أكاديمية كاوست
KAUST ACADEMY

جامعة الملك عبدالله للعلوم والتقنية
King Abdullah University of Science and Technology

Toward SED Reconstruction of Type Ia Supernovae Using Rubin Observatory Simulations

Sulaiman Alangari,
Renee Hlozek

sulaiman.alangari@kaust.edu.sa, hlozek@dunlap.utoronto.ca

Abstract

Type Ia supernovae are useful cosmological distance indicators because their peak brightness can be standardized once a small number of light-curve properties are known. Two very different observing strategies can supply those properties: spectroscopy, which resolves the full spectrum of a supernova at each epoch but is expensive and cannot keep pace with the number of events upcoming surveys will find, and photometry, which measures only broad-band brightness in a handful of filters but scales to hundreds of thousands of objects. The Vera C. Rubin Observatory's Legacy Survey of Space and Time (LSST) will observe almost exclusively in this photometric regime, which motivates inferring supernova properties directly from sparse, multi-band light curves rather than assuming spectroscopic follow-up. This project builds an inverse-inference pipeline for that problem, working within the `cosmographi` framework and its underlying SALT2 spectral energy distribution (SED) model, which represents each supernova by a small set of local parameters (stretch x_1, colour c, time of peak t_0, redshift z) applied to a fixed, shared phase-wavelength surface.

The pipeline was built in stages. Ordinary SALT2 parameters were first recovered with interpretable χ^2 grid searches, then with a benchmark of machine learning point estimators scaled to 20,000 simulated supernovae. A conditional normalizing flow was then implemented to move beyond point estimates toward full posterior distributions, though its raw credible intervals showed mild empirical undercoverage. The project's central and newest direction moves beyond treating the SALT2 surface as fixed: three shared correction parameters, an overall amplitude a_0, a wavelength-dependent tilt a_1, and a phase-dependent tilt a_2, were introduced and inferred jointly from multiple supernovae's photometry using BlackJAX's implementation of the No-U-Turn Sampler (NUTS). In a controlled injection-and-recovery test with 20 simulated supernovae and 874 total photometric observations, all generated with $a_{0,\text{true}} = a_{1,\text{true}} = a_{2,\text{true}} = 0$, the recovered posterior was $a_0 = -0.00122 \pm 0.00104$, $a_1 = 0.00206 \pm 0.00293$, and $a_2 = -0.00011 \pm 0.00324$, with all three injected values contained within their 95% credible intervals and healthy sampler convergence diagnostics. This is presented as a controlled proof of concept toward SED reconstruction, not as a demonstration of complete SED reconstruction: it shows that broad-band photometry, combined across supernovae through a shared likelihood, can constrain simple corrections to an assumed SED surface. Immediate future work is nonzero injection testing, freeing the local supernova parameters, and generalizing toward a smooth basis expansion of the SED surface.

Date: August 20, 2026

Code: https://github.com/cosmographi/cosmographi

Contents

1 Executive Summary **5**

2 Introduction and Motivation **5**
2.1 Light curves, spectroscopy, and photometry . . . 5
2.2 Why this matters at Rubin scale . . . 6
2.3 SALT2 and the SED surface . . . 7
2.4 Novelty and scope of this project . . . 7
2.5 Structure of this report . . . 7

3 Scientific Background **7**
3.1 Light curve, phase, and wavelength . . . 8
3.2 Rubin filters and cadence . . . 8
3.3 Measurement uncertainty . . . 8
3.4 SALT2 . . . 8
3.5 Local versus global parameters . . . 9

4 Related Work **9**

5 Forward Simulation **10**
5.1 Running the end-to-end simulation tutorial . . . 10
5.2 The forward-simulation procedure . . . 10
5.3 Daily-cadence reference light curve . . . 11
5.4 Forward-model validation: diagnosing and correcting the instrument noise floor . 11
5.5 Realistic OpSim cadence and variable-length observations . . . 13

6 Datasets **13**
6.1 Individual-parameter dataset (20,000 supernovae) . . . 13
6.2 Shared-SED dataset (20 supernovae, 874 observations) . . . 14

7 Grid-Search Inference **14**
7.1 The χ^2 statistic . . . 15
7.2 Single-parameter grid search . . . 15
7.3 Joint four-parameter grid search . . . 15

8 Machine-Learning Benchmarks **16**
8.1 Models and task . . . 17
8.2 Adapting the benchmark to irregular, real-cadence photometry . . . 17
8.3 Why point-estimate models are a supporting, not final, result . . . 17

9 Conditional Normalizing Flow **18**
9.1 What a normalizing flow does . . . 18
9.2 Architecture . . . 18
9.3 Training objective . . . 19
9.4 Evaluation . . . 19

10 Bayesian Shared-SED Inference **21**
10.1 Global amplitude correction, a_0 . . . 21
10.1.1 Sensitivity to the single dense object . . . 22
10.2 Wavelength correction, a_1 . . . 23
10.3 Phase correction, a_2 . . . 23
10.4 Full correction model . . . 24

10.5 Efficient implementation . . . 24
10.6 Prior and likelihood . . . 24
10.7 MCMC and NUTS . . . 25

11 Final a_0-a_1-a_2 Results **26**
11.1 Posterior summary . . . 26
11.2 Convergence diagnostics . . . 26
11.3 Correlations between parameters . . . 27
11.4 Interpretation . . . 27
11.5 Posterior and chain figures . . . 27

12 Discussion **28**
12.1 From grid search to machine learning to normalizing flows to MCMC . . . 28
12.2 Why shared inference is different from per-object fitting . . . 29
12.3 Why this is a controlled proof of concept . . . 30
12.4 What the evidence does, and does not, support . . . 30

13 Limitations **30**

14 Future Work **31**

15 Conclusion **32**

References **32**

A Technical Appendix **33**
A.1 Preprocessing transformations . . . 33
A.2 MCMC configuration . . . 34
A.3 Project script map . . . 34
A.4 Marginal versus joint credible regions . . . 34

1 Executive Summary

This report describes a project that builds, stage by stage, a pipeline for inferring the properties of Type Ia supernovae from the kind of sparse, multi-filter brightness measurements that the Vera C. Rubin Observatory will collect for hundreds of thousands of events. The pipeline starts from an existing, fixed model of what a Type Ia supernova's light output looks like over time and wavelength (SALT2), and asks two increasingly ambitious questions of it.

The first question is: given noisy photometry of one supernova, can its four defining numbers, stretch, colour, time of peak brightness, and redshift, be recovered? This was demonstrated with three progressively more scalable methods: a simple, fully interpretable grid search; a set of machine learning models trained to predict these numbers directly; and a conditional normalizing flow that returns a distribution of plausible values rather than a single best guess.

The second, newer question is more ambitious: rather than only inferring properties for one supernova at a time using a fixed model, can broad-band photometry, combined across many supernovae, tell us anything about whether the shared model itself needs correcting? This project introduced three simple, shared correction parameters, controlling the overall brightness, a wavelength-dependent tilt, and a phase-dependent tilt of the model surface, and used a modern Bayesian sampling method (NUTS, via the BlackJAX library) to infer them jointly from a controlled dataset of 20 simulated supernovae. In this proof-of-concept test, where the true corrections were all set to zero, the method correctly found no significant deviation from the original model, with healthy statistical diagnostics throughout.

This result does not mean the SED has been reconstructed; it means that the machinery for asking this kind of question, combining photometry from many objects to constrain a shared physical model, has been built and validated on a case where the right answer is known in advance. The next steps, described in this report, are to test the same machinery on cases where a real correction has been injected, and then to gradually relax the simplifications made along the way.

2 Introduction and Motivation

A Type Ia supernova is the thermonuclear explosion of a white dwarf star that has been pushed past a critical mass, most likely by accreting material from a companion star or merging with another white dwarf. Because this critical mass is nearly the same for every such star, the explosions release a remarkably consistent amount of energy, making Type Ia supernovae close to "standard candles": objects whose true brightness is known well enough that their apparent brightness, as seen from Earth, reveals their distance. After a further, well-established correction for how the brightness evolves over time and how much dust reddens the light along the way, Type Ia supernovae become "standardizable candles" precise enough to trace the expansion history of the universe and constrain the properties of dark energy [1].

2.1 Light curves, spectroscopy, and photometry

A supernova's *light curve* is simply its brightness plotted as a function of time: it rises to a peak over roughly two to three weeks and then fades over the following months. The full physical description of a supernova at any given moment is its spectral energy distribution (SED), the amount of light it emits at every wavelength, which changes shape as the explosion evolves. There are two fundamentally different ways to measure this.

Spectroscopy disperses a supernova's light into its component wavelengths, typically producing a spectrum with hundreds or thousands of individual wavelength measurements at each epoch observed. This is extremely informative: it reveals detailed physical information, and it is how Type Ia supernovae are conclusively distinguished from other supernova types. It is also expensive, requiring a large telescope and significant exposure time for each object, which limits how many supernovae can realistically be followed up spectroscopically.

Photometry, in contrast, measures the total brightness passing through a small number of broad wavelength filters, typically five or six for an optical survey. This throws away the detailed spectral shape within each filter, keeping only a coarse, band-integrated summary. What photometry loses in detail, it gains in speed: a single wide-field image can measure thousands of objects across many filters in a fraction of the time a single spectrum would take, which is what allows a large photometric survey to observe far more supernovae than any spectroscopic programme could follow up individually.

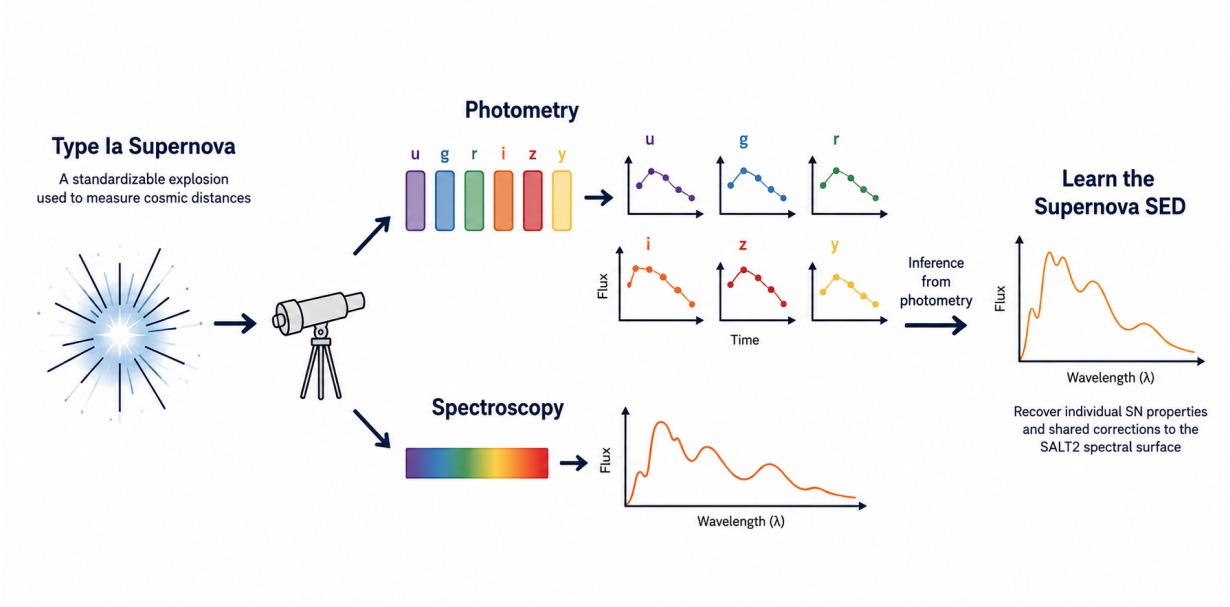


Figure 1: Spectroscopy resolves a supernova's full spectrum but is expensive; photometry measures only coarse, band-integrated brightness in each of six Rubin filters but scales to far more objects. This project infers supernova properties, and, in Section 10, shared corrections to the SALT2 model itself, directly from photometry.

2.2 Why this matters at Rubin scale

The Legacy Survey of Space and Time (LSST), to be conducted at the Vera C. Rubin Observatory [2], is expected to discover on the order of hundreds of thousands of supernovae over its ten-year survey. Only a small fraction of these will ever receive a spectrum. The overwhelming majority will be observed purely photometrically: sparse, irregularly timed measurements in six broad filters (u, g, r, i, z, y), rather than the dense, well-sampled light curves that older, smaller supernova surveys could obtain. This is precisely the regime in which spectroscopic-style detailed analysis is impossible and photometric inference becomes not just convenient but necessary.

Because only a fraction of candidates can be spectroscopically confirmed as genuine Type Ia events, any sample built primarily from photometry will be contaminated by other supernova types (typically grouped as "non-Ia") that mimic Type Ia behaviour closely enough to pass simple photometric cuts. BEAMS (Bayesian Estimation Applied to Multiple Species) [3] was developed for exactly this situation: rather than discarding uncertain candidates, it assigns each one a probability P(Ia) of being a genuine Type Ia and models both the Ia and contaminant populations simultaneously in the cosmological likelihood, so that the inferred cosmological parameters remain unbiased even when individual P(Ia) values are imperfect, provided those probabilities are well calibrated. BEAMS-compatible classification remains a longer-term motivation for this project, but it is not the present technical focus: no non-Ia population has been simulated, and no Ia/non-Ia classifier or P(Ia) values currently exist. This distinction is maintained throughout the report.

