Next-Generation Model Comparison for Primordial Cosmology

Will Handley
<wh260@cam.ac.uk>

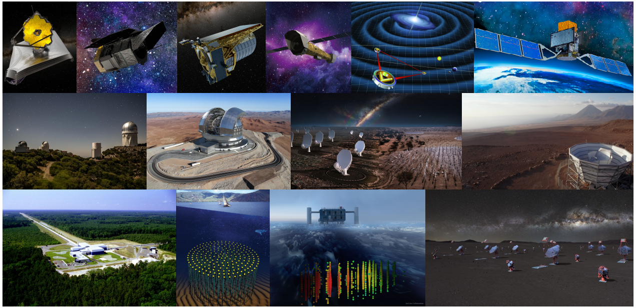
Royal Society University Research Fellow
Institute of Astronomy, University of Cambridge
Kavli Institute for Cosmology, Cambridge
Gonville & Caius College
willhandley.co.uk/talks

16th September 2026

Beginning the golden age of astronomy data

- Over our research lifetimes we will see next-generation data rates across the electromagnetic spectrum & beyond:

Radio	SKA *et al.*
Micro	SO/CMB-S4/LiteBIRD.
IR	JWST, Roman.
Optical	Euclid, DESI, Rubin, EELT.
X-ray	Athena.
Gamma-ray	e-ASTROGAM.
Gravitational	LIGO/ET/LISA.
Particle	CTA, IceCube, KM3NeT.



- This data deluge creates unprecedented computational challenges for model comparison and parameter estimation.
- Traditional computing approaches will not scale to these next-generation data volumes and complexity.

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× 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.
- **Boltzmann solvers**: Vectorized across k-modes.
- **N-body sims**: GPU acceleration.

Capability 2: Massive 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 parallelize without gradients.
They serve different purposes.

BlackJAX: GPU Native Sampling [2402.10797]

Gradient descent: inference at speed

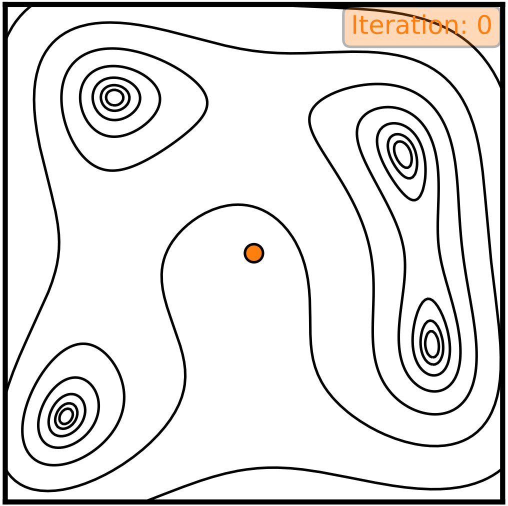
David Yallup

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 [2402.10797]

Gradient descent: inference at speed

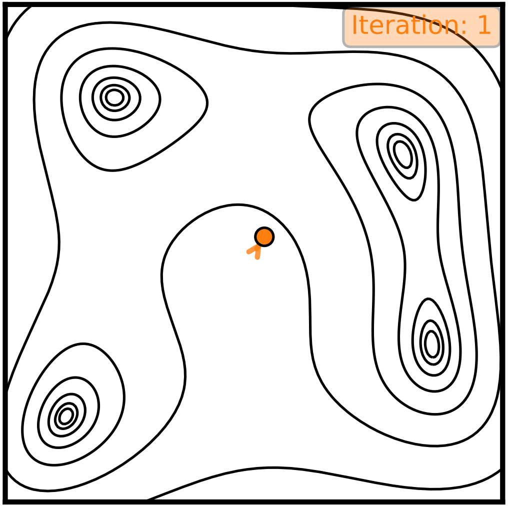
David Yallup

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 [2402.10797]

Gradient descent: inference at speed

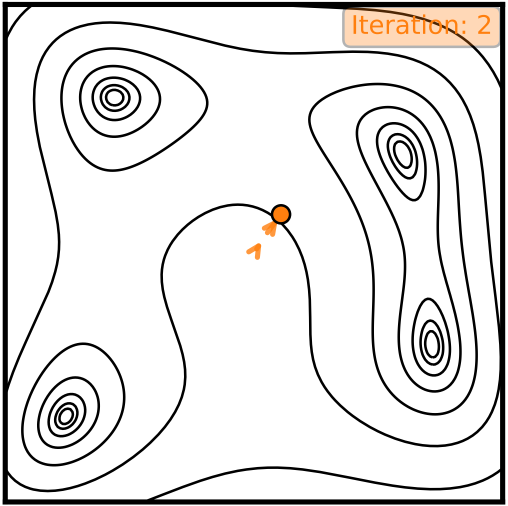
David Yallup

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 [2402.10797]

Gradient descent: inference at speed

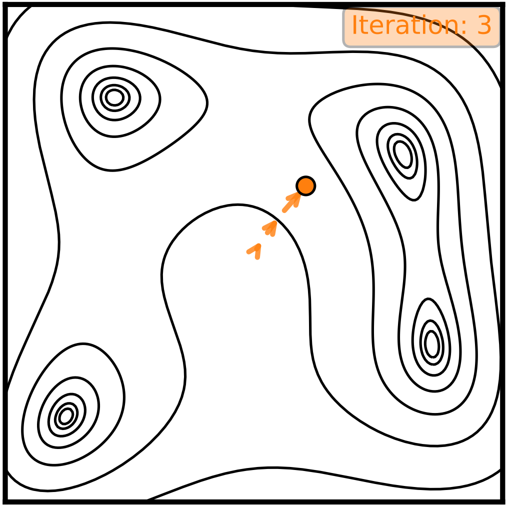
David Yallup

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 [2402.10797]

