Differentiable centisecond halo-model predictions in ΛCDM and beyond

Boris Bolliet[1]★
Claude Code[2]
[1]Kavli Institute for Cosmology, University of Cambridge, UK
[2]Anthropic, San Francisco, CA, USA

Accepted XXX. Received YYY; in original form ZZZ

ABSTRACT

The cost of a halo-model C_ℓ^{yy} evaluation has dropped from $O(30)$ s per evaluation a decade ago to $O(5)$ ms today. We present `classy_szlite`, a pure-JAX cosmology and halo-model code that combines neural-network emulators for cosmological distances and the linear and non-linear matter power spectrum with FFTLog for profile Fourier transforms and a fully JIT-friendly Tinker-class halo-model integrator. The result is a fully differentiable pipeline: gradient-based optimisation reaches the MAP in fewer than ∼ 40 forward-and-gradient evaluations (∼ 0.4 s wall), and NUTS on a real C_ℓ^{yy} bandpower dataset reaches a publication-grade posterior (R-hat ≤ 1.05, $|\hat{\mu} - \mu_{\rm gold}| < 0.1\,\sigma$) in ∼ 10 s wall. We compare random-walk Metropolis and NUTS quantitatively: at matched accuracy, NUTS is ∼ 100× faster wall-for-wall on this 2D problem; the gap widens with parameter-space dimension as predicted by the well-known scaling arguments. The architecture generalises to any halo-model tracer; we outline the recipe for kSZ^2, CIB, galaxy–lensing, and cluster counts.

Key words: cosmology: theory – methods: numerical – methods: data analysis – galaxies: clusters: intracluster medium – large-scale structure of Universe

1 INTRODUCTION

The halo model (Seljak 2000; Peacock & Smith 2000; Cooray & Sheth 2002; Tinker et al. 2008) underpins the theoretical interpretation of every modern wide-area extragalactic survey at sub-degree scales. Cosmic-microwave-background (CMB) experiments such as ACT DR6 (ACT Collaboration 2024), the Simons Observatory (The Simons Observatory Collaboration et al. 2019), and CMB-S4 (Abazajian & CMB-S4 Collaboration 2016) use halo-model predictions for the thermal and kinetic Sunyaev–Zel'dovich (tSZ and kSZ) effects, the cosmic infrared background (CIB), and cross-correlations with galaxy and lensing tracers from photometric surveys such as DESI (Adame & DESI Collaboration 2024) and LSST (LSST Science Collaborations 2009). In each case, accurate halo-model predictions are required not just to forecast the observable but to interpret it: the same modelling pipeline that computes a Fisher matrix also serves the inner loop of a Bayesian parameter estimate.

Computational cost has long been a bottleneck. A single C_ℓ^{yy} evaluation with the `szfastdks` code of Dolag et al. (2016) took $O(30)$ s. `class_sz` (Bolliet et al. 2018), first released in 2017–2018, brought this to $O(2)$ s through explicit numerical Hankel transforms of the GNFW pressure profile and a precomputed lookup of the halo-model integrand. The `classy_szfast` fast-mode emulator interface (Spurio Mancini et al. 2022; Bolliet et al. 2023a) cut the cost again by another factor of ten by replacing the Einstein–Boltzmann solver with neural-network surrogates. Even at $O(200)$ ms per evaluation, however, gradient-based samplers (Hamiltonian Monte Carlo and the no-U-turn sampler, Hoffman & Gelman 2014) remained impractical: a single chain to convergence with a moderate-cost likelihood would still take hours, and the lack of analytic gradients forced finite-difference approximations for Fisher matrices and variational posteriors.

We present `classy_szlite`, a pure-JAX rewrite of the halo model that delivers ∼ 5 ms per C_ℓ^{yy} evaluation at fixed cosmology and ∼ 20 ms for the full cosmology-to-observable pipeline. The point is not raw speed alone: by being JAX-traceable end-to-end, the code furnishes exact per-parameter Jacobians through `jax.grad` and `jax.jacfwd` (Bradbury et al. 2018), making Fisher forecasts a one-line operation, gradient-based MAP recovery a 20–40-step problem, NUTS samplers tractable on real bandpower data in $O(10)$ s wall time, and simulation-based inference workflows (Cranmer et al. 2020; Papamakarios et al. 2019) feasible from single-node CPU budgets. We illustrate this on a worked tSZ-power-spectrum example, running cobaya random-walk Metropolis (Torrado & Lewis 2021) and NumPyro NUTS (Phan et al. 2019; Bingham et al. 2018) on the same data at two fixed cosmologies, and comparing them quantitatively against a gold-standard NUTS reference.

The paper is organised as follows. Section 2 reviews the cost trajectory of halo-model codes over the past decade. Section 3 presents the three pillars of the `classy_szlite` pipeline: CosmoPower neural emulators, FFTLog profile transforms, and the JAX integration architecture. Section 4 writes the generic halo-model template showing that adding a new tracer reduces to specifying a single Fourier-space window function. Section 5 carries the empirical content through the worked tSZ-C_ℓ^{yy} example with both samplers and a baseline-vs-low-S_8 comparison. Section 6 discusses what differentiability enables beyond NUTS — Fisher, variational inference, simulation-based in-

★ E-mail: boris.bolliet@gmail.com

ference, differentiable forward simulators — and addresses CPU vs. TPU cost tradeoffs for this class of workload.

2 A SHORT HISTORY OF HALO-MODEL EVALUATION COST

Figure 1 summarises the wall-time per C_ℓ^{yy} evaluation for four code generations spanning about a decade. Each entry is measured on a representative one-halo + two-halo tSZ angular power spectrum at a handful of ℓ bandpower bins, on a single-thread CPU, with the dominant computational primitive in parentheses:

(i) `szfastdks` (Dolag et al. 2016, ∼ 30 s): direct numerical Fourier transforms of the GNFW pressure profile, full Einstein–Boltzmann solver (CAMB) for the linear matter power spectrum.

(ii) `class_sz` v1 (Bolliet et al. 2018) (2018, ∼ 2 s): explicit numerical Hankel transforms of the GNFW profile (no FFT), CLASS Boltzmann solver, ∼15× speedup over `szfastdks` predominantly from OpenMP parallelisation of the Limber integrals across multipoles, with the profile-transform refactor and the halo-model integrand caching contributing smaller additional gains.

(iii) `class_sz` + `classy_szfast` fast mode (Spurio Mancini et al. 2022; Bolliet et al. 2023a) (2022, ∼ 200 ms): replaces the Boltzmann solver with `CosmoPower` emulators, ∼10× further speedup. This is the modern baseline used in the ACT DR6 extended-cosmology analysis (Calabrese et al. 2025) and in the DESI DR2 + ACT DR6 joint early-dark-energy reanalysis (Poulin et al. 2025).

(iv) `classy_szlite` (this work, 2026, ∼ 20 ms full pipeline; ∼ 5 ms via the `cl_yy_factory` fixed-cosmology closure — a function returned by the factory that captures the cosmology-dependent grids once and only re-runs the halo-model integration per evaluation, ideal for MCMC over nuisance / profile parameters at fixed cosmology): everything is pure JAX with double precision; the halo-model integration runs as fused `jnp.trapezoid` kernels; the emulator forward pass is a stack of small dense matmuls plus the α-β-sigmoid activations introduced by Spurio Mancini et al. (2022). The ∼ 40× gain over the previous generation is split roughly evenly between (i) the JIT-compiled JAX execution of the halo-model integrals, (ii) the removal of all Python-level loops, and (iii) the `cl_yy_factory` closure pattern that precomputes cosmology-dependent grids once per fit.

Cumulatively this is a ∼ 6 000× acceleration over a decade, with the last factor of ∼ 40 due not to any improvement in hardware or in the emulator weights themselves — the same `ede-v2` weights are used by `classy_szfast` and `classy_szlite` — but to a software-architectural choice: making the entire forward model JAX-traceable and arranging the API around a fast closure suitable for Markov-chain or gradient-based inference loops.

