Skepthical review: *Predicting Halo Assembly Bias from Merger Trees using Graph Neural Networks with Formation Time Regularization*

Summary

The manuscript applies a GCN-based Graph Neural Network to dark-matter halo merger trees, encoded as graphs with node features (e.g., mass, concentration, Vmax, scale factor) and edge features (accretion-rate statistics) (Sec. 2.1–2.2). The target is a graph-level "formation-time" proxy computed from nodes on the main progenitor branch (described as the median scale factor, but inconsistently defined across sections) (Sec. 2.1.2, Sec. 3.1). The model is trained with a composite loss: an MSE term on the formation-time proxy plus node- and edge-level Pearson-correlation regularizers intended to align node embeddings with scale factor and edge embeddings with accretion rates, respectively (Sec. 2.3.2–2.3.4). On 1000 trees with an 80/10/10 split, the model achieves moderate performance (test $R^2 \approx 0.48$) (Sec. 3.3). Training diagnostics show the node-regularization rapidly drives embedding–scale-factor correlation to ~ 1, while the edge-regularization remains ~ 0 and appears inoperative (Sec. 3.4). The work is a promising proof-of-concept for representation learning on merger trees, but key aspects of scientific framing (assembly bias vs. formation time), target/feature leakage, missing baselines/ablations, unclear edge-feature/regularizer implementation, limited dataset description, and lack of robustness/uncertainty quantification currently limit interpretability, reproducibility, and the strength of the astrophysical claims (Sec. 1, Sec. 2.1.2, Sec. 2.2–2.5, Sec. 3.3–3.5, Sec. 4).

Strengths

- Timely application of GNNs to merger-tree data; trees are inherently graphical/temporal, making the approach well-motivated (Sec. 1–2).
- Clear end-to-end pipeline presentation from data preparation to training and evaluation, with informative diagnostics (Sec. 2–3).
- The node-level temporal regularization is a simple, interesting mechanism and its effect is clearly demonstrated in training curves (Sec. 2.3.2, Sec. 3.4).
- Nontrivial predictive accuracy on the chosen proxy given a modest sample size ($N = 1000$) (Sec. 3.3).
- The manuscript candidly reports the edge-regularization failure and discusses possible reasons, which is useful for future work (Sec. 3.4–3.5).
- Figures are generally well-organized and, where readable, convey key patterns such as outliers and residual structure (Sec. 3.3–3.4).

Major issues

1. **Scientific framing: the paper repeatedly presents the task as "predicting halo assembly bias," but the model actually predicts an internal formation-time proxy. Assembly bias is fundamentally a statement about clustering/large-scale bias dependence at fixed mass, which is not evaluated here (Sec. 1, Sec. 3.5, Sec. 4).**

   Recommendation: Either (a) reframe throughout (Abstract, Sec. 1, Sec. 3.5, Sec. 4, and figure titles/captions) as "predicting a formation-time proxy from merger trees," explicitly treating assembly bias only as motivation; or (b) add a direct assembly-bias validation: at fixed mass, split halos by true and by predicted formation-time proxy and measure a standard assembly-bias observable (e.g., large-scale bias from the two-point correlation function or bias estimator), reporting the strength of the signal and how well predictions reproduce it (Sec. 3.5). Ensure title and conclusions match the final scope.

2. **Target definition is inconsistent and insufficiently motivated. Sec. 2.1.2 describes both the "earliest" main-branch node and the "median scale factor" of main-branch nodes, while elsewhere it is described as the median (Sec. 2.1.2, Sec. 3.1, Sec. 3.3). This undermines reproducibility and comparability to standard formation-time definitions in the literature.**

   Recommendation: Unify the definition in Sec. 2.1.2 with a precise formula/pseudocode (using `mask_main` and scale factor), and ensure consistent terminology in Sec. 3.1, Sec. 3.3, Sec. 3.5, and all figure labels. Add brief motivation and comparison to standard definitions (e.g., $a_{1/2}$ from main-branch mass growth, or formation redshift when a given mass fraction is assembled). If feasible, compute at least one conventional formation-time metric on the same halos and report its correlation with the chosen proxy (Sec. 3.5).

