Skepthical review: *Hierarchical Contrastive Graph Representation Learning for Cosmological Merger Trees and Parameter Inference*

Summary

The manuscript proposes a GraphSAGE-based graph representation learning pipeline for dark-matter halo merger trees, with a hierarchical contrastive objective combining node-level and graph-level InfoNCE losses and an adaptive hard-negative sampling strategy (Sec. 2.4–2.6). Merger trees are grouped by cosmological parameters (Ω_m, σ_8), and positives/negatives are defined primarily by identical vs different cosmology labels; learned 64-d graph embeddings are then frozen and used for downstream regression of (Ω_m, σ_8) via an MLP (Sec. 2.7), plus gradient/perturbation feature-importance analyses and PCA/t-SNE embedding visualizations (Sec. 2.8, 3.4–3.5). On a simulation-level split (train/val/test at the simulation granularity), the paper reports excellent performance for Ω_m ($R^2 \approx 0.98$) and good performance for σ_8 ($R^2 \approx 0.79$) (Sec. 3.3), along with qualitative evidence that graph embeddings vary smoothly with cosmology (Sec. 3.5). However, multiple methodological and reporting issues substantially weaken confidence in the results as currently presented: (i) the contrastive encoder is trained for only two epochs and explicitly not converged (Sec. 2.6, 3.2, 3.6–3.7); (ii) the stated InfoNCE objective is internally inconsistent with negative "validation loss" values and contains apparent formula/notation errors (Sec. 2.5.3, 3.2); (iii) the node-level contrastive component appears empirically ineffective under the current pairing strategy (Sec. 2.5.1, 3.2); and (iv) there are insufficient baselines/ablations to rule out simpler explanations (e.g., feature-only summary statistics, random/frozen encoders, or supervised end-to-end models) and to quantify which components drive the reported performance (Sec. 3.3, 3.6). Strengthening objective correctness, convergence, documentation, and benchmarking would significantly improve rigor, interpretability, and impact.

Strengths

- Timely and well-motivated direction: learning compact, cosmology-sensitive representations from complex merger-tree structure using GNNs and contrastive learning (Sec. 1).
- Meaningful simulation-level splitting strategy that reduces obvious leakage across train/validation/test by keeping simulations (cosmologies) separated (Sec. 2.3, 3.1).
- Promising downstream results on held-out simulations using a simple regressor, suggesting the learned graph representations encode cosmological signal (Sec. 3.3).
- Useful interpretability attempts (gradient-based and perturbation-based feature importance) with qualitatively plausible physical patterns (Sec. 2.8, 3.4).
- Embedding visualizations (PCA/t-SNE) provide intuitive qualitative support for cosmology-related organization of graph representations (Sec. 3.5).

- The paper acknowledges several limitations (short training, weak node-level behavior) and outlines reasonable next steps, which provides a good basis for revision (Sec. 3.6–3.7, 4).

Major issues

1. **The contrastive training objective is internally inconsistent and likely mis-specified and/or mis-logged. Sec. 2.5.3 (Eq. (1)) contains a denominator term written as $\mathrm{sim}(z_k)$ that omits the anchor–negative similarity $\mathrm{sim}(z_i, z_{n,k})$, and Sec. 3.2 reports negative validation "InfoNCE losses" (e.g., -3.8) even though the defined loss $L = -\log\left(\frac{\exp(\mathrm{pos})}{\exp(\mathrm{pos})+\Sigma\exp(\mathrm{neg})}\right)$ is non-negative by construction.** This is a critical correctness/reproducibility problem that undermines confidence in the learned embeddings and all downstream claims (Sec. 2.5.3–2.6, 3.2).

   Recommendation: Audit the implementation and reporting of the contrastive objective end-to-end. In Sec. 2.5.3–2.5.4, rewrite Eq. (1) with correct anchor–positive and anchor–negative similarities and explicitly state: (i) whether the positive is included in the denominator; (ii) whether a symmetric loss is used; (iii) reduction/averaging conventions over anchors and (if applicable) multiple positives. In Sec. 3.2, clearly define what scalar is plotted/reported as "validation loss" (loss vs log-likelihood vs negative loss). After correction, rerun training and provide train/validation curves that behave consistently with the stated definition. Update all quantitative results and figures that depend on these embeddings (Sec. 3.2–3.5) and revise the Discussion (Sec. 3.6–3.7) accordingly.

2. **The encoder is trained for only two epochs and explicitly not converged (Sec. 2.6, 3.2, 3.6–3.7), yet all main results (R^2, similarity distributions, feature importance, PCA/t-SNE) are drawn from this undertrained state.** This makes outcomes potentially unstable with respect to initialization, batching, and early-training dynamics, and makes it difficult to interpret the reported strong regression performance as a property of the intended method rather than an artifact.

   Recommendation: Train the contrastive encoder to a clear convergence criterion (e.g., stabilization of validation contrastive loss and/or a proxy validation metric such as linear/MLP probe performance on validation embeddings). Report full learning curves (train/val) over substantially more epochs and across multiple random seeds. Recompute and report all downstream metrics (Sec. 3.3), similarity-distribution diagnostics (Sec. 3.2), feature-importance analyses (Sec. 3.4), and embedding visualizations (Sec. 3.5) using the converged encoder; optionally include 2-epoch results only as an explicit ablation/reference point.

3. **Insufficient baselines to demonstrate that hierarchical contrastive learning (and topology-aware message passing) is responsible for the performance.** Given the surprisingly strong regression after minimal training, it is plausible that

simple correlates (e.g., aggregated node-feature statistics) or even random/frozen encoders could produce comparable results, or that topology is not contributing substantially (Sec. 3.3, 3.6).

Recommendation: Add a compact but decisive baseline suite evaluated on the same simulation-level splits (Sec. 2.3.2, 3.1): (a) random-initialized GraphSAGE encoder (frozen) + same pooling + same regressor; (b) feature-only baselines using hand-crafted summaries (mean/variance/max of node features; tree size; depth; mass-function-like summaries) with MLP/RF; (c) a supervised end-to-end GNN regressor trained directly on (Ω_m, σ_8) without contrastive pretraining; and (d) a topology ablation such as shuffled edges (or an MLP applied per node with pooling, no message passing) to quantify how much graph structure matters. Report test R^2/MAE (Sec. 3.3), plus computational cost. Use these results to substantiate (or appropriately temper) the claimed benefits of the proposed approach (Sec. 1, 3.6, 4).

4. **The current positive/negative construction is label-based (identical cosmology parameters define positives) and thus is closer to supervised contrastive learning than self-supervised learning (Sec. 2.5.1). In addition, because positives are defined by exact equality of (Ω_m, σ_8), the method may primarily learn to cluster by simulation/cosmology class rather than learn a representation that supports smooth interpolation in continuous parameter space—yet the downstream task is continuous regression (Sec. 2.7, 3.3, 3.5).**

   Recommendation: Clarify terminology and intent: explicitly describe the approach as label-/group-supervised contrastive learning (or justify "self-supervised" usage). Then evaluate continuous generalization more directly: (i) structured splits in Ω_m–σ_8 space (e.g., leave-out a region/corner; train on low Ω_m and test on high Ω_m) rather than only a random holdout of 5 simulations (Sec. 3.1, 3.3); (ii) error/residual analysis as a function of parameter-space location (Sec. 3.3–3.6); (iii) optionally, embedding-space smoothness diagnostics such as correlation between embedding distances and parameter distances or kNN regression in embedding space. State clearly whether extrapolation beyond the sampled parameter range is supported (Sec. 3.6–3.7).

5. **The hierarchical/node-level contrastive component appears ineffective and under-justified. Node-level positives are formed by randomly sampling nodes from different trees with the same cosmology (Sec. 2.5.1), but Sec. 3.2 shows strong overlap between node-level positive and negative similarity distributions even after training.** This raises the possibility that the node-level loss contributes little or adds noise, and the paper currently does not quantify its value relative to a graph-only objective.

   Recommendation: Quantify the contribution of node-level contrastive learning via ablations: graph-only ($\alpha = 0$), node-only ($\alpha = 1$), and several intermediate α values in the combined loss (Sec. 2.5.4), all trained to convergence. For each, report: node- and

graph-level similarity separation metrics (not only histograms), downstream regression performance (Sec. 3.3), and qualitative embedding plots (Sec. 3.5). If node-level contrastive is retained, revise the node-positive definition to enforce semantic correspondence (e.g., matching by scale-factor bin, mass percentile, depth-from-root, main-branch nodes) and document the exact sampling algorithm (Sec. 2.5.1). If it does not help, simplify the method and reframe the contribution accordingly (Sec. 3.6, 4).

6. **Adaptive hard-negative sampling is introduced as a key design (Sec. 2.5.2) but is not convincingly justified or validated; the similarity window $[0.2, 0.6]$ and K_{neg} appear ad hoc, and it is unclear how many eligible negatives exist during training and whether the scheme is stable as embeddings evolve.**

   Recommendation: Provide a focused ablation where only the negative sampling strategy changes: (i) adaptive windowed hard negatives (current method), (ii) random in-batch negatives, (iii) all valid in-batch negatives, and (optionally) a true "hardest negatives" variant. Report contrastive learning curves, fraction of eligible negatives over training, and downstream regression metrics (Sec. 3.2–3.3). Explain how the window and K_{neg} were chosen (validation sweep or heuristic), and include these settings in a consolidated hyperparameter/config table (Sec. 2.9 or Appendix).

7. **Dataset provenance and simulation/merger-tree construction details are insufficient to assess physical representativeness and external validity.** Sec. 2.1 and 3.1 cite Parkinson et al. (2007) and Jiang & van den Bosch (2013), but do not clearly state which simulation suite produced the trees, how halos were identified, snapshot spacing, mass resolution, box size, whether graphs are directed/undirected in the GNN, and how the 25 trees per cosmology were selected (random halos? fixed mass bin?). This makes it hard to interpret what regimes the reported results apply to and whether there are hidden confounds.

   Recommendation: Expand Sec. 2.1, 2.2.2, and 3.1 with a concrete, reproducible dataset description: simulation suite(s), box size, particle mass, halo finder, tree builder, snapshot/redshift grid, selection criteria for the 25 trees per cosmology, and cosmology sampling scheme/ranges for Ω_m and σ_8 (grid vs Latin hypercube, etc.). Include a plot of Ω_m–σ_8 coverage and a short statement about how this range relates to observational constraints (and that extrapolation is not validated). Also specify whether edges are treated as directed or converted to undirected for GraphSAGE (Sec. 2.4).

8. **Regression evaluation is currently limited to aggregate test metrics and a small number of held-out cosmologies (5 simulations).** For cosmological inference, it is important to quantify uncertainty, variability across splits/seeds, and performance heterogeneity across parameter space (Sec. 3.3, 3.6).

Recommendation: In Sec. 3.3, add: (i) per-test-cosmology metrics (one point per held-out simulation) and residual/bias plots; (ii) binned error analysis across Ω_m and σ_8 (1D bins or a 2D grid); (iii) uncertainty estimates via multiple random seeds and preferably multiple simulation-level splits (cross-validation across the 40 cosmologies). If some systematic biases appear (e.g., at high σ_8), quantify them and incorporate into the limitations (Sec. 3.6–3.7).

9. **Presentation/consistency issues significantly impede verification and actionability: several figures/tables use placeholders ("illustrative values", "Figures ??"), figure numbering/cross-references appear broken, and some plots lack normalization/sample sizes and quantitative separation measures (Sec. 2.2.1–2.2.3, 3.2–3.7).**

   Recommendation: Remove or relocate "illustrative" tables (Tables 1–3 in Sec. 2.2.1–2.2.3) unless replaced with real dataset statistics. Fix all figure labels/references using consistent \label/\ref (Sec. 3.2–3.7). For similarity-distribution plots (Sec. 3.2), normalize histograms (probability density), report N for each class, specify binning, and add quantitative separation metrics (e.g., AUROC, KS distance, overlap coefficient). Ensure colorblind-safe palettes and readable typography throughout.

Minor issues

1. Edge features are mentioned but then not used ("edge_attr ... were not explicitly utilized"), despite merger trees naturally encoding temporal/relational information along edges (Sec. 2.1, 2.4). It remains unclear whether discarding edge information is harmless or leaves performance on the table.

   Recommendation: Either justify explicitly why edge_attr is redundant in this dataset (e.g., scale factor already in node features), or add a small ablation that incorporates edge features (edge-conditioned messages, time-difference encoding) and report whether it changes contrastive separation and downstream regression (Sec. 3.6).

2. Batch construction and multi-positive handling are not specified clearly. Sec. 2.5.1 implies potentially multiple positives per anchor ("any other graph with identical cosmological parameters"), but Sec. 2.5.3 writes a single-positive InfoNCE form. Contrastive learning behavior depends strongly on how many cosmologies per batch are included and how positives are sampled/aggregated.

   Recommendation: In Sec. 2.5.1–2.5.4, provide an explicit algorithm: number of anchors per batch, number of cosmologies per batch and whether balanced, number of positives per anchor and how selected (sample one vs average over all), and number of negatives considered (before/after filtering).

3. The relationship between backbone embeddings, projection-head outputs used for contrastive learning, and embeddings used for regression/feature-importance is inconsistently described (Sec. 2.4, 2.5, 2.7, 2.8, 3.4). This ambiguity affects interpretability of

both performance and gradients.

Recommendation: Standardize notation (e.g., h_G for pooled pre-projection, z_G for projected embedding). State explicitly in Sec. 2.7 and Sec. 3.3–3.5 which representation is frozen and fed into the regressor/visualizations, and in Sec. 2.8/3.4 which representation the gradients are taken through.

4. Feature-importance definitions and normalization are under-specified and potentially confounded by graph size and feature scaling. Sec. 2.8 defines a sum of absolute gradients across nodes and embedding dimensions, while Sec. 3.4 describes an "average magnitude," and perturbation magnitudes are not clearly specified.

   Recommendation: Decide and document whether importance is a sum or an average; if comparing across graphs, normalize by $\mathbf{num_nodes}(G)$ (and possibly embedding dim). Specify whether gradients/perturbations are computed in standardized feature space or physical units, and give perturbation magnitude (e.g., $+\mathbf{1}$ std) and whether applied to all nodes or subsets. Add uncertainty/error bars across graphs (Sec. 3.4).

5. Pooling choice (global mean pooling) may wash out important temporal/hierarchical structure, especially with variable-sized trees, and the paper does not assess sensitivity to pooling (Sec. 2.4, 2.7).

   Recommendation: Add a lightweight ablation comparing mean vs sum vs attention pooling (or Set2Set) and report the impact on regression and/or contrastive separation (Sec. 3.3, 3.6).

6. Reproducibility/configuration details are scattered and incomplete (architecture, τ, α, K_{neg}, similarity window, optimizer settings, seeds, early stopping, etc.) (Sec. 2.4–2.7, 2.9, 3.2–3.3).

   Recommendation: Provide a consolidated configuration table (Sec. 2.9 or Appendix) listing final hyperparameters for the GNN, projection head, sampling, and regressor; include compute environment and approximate training time. Report mean±std over multiple seeds at least for the main metrics (Sec. 3.3).

7. Terminology and positioning: the method is described as self-supervised, but positives are constructed using cosmology labels (Sec. 2.5.1, Abstract/Sec. 1).

   Recommendation: Rephrase accurately as label-/group-supervised contrastive learning (or carefully justify the self-supervised framing). If feasible, add a truly self-supervised variant based on within-graph augmentations (subtree sampling, edge dropout, feature masking) as a comparison (Sec. 3.6).

8. Figures in Sec. 3.2–3.5 often omit key plot metadata (bin widths, normalization, sample sizes), and some palettes/typography reduce accessibility (color-vision deficiency, grayscale printing).

Recommendation: Standardize figure styling: probability-normalized histograms or KDE overlays, include N and binning in captions, use colorblind-safe palettes, and increase font sizes/DPI (or export vector graphics). Annotate key quantitative metrics directly on plots where relevant (R^2, MAE, AUROC/KS, etc.).

9. Downstream regressor details are too brief given that regression performance is the main quantitative proxy of embedding quality (Sec. 2.7, 3.3).

   Recommendation: In Sec. 2.7, specify MLP depth/width, activations, regularization (dropout/weight decay), optimizer, learning rate, epochs, early stopping, and whether targets are normalized/denormalized during training/evaluation. In Sec. 3.3, report whether results are from one run or averaged over seeds.

Very minor issues

1. Numerous LaTeX/formatting and cross-referencing problems remain (e.g., "Figures ??", broken labels like "Figures 4–LABEL:...", placeholder citations such as "Li et al. 2025,?,?", unintended line breaks, inconsistent notation for $\mathbf{log_{10}(mass)}$ and V_{max}) across Sec. 1–4 and the figure captions.

   Recommendation: Perform a full proofreading pass and resolve all placeholders, citations, and cross-references. Standardize notation and units throughout (Sec. 2.1, 3.1, 3.4), and ensure every figure/table referenced in Sec. 3.2–3.7 exists with correct numbering and captions.

2. Section heading markup/numbering is inconsistent (mixed styles such as "# 2.2 ...", "### 2.2 ...", and inconsistent subsection numbering), which makes navigation and references harder.

   Recommendation: Unify heading/numbering style (LaTeX \section/\subsection or a consistent Markdown scheme) so that in-text references to Sec. 2.2.1, 3.1–3.4 match the compiled document.

3. Minor figure/table presentation issues (small fonts, inconsistent decimal precision, legend redundancy, non-colorblind-safe red/green contrasts, axis limits that may truncate distributions) appear across multiple figures (Sec. 3.2–3.5).

   Recommendation: Increase typography and line weights, standardize decimal precision, simplify legends/titles, adopt accessible palettes, and justify or remove axis truncations; ensure readability in grayscale/print.

4. Embedding notation is overloaded (graph embedding sometimes refers to pooled backbone output and sometimes to projection-head output), and anchor notation varies (z_a vs z_i) (Sec. 2.5.2–2.5.3, 2.8).

   Recommendation: Introduce distinct symbols for pre-projection vs projected embeddings and use consistent anchor notation across Sec. 2.4–2.8 and Results.

Key statements and references

- • **Constructing and analyzing dark matter halo merger trees that accurately reflect cosmological parameter dependencies is challenging due to their complex, high-dimensional, and hierarchical nature, as shown by recent work on merger tree construction and analysis in cosmological simulations.**
- *Reference(s):* Robles et al., 2022, Bose et al., 2022, Nguyen et al., 2024
- • **Graph Neural Networks, including GraphSAGE-style architectures, have been demonstrated to effectively process graph-structured astrophysical data and capture relationships between halos and their environments, motivating their use for merger tree representation learning in this study.**
- *Reference(s):* Tang and Ting, 2022, Wu et al., 2024, Jespersen et al., 2022
- • **Hierarchical contrastive learning frameworks that combine node-level and graph-level contrastive objectives have been proposed in the machine learning literature to unify multiple granularities of representation, and this work adapts that idea to cosmological merger trees to learn embeddings sensitive to underlying cosmology.**
- *Reference(s):* Li et al., 2025, Li et al., 2025b, Li et al., 2025a
- • **Contrastive learning in astrophysics has recently been reviewed and applied to tasks such as representation learning for galaxy images and radio astronomy data, providing evidence that contrastive objectives and hard-negative mining can yield useful, discriminative embeddings for downstream scientific inference.**
- *Reference(s):* Huertas-Company et al., 2023, Slijepcevic et al., 2022, Desmons et al., 2024
- • **The 1000 merger trees used in this work are generated from dark matter halo merger tree algorithms developed for N-body simulations, which have been previously validated as accurate methods for constructing halo assembly histories across cosmological parameter variations.**
- *Reference(s):* Parkinson et al., 2007, Jiang and van den Bosch, 2013, Ángel Chandro-Gómez et al., 2025
- • **The choice to split the dataset at the simulation (cosmology) level rather than at the individual tree level follows best practices from recent cosmological machine learning studies, which emphasize avoiding information leakage when constraining cosmological parameters from simulations.**
- *Reference(s):* Lazanu 2021, Hortua 2021, Burger et al., 2024

- • **Using learned low-dimensional summaries (deep or simulation-based summaries) of cosmological data for parameter estimation has been successfully demonstrated in prior work, supporting the use of graph embeddings here as inputs to a downstream regressor for inferring Ω_m and σ_8.**
- *Reference(s):* Fluri et al., 2021, de Santi et al., 2025, Chatterjee and Villaescusa-Navarro, 2025

Mathematical consistency audit

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

Maths relevance: light

The paper's analytic content centers on (i) defining hierarchical contrastive objectives via a standard InfoNCE loss at node and graph levels, combined linearly, and (ii) defining gradient-based feature-importance scores as sums/averages of partial derivatives of an embedding with respect to input features. There are few explicit derivations, but there is a key internal inconsistency between the stated InfoNCE definition (non-negative) and the reported negative validation losses.

Checked items

1. ✔ **InfoNCE definition (single-positive, temperature-scaled)** (Sec. 2.5.3, p.5)
    - **Claim:** Defines InfoNCE as $-\log\left(\frac{\exp(\mathrm{sim}(z_i,z_p)/\tau)}{\exp(\mathrm{sim}(z_i,z_p)/\tau)+\sum_k \exp(\mathrm{sim}(z_i,z_{n,k})/\tau)}\right)$.
    - **Checks:** algebra, constraint/sanity (loss sign)
    - **Verdict:** PASS; confidence: high; impact: critical
    - **Assumptions/inputs:** $\tau > 0$, Denominator includes the positive term plus at least zero negative terms, $\mathbf{sim}(\cdot,\cdot)$ is finite (cosine similarity)
    - **Notes:** As written, the fraction is in $(0,1]$, so the log is ≤ 0 and the leading minus makes the loss ≥ 0. This property is later contradicted by reported negative validation losses (checked separately).
2. ✖ **Non-negativity implication vs reported negative validation loss** (Sec. 3.2, p.7 (validation loss values) vs Sec. 2.5.3, p.5 (InfoNCE formula))
    - **Claim:** Validation loss is reported as negative while using InfoNCE-based objectives.
    - **Checks:** definition consistency, constraint/sanity (loss sign)
    - **Verdict:** FAIL; confidence: high; impact: critical
    - **Assumptions/inputs:** Validation loss is intended to be the same L_{total} (or an average of InfoNCE terms) defined in Methods, No additional constant offsets or sign flips are applied

- **Notes:** Given the stated InfoNCE form, neither InfoNCE nor a convex combination of two InfoNCE losses can be negative. A negative validation loss indicates the reported metric is not the defined loss (e.g., sign error, different objective, or mis-reporting). This undermines interpretability of training/early-stopping selection as described.

3. ✔ **Combined hierarchical loss as convex combination** (Sec. 2.5.4, p.5)
   - **Claim:** Total loss is $L_{\text{total}} = \alpha L_{\text{node}} + (1 - \alpha) L_{\text{graph}}$ with $\alpha \in [0, 1]$.
   - **Checks:** algebra, constraint consistency
   - **Verdict:** PASS; confidence: high; impact: moderate
   - **Assumptions/inputs:** L_{node} and L_{graph} are scalar losses computed on the batch, α is a scalar hyperparameter in $[0, 1]$
   - **Notes:** Formula is algebraically consistent and preserves non-negativity if L_{node} and L_{graph} are non-negative InfoNCE losses.

4. ✔ **Adaptive negative sampling based on cosine similarity window** (Sec. 2.5.2, p.5)
   - **Claim:** Selects negatives with cosine similarity to anchor in a prescribed interval, then tops up with random negatives to reach K_{neg}.
   - **Checks:** definition consistency, constraint/sanity
   - **Verdict:** PASS; confidence: medium; impact: minor
   - **Assumptions/inputs:** Cosine similarity $\mathbf{sim}(\cdot, \cdot) \in [-1, 1]$, Specified window (e.g., 0.2 to 0.6) is within $[-1, 1]$, K_{neg} is a positive integer
   - **Notes:** Sampling rule is well-defined at a high level. Exact distribution induced is unspecified but not an internal algebraic issue.

5. ⚠ **Multiple positives described vs single-positive InfoNCE formula** (Sec. 2.5.1 (pair definition), p.4–5 and Sec. 2.5.3 (loss), p.5)
   - **Claim:** Positives are defined as any other graph/node from the same cosmology, while the loss is written for one positive z_p per anchor.
   - **Checks:** definition consistency, missing-derivation/omitted-step check
   - **Verdict:** UNCERTAIN; confidence: medium; impact: moderate
   - **Assumptions/inputs:** Batches may contain multiple graphs from the same cosmology, Implementation must choose how to incorporate multiple positives
   - **Notes:** To be fully consistent, the paper must specify whether it samples a single positive per anchor, averages across all positives, or uses a multi-positive InfoNCE variant. Without this, L_{node} and L_{graph} are not uniquely defined from the text.

6. ✔ **Feature-importance sensitivity score $S_{G,j}$ (gradient sum)** (Sec. 2.8, p.6)
   - **Claim:** Defines $S_{G,j} = \sum_{k=1}^{\dim(Z_G)} \sum_{i=1}^{\text{num}_\text{n}\text{odes(G)}} \left| \frac{\partial Z_{G,k}}{\partial x_{i,j}} \right|$.

- **Checks:** definition consistency, dimensional/sanity (indexing)
- **Verdict:** PASS; confidence: medium; impact: minor
- **Assumptions/inputs:** Z_G is differentiable with respect to inputs $\boldsymbol{x}$, $\boldsymbol{x}_{i,j}$ denotes feature $\boldsymbol{j}$ at node i, $\mathbf{dim}(Z_G)$ and $\mathbf{num_nodes}(G)$ are finite
- **Notes:** Mathematically well-defined (up to typographic clarity of summation limits). The resulting magnitude depends on graph size and embedding dimension unless normalized.

7. ✔ **Global feature importance as average over test graphs** (Sec. 2.8, p.6)
    - **Claim:** Defines ${\rm Importance}$j$ = \frac{1}{|\text{Test Dataset}|} \sum$.}} S_{G,j
    - **Checks:** algebra, normalization
    - **Verdict:** PASS; confidence: high; impact: minor
    - **Assumptions/inputs:** $|\mathbf{Test\ Dataset}| > 0$
    - **Notes:** Averages per-graph sensitivity scores. Interpretability depends on whether $S_{G,j}$ is normalized for graph size (not addressed).

8. ✖ **Sum vs average wording for gradient-based importance** (Sec. 2.8, p.6 (sum definition) vs Sec. 3.4, p.10–11 (described as average magnitude))
    - **Claim:** Results are described as averages while the method defines sums.
    - **Checks:** definition consistency
    - **Verdict:** FAIL; confidence: medium; impact: moderate
    - **Assumptions/inputs:** The reported numbers/plots in Results correspond to the metric defined in Methods
    - **Notes:** Sec. 2.8 defines a sum across nodes and embedding dimensions; Sec. 3.4 repeatedly calls the metric an average magnitude. These differ by scaling factors and can change comparisons across graphs with different $\mathbf{num_nodes}$.

9. ⚠ **Embedding symbol overloading (raw vs projected embeddings)** (Sec. 2.4 (graph embeddings via mean pooling), p.4; Sec. 2.4 (projection heads), p.4; Sec. 2.8 (raw embedding before projection head), p.6; Sec. 2.7 (embeddings used are projection-head output), p.6)
    - **Claim:** Z_G/graph embedding notation refers to multiple stages (pooled representation, projection-head output).
    - **Checks:** notation/definition consistency
    - **Verdict:** UNCERTAIN; confidence: medium; impact: minor
    - **Assumptions/inputs:** There are at least two distinct vectors: pooled pre-projection and projected 64-D embedding
    - **Notes:** The pipeline implies multiple representations, but notation does not cleanly distinguish them. This is mainly a clarity/consistency issue; it becomes material in Sec. 2.8 where gradients depend on which embedding is

used.

Limitations

- The provided PDF text contains no equation numbering; locations are given by section and page only.
- Several computations referenced in Results (e.g., Euclidean shift under perturbations, regression-output sensitivity) are described qualitatively without explicit mathematical definitions, limiting symbolic verification to the formulas that are actually written.
- This audit is analytic only and does not validate numerical values, plots, or implementation behavior beyond checking whether reported signs/properties contradict stated mathematical definitions.

Numerical results audit

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

Thirteen internal-consistency checks were executed across dataset composition/splitting statements, Table 4 range sanity checks, and R^2-to-percentage interpretation statements; all checks passed within the stated tolerances, with only small rounding/phrasing differences for the R^2-to-percent items.

Checked items

1. ✔ **C1_dataset_grouping_total** (Methods §2.1 (page 3) and Results §3.1 (page 7))
    - **Claim:** The 1000 trees originate from 40 distinct N-body simulations, with 25 trees sampled from each simulation.
    - **Checks:** parts_vs_total
    - **Verdict:** PASS
    - **Notes:** Exact integer equality expected.
2. ✔ **C2_split_counts_by_simulation** (Methods §2.3.2 (page 4) and Results §3.1 (page 7))
    - **Claim:** Simulation-level split: Training 30 simulations, Validation 5 simulations, Test 5 simulations (total 40 simulations).
    - **Checks:** parts_vs_total
    - **Verdict:** PASS
    - **Notes:** Exact integer equality expected.
3. ✔ **C3_split_counts_by_trees** (Methods §2.3.2 (page 4) and Results §3.1 (page 7))
    - **Claim:** Split in trees: Training 750 trees, Validation 125 trees, Test 125 trees (total 1000 trees).

- **Checks:** parts_vs_total
- **Verdict:** PASS
- **Notes:** Exact integer equality expected.

4. ✔ **C4_split_tree_counts_from_sims_times_25** (Methods §2.1 (page 3) + Methods §2.3.2 (page 4))
    - **Claim:** Because there are 25 trees per simulation, the stated split in simulations implies 750/125/125 trees.
    - **Checks:** derived_quantity_recompute
    - **Verdict:** PASS
    - **Notes:** Exact integer equality expected for each split.
5. ✔ **C5_table4_mean_within_minmax_mass** (Results §3.1 Table 4 (page 7))
    - **Claim:** Table 4 gives mean/std/min/max for $\log_{10}$**(mass)**. Mean should lie within [min,max].
    - **Checks:** range_check
    - **Verdict:** PASS
    - **Notes:** Strict inequality check; floating values as stated.
6. ✔ **C6_table4_mean_within_minmax_conc** (Results §3.1 Table 4 (page 7))
    - **Claim:** Table 4 gives mean/std/min/max for $\log_{10}$**(concentration)**. Mean should lie within [min,max].
    - **Checks:** range_check
    - **Verdict:** PASS
    - **Notes:** Strict inequality check.
7. ✔ **C7_table4_mean_within_minmax_vmax** (Results §3.1 Table 4 (page 7))
    - **Claim:** Table 4 gives mean/std/min/max for $\log_{10}(V_{\max})$. Mean should lie within [min,max].
    - **Checks:** range_check
    - **Verdict:** PASS
    - **Notes:** Strict inequality check.
8. ✔ **C8_table4_mean_within_minmax_scale** (Results §3.1 Table 4 (page 7))
    - **Claim:** Table 4 gives mean/std/min/max for Scale Factor. Mean should lie within [min,max].
    - **Checks:** range_check
    - **Verdict:** PASS
    - **Notes:** Strict inequality check.
9. ✔ **C9_training_label_stats_mean_within_minmax_omegam** (Results §3.1 (page 7, paragraph after Table 4))

- **Claim:** Training-set Ω_m: mean 0.2678, min 0.1030, max 0.4734; mean should lie within [min,max].
- **Checks:** range_check
- **Verdict:** PASS
- **Notes:** Strict inequality check.

10. ✔ **C10_training_label_stats_mean_within_minmax_sigma8** (Results §3.1 (page 7, same paragraph))
    - **Claim:** Training-set σ_8: mean 0.8173, min 0.6030, max 0.9786; mean should lie within [min,max].
    - **Checks:** range_check
    - **Verdict:** PASS
    - **Notes:** Strict inequality check.
11. ✔ **C11_nodes_summary_mean_within_range** (Results §3.1 (page 7, same paragraph) and Methods Table 3 (page 3, illustrative))
    - **Claim:** Merger tree size described as mean 250 nodes, range min 50 to max 500; mean should lie within [min,max].
    - **Checks:** range_check
    - **Verdict:** PASS
    - **Notes:** Exact integers.
12. ✔ **C12_regression_table5_r2_vs_percent_variance_omegam** (Results §3.3 (page 9, Table 5 and paragraph below))
    - **Claim:** They report $R^2 = 0.977852$ for Ω_m and state this indicates nearly 98% of the variance explained.
    - **Checks:** percentage_from_fraction
    - **Verdict:** PASS
    - **Notes:** Allow rounding/wording; check $100 \times R^2 = 97.7852\%$ is consistent with 'nearly 98%'.
13. ✔ **C13_regression_table5_r2_vs_percent_variance_sigma8** (Results §3.3 (page 9, paragraph describing σ_8))
    - **Claim:** They report $R^2 = 0.79462$ for σ_8 and state this signifies about 80% of the variance captured.
    - **Checks:** percentage_from_fraction
    - **Verdict:** PASS
    - **Notes:** About/rounding; $100 \times R^2 = 79.462\%$ should map to 'about 80%'.

Limitations

- Only the provided PDF text was used; no external datasets, code repositories, or supplementary files were accessed.

- Checks that require reconstructing metrics from underlying predictions/targets (e.g., deriving R^2 from MSE without target variance) are not feasible with the information given.
- Image/plot value extraction is excluded; numerical verification relies on explicitly stated numbers in the text/tables.
- Cannot verify the negative validation-loss values without the exact loss implementation details and underlying logits/similarities.
- Cannot verify mutual consistency among MSE/MAE/R^2/Pearson correlation values without the variance of targets and exact metric definitions/averaging.
- Cannot verify the approximately stated final validation MSE ($\sim \mathbf{0.0022}$) without the underlying training history/logs.
- Cannot cross-validate feature-importance values shown in figures because figure value extraction is disallowed and no alternative tabulated source is provided.