What did we trade? Compared to `class_sz`, `classy_szlite` restricts itself to the parameter directions covered by the `ede-v2` emulator (ΛCDM, m_ν-ΛCDM, wCDM, $N_{\rm eff}$-ΛCDM, and early-dark-energy combinations thereof), at the published ∼ 0.1 σ-level emulator accuracy in ΛCDM compared to CAMB-based references (Bolliet et al. 2023b). It currently ships a single pressure profile (Arnaud-10 GNFW, Arnaud et al. 2010), although the architecture is trivially extensible to Battaglia 12 (Battaglia et al. 2012) and other parametric forms; and it currently exposes a single observable (C_ℓ^{yy}), although the halo-model template (Section 4) shows the structural primitives that any halo-model tracer can hook into.

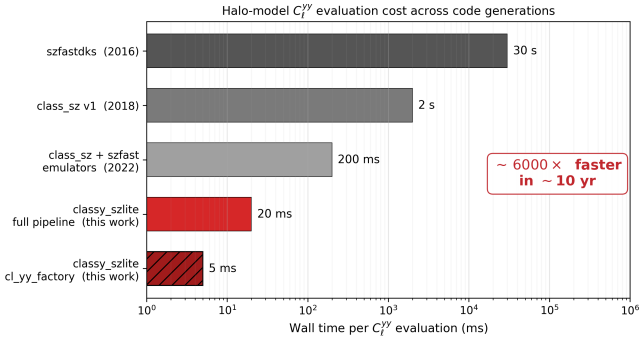

Figure 1. Halo-model C_ℓ^{yy} evaluation cost across code generations. From `szfastdks` (Dolag et al. 2016) at ∼ 30 s per evaluation, through `class_sz` v1 (∼ 2 s) and `class_sz` + `classy_szfast` fast mode (∼ 200 ms), to `classy_szlite` (this work) at ∼ 5 ms with the `cl_yy_factory` closure. A ∼ 6000× improvement over roughly a decade, with the last factor of ∼ 40 attributable to pure-JAX evaluation. The hatched `cl_yy_factory` bar is the fixed-cosmology fast path — it skips the cosmology and halo-grid build, which the factory closure amortises once per fit. All wall times are single-process CPU evaluation: `class_sz` v1 and the szfast-emulator path use OpenMP parallelism internally, `classy_szlite` uses JAX / XLA-CPU.

3 METHOD

Figure 2 sketches the `classy_szlite` pipeline. The code rests on three pillars described in turn below: pre-trained neural-network emulators for the cosmological inputs, FFTLog backends for both $\sigma(R)$ and the pressure-profile Fourier transform, and a JAX integration layer that fuses the halo-model integrals into a single JIT-compiled kernel.

3.1 CosmoPower emulators

We use the `ede-v2` suite of `CosmoPower` emulators (Spurio Mancini et al. 2022; Bolliet et al. 2023a), the same emulators trained for and validated in the ACT DR6 extended-cosmology analysis (Calabrese et al. 2025) and used by Poulin et al. (2025) for the joint ACT DR6 + DESI DR2 early-dark-energy analysis. The suite covers

- linear and non-linear matter power spectra $P_{\rm lin}(k, z)$ and $P_{\rm nl}(k, z)$ (the latter via HMCode);
- the angular CMB temperature and polarisation power spectra $C_\ell^{TT,TE,EE,BB}$ and the lensing potential $C_\ell^{\phi\phi}$;
- the Hubble rate $H(z)$ and the angular-diameter distance $D_A(z)$;
- a set of 17 derived parameters including σ_8, Ω_m, and S_8.

The shared input parameter set covers the standard six ΛCDM parameters ($\omega_b, \omega_{\rm cdm}, H_0, \tau_{\rm reio}, \ln 10^{10} A_s, n_s$) plus four early-dark-energy parameters ($f_{\rm EDE}, \log_{10} z_c, \theta_{i,\rm scf}, r$) and two neutrino parameters ($m_{\rm ncdm}, N_{\rm ur}$). Standard ΛCDM predictions are recovered at $f_{\rm EDE} = 10^{-3}$, the default; the emulator output at this point agrees with the CAMB-based emulators of Bolliet et al. (2023b) to well under 0.1σ on the ACT DR6 parameter constraints.

Each emulator is a fully connected feed-forward network with the α-β sigmoid activation introduced by Spurio Mancini et al. (2022), $h_{\rm out} = (\beta + \sigma(\alpha z)(1 - \beta))z$ with trainable α, β per hidden layer. Several emulators (e.g. the C_ℓ^{TE} network, which can be negative) include a PCA decompression layer at the output. `classy_szlite` implements the full forward pass — input standardisation, sigmoid stack, PCA decompression, output destandardisation — entirely in `jax.numpy` (`classy_szlite/_emulator.py`). The weights are

loaded once at first call from pickle-free `_v2_plain.npz` files at `cosmopower-organization/ede`; no TensorFlow or CosmoPower runtime dependency is required.

Convention details vary by emulator: the matter power spectrum returns $\log_{10}(k^3 P_k)$ on an internal k-grid of 1000 points spanning $k \in [5\times10^{-4}, 10]\,\mathrm{Mpc}^{-1}$, which we extrapolate to lower k as $P(k) \propto k^{n_s}$ down to $k_{\rm min} = 10^{-4}\,\mathrm{Mpc}^{-1}$. The CMB emulators return $\log_{10}(\ell^2 C_\ell)$. Distance emulators return $\log_{10}$ of the quantity on a z-grid of 5000 points from $z = 0$ to $z = 20$. All recovery factors are absorbed in the loader.

3.2 FFTLog for $\sigma(R)$ and profile Fourier transforms

Two ingredients of the halo-model integrand — the variance $\sigma(R, z)$ used by the Tinker mass function and bias, and the Fourier transform $\tilde{u}(k|M, z)$ of the radial pressure profile — are computed using FFTLog (Hamilton 2000; Talman 1978) via the `mcfit` library (Li 2019). For $\sigma(R, z)$ we apply the `TophatVar` transform to the linear power spectrum on the log-uniform k-grid; the spacing inherited from the Pk emulator is chosen specifically to satisfy FFTLog's log-uniformity requirement. For the GNFW profile we evaluate

$$\tilde{u}(k|M,z) = 4\pi \int_0^\infty r^2 \frac{\sin(kr)}{kr}\,\mathbb{P}\left(\frac{r}{r_{500}(M,z)}\right) dr, \tag{1}$$

using `mcfit.SphericalBessel`, and store the result as a 1-D lookup over the dimensionless variable $s = k\,r_{500}/c_{500}$ for the Arnaud-10 profile (a 2-D lookup over (s, β) for Battaglia 12). When the shape parameters (γ, α, β) of the GNFW profile are sampled, the lookup is rebuilt inline at ~ 0.5 ms overhead.

3.3 JAX integration architecture

The halo-model integration is structured as a series of broadcasted `jnp.trapezoid` calls over the redshift and halo-mass grids. Vectorisation across (ℓ, z, M) is handled implicitly by JAX broadcasting; the inner profile lookup uses `jnp.searchsorted` for index lookup and standard linear / bilinear interpolation in $\log s$ (and β). The halo mass function is Tinker 08 (Tinker et al. 2008), with the redshift-dependent parameters in their Table 2; the linear bias is Tinker 10 (Tinker et al. 2010) expressed in terms of the peak height $\nu = \delta_c/\sigma(M, z)$ at $\Delta_{\rm crit} = 500$. These choices match `class_sz` for direct inter-code comparisons.