3. **Potential information leakage / tautology: the target is computed from scale factor values in the tree, while scale factor is also an input node feature and the node-regularizer explicitly forces embeddings to be maximally correlated with scale factor (Sec. 2.1.2, Sec. 2.3.2, Sec. 3.4). This raises the possibility that the network is largely learning to read out a statistic of an already-provided feature rather than learning assembly history in a physically meaningful sense.**

   Recommendation: Add targeted baselines and ablations in Sec. 3.3–3.4: (i) non-graph baselines using only summary statistics of node scale factors (and other node features) without message passing; (ii) remove scale factor from node inputs and retrain; (iii) remove the node-regularizer (MSE-only) and compare; (iv) optionally, remove `mask_main`-related information if it is provided to/used by the model at inference

(clarify in Methods). Report MSE/R^2/MAE for each setting. This will demonstrate whether graph structure and non-time features contribute beyond directly encoding/regularizing time.

4. **Missing baselines and loss/architecture ablations make it hard to assess the added value of (a) the GNN versus simpler models and (b) the custom regularizers. With test $R^2 \approx 0.48$, it is unclear how much the approach improves over straightforward regressors on hand-crafted features or pooled node features (Sec. 2.2–2.3, Sec. 3.3–3.5).**

   Recommendation: In Sec. 3.3, include at least: (1) mean-predictor baseline; (2) linear regression and/or random forest on hand-crafted tree summaries (e.g., main-branch mass history summaries, node-feature moments, accretion-rate summaries); (3) an MLP on globally pooled node features; (4) GNN with MSE-only; (5) GNN with MSE+node-reg only; (6) GNN with MSE+edge-reg only (if meaningful). Summarize results in a table and discuss what each comparison implies about the contribution of graph structure and regularization (Sec. 3.5).

5. **Core methodological details are underspecified or internally inconsistent, limiting reproducibility and diagnosis of the edge-regularization failure. Notably: (i) the exact GCN layer/variant is not stated; (ii) how edge attributes enter the convolution is described inconsistently (per-message concatenation vs. pre-aggregation of incoming edges before the first layer); (iii) Pearson-correlation regularizers lack precise definitions (tensor shapes, scalarization, masking, per-graph vs batch computation) (Sec. 2.2, Sec. 2.3.2–2.3.4, Sec. 3.2–3.4).**

   Recommendation: Expand Sec. 2.2 and Sec. 2.3 with implementable specificity: name the library layer (e.g., PyG `GCNConv`) and its options (self-loops, normalization), clarify directed vs undirected handling for merger-tree edges, and provide explicit formulas for the node/edge regularizers: define the scalar node score used (e.g., mean over embedding dims or a learned projection), the index set over which correlation is computed (all nodes vs main-branch nodes), and how correlations are aggregated across graphs/batches (with variance safeguards). Make Sec. 2.2 and Sec. 3.2 fully consistent and include pseudocode in an appendix if needed.

6. **Edge-level regularization appears inoperative (≈ 0 throughout training), raising the possibility of a bug (e.g., detached tensors), a constant/zero-variance statistic, or a definition that yields vanishing gradients. Additionally, it is unclear how "edge embeddings" are obtained, since a vanilla GCN produces node embeddings (Sec. 2.3.3–2.3.4, Sec. 3.4).**

   Recommendation: In Sec. 3.4 (and Methods), precisely define how edge embeddings are constructed (e.g., MLP on $[\boldsymbol{h}_u, \boldsymbol{h}_v, \boldsymbol{e}_{uv}]$ or a separate edge network), and verify gradient flow from the edge term (e.g., report nonzero gradient norms attributable to that term on a batch). Add a diagnostic: compute the relevant correlations outside

training using the current definitions to check variance and expected magnitude; include a zoomed/inset plot for the edge-regularizer curve and report representative values with adequate precision. If the term is fundamentally ill-posed for the current architecture, remove it from claims and present it as future work, or replace it with a better-defined edge-aware auxiliary task (Sec. 3.5).

