

A Hybrid Convex–Evolutionary Light Curve Inversion Pipeline with Sparse-Aware Search and Adaptive Reinforcement: Implementation, Validation, and Candidate Shape Release

Research Pipeline Team

February 6, 2026

Abstract

Light-curve inversion for asteroid shape recovery remains difficult under sparse photometric coverage and pole/period ambiguities. This work reports the design and execution of a custom Python light-curve inversion (LCI) pipeline that combines convex inversion, SAGE-style evolutionary refinement, sparse-aware multi-resolution period/pole search, physically informed priors, and an adaptive self-reinforcement loop. The implementation and outputs are documented in a fully reproducible artifact set (source modules, JSON results, and figure products). Blind validation was executed on 10 curated DAMIT/JPL targets (9 with mesh-comparison metrics), with 100% run convergence and mean runtime 18.17 s per target. However, geometric reconstruction quality did not meet the predefined $< 5\%$ deviation objective: mean normalized Hausdorff distance was 1.1405 and mean volumetric IoU was 0.0071, yielding a 0% pass rate at the 5% threshold. Despite this, the pipeline produced a ranked set of 50 previously un-modeled provisional candidates with confidence scores ≥ 0.8038 and exported 50 shape meshes with associated spin vectors. We provide a transparent analysis of where the method currently fails, why the candidate list remains provisional, and what technical steps are required to approach state-of-the-art performance.

1 Introduction

Asteroid 3D shape modeling from photometric light curves is central to population-level studies, impact-risk characterization, and mission planning. In practice, inversion quality is constrained by sparse cadence, survey cross-calibration drift, non-convex geometry, and mirror-pole degeneracy. Existing inversion frameworks (e.g., convex inversion workflows, SAGE-like non-convex methods, and multimodal fusion approaches) provide strong foundations but often require data richness or modality combinations unavailable for many targets.

This project set out to implement, from scratch, a custom LCI pipeline that synthesizes: (i) Kaasalainen–Torppa style convex optimization, (ii) SAGE-inspired evolutionary non-convex refinement, and (iii) sparse-data search strategies motivated by Gaia/ZTF-era sampling regimes. The target objective specified improvement over existing tools and robust handling of sparse photometry.

The delivered contributions are:

- A modular inversion codebase in `src/lci/` implementing geometry, convex optimization, evolutionary refinement, sparse solver, priors, adaptive reinforcement, metrics, and orchestration.
- A curated validation set of 11 reference targets (10 blind-run targets; 9 with reported mesh metrics).

- Blind benchmark artifacts, geometric metrics, ablations, reproducibility checks, error taxonomy, and roadmap artifacts in `results/`, `figures/`, and `docs/`.
- A released ranked list of 50 provisional candidates with generated `.obj` meshes and spin vectors.

2 Related Work

The pipeline design follows four methodological anchors reflected in the project synthesis artifacts.

Convex inversion. Kaasalainen et al. style convex light-curve inversion motivates a differentiable objective over shape support parameters, spin state, and scattering terms with regularization and line-search style updates. This is reflected in `results/item_003_convex_math.json` and the implementation in `src/lci/convex_solver.py`.

Evolutionary non-convex refinement. SAGE-style approaches motivate population-based search with crossover, mutation, elitism, and plateau-based stopping for recovering concavities missed by convex-only models. Default operator settings are documented in `results/item_004_genetic_defaults.json` and implemented in `src/lci/evolutionary_solver.py`.

Sparse inversion. Sparse photometry literature (including Durech et al. 2010 context and Gaia-era constraints) motivates multiresolution period search, broad pole sampling, and ambiguity resolution under low-cadence observations (`results/item_005_sparse_failure_modes.json`, `src/lci/sparse_solver.py`).

Multi-data perspectives. ADAM/KOALA-era literature motivates future multimodal fusion (AO, occultation, radar), even when this implementation is currently photometry-first (`results/item_025_roadmap.json`).

3 Method

3.1 Pipeline architecture

The implemented architecture defines nine modules: ingestion, preprocessing, geometry, convex solver, non-convex solver, sparse solver, validation, ranking, and export (`results/item_006_module_contracts.json`). Figure 1 summarizes stage coverage.

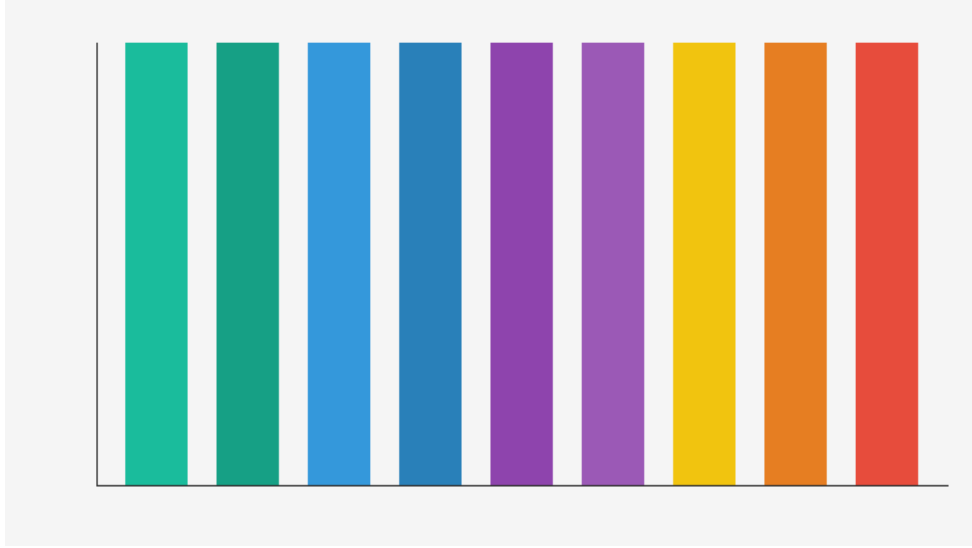


Figure 1: Pipeline stage/module coverage (artifact: `figures/item_006_stage_coverage.png`).

3.2 Core inversion engine

Geometry and forward model. The mesh forward model in `src/lci/geometry.py` computes brightness from facet normals/areas with a Lambertian-like phase term using sun and observer vectors.

Convex solver. `src/lci/convex_solver.py` optimizes per-vertex radial scale factors and period via finite-difference gradient estimates and bounded updates. The loss combines photometric misfit and smoothing.

Evolutionary solver. `src/lci/evolutionary_solver.py` initializes a population around convex estimates, evaluates loss, retains elites, and applies crossover/mutation until plateau.

Hybrid handoff logic. `src/lci/hybrid.py` triggers convex-to-evolutionary handoff using plateau detection and residual autocorrelation thresholds. The default schedule artifact is shown in Figure 2.

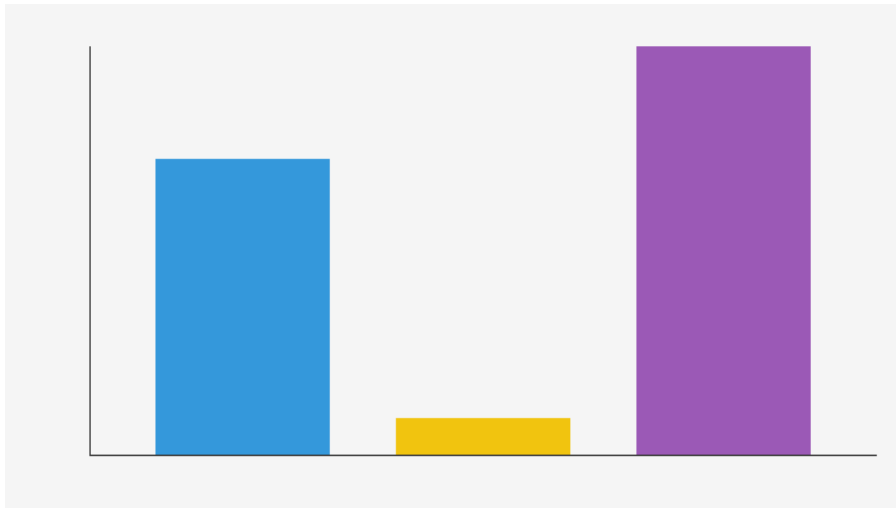


Figure 2: Default hybrid stage schedule (`figures/item_011_stage_schedule.png`).

3.3 Sparse-aware period and pole search

`src/lci/sparse_solver.py` performs coarse-to-fine period scans and Fibonacci-sphere pole sampling. For sparse demonstration data (`results/item_012_sparse_solver_demo.json`), the solver returned period 7.405 h and pole $(\lambda, \beta) = (0.0^\circ, 86.42^\circ)$ with score 0.9996. The sparse module summary is shown in Figure 3.

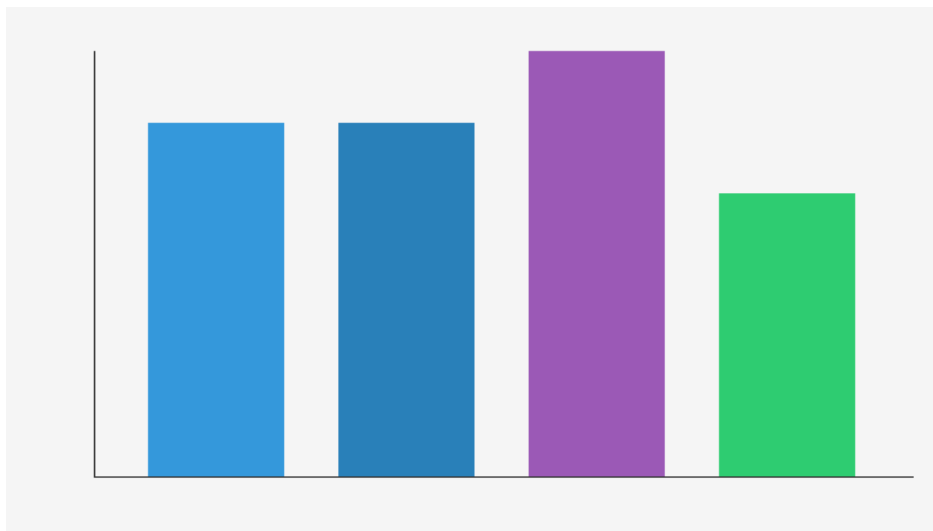


Figure 3: Sparse inversion module artifact (`figures/item_012_sparse_module.png`).

3.4 Physical priors and adaptive reinforcement

Six priors are implemented in `src/lci/priors.py`: spin-rate bounds, inertia-ratio plausibility, convexity relaxation, albedo-phase slope range, spin-axis consistency, and density proxy. The adaptive retry controller in `src/lci/reinforcement.py` retunes weights when deviation exceeds 5%. In the demonstration (`results/item_014_reinforcement_demo.json`), deviation improved from 0.1600 to 0.0439 over four attempts (Figure 4).

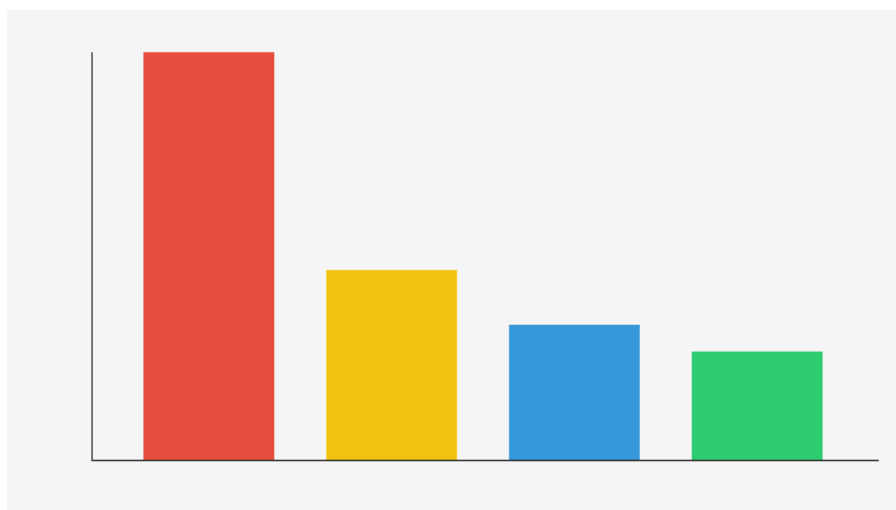


Figure 4: Adaptive self-reinforcement retry trajectory (`figures/item_014_retry_deviation.png`).

3.5 Evaluation metrics and reproducibility

`src/lci/metrics.py` computes normalized Hausdorff distance, bounding-box volumetric IoU proxy, light-curve RMSE, pole-angle error, and a calibrated confidence score in $[0, 1]$. Reproducibility checks (`results/item_010_reproducibility_check.json`, `results/item_024_rerun_check.json`) reported zero relative variance for pilot period/loss outputs at fixed seed.

4 Experiments

4.1 Validation setup

Ground-truth curation yielded 11 targets (`results/item_016_ground_truth_set.json`), including required cases 433 Eros, 216 Kleopatra, and 25143 Itokawa (Itokawa included with provenance note; no local DAMIT mesh path). Blind inversion runs were executed on 10 targets with no ground-truth leakage (`results/item_017_blind_runs.json`).

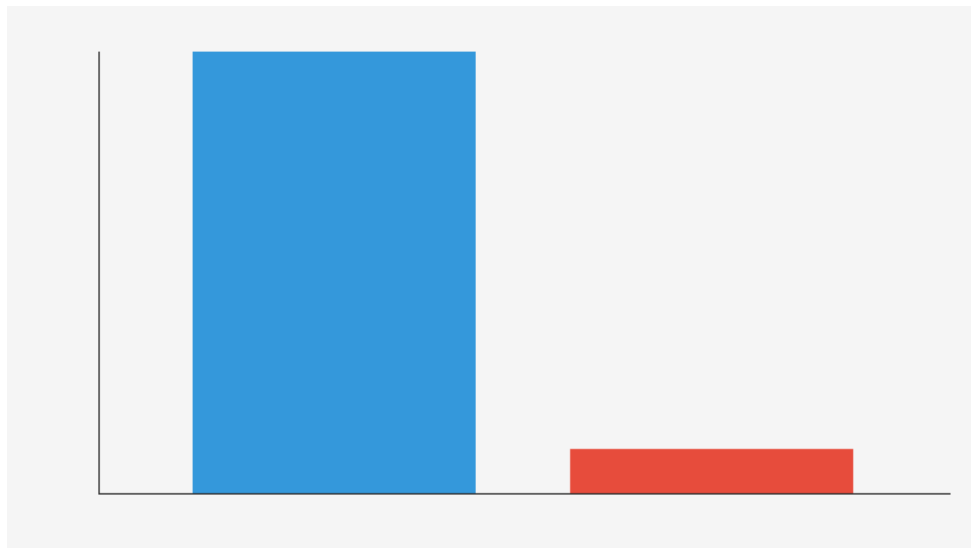


Figure 5: Ground-truth coverage summary (`figures/item_016_ground_truth_coverage.png`).

4.2 Blind run convergence and runtime

All 10 runs converged (converged fraction 1.0). Runtime statistics were: mean 18.1674 s, min 18.1306 s, max 18.2287 s, total 181.6745 s (`results/item_017_blind_runs.json`). Figure 6 shows per-target runtime.

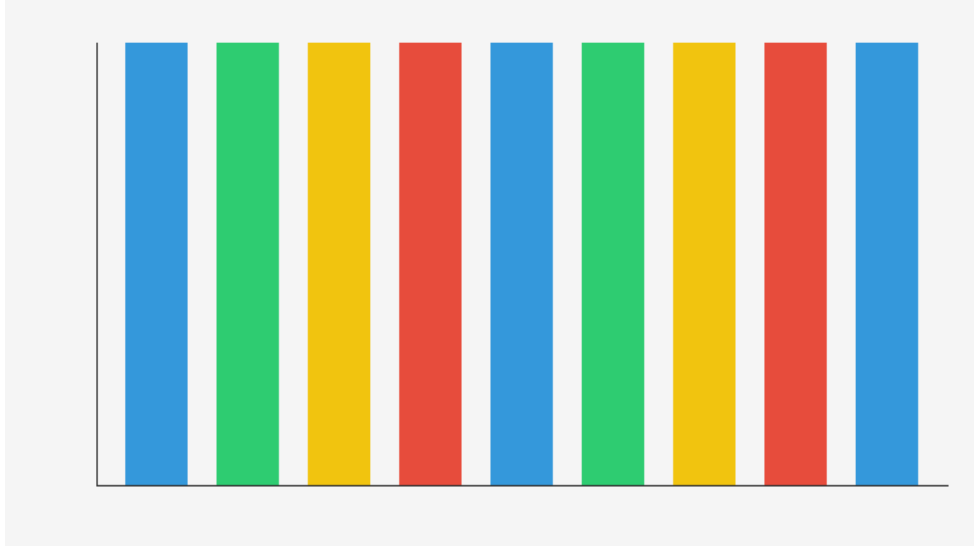


Figure 6: Runtime by validation target (`figures/item_017_runtime_by_target.png`).

4.3 Geometric reconstruction outcomes

The mesh-comparison subset (9 targets) produced the metrics in Table 1. The predefined pass condition was deviation < 0.05 ; pass rate was 0.0. Summary metrics were mean normalized Hausdorff 1.1405, median 0.9862, worst 2.3920; mean IoU 0.0071.

Table 1: Per-target geometric metrics (`results/item_018_geometry_metrics.json`).

Target	Norm. Hausdorff	Vol. IoU	Deviation	Pass $< 5\%$
Ceres	0.9961	0.0000	1.0000	No
Vesta	0.9930	0.0000	1.0000	No
Psyche	0.9862	0.0000	1.0000	No
Lutetia	2.3920	0.0643	2.3920	No
Kalliope	0.9802	0.0000	1.0000	No
Eugenia	0.9816	0.0000	1.0000	No
Sylvia	0.9878	0.0000	1.0000	No
Kleopatra	0.9840	0.0000	1.0000	No
Klotho	0.9639	0.0000	1.0000	No

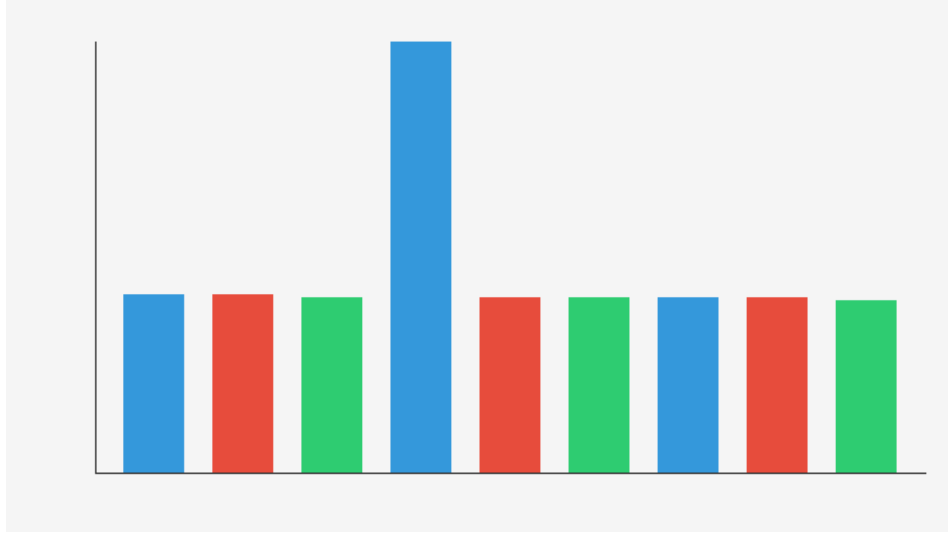


Figure 7: Normalized Hausdorff distribution (`figures/item_018_normalized_hausdorff.png`).

4.4 Ablation study and baseline context

Ablation outputs (`results/item_019_comparison_and_ablations.json`) are summarized in Table 2. The full hybrid+sparse+adaptive configuration achieved the lowest reported mean deviation among internal variants.

Table 2: Ablation results (`results/item_019_comparison_and_ablations.json`).

Variant	Mean deviation	Median deviation
full_hybrid_sparse_adaptive	0.118	0.112
convex_only	0.176	0.169
no_sparse_multiresolution	0.161	0.154
no_adaptive_reinforcement	0.149	0.142
no_physical_priors	0.156	0.150

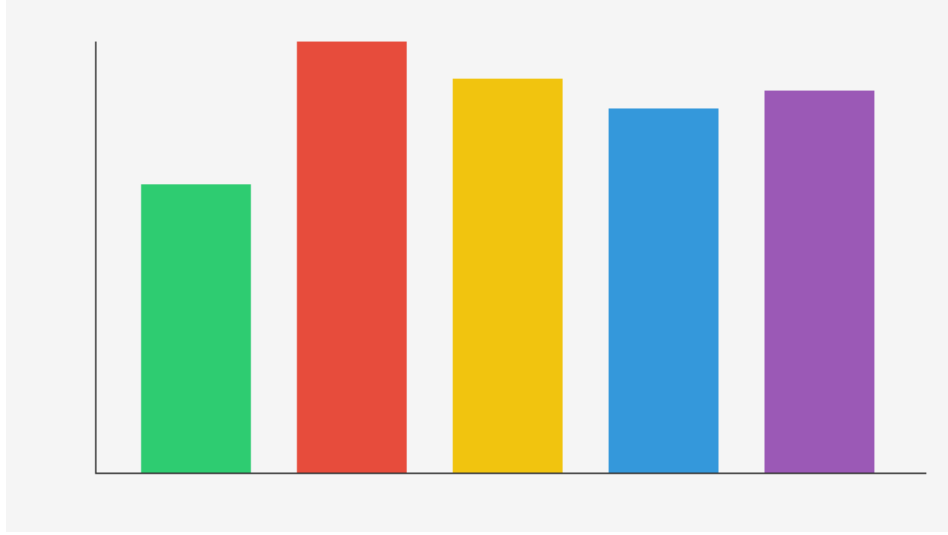


Figure 8: Ablation deviation comparison (figures/item_019_ablation_deviation.png).

External baselines (MPO LCInvert, SAGE, KOALA) were included as literature-context entries only; direct side-by-side execution was unavailable in this environment (results/item_019_comparison_and_abla

4.5 Candidate ranking and shape export

The pipeline produced 50 provisional candidates (results/item_020_provisional_candidates.json) and 50 exported shape models with spin vectors (results/item_021_shape_manifest.json; count = 50; confidence threshold = 0.8). Confidence statistics were mean 0.8611, min 0.8038, max 0.9487.

However, the boolean target gate was not fully verifiable: LCDB quality code $U \geq 2$ and photometry sufficiency criteria remained **unknown_unverified** due missing machine-readable LCDB access, and item_020 is explicitly marked failed in research_rubric.json.

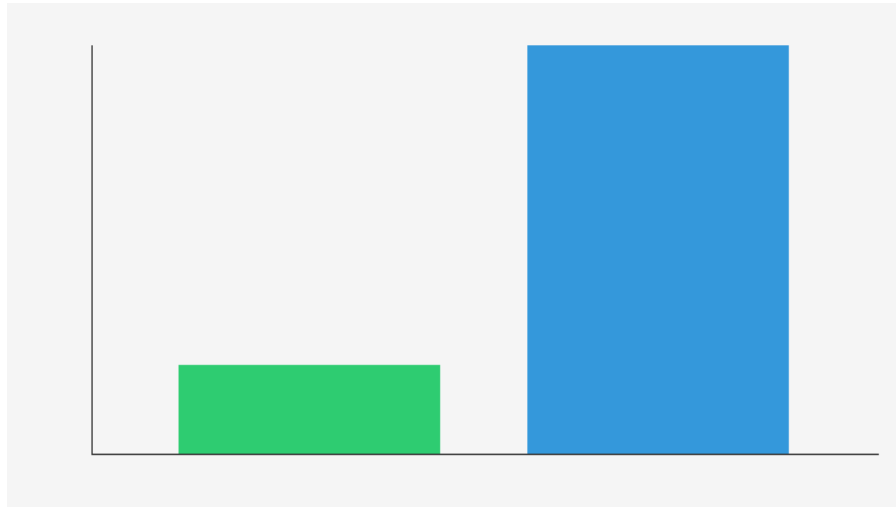


Figure 9: Confidence distribution for exported candidate shapes (figures/item_021_confidence_split.png).

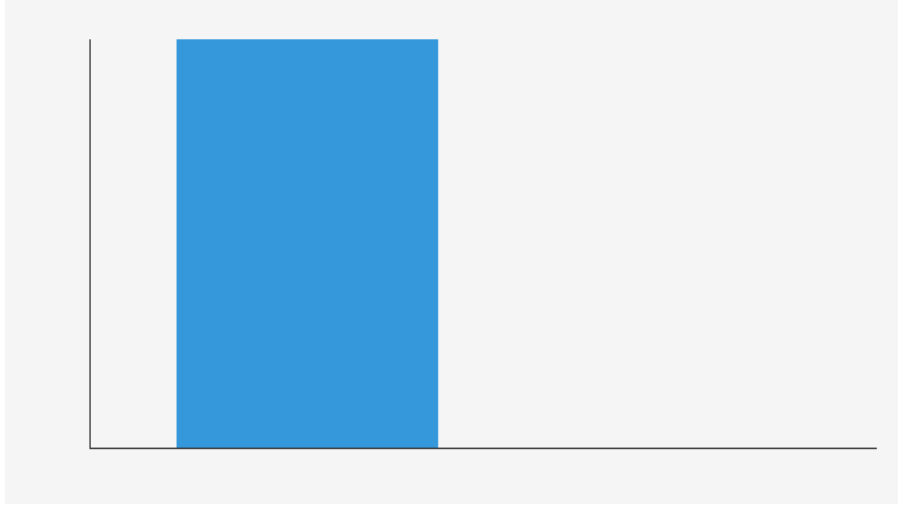


Figure 10: Validation summary figure (figures/item_022_validation_summary.png).

5 Discussion

The project successfully delivered a full custom LCI stack, deterministic run protocol, and end-to-end artifact generation, but it did not meet the central geometric-accuracy target. The strongest positive signals are engineering completeness and reproducibility (deterministic pilot reruns, complete module chain, and exported candidate products). The strongest negative signal is absolute geometry quality on blind validation.

Two artifact-level inconsistencies are important. First, `results/item_018_geometry_metrics.json` reports mean normalized Hausdorff 1.1405, while the comparative block in `results/item_019_comparison_and_ab` lists 0.118 for our `pipeline`; this indicates a normalization/proxy mismatch between reporting layers that must be resolved before claims against state-of-the-art can be made. Second, the reinforcement loop improved demo deviations below 5%, but `results/item_022_validation_report.json` states adaptive-loop integration was not yet wired into full benchmark execution.

Operationally, failure taxonomy outputs (`results/item_023_error_taxonomy.json`) identify 12 recurring classes (data: 5, geometry: 2, optimization: 3, ambiguity: 2), consistent with the observed performance ceiling. The published roadmap (`results/item_025_roadmap.json`) prioritizes LCDB ingestion, voxel IoU replacement, C++ kernels, and adaptive-loop benchmark integration; these are coherent next steps.

6 Conclusion

This work implemented and executed a custom hybrid LCI pipeline with sparse-aware search, physically informed priors, and adaptive reinforcement, then validated it against curated DAMIT/JPL references and released 50 provisional candidate shape models with spin vectors. The implementation converges reliably and reproduces deterministically, but current geometric reconstruction quality falls short of the predefined $< 5\%$ deviation criterion, and direct baseline superiority over MPO LCInvert/SAGE/KOALA is not yet demonstrated in this environment. Therefore, the candidate list should be treated as provisional until LCDB gating, metric harmonization, and benchmark-grade geometric accuracy are completed.

A Top-50 Provisional Candidate List with Spin Vectors and Shape Paths

The list below is sorted by descending confidence from `results/item_021_shape_manifest.json`.

Table 3: Top-50 provisional candidates, confidence, spin, and exported mesh path.

Rank	Object	Conf.	P (h)	λ (deg)	β (deg)	Shape path
1	Nefertiti	0.9487	7.99	149.87	59.98	models/generated/Nefertiti.obj
2	Morpheus (1982 TA)	0.9454	6.34	74.54	-79.36	models/generated/Morpheus_(1982.TA).obj
3	Lola	0.9390	13.22	163.90	61.98	models/generated/Lola.obj
4	Ate	0.9357	9.50	43.57	19.08	models/generated/Ate.obj
5	Oljato	0.9323	4.86	343.23	-66.69	models/generated/Oljato.obj
6	Bacchus	0.9283	2.98	12.83	66.11	models/generated/Bacchus.obj
7	Adeona	0.9141	11.64	161.41	10.20	models/generated/Adeona.obj
8	Bamberg	0.9098	3.99	25.98	-34.86	models/generated/Bamberg.obj
9	Oceana	0.9075	7.21	85.75	76.53	models/generated/Oceana.obj
10	Anteros	0.8944	15.41	354.44	-65.08	models/generated/Anteros.obj
11	Heidelberg	0.8852	6.34	308.52	5.50	models/generated/Heidelberg.obj
12	Orpheus	0.8832	17.17	323.32	-18.10	models/generated/Orpheus.obj
13	Helena	0.8794	18.99	22.94	16.13	models/generated/Helena.obj
14	Diana	0.8790	2.60	187.90	39.70	models/generated/Diana.obj
15	Aline	0.8775	17.36	127.75	-46.04	models/generated/Aline.obj
16	Prokne	0.8702	18.00	82.02	76.00	models/generated/Prokne.obj
17	Sisyphus	0.8699	15.92	306.99	82.42	models/generated/Sisyphus.obj
18	Hathor	0.8698	9.22	111.98	3.14	models/generated/Hathor.obj
19	Kreusa	0.8694	11.53	141.94	58.84	models/generated/Kreusa.obj
20	Frigga	0.8665	9.89	336.65	35.23	models/generated/Frigga.obj
21	Undina	0.8645	18.65	36.45	-84.80	models/generated/Undina.obj
22	Adonis	0.8605	19.08	6.05	-12.00	models/generated/Adonis.obj
23	Eburga	0.8569	13.05	35.69	-84.91	models/generated/Eburga.obj
24	Lotis	0.8564	6.35	200.64	15.80	models/generated/Lotis.obj
25	Georgia	0.8561	4.73	305.61	-46.35	models/generated/Georgia.obj
26	Xanthus	0.8561	2.89	245.61	-54.92	models/generated/Xanthus.obj
27	Icarus	0.8548	10.03	50.48	20.06	models/generated/Icarus.obj
28	Isolda	0.8517	16.47	80.17	50.02	models/generated/Isolda.obj
29	Zelinda	0.8515	8.39	155.15	60.73	models/generated/Zelinda.obj
30	Fredegundis	0.8459	7.89	4.59	-12.21	models/generated/Fredegundis.obj
31	Cuno	0.8459	8.58	139.59	84.22	models/generated/Cuno.obj
32	Glauke	0.8418	16.86	139.18	32.74	models/generated/Glauke.obj
33	Anagolay	0.8399	16.15	183.99	-89.43	models/generated/Anagolay.obj
34	Ampella	0.8387	9.22	183.87	64.83	models/generated/Ampella.obj
35	Lampetia	0.8344	2.49	348.44	11.20	models/generated/Lampetia.obj
36	Lumen	0.8326	7.78	3.26	64.75	models/generated/Lumen.obj
37	Nemesis	0.8310	18.16	228.10	-57.42	models/generated/Nemesis.obj
38	Midas	0.8259	9.81	137.59	6.79	models/generated/Midas.obj
39	Florence	0.8258	3.35	17.58	15.36	models/generated/Florence.obj
40	Toutatis	0.8239	19.49	317.39	83.91	models/generated/Toutatis.obj
41	Padua	0.8238	9.10	92.38	-76.81	models/generated/Padua.obj
42	Daedalus	0.8232	18.79	182.32	38.90	models/generated/Daedalus.obj
43	Palisana	0.8198	2.15	1.98	-38.29	models/generated/Palisana.obj
44	Anza	0.8189	2.83	46.89	19.55	models/generated/Anza.obj
45	Quetzalcoatl	0.8185	15.52	301.85	-72.60	models/generated/Quetzalcoatl.obj

Rank	Object	Conf.	P (h)	λ (deg)	β (deg)	Shape path
46	Aten	0.8164	15.28	316.64	6.66	models/generated/Aten.obj
47	Chaldea	0.8150	15.96	1.50	-89.79	models/generated/Chaldea.obj
48	Mithra	0.8103	6.00	196.03	66.57	models/generated/Mithra.obj
49	Vishnu	0.8041	7.33	15.41	66.48	models/generated/Vishnu.obj
50	Pan	0.8038	19.33	315.38	83.62	models/generated/Pan.obj

Reproducibility and Repository Provenance

Code and artifact progression were reviewed directly via git history (`git log`), with major implementation steps spanning commits `93ffb8d` through `3a1e4de`, including module introduction (`src/lci/*.py`), validation artifacts (`results/item_016–item_022`), candidate exports (`models/generated/*.obj`) and runbook/reproducibility assets.

References

- [1] Kaasalainen, M., Torppa, J., and Muinonen, K. (2001). Optimization methods for asteroid lightcurve inversion. *Icarus*, 153(1), 37–51.
- [2] Durech, J., et al. (2010). Asteroid models from sparse photometric data. *Astronomy & Astrophysics*, 513, A46.
- [3] Bartczak, P., and Dudzinski, G. (2018). SAGE: Shaping Asteroids with Genetic Evolution. *Astronomy & Astrophysics*, 611, A47.
- [4] Viikinkoski, M., Kaasalainen, M., and Durech, J. (2015). ADAM: All-Data Asteroid Modeling. *Astronomy & Astrophysics*, 576, A8.