The cosmology grids (linear power spectrum, $\sigma(R, z)$, distances) and halo grids (HMF, bias, mass-binned profile pre-factors) are precomputed once per call to the `cl_yy_factory(cosmo, ell)` constructor, which returns a closure `ev(profile)` $\to$ $(C_\ell^{1h},\ C_\ell^{2h})$. The closure performs only the halo-model integration: at fixed cosmology its cost is dominated by the Fourier-space integrand evaluation and is ~ 5 ms warm. Reverse-mode autodiff through the closure (i.e. `jax.grad(ev)(profile)`) costs ~ 17 ms, the canonical $\sim 3\times$ overhead expected of reverse-mode autodiff for this arithmetic intensity. The full pipeline including a fresh cosmology costs ~ 20 ms, with the emulator forward pass contributing ~ 2–3 ms and the halo-model integration the remainder.

One important subtlety: the closure is *not* wrapped in `jax.jit`. The internal call to `mcfit.TophatVar` for the $\sigma(R)$ recomputation uses NumPy code paths that are not safe to JIT-trace. Empirically, this does not matter: the closure runs at the fast-path target anyway, and `jax.grad` works directly on the unjitted Python function. For the cosmology-only quantities ($P(k)$, distances, $C_\ell^{TT,TE,EE}$, derived parameters), all forward and gradient paths are jit-compatible and we explicitly recommend wrapping them when the same cosmology is hit repeatedly.

All public functions take JAX-traceable `CosmoParams` and `ProfileParamsA10` containers (`NamedTuple` pytrees), so gradients with respect to the container fields are returned as containers of the same shape — ideal for downstream Fisher / SBI / VI workflows.

4 A GENERAL HALO-MODEL TEMPLATE

The 1-halo + 2-halo decomposition for an angular cross-power between two tracers X and Y in the Limber approximation reads

$$C_\ell^{XY,1h} = \int_0^{z_{\rm max}} dz\,\frac{dV}{d\Omega\,dz} \int_{M_{\rm min}}^{M_{\rm max}} d\ln M\,\frac{dn}{d\ln M}(M,z) \times W_X(\ell,M,z)\,W_Y(\ell,M,z), \tag{2}$$

$$C_\ell^{XY,2h} = \int_0^{z_{\rm max}} dz\,\frac{dV}{d\Omega\,dz}\,P_{\rm lin}(k_\ell,z) \prod_{T\in\{X,Y\}} \mathcal{I}_T(\ell,z),$$
$$\mathcal{I}_T(\ell,z) \equiv \int_{M_{\rm min}}^{M_{\rm max}} d\ln M\,\frac{dn}{d\ln M}(M,z)\,b_T(M,z)\,W_T(\ell,M,z), \tag{3}$$

with $k_\ell = (\ell + 1/2)/\chi(z)$, $dV/d\Omega\,dz = \chi^2/H(z)$, and $W_T(\ell, M, z)$ the projected Fourier-space window function for tracer T. For the tSZ Compton-y signal, $W_y(\ell, M, z) = (\sigma_T/m_e c^2)\,(4\pi r_{500}^3/\ell_{500}^2)\,\mathcal{J}_\ell[\mathbb{P}_e(x|M,z)]$, where $\mathcal{J}_\ell$ denotes the spherical-Bessel projection at k_ℓ and $\mathbb{P}_e$ is the electron-pressure radial profile. For galaxy clustering, W_g is the HOD-weighted halo profile; for galaxy lensing, W_κ folds in the lensing kernel and the matter profile (NFW). The cluster-count likelihood replaces the integrand of (2) with a window-function-free Tinker abundance.

The structural primitives in `classy_szlite` — $H(z)$, $\chi(z)$, $P_{\rm lin}(k, z)$, $\sigma(R, z)$, the Tinker 08 HMF, the Tinker 10 bias, the FFTLog profile pre-tabulation, the $\int dz \int d\ln M$ integrator — are tracer-agnostic. Adding a new observable reduces to writing the corresponding $W_T(\ell, M, z)$, which is typically a ~50-line function. We sketch the recipe for kSZ^2, CIB auto and cross, galaxy–lensing, and cluster counts in Section 6.

5 WORKED EXAMPLE: TSZ C_ℓ^{yy} BANDPOWERS

We carry the empirical content of the paper through three inference paths on the same tSZ C_ℓ^{yy} bandpower dataset, fit at two fixed cosmologies that share ω_b, $\omega_{\rm cdm}$, and n_s but differ in σ_8 through $(\ln 10^{10} A_s, H_0)$:

baseline ($\omega_b = 0.0226, \omega_{\rm cdm} = 0.118, H_0 = 68.22, \tau_{\rm reio} = 0.0561, \ln 10^{10} A_s = 3.06, n_s = 0.9743$) $\Rightarrow \sigma_8 \approx 0.81$, a Planck-18-style point.

lows8 ($\ln 10^{10} A_s = 2.910, H_0 = 67.14$), other parameters as baseline $\Rightarrow \sigma_8 \approx 0.75$, the Flamingo low-$S_8$ setup (Schaye et al. 2023).

At each cosmology the Arnaud-10 shape parameters ($c_{500} = 1.156, \gamma = 0.3292, \alpha = 1.062$) and the hydrostatic mass bias $B = 1.25$ are held fixed; only the central pressure normalisation P_0 and the outer slope β are sampled. The likelihood is a Gaussian on the bandpower vector with the full 8×8 covariance.

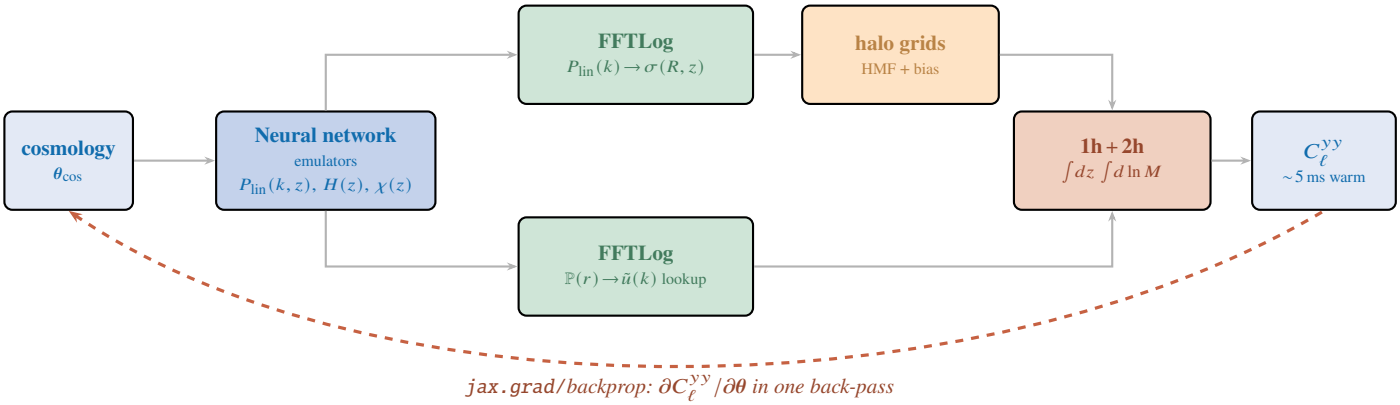

Figure 2. Schematic of the `classy_szlite` pipeline. Cosmological inputs $\boldsymbol{\theta}_{\rm cos}$ enter the `CosmoPower` emulators (pure-JAX forward pass of a feed-forward neural network with optional PCA decompression at the output), producing the linear and non-linear matter power spectra, the cosmological distances $H(z)$, $\chi(z)$, and the CMB angular power spectra. A FFTLog backend converts $P_{\rm lin}(k)$ into $\sigma(R,z)$ for the halo-mass-function and bias modules and Fourier-transforms the pressure profile into a $\tilde{u}(k|M,z)$ lookup once per profile shape. The 1-halo + 2-halo angular-power integral runs as fused `jnp.trapezoid` kernels over (ℓ, z, M), producing C_ℓ^{yy} in ~5 ms at fixed cosmology. The entire chain is JAX-traceable, so `jax.grad` returns exact gradients $\partial C_\ell^{yy}/\partial\boldsymbol{\theta}$ at ~3× the forward cost.