7. **Robustness/uncertainty is not quantified: results are reported for a single split and (apparently) a single seed on only 1000 graphs, limiting confidence in the stated performance and conclusions (Sec. 2.1.4, Sec. 2.4–2.5, Sec. 3.3–3.4).**

   Recommendation: Run multiple seeds and/or repeated random splits (or k-fold CV) and report mean ± std for MSE/R^2/MAE on the test set (Sec. 3.3). Mark the model-selection criterion (best validation epoch) and specify whether selection is by validation MSE or total loss (Sec. 2.4–2.5, Sec. 3.2). If compute is limited, at minimum add a sensitivity check over a few seeds and explicitly caveat conclusions in Sec. 3.5 and Sec. 4.

8. **Dataset provenance and representativeness are inadequately described. The manuscript does not clearly state the simulation name/code, cosmology, resolution, halo finder, merger-tree builder, selection cuts, mass/redshift range, or tree-size distributions—key context for assessing generality and potential leakage (e.g., correlated objects across splits) (Sec. 2.1, Sec. 3.1, Sec. 3.5, Sec. 4).**

   Recommendation: In Sec. 2.1, add a concise dataset paragraph: simulation and cosmology, box size and mass resolution, halo finder and tree builder, snapshots/redshift range, halo selection (mass cuts, central/satellite, etc.), and how 1000 trees were sampled. In Sec. 3.1, report distributions of halo mass, proxy values, and graph sizes (nodes/edges/depth). Briefly discuss possible split leakage risks (e.g., if trees share ancestry or are environmentally correlated) and how splits were constructed to mitigate this.

Minor issues

1. Edge feature usage is described inconsistently: Sec. 2.2 suggests per-edge concatenation "during message passing," while Sec. 3.2 suggests concatenating mean incoming edge attributes before the first GCN layer. These are materially different architectures and affect interpretability (Sec. 2.2, Sec. 3.2).

   Recommendation: Make the description consistent and match the implemented model. State the exact input dimension to the first GCN layer and where/when edge information is injected. If multiple variants were tried, separate them explicitly and indicate which produced the reported results.

2. Merger trees are directed acyclic graphs in time, but it is unclear whether directionality is preserved or lost (e.g., if the graph is symmetrized for GCN). Direction can be crucial for temporal/causal interpretation (Sec. 2.1–2.2).

   Recommendation: Clarify edge direction in `edge_index` and whether the convolution treats edges as directed. If the model uses an undirected GCN, discuss what information is lost and consider a directed/edge-aware alternative (e.g., separate in/out aggregations, GraphSAGE with direction tags, attention with edge features).

3. Pooling choice and variable tree sizes: global mean pooling can dilute information for graphs with widely varying node counts and depths (Sec. 2.2, Sec. 3.5).

   Recommendation: Report the distribution of node counts and consider comparing mean pooling with an alternative (attention pooling, Set2Set, or size-normalized variants). If not feasible, discuss this as a limitation and motivate mean pooling for this dataset (Sec. 3.5).

4. Performance analysis is largely global; it is unclear where the model succeeds/fails (e.g., early vs late forming halos, different mass ranges). This matters for any downstream assembly-bias use (Sec. 3.3, Sec. 3.5).

   Recommendation: Add stratified metrics (R^2/RMSE/MAE) in bins of true formation-time proxy and/or halo mass (Sec. 3.3 or Sec. 3.5). Add a calibration plot (binned mean predicted vs mean true). Optionally translate RMSE in scale factor to an approximate lookback-time/redshift uncertainty for interpretability.

5. Training/evaluation protocol details are scattered or missing (e.g., batch size, optimizer betas, initialization, dropout, scheduler, early stopping/model selection). This hinders reproduction (Sec. 2.4–2.5, Sec. 3.2–3.4).