2.3 SALT2 and the SED surface

SALT2 [4] is the standard empirical model used to connect a Type Ia supernova's spectral energy distribution to a small number of parameters. It represents the SED as a set of phase-wavelength surfaces (introduced formally in Section 3) that are shared, in the sense that the same underlying surface is used to model every Type Ia supernova. What varies from object to object is a handful of local parameters describing how that shared surface is stretched, shifted, and dimmed for a particular event. Standard practice fits an observed light curve with this fixed, shared surface to recover the local parameters, then hands those parameters to a separate cosmological analysis. The shared surface itself is treated as known and correct throughout this process.

This is a reasonable simplification, but it is a simplification. If the shared surface has systematic errors, for example because it was calibrated on a particular historical sample of supernovae with particular observing conditions, those errors could propagate into every supernova's inferred parameters in a correlated way, since all objects use the same underlying surface. A single supernova's photometry cannot distinguish an error in the shared surface from an error in that object's own parameters; there is not enough information in one object's data to tell the two apart. Combining photometry from *many* supernovae, however, can: if the same discrepancy appears consistently across many different objects at many different redshifts and phases, it is more likely to reflect something about the shared model rather than about any one supernova.

2.4 Novelty and scope of this project

This project investigates that idea directly: rather than assuming the SALT2 surface is exactly correct and only inferring local, per-supernova parameters from it, can broad-band photometry, combined across a population of supernovae, be used to constrain controlled corrections to the shared surface itself? This is not the same question as asking whether any single supernova's spectrum matches the model; it is a question about the model's own validity as inferred from a population of photometric observations. This project does not claim to be the first to consider corrections to supernova spectral models in general, and no such broad claim is made here. What this report investigates is a specific and less commonly explored direction: treating the shared SED surface itself as an object of Bayesian inference, informed only by broad-band photometry rather than by spectroscopy.

2.5 Structure of this report

The remainder of this report follows the progression summarized in the Executive Summary. Section 3 introduces the technical background and notation needed for everything that follows. Section 4 reviews related work. Section 5 describes the forward-simulation pipeline, including a units-error correction to the instrument noise model. Section 6 describes the two datasets used later in the report. Section 7 and Section 8 cover ordinary SALT2 parameter inference by grid search and by machine learning. Section 9 covers an exploratory normalizing-flow approach to posterior uncertainty. Section 10 is the central section of the report: it introduces the shared-SED correction model and the Bayesian inference machinery used to constrain it. Section 11 presents the current main result, a joint three-parameter test of that machinery. Sections 12 through 15 discuss, qualify, and conclude the work, and the appendix records implementation details for reproducibility.

3 Scientific Background

This section defines, in plain language and with the minimum necessary equations, the quantities used throughout the rest of the report.

3.1 Light curve, phase, and wavelength

A supernova's *light curve* is its measured brightness as a function of time. *Phase*, written p, is time measured relative to the supernova's peak brightness in its own rest frame: $p = 0$ at peak, $p < 0$ before peak, and $p > 0$ after peak. *Wavelength*, written λ, is measured in the supernova's rest frame as well, in nanometres. A supernova's SED is a function of both: $F(p, \lambda)$ gives the flux emitted at phase p and rest-frame wavelength λ.

3.2 Rubin filters and cadence

Rubin/LSST observes through six broad-band optical filters, conventionally labelled u, g, r, i, z, and y in order of increasing central wavelength (near-ultraviolet through near-infrared). Each filter has a throughput curve describing what fraction of light at each wavelength it transmits; the flux actually measured in a given filter is the SED integrated against that filter's throughput curve, not the flux at any single wavelength. *Cadence* refers to the pattern and frequency with which the survey revisits a given patch of sky: real Rubin cadence is irregular, with gaps of days to weeks between visits to a given field and with different filters used on different visits, rather than a synchronized daily observation in every band.

3.3 Measurement uncertainty

Every photometric measurement comes with an associated *uncertainty* σ, reflecting the combined effect of Poisson counting statistics from the source itself, the brightness of the night sky background in that filter and at that time, and instrumental effects such as read noise. A measurement with a small σ relative to its flux is a high-signal-to-noise, well-constrained observation; a measurement with σ comparable to or larger than the flux itself is a low-signal-to-noise, weakly constraining observation. This uncertainty is what turns a single flux measurement into a constraint on the underlying model rather than an exact statement of the true flux.

3.4 SALT2

SALT2 [4] models a Type Ia supernova's rest-frame SED as

$$F(p, \lambda) = x_0 \left[M_0(p, \lambda) + x_1 M_1(p, \lambda) \right] \times 10^{-0.4\, c\, CL(\lambda)},$$

where $M_0(p, \lambda)$ is a shared, average phase-wavelength surface common to all Type Ia supernovae, and $M_1(p, \lambda)$ is a second shared surface capturing the dominant way individual supernovae deviate from that average. $CL(\lambda)$ is a shared colour-law function describing how dust-like reddening changes the SED shape with wavelength. The parameters that vary from one supernova to the next are:

- x_0: an overall amplitude, related to the supernova's peak brightness and, through the standard distance-modulus relation, to its distance;
- x_1: the light-curve shape or "stretch" parameter, controlling how broad or narrow the light curve is around peak;
- c: the colour parameter, controlling how much the colour law is applied;
- t_0: the time of peak brightness, fixing where $p = 0$ falls in observer-frame time.

Combined with the supernova's redshift z, which stretches both the wavelength axis (via $1 + z$) and the time axis of the observed light curve relative to the rest frame, these local parameters (x_1, c, t_0, z) are what standard SALT2 fitting infers for each individual supernova. Throughout this report, M_0 and M_1 together are referred to as the *shared SALT2 surfaces*, or simply the SED surface, since it is these two functions, not the local parameters, that this project investigates corrections to.

3.5 Local versus global parameters

This distinction is central to the rest of the report. *Local parameters*, x_1, c, t_0, and z, describe one specific supernova and are, in general, different for every object. *Global parameters*, introduced in Section 10 and written a_0, a_1, a_2, describe a correction to the shared surfaces M_0 and M_1 themselves, and by construction take the same value for every supernova in a given inference. A local parameter answers "what does this object look like"; a global parameter answers "is the shared model itself correct." Distinguishing these is what allows combining photometry from many supernovae to be informative about the model, rather than merely about any one object.

4 Related Work

Type Ia supernova cosmology. Leibundgut and Sullivan [1] review the broader landscape of Type Ia supernova cosmology and the systematics that dominate as samples grow, providing context for why light-curve modelling accuracy matters as statistical uncertainties shrink with larger surveys.

SALT2 and light-curve modelling. SALT2 [4] is the standard empirical model for Type Ia supernova light curves and spectra, introduced in Section 3. Reconstructing its local parameters from photometry is effectively inverting the lossy compression that broadband filters perform on the underlying spectral time series; this project's later sections instead ask whether photometry can constrain the shared surface used by that inversion.

Supernova cosmology with contaminated photometric samples. Hlozek et al. [3] introduced BEAMS and applied it to photometrically classified supernovae from the SDSS-II survey, showing that a probabilistic treatment of classification recovers unbiased cosmological parameters from a contaminated sample. The key insight is that the non-Ia population is not itself the source of bias; the bias comes from ignoring it. By modelling both populations with continuous probability weights rather than a hard classification cut, BEAMS keeps every candidate in the analysis. As stated in Section 2, BEAMS is retained here only as longer-term motivation: no non-Ia population has been simulated in this project, and no P(Ia) classifier has been produced.

Rubin/LSST instrument characterization. The expected photometric performance of LSST, including filter throughputs, zeropoints, sky brightness, and point-spread-function properties, is documented in the official LSST technical notes [5] and the survey overview [2]. These specifications define what realistic simulated training data must look like, and they directly informed the noise-model diagnosis and correction described in Section 5.

Probabilistic inference and posterior calibration. The problem of inferring physical quantities from broadband fluxes with calibrated uncertainty has a close analogue in photometric redshift estimation, where hierarchical probabilistic treatments [6] have shown the value of propagating full probability distributions rather than point estimates. More generally, Kuleshov, Fenner, and Ermon [7] show that regression models, including deep learning models, commonly produce systematically overconfident or underconfident credible intervals unless explicitly checked and recalibrated against empirical coverage, which is exactly the diagnostic used to evaluate the normalizing flow in Section 9.

Normalizing flows and simulation-based inference. Conditional normalizing flows, of the kind used in Section 9, learn an invertible, differentiable transformation between a simple base distribution and a complex target posterior; this project uses the RealNVP coupling-layer architecture of Dinh, Sohl-Dickstein, and Bengio [8]. More broadly, learning posteriors or likelihoods directly from simulated data, rather than evaluating an explicit likelihood, is the subject of the growing field of simulation-based inference, reviewed by Cranmer, Brehmer, and Louppe [9]; the normalizing-flow branch of this project is one instance of that broader approach, while the MCMC-based work in Section 10 instead uses an explicit, tractable likelihood.

Hamiltonian Monte Carlo and NUTS. Hamiltonian Monte Carlo (HMC) uses the gradient of the log posterior to propose distant, high-acceptance moves through parameter space, avoiding the slow, diffusive exploration of simpler random-walk samplers; Neal [10] gives a thorough

introduction. The No-U-Turn Sampler (NUTS) of Hoffman and Gelman [11] removes HMC's need to hand-tune a fixed number of integration steps, adaptively choosing when to stop each trajectory, and is the sampling algorithm used throughout Section 10.

BlackJAX. BlackJAX [12] is a library of composable Bayesian sampling algorithms, including NUTS, built on the JAX automatic-differentiation framework. Its JAX foundation is what makes the differentiable, precomputed-basis likelihood construction described in Section 10.5 practical to sample efficiently.

Cosmographi. This project works within `cosmographi`, a Bayesian forward-modelling framework for supernova cosmology; the codebase is available at `https://github.com/cosmographi/cosmographi`. At the time of writing, no independently citable paper describing `cosmographi` itself could be verified, so it is referenced here by its code repository rather than by a bibliographic entry; this should be confirmed with the project team before final submission.

5 Forward Simulation

Before any parameters can be inferred from data, the pipeline must be able to generate realistic simulated data in the first place: a forward simulation that starts from known supernova and survey properties and produces the kind of sparse, noisy photometry Rubin will actually deliver. This section describes that forward simulation, its validation, and a units error that was found and corrected in it.

5.1 Running the end-to-end simulation tutorial

The `cosmographi` package ships with a tutorial notebook that simulates supernovae from realistic Rubin/LSST observing conditions and runs a full inference chain through to posterior contours. This tutorial was run end-to-end, producing working posterior contours and MCMC chains. Beyond confirming that the pipeline works, this was an effective way to learn the package's existing structure before extending it.

5.2 The forward-simulation procedure

For a single simulated supernova, the forward simulation proceeds through the following steps:

1. Load the SALT2 model, meaning the shared surfaces $M_0(p, \lambda)$ and $M_1(p, \lambda)$ and the colour law $CL(\lambda)$ described in Section 3.

2. Specify local parameters x_1, c, t_0, and z for this simulated supernova, drawn either by hand (for controlled test cases) or from a population model (for bulk dataset generation).

3. Generate the underlying, noiseless rest-frame phase- and wavelength-dependent SED signal, $F(p, \lambda)$, from the SALT2 equation using these parameters.

4. Match this simulated supernova to real Rubin OpSim visits: the OpSim database is a simulated log of when and how the actual Rubin survey is planned to observe each part of the sky, so matching to it means adopting a real, physically plausible observing history for this supernova's sky position rather than an idealized or invented one.

5. Use the visit times, filters, airmass, exposure time, sky brightness, and seeing recorded for each matched visit, since each of these affects how bright the supernova appears and how noisy that measurement will be.

6. Predict the band-integrated flux at each visit by evaluating the SALT2 SED at that visit's observer-frame time and integrating it against that visit's filter throughput curve.

7. Calculate the visit-dependent measurement uncertainty σ for each predicted flux, combining source Poisson noise, sky-background noise, and instrumental terms, all of which depend on the visit's specific observing conditions.

8. Add Gaussian observational noise to the predicted flux, drawn with standard deviation equal to that visit's σ, to produce a realistic noisy measurement.

9. Store the resulting sparse, multi-band light curve: a set of (time, band, flux, flux uncertainty) tuples, one per matched visit, rather than a dense, regularly sampled curve.

This procedure keeps three distinct quantities clearly separate throughout the pipeline: the noiseless underlying SALT2 signal (a smooth function of phase and wavelength, never observed directly), the sparse, irregular times and filters at which Rubin actually samples that signal (fixed by the OpSim match, not chosen by the supernova), and the noisy observed photometry finally recorded (the signal at those sparse times, corrupted by measurement noise). Later sections that discuss "photometry" always mean this last, noisy, sparse quantity.

5.3 Daily-cadence reference light curve

In addition to the irregular, sparse OpSim-matched observations a real survey would produce, a daily-sampled noiseless reference curve was added to the single-supernova workflow, following an implementation shared by a collaborator. This daily curve is not produced by interpolating the sparse OpSim measurements; instead, step 3 above (evaluating the SALT2 SED) is repeated independently at one-day intervals across the visible time range, giving a smooth "true" light curve against which the sparse, noisy OpSim measurements can later be compared. Figure 2 shows this reference curve for a representative simulated supernova.

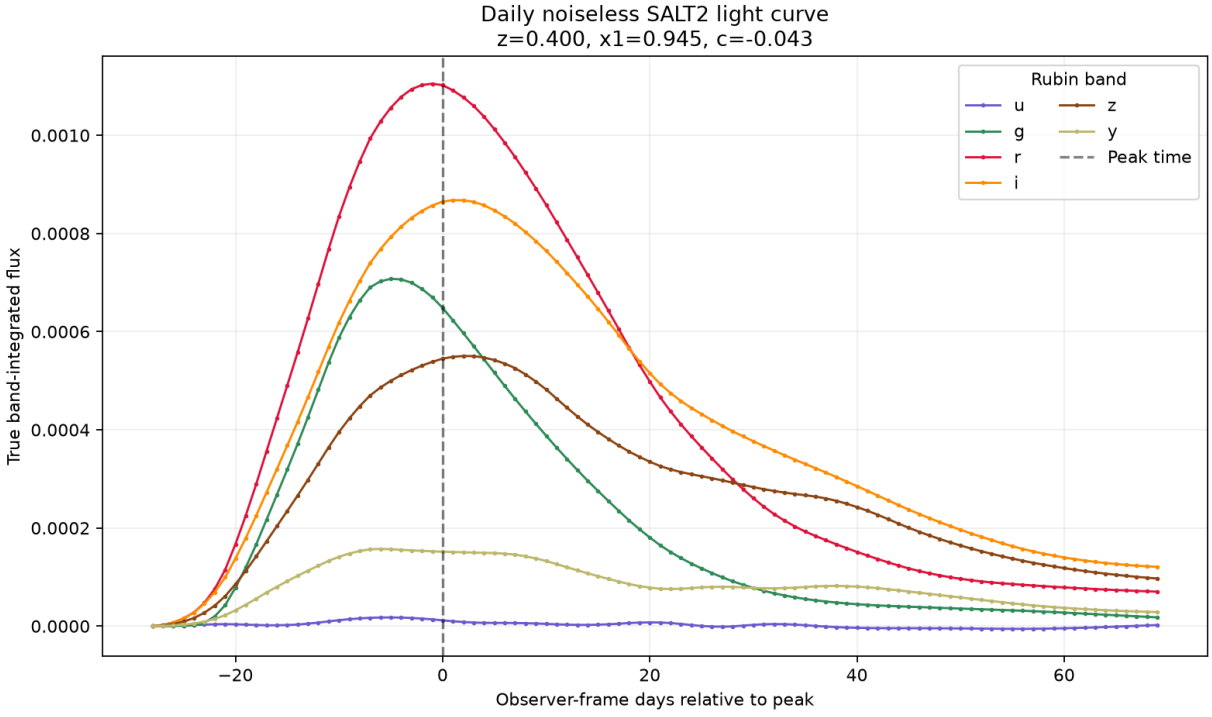


Figure 2: SALT2 forward-simulated light curve evaluated at one-day cadence across six Rubin bands, for a simulated supernova with $z = 0.400$, $x_1 = 0.945$, $c = -0.043$. This deterministic daily curve is the noiseless reference against which sparse, irregular Rubin photometry is later compared; it is evaluated directly from the SALT2 model (step 3 of Section 5.2), not interpolated from sparse observations.

5.4 Forward-model validation: diagnosing and correcting the instrument noise floor

Simulated data are only useful for testing an inference pipeline if the forward model that generates them is itself correct. Early in the project, benchmark results (Section 8) suggested the sim-

ulated dataset was unrealistically easy, in the sense that essentially every simulated supernova had very high signal-to-noise. This was traced to a specific bug rather than to a redshift-coverage issue, through a two-part validation of the flux and noise unit chain.

First, simulated flux was converted back to a real apparent magnitude for a standard, nearby supernova ($x_1 = 0$, $c = 0$, $z = 0.05$) and compared against the textbook expectation (SN Ia peak absolute magnitude ≈ -19.3, plus the standard distance modulus for that redshift). The result matched closely (≈17.3–18 mag across bands versus ≈17.4 mag expected), confirming that the signal-flux calculation itself (step 6 of Section 5.2) was correct.

Second, the simulator's own 5σ single-visit detection-limit magnitude was computed per band and compared directly against LSST's official published depths from SMTN-002 [5], under matching observing conditions (30 s exposure, dark sky, zenith). Table 1 shows that, before correction, the simulator's computed limiting magnitude was systematically 1–1.9 magnitudes deeper than the official depth in every band except the bluest, meaning the simulator was treating LSST as more sensitive than it will actually be.

Table 1: Simulator's computed 5σ single-visit limiting magnitude versus LSST's official published depth (SMTN-002), before correction.

Band	**Official m_5**	**Simulator's m_5 (before)**	**Gap**
u	23.70	23.90	+0.20 mag
g	24.97	25.99	+1.02 mag
r	24.52	25.91	+1.39 mag
i	24.13	25.77	+1.64 mag
z	23.56	25.40	+1.84 mag
y	22.55	24.40	+1.85 mag

Since the signal-flux calculation had already been checked and found correct, the problem was isolated to step 7, the sky-background variance term (part of σ). The old calculation effectively multiplied sky brightness, still expressed as a logarithmic magnitude, by the linear point-spread-function area before converting it to electron flux; magnitudes cannot be scaled this way, and the effect was to make the simulated sky-background noise contribution collapse to near zero, making every simulated observation appear far less noisy than it should. The corrected calculation instead converts the sky-brightness magnitude to linear flux first, and only then multiplies by the point-spread-function area to obtain the total sky flux. *This fix was implemented by a collaborator working on the shared* `cosmographi` *instrument code, not by the author of this report; the diagnosis of the bug, described above, was this project's contribution.*

After the correction, the simulator's computed depths agree closely with the official LSST values in every band, as shown in Table 2. This closed the largest simulation-validity problem identified in this project, and Section 6 shows its effect on the resulting simulated population.

Table 2: Simulator's computed 5σ single-visit limiting magnitude versus LSST's official published depth (SMTN-002), after correction.

Band	**Official m_5**	**Simulator's m_5 (after)**	**Remaining gap**
u	23.70	23.73	+0.03 mag
g	24.97	25.06	+0.09 mag
r	24.52	24.59	+0.07 mag
i	24.13	24.19	+0.06 mag
z	23.56	23.61	+0.05 mag
y	22.55	22.63	+0.08 mag

5.5 Realistic OpSim cadence and variable-length observations

The bulk population generator used to build the datasets in Section 6 was rebuilt around the same per-supernova workflow described in Section 5.2, used for the single-object reference case. Each simulated supernova draws physical parameters from the SALT2 prior and cosmological population model, receives a sky location and time of peak, is cross-matched against the Rubin OpSim database, and uses the actual matched visit times, bands, exposure times, airmass, sky brightness, and seeing conditions from that match. Because each supernova has a different survey history, it has a different number of observations, so the stored data representation was changed from a synchronized dense six-band grid to a variable-length sequence of individual visits, with a per-visit band index, observation time, measured flux, flux uncertainty, and a mask flag distinguishing real observations from padding (padding and masking are described further in Section 6). An intermediate real-survey test dataset built with this corrected pipeline contained 1,992 simulated supernovae, 65,446 real photometric observations, and a mean of 32.85 observations per supernova.

6 Datasets

Two distinct simulated datasets are used later in this report, built for two different purposes. This section describes both, since keeping their sizes and roles clearly separate matters for interpreting the results in Sections 7 through 11.

6.1 Individual-parameter dataset (20,000 supernovae)

After correcting the sky-background calculation (Section 5.4), the real-cadence generator described in Section 5.5 was scaled up to 20,000 simulated supernovae, each with its own real, matched OpSim visit history. This dataset is used for the grid-search demonstration, the machine-learning benchmark, and the normalizing-flow training in Sections 7 through 9. It contains 651,778 total photometric observations, a mean of 32.59 observations per supernova, and between 5 and 100 stored observations per supernova. The generated redshift range was configured as $0.01 < z < 1.0$; the realized range in the generated sample was approximately $0.041 < z < 0.999$. It is split into 16,000 supernovae for training, 2,000 for validation, and 2,000 held out for testing; the normalizing-flow evaluation in Section 9 uses this same 2,000-object test split.

Each stored observation carries four pieces of information: the **band** (which of the six Rubin filters), the **observation time** (relative to the first real visit for that object), the **flux** measured, and the **flux uncertainty** σ. A fifth, per-slot **mask** value distinguishes real observations from padding. Padding is necessary because different supernovae have different numbers of observations (5 to 100, as above), while machine learning models generally require fixed-size input tensors to process a batch of objects together efficiently; every object's sequence is padded up to the same maximum length with placeholder entries, and the mask records which entries are real. This mask must be propagated through every downstream operation, particularly attention and pooling layers (Section 9), so that padded placeholder entries are excluded from any computed average, sum, or attention weight; failing to mask correctly would let padding silently distort a model's predictions.

Using the maximum measured per-visit signal-to-noise ratio for each supernova as a diagnostic, Table 3 and Figure 3 show the resulting distribution.

This is the opposite of the earlier, nearly-all-bright result obtained before the noise-model correction (Section 5.4), and it was obtained without any manual rebalancing toward target bright, medium, or faint proportions: the corrected instrument model naturally produces a broad range of detection difficulty once the underlying physics is right. These bins remain a diagnostic, not a finalized survey selection function; a complete Rubin detection and selection rule has not yet been applied to this dataset, so the below-5 group should be described as having low measured peak SNR, and potentially undetected, rather than as a finalized detected/non-detected classifi-

Table 3: Measured peak-SNR bins for the corrected 20,000-supernova dataset.

Measured peak-SNR range	Count	Fraction
Below 5	8,603	43.0%
5–15	7,570	37.9%
15–50	2,971	14.9%
50 or above	856	4.3%

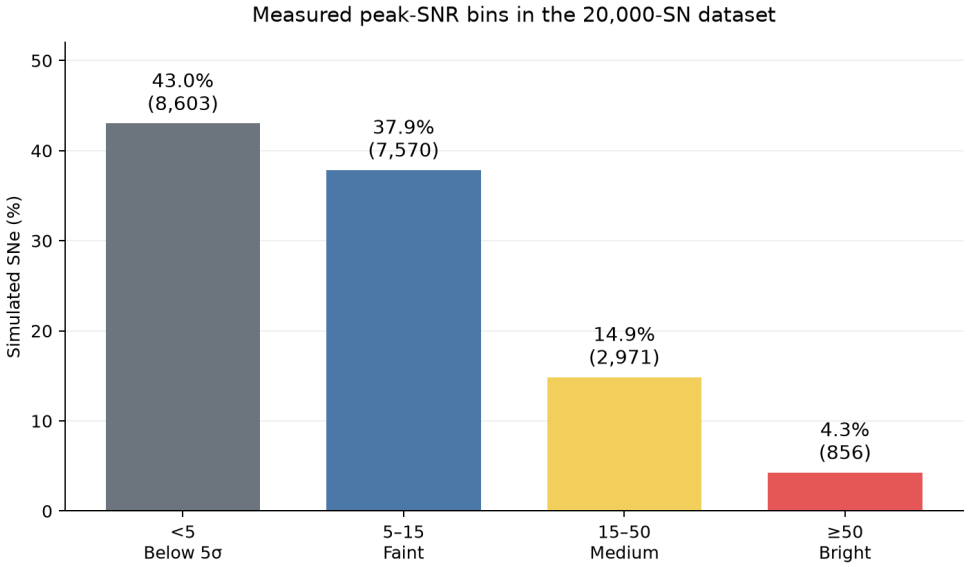


Figure 3: Measured peak-SNR distribution across the corrected 20,000-supernova dataset. Most simulated events now fall below or near the nominal 5σ detection threshold, in contrast to the earlier, uncorrected dataset, which was almost entirely bright.

cation.

6.2 Shared-SED dataset (20 supernovae, 874 observations)

A second, much smaller and deliberately controlled dataset was built specifically for the shared-SED inference in Section 10. It contains 20 simulated supernovae, generated with different redshifts (evenly spaced from 0.1 to 0.6) and different times of peak, each with its own matched OpSim cadence, but all sharing a single global set of correction parameters a_0, a_1, a_2, applied identically to every object. Each object's local parameters x_1, c, t_0, and z are treated as exactly known during this initial experiment, rather than inferred, so that the only unknowns in the resulting Bayesian problem are the three shared parameters. The dataset was generated with the injected truth $a_{0,\text{true}} = a_{1,\text{true}} = a_{2,\text{true}} = 0$, meaning the original, unmodified SALT2 surface.

The full dataset contains **874 total photometric observations across the 20 supernovae**; this is not 874 separate supernovae, but 874 individual visits contributed by 20 objects, each with its own OpSim-matched cadence. One unusually dense event contributed 303 of the 874 visits on its own, which is addressed directly as a sensitivity check in Section 10.1.1.

7 Grid-Search Inference

The first and simplest way this project tested whether SALT2 parameters can be recovered from simulated photometry was a χ^2 grid search: evaluating the forward model at a fixed grid of candidate parameter values and finding which candidate best matches the observed data. This approach is slow and does not scale, for reasons explained below, but it is fully interpretable and

requires no training, which makes it a useful first check that the forward and inverse problems are consistent with each other before moving to more scalable methods.

7.1 The χ^2 statistic

For a candidate parameter value θ, the forward model predicts a flux $F_{\text{model},j}(\theta)$ at each observed visit j. The goodness of fit between this prediction and the actual observed flux $F_{\text{obs},j}$ is measured by

$$\chi^2(\theta) = \sum_j \left(\frac{F_{j,\text{obs}} - F_{j,\text{model}}(\theta)}{\sigma_j} \right)^2 .$$

Each term is divided by that observation's uncertainty σ_j so that precise, low-noise measurements are weighted more heavily in the sum than imprecise, high-noise ones; without this division, a handful of noisy points could dominate the fit even though they carry little real information. A lower χ^2 means the model prediction is, on average, closer to the data relative to the size of the measurement uncertainties, so the candidate parameter value with the lowest χ^2 is the best fit on the grid. An approximate likelihood can be recovered from χ^2 via

$$L(\theta) \propto \exp\left[-\frac{\chi^2(\theta) - \chi^2_{\text{min}}}{2} \right],$$

which follows from the fact that Gaussian measurement noise makes χ^2 (up to an additive constant) equal to $-2 \log L$. *Marginalization* means summing (or, for a continuous parameter, integrating) this likelihood over all values of the parameters not currently of interest, in order to obtain the likelihood, or posterior, for one parameter alone, independent of the specific values the others happened to take on the grid.

Two effects prevent even a well-implemented grid search from recovering the injected truth exactly. First, the observations contain real Gaussian noise, so the data being fit are not identical to the noiseless forward-model prediction at the true parameters; the best fit to a particular noisy realization is generally slightly offset from the truth, by an amount set by the size of the noise. Second, the grid itself is discrete: the true best-fit value may fall between two grid points, in which case the closest grid point, not the true optimum, is what gets reported.

7.2 Single-parameter grid search

Following our supervisor's suggestion to begin with the simplest possible case, a χ^2 grid search was implemented for a single parameter, x_1, with the other parameters held fixed at their known simulated values ($c = -0.043$, $z = 0.4$, $t_0 = 61200$) and the injected truth $x_1 = 0.945$. A grid of 121 candidate values between -3 and 3 was evaluated, summing $\chi^2(x_1)$ over the observed photometric points for this object. The best grid value was approximately $x_1 = 0.80$, offset from the injected truth for the reasons above (noise and discrete grid spacing), and because this initial inference uses a simplified fixed-airmass reconstruction rather than the full per-visit airmass treatment. The purpose of this stage was to verify the forward/inverse loop using the simplest possible method before scaling up.

7.3 Joint four-parameter grid search

The same method was extended to all four SALT2 parameters (x_1, c, t_0, z) jointly, using 14 candidate values in each dimension. Evaluating every combination of 4 parameters at 14 values each gives $14^4 = 38{,}416$ parameter combinations, following the general rule that a grid search's cost grows as

$$N_{\text{combinations}} = N_{\text{grid}}^{N_{\text{parameters}}},$$

where N_{grid} is the number of candidate values per dimension and $N_{\text{parameters}}$ is the number of parameters being searched over jointly. This exponential growth is known as the *curse of*

dimensionality: doubling the number of parameters does not double the cost, it squares it (for fixed grid resolution), and increasing the grid resolution to get a finer, more accurate answer multiplies the cost by the same factor in every dimension at once. With 23 observed photometric points for the test supernova used here, the joint search required approximately 883,568 model-flux evaluations to fit this single object. Table 4 shows the injected and recovered values.

Table 4: Injected versus best-fit grid values for the joint four-parameter χ^2 grid search on a single simulated supernova.

Parameter	Injected value	Best grid value
x_1	0.945	1.1538
c	−0.043	−0.0385
t_0	61200.0	61198.46
z	0.400	0.400

The likelihood approximation above was summed (marginalized) over the other three dimensions to obtain a one-dimensional marginal posterior for each parameter individually, shown in Figure 4.

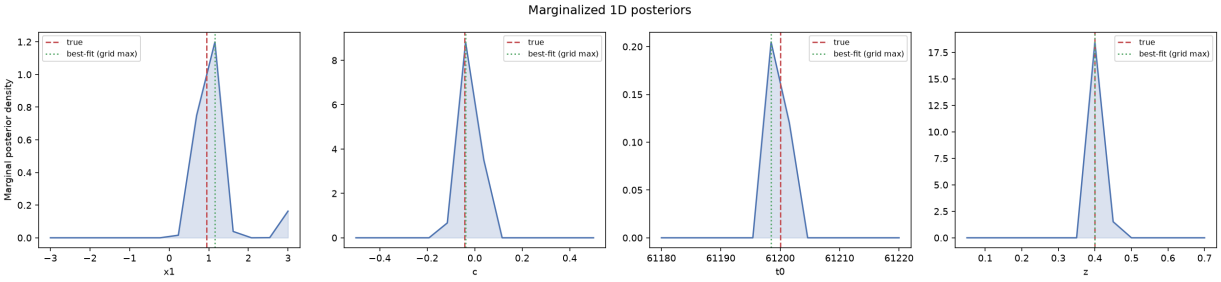


Figure 4: Marginalized one-dimensional posteriors from the joint four-parameter grid search, for the same simulated supernova as Table 4. The dashed red line marks the injected truth and the dotted green line marks the grid maximum for each parameter. t_0, c, and z are recovered tightly and close to their injected values; x_1 shows a broader, slightly asymmetric marginal, including some residual density near the edge of the searched range, illustrating the coarseness of a 14-point grid relative to a continuous parameter.

This demonstrated joint recovery of all four parameters together and the expected correlations between them (for instance, between x_1 and c), but it also demonstrated directly why brute-force grid search does not scale: the curse of dimensionality means that any attempt to search over more parameters, or to use a finer grid for better precision, quickly becomes computationally impractical. This motivated the scalable point-estimation methods of Section 8 and, later, the gradient-based Bayesian methods of Section 10, neither of which requires evaluating every point on a fixed grid.

8 Machine-Learning Benchmarks

The grid search in Section 7 demonstrated that SALT2 parameters can, in principle, be recovered from photometry, but its computational cost scales so badly that it cannot be applied to a large dataset such as the 20,000-supernova sample of Section 6.1. This section describes a benchmark of machine learning models trained to perform the same task, photometry $\rightarrow$ estimated (x_1, c, t_0, z), but scalably: once trained, each model can predict parameters for a new object in a single fast forward pass, rather than by re-evaluating the forward model thousands of times.

8.1 Models and task

Eight point-prediction models were benchmarked on the individual-parameter dataset (Section 6.1): a multilayer perceptron, a Transformer, a convolutional neural network (CNN1D), an LSTM, a GRU, a temporal convolutional network (TCN), a random forest, and XGBoost (gradient-boosted trees). All eight take a single simulated supernova's padded, masked photometric sequence as input and output a single point estimate of (x_1, c, t_0, z). Figure 5 shows the resulting validation mean-squared error (MSE) per model and per parameter.

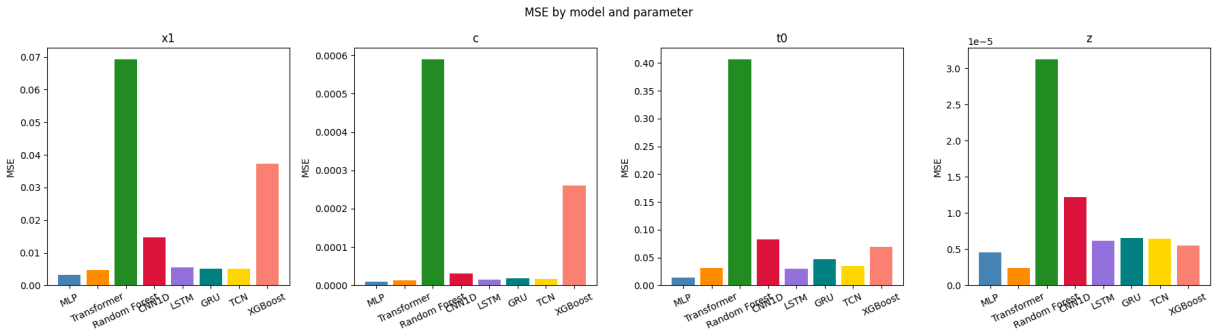


Figure 5: Validation MSE by model and parameter for the eight-model benchmark. Random forest shows by far the largest error across all four parameters, and XGBoost is consistently the second-highest; the remaining models (MLP, Transformer, CNN1D, LSTM, GRU, TCN) form a substantially lower-error cluster, with CNN1D somewhat elevated relative to the other neural architectures on t_0 and z.

The clear separation between the two tree-based models (random forest and XGBoost) and the neural architectures is consistent with the general expectation that sequence-aware neural models are better suited to irregular, variable-length photometric input than models that do not explicitly model sequence structure. Beyond this qualitative ranking, this report does not report precise numerical MSE values for individual models, since the benchmark's role here is supporting rather than central: it establishes that a scalable inverse-inference pipeline for irregular Rubin cadence is achievable, motivating the probabilistic extensions that follow.

8.2 Adapting the benchmark to irregular, real-cadence photometry

Because the dataset uses the variable-length, padded, masked representation described in Section 6.1, several adaptations were required before these models could be applied to it directly. Band identity, which is a discrete category rather than a number, is represented as a learned embedding rather than encoded numerically. Flux is transformed with `arcsinh`, which behaves like a logarithm for large values but remains well-defined and smooth for the negative, noise-dominated flux values that low-signal-to-noise observations regularly produce (unlike an ordinary logarithm, which is undefined for negative numbers). Flux uncertainty is transformed with `log1p` (the logarithm of one plus the value), which compresses its dynamic range while remaining well-behaved near zero. Attention and pooling operations, used in the Transformer and, later, in the normalizing flow (Section 9), are explicitly masked so that padded, non-real visits are excluded from any weighted average or attention score. The recurrent architectures (LSTM, GRU) use packed sequences, a standard technique that lets a recurrent model process a batch of different-length sequences without wasting computation on padding.

8.3 Why point-estimate models are a supporting, not final, result

Training a model on 16,000 examples is a one-time cost; once trained, that same model can predict parameters for any new simulated (or eventually real) object in a fraction of a second, which is what makes this class of method suitable for a survey the size of Rubin's. This is a

fundamentally different trade-off than the grid search: most of the computational cost is paid once, during training, rather than being paid again for every new object.

However, a point estimate is exactly that: a single best-guess number for each parameter, with no accompanying statement of how confident that guess is. This matters for two reasons relevant to the rest of this report. First, without an uncertainty estimate, it is not possible to tell whether a given prediction should be trusted, or how much it should be weighted relative to other, more or less confident, predictions, which is essential for any downstream cosmological use. Second, parameter normalization (rescaling each parameter to a comparable numerical range before training) matters a great deal for how well these models train in practice, and mismatches between the simulated training distribution and any real data a model is eventually applied to (simulated-to-real mismatch) are a known limitation of any model trained purely on simulations, discussed further in Section 13. These two gaps, no calibrated uncertainty and no test against real data, motivated the probabilistic approaches described in Sections 9 and 10.

9 Conditional Normalizing Flow

The point-prediction benchmark in Section 8 produces a single best-guess value for each parameter but no accompanying uncertainty. This section describes a conditional normalizing flow, implemented as a first attempt at recovering a full posterior distribution, $p(x_1, c, t_0, z \mid \text{photometry})$, directly from a neural network, before the project moved to the explicit, likelihood-based Bayesian approach of Section 10.

9.1 What a normalizing flow does

A normalizing flow learns an invertible, differentiable transformation between a simple, fixed base distribution (here, a standard four-dimensional Gaussian, $\mathcal{N}(0, I)$) and a complex target distribution that is hard to write down directly, here the posterior over (x_1, c, t_0, z) given one supernova's photometry. Because the transformation is invertible, it can be run in two directions:

$$\textbf{Training direction:} \quad \theta \xrightarrow{\text{conditional RealNVP}} z_{\text{base}} \sim \mathcal{N}(0, I),$$

$$\textbf{Posterior sampling:} \quad z_{\text{base}} \sim \mathcal{N}(0, I) \xrightarrow{\text{inverse conditional RealNVP}} \theta \text{ samples},$$

where $\theta = (x_1, c, t_0, z)$. During training, known parameter values from the simulated dataset are pushed through the flow toward the base Gaussian, and the flow is adjusted so that this mapping is as close as possible to the true base distribution. Once trained, the direction is reversed: samples are drawn from the simple base Gaussian and pushed backward through the same, now-fixed transformation to produce samples of θ conditioned on a given photometric input. This is fundamentally different from the point-prediction networks of Section 8, which learn a single deterministic function from input to output; a normalizing flow learns a whole distribution, and can be sampled repeatedly to characterize its shape and width.

9.2 Architecture

Figure 6 shows the full architecture described in this subsection, from raw sparse photometry through to posterior samples.

Each photometric visit, band, time, flux, and flux uncertainty, is first preprocessed as in Section 8 (relative time, `arcsinh`-transformed flux, `log1p`-transformed flux uncertainty, and a learned band embedding), then linearly embedded and passed through two masked Transformer [13] encoder layers with four attention heads, using the same padding mask described in Section 6.1 so that padded, non-real visits do not contribute to any attention weight. The output is combined by masked mean pooling (an average over only the real, unmasked visits) into a single 64-dimensional photometry context vector, h_{phot}, summarizing that object's entire light curve.

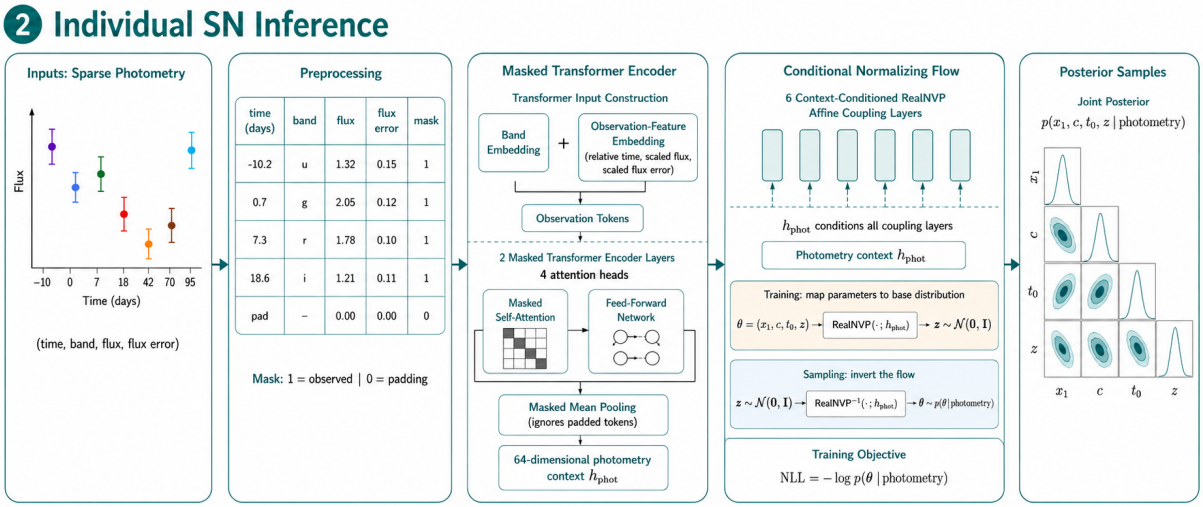


Figure 6: Individual-supernova inference architecture: sparse photometry is preprocessed (relative time, transformed flux and flux uncertainty, band embedding, padding mask), encoded by a masked Transformer into a 64-dimensional context vector h_{phot}, which conditions six RealNVP coupling layers. Reversing the flow turns base-Gaussian samples into posterior samples of (x_1, c, t_0, z).

This context vector conditions six RealNVP affine-coupling layers [8], with alternating masks (each layer transforms half of the four parameter dimensions using an affine function of the other half, and successive layers alternate which half is transformed, so that all four dimensions are eventually coupled to one another). Critically, every one of the six coupling layers is conditioned on the same h_{phot}, so the learned transformation between θ and the base Gaussian depends on that specific object's photometry throughout, not only at the input or output.

9.3 Training objective

The flow is trained by maximizing the log-likelihood it assigns to the true, known parameter values in the training set, or equivalently minimizing the negative log-likelihood,

$$\text{NLL} = -\log p(\theta \mid \text{photometry}),$$

computed using the flow's exact, tractable density (a property RealNVP guarantees by construction, since each coupling layer's Jacobian determinant is cheap to compute). This is a different objective from the mean-squared-error loss typically used to train the point-prediction models in Section 8: minimizing NLL rewards a model for assigning high probability density to the correct answer specifically, and implicitly penalizes overconfidence, since a narrow predicted distribution that misses the true value receives an especially large NLL penalty.

9.4 Evaluation

Evaluated on the held-out 2,000-object test set described in Section 6.1, the flow gave the results in Table 5 and Figure 7.

RMSE (root-mean-square error) here is computed between each test object's true parameter value and the mean of that object's posterior samples from the flow, summarizing the flow's point-estimate accuracy in the same units as the parameter itself. **Empirical coverage** answers a different question: of all the test objects, what fraction had their true parameter value actually fall inside the flow's own stated 68% (or 95%) credible interval for that object? If the flow were perfectly calibrated, roughly 68% and 95% of test objects would show their true value inside those respective intervals, by definition of what a credible interval means.

Table 5: Conditional normalizing-flow evaluation on the held-out 2,000-object test set.

Parameter	Posterior-mean RMSE	68% empirical coverage	95% empirical coverage
x_1	0.9766	62.3%	92.9%
c	0.0720	65.3%	92.8%
t_0	11.84 days	68.3%	94.9%
z	0.0747	66.3%	93.7%

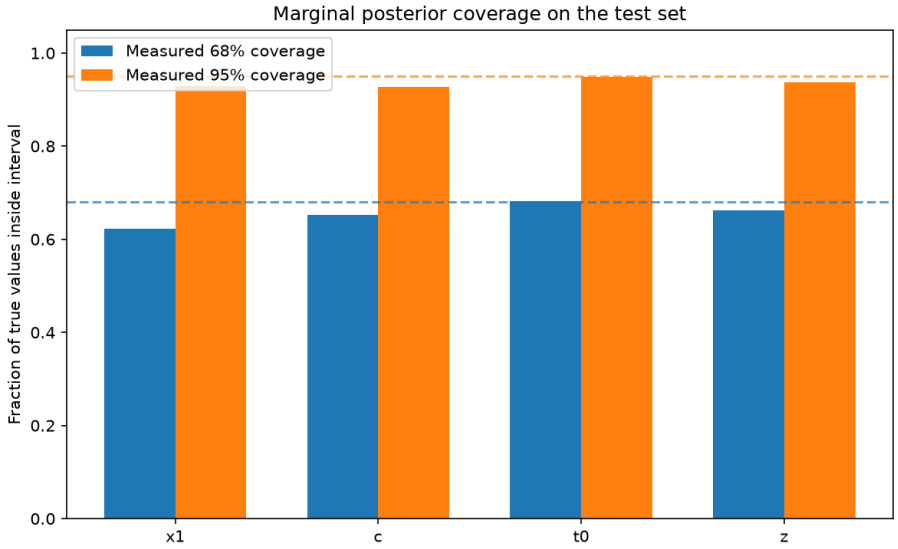


Figure 7: Measured empirical coverage of the flow's 68% and 95% marginal credible intervals, per parameter, against the nominal target levels (dashed lines). Every parameter falls below its nominal line at both coverage levels, showing the flow is consistently, mildly overconfident across all four parameters rather than only for one or two of them.

It is worth being precise about what these intervals are and are not. A *Bayesian credible interval* is a statement from a single posterior distribution: it says an X% credible interval contains X% of that distribution's own probability mass. *Empirical coverage*, as measured here, is a repeated-experiment, frequentist-style check applied on top of that: across many different test objects, each with its own posterior from the flow, does the stated credible level actually match the fraction of times the true value falls inside it? A flow can produce perfectly valid credible intervals for each individual object in the Bayesian sense (correctly summarizing that object's own posterior, given the flow's internal beliefs) while still being poorly calibrated in the empirical coverage sense, if those internal beliefs are themselves systematically overconfident or underconfident across the population of objects it was trained on. This is exactly the effect Kuleshov, Fenner, and Ermon [7] describe as a general property of learned regression uncertainty, and it is why the empirical check is necessary rather than assuming a flow's raw output is automatically trustworthy.

The empirical coverage falling somewhat below the nominal 68% and 95% levels for every parameter in Table 5 indicates that the raw flow is mildly overconfident: its intervals are, on average, narrower than they should be given how often the true value actually falls outside them. This would need additional calibration and validation before scientific use. This branch remains a useful uncertainty-estimation baseline, but it is not the current main result: after discussion with a collaborator, an explicit, likelihood-based Bayesian approach using Markov Chain Monte Carlo (MCMC) was selected as the next method, described in Section 10, in order to establish a transparent baseline with directly interpretable diagnostics before returning to more complex

amortized models such as this flow.

10 Bayesian Shared-SED Inference

Every method described so far, grid search, the machine-learning benchmark, and the normalizing flow, answers the same kind of question: given one supernova's photometry, and assuming the shared SALT2 surface M_0, M_1 is exactly correct, what are that object's local parameters? This section asks a different question, introduced conceptually in Section 2: given photometry from *many* supernovae, is there evidence that the shared surface itself needs correcting?

Formally, this is a shift from

$$p(x_1, c, t_0, z \mid \text{one supernova's photometry})$$

to

$$p(a_0, a_1, a_2 \mid \text{photometry from all supernovae in the dataset}),$$

where a_0, a_1, a_2 are global correction parameters, shared by every object, introduced below. Figure 8 shows how these global corrections fit into the forward simulation described in Section 5: the same SALT2 surface and per-object local parameters used throughout this report are combined with the global corrections into a single corrected surface, which is then observed through the same realistic Rubin cadence and noise model as every other simulated dataset in this project.

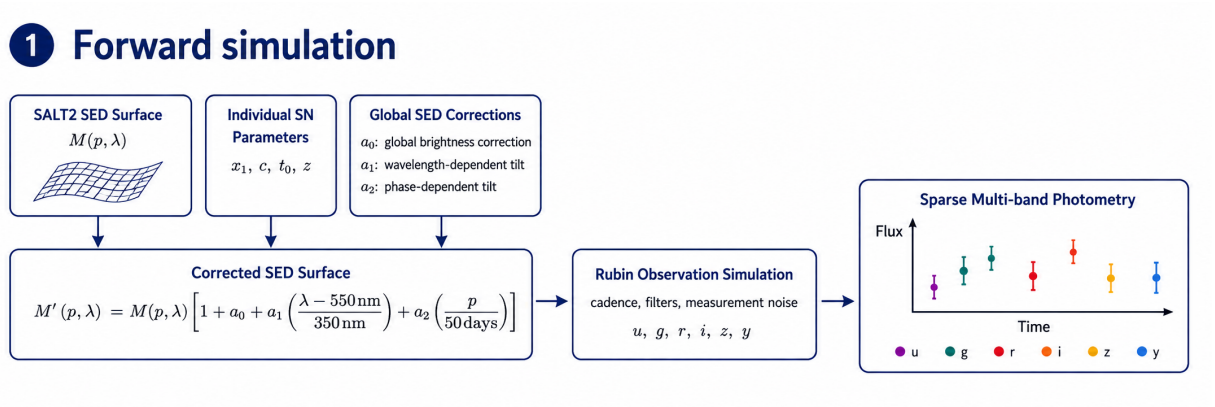


Figure 8: The forward-simulation pipeline extended with global SED corrections. The fixed SALT2 surface $M(p, \lambda)$, an individual supernova's local parameters (x_1, c, t_0, z), and the shared global corrections (a_0, a_1, a_2) combine into a corrected surface $M'(p, \lambda)$ (Section 10.4), which is then observed through the same Rubin cadence, filters, and measurement noise as every other simulated dataset in this report, producing sparse multi-band photometry.

This shift is what makes combining data across supernovae informative in a new way: a single supernova's photometry genuinely cannot distinguish "this object is unusual" from "the shared model is wrong," because there is only one object's worth of information to work with. But if the *same* systematic offset appears consistently across the photometry of many different supernovae, at many different redshifts, phases, and cadences, that consistency is itself evidence about the shared model, since it is far more likely to arise from something common to all of them (the surface they all share) than from many unrelated, independent coincidences. A shared parameter therefore gains statistical power precisely by pooling observations that would otherwise each only inform their own object's local parameters.

10.1 Global amplitude correction, a_0

As an initial, deliberately simple test of this idea, a single global scaling correction was defined,

$$M' = M(1 + a_0),$$

applied identically to both SALT2 surfaces M_0 and M_1, where $a_0 = 0$ recovers the original, unmodified SALT2 surface, $a_0 > 0$ gives a uniformly brighter surface, and $a_0 < 0$ gives a uniformly fainter surface. This is a calibration and proof-of-concept test, not a reconstruction of SED shape: it changes only the overall amplitude of the surface, uniformly across every wavelength and phase, and was chosen specifically because a problem with only one unknown is the simplest possible version of the shared-inference idea, and the easiest to validate against an independent check (Section 10.1.1).

An initial, one-parameter-only version of the shared-SED dataset (Section 6.2) was generated with $a_{0,\text{true}} = 0$ and inferred using an intentionally offset prior, $a_0 \sim \mathcal{N}(0.05, 0.1^2)$, so that the posterior would have real distance to move from the prior toward the truth if the method were working correctly. BlackJAX NUTS recovered $a_0 = -0.00136 \pm 0.00083$, closely matching the independent analytical posterior of $a_0 = -0.00138 \pm 0.00084$ available for this simple, linear-Gaussian case, with $\hat{R} = 1.0001$, an effective sample size of approximately 2,596, a mean acceptance probability of 0.944, and zero divergences. Figures 9 and 10 show this result. The injected truth of zero fell within the 95% credible interval, though not within the central 68% interval, an outcome consistent with the small shift expected from one particular finite noisy realization of the data, and discussed further in Section 10.1.1.

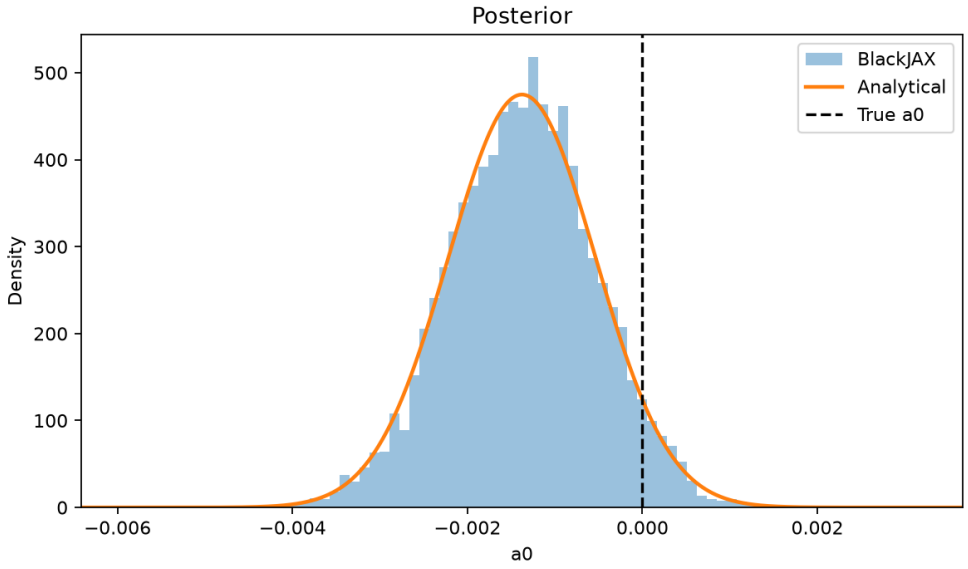


Figure 9: BlackJAX NUTS posterior for the global surface-scaling parameter a_0 alone (blue histogram), from the initial one-parameter validation test, against the analytical Gaussian posterior (orange line) and the injected truth $a_{0,\text{true}} = 0$ (dashed line). The posterior moved from the intentionally offset prior toward the injected value.

Close agreement between the independently computed analytical posterior and the MCMC result is what validates the sampling implementation itself, independent of any question about the astrophysics: it confirms that the likelihood, prior, and sampler are all working correctly together on a case simple enough to check by hand, before extending to the higher-dimensional case where no analytical answer is available. That extension is described next.

10.1.1 Sensitivity to the single dense object

As noted in Section 6.2, one unusually dense simulated supernova contributed 303 of the 874 total visits in the shared-SED dataset on its own. As a sensitivity check, the a_0-only analytical result was recomputed with that object removed entirely: this gives $a_0 = -0.00212 \pm 0.00172$, broader, as expected with less data, but still consistent with zero within the credible interval. This shows that the dense event contributes substantial statistical information to the posterior,

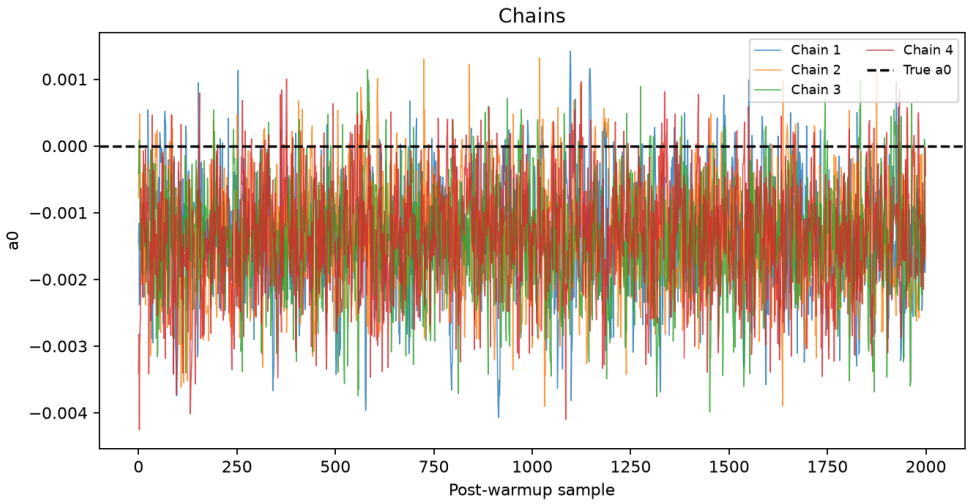


Figure 10: The four BlackJAX NUTS chains for a_0 alone, after warm-up, from the initial one-parameter validation test, shown against the injected truth (dashed line). The chains mix well and show no sign of getting stuck.

since removing it visibly widens the uncertainty, but is not solely responsible for the near-zero result, since the recovered value remains consistent with the full-dataset result even without it.

10.2 Wavelength correction, a_1

A purely global amplitude scale cannot capture a large family of plausible SED errors: a survey calibration issue, for instance, might make the model too bright at blue wavelengths and too faint at red wavelengths (or vice versa) rather than uniformly off by the same fraction everywhere. To capture this, a second global parameter, a_1, was introduced as a wavelength-dependent linear tilt,

$$a_1 \frac{\lambda - 550\,\mathrm{nm}}{350\,\mathrm{nm}},$$

where 550 nm is a *pivot wavelength*, roughly in the middle of the optical range these observations probe, and 350 nm is a *wavelength scale*. Centring the correction on the pivot wavelength means the tilt is exactly zero at $\lambda = 550$ nm regardless of the value of a_1; this is a deliberate choice that keeps a_1 distinguishable from a_0, since a_0 shifts the surface at every wavelength equally while a_1 leaves the pivot wavelength untouched and tilts flux away from it in opposite directions on either side. Dividing by the wavelength scale keeps a_1 a small, dimensionless number of a similar order of magnitude to a_0, rather than a value that would otherwise need to be extremely small simply because raw wavelengths in nanometres are numerically large; without this scaling, a numerically tiny, physically meaningless value of a_1 could produce an enormous, unphysical correction. A positive a_1 tilts the surface brighter at longer (redder) wavelengths and fainter at shorter (bluer) wavelengths relative to the pivot; a negative a_1 tilts the other way. Because SALT2's own colour parameter c already changes an individual supernova's flux as a function of wavelength through the colour law $CL(\lambda)$, a shared wavelength tilt a_1 may be partially degenerate with c, a point returned to in Section 13.

10.3 Phase correction, a_2

A third global parameter, a_2, was introduced as a phase-dependent linear tilt,

$$a_2 \frac{p}{50\,\mathrm{days}},$$

using the phase convention from Section 3: $p = 0$ at peak brightness, $p < 0$ before peak, and $p > 0$ after peak. Fifty days is used purely as a numerical scale, chosen to keep a_2 a similarly

small, dimensionless number over the phase range these light curves typically cover, in the same way the 350 nm wavelength scale does for a_1. Because the correction is proportional to p itself, it is exactly zero at peak brightness and changes sign across peak: a positive a_2 makes the model brighter than SALT2 predicts after peak and fainter before peak (or the reverse for negative a_2), rather than uniformly brighter or fainter at all phases. Since SALT2's stretch parameter x_1 and time-of-peak parameter t_0 also control how a supernova's brightness evolves before and after peak, a shared phase tilt a_2 may be partially degenerate with either of these local parameters, again returned to in Section 13.

10.4 Full correction model

Combining all three terms gives the correction model actually implemented and used for the results in Section 11:

$$M'(p,\lambda) = M(p,\lambda)\left[1 + a_0 + a_1\frac{\lambda - 550\,\mathrm{nm}}{350\,\mathrm{nm}} + a_2\frac{p}{50\,\mathrm{days}}\right],$$

applied identically to both relevant SALT2 surfaces along the full phase-wavelength grid. This is still a low-dimensional, linear correction, not an arbitrary reconstruction of every individual grid cell of the SED surface: it can only stretch, tilt in wavelength, and tilt in phase, and cannot represent, for example, a correction localized to one narrow wavelength range or one specific phase. What it does test is whether broad-band photometry contains information about these three specific, physically motivated kinds of shared deviation from the assumed surface, which is a meaningfully different and much more constrained question than free-form SED reconstruction.

10.5 Efficient implementation

Rebuilding the entire SALT2 surface from scratch at every candidate (a_0, a_1, a_2) visited during MCMC sampling would be needlessly expensive, since thousands of candidate values are evaluated over the course of sampling. Because the correction model above is linear in a_0, a_1, and a_2, this cost can be avoided: the baseline model flux F_{baseline} (at $a_0 = a_1 = a_2 = 0$) is calculated once, and the flux response associated with a unit change in a_1 alone, $F_{\mathrm{wavelength}}$, and in a_2 alone, F_{phase}, are each also calculated once. Every candidate model flux visited during sampling is then constructed as the linear combination

$$F_{\mathrm{model}} = F_{\mathrm{baseline}}(1 + a_0) + a_1 F_{\mathrm{wavelength}} + a_2 F_{\mathrm{phase}},$$

which is cheap to evaluate for any candidate (a_0, a_1, a_2) without recomputing the underlying SALT2 surface itself. This keeps the likelihood function differentiable end-to-end in JAX (necessary for NUTS, which requires the gradient of the log posterior, as explained in Section 10.7), and keeps each individual NUTS step computationally cheap despite the thousands of steps a full run requires.

10.6 Prior and likelihood

Independent, intentionally offset priors were used for all three global parameters,

$$a_j \sim \mathcal{N}(0.05, 0.1^2), \qquad j = 0, 1, 2,$$

offset from the injected truth of zero for the same reason as in the a_0-only test (Section 10.1): if inference correctly moves the posterior away from this prior and back toward the true, injected value, that is itself evidence the method is doing something real rather than simply reporting its prior back unchanged.

For observation i (using i to index all 874 observations across all 20 supernovae, per Section 6.2), the model prediction $F_{\mathrm{model},i}(a_0, a_1, a_2)$ is evaluated using that observation's own phase

and wavelength via the efficient decomposition above, and the observation is assumed Gaussian-distributed around that prediction,

$$F_{\mathrm{obs},i} \sim \mathcal{N}(F_{\mathrm{model},i}(a_0, a_1, a_2),\, \sigma_i^2).$$

The combined log-likelihood over all 874 observations is

$$\log L_{\mathrm{total}} = \sum_i \log p(F_{\mathrm{obs},i} \mid a_0, a_1, a_2),$$

and the quantity actually sampled by NUTS is the log posterior (up to an additive normalizing constant that does not depend on the parameters and can be ignored during sampling),

$$\log p(a \mid \text{all photometry}) = \log L_{\mathrm{total}} + \log p(a) + \text{constant}, \qquad a = (a_0, a_1, a_2),$$

where $\log p(a)$ is the sum of the three independent Gaussian log-priors above. Every one of the 874 observations, drawn from all 20 supernovae, contributes to this single sum, which is precisely the mechanism by which shared information from many objects jointly constrains the three global parameters, as described conceptually at the start of this section.

10.7 MCMC and NUTS

Markov Chain Monte Carlo (MCMC) is a family of methods for drawing samples from a probability distribution, here the posterior $p(a \mid \text{all photometry})$, when that distribution cannot easily be sampled from directly. Each individual draw is called a *posterior sample*: one plausible combination (a_0, a_1, a_2) consistent with the data and priors, not a simulated supernova and not itself a new photometric observation. A *chain* is a sequence of such samples produced by running the algorithm forward step by step, where each new sample depends on the previous one; running several independent chains from different starting points, as done here, is a standard way to check that the sampler is exploring the full posterior rather than getting stuck in one region.

Hamiltonian Monte Carlo (HMC) [10] is the specific family of MCMC methods used here. Rather than proposing new candidate parameter values randomly, HMC uses the gradient of the log posterior (how steeply the posterior's probability changes in every direction) to simulate a physical analogy: the current parameter values are treated as a position, a random "momentum" is drawn, and the pair evolves for a while under simulated Hamiltonian dynamics, in which regions of higher posterior probability act like regions of lower potential energy that the trajectory is pulled toward, similar to a ball rolling on a landscape. This simulated trajectory is computed numerically using *leapfrog integration*, a discrete step-by-step approximation to the true, continuous dynamics. Because the trajectory uses gradient information rather than a random guess, HMC can propose large, high-acceptance moves through parameter space far more efficiently than simpler random-walk MCMC methods, especially in problems with several correlated parameters, as this one has (Section 11).

Ordinary HMC requires the user to choose, in advance, how many leapfrog steps each trajectory should take; too few steps under-explores the posterior, while too many wastes computation retracing ground already covered. The No-U-Turn Sampler (NUTS) [11] removes this requirement by monitoring the trajectory as it runs and automatically stopping it once it detects that continuing would start doubling back on itself, a "U-turn" in the trajectory's path, rather than continuing to explore new territory. This is the sampler actually used throughout this section, via its implementation in BlackJAX [12].

Before collecting any samples that are kept for analysis, a *warm-up* phase is run, during which the sampler adjusts its own internal step-size and mass-matrix settings (parameters controlling how large a step the leapfrog integrator takes, and how it weights different parameter directions) to match the specific shape of this problem's posterior; these warm-up samples are then discarded, and only the samples collected after warm-up, the *retained samples*, are used for the results reported in Section 11. The specific configuration used throughout is four independent

chains, 1,000 warm-up steps per chain, and 2,000 retained samples per chain, for 8,000 total retained posterior samples. To be explicit: NUTS, as used here, is a gradient-based Hamiltonian Monte Carlo method, not simple random-walk Metropolis sampling, and running it is not a form of machine-learning training; no loss function is minimized and no weights are learned, only samples from an already fully specified posterior distribution are drawn.

11 Final a_0-a_1-a_2 Results

This section presents the current main result of the project: the joint, three-parameter version of the shared-SED inference described in Section 10, run on the full shared-SED dataset (20 supernovae, 874 observations, Section 6.2) with the injected truth

$$a_{0,\text{true}} = 0, \qquad a_{1,\text{true}} = 0, \qquad a_{2,\text{true}} = 0,$$

using the same BlackJAX NUTS configuration described in Section 10.7: four chains, 1,000 warm-up steps per chain, 2,000 retained samples per chain, for 8,000 total retained posterior samples of (a_0, a_1, a_2).

11.1 Posterior summary

Table 6 reports the posterior mean and standard deviation for each parameter, together with the marginal 68% and 95% credible intervals. As in Section 10, these are Bayesian credible intervals from a single joint posterior, not repeated-experiment confidence intervals.

Table 6: Posterior summary for the joint a_0-a_1-a_2 inference. All quantities are dimensionless, per the definitions in Section 10.

	a_0	a_1	a_2
Posterior mean $\pm$ SD	-0.00122 ± 0.00104	0.00206 ± 0.00293	-0.00011 ± 0.00324
68% credible interval	$[-0.00226, -0.00019]$	$[-0.00085, 0.00496]$	$[-0.00328, 0.00311]$
95% credible interval	$[-0.00327, 0.00083]$	$[-0.00368, 0.00778]$	$[-0.00644, 0.00629]$

11.2 Convergence diagnostics

Table 7 reports sampler diagnostics for each parameter individually, plus the two diagnostics that summarize the sampler run as a whole.

Table 7: Convergence diagnostics for the joint a_0-a_1-a_2 inference.

Parameter	$\hat{R}$	**Effective sample size (ESS)**
a_0	1.00019	$\approx$5,182
a_1	0.99990	$\approx$3,368
a_2	1.00012	$\approx$2,940

Mean acceptance probability: 0.9377 Divergences: 0

$\hat{R}$ (R-hat) compares the variance of samples within each chain to the variance across all chains combined; a value close to 1 for every parameter, as seen here, indicates that the four independent chains have converged to the same distribution rather than each exploring a different region. Effective sample size (ESS) estimates how many independent samples the 8,000 correlated retained samples are actually worth, since successive MCMC samples are not fully independent of one another; an ESS in the low thousands out of 8,000 raw samples, as seen for all three parameters, is a healthy result for this kind of correlated chain. Zero divergences (a diagnostic specific to Hamiltonian methods, flagging numerically unstable trajectories where the

simulated dynamics broke down) and a mean acceptance probability close to the standard target of 0.8–0.95 both indicate that the sampler explored this posterior without numerical difficulty.

11.3 Correlations between parameters

Table 8 reports the pairwise correlation coefficients between the three parameters, computed from the retained posterior samples.

Table 8: Pairwise posterior correlation coefficients for the joint a_0-a_1-a_2 inference.

Parameter pair	Correlation
$\mathrm{corr}(a_0, a_1)$	$+0.335$
$\mathrm{corr}(a_0, a_2)$	-0.545
$\mathrm{corr}(a_1, a_2)$	-0.293

The strongest correlation is between a_0 (overall amplitude) and a_2 (phase tilt); this is a physically sensible degeneracy, since a supernova sample with limited or asymmetric phase coverage relative to peak brightness can partially reproduce the effect of a uniform brightness shift by instead tilting brightness across phase, and vice versa. Weaker correlations appear between a_0 and a_1, and between a_1 and a_2.

11.4 Interpretation

All three injected truths, $a_{0,\mathrm{true}} = a_{1,\mathrm{true}} = a_{2,\mathrm{true}} = 0$, are contained within their respective marginal 95% credible intervals in Table 6. The a_1 and a_2 truths additionally fall within their marginal 68% intervals; the a_0 truth falls just outside its central 68% interval but comfortably inside its 95% interval, an outcome consistent with a particular finite noisy realization of the data rather than with a flaw in the method, and directly analogous to the same pattern seen in the one-parameter-only validation test in Section 10.1. This should be described as consistency with the injected zero corrections within the 95% credible interval, not as exact recovery of zero.

Comparing the widths of the credible intervals across the three parameters, a_0 is the most tightly constrained (posterior standard deviation ≈ 0.00104), and a_2 is the least tightly constrained (posterior standard deviation ≈ 0.00324), with a_1 in between. This ordering is consistent with the correlation structure above: because a_0 affects every observation identically regardless of phase or wavelength, it benefits most directly from combining all 874 observations, whereas a_1 and a_2 each depend on how well the data actually span the wavelength or phase range needed to distinguish them from a_0 and from each other, and are correspondingly less tightly determined by this particular dataset.

Finally, and most importantly, the posterior moved substantially away from the intentionally offset prior centred at 0.05 for all three parameters, toward the region around zero, and did not falsely detect a significant nonzero correction where none was injected. Together with the healthy convergence diagnostics in Table 7, this is what supports treating the joint three-parameter inference as validated on this null test, ready for the nonzero-injection tests described in Section 14.

11.5 Posterior and chain figures

Figure 11 shows the joint posterior: marginal histograms for a_0, a_1, and a_2 across the top row, each with the injected truth marked by a dashed line, and pairwise scatter plots of the retained posterior samples across the bottom row, with the injected truth marked by a black cross on each panel.

Figure 12 shows the four sampled chains for all three parameters after warm-up.

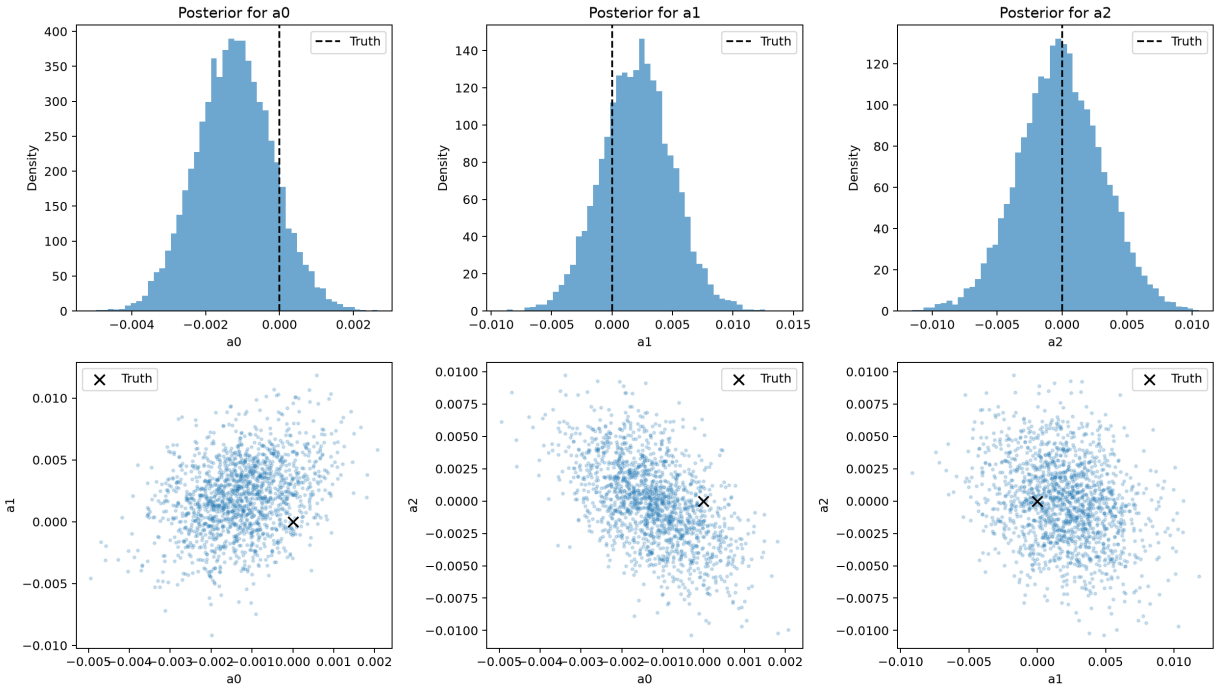


Figure 11: Top row: marginal posteriors for a_0, a_1, and a_2 from the joint three-parameter inference, with the injected truth (dashed line) at zero for all three. Bottom row: pairwise scatter plots of the same retained posterior samples, with the injected truth marked by a black cross. The visible tilt in the a_0-a_2 panel reflects the negative correlation reported in Table 8; the a_0-a_1 and a_1-a_2 panels show milder tilts, consistent with their smaller correlation coefficients. These pairwise panels show raw posterior samples, not formally computed joint contours (Appendix A).

12 Discussion

12.1 From grid search to machine learning to normalizing flows to MCMC

The project's methodological progression, grid search, then machine learning point estimates, then a normalizing flow, then explicit MCMC, was not arbitrary; each stage exposed a specific limitation that motivated the next. The grid search (Section 7) confirmed that the forward and inverse problems were consistent with each other, but its cost scales exponentially with the number of parameters, ruling it out for any large dataset. The machine learning benchmark (Section 8) solved the scaling problem, but point estimates alone say nothing about how confident a given prediction is. The normalizing flow (Section 9) addressed that by learning a full posterior distribution, but its empirical coverage showed it was mildly overconfident, meaning its stated uncertainties could not yet be taken at face value. Explicit, likelihood-based MCMC (Section 10) was adopted specifically because, unlike a learned amortized model, its behaviour can be checked directly: the prior and likelihood are written down explicitly, standard convergence diagnostics apply, and, in the simplest case, an independent analytical solution is even available to check the sampler against (Section 10.1). This is why the project's most detailed and central result uses MCMC rather than the flow, even though the flow is in principle a faster method once trained.

This progression also illustrates a distinction worth stating plainly: point accuracy and uncertainty calibration are different goals, and improving one does not automatically improve the other. A model can have excellent point-estimate accuracy (low RMSE) while still being poorly calibrated (its stated confidence not matching how often it is actually correct), exactly the pattern seen for the normalizing flow in Table 5. Explicit likelihood-based MCMC sidesteps this particular failure mode by construction, since its output is not a learned approximation to a posterior

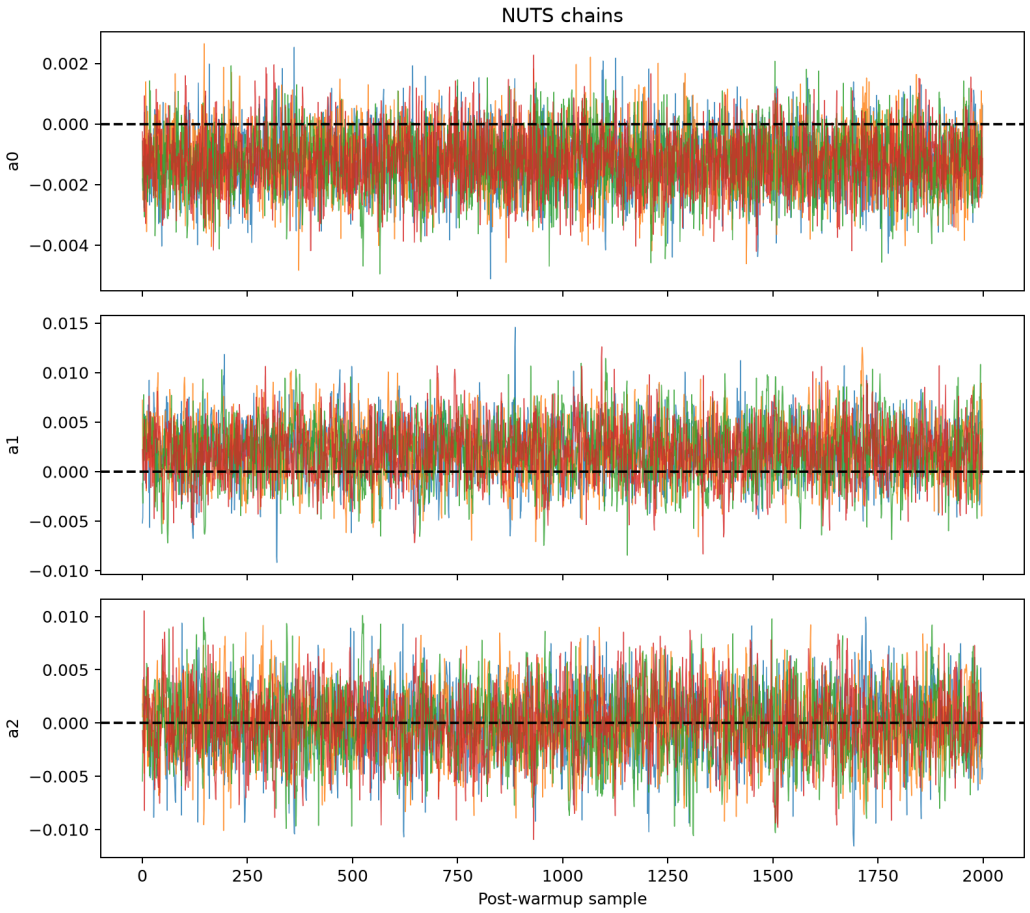


Figure 12: The four BlackJAX NUTS chains for a_0, a_1, and a_2 after warm-up, shown against the injected truth (dashed lines at zero). All four chains mix well and explore the same region for every parameter, with no visible trend, drift, or long excursion by any single chain, consistent with the healthy $\hat{R}$ and ESS values in Table 7. Good mixing of this kind, chains that overlap and move freely rather than getting stuck in separate regions, is what these convergence diagnostics are designed to detect quantitatively.

but samples from an exactly specified one; what it cannot sidestep is whether the likelihood and priors themselves are the right model for the data, which is a separate, ongoing question.

12.2 Why shared inference is different from per-object fitting

Learning shared global parameters a_0, a_1, a_2 from multiple supernovae's photometry, as described in Section 10, is qualitatively different from fitting each supernova independently and comparing the results afterward. Fitting each object independently could, at best, reveal that some particular object's data looked unusual relative to the model; it could never, on its own, distinguish a genuinely unusual object from a systematic problem shared across the whole population, because each fit only has access to that one object's information. The shared-likelihood construction in Section 10.6, in which all 874 observations from 20 different supernovae jointly inform the same three parameters, is what makes this distinction possible: a consistent, repeated pattern across many independent objects is far more informative about a shared surface than the same pattern seen in only one object could ever be.

This also explains why the specific composition of the shared-SED dataset (Section 6.2) matters for what a_1 and a_2 can be expected to constrain. Broad redshift coverage helps constrain a_1 specifically, because redshift stretches the observed-frame wavelength range relative to the rest-frame wavelength range the SED is defined on; supernovae at different redshifts probe different

rest-frame wavelengths through the same fixed set of observer-frame Rubin filters, so combining a range of redshifts is what lets a wavelength-dependent tilt be distinguished from a simple offset. Similarly, broad phase coverage, meaning photometric observations spanning a wide range of p both before and after peak brightness, is necessary for a_2 to be constrained at all, since a phase tilt is by definition invisible in data taken only near $p = 0$, where the correction itself is zero regardless of a_2's value.

The posterior correlations reported in Table 8 matter physically, not just statistically, for the same reason: the strongest correlation, between a_0 and a_2, reflects a genuine, expected physical degeneracy given this dataset's particular phase coverage, rather than a coding artefact, and understanding it is part of correctly interpreting the result rather than a separate concern.

12.3 Why this is a controlled proof of concept

Every local supernova parameter, x_1, c, t_0, and z, was held fixed at its known, injected value throughout the shared-SED experiments in Sections 10 and 11. This is what makes the result a *controlled* proof of concept: it isolates the question of whether the global surface parameters can be recovered at all, in the cleanest possible setting, before asking the much harder question of whether they can still be recovered once every object's local parameters are also unknown and must be inferred simultaneously. Freeing the local parameters introduces new possible degeneracies (Section 13) that this controlled test was deliberately designed to avoid, so that a first success or failure would be unambiguous.

12.4 What the evidence does, and does not, support

The evidence in Section 11 supports a specific, bounded claim: broad-band, multi-supernova photometry, combined through an explicit shared likelihood, can constrain simple global corrections to an assumed SED surface, tightly enough that a deliberately offset prior is pulled back toward the correct, injected answer with healthy sampler diagnostics. A correctly recovered null result is informative because it demonstrates the same machinery would also have detected a genuine nonzero correction had one been present, which Section 14 proposes testing directly.

What the evidence does not yet support is any claim about practical SED reconstruction on real survey data: nonzero injected corrections have not been tested (so detection sensitivity is unknown), local supernova parameters have not been freed (so realistic degeneracies with x_1, c, and t_0 remain unprobed), and no real Rubin observations have been analyzed. Each is a distinct, necessary step (Section 14) before the current result could support a broader scientific claim.

13 Limitations

Several limitations apply to the results in this report, each stated with its practical consequence rather than left implicit.

- **Simulation-only results.** All results come from the same forward model being tested, so they demonstrate internal consistency, not yet validity against reality.
- **Type Ia only, no P(Ia) classification.** No non-Ia population or classifier exists; BEAMS-compatible classification (Section 2) remains a longer-term goal, not current work.
- **No real Rubin observations.** All photometry analyzed is simulated; performance on real data is not yet demonstrated.
- **Local parameters fixed in the shared-SED experiments.** x_1, c, t_0, z were treated as exactly known (Sections 10.1, 11), so simultaneous recovery of local and global parameters has not been tested.

- **Possible degeneracies once local parameters are freed.** a_0 may be degenerate with amplitude, absolute magnitude, or calibration; a_1 with colour c and dust; a_2 with x_1 and t_0. This is why the next experiment keeps local parameters fixed before freeing them (Section 14).
- **Single noise realization.** Validates the implementation but is not yet a population-level calibration study.
- **Zero-injection test only.** Correctly reporting no correction when none exists does not yet show the method detects a real correction when one exists.
- **Linear, low-dimensional correction only.** The model can scale and tilt in wavelength and phase, but cannot represent a correction localized to one wavelength range or phase, nor any nonlinear deviation.
- **Unequal observations per supernova.** One dense object contributed over a third of all visits (Section 10.1.1), so the posterior is not equally informed by every object.
- **Selection effects not fully modelled.** The measured-SNR bins (Section 6.1) are a diagnostic, not a complete Rubin selection function.
- **Normalizing-flow undercoverage.** Its raw credible intervals (Table 5) are mildly overconfident and not yet suitable for scientific use without recalibration.
- **No posterior-predictive validation.** Not yet performed for either shared-SED result; would catch model misspecification that $\hat{R}$ and ESS alone cannot.
- **Detection sensitivity unknown.** The strength at which a genuine nonzero correction becomes reliably distinguishable from noise has not been established.
- **No full hierarchical population model.** a_0, a_1, a_2 are fixed, shared unknowns, not parameters of a broader model allowing, for example, redshift evolution.

14 Future Work

The limitations above motivate a specific, ordered validation sequence:

1. Repeat the joint experiment with nonzero injected a_1, a_2, to test recovery of a genuine correction, not only a correctly reported null result.
2. Test several injected strengths, to find the minimum detectable correction given this dataset's size.
3. Repeat across several noise realizations, to test whether reported uncertainties are calibrated on average.
4. Vary the number of supernovae, to characterize how constraints tighten with sample size.
5. Vary the redshift distribution, to test whether redshift coverage specifically constrains a_1 (Section 12).
6. Vary cadence and phase coverage, to test the analogous claim for a_2.
7. Free the local colour parameter c and study the resulting a_1-c degeneracy directly.
8. Free x_1 and t_0 and study the resulting phase-related degeneracies with a_2.
9. Introduce realistic amplitude or distance uncertainty per supernova, rather than treating x_0 as exactly known.

10. Use posterior-predictive checks against simulated data generated from the posterior, beyond $\hat{R}$ and ESS alone.

11. Generalize to a smooth basis expansion,

$$M'(p,\lambda) = M(p,\lambda)\left[1 + \sum_k a_k\, g_k(p,\lambda)\right],$$

allowing progressively more flexible corrections than the current amplitude and two linear tilts.

12. Apply regularization or smoothness priors as the number of coefficients a_k grows, since too many correction parameters too quickly risks an underconstrained problem where very different corrections fit the data comparably well.

13. Test on real Rubin photometry once the steps above are validated, incorporating a realistic survey selection function at that stage.

14. Consider non-Ia simulation and BEAMS-style P(Ia) classification only after the SED-inference problem itself is validated; this remains a longer-term goal.

15 Conclusion

This project has built a realistic Rubin/SALT2 simulation foundation, validated it against official LSST depth calculations, and used it to develop an inverse-inference pipeline for Type Ia supernova photometry. Individual SALT2 parameters were successfully inferred from simulated light curves using both an interpretable χ^2 grid search and a scalable machine-learning benchmark, establishing that the forward and inverse problems are consistent with each other and that the resulting pipeline scales to a realistic, irregular-cadence dataset of tens of thousands of objects. A conditional normalizing flow provided a first joint posterior baseline over these local parameters, but its measured empirical coverage showed mild undercoverage, motivating a move to explicit, likelihood-based Bayesian inference for the project's central result.

That central result extends Bayesian inference from a single global amplitude correction, a_0, to a joint three-parameter correction, a_0, a_1, a_2, capturing overall amplitude, wavelength tilt, and phase tilt of the shared SALT2 surface. In a controlled null test, with all three corrections injected as zero, BlackJAX NUTS recovered all three parameters with every injected value contained inside its marginal 95% credible interval, and with healthy convergence diagnostics throughout, no divergences, $\hat{R}$ close to 1 for every parameter, and reasonable effective sample sizes. This shows that broad-band photometry, combined across multiple supernovae through a shared likelihood, can constrain simple, physically motivated corrections to an assumed SED surface, and that the statistical machinery for asking this question has been implemented and validated correctly on a case with a known answer.

This is explicitly framed as work *toward* SED reconstruction, not as SED reconstruction itself: the correction model is low-dimensional and linear, local supernova parameters were held fixed throughout, and only a null test has been performed so far. Nonzero injection testing, freeing the local parameters one at a time, and generalizing toward a smooth basis expansion of the SED surface are the required next steps before this pipeline could support any broader claim about reconstructing real supernova SEDs from Rubin photometry.

References

[1] Bruno Leibundgut and Mark Sullivan. "Type Ia Supernova Cosmology". In: *Space Science Reviews* 214.1 (2018), p. 57. DOI: 10.1007/s11214-018-0491-8.

[2] Željko Ivezić et al. "LSST: From Science Drivers to Reference Design and Anticipated Data Products". In: *The Astrophysical Journal* 873.2 (2019), p. 111. DOI: 10.3847/1538-4357/ab042c.

[3] Renee Hlozek et al. "Photometric Supernova Cosmology with BEAMS and SDSS-II". In: *The Astrophysical Journal* 752.2 (2012), p. 79. DOI: 10.1088/0004-637X/752/2/79. arXiv: 1111.5328.

[4] Julien Guy et al. "SALT2: using distant supernovae to improve the use of type Ia supernovae as distance indicators". In: *Astronomy & Astrophysics* 466.1 (2007), pp. 11–21. DOI: 10.1051/0004-6361:20066930.

[5] Lynne Jones and the Rubin Observatory Survey Strategy Team. *Calculating LSST limiting magnitudes and SNR*. Tech. rep. SMTN-002. Vera C. Rubin Observatory, 2023. URL: https://smtn-002.lsst.io.

[6] Boris Leistedt and David W. Hogg. "Data-driven, Interpretable Photometric Redshifts Trained on Heterogeneous and Unrepresentative Data". In: *The Astrophysical Journal* 838.1 (2017), p. 5. DOI: 10.3847/1538-4357/aa6332.

[7] Volodymyr Kuleshov, Nathan Fenner, and Stefano Ermon. "Accurate Uncertainties for Deep Learning Using Calibrated Regression". In: *Proceedings of the 35th International Conference on Machine Learning (ICML)*. Vol. 80. PMLR, 2018, pp. 2796–2804.

[8] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. "Density Estimation Using Real NVP". In: *International Conference on Learning Representations (ICLR)*. 2017. arXiv: 1605.08803.

[9] Kyle Cranmer, Johann Brehmer, and Gilles Louppe. "The Frontier of Simulation-Based Inference". In: *Proceedings of the National Academy of Sciences* 117.48 (2020), pp. 30055–30062. DOI: 10.1073/pnas.1912789117.

[10] Radford M. Neal. "MCMC Using Hamiltonian Dynamics". In: *Handbook of Markov Chain Monte Carlo*. Ed. by Steve Brooks et al. Chapman and Hall/CRC, 2011. arXiv: 1206.1901.

[11] Matthew D. Hoffman and Andrew Gelman. "The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo". In: *Journal of Machine Learning Research* 15.47 (2014), pp. 1593–1623.

[12] Alberto Cabezas et al. "BlackJAX: Composable Bayesian inference in JAX". In: *arXiv preprint* (2024). arXiv: 2402.10797.

[13] Ashish Vaswani et al. "Attention Is All You Need". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 30. 2017. arXiv: 1706.03762.

A Technical Appendix

This appendix records implementation details for reproducibility, and maps the project scripts referenced throughout this report to the stage of the pipeline each one performs. These are project-specific scripts written for this pipeline; they are not part of `cosmographi` itself, which is a shared, pre-existing framework this project builds on rather than authored.

A.1 Preprocessing transformations

As described in Sections 8 and 9: observation time is stored relative to each object's first real visit; flux is transformed with `arcsinh`, which remains well-defined for the negative, noise-dominated values low-signal-to-noise observations regularly produce; flux uncertainty is transformed with `log1p`; band identity is represented as a learned embedding rather than a raw numeric code; and every sequence is padded to a fixed maximum length with an accompanying boolean mask distinguishing real observations from padding.

A.2 MCMC configuration

The configuration used for every BlackJAX NUTS run reported in Sections 10 and 11: four independent chains, 1,000 warm-up steps per chain (during which step-size and mass-matrix adaptation occur, as described in Section 10.7), and 2,000 retained samples per chain, for 8,000 total retained posterior samples. Chains were initialized at different starting positions to support the convergence check described in Section 10.7.

A.3 Project script map

Table 9 lists the project scripts referenced during this work and the pipeline stage each corresponds to.

Table 9: Project scripts and the pipeline stage each performs.

Script	Stage
`generate_dataset_20k.py`	Individual-parameter dataset generation (Section 6.1)
`benchmark.py`	Machine-learning point-prediction benchmark (Section 8)
`train_normalizing_flow.py`	Conditional normalizing-flow training and evaluation (Section 9)
`generate_global_a0_dataset.py`	Shared-SED dataset generation (Section 6.2)
`global_a0_salt2.py`	Global amplitude-correction SED model (Section 10.1)
`infer_global_a0_blackjax.py`	One-parameter a_0-only BlackJAX NUTS inference (Section 10.1)
`infer_global_a0_a1_blackjax.py`	Two-parameter a_0-a_1 BlackJAX NUTS inference
`infer_global_a0_a1_a2_blackjax.py`	Joint three-parameter BlackJAX NUTS inference (Section 11)

A.4 Marginal versus joint credible regions

A *marginal* credible interval for one parameter (Table 6) considers that parameter's posterior samples alone. A *joint* credible region considers several parameters together, accounting for correlation (Table 8); it is generally not the rectangular product of the marginal intervals, since correlated parameters trace a tilted region rather than a rectangle. Formal joint contours require an explicit density estimate or contouring procedure; as noted in Section 11, none was performed here, so any pairwise scatter plot of samples should be read as raw samples, not a computed contour.