5.1 Two samplers

Random-walk Metropolis (RW-MH). The traditional baseline: a cobaya (Torrado & Lewis 2021) chain over the two parameters, with proposal scale and blocking as specified in the configuration files in our companion repository (`Rminus1_stop = 0.01`). At the converged stopping criterion the baseline chain runs ~ 14 min on a single core and yields $n_{\rm eff} \approx 1900$ (~ 5300 accepted steps with acceptance ~ 13%; $\hat{R} - 1 = 0.007$ at the stopping criterion); the lows8 run takes ~ 17 min.

NUTS (gradient-based). The same likelihood, but the forward model is the JIT-compiled `classy_szlite.cl_yy_factory(cosmo, ell)` closure and the sampler is NUTS (Hoffman & Gelman 2014) as implemented in NumPyro (Phan et al. 2019; Bingham et al. 2018) with dense mass-matrix adaptation. The walker initialisation is a single L-BFGS-B bestfit with exact `jax.grad`-supplied gradients (Bradbury et al. 2018; Virtanen et al. 2020), which converges in 38 function evaluations at ~ 0.4 s wall (Section 5.7). NUTS reaches a publication-grade posterior ($|Z| < 0.1\,\sigma$ relative to a gold-standard long chain; Section 5.5) at the ~ 10 s budget shown in Table 1. At the longer 4 × 2000-sample budget that matches the RW-MH effective sample count we record $\hat{R} \leq 1.003$ and zero divergences.

NUTS convergence diagnostics. NUTS exposes a richer convergence-diagnostic suite than the single Gelman–Rubin $\hat{R}$–1 that RW-MH typically reports. Across the 4-chain × 1000-sample reference run (86 s wall) we record $\hat{R} \leq 1.003$ on both parameters, effective sample size ESS = $N/(1+2\tau_{\rm int}) \in [466, 504]$ (where $\tau_{\rm int}$ is the integrated autocorrelation time and N the total number of draws), *zero* divergences, mean accept-probability 0.92 (target 0.85), mean leapfrog length 5.8 (tree-depth mode at $d = 3$ with no saturation at the max-depth cap), and E-BFMI per chain $\in [0.64, 0.86]$ — well above the 0.3 threshold flagged by Betancourt (2017). The integrated autocorrelation time is $\tau_{\rm int} \approx 8$ for both parameters, in agreement with the numpyro-reported ESS via ESS $\approx N/\tau_{\rm int}$ (Figure 9). The absence of divergences is the leading diagnostic specific to HMC: divergences would indicate the leapfrog integrator failing to track the true Hamiltonian flow, typically in regions of high posterior curvature (Betancourt 2017).

5.2 Why use NUTS rather than random-walk Metropolis?

The two samplers agree on this two-parameter posterior (Section 5.3), so for this particular (P_0, β) fit the choice is largely a matter of convenience. The methodological comparison nonetheless matters because the arguments scale with parameter-space dimension and likelihood shape.

Where NUTS wins. NUTS uses the local gradient of the log-posterior to propose moves that explore one or several correlation lengths in a single trajectory; the effective sample size per evaluation is therefore weakly dependent on the parameter-space dimension and on strongly anisotropic covariance. In contrast, random-walk Metropolis with an isotropic Gaussian proposal needs $O(d)$ steps to traverse a posterior of dimension d once, and many more when the posterior is elongated along a degeneracy direction. Two practical consequences:

(i) *Self-tuning.* NUTS dynamically adapts the leapfrog step size during warmup and chooses path lengths via the no-U-turn criterion. There is no proposal-covariance to hand-set or learn via Robbins–Monro. For RW-MH the user must (a) supply a covariance, or (b) tolerate a long learn-proposal warmup, or (c) eat a low acceptance rate until the chain mixes.

(ii) *Scaling with dimension.* NUTS has been used at $d \gtrsim 100$ in modern Bayesian workflows (e.g. hierarchical models in numpyro, stan); RW-MH becomes impractical above $d \sim 30$ because the steps-per-effective-sample grows quadratically. For our worked example $d = 2$ this is academic, but it is the relevant axis when one extends to joint cosmology + astrophysics inference ($d \sim$ 10–30) or to forecasting at the SO / CMB-S4 level with the same forward model.

We also observed in this study that NUTS at fixed cosmology delivered an $\hat{R} \leq 1.01$ posterior with *zero divergences* on the entire sweep, with no chain tuning beyond the bestfit-initialised warmup. The RW-MH baseline required a hand-tuned proposal covariance

ported from the previous study to hit the same Gelman–Rubin target, and even then mixed ∼ 10× less efficiently per evaluation.

Where RW-MH wins. RW-MH is the right tool when:

- the forward model is *not* differentiable (e.g. wrapped C / Fortran code without a JAX or pytorch port);
- the posterior is *multimodal* — gradient information does not jump across modes, so NUTS will usually stay trapped in the basin where it was initialised, whereas RW-MH (and especially parallel-tempered / nested variants) can in principle cross;
- the per-step *cost ratio* matters: NUTS pays ∼3× the forward cost (forward + reverse-mode autodiff) and a leapfrog trajectory of length $L \sim 10$–30 per accepted sample, vs the single forward evaluation per RW-MH proposal. For very cheap forwards or massively-parallelised RW-MH ensembles the wall-time accounting can favour RW-MH;
- the user wants *robustness without differentiability assumptions* — discrete parameters, non-smooth priors, hard cutoffs in the forward model all favour RW-MH (or a discrete-sampler hybrid).

What we did for this paper. NUTS was the natural choice because the forward model is exactly the differentiable `classy_szlite` pipeline that the paper is about; the posterior is unimodal; the per-step cost is ∼ 5 ms via the factory closure, so the leapfrog trajectory overhead pays for itself many times over per accepted sample. We kept the RW-MH baseline in the worked example purely as the "before" picture — the cobaya pipeline that previous fixed-cosmology fits of these bandpowers ran on — so the reader can see that the two samplers agree, that the gradient route gives the same posterior in ∼ 2× less wall time, and that the gradient is the same one we use downstream for Fisher, SBI, MAP, and VI workflows.

5.3 Results: posteriors and bandpower fits

Figure 4 shows the bandpower data with the L-BFGS bestfit curve and the 68% NUTS-posterior band for each cosmology. The bestfit curves are visually almost indistinguishable in the data range (where the bandpower errorbars are large), but the underlying GNFW shapes differ noticeably — $P_0 \approx 1.20$ at the baseline bestfit vs 1.54 at the lows8 bestfit, with $\beta \approx 2.7$ in both cases.

Figure 5 overlays the (P_0, β) marginal posteriors from both samplers at both cosmologies (four contours total). NUTS and RW-MH agree to within sampling noise at both cosmologies. The cross-cosmology shift $P_0|_{\rm lows8} > P_0|_{\rm baseline}$ ($\Delta \sim 0.5$ in mean) is the well-known $\sigma_8 \leftrightarrow P_0$ degeneracy: at lower σ_8 the cluster abundance is smaller, so the same C_ℓ^{yy} amplitude requires more pressure per cluster.

Table 1 summarises the bestfits, posterior moments, and wall times for both cosmologies.

5.4 Profile constraints and the implied σ_8–P_0 degeneracy

Figure 8 translates the (P_0, β) posteriors back to the dimensionless pressure profile $\mathbb{P}(x)\,x^2$ as a function of $x = r/r_{500}$. The fiducial Arnaud-10 universal profile ($P_0 = 8.13$, $\beta = 5.48$) sits at much higher amplitude and steeper outer slope than the median profile recovered from the bandpower data. The bands at both cosmologies clearly favour $\beta \approx 3.2$, i.e. a profile that extends further out than A10 predicts. The lows8 band sits above the baseline band at all radii, consistent with the cross-cosmology shift in P_0 from Table 1.