   Recommendation: Consolidate all hyperparameters and training details into a single table in Sec. 2.4–2.5 (batch size, optimizer params, epochs, clipping, λ weights, hidden dims, layers, pooling, seed handling). State explicitly the validation metric used for selecting the reported checkpoint (Sec. 3.2).

6. Figure terminology and annotations are sometimes inconsistent ("assembly bias proxy" vs "formation time proxy"), and key quantitative context is often missing (sample size per panel, R^2/RMSE/MAE on the plot, binning choices). The edge-regularizer curve near zero is hard to interpret without higher precision (Figures 1–4; Sec. 3.3–3.4).

   Recommendation: Standardize naming across axes/titles/captions; define the proxy once and reuse the same label. Add on-plot annotations for N, R^2, RMSE/MAE, and binning. For the edge-regularizer, add an inset or rescaled y-axis and print representative values with sufficient decimal precision.

7. The manuscript discusses the edge-regularizer failure but does not empirically test proposed explanations (variance, batch effects, signal attenuation), leaving the discussion speculative (Sec. 3.4–3.5).

   Recommendation: Add at least one simple diagnostic: (i) direct correlation between raw edge features and candidate edge-embedding summaries; (ii) evaluate per-graph (not batch) correlations; (iii) test an alternative, more local edge objective on a subset. If not possible, explicitly label hypotheses as untested and outline concrete next steps (Sec. 3.5).

Very minor issues

1. Minor typographical/formatting inconsistencies: capitalization (e.g., "edge-Level Regularization"), broken line breaks within words, inconsistent heading styles, occasional notation inconsistencies (R-squared/$R2$/R^2; $y/z/\hat{y}$), and a possibly truncated loss expression with an undefined symbol (Sec. 2.3–2.5, Sec. 3.2, References).

   Recommendation: Proofread and standardize terminology, section formatting, and notation. Ensure the loss equation in Sec. 3.2 matches the formal definition in Sec. 2.3.4 exactly and remove/define any stray symbols. Use one consistent notation for R^2 and for targets/predictions (add a short notation table if helpful).

2. Keywords include topics not reflected in the manuscript (e.g., Black holes, Nucleosynthesis, Nebulae) (Abstract).

   Recommendation: Replace with content-accurate keywords (e.g., dark matter halos, merger trees, graph neural networks, halo formation time, assembly bias, cosmological simulations).

3. Citations and reference formatting: duplicated-year suffixes appear inconsistent in-text vs bibliography, and some references/attributions may not match the specific statements they support (Sec. 1, Sec. 2.1.1, References).

   Recommendation: Audit citations for relevance and correctness, standardize year-suffix usage (2024a/2024b, etc.), and ensure every in-text citation maps uniquely to a reference entry.

4. Figure readability: some plots appear low resolution with small fonts; axis limits/aspect ratios are not always ideal for 1:1 comparisons (e.g., predicted vs true) (Figures 1–4).

   Recommendation: Export higher-DPI or vector figures, increase font sizes, enforce 1:1 aspect ratio and matching axis limits for parity plots, and add light grid/reference lines where helpful.

Key statements and references

- • **Traditional approaches to modeling halo assembly bias typically use simplified proxies such as halo concentration or formation time, but recent work has shown that these proxies fail to fully capture the detailed information encoded in the full halo formation history, motivating the use of richer representations like merger trees instead (Montero-Dorta et al., 2021; Smith et al., 2024).**
- *Reference(s):* Montero-Dorta et al., 2021, Smith et al., 2024
- • **The merger tree, which traces the hierarchical assembly of a halo over cosmic time, has been demonstrated as a more complete and informative representation of halo formation history than simple scalar proxies, and recent studies have developed methods to exploit this structure for modeling halo assembly (Hearin et al., 2022; Nguyen et al., 2024; Ángel Chandro-Gómez et al., 2025).**
- *Reference(s):* Hearin et al., 2022, Nguyen et al., 2024, Ángel Chandro-Gómez et al., 2025
- • **Recent cosmological machine-learning studies have shown that Graph Neural Networks are effective for learning from graph-structured data such as halos and merger trees, including applications to inferring halo masses and galaxy/halo properties from graph representations (Villanueva-Domingo et al., 2023; Jagvaral et al., 2022; Garuda et al., 2024).**
- *Reference(s):* Villanueva-Domingo et al., 2023, Jagvaral et al., 2022, Garuda et al., 2024
- • **The definition of the new target variable z as a proxy for halo assembly bias—based on halo formation time approximated by the median scale factor of main-branch nodes in the merger tree—follows recent work that uses formation-time-based metrics derived from merger histories to characterize assembly bias (Contreras et al., 2021; Sunayama et al., 2022; Montero-Dorta et al., 2024).**
- *Reference(s):* Contreras et al., 2021, Sunayama et al., 2022, Montero-Dorta et al., 2024

