Skepthical review: *3.2. Baseline model performance*

Summary

The paper tackles a well-motivated astrophysical/ML problem: predicting, for each progenitor→descendant transition in CAMELS-SAM merger trees, whether a halo's concentration increases or instead decreases/does not change (Sec. 2.1–2.1.3). The authors propose a merger-tree GNN that ingests node-level halo properties plus global cosmological parameters and is trained with a combined supervised BCE classification loss and an NT-Xent–style *label-supervised* contrastive loss (Sec. 2.3–2.3.3). They compare against a feature-engineered Random Forest baseline that includes intrinsic/temporal/cosmology features and explicit environment/merger descriptors (Sec. 2.2, Sec. 2.4). Empirically, the RF reaches weighted F1 $\approx \mathbf{0.63}$ on a validation subset (Sec. 3.1–3.2), while the GNN—trained under severe compute limits on only **10–50** trees for $\sim \mathbf{2}$ epochs and evaluated on a very small reduced test subset—achieves weighted F1 $\approx \mathbf{0.49}$ with strong class-1 bias (Sec. 3.3–3.5). As written, the study is a promising proof-of-concept, but key elements of the task definition, leakage-safety/causality assumptions, and unified evaluation protocol need tightening, and the current GNN evidence is too preliminary to support strong comparative claims versus the RF.

Strengths

- The prediction task (direction of concentration change) is clearly motivated and potentially useful as an alternative to full regression, with direct physical relevance (Introduction, Sec. 1).
- Using merger trees as graphs and a GNN to exploit hierarchical assembly history is conceptually well aligned with the data modality (Sec. 2.1–2.3).
- The RF baseline is thoughtfully constructed and interpretable, with engineered environment/merger-history features and hyperparameter tuning/feature-importance analysis (Sec. 2.2, Sec. 2.4, Sec. 3.2).
- The manuscript is candid about computational constraints and includes ablations (cosmology inputs; contrastive term) that expose failure modes like class bias and embedding non-separation (Sec. 3.3–3.5).
- Preprocessing/normalization is discussed, including the intent to use train-only statistics, which is good practice (Sec. 2.5).
- The presentation includes many diagnostic plots (feature distributions, confusion matrices, embeddings) that can become highly informative once paired with clearer protocols and quantitative annotations (Sec. 3; Figs. 1–14).

Major issues

1. **Central methodological claim is not supported by the current GNN experiments: the GNN is trained on only $\sim$ 10–50 trees for $\sim$ 2 epochs (a tiny fraction of the $\sim$ 15k available trees), and results are reported on an extremely small reduced test subset, yielding clear underfitting and strong class-1 bias (Sec.**

2.5.1–2.5.2, Sec. 3.3–3.5, Conclusion). With such limited training, it is not possible to assess whether the architecture/contrastive objective is competitive or whether observed ablation effects are robust.

Recommendation: Either (i) substantially scale up GNN training (more trees and epochs, to near-convergence) and report learning curves (train/val loss, macro/weighted F1 vs epoch) plus final results on the official held-out test set (e.g., CS_tree_test.pt), or (ii) explicitly reframe the paper as a compute-limited pilot/proof-of-concept and avoid comparative performance conclusions. In both cases, quantify for every GNN run: #trees, #transitions, class counts, #epochs, hardware/time budget, and provide variability across multiple random seeds/subsamples (mean±std), since small-subset metrics are very noisy.

2. **Evaluation protocol and data partitioning are not unified across RF and GNN, undermining comparability and risking leakage/confounding from correlated transition samples within trees (Sec. 2.1, Sec. 2.5.1, Sec. 3.1–3.3).** The RF is tuned/evaluated on larger validation subsets, while the GNN reports results on a reduced test subset; it is unclear whether both models are ever evaluated on the same trees/transitions or on the full provided test split.

   Recommendation: In Sec. 2.1 and Sec. 2.5, provide a single explicit splitting description at the *tree level* (no tree in multiple splits) and report, for each split used in each experiment: #trees, #nodes, #edges/transitions, and per-class supports. Then enforce a common evaluation: (a) evaluate both RF and GNN on the same held-out test set (preferably the full CS_tree_test.pt), or (b) if compute forces reduced subsets, use identical reduced subsets (same tree IDs) for both models. Add uncertainty estimates (multiple seeds/subsamples) for all headline metrics in Sec. 3.2–3.5.