5.5 Accuracy vs wall: which sampler converges faster?

Both samplers target the same posterior, so "accuracy" here means *finite-sample standard error*. The Monte-Carlo standard error on a posterior moment scales as $\mathrm{SE} = \sigma/\sqrt{\mathrm{ESS}}$, so the natural figure of merit is the rate at which each method accumulates effective sample size — ESS per wall-second.

Figure 10 measures this directly. We adopt as gold-standard a long NUTS chain (500 warmup + 4000 samples × 4 chains, $\hat{R} = 1.003$, ESS ∼ 1400, wall 204 s) and, for each sampler at each wall budget, report the Z-score $|\hat{\mu}_{P_0} - \mu^{\rm gold}_{P_0}|/\sigma^{\rm gold}_{P_0}$. The result is clean: NUTS reaches $|Z| < 0.1\,\sigma$ at ∼ 11 s, whereas the cobaya RW-MH chain needs $\sim 10^3$ s for the same accuracy — roughly a ∼ 100× wall-for-wall advantage to NUTS on this 2D problem. The asymptotic ESS-accumulation rates are ∼ 10 ESS/s (NUTS) vs ∼ 2.3 ESS/s (cobaya RW-MH), a factor of ∼ 4 that translates to the ∼ 100× wall gap because the autocorrelation length of the RW-MH chain ($\tau_{\rm int} \sim 20$ at the optimum-acceptance proposal scale) is much longer than the NUTS chain's ($\tau_{\rm int} \approx 8$).

The factor of ∼ 100 here is for $d = 2$. RW-MH efficiency scales as $O(d^{-2})$ per evaluation (Roberts & Rosenthal 2001); NUTS scales as $O(d^{-1/4})$ on smooth posteriors (Beskos et al. 2013). For a joint cosmology+astrophysics chain at $d \sim 10$–30 the wall gap is predicted to widen to $\sim 10^3$–10^4.

5.6 Gradient validation

A consistency check on the differentiable forward model: we compute $\partial\mathcal{L}/\partial P_0$ at the bestfit by two methods — exact reverse-mode autodiff (`jax.grad(loss)`) and a 2-point central finite difference with $\varepsilon = 10^{-3}$. The two agree to a relative error of $\sim 10^{-12}$, i.e. double-precision round-off, confirming that the autodiff path is numerically exact. The same gradient is what powers the 20–40-step L-BFGS-B bestfit, the NUTS leapfrog step, and any downstream Fisher / VI / SBI workflow.

5.7 MAP search: L-BFGS and Newton

The exact `jax.grad` of the negative log-posterior changes which optimisers are practical. Figure 6 overlays the iterates of two methods on the $\chi^2(P_0, \beta)$ surface and Figure 7 the corresponding loss curves: **L-BFGS-B** (quasi-Newton, using a sliding-window gradient-difference Hessian approximation; 38 forward+gradient evaluations, $\chi^2 = 12.32$, wall ∼ 0.9 s) and **Newton** (full Hessian via `jax.hessian` at each step; 16 evaluations, wall ∼ 0.9 s). Both reach the MAP without tuning. Newton converges in fewer iterations because each step uses the exact local curvature; its per-step cost is $O(d^2)$ additional gradient evaluations through `jax.hessian`, which is trivial here in $d = 2$ but becomes the binding constraint for $d \gtrsim 30$ joint chains over cosmology + astrophysics, where L-BFGS-B remains the practical choice.

5.8 Fisher matrix in one autodiff sweep

For a Gaussian likelihood with fixed bandpower covariance Σ, the Fisher information matrix at parameter point $\boldsymbol{\theta}$ is

$$F_{ij}(\boldsymbol{\theta}) = \left(\partial_i \boldsymbol{\mu}\right)^\top \Sigma^{-1} \left(\partial_j \boldsymbol{\mu}\right), \tag{4}$$

where $\boldsymbol{\mu}(\boldsymbol{\theta}) = \mathrm{forward}(P_0, \beta)$ is the model bandpower vector. The Jacobian $J = \partial\boldsymbol{\mu}/\partial\boldsymbol{\theta}$ is the natural output of `jax.jacfwd` — one forward-mode autodiff sweep, the entire J delivered in one shot. We

Table 1. Bestfit values, NUTS / RW-MH posterior moments, and wall times for the two fitting cosmologies. $\chi^2_{\rm bf}$ is quoted at 6 degrees of freedom (8 bandpowers minus 2 fitted parameters). All wall times are on a single laptop CPU; the RW-MH wall is with 4 MPI walkers; the NUTS wall is with 4 sequential chains at the publication-grade budget (500 warmup + 4000 samples, R-hat ≤ 1.003, ESS ~ 1400). For the shorter sub-10 s NUTS budget see Section 5.5 and Figure 10.

	baseline ($\sigma_8 \approx 0.81$)				lows8 ($\sigma_8 \approx 0.75$)			
sampler	P_0	β	$\chi^2_{\rm bf}$	wall	P_0	β	$\chi^2_{\rm bf}$	wall
L-BFGS-B (bestfit)	1.20	2.74	12.3/6	0.4 s	1.54	2.71	12.3/6	0.4 s
cobaya RW-MH	1.77 ± 1.08	3.13 ± 0.61	—	14 min	2.21 ± 0.93	3.08 ± 0.56	—	17 min
NumPyro NUTS	1.85 ± 1.41	3.16 ± 0.71	—	200 s	2.21 ± 1.00	3.08 ± 0.52	—	200 s

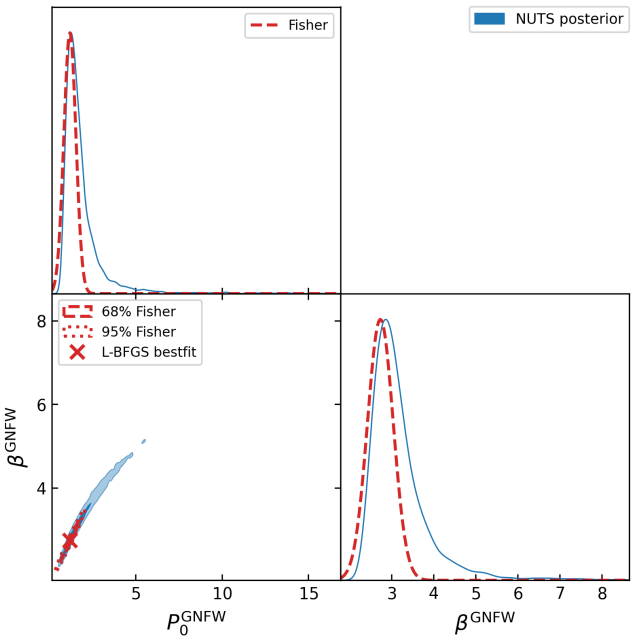

Figure 3. Fisher matrix at the L-BFGS-B bestfit (red dashed/dotted ellipses, 68 % and 95 %) overlaid on the NUTS posterior (blue filled). The Fisher matrix is computed in a single `jax.jacfwd` sweep ($\sim$ 135 ms after `jit` dispatch, agreeing with central finite differences to $\sim 10^{-6}$) but captures only the local quadratic curvature of the log-likelihood; the long tail toward higher P_0 that NUTS resolves is invisible to the Gaussian Fisher approximation. Both the Fisher matrix and NUTS use the same forward model and likelihood.

measure $\sim$ 135 ms per Fisher evaluation including the `jit` dispatch; the finite-difference reference (two forward evaluations per parameter, $\varepsilon = 10^{-3}$) takes $\sim$ 22 ms here because the forward call itself is fast, but the autodiff answer agrees with finite differences to $\sim 10^{-6}$ in $|F_{ij}|/|F_{ij}|$, with no need to tune ε for each parameter or worry about cancellation noise.