Gradient descent: inference at speed

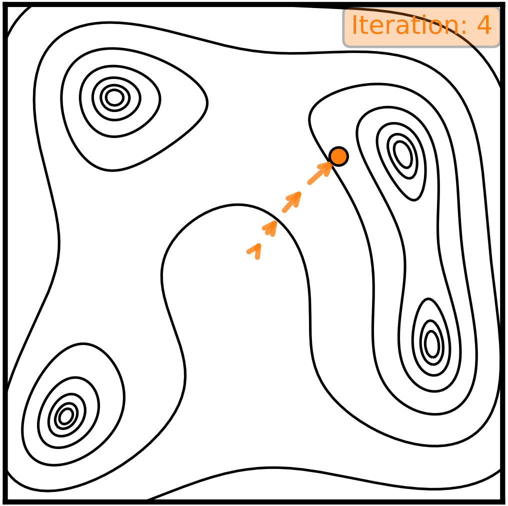
David Yallup

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 [2402.10797]

Gradient descent: inference at speed

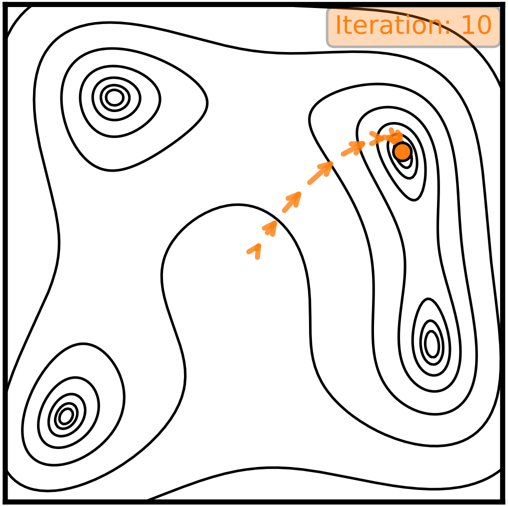
David Yallup

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 [2402.10797]

Gradient descent: inference at speed

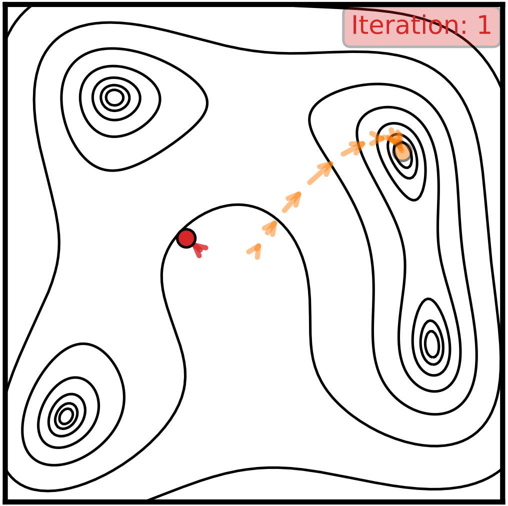
David Yallup

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 [2402.10797]

Gradient descent: inference at speed

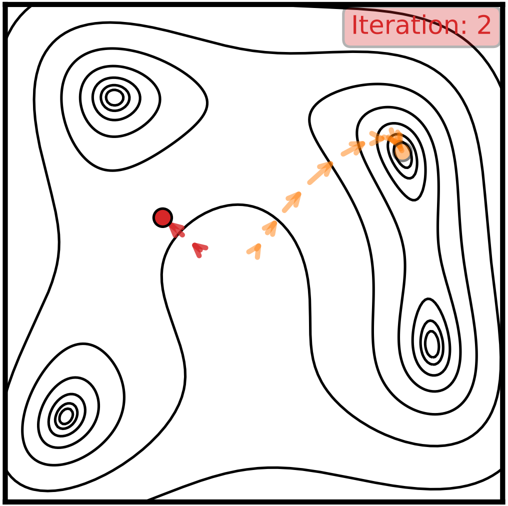
David Yallup

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 [2402.10797]

Gradient descent: inference at speed

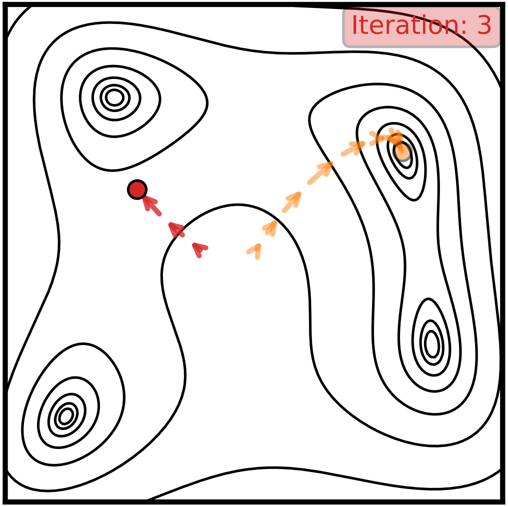
David Yallup

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 [2402.10797]

Gradient descent: inference at speed

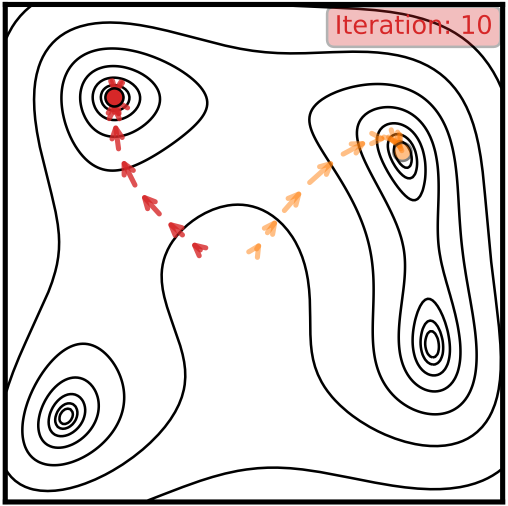
David Yallup

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 [2402.10797]

Metropolis-Hastings: error bars

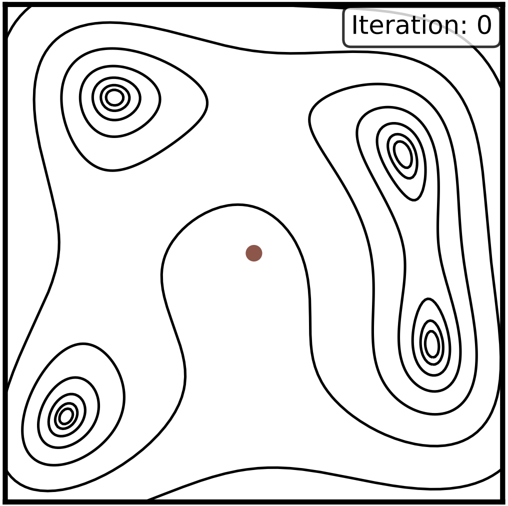
David Yallup

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 [2402.10797]