Mathematical consistency audit

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

Maths relevance: light

The paper contains limited explicit mathematics: a single explicit loss-sum equation and verbal definitions of a target proxy (formation time) and correlation-based regularizers. The main audit focus is consistency of definitions (formation-time proxy), sign/optimization logic for correlation regularization, and whether the quantities being correlated are well-defined (scalars vs vectors, per-graph vs per-batch). Several key steps needed to verify the regularizers are omitted, leading to UNCERTAIN verdicts for core components.

Checked items

1. ✔ **Total loss weighted-sum form** (Intro (p.2) and Sec. 2.3.4 (p.4) and Eq. (1) Sec. 3.2 (p.5))
    - **Claim:** Total Loss = MSE Loss $+\lambda_{\text{node}}\cdot$ Node-Level Regularization $+\lambda_{\text{edge}}\cdot$ Edge-Level Regularization.
    - **Checks:** definition consistency, algebraic form
    - **Verdict:** PASS; confidence: high; impact: moderate
    - **Assumptions/inputs:** λ_{node} and λ_{edge} are scalar weights (hyperparameters)., All three terms are scalar losses.
    - **Notes:** The additive form is consistent across sections and is algebraically valid as written (a scalar sum of scalar terms).
2. ✖ **Formation-time proxy definition inconsistency** (Sec. 2.1.2 (p.3); Intro (p.2); Results Sec. 3.1 (p.4))
    - **Claim:** The target formation time proxy is defined from main-branch node scale factors.
    - **Checks:** symbol/definition consistency
    - **Verdict:** FAIL; confidence: high; impact: critical
    - **Assumptions/inputs:** mask_main identifies the main branch nodes., Each node has a scale factor feature.
    - **Notes:** Sec. 2.1.2 states it is 'approximated using the scale factor of the earliest node in the main branch' but then immediately says the median of main-branch scale factors is computed and assigned. Intro (p.2) and Results (p.4) describe the proxy as the median. These are different statistics (min vs median), so the target is not uniquely defined in the manuscript.
3. ✔ **Log-transform then global standardization** (Sec. 2.1.1 (p.2))
    - **Claim:** Apply $x[:,0] = \log(\text{mass})$, $x[:,1] = \log(\text{concentration})$, then normalize each feature to zero mean and unit variance using global dataset statistics; similarly normalize $\text{edge}_{\text{a}}\text{ttr}$.
    - **Checks:** domain sanity, definition coherence
    - **Verdict:** PASS; confidence: medium; impact: minor
    - **Assumptions/inputs:** mass and concentration are strictly positive so log is defined., Standardization uses (value-mean)/std with std> 0.

- **Notes:** As an analytic preprocessing pipeline, log then z-score is coherent. The paper does not state positivity explicitly, so correctness depends on the dataset, but no internal contradiction is present.

4. ✔ **Node-level regularization: sign logic** (Sec. 2.3.2 (p.3); Sec. 3.2 (p.5); Sec. 3.4 (p.7))
    - **Claim:** Node-level regularization is the negative Pearson correlation between (a scalar derived from) node embeddings and (normalized) scale factor, so minimizing total loss encourages correlation $\to +1$.
    - **Checks:** optimization sign sanity
    - **Verdict:** PASS; confidence: high; impact: moderate
    - **Assumptions/inputs:** Pearson correlation $\rho \in [-1, 1]$., Regularization term is defined as $-\rho$ (negative correlation).
    - **Notes:** If the regularizer is $-\mathbf{corr}$, then reducing the loss pushes **corr** upward toward $+1$; the paper's interpretation that a strongly negative term indicates **corr** close to $+1$ is mathematically consistent.

5. ⚠ **Node-level regularization: object being correlated (vector vs scalar)** (Sec. 2.3.2 (p.3) vs Sec. 3.2 (p.5))
    - **Claim:** Compute Pearson correlation between node embeddings and scale factor for each graph.
    - **Checks:** well-definedness, notation/definition consistency
    - **Verdict:** UNCERTAIN; confidence: high; impact: critical
    - **Assumptions/inputs:** Node embeddings are vectors $\boldsymbol{h}_v \in \mathbb{R}^d$.
    - **Notes:** Pearson correlation is defined between two scalar-valued lists. Sec. 3.2 clarifies a scalarization (mean over embedding dimensions) but Sec. 2.3.2 does not. Without an explicit equation for the scalar node score and the index set (all nodes vs main-branch nodes), the regularizer is not verifiable from the manuscript.

6. ⚠ **Node-level correlation computed 'for each graph'** (Sec. 2.3.2 (p.3); Sec. 3.2 (p.5))
    - **Claim:** Correlation is computed per graph (across nodes) between node scalar embedding values and node scale factors.
    - **Checks:** well-definedness, edge-case sanity
    - **Verdict:** UNCERTAIN; confidence: medium; impact: moderate
    - **Assumptions/inputs:** Each graph has at least **2** nodes included in the correlation., Variance of each variable across nodes is non-zero or a fallback is defined.
    - **Notes:** Pearson correlation is undefined if one variable has zero variance, or if fewer than **2** samples exist. The manuscript does not state masking rules or degeneracy handling (e.g., what happens for tiny trees or constant scale

factors in the selected node subset).

7. ⚠ **Edge-level regularization: definition depends on undefined 'edge embeddings'** (Sec. 2.3.3 (p.3); Sec. 3.2 (p.5); Sec. 3.4 (p.7))
    - **Claim:** Edge-level regularization is negative Pearson correlation between mean edge embeddings and median accretion rate (graph-level), computed across graphs in a batch.
    - **Checks:** well-definedness, missing derivation/definition
    - **Verdict:** UNCERTAIN; confidence: high; impact: moderate
    - **Assumptions/inputs:** The model produces an embedding per edge $e_{uv} \in \mathbb{R}^d$., A scalar per graph is derived from its edge embeddings.
    - **Notes:** The paper does not define how edge embeddings are computed (GCN layers typically output node embeddings, not edge embeddings) nor the exact scalarization/aggregation steps. Because the correlated quantities are not explicitly defined, the term cannot be audited symbolically.
8. ✖ **Edge-feature incorporation into GCN: inconsistent description** (Sec. 2.2 (p.3) vs Sec. 3.2 (p.5))
    - **Claim:** Edge attributes are incorporated into the GCN by concatenation in message passing.
    - **Checks:** definition consistency
    - **Verdict:** FAIL; confidence: high; impact: minor
    - **Assumptions/inputs:** Destination node features are augmented with edge-derived features.
    - **Notes:** Sec. 2.2 describes concatenating edge features to destination node features during message passing, implying a per-edge operation. Sec. 3.2 instead describes concatenating the mean of incoming edge attributes before the first GCN layer (a per-node pre-aggregation). These are materially different computations.
9. ✔ **Use of normalized vs raw scale factor in different roles** (Sec. 2.1.2 (p.3); Sec. 3.2 (p.5))
    - **Claim:** Target formation time proxy uses scale factor; node regularization correlates embeddings with normalized scale factor.
    - **Checks:** definition coherence, units/dimensional sanity
    - **Verdict:** PASS; confidence: medium; impact: minor
    - **Assumptions/inputs:** Scale factor is a node feature that is standardized before model input.
    - **Notes:** Using raw scale factor for the target while using normalized scale factor in a regularizer is not mathematically inconsistent (correlation is scale-invariant to affine transforms with positive scaling). The manuscript should

still state explicitly whether the target z is raw or normalized to avoid confusion.

10. ✔ **Interpretation of node-regularization value near -1** (Sec. 3.4 (p.7); Table 2 (p.6))
    - **Claim:** A node-level regularization term near -0.998 implies Pearson correlation near $+0.998$.
    - **Checks:** algebraic implication
    - **Verdict:** PASS; confidence: high; impact: minor
    - **Assumptions/inputs:** Regularization term = $-\mathbf{corr}$.
    - **Notes:** If $\mathbf{reg} = -\mathbf{corr}$, then $\mathbf{reg} \approx -0.998$ corresponds to $\mathbf{corr} \approx +0.998$, consistent with the stated interpretation.

Limitations

- The PDF provides only a high-level verbal description of the Pearson-correlation regularizers; it does not provide explicit mathematical formulas, masking/indexing conventions, or degeneracy handling, preventing full symbolic verification.
- No explicit GCN layer equations or message passing equations are given, so the correctness of the claimed edge-feature integration and any derived 'edge embeddings' cannot be checked analytically from the manuscript alone.
- No formal notation section is provided; several quantities ($\boldsymbol{y}$ vs $\boldsymbol{z}$, node embedding vs scalar node score) are referenced informally, which limits definitive consistency checking.

Numerical results audit

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

11 checks passed and **2** checks were uncertain due to inability to verify repeated-constant consistency across two separately extracted locations. No numeric mismatches were detected in the performed arithmetic/logic checks.

Checked items

1. ✔ **C1** (Page 3, Sec. 2.1.4 Data Splitting)
    - **Claim:** Used an 80/10/10 split, allocating 80% of the data to the training set, 10% to the validation set, and 10% to the test set (dataset size 1000 merger trees).
    - **Checks:** parts_vs_total
    - **Verdict:** PASS
    - **Notes:** Fractions sum to 1.0; implied counts are exactly 800/100/100 and sum to 1000.

2. ✔ **C2** (Page 5, Sec. 3.2 Model Architecture and Training (training split stated with counts))
    - **Claim:** The dataset was split into 80% training (800 graphs), 10% validation (100 graphs), and 10% test (100 graphs) out of 1000 merger trees.
    - **Checks:** percent_to_count_consistency
    - **Verdict:** PASS
    - **Notes:** Percent-to-count conversions and total count sum match exactly.
3. ✔ **C3** (Page 6, Sec. 3.3 Model Performance on Test Set + Table 2)
    - **Claim:** An R^2 score of 0.4838 indicates the model can explain approximately 48.38% of the variance.
    - **Checks:** decimal_to_percent_conversion
    - **Verdict:** PASS
    - **Notes:** 48.38% equals 100×0.4838 exactly.
4. ✔ **C4** (Page 6, Sec. 3.3 Model Performance on Test Set)
    - **Claim:** The square root of the MSE (RMSE) would be 0.02, given MSE = 0.0004.
    - **Checks:** derived_quantity_sqrt
    - **Verdict:** PASS
    - **Notes:** $\sqrt{0.0004} = 0.02$ matches the reported RMSE.
5. ✔ **C5** (Page 6, Sec. 3.3 Model Performance on Test Set + Page 5 Table 1)
    - **Claim:** The MSE of 0.0004 is relatively small, considering the target variable's standard deviation of 0.025; RMSE 0.02 is comparable to the target's standard deviation.
    - **Checks:** unit_consistent_comparison
    - **Verdict:** PASS
    - **Notes:** RMSE computed from MSE equals 0.02; ratio RMSE/target_std = 0.8, consistent with 'comparable'.
6. ✔ **C6** (Page 7, Sec. 3.4 Analysis of Loss Function and Training Dynamics)
    - **Claim:** The node-level regularization term is the negative Pearson correlation; a value approaching -0.9987 signifies the Pearson correlation itself became very close to $+1$.
    - **Checks:** sign_inversion_consistency
    - **Verdict:** PASS
    - **Notes:** Arithmetic sign flip is consistent: $\text{corr} = -(-0.9987) = +0.9987$.
7. ✔ **C7** (Page 6 Table 2 + Page 7 narrative)

- **Claim:** Table 2 reports node-level regularization term -0.9985; this corresponds to Pearson correlation $+0.9985$ (since term is negative Pearson correlation).
- **Checks:** sign_inversion_consistency
- **Verdict:** PASS
- **Notes:** Arithmetic sign flip is consistent: $\mathbf{corr} = -(-0.9985) = +0.9985$.

8. ⚠ **C8** (Page 4, Sec. 2.4 Training Procedure + Page 5, Sec. 3.2 Training)
   - **Claim:** Training used Adam optimizer with learning rate 0.001, weight decay 0.0001, and gradient clipping value 1.0; trained for 100 epochs.
   - **Checks:** repeated_constant_match
   - **Verdict:** UNCERTAIN
   - **Notes:** Only one extraction available; verifying equality across both cited sections requires separately extracted values from each location.

9. ✔ **C9** (Page 5, Sec. 3.1 + Table 1)
   - **Claim:** Raw formation time proxy ranges from approximately 0.500 to 0.684, with mean= 0.550, median= 0.550, and std dev= 0.025.
   - **Checks:** range_and_summary_self_consistency
   - **Verdict:** PASS
   - **Notes:** min≤mean≤max and min≤median≤max; std is non-negative.

10. ✔ **C10** (Page 5 Table 1 (Normalized Node Features))
    - **Claim:** Normalized node features have std dev of 1.000 for each feature ($\mathbf{log(mass)}$, $\mathbf{log(concentration)}$, V_{max}, scale factor).
    - **Checks:** repeated_constant_match
    - **Verdict:** PASS
    - **Notes:** All four listed std dev values are identical and equal to 1.0.

11. ✔ **C11** (Page 5 Table 1 (Normalized Edge Feature))
    - **Claim:** Normalized accretion rate shows median $= -0.142$ and min $= -0.142$.
    - **Checks:** median_equals_min_check
    - **Verdict:** PASS
    - **Notes:** Median equals minimum exactly as reported (a plausibility flag noted in the check instructions, not an arithmetic inconsistency).

12. ✔ **C12** (Page 7, Sec. 3.4)
    - **Claim:** Best validation loss of -0.0997 was achieved at epoch 94 within 100 epochs.
    - **Checks:** bound_check
    - **Verdict:** PASS

- **Notes:** Best epoch (**94**) is within **1..100**; best validation loss is finite.

13. ⚠ **C13** (Page 5, Sec. 3.2 (loss definition) + Page 5 Eq. (1))
    - **Claim:** Total Loss is defined as MSE Loss $+\lambda_{\text{node}}\cdot$Node-Level Regularization $+\lambda_{\text{edge}}\cdot$Edge-Level Regularization, with $\lambda_{\text{node}} = \mathbf{0.1}$ and $\lambda_{\text{edge}} = \mathbf{0.1}$.
    - **Checks:** repeated_constant_match
    - **Verdict:** UNCERTAIN
    - **Notes:** Only one extraction available; verifying consistency between equation and narrative requires separately extracted values from each mention.

Limitations

- Only parsed text from the provided PDF pages was used; no external data or code execution was available.
- Exact numerical verification of results that depend on underlying datasets, model outputs, or training logs cannot be performed from the PDF alone.
- Plot-based numeric extraction was avoided per instructions, so any values only visible in figures (without explicit text/tabulation) were not proposed as fast checks.
- Some repeated-constant cross-references (e.g., hyperparameters and loss weights) could not be fully checked because only a single extracted numeric instance was available for comparison.