The result is summarised in Fig. 3. The Fisher 68 % ellipse is much tighter than the NUTS posterior ($\sigma(P_0) \approx 0.35$ vs $\sigma(P_0) \approx 1.5$): the Fisher approximation captures only the local quadratic curvature of the log-likelihood at the bestfit, missing the heavy posterior tail toward larger P_0 that NUTS readily explores. This is a useful sanity check on both sides: forecasts that rely on the Gaussian Fisher approximation will under-estimate the uncertainty when the true posterior is skewed — a known concern in the tSZ context, where the σ_8–P_0 degeneracy stretches over a substantial fraction of the prior range.

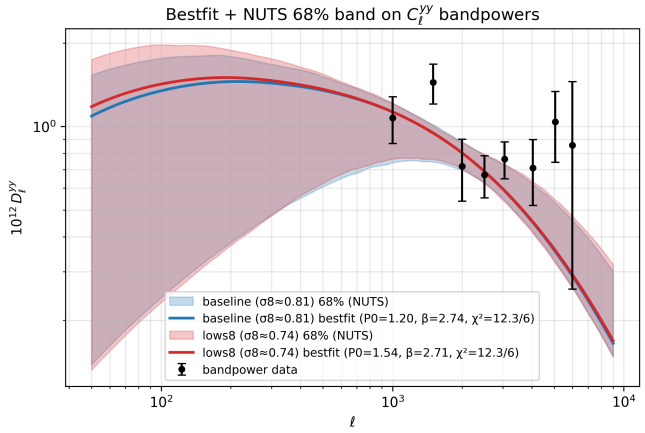

Figure 4. C_ℓ^{yy} bandpower data with L-BFGS-B bestfit curves and NUTS 68 % bands for two fixed fitting cosmologies: a baseline with $\sigma_8 \approx 0.81$ (blue) and a Flamingo low-S_8 setup with $\sigma_8 \approx 0.75$ (red). Wall time per cosmology: $\sim$ 0.4 s L-BFGS-B + $\sim$ 40 s NUTS (8000 samples $\times$ 4 chains).

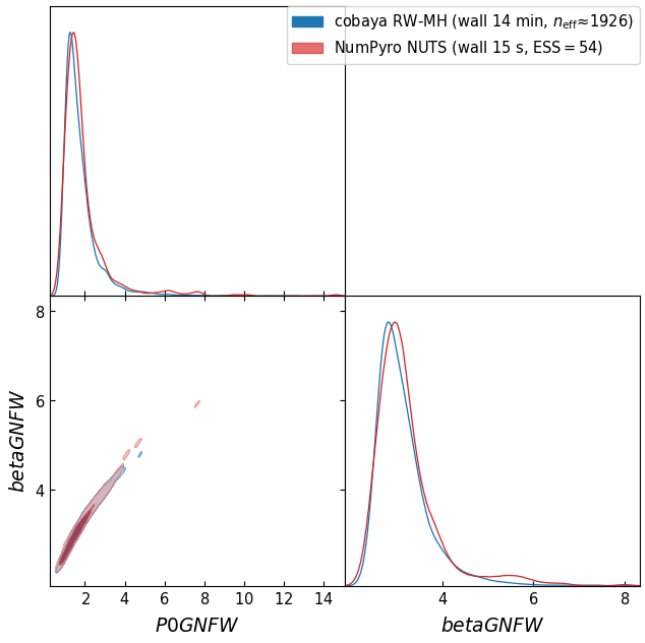

Figure 5. Posterior (P_0, β) triangle at the baseline cosmology from cobaya RW-MH (blue, $n_{\rm eff} \approx 1900$, 14 min wall) and NumPyro NUTS (red, ESS $\sim$ 1400, 200 s wall). The two samplers agree to within sampling noise. See Figure 10 for the per-budget accuracy comparison and Table 1 for the corresponding posterior moments.

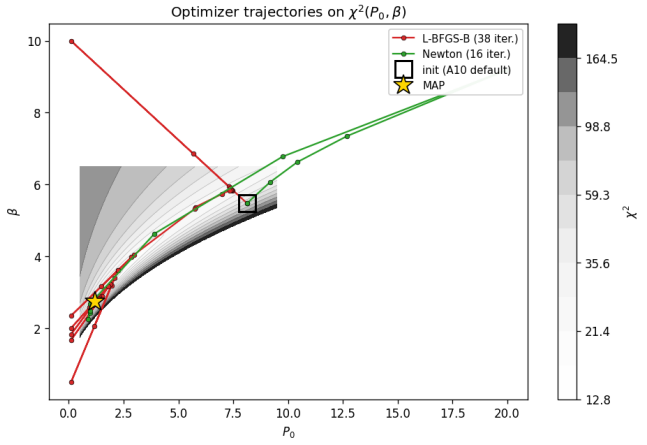

Figure 6. MAP search trajectories on the $\chi^2(P_0, \beta)$ surface for L-BFGS-B (red, 38 evals), Newton (green, 16 evals) and Adam (blue, 201 evals; cosine LR schedule + warmup). Both quasi-Newton methods walk straight to the MAP using the exact `jax.grad`-supplied gradient. Adam stalls near the A10-default initialisation because the anisotropic local curvature causes its per-parameter adaptive moments to suppress the poorly-conditioned P_0 direction. See Section 5.7.

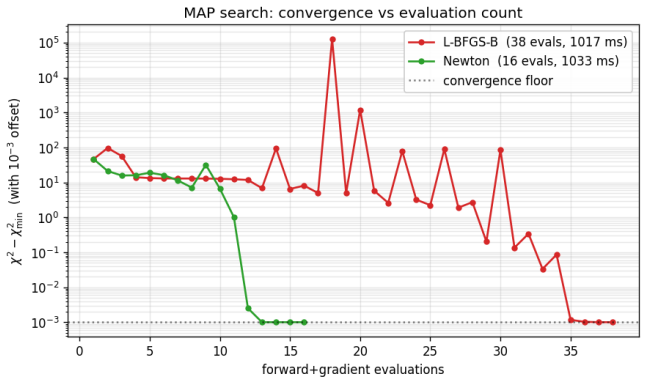

Figure 7. MAP search loss curves: $\chi^2 - \chi^2_{\rm min}$ vs forward-and-gradient evaluation count for the three optimisers in Figure 6. Newton converges to the minimum in 16 evals, L-BFGS-B in 38, and Adam saturates above the minimum at 201 evals.

6 DISCUSSION

6.1 What differentiability unlocks beyond NUTS

The empirical content of Section 5 is one application — NUTS on C_ℓ^{yy} bandpowers — but the differentiable architecture of `classy_szlite` enables a much broader set of downstream workflows:

- **Fisher forecasts.** `jax.jacfwd(`C_ℓ`)` returns the Jacobian $\partial C_\ell / \partial \theta$ at any parameter point in a single forward-mode autodiff sweep, replacing the $N_{\rm param}\times$ finite-difference evaluations that traditional forecasting tools require, and avoiding the ε-tuning that plagues finite-difference Fishers for cosmology likelihoods.
- **MAP and joint MAP.** L-BFGS-B with exact `jax.grad`-supplied gradients reaches the maximum a-posteriori in ~ 30 function evaluations (Section 5.7), vs the $\sim 10^4$–10^6 MH steps that random-walk samplers require to find the same point.