3. **Task definition and label construction are under-specified in a way that affects the physical meaning of samples and the learning problem (Sec. 2.1.3, Sec. 3.1).** In particular: (i) merger trees can have multiple progenitors per descendant—it's unclear whether transitions are drawn from all progenitor→descendant edges or only from the main progenitor branch; (ii) grouping "decrease" and "no change" into class 0 is not quantified (ties may be rare in float), and threshold sensitivity is not explored; (iii) concentration comparisons appear to use $\boldsymbol{x}[u, 1]$ where $\boldsymbol{x}$ is $\log_{10}$(concentration) (and possibly standardized), which is monotonic but matters if any ϵ/rounding is used.

   Recommendation: In Sec. 2.1.3, specify precisely which edges become samples (all progenitor edges vs main branch only) and justify the physical interpretation for minor-progenitor edges if included. Report the fraction of exact ties and consider a tolerance δ (increase if $\Delta c > \delta$) or a three-way split (decrease / ∼constant / increase) at least as a sensitivity check. Explicitly state whether $Y_{\text{transition}}$ uses linear concentration c, $\log_{10}(c)$, or standardized $\log_{10}(c)$, and keep notation consistent.

4. **Potential information leakage / unclear forecasting setting: the RF feature set appears to include descendant and time-of-descendant information (e.g., mass_v, sf_v, Δsf), and the GNN message passing may aggregate from "future" (descendant) nodes depending on edge direction and whether reverse edges/undirected conversion is used (Sec. 2.2, Sec. 2.3.1, Sec. 2.5.1, Sec. 3.2–3.4).** Without an ex-

plicit statement of whether the task is (a) causal forecasting using information available at time $\mathbf{sf}_u$ only or (b) retrospective classification using both endpoints, the scientific interpretation and fairness of comparisons are unclear.

Recommendation: Define the intended prediction setting in Sec. 1/Sec. 2.1.3: forecasting (use only information available at $\mathbf{sf}_u$) vs retrospective (allow using $\mathbf{sf}_v$ and/or node v features). If forecasting, remove descendant features from the RF (or clearly separate results), and for the GNN restrict message passing and subgraphs so embeddings for u cannot aggregate from nodes with $\mathbf{sf} > \mathbf{sf}_u$ (and explicitly document whether reverse edges are added and which PyG operators are used). If retrospective, make that explicit and ensure both RF and GNN have access to comparable endpoint information.

5. **Mismatch between the edge-level target and the stated GNN prediction head: the label is per transition ($u \to v$), but the classifier is described as using only the progenitor embedding z_u (Sec. 2.3.3, Sec. 3.3).** This discards explicit information about v and about the time interval $\Delta\mathbf{sf}$, and makes it harder to interpret what the model is learning—especially in trees where nodes can participate in multiple edges.

   Recommendation: Reformulate the GNN as an explicit edge/transition classifier: predict using a function of both endpoints and interval, e.g., $h = \mathrm{MLP}([z_u, z_v, z_u - z_v, z_u \odot z_v, \Delta\mathbf{sf}])$ or a dedicated edge network. Clearly state the transition-to-embedding mapping under PyG batching (Sec. 2.5.1). If you keep a node-only head, justify why z_u alone should determine the sign of concentration change and clarify how multi-edge cases are handled.

6. **GNN architecture/training and contrastive objective are underspecified, limiting reproducibility and making it difficult to diagnose class bias and underfitting (Sec. 2.3.1–2.3.3, Sec. 2.5.1–2.5.2).** There are inconsistencies in wording (GraphConv vs GraphSAGE), missing hyperparameters (hidden sizes, #layers, aggregation type, dropout/residuals, normalization), and incomplete contrastive details (multi-positive handling, what happens with zero positives in a batch, batch size in transitions, label balance). The loss is effectively supervised contrastive learning, but is not labeled as such.

   Recommendation: Expand Sec. 2.3 and Sec. 2.5 to fully specify: exact PyG conv operator(s) and settