Skepthical review: *Parameterized Manifold Learning and Sparse Tensor Train Regression for Cosmological Parameter Inference from Merger Trees*

Summary

This manuscript proposes a modular ML pipeline to infer cosmological parameters (Ω_m, σ_8) from dark-matter halo merger trees. Halo-node features (e.g., log mass, log concentration, log $V_{\max}$, scale factor) are first embedded using a "parameterized"/target-conditioned UMAP (Sec. 2.2; see also Sec. 2.1.1–2.1.2). Each tree's variable-size set of embedded nodes is then converted into a fixed-size tensor by evaluating a KDE on a fixed grid in the embedding space (Sec. 2.3), and sparse Tensor-Train (TT) regression models are trained to predict Ω_m and σ_8 from these tensors (Sec. 2.4, Sec. 2.6.2). Experiments on **1000** trees across **40** cosmologies with a cosmology-wise split (Sec. 3.1) report strong accuracy (Sec. 3.2–3.3) and offer interpretability via manifold visualizations and TT-weight "feature importance" (Sec. 3.4–3.5). However, several core methodological choices are currently ambiguous or internally inconsistent—most critically the use of target-conditioned UMAP for a task where targets are unknown at inference time (Sec. 2.2 vs. Sec. 3)—and many implementation details (UMAP/KDE/TT hyperparameters, KDE tractability in 8D, TT optimization) are under-specified or left as placeholders (Sec. 2). Addressing these points is necessary to validate that the evaluation matches the intended inference setting, to ensure reproducibility, and to support claims about robustness, scalability, and physical insight (Sec. 3–4).

Strengths

- Scientifically relevant goal: inferring Ω_m and σ_8 from complex merger-tree data beyond simple summary statistics (Sec. 1, Sec. 2.1).
- Clear modular design (embedding → distributional/tensor summary → regression) that is easy to ablate and potentially reusable (Sec. 2.2–2.4).
- Cosmology-wise train/validation/test split by unique (Ω_m, σ_8) pairs is an appropriate generalization test and reduces leakage compared to tree-wise random splits (Sec. 2.1.2, Sec. 3.1).
- Reported performance improvements over tree-level baselines (Random Forest / Gradient Boosting on aggregate statistics) are substantial on the presented split (Sec. 3.2–3.3).
- Interpretability-oriented components (manifold visualization; TT-weight-based importance; ablations) are a good direction for connecting ML outputs to astrophysical questions (Sec. 2.6, Sec. 3.4–3.5).

Major issues

1. **Potential label leakage / ill-defined inference setting due to target-conditioned ("parameterized") UMAP. Sec. 2.2 describes an embedding conditioned on the target cosmological parameters using UMAP's supervised/target mechanism (e.g., `target_weight`, `target_metric`). But at inference time the cosmological parameters are unknown (they are what the model must predict). It is currently unclear whether, during `.transform()` on validation/test nodes, the true (Ω_m, σ_8) values are provided to UMAP (which would leak labels into the representation), or whether a different procedure is used (which could materially change results and the "global consistency across cosmologies" claim in Sec. 3.4 and Sec. 4).**

   Recommendation: Make the information flow explicit with a step-by-step description (training vs. validation/test) in Sec. 2.2 and Sec. 3.1: (i) confirm UMAP is fit only on training nodes; (ii) state exactly what is passed as y/targets during fit and during $transform$ for val/test. If true targets are used at val/test, redo the evaluation without target access (to match the intended inference task). Include an ablation with unsupervised UMAP (no target conditioning) and report the performance gap (Sec. 3.2–3.3). If the goal is instead interpolation within a labeled grid (i.e., supervised dimensionality reduction), reframe claims accordingly and clearly state that the embedding uses cosmology labels.

2. **Methods contain numerous unresolved placeholders and inconsistencies, preventing reproducibility. Sec. 2.1.1–2.1.2, Sec. 2.2–2.5 and Tables 1–4 include "[Value]", "[e.g., 8]", "[e.g., Gaussian]", etc., while later Results tables (Sec. 3) appear more concrete. As written, readers cannot tell which settings produced the reported results (e.g., final D_{embed}, grid resolution d_k, kernel/bandwidth choice, TT ranks/regularization, baseline hyperparameters).**

   Recommendation: Replace all placeholders in Sec. 2.1–2.5 and fully populate (or remove) Tables 1–4. Add a single consolidated "Experiment configuration" table listing: dataset split, D_{embed}, UMAP hyperparameters ($n_{\text{neighbors}}$, min_{dist}, $metric$, $target_weight$, target scaling), KDE kernel + bandwidth selection rule, grid resolution per dimension, TT ranks and regularization strength(s), optimizer/ALS settings, and baseline hyperparameters. Ensure Methods and Results refer to the same authoritative configuration.

3. **KDE tensorization feasibility and exact tensor dimensionality are unclear (and may be intractable as described). With $D_{\text{embed}} = 8$ (Sec. 3.4) and, e.g., $d_k \approx 6$ bins per dimension (Sec. 2.3), the grid has $6^8 \approx 1.68$ million cells per tree. Storing/evaluating this densely for 1000 trees is potentially many GB and expensive; the manuscript does not explain memory layout, sparse storage, or how KDE evaluation avoids the full grid cost (Sec. 2.3, Sec. 4).**

Recommendation: In Sec. 2.3 and Sec. 3 (new short "Computational cost" subsection), report the exact D_{embed} and $(d_1, \ldots, d_D)$ used for the main results; estimate tensor size per tree; state whether tensors are stored densely or sparsely; and provide runtime/memory measurements for (i) UMAP fit/transform, (ii) KDE evaluation per tree, and (iii) TT training. If TT is intended to avoid dense materialization, explain concretely how you compute $\langle W, H_i \rangle$ without explicitly storing all of H_i, or clarify that tensors are in fact dense and show it is feasible on the reported hardware.

4. **"Adaptive KDE" is described ambiguously and appears inconsistent with the cited implementation. The text calls the KDE step "adaptive" (Sec. 2.3, Sec. 2.6.2, Sec. 4) but also references `sklearn.neighbors.KernelDensity`, which is fixed-bandwidth. Bandwidth selection (global vs per-tree vs per-point), whether bandwidths vary across embedding regions, and boundary handling on a finite grid are not specified, even though this representation is central.**

   Recommendation: Clarify in Sec. 2.3 whether the KDE is (a) fixed-bandwidth, (b) per-tree bandwidth selected by CV, or (c) truly adaptive (e.g., balloon/sample-point bandwidth based on kNN distances). If adaptive, describe the exact rule, hyperparameters, and how it is implemented with (or beyond) scikit-learn. If fixed-bandwidth, remove "adaptive" language throughout and specify the bandwidth selection procedure (search range, CV criterion on training data). Also state whether you store densities or cell masses (density $\times$ cell volume) and how normalization is handled (see also minor issue on KDE normalization).

5. **Sparse TT regression is under-specified (configuration, optimization, and what sparsity means), limiting reproducibility and weakening interpretability claims. Sec. 2.4 omits TT-rank choices, initialization, the exact regularized objective, and how ALS handles an L1 penalty (ALS subproblems become non-smooth and need a specific solver/prox step). Moreover, L1 sparsity in TT cores does not automatically imply localized sparsity in the reconstructed full weight tensor W, so the interpretation "top bins by $|W|$" (Sec. 3.5) needs justification.**

   Recommendation: Expand Sec. 2.4 with: (i) explicit TT definition with ranks r_k; (ii) the full training objective (MSE + regularizer), specifying whether L1 is on cores or on W; (iii) the ALS/proximal update used for L1 (or cite a specific sparse TT/MPS regression method and match it); (iv) rank/λ selection procedure and validation protocol; and (v) training/validation curves or train–test gaps to assess overfitting with $\sim$800 training trees (Sec. 3.1). For interpretability (Sec. 3.5), add a sanity check: reconstruct representative slices/marginals of W (or compute effective per-bin weights) and show that "important bins" are stable across seeds/splits.

6. **Baseline comparisons are likely too weak to support broader performance/novelty claims. Sec. 2.5.2 and Sec. 3.3 compare mainly to Random Forest / Gradient Boosting on 20 aggregate statistics. This does not test whether gains come from (i) UMAP conditioning, (ii) using a distributional/histogram representation, or simply (iii) using a much richer representation than the baselines. It also omits strong set-/graph-based baselines commonly used for irregular astrophysical data.**

   Recommendation: In Sec. 2.5.2 and Sec. 3.3, add at least one stronger baseline that operates on node-level data under the same cosmology-wise split, e.g.: Deep Sets (pooled MLP), Set Transformer, or a simple GNN on the merger-tree graph (if edges are available). Also add a "fair summary" baseline closer to your representation, e.g., histogram/KDE in the original 4D feature space (or PCA space) without UMAP, to isolate the contribution of manifold learning. If adding baselines is infeasible, explicitly limit claims to the tested baselines and adjust wording in Sec. 3–4 accordingly.

7. **Statistical robustness, uncertainty, and split sensitivity are not evaluated, which is important given only 40 cosmologies. Sec. 3.2–3.3 report single-point MSE/R² on one 32/4/4 cosmology split. With 40 cosmologies, results may vary depending on which cosmologies are held out, and "parameter inference" typically requires uncertainty quantification rather than point prediction alone.**

   Recommendation: Repeat experiments across multiple random cosmology splits or perform k-fold cross-validation over the 40 (Ω_m, σ_8) pairs (Sec. 3.1–3.3), reporting mean±std (or confidence intervals) for MSE/R^2. Add uncertainty estimates for predictions (e.g., bootstrap over trees, ensembling over seeds, or conformal intervals) and report calibration diagnostics if framing as "inference." Provide predicted-vs-true scatter plots per held-out cosmology to reveal systematic biases (Sec. 3.2–3.3).

8. **Merger-tree information content vs. "bag of nodes" is unclear, affecting scope and interpretation. The current pipeline appears to ignore explicit parent–child edges/topology after extracting nodes and a scale-factor feature (Sec. 2.1.1, Sec. 2.3). Given the title emphasizes "merger trees," readers need to know whether the method uses the tree structure or only the multiset of nodes across time, and what is lost by discarding topology.**

   Recommendation: State explicitly in Sec. 2.1.1 and/or Sec. 2.3 whether edges/topology are used anywhere downstream. If not, rephrase claims to avoid implying topological modeling, and discuss limitations (Sec. 4). Optionally include a topology-aware baseline (e.g., GNN) or add simple topology/MAH engineered features (formation time, main-branch mass accretion history, branching ratios) to test whether explicit structure improves inference.

9. **Dataset provenance and astrophysical context are insufficiently described, limiting reproducibility and scientific interpretation. Sec. 2.1 and Sec. 3.1 list counts and parameter ranges but not the simulation suite, box size, mass resolution, redshift outputs, halo finder, tree-building method, or selection criteria. This also prevents assessing sensitivity to known systematics (resolution, halo finder, tree construction).**

   Recommendation: Add a dedicated dataset subsection in Sec. 2.1 describing: simulation code/name, box size, particle mass, snapshot/redshift range, halo finder and parameters, tree builder, cosmology grid design, and selection cuts (mass thresholds, centrals vs satellites, pruning). Report the distribution of nodes per tree (median/IQR) and how many snapshots contribute. In Sec. 4, discuss how these choices might affect transfer to other simulations or to observations.

Minor issues

1. Preprocessing and normalization are ambiguous for both inputs and targets. Sec. 2.1.1 and Sec. 2.2 do not clearly state whether each node inherits its parent tree's (Ω_m, σ_8) label, how features are standardized (and whether statistics are computed on training data only), and whether Ω_m/σ_8 are scaled for UMAP conditioning and/or TT regression.

   Recommendation: In Sec. 2.1.1–2.2, explicitly define $\boldsymbol{X}_{\text{all nodes}}$ and $\boldsymbol{Y}_{\text{all nodes}}$ (node labels inherited from tree labels), specify feature scaling (fit on training nodes only), and specify target scaling (if any) separately for (i) UMAP conditioning and (ii) regression targets. Clarify any per-tree/per-cosmology weighting when sampling nodes for UMAP fitting.

2. KDE tensor semantics and normalization are not specified. KDE evaluates a density with units inverse-volume in embedding space, but the tensor $\boldsymbol{H}_i$ is described as a "PDF/fingerprint" without stating whether values represent densities, probability masses per grid cell (density × cell volume), or a normalized tensor. This affects comparability across different grid resolutions $\boldsymbol{d}_k$ (Sec. 2.3).

   Recommendation: In Sec. 2.3, state exactly what $\boldsymbol{H}_i$ stores (density vs mass), whether $\sum \boldsymbol{H}_i = 1$ is enforced, and whether features are standardized across trees. If you want invariance to grid resolution, use (or justify not using) density×cell-volume and/or normalization.

3. Interpretability claims are currently more qualitative than quantitative. Sec. 3.4–3.5 discuss manifold structure and "important bins" but do not provide a clear, quantitative mapping from important embedding bins back to original physical ranges (mass, concentration, V_{max}, scale factor/redshift), nor stability of importance across seeds/splits.

Recommendation: Augment Sec. 3.4–3.5 with quantitative summaries: correlations between UMAP coordinates and physical variables, distributions of original features for nodes falling into top-importance bins vs. low-importance bins, and stability of the selected bins across seeds/splits. Provide at least one concrete example (table/figure) translating a set of high-weight bins into physical regimes (e.g., mass and redshift ranges).

4. Ablation protocol could be strengthened and clarified. Sec. 3.5 describes keeping "top X%" of features by $|\boldsymbol{W}|$, but it is unclear what constitutes a "feature" (full 8D grid cell?), whether masking is applied to $\boldsymbol{H}$, $\boldsymbol{W}$, or both, and how performance varies with sparsity level.

   Recommendation: In Sec. 3.5, define the ablated unit precisely (grid bin / tensor cell), report the number of nonzero/active bins per tree, and show performance as a function of retained fraction (e.g., $\mathbf{1\%}$, $\mathbf{5\%}$, $\mathbf{10\%}$, $\mathbf{20\%}$, $\mathbf{50\%}$). Add a random-mask control at matching sparsity to demonstrate that importance-based masking is meaningful.

5. Error metrics are not contextualized relative to parameter ranges. Sec. 3.2–3.3 report MSE and R^2, but RMSE/MAE and relative errors would be easier to interpret given $\boldsymbol{\Omega_m \in [0.1, 0.5]}$, $\boldsymbol{\sigma_8 \in [0.6, 1.0]}$ (ranges appear in multiple places).

   Recommendation: Report RMSE (and optionally MAE) alongside MSE and R^2 in Sec. 3.2–3.3, and express typical errors as a fraction of the parameter range. Add predicted-vs-true plots and/or per-cosmology residual summaries for the held-out cosmologies.

6. Redundancy/inconsistency in tables across Sec. 2 and Sec. 3. Dataset/split tables in Sec. 2 include placeholders while Sec. 3.1 includes final numbers; this duplication can confuse which values are authoritative.

   Recommendation: Remove or update placeholder tables in Sec. 2 and keep one authoritative dataset/split table (either in Sec. 2.1.2 or Sec. 3.1). Ensure all in-text references point to that single source of truth.

7. Mathematical definitions are often verbal rather than explicit. Key objects (UMAP objective with targets; KDE estimator; TT decomposition; regularized optimization problem) are described informally (Sec. 2.2–2.4).

   Recommendation: Add explicit equations with indices in Sec. 2.2–2.4: the UMAP supervised/target term and how (Ω_m, σ_8) enters; KDE formula and normalization; TT factorization of $\boldsymbol{W}$ with TT-ranks; and the full regularized loss minimized (including how L1 is applied). This will also help readers verify correctness and implement variants.

8. Related-work framing under-emphasizes cosmology/astro ML alternatives. Sec. 1 focuses more on UMAP/KDE/TT components than on alternative cosmological inference pipelines (emulators, likelihood-free inference, field-level methods, set/graph

models for halos/trees).

Recommendation: Expand Sec. 1 to position the contribution relative to: (i) summary-statistic emulators, (ii) likelihood-free inference approaches, and (iii) set/graph-based models for halo catalogs/merger trees. Clarify what is novel (representation + interpretability via TT sparsity) and what is a design choice among alternatives.

Very minor issues

1. Typographical/formatting issues: broken words at line breaks, inconsistent hyphenation ("high dimensional" vs "high-dimensional"), and inconsistent quotation/scare-quote usage (Sec. 1–3).

   Recommendation: Proofread and standardize typography and hyphenation throughout; remove unnecessary scare quotes; ensure consistent formatting for emphasized terms.

2. Section heading formatting is inconsistent (e.g., stray markdown-style hashes in Sec. 2–3 headings).

   Recommendation: Normalize headings to the target publication style and ensure "Sec. X.Y" references match the compiled numbering.

3. API/class names and citations appear inconsistently formatted in places (e.g., `KernelDensity` vs variants; mixed citation years/labels).

   Recommendation: Standardize code/API formatting (backticks or) and audit the reference list so in-text citations match bibliography entries (years, author spellings, disambiguation such as 2024a/2024b).

4. UMAP dimension for visualization vs. main embedding is not clearly explained (Sec. 2.6.1 vs. Sec. 3.4).

   Recommendation: Clarify whether visualizations use a separate $D_{\text{embed}} = 2/3$ run or a projection of the main $D_{\text{embed}} > 3$ embedding (and specify the projection method).

Key statements and references

- ✔ **Merger trees are inherently complex, graph-like data structures with variable numbers of nodes and intricate hierarchical relationships, and directly applying standard machine learning techniques to such irregular, high-dimensional, multi-scale data is often inefficient or leads to loss of crucial structural information (Nguyen et al., 2025; Hui et al., 2018; Robles et al., 2022; Nguyen et al., 2024).**
- *Reference(s):* Nguyen et al., 2025, Hui et al., 2018, Robles et al., 2022

- *Justification:* Nguyen et al. (2025) describe merger trees as hierarchical, tree-like graphs and state that generic graph generative models require fixed graph sizes, making them poorly suited to the variable branching of merger trees; they also note that image-based GANs (as in Robles et al., 2022) cannot enforce key physical constraints and limit redshift flexibility, i.e., they lose structural information. Hui et al. (2018) treat merger trees as graphs and emphasize that adapting standard ML/feature extraction to such data is nontrivial, requiring engineered encodings of tree structure into vectors before SVMs can be applied. Robles et al. (2022) further illustrate the need to coerce trees into fixed-size matrices (limited by branch count and memory), underscoring the mismatch between standard ML inputs and irregular, multi-scale merger trees.

- ⚠ **Adaptive Kernel Density Estimation has been proposed and applied in astrophysical contexts, including gravitational-wave data analysis and gamma-ray astronomy, to improve sky-map or density estimation by using bandwidths that adapt to local data density (Falxa et al., 2022; Holler et al., 2024).**

- *Reference(s):* Falxa et al., 2022, Holler et al., 2024
- *Justification:* Holler et al., 2024 proposes and applies an adaptive KDE method in gamma-ray astronomy, using event-wise kernel widths based on estimated direction uncertainty to improve sky maps. The paper does not discuss gravitational-wave data analysis, nor does it frame the bandwidth adaptation as based on local data density. Thus only the gamma-ray astronomy part is supported.

- ✔ **Tensor Train (matrix-product-state) decompositions provide an efficient representation for high-dimensional tensors and have been successfully used for time-series and other machine-learning tasks, mitigating the curse of dimensionality (Moore et al., 2025; Chen et al., 2023).**

- *Reference(s):* Moore et al., 2025, Chen et al., 2023
- *Justification:* Moore et al., 2025 explicitly identify matrix-product states as tensor trains and state they capture correlations with polynomial (tractable) scaling, truncating an exponentially large space, and demonstrate successful time-series machine-learning tasks (classification and imputation) with competitive performance. Chen et al., 2023 describes TT as an efficient representation whose storage scales $O(dnr^2)$ rather than exponentially, stating TT is not affected by the curse of dimensionality, and presents efficient randomized algorithms with experiments on real data. Together, they support that TT/MPS efficiently represent high-dimensional tensors and have been successfully used in time-series and other ML contexts, mitigating the curse of dimensionality.

- ✔ **Recent work has demonstrated that domain-generalized or symmetry-preserving machine-learning methods can accurately infer cosmological parameters from large-scale-structure data and galaxy surveys, motivating**

advanced architectures for cosmological inference (Lee et al., 2024; Balla et al., 2024).

- *Reference(s):* Lee et al., 2024, Balla et al., 2024
- *Justification:* Lee et al., 2024 demonstrate domain-generalized neural networks (with semantic alignment) inferring Ω_m and σ_8 directly from the SDSS BOSS LOWZ NGC survey, reporting $\Omega_m = 0.339 \pm 0.056$ and $\sigma_8 = 0.801 \pm 0.061$ and showing improved accuracy over non-generalized models and even tighter constraints for the best-adapted model. Balla et al., 2024 benchmark $E(3)$-equivariant (symmetry-preserving) GNNs for predicting Ω_m and σ_8 from simulated galaxy point clouds, showing they outperform non-equivariant baselines and are simulation-efficient; they also identify limitations for long-range correlations and show that incorporating 2PCF improves results, motivating development of advanced architectures. Together, these works support that domain-generalized or symmetry-preserving ML methods can accurately infer cosmological parameters from LSS/galaxy data and motivate improved architectures.

- **✔ Random Forest and Gradient Boosting regressors are widely used baseline models in empirical comparisons of supervised machine-learning algorithms and have been benchmarked across diverse application domains, including general tabular prediction and travel-behavior modeling (Liu et al., 2022; Shiri et al., 2025; Wang et al., 2025).**

- *Reference(s):* Liu et al., 2022, Shiri et al., 2025, Wang et al., 2025
- *Justification:* Liu et al., 2022 explicitly identify Random Forest (RF) and Gradient Boosting Machines (GBM/XGBoost) as among the most popular supervised ML methods for structured (tabular) data and benchmark them extensively against neural networks on simulated tabular prediction tasks with both continuous (regression) and binary responses (e.g., Introduction: RF, GBM, and FFNN are among the most popular SML methods; Sections 5.1–5.6 report comparative performance). Wang et al., 2025 includes Random Forests and Boosting among **12** model families (Table 2) and benchmarks them in $\mathbf{6,970}$ experiments across multiple travel demand datasets and choice settings, showing ensemble methods and DNNs often outperform DCMs. Together, these papers show RF and Gradient Boosting are widely used baseline comparators and have been benchmarked in general tabular prediction and travel-behavior modeling. (Shiri et al., 2025 is a DL survey and not needed for this support.)

Mathematical consistency audit

This section audits **symbolic/analytic** mathematical consistency (algebra, derivations, dimensional/unit checks, definition consistency).

Maths relevance: light

The paper is primarily methodological and descriptive, with very few explicit equations. The main explicit mathematical statement is a linear regression on KDE-feature tensors: $y_{\text{pred}} = \langle W, H_i \rangle + b$, with W represented in Tensor Train form. Other key mathematical components (parameter-conditioned UMAP objective, adaptive KDE formula/bandwidth, TT decomposition details, and the exact regularized optimization solved by ALS) are described verbally without explicit equations, limiting auditability of internal derivations.

Checked items

1. ✔ **Tree-count arithmetic and split consistency** (Secs. 2.1.2 (p. 3) and 3.1 (p. 6))
    - **Claim:** There are 40 unique (Ω_m, σ_8) pairs with 25 trees each (1000 trees total), split into $32/4/4$ unique pairs for train/val/test corresponding to $800/100/100$ trees.
    - **Checks:** algebra/arithmetic consistency, definition consistency
    - **Verdict:** PASS; confidence: high; impact: minor
    - **Assumptions/inputs:** 25 trees per unique cosmology, $32 + 4 + 4 = 40$ unique cosmologies
    - **Notes:** Counts match exactly: $40 \times 25 = 1000$ and $(32 \times 25, 4 \times 25, 4 \times 25) = (800, 100, 100)$.
2. ✔ **Z-score normalization definition** (Sec. 2.1.1 (p. 3))
    - **Claim:** Each raw node feature is normalized to mean 0 and standard deviation 1 using global dataset statistics.
    - **Checks:** definition consistency, dimensional/units sanity
    - **Verdict:** PASS; confidence: high; impact: minor
    - **Assumptions/inputs:** Standard normalization $x_{\text{norm}} = \frac{x-\mu}{\sigma}$ per feature
    - **Notes:** Statement is standard and internally consistent; no conflicting alternative normalization is given elsewhere.
3. ✔ **UMAP input/target tensor alignment** (Sec. 2.2 (pp. 3–4))
    - **Claim:** UMAP is trained on $X_{\text{nodes,train,norm}}$ with conditioning targets $Y_{\text{nodes,train}}$ where each node is labeled by its parent tree's (Ω_m, σ_8).
    - **Checks:** symbol/definition consistency, shape/compatibility sanity
    - **Verdict:** PASS; confidence: medium; impact: moderate
    - **Assumptions/inputs:** Each node has exactly one parent tree cosmology label, $X_{\text{nodes,train,norm}}$ and $Y_{\text{nodes,train}}$ have the same number of rows (nodes)
    - **Notes:** The mapping 'node → parent cosmology' is consistent with earlier definitions. Exact shapes are not explicitly provided but the construction described implies compatibility.
4. ✔ **KDE grid tensor shape definition** (Sec. 2.3 (p. 4))

- **Claim:** A D_{embed}-dimensional grid discretized into d_k bins per dimension yields a tensor H_i of shape $(d_1, \ldots, d_{D_{\text{embed}}})$ per tree.
- **Checks:** dimensionality/shape consistency, notation consistency
- **Verdict:** PASS; confidence: high; impact: moderate
- **Assumptions/inputs:** Each dimension k uses d_k bins, Evaluation is at grid-cell centers
- **Notes:** The stated tensor shape follows directly from the discretization. If all $d_k = 6$ and $D_{\text{embed}} = 8$, the implied feature count is 6^8, which is large but not a mathematical inconsistency.

5. ⚠ **KDE as 'probability density function' on a discretized grid** (Sec. 2.3 (p. 4))

- **Claim:** Adaptive KDE estimates a probability density function for each tree and the evaluated densities populate H_i as a tree 'fingerprint'.
- **Checks:** units/dimensional consistency, normalization/measure consistency
- **Verdict:** UNCERTAIN; confidence: medium; impact: moderate
- **Assumptions/inputs:** KDE returns a density $\hat{h}(z)$ in embedding space, Grid evaluation uses cell centers
- **Notes:** A KDE yields a density (integrates to 1 over continuous space), but the paper does not specify whether H_i stores densities or discretized probability masses (density×cell volume), nor any normalization to make H_i comparable across grid resolutions. This is an interpretational/dimensional gap rather than a provable error.

6. ✔ **Linear tensor regression formula** (Sec. 2.4 (p. 4))

- **Claim:** Predictions are computed as $y_{\text{pred}} = \langle W, H_i \rangle + b$ with W the same shape as H_i.
- **Checks:** algebraic form sanity, shape/compatibility sanity
- **Verdict:** PASS; confidence: high; impact: critical
- **Assumptions/inputs:** $\langle W, H \rangle$ denotes full tensor inner product: sum over all indices of $W \odot H$, W and H_i are conformable
- **Notes:** Given conformable shapes, the expression defines a scalar prediction and is internally consistent for both targets when trained separately.

7. ⚠ **Tensor Train representation of W** (Sec. 2.4 (p. 4))

- **Claim:** The weight tensor W is represented in TT format via cores $G_1, \ldots, G_{D_{\text{embed}}}$, reducing parameter count.
- **Checks:** notation/definition completeness, derivation verifiability
- **Verdict:** UNCERTAIN; confidence: low; impact: critical
- **Assumptions/inputs:** A TT decomposition exists for the target tensor shape, TT-ranks are specified/used in training

- **Notes:** No explicit TT index formula is provided (e.g., $W[i_1, \ldots, i_D] = G_1[:, i_1, :] \ldots G_D[:, i_D, :]$) and TT-rank symbols/ranges are not defined, so the correctness of subsequent claims about parameter counts and regularization effects cannot be verified from the PDF.

8. ⚠ **Regularized objective (MSE + L1 on cores)** (Sec. 2.4 (p. 4))
   - **Claim:** Training minimizes MSE augmented with an L1 penalty on TT core elements to induce sparsity.
   - **Checks:** objective definition completeness, symbol consistency
   - **Verdict:** UNCERTAIN; confidence: medium; impact: critical
   - **Assumptions/inputs:** Loss $L = \frac{1}{N} \sum (y_i - \langle W, H_i \rangle - b)^2 + \lambda \sum_k ||G_k||_1$ (or similar)
   - **Notes:** The exact objective is not written; it is unclear whether L1 applies to cores, to W, or to both, and whether any scaling is used. This blocks analytic checking of the optimization statements.

9. ⚠ **ALS optimization with L1 penalty** (Sec. 2.4 (p. 4))
   - **Claim:** ALS iteratively updates each TT core while keeping others fixed, minimizing the regularized MSE.
   - **Checks:** derivation/algorithmic consistency, non-smooth optimization sanity
   - **Verdict:** UNCERTAIN; confidence: medium; impact: moderate
   - **Assumptions/inputs:** Each core update solves a well-defined subproblem
   - **Notes:** With an L1 term, per-core subproblems become non-smooth (Lasso-like). The paper does not specify how these are solved within ALS, so the stated minimization procedure is mathematically underspecified.

10. ⚠ **Sparsity implies feature importance in W bins** (Sec. 2.6.2 (p. 5) and Sec. 3.5 (p. 7))
    - **Claim:** Magnitude of reconstructed W elements indicates importance of corresponding KDE bins; sparsity makes many irrelevant regions effectively zeroed.
    - **Checks:** definition consistency, implication validity
    - **Verdict:** UNCERTAIN; confidence: medium; impact: moderate
    - **Assumptions/inputs:** A well-defined reconstructed W exists from TT cores, Importance defined by $|W|$ magnitude
    - **Notes:** Using $|W|$ as an importance score is coherent for linear models. However, the linkage between L1 on cores and elementwise sparsity/interpretability of W is not formally justified in the text, so the strength of the implication is uncertain.

11. ✔ **Ablation masking procedure consistency** (Sec. 2.6.2 (p. 5) and Sec. 3.5 (p. 7))

- **Claim:** Ablating low-importance bins by setting corresponding entries in H_i to zero yields $H_{i,\text{test, ablated}}$; re-evaluating the same model measures feature relevance.
- **Checks:** algebraic consistency, shape consistency
- **Verdict:** PASS; confidence: high; impact: minor
- **Assumptions/inputs:** Mask M has same shape as H_i, $H_{\text{ablated}} = M \odot H$
- **Notes:** Masking tensor entries is algebraically consistent and compatible with $y_{\text{pred}} = \langle W, H \rangle + b$. Interpretability conclusions still depend on how importance is defined, but the masking operation itself is consistent.

12. ✔ **Visualization discussion vs chosen D_{embed}** (Sec. 2.6.1 (p. 5) and Sec. 3.4 (p. 7))
    - **Claim:** Visualization uses 2D/3D scatter plots if D_{embed} is 2 or 3, while experiments use $D_{\text{embed}} = 8$ with projections onto 2D planes.
    - **Checks:** definition consistency, logical consistency
    - **Verdict:** PASS; confidence: medium; impact: minor
    - **Assumptions/inputs:** Projection onto first two dimensions is allowed even when $D_{\text{embed}} = 8$
    - **Notes:** Not a strict contradiction: Sec. 3.4 explicitly mentions projecting 8D embeddings to 2D planes. Sec. 2.6.1 could be clarified but is not mathematically inconsistent.

Limitations

- The PDF text contains almost no explicit mathematical equations for UMAP conditioning, KDE, TT decomposition, or the ALS+L1 optimization; this prevents verifying derivations beyond basic shape/algebra sanity checks.
- No equation numbering is present in the provided PDF text, so locations are cited by section and page rather than equation numbers.
- This audit intentionally does not assess reported numerical results (MSE/R2 values), simulations, hyperparameter choices, or implementation correctness.

Numerical results audit

This section audits **numerical/empirical** consistency: reported metrics, experimental design, baseline comparisons, statistical evidence, leakage risks, and reproducibility.

19 numerical checks were executed: 8 PASS, 10 FAIL, 1 UNCERTAIN. Passes include dataset split totals (trees and unique pairs), implied tree counts from unique-pair split, feature-vector dimensionality (5 stats × 4 features = 20), and Table 5 range consistency (min ≤ mean ≤ max). Failures include the total dataset size from pairs×trees-per-pair, the ablation percentage complement, and several delta/recomputed-difference checks for R^2 and MSE changes (with sign/delta-definition inconsistencies indicated by the execution output).

Checked items

1. ✖ **C1_total_trees_from_pairs** (Page 3 (Sec. 2.1.1) and Page 6 (Sec. 3.1))
    - **Claim:** Dataset contains 40 unique (Ω_m, σ_8) pairs, with 25 merger trees simulated for each unique pair (total dataset size 1000 trees).
    - **Checks:** parts_vs_total
    - **Verdict:** FAIL
    - **Notes:** Execution output computed 65 (op recorded as sum of unique_pairs and trees_per_pair) versus reported total 1000.
2. ✔ **C2_split_counts_sum_to_total** (Page 3 (Sec. 2.1.2) and Page 6 (Sec. 3.1))
    - **Claim:** Split is 800 training + 100 validation + 100 test = 1000 total trees.
    - **Checks:** parts_vs_total
    - **Verdict:** PASS
    - **Notes:** Exact integer equality.
3. ✔ **C3_split_unique_pairs_sum** (Page 3 (Sec. 2.1.2))
    - **Claim:** Split by cosmology pairs: 32 unique pairs train, 4 validation, 4 test (total 40).
    - **Checks:** parts_vs_total
    - **Verdict:** PASS
    - **Notes:** Exact integer equality.
4. ✔ **C4_split_trees_implied_by_pairs** (Page 3 (Sec. 2.1.1–2.1.2) and Page 6 (Sec. 3.1))
    - **Claim:** Because there are 25 trees per unique (Ω_m, σ_8) pair, the split by unique pairs implies $32 \times 25 = 800$ train trees, $4 \times 25 = 100$ val trees, $4 \times 25 = 100$ test trees.
    - **Checks:** derived_counts
    - **Verdict:** PASS
    - **Notes:** Exact integer equality across train/val/test.
5. ✔ **C5_feature_vector_dim_aggregate_stats** (Page 5 (Sec. 2.5.2))
    - **Claim:** Aggregate stats: mean, std, min, max, median for each of four node features yields a 20-dimensional feature vector per tree.
    - **Checks:** dimensionality_arithmetic
    - **Verdict:** PASS
    - **Notes:** Exact integer equality.
6. ✖ **C6_ablation_percent_complements** (Page 7 (Sec. 3.5) and Page 8 (Conclusions))
    - **Claim:** Ablation keeps top 20% features and ablates remaining 80%.

- **Checks:** percentage_complement
- **Verdict:** FAIL
- **Notes:** Execution output computed **20** from kept_percent (op recorded as sum) versus reported ablated_percent **80**.

7. ✔ **C7_table5_min_le_mean_le_max_logmass** (Page 6, Table 5)
   - **Claim:** For $\log_{10}$(mass), min= **9.02**, mean= **11.53**, max= **14.07** (should satisfy min ≤ mean ≤ max).
   - **Checks:** range_consistency
   - **Verdict:** PASS
   - **Notes:** Ordering condition satisfied.
8. ✔ **C8_table5_min_le_mean_le_max_logconc** (Page 6, Table 5)
   - **Claim:** For $\log_{10}$(concentration), min= **0.15**, mean= **0.82**, max= **1.55** (should satisfy min ≤ mean ≤ max).
   - **Checks:** range_consistency
   - **Verdict:** PASS
   - **Notes:** Ordering condition satisfied.
9. ✔ **C9_table5_min_le_mean_le_max_logvmax** (Page 6, Table 5)
   - **Claim:** For $\log_{10}(V_{\max})$, min= **1.58**, mean= **2.51**, max= **3.52** (should satisfy min ≤ mean ≤ max).
   - **Checks:** range_consistency
   - **Verdict:** PASS
   - **Notes:** Ordering condition satisfied.
10. ✔ **C10_table5_min_le_mean_le_max_scalefactor** (Page 6, Table 5)
    - **Claim:** For scale factor, min= **0.10**, mean= **0.63**, max= **1.00** (should satisfy min ≤ mean ≤ max).
    - **Checks:** range_consistency
    - **Verdict:** PASS
    - **Notes:** Ordering condition satisfied.
11. ⚠ **C11_cosmology_ranges_match** (Page 2 (Intro/overview) and Page 6 (Sec. 3.1))
    - **Claim:** Cosmological parameter ranges are stated consistently: Ω_m **0.1–0.5** and σ_8 **0.6–1.0**.
    - **Checks:** repeated_constants_match
    - **Verdict:** UNCERTAIN
    - **Notes:** Only one instance of each range value was available to the checker; repeated match across two locations could not be verified.

12. ✖ **C12_improvement_R2_Om_vs_RF** (Page 6 (Table 6 vs Table 7))
    - **Claim:** Sparse TT improves Ω_m R^2 from **0.85** (RF) to **0.95** (TT).
    - **Checks:** difference_recompute
    - **Verdict:** FAIL
    - **Notes:** Execution output shows computed delta **0.10** but reported expected_delta recorded as **2.0**.
13. ✖ **C13_improvement_R2_Om_vs_GB** (Page 6 (Table 6 vs Table 7))
    - **Claim:** Sparse TT improves Ω_m R^2 from **0.88** (GB) to **0.95** (TT).
    - **Checks:** difference_recompute
    - **Verdict:** FAIL
    - **Notes:** Execution output shows computed delta **0.07** but reported expected_delta recorded as **2.0**.
14. ✖ **C14_improvement_R2_sigma8_vs_RF** (Page 6 (Table 6 vs Table 7))
    - **Claim:** Sparse TT improves σ_8 R^2 from **0.88** (RF) to **0.97** (TT).
    - **Checks:** difference_recompute
    - **Verdict:** FAIL
    - **Notes:** Execution output shows computed delta **0.09** but reported expected_delta recorded as **2.0**.
15. ✖ **C15_improvement_R2_sigma8_vs_GB** (Page 6 (Table 6 vs Table 7))
    - **Claim:** Sparse TT improves σ_8 R^2 from **0.90** (GB) to **0.97** (TT).
    - **Checks:** difference_recompute
    - **Verdict:** FAIL
    - **Notes:** Execution output shows computed delta **0.07** but reported expected_delta recorded as **2.0**.
16. ✖ **C16_ablation_Om_MSE_delta** (Page 7 (Sec. 3.5))
    - **Claim:** After ablating **80%** features, Ω_m MSE increases from **0.0005** to approximately **0.0008**.
    - **Checks:** difference_recompute
    - **Verdict:** FAIL
    - **Notes:** Execution output computed -0.0003 with op recorded as MSE_before - MSE_after, conflicting with the stated 'increases' direction.
17. ✖ **C17_ablation_Om_R2_drop** (Page 7 (Sec. 3.5) and Page 8 (Conclusions))
    - **Claim:** After ablation, Ω_m R^2 drops from **0.95** to **0.91**.
    - **Checks:** difference_recompute
    - **Verdict:** FAIL

- **Notes:** Execution output shows computed drop **0.04** but reported expected_delta recorded as **2.0**.

18. ✖ **C18_ablation_sigma8_MSE_delta** (Page 7 (Sec. 3.5))
    - **Claim:** After ablation, σ_8 MSE increases from **0.0003** to approximately 0.0005.
    - **Checks:** difference_recompute
    - **Verdict:** FAIL
    - **Notes:** Execution output computed -0.0002 with op recorded as MSE_before - MSE_after, conflicting with the stated 'increases' direction.
19. ✖ **C19_ablation_sigma8_R2_drop** (Page 7 (Sec. 3.5))
    - **Claim:** After ablation, σ_8 R^2 decreases from **0.97** to **0.94**.
    - **Checks:** difference_recompute
    - **Verdict:** FAIL
    - **Notes:** Execution output shows computed drop **0.03** but reported expected_delta recorded as **2.0**.

Limitations

- Audit is based only on the provided parsed PDF text; no access to underlying data, code, or supplementary materials to recompute statistics beyond simple arithmetic checks.
- Several method-table entries contain placeholders like '[Value]' which prevents numerical consistency checks in those parts.
- No plot-digitization or image-based numeric extraction was used; only numbers explicitly present in the text/tables were considered.