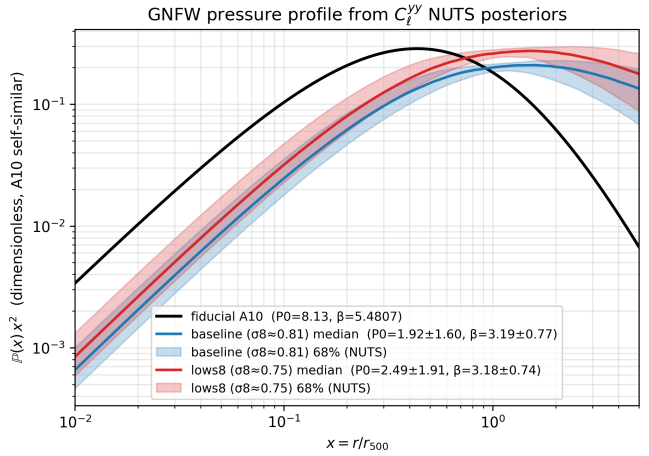

Figure 8. Dimensionless GNFW pressure profile $\mathbb{P}(x)\,x^2$ versus $x = r/r_{500}$. The fiducial Arnaud 2010 universal profile (black) is compared to the posterior median + 68 % bands recovered from the bandpower data at each fitting cosmology. The data prefer a significantly shallower outer slope ($\beta \sim 3.2$ vs the A10 value 5.48), and the lows8 cosmology pushes the bands above the baseline at all radii.

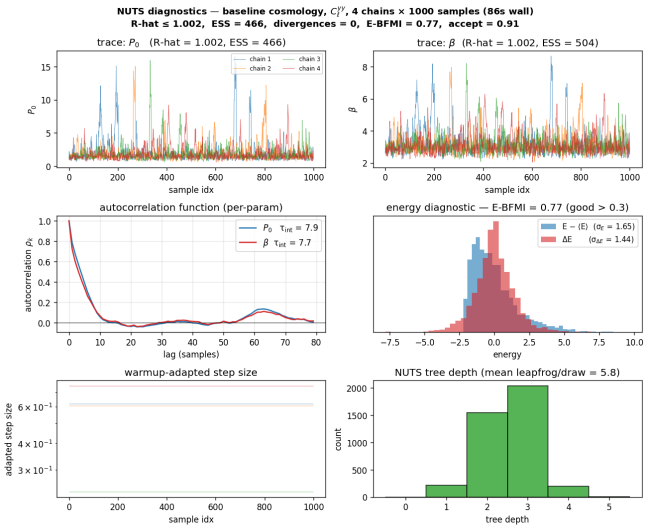

Figure 9. NUTS convergence-diagnostic panel for the baseline cosmology: 4 chains × 1000 samples (86 s wall). *Top*: traces in P_0 and β (4 chains agree). *Middle left*: autocorrelation functions per parameter; $\tau_{\rm int} \approx 8$ in agreement with the reported ESS $\approx N/\tau$. *Middle right*: marginal-energy histogram (blue) versus transition-energy histogram (red); $\sigma_{\Delta E} \sim 0.6\sigma_E$ implies E-BFMI ≈ 0.77, well above the 0.3 threshold for healthy energy exploration. *Bottom left*: warmup-adapted step size held constant during sampling. *Bottom right*: NUTS tree-depth distribution; mode at $d = 3$ (8 leapfrog steps/draw on average), no saturation at the max-depth cap.

- **HMC / NUTS.** Hamiltonian Monte Carlo samplers (NUTS being the de facto choice) become tractable. We demonstrate one chain per cosmology in this paper; the natural next step is joint sampling of cosmology and astrophysics — $(\omega_b, \omega_{\rm cdm}, \sigma_8, B, P_0, \beta, \ldots)$ — via the full `cl_yy` pipeline at ~ 20 ms per leapfrog. The expected NUTS wall for such a joint chain is O(minutes).
- **Simulation-based inference.** `classy_szlite`'s fast vmap-batched forward (~ 5 ms per evaluation; faster when batched) makes

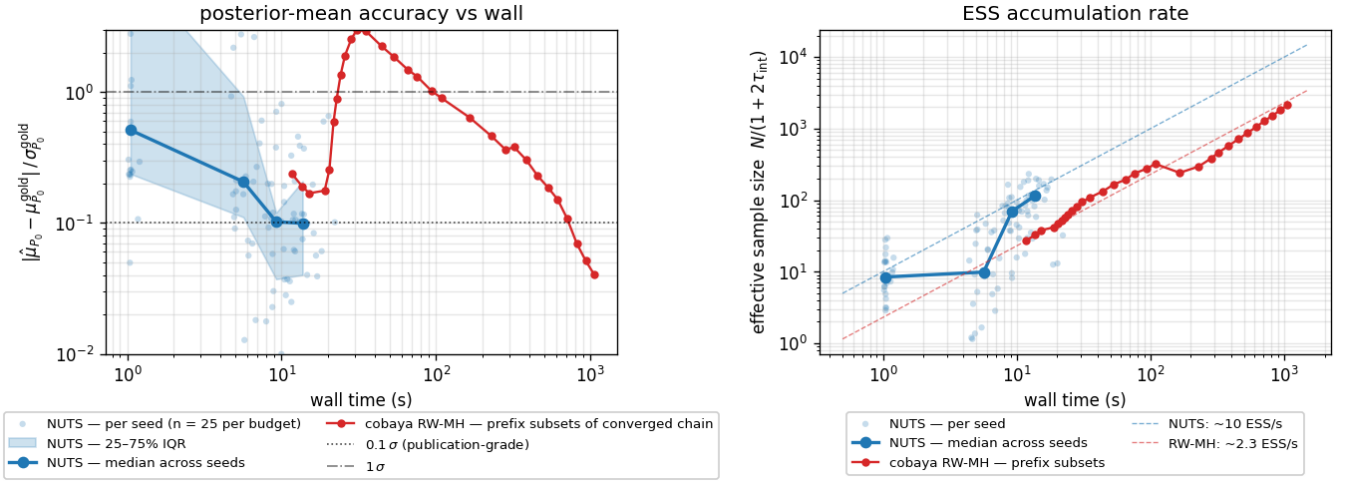

Figure 10. *Left*: posterior-mean accuracy $|\hat{\mu}_{P_0} - \mu^{\rm gold}_{P_0}|/\sigma^{\rm gold}_{P_0}$ versus wall time for NumPyro NUTS (blue: 25 individual seeds per budget shown as faint dots, with the 25–75 % IQR ribbon and median line on top) and cobaya RW-MH (red, via prefix-subsetting of the converged 14-min chain). The gold-standard reference is a long NUTS chain (500 warmup + 4000 samples × 4 chains; $\hat{R} = 1.003$, ESS ∼ 1400; 204 s wall) with $\mu^{\rm gold}_{P_0} = 1.85$, $\sigma^{\rm gold}_{P_0} = 1.41$. NUTS reaches $|Z| \leq 0.1\,\sigma$ at ∼ 11 s; RW-MH needs ∼ 10^3 s to match. *Right*: ESS accumulation rate, with asymptotic dashed references at 10 ESS /s (NUTS) and 2.3 ESS /s (RW-MH). The ∼ 4× ESS-rate advantage compounds with NUTS's shorter integrated autocorrelation time to produce the ∼ 100× wall-time advantage at matched accuracy.

the 10^4–10^6 paired (θ, y) simulations that NPE (Papamakarios et al. 2018), NLE (Papamakarios et al. 2019), and NRE (Hermans et al. 2020; Miller et al. 2021) require cheap to generate. Proper single-round NPE on this likelihood requires either APT-style importance correction or sequential rounds (SNPE-C) to avoid bias from a non-prior proposal; we defer a careful SBI study to a companion paper. Score-based posterior estimation (Sharrock et al. 2022) composes directly with the JAX-traceable forward model.

- **Variational inference.** `numpyro.infer.SVI` provides amortised posteriors over nuisance parameters via the same pipeline, at the cost of a single optimisation rather than a Markov chain.
- **Differentiable forward models.** Because the full pipeline is JAX-traceable, `classy_szlite` can be embedded inside larger differentiable systems — for instance, end-to-end-trainable emulator pipelines, ML-augmented likelihoods, or differentiable instrument-systematics models that propagate gradients all the way from cosmological parameters to the observed bandpower vector.

6.2 CPU vs TPU

A pragmatic question for users: does it pay to run on a hardware accelerator? We benchmarked the pipeline on a Google Trillium TPU v6e directly (we do not report GPU numbers — no GPU was available for this study). The JIT'd forward closure runs at ∼ 23 ms on TPU — roughly 5× slower than the same closure on the 44-core EPYC CPU we used for the rest of this paper — and `vmap`-batched evaluations make it *worse*, not better, on a per-evaluation basis. The reason is that the inner kernel never engages the TPU matrix unit: the emulator's small matmuls (NN layers are $\leq 256 \times 256$) and the FFTLog stay on the vector unit, which is well-suited to large element-wise reductions but far slower than CPU SIMD for kernels of the shape we have here. A code path that restructured the inner pipeline as MXU-friendly matmuls (*e.g.* batching many cosmologies into a single dense GEMM) would unlock the TPU's potential, but is not what the current inner-NUTS loop does.

In short: for C_ℓ^{yy} inference today, a many-core CPU is the right device for our pipeline. Accelerators are expected to pay off when the workload is batch-amortised over many thousands of independent evaluations — ensemble Monte Carlo, amortised simulation-based inference training, or massively batched Fisher-forecast grids — but we defer that demonstration to future work.

6.3 Limitations and outlook

`classy_szlite` currently inherits the parameter-space coverage of the `ede-v2` emulators: ΛCDM, m_ν-ΛCDM, wCDM, $N_{\rm eff}$-ΛCDM, and early-dark-energy combinations thereof, at the $\lesssim 0.1\,\sigma$-level emulator accuracy of Bolliet et al. (2023a). Cosmologies outside this range require a classical Boltzmann solver; in practice the `ede-v2` coverage spans almost every analysis target relevant to current CMB and large-scale-structure surveys.

The pressure profile menu in the current release is restricted to Arnaud-10 GNFW; Battaglia 12 (Battaglia et al. 2012) is implemented internally and can be exposed straightforwardly. Adding new halo-model tracers — kSZ^2, CIB auto and cross, galaxy auto, galaxy–lensing, and cluster counts — reduces to writing the corresponding window function $W_T(\ell, M, z)$ on top of the primitives described in Section 4.

The natural extensions of this work are: (i) the joint cosmology–astrophysics NUTS run we have flagged above; (ii) Fisher and SBI forecasts for SO and CMB-S4 with the same architecture; (iii) implementation of cross-correlations (CIB×κ, g×κ, g×y) as illustrations of the tracer-agnostic template; and (iv) systematic SO and CMB-S4 forecast tables using the one-line Fisher recipe.

ACKNOWLEDGEMENTS

We are grateful to Google and ai@cam for supporting our work.

DATA AVAILABILITY

`classy_szlite` (v0.2.9+) is on PyPI and at https://github.com/CLASS-SZ/classy_szlite; documentation at https://classy-szlite.readthedocs.io. The `CosmoPower ede-v2` emulator weights live at https://github.com/cosmopower-organization/ede. All scripts, chains, and figure-generation code used here are at https://github.com/CLASS-SZ/classy_szlite-paper.

REFERENCES

ACT Collaboration 2024, in prep.

Abazajian K. N., CMB-S4 Collaboration 2016, CMB-S4 science book, first edition (arXiv:1610.02743)

Adame A. G., DESI Collaboration 2024, JCAP, 2025, 021

Arnaud M., Pratt G. W., Piffaretti R., Böhringer H., Croston J. H., Pointecouteau E., 2010, A&A, 517, A92

Battaglia N., Bond J. R., Pfrommer C., Sievers J. L., 2012, ApJ, 758, 75

Beskos A., Pillai N. S., Roberts G. O., Sanz-Serna J. M., Stuart A. M., 2013, Bernoulli, 19, 1501

Betancourt M., 2017, A conceptual introduction to Hamiltonian Monte Carlo (arXiv:1701.02434)

Bingham E., et al., 2018, Pyro: Deep universal probabilistic programming (arXiv:1810.09538)

Bolliet B., Comis B., Komatsu E., Macías-Pérez J. F., 2018, MNRAS, 477, 4957

Bolliet B., Spurio Mancini A., Hill J. C., Madhavacheril M., Jense H. T., Calabrese E., Dunkley J., 2023a, High-accuracy emulators for observables in ΛCDM, $N_{\rm eff}$, Σm_ν, and w cosmologies (arXiv:2303.01591)

Bolliet B., Spurio Mancini A., Hill J. C., Madhavacheril M., Jense H. T., Calabrese E., Dunkley J., 2023b, High-accuracy emulators for observables in ΛCDM, $N_{\rm eff}$, Σm_ν, and w cosmologies (arXiv:2303.01591)

Bradbury J., et al., 2018, JAX: composable transformations of Python+NumPy programs, http://github.com/google/jax

Calabrese E., Hill J. C., Jense H. T., ACT Collaboration 2025, The Atacama Cosmology Telescope: DR6 constraints on extended cosmological models (arXiv:2503.14454)

Cooray A., Sheth R., 2002, Physics Reports, 372, 1

Cranmer K., Brehmer J., Louppe G., 2020, PNAS, 117, 30055

Dolag K., Komatsu E., Sunyaev R., 2016, MNRAS, 463, 1797

Hamilton A. J. S., 2000, MNRAS, 312, 257

Hermans J., Begy V., Louppe G., 2020, Likelihood-free MCMC with amortized approximate ratio estimators (arXiv:1903.04057)

Hoffman M. D., Gelman A., 2014, J. Mach. Learn. Res., 15, 1593

LSST Science Collaborations 2009, LSST science book, version 2.0 (arXiv:0912.0201)

Li Y., 2019, mcfit: multiplicatively convolutional fast integral transforms, https://github.com/eelregit/mcfit

Miller B. K., Cole A., Forré P., Louppe G., Weniger C., 2021, Truncated marginal neural ratio estimation (arXiv:2107.01214)

Papamakarios G., Pavlakou T., Murray I., 2018, Masked autoregressive flow for density estimation (arXiv:1705.07057)

Papamakarios G., Sterratt D. C., Murray I., 2019, Sequential neural likelihood: fast likelihood-free inference with autoregressive flows (arXiv:1805.07226)

Peacock J. A., Smith R. E., 2000, MNRAS, 318, 1144

Phan D., Pradhan N., Jankowiak M., 2019, Composable effects for flexible and accelerated probabilistic programming in NumPyro (arXiv:1912.11554)

Poulin V., Smith T. L., Calderón R., Simon T., 2025, Impact of ACT DR6 and DESI DR2 for early dark energy and the Hubble tension (arXiv:2505.08051)

Roberts G. O., Rosenthal J. S., 2001, Statistical Science, 16, 351

Schaye J., Kugel R., Schaller M., Helly J. C., Braspenning J., Elbers W., McCarthy I. G., et al., 2023, MNRAS, 526, 4978

Seljak U., 2000, MNRAS, 318, 203

Sharrock L., Simons J., Liu S., Beaumont M., 2022, Sequential neural score estimation: likelihood-free inference with conditional score based diffusion models (arXiv:2210.04872)

Spurio Mancini A., Piras D., Alsing J., Joachimi B., Hobson M. P., 2022, MNRAS, 511, 1771

Talman J. D., 1978, J. Comput. Phys., 29, 35

The Simons Observatory Collaboration Ade P., Aguirre J., Ahmed Z., et al., 2019, JCAP, 2019, 056

Tinker J. L., Kravtsov A. V., Klypin A., Abazajian K., Warren M. S., Yepes G., Gottlöber S., Holz D. E., 2008, ApJ, 688, 709

Tinker J. L., Robertson B. E., Kravtsov A. V., Klypin A., Warren M. S., Yepes G., Gottlöber S., 2010, ApJ, 724, 878

Torrado J., Lewis A., 2021, JCAP, 2021, 057

Virtanen P., et al., 2020, Nature Methods, 17, 261