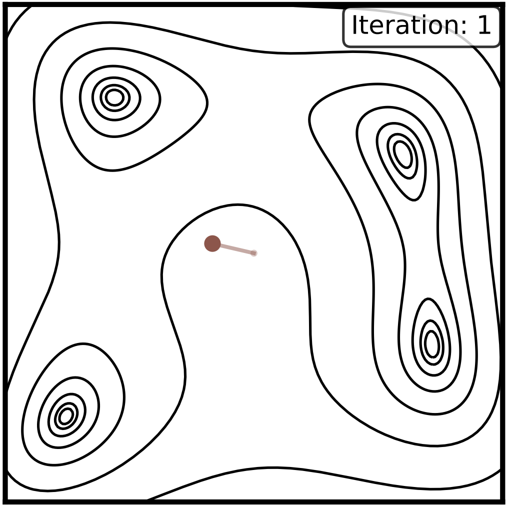
Metropolis-Hastings: error bars

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`.



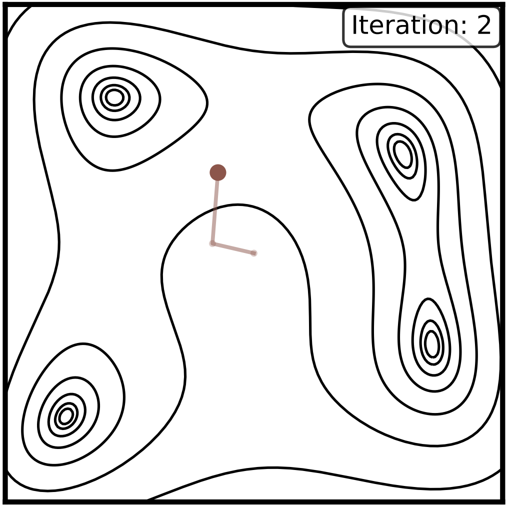
BlackJAX: GPU Native Sampling [2402.10797]

Metropolis-Hastings: error bars

- 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 [2402.10797]

Metropolis-Hastings: error bars

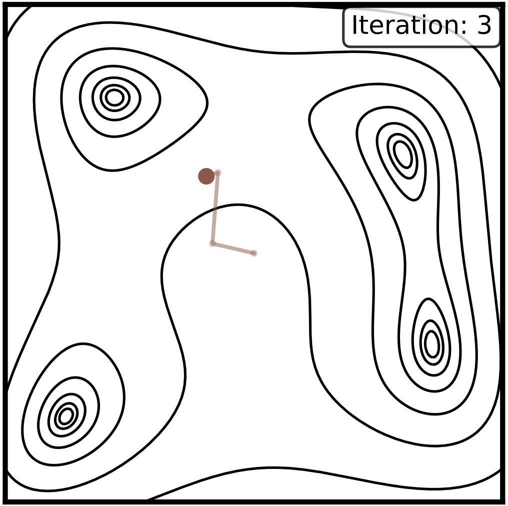
David Yallup

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 [2402.10797]

Metropolis-Hastings: error bars

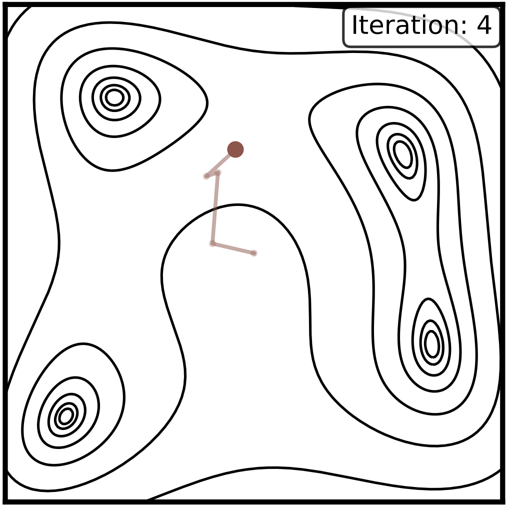
David Yallup

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 [2402.10797]

Metropolis-Hastings: error bars

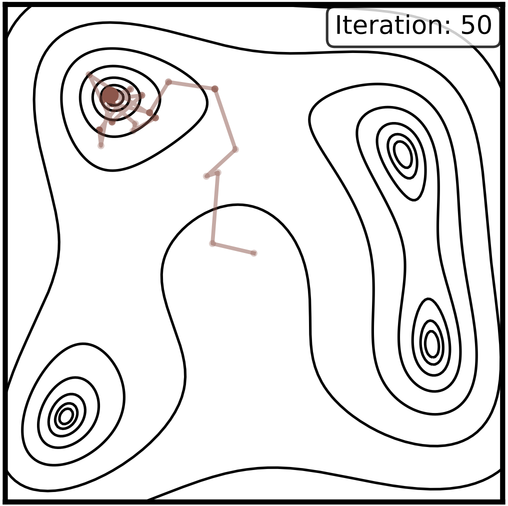
David Yallup

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 [2402.10797]

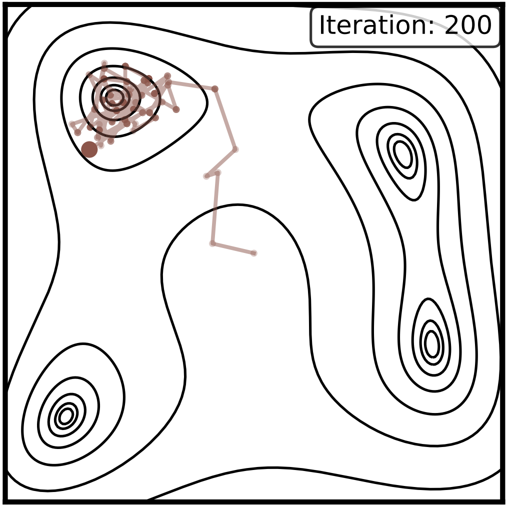
Metropolis-Hastings: error bars

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 [2402.10797]

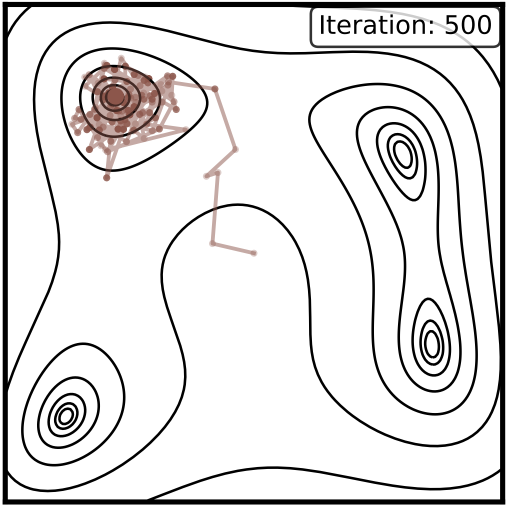
Metropolis-Hastings: error bars

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`.



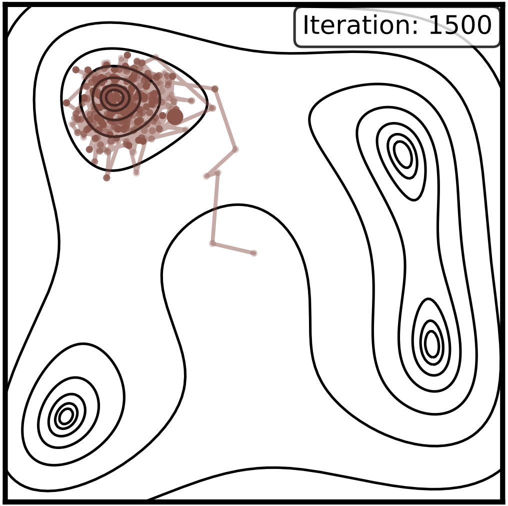
BlackJAX: GPU Native Sampling [2402.10797]

Metropolis-Hastings: error bars

- 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 [2402.10797]

emcee: adaptive ensemble algorithms

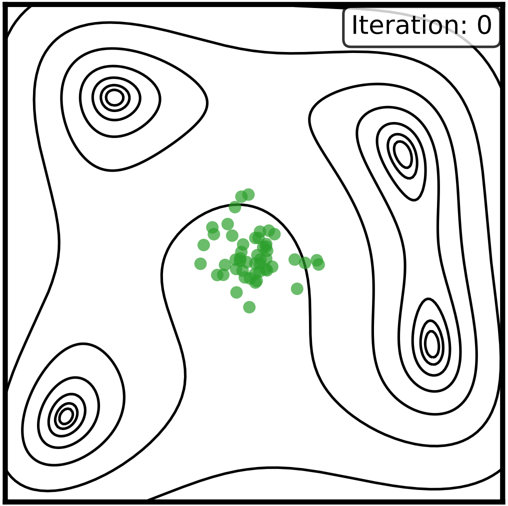
David Yallup

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 [2402.10797]

emcee: adaptive ensemble algorithms

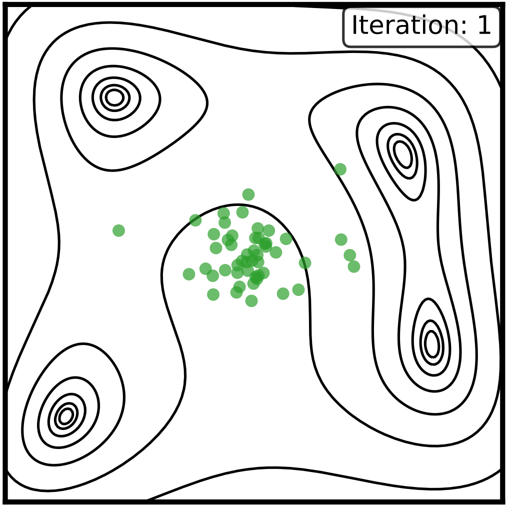
David Yallup

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 [2402.10797]

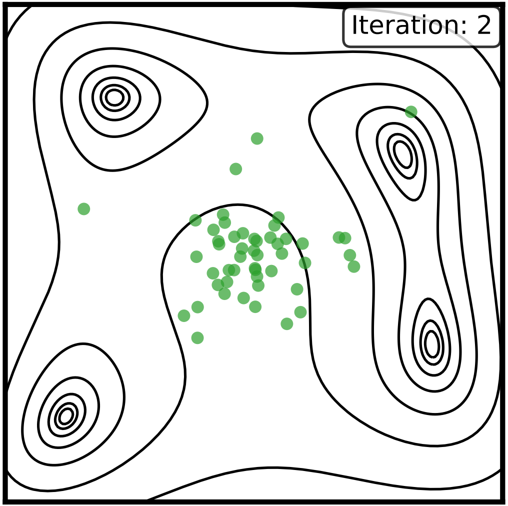
emcee: adaptive ensemble algorithms

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 [2402.10797]

emcee: adaptive ensemble algorithms

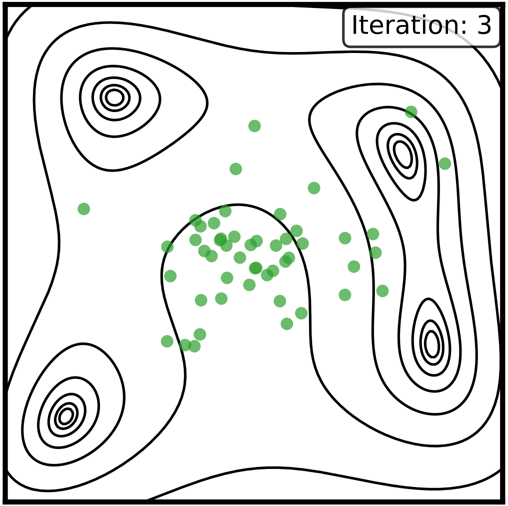
David Yallup

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 [2402.10797]

emcee: adaptive ensemble algorithms

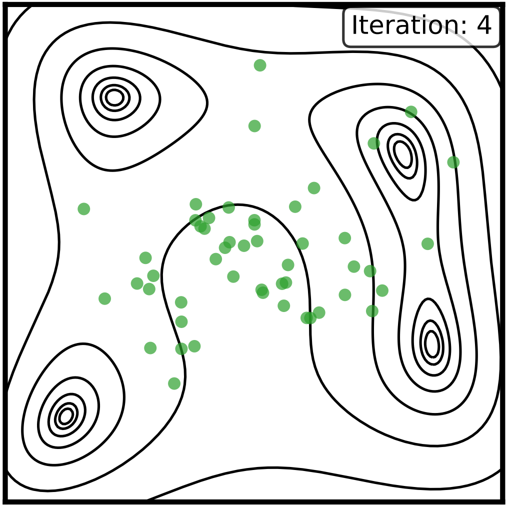
David Yallup

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 [2402.10797]

emcee: adaptive ensemble algorithms

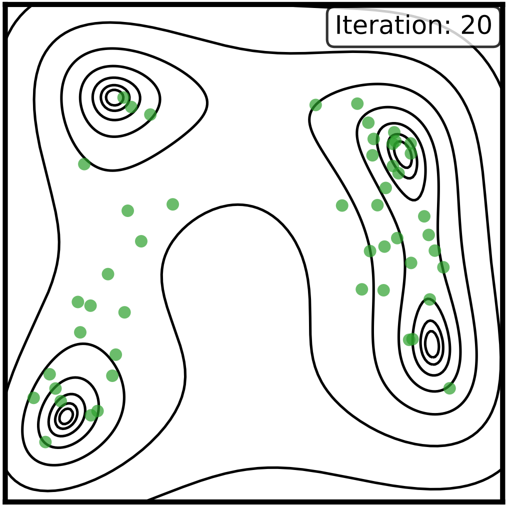
David Yallup

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 [2402.10797]

emcee: adaptive ensemble algorithms

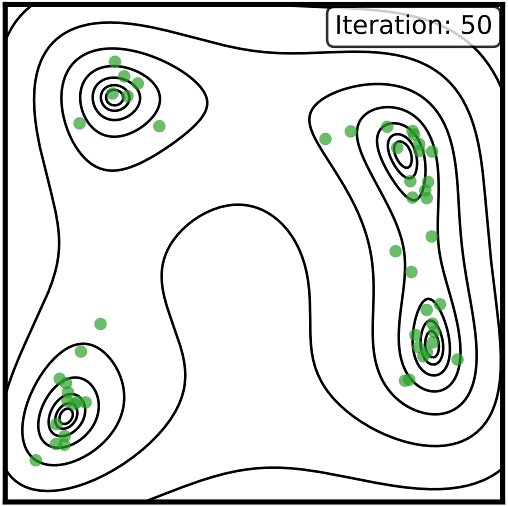
David Yallup

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 [2402.10797]

emcee: adaptive ensemble algorithms

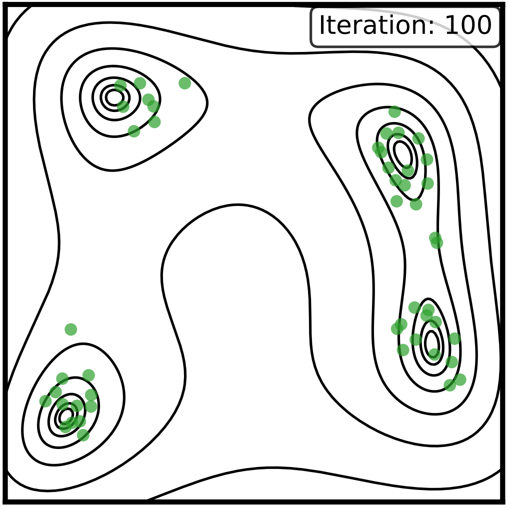
David Yallup

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 [2402.10797]

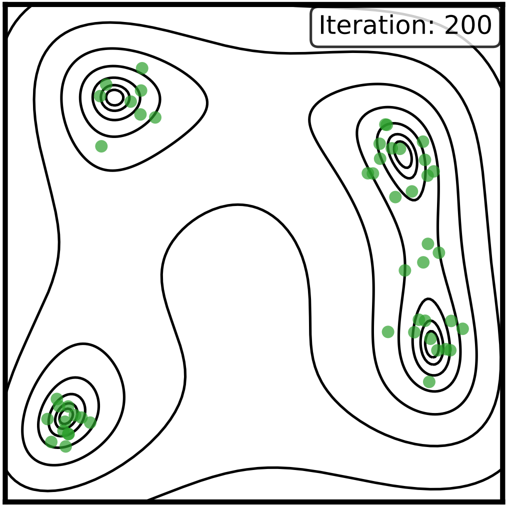
emcee: adaptive ensemble algorithms

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 [2402.10797]

Nested sampling: model comparison

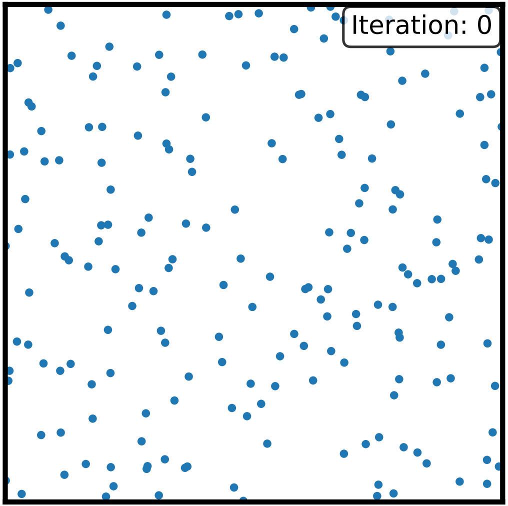
David Yallup

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 [2402.10797]

Nested sampling: model comparison

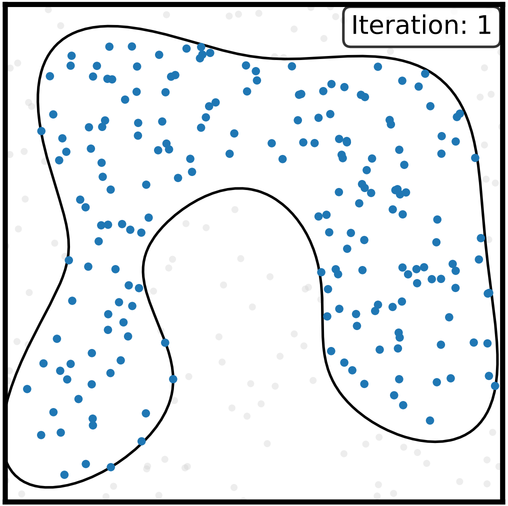
David Yallup

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 [2402.10797]

Nested sampling: model comparison

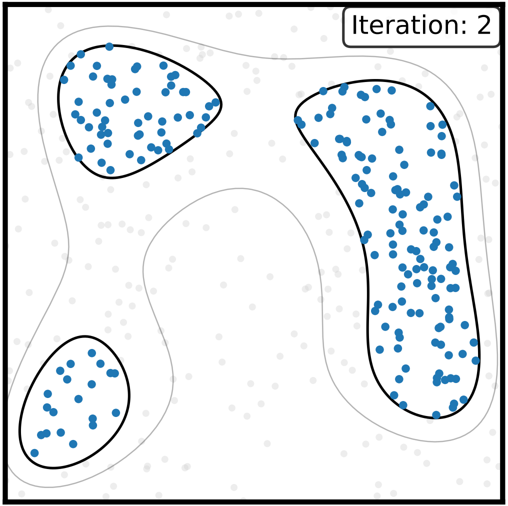
David Yallup

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 [2402.10797]

Nested sampling: model comparison

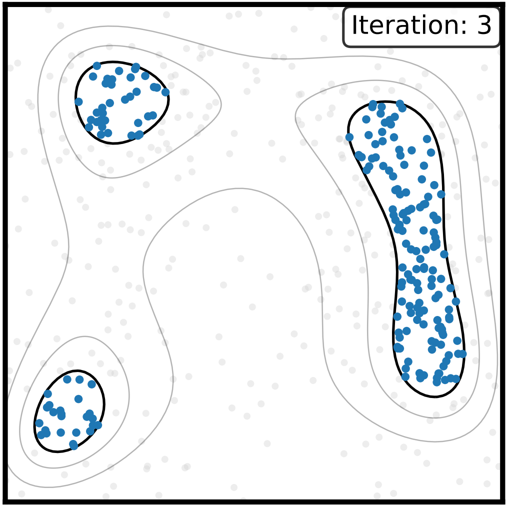
David Yallup

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 [2402.10797]

Nested sampling: model comparison

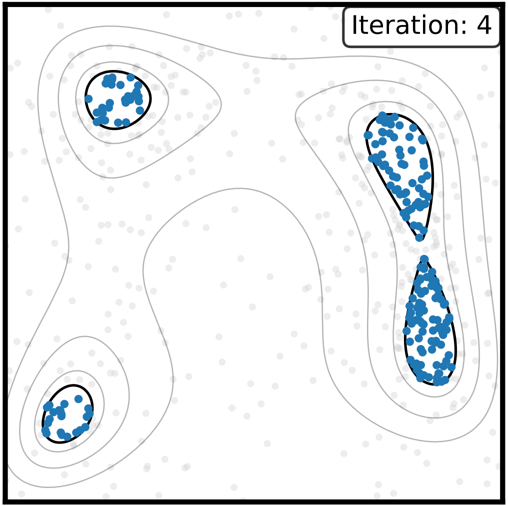
David Yallup

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 [2402.10797]

Nested sampling: model comparison

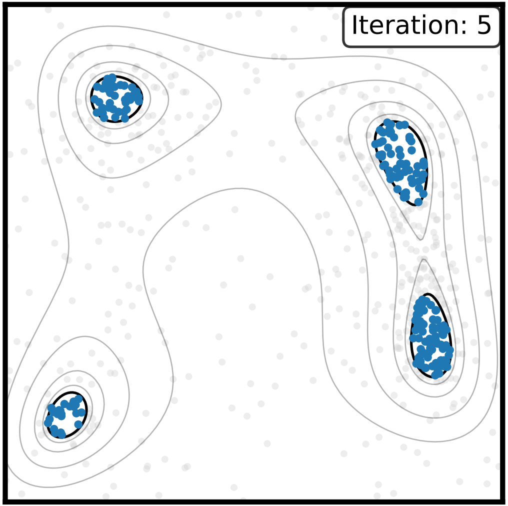
David Yallup

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 [2402.10797]

Nested sampling: model comparison

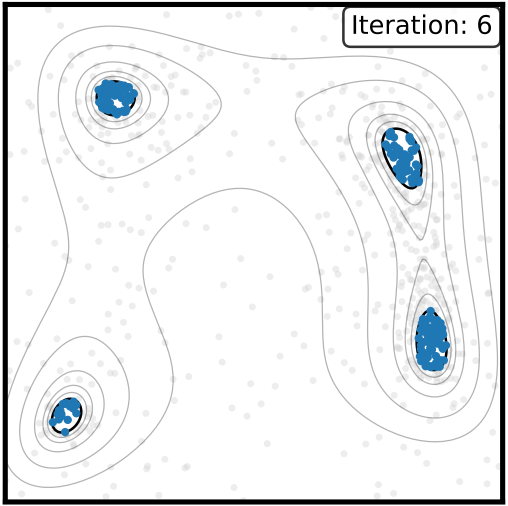
David Yallup

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 [2402.10797]

Nested sampling: model comparison

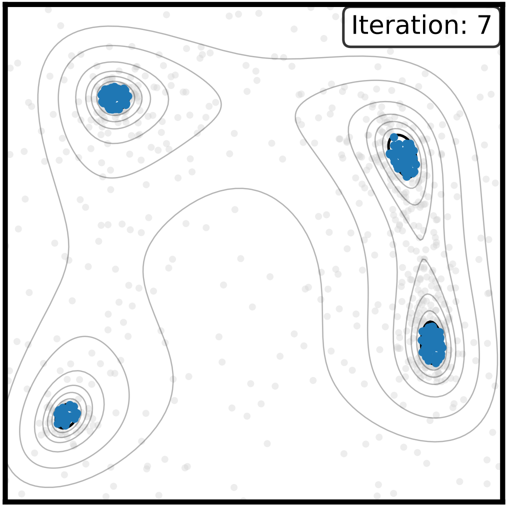
David Yallup

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 [2402.10797]

Hamiltonian Monte Carlo: inference with gradients

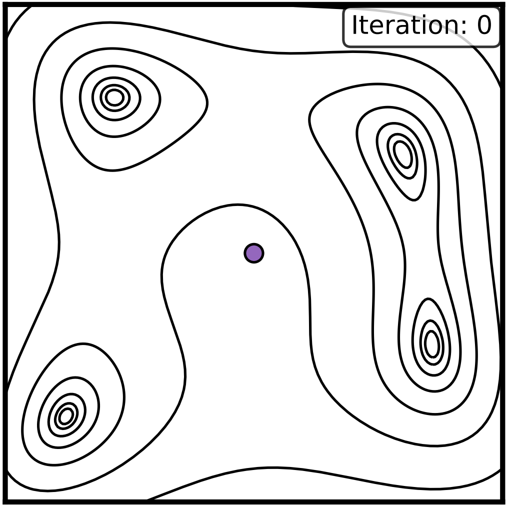
David Yallup

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 [2402.10797]

Hamiltonian Monte Carlo: inference with gradients

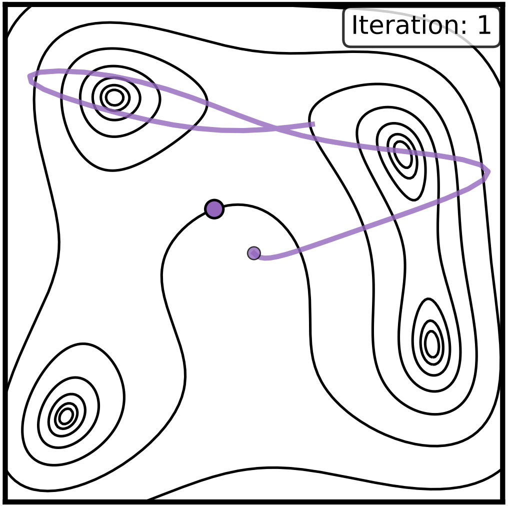
David Yallup

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 [2402.10797]

Hamiltonian Monte Carlo: inference with gradients

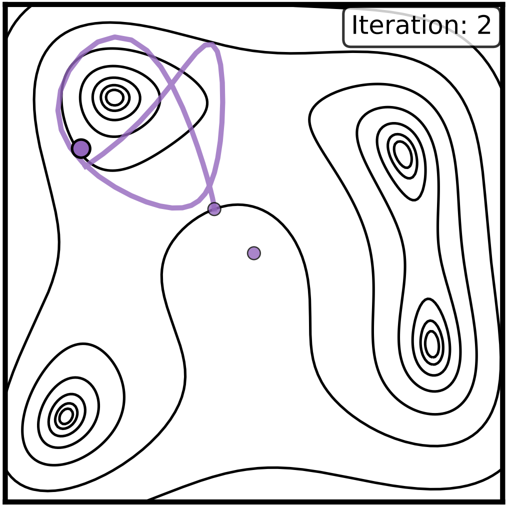
David Yallup

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 [2402.10797]

Hamiltonian Monte Carlo: inference with gradients

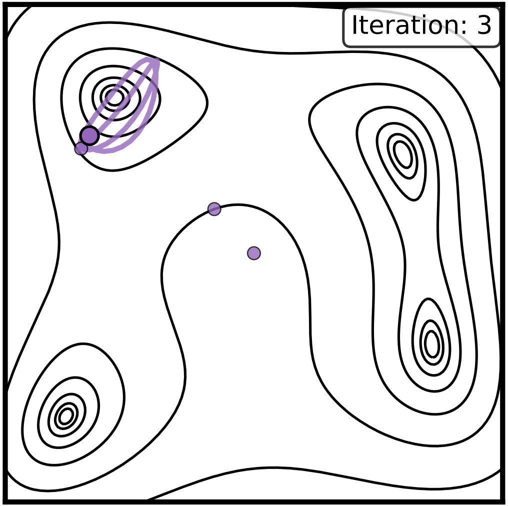
David Yallup

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 [2402.10797]

Hamiltonian Monte Carlo: inference with gradients

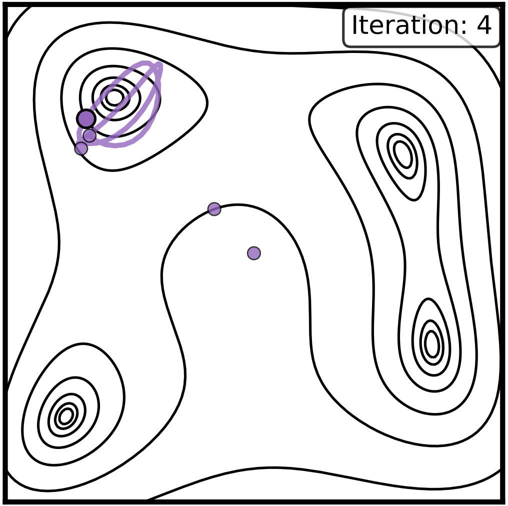
David Yallup

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 [2402.10797]

Hamiltonian Monte Carlo: inference with gradients

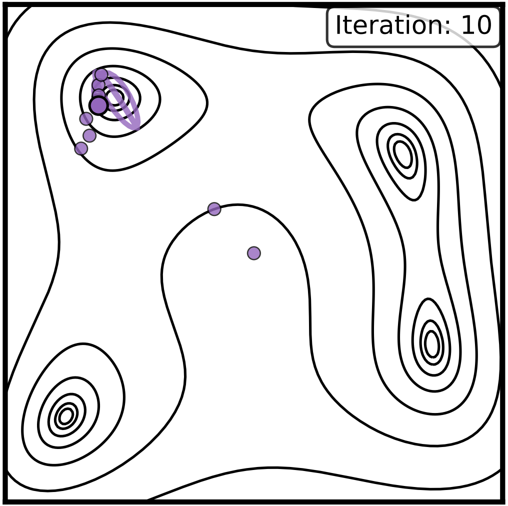
David Yallup

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 [2402.10797]

Hamiltonian Monte Carlo: inference with gradients

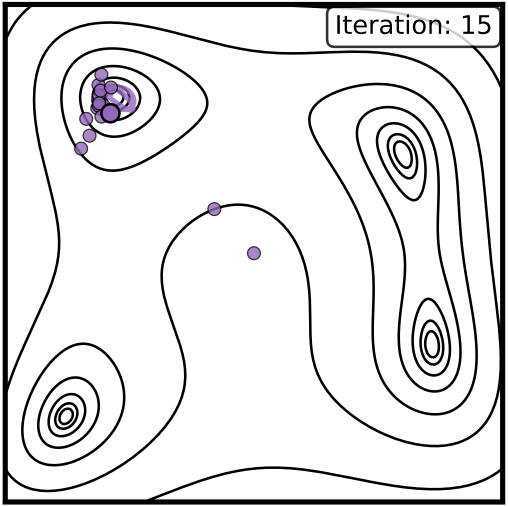
David Yallup

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 [2402.10797]

Hamiltonian Monte Carlo: inference with gradients

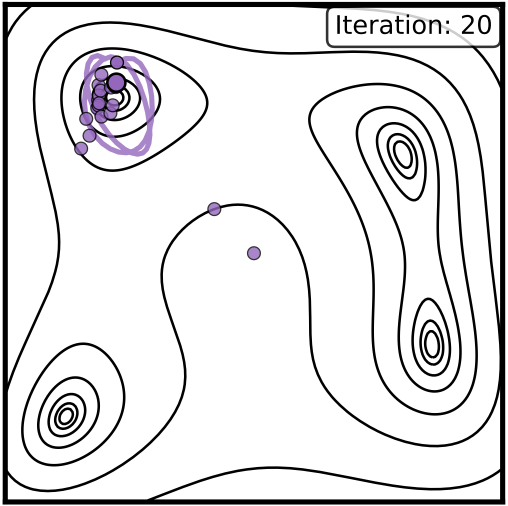
David Yallup

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`.



Classical Algorithms on GPU

Toby Lovick

PhD

CMB Power Spectrum (6 params)

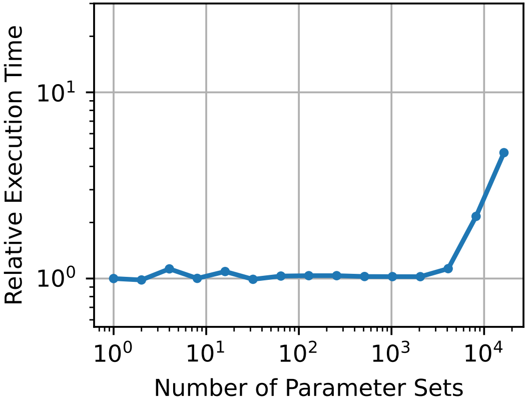
- **PolyChord (CPU)**: 1 hour
- **BlackJAX (GPU)**: 12 seconds

300× speedup

Cosmic Shear (37 params)

- **PolyChord (48 CPUs)**: 8 months
- **NUTS (12 A100 GPUs)**: 2 days
- **BlackJAX (1 A100 GPU)**: 11 hours

Significant improvement over previous GPU methods [2405.12965]



Classical Algorithms on GPU

Toby Lovick

PhD

CMB Power Spectrum (6 params)

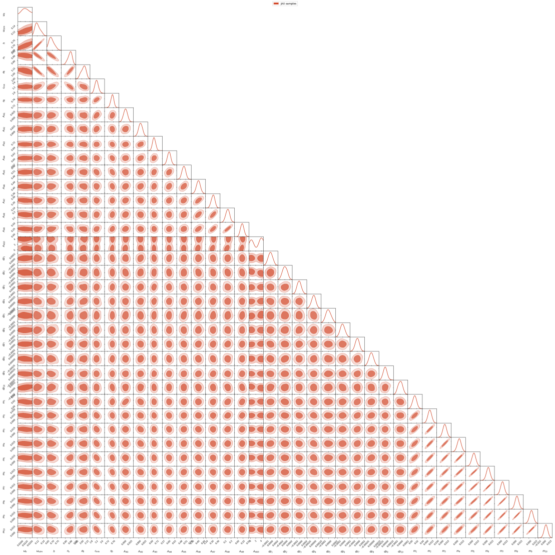
- **PolyChord (CPU)**: 1 hour
- **BlackJAX (GPU)**: 12 seconds

300× speedup

Cosmic Shear (37 params)

- **PolyChord (48 CPUs)**: 8 months
- **NUTS (12 A100 GPUs)**: 2 days
- **BlackJAX (1 A100 GPU)**: 11 hours

Significant improvement over previous GPU methods [2405.12965]



Classical Algorithms on GPU

Toby Lovick

PhD

CMB Power Spectrum (6 params)

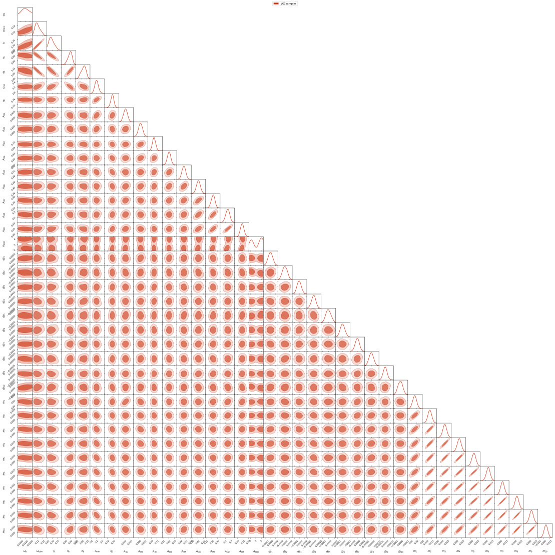
- **PolyChord (CPU)**: 1 hour
- **BlackJAX (GPU)**: 12 seconds

300× speedup

Cosmic Shear (37 params)

- **PolyChord (48 CPUs)**: 8 months
- **NUTS (12 A100 GPUs)**: 2 days
- **Toby's NUTS (1 A100 GPU)**: 6 hours
- **BlackJAX (1 A100 GPU)**: 11 hours

Significant improvement over previous GPU methods [2405.12965]



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× development speedup enables widespread adoption.