same GPU challenge.
- Need unified GPU-native framework.
- From optimization to model comparison.

- BlackJAX: Full JAX ecosystem.
- All algorithms GPU-accelerated.
- Gradient descent through nested sampling.
- Unified interface, maximum performance.

- Framework: more like `numpy` or `scipy` than `cobaya` or `cosmosis`.



BlackJAX: GPU Native Sampling

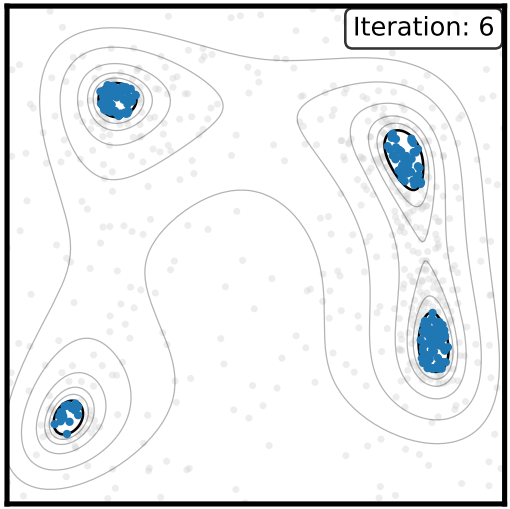
Nested sampling: model comparison

David Yallup
Postdoc

- Sampling traditionally CPU-bound.
- Different algorithms, same GPU challenge.
- Need unified GPU-native framework.
- From optimization to model comparison.

- BlackJAX: Full JAX ecosystem.
- All algorithms GPU-accelerated.
- Gradient descent through nested sampling.
- Unified interface, maximum performance.

- Framework: more like `numpy` or `scipy` than `cobaya` or `cosmosis`.



BlackJAX: GPU Native Sampling

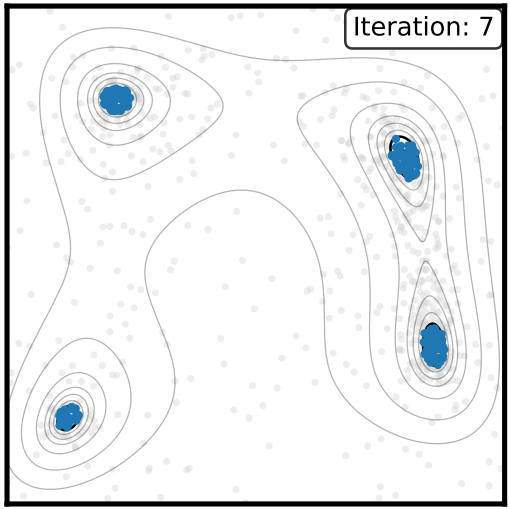
Nested sampling: model comparison

David Yallup
Postdoc

- Sampling traditionally CPU-bound.
- Different algorithms, same GPU challenge.
- Need unified GPU-native framework.
- From optimization to model comparison.

- BlackJAX: Full JAX ecosystem.
- All algorithms GPU-accelerated.
- Gradient descent through nested sampling.
- Unified interface, maximum performance.

- Framework: more like `numpy` or `scipy` than `cobaya` or `cosmosis`.



BlackJAX: GPU Native Sampling

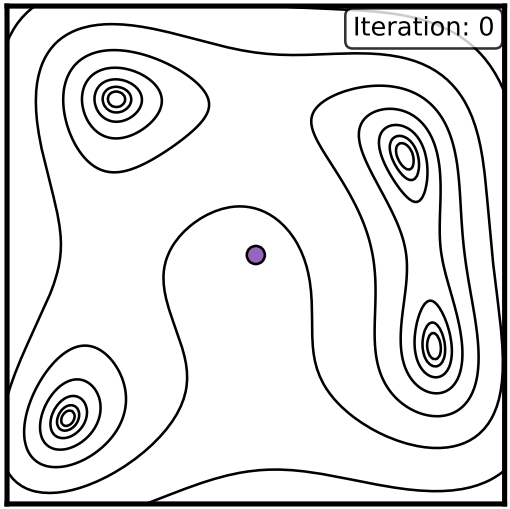
Hamiltonian Monte Carlo: inference with gradients

David Yallup
Postdoc

- Sampling traditionally CPU-bound.
- Different algorithms, same GPU challenge.
- Need unified GPU-native framework.
- From optimization to model comparison.

- BlackJAX: Full JAX ecosystem.
- All algorithms GPU-accelerated.
- Gradient descent through nested sampling.
- Unified interface, maximum performance.

- Framework: more like `numpy` or `scipy` than `cobaya` or `cosmosis`.



BlackJAX: GPU Native Sampling

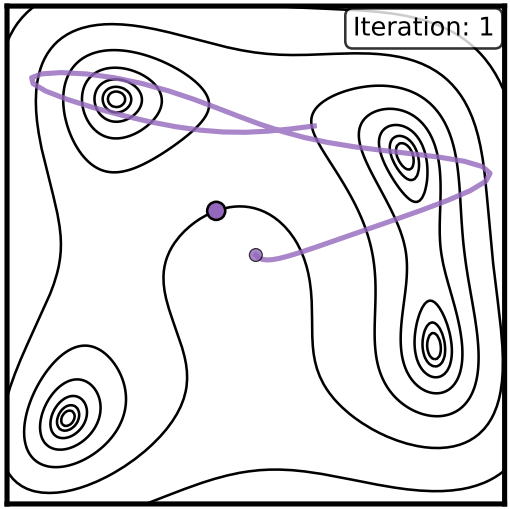
Hamiltonian Monte Carlo: inference with gradients

David Yallup
Postdoc

- Sampling traditionally CPU-bound.
- Different algorithms, same GPU challenge.
- Need unified GPU-native framework.
- From optimization to model comparison.

- BlackJAX: Full JAX ecosystem.
- All algorithms GPU-accelerated.
- Gradient descent through nested sampling.
- Unified interface, maximum performance.

- Framework: more like `numpy` or `scipy` than `cobaya` or `cosmosis`.



BlackJAX: GPU Native Sampling

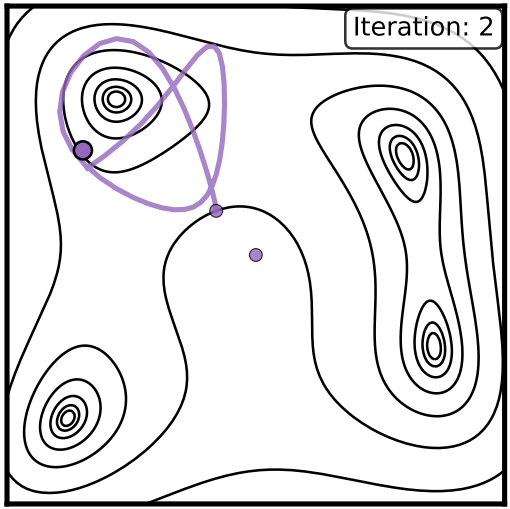
Hamiltonian Monte Carlo: inference with gradients

David Yallup
Postdoc

- Sampling traditionally CPU-bound.
- Different algorithms, same GPU challenge.
- Need unified GPU-native framework.
- From optimization to model comparison.

- BlackJAX: Full JAX ecosystem.
- All algorithms GPU-accelerated.
- Gradient descent through nested sampling.
- Unified interface, maximum performance.

- Framework: more like `numpy` or `scipy` than `cobaya` or `cosmosis`.



BlackJAX: GPU Native Sampling

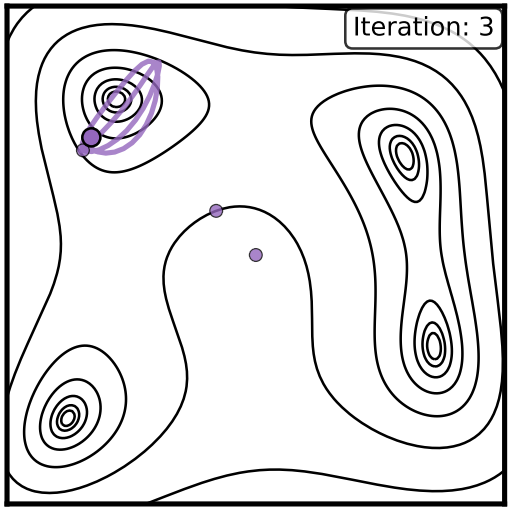
Hamiltonian Monte Carlo: inference with gradients

David Yallup
Postdoc

- Sampling traditionally CPU-bound.
- Different algorithms, same GPU challenge.
- Need unified GPU-native framework.
- From optimization to model comparison.

- BlackJAX: Full JAX ecosystem.
- All algorithms GPU-accelerated.
- Gradient descent through nested sampling.
- Unified interface, maximum performance.

- Framework: more like `numpy` or `scipy` than `cobaya` or `cosmosis`.



BlackJAX: GPU Native Sampling

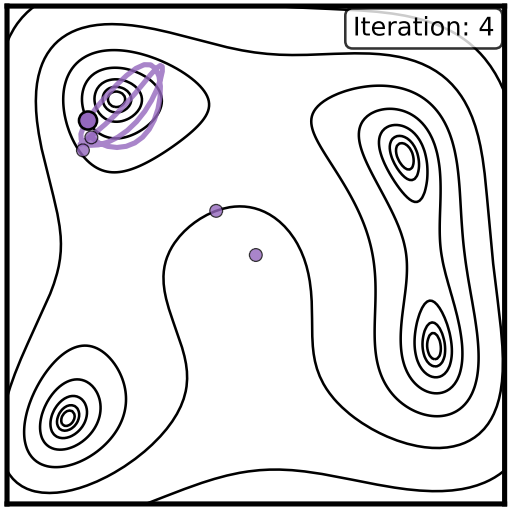
Hamiltonian Monte Carlo: inference with gradients

David Yallup
Postdoc

- Sampling traditionally CPU-bound.
- Different algorithms, same GPU challenge.
- Need unified GPU-native framework.
- From optimization to model comparison.

- BlackJAX: Full JAX ecosystem.
- All algorithms GPU-accelerated.
- Gradient descent through nested sampling.
- Unified interface, maximum performance.

- Framework: more like `numpy` or `scipy` than `cobaya` or `cosmosis`.



BlackJAX: GPU Native Sampling

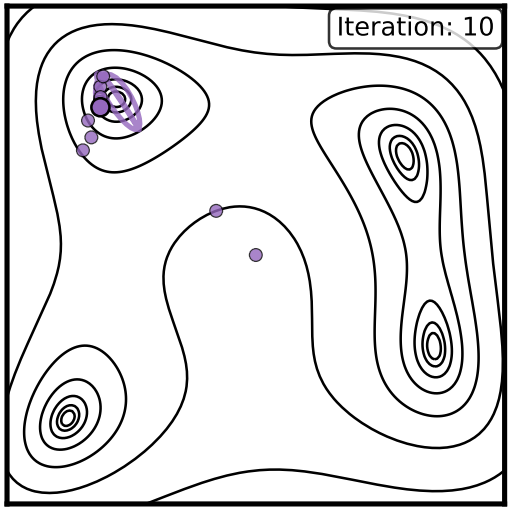
Hamiltonian Monte Carlo: inference with gradients

David Yallup
Postdoc

- Sampling traditionally CPU-bound.
- Different algorithms, same GPU challenge.
- Need unified GPU-native framework.
- From optimization to model comparison.

- BlackJAX: Full JAX ecosystem.
- All algorithms GPU-accelerated.
- Gradient descent through nested sampling.
- Unified interface, maximum performance.

- Framework: more like `numpy` or `scipy` than `cobaya` or `cosmosis`.



BlackJAX: GPU Native Sampling

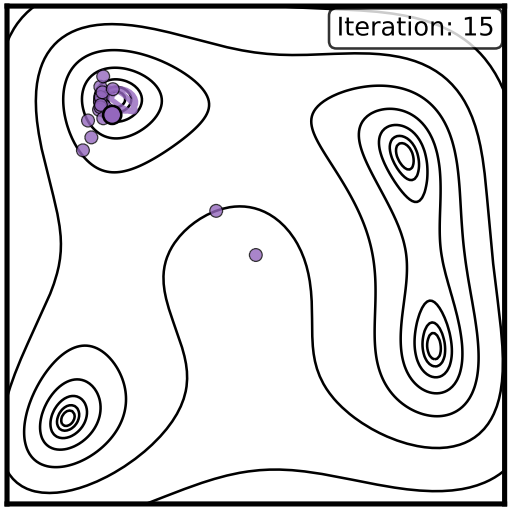
Hamiltonian Monte Carlo: inference with gradients

David Yallup
Postdoc

- Sampling traditionally CPU-bound.
- Different algorithms, same GPU challenge.
- Need unified GPU-native framework.
- From optimization to model comparison.

- BlackJAX: Full JAX ecosystem.
- All algorithms GPU-accelerated.
- Gradient descent through nested sampling.
- Unified interface, maximum performance.

- Framework: more like `numpy` or `scipy` than `cobaya` or `cosmosis`.



BlackJAX: GPU Native Sampling

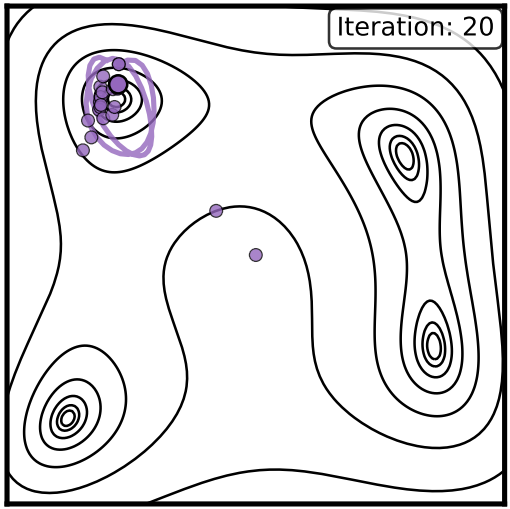
Hamiltonian Monte Carlo: inference with gradients

David Yallup
Postdoc

- Sampling traditionally CPU-bound.
- Different algorithms, same GPU challenge.
- Need unified GPU-native framework.
- From optimization to model comparison.

- BlackJAX: Full JAX ecosystem.
- All algorithms GPU-accelerated.
- Gradient descent through nested sampling.
- Unified interface, maximum performance.

- Framework: more like `numpy` or `scipy` than `cobaya` or `cosmosis`.

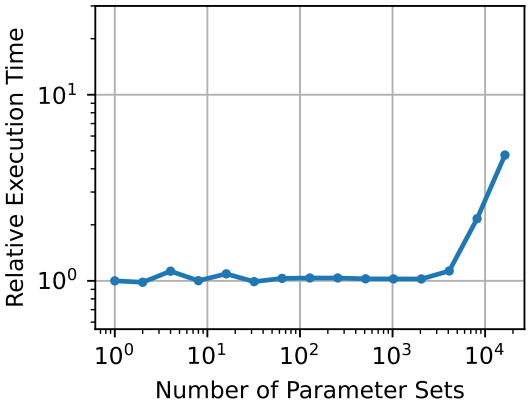


Recent GPU-Accelerated Applications

Case study 1/4: CMB and Cosmic Shear [2509.13307]

Toby Lovick
PhD

- **CMB (6 params)**: 300× speedup vs CPU PolyChord
- **Cosmic Shear (37 params)**: Days vs months
- **Method**: JAX neural emulators + GPU NS
- **Evidence**: Direct calculation with error bars
- **Models**: ΛCDM vs $w_0 w_a$ comparison
- **Impact**: NS competitive with MCMC+evidence methods

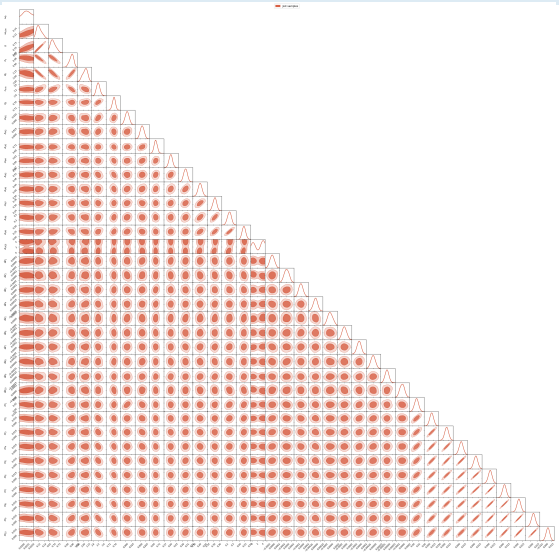


Recent GPU-Accelerated Applications

Case study 1/4: CMB and Cosmic Shear [2509.13307]

Toby Lovick
PhD

- **CMB (6 params)**: 300× speedup vs CPU PolyChord
- **Cosmic Shear (37 params)**: Days vs months
- **Method**: JAX neural emulators + GPU NS
- **Evidence**: Direct calculation with error bars
- **Models**: ΛCDM vs $w_0 w_a$ comparison
- **Impact**: NS competitive with MCMC+evidence methods

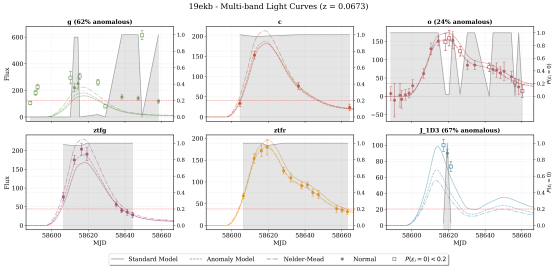


Recent GPU-Accelerated Applications

Case study 2/4: Bayesian Anomaly Detection for Type Ia Supernovae [2509.13394]

Sam Leeney
PhD

- **Problem**: Manual photometric rejection not scalable for LSST
- **Solution**: Bayesian anomaly detection integrated into SALT3 fitting
- **Method**: Model contamination probability per measurement
- **Result**: Automatic outlier/corrupted band rejection
- **Finding**: Contaminants systematically brighter/bluer
- **Impact**: Essential for unbiased cosmology at scale

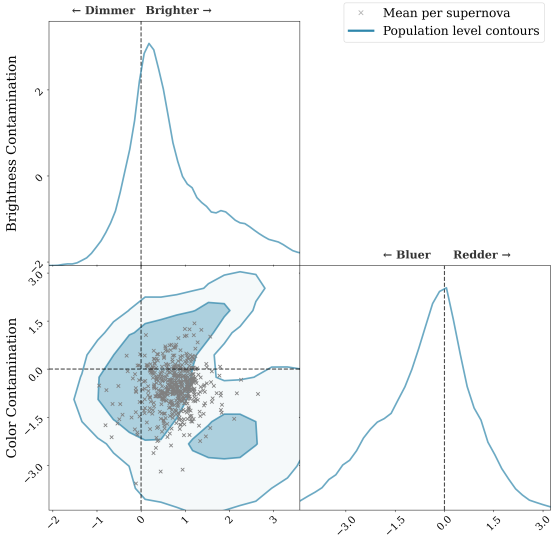


Recent GPU-Accelerated Applications

Case study 2/4: Bayesian Anomaly Detection for Type Ia Supernovae [2509.13394]

Sam Leeney
PhD

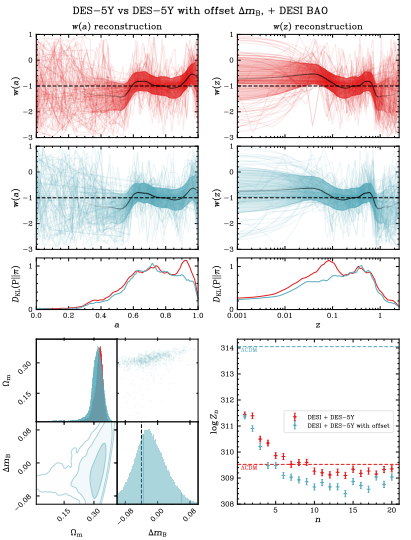
- **Problem**: Manual photometric rejection not scalable for LSST
- **Solution**: Bayesian anomaly detection integrated into SALT3 fitting
- **Method**: Model contamination probability per measurement
- **Result**: Automatic outlier/corrupted band rejection
- **Finding**: Contaminants systematically brighter/bluer
- **Impact**: Essential for unbiased cosmology at scale



Recent GPU-Accelerated Applications

Case study 3/4: Dark Energy vs Supernova Systematics [2509.13220]

- **Question**: DESI+DES $w_0 w_a$ preference - new physics or systematics?
- **Method**: Bayesian model comparison
- **Models**: Dynamic DE vs redshift-dependent SN bias
- **Result**: Systematics fit equally well with lower complexity
- **Evidence**: Favors systematic explanation
- **Lesson**: Test mundane before claiming exotic

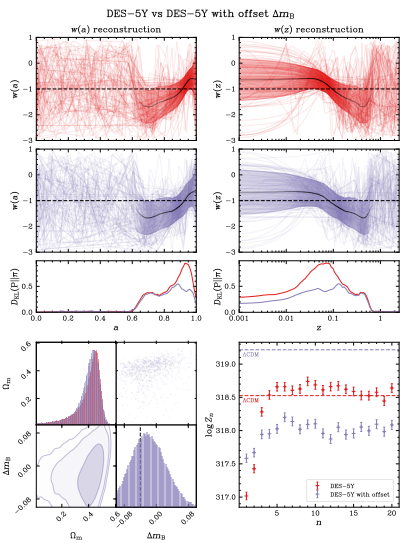


Recent GPU-Accelerated Applications

Case study 3/4: Dark Energy vs Supernova Systematics [2509.13220]

Adam Ormondroyd
PhD

- **Question**: DESI+DES $w_0 w_a$ preference - new physics or systematics?
- **Method**: Bayesian model comparison
- **Models**: Dynamic DE vs redshift-dependent SN bias
- **Result**: Systematics fit equally well with lower complexity
- **Evidence**: Favors systematic explanation
- **Lesson**: Test mundane before claiming exotic

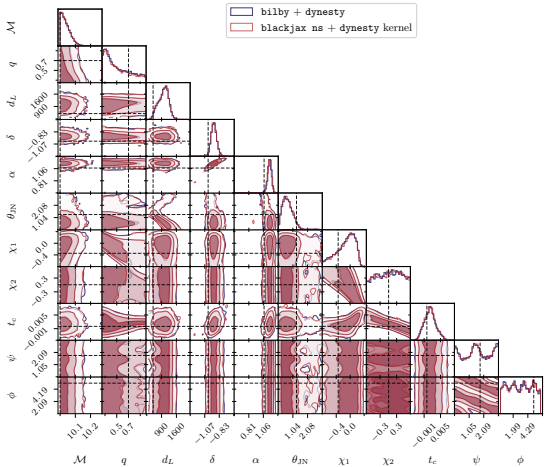


Recent GPU-Accelerated Applications

Case study 4/4: Gravitational Wave Inference [2509.04336]

Metha Prathaban
PhD

- **Goal**: GPU-accelerate bilby's acceptance-walk NS
- **Implementation**: Faithful port to blackjax-ns
- **Performance**: 20-40× speedup for BBH
- **Validation**: Identical posteriors/evidences
- **Hardware**: Single GPU vs CPU clusters
- **Impact**: Clean baseline for future methods

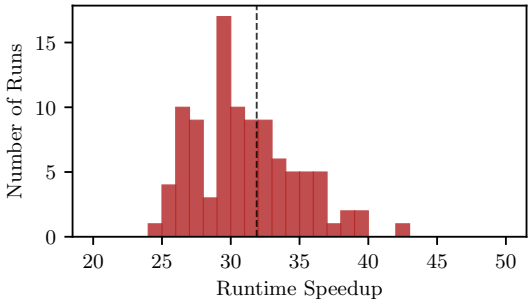


Recent GPU-Accelerated Applications

Case study 4/4: Gravitational Wave Inference [2509.04336]

Metha Prathaban
PhD

- **Goal**: GPU-accelerate bilby's acceptance-walk NS
- **Implementation**: Faithful port to blackjax-ns
- **Performance**: 20-40× speedup for BBH
- **Validation**: Identical posteriors/evidences
- **Hardware**: Single GPU vs CPU clusters
- **Impact**: Clean baseline for future methods



The Future: AI in Scientific Code Development

The Real AI Revolution: LLMs

The biggest impact of AI will not be in analyzing data, but in helping us write the code to do it.

- **Automated code translation**: LLMs can help port legacy Fortran/C++ models to modern, GPU-friendly & differentiable frameworks like JAX or PyTorch.

The 80/20 Rule of Scientific Work

- **80% "boring" tasks**: Writing code, debugging, drafting & reviewing papers, munging data, organising meetings...
- **20% "hard thinking"**: The actual scientific insight.

AI's biggest immediate impact is automating and accelerating the 80%, freeing up human time for the 20%.

Key Message

AI is not just a tool for analysis; it's about to fundamentally change how we develop, optimize, and deploy our science

Conclusions

github.com/handley-lab/group

1. **GPU $\neq$ Machine Learning: Two Independent Capabilities**
   - GPUs accelerate any parallel algorithm.
   - Automatic differentiation + massive parallelization.
   - Often confused, serve different purposes.
2. **Classical Algorithms on GPU Competitive with ML State of the Art**
   - Traditional physics methods + GPU = superior performance.
3. **AI Accelerates Development as well as Computation**
   - LLMs solve the GPU porting challenge at scale.
   - 10$\times$ development speedup enables widespread adoption.

Get Started with GPU-Accelerated Sampling

handley-lab.co.uk/nested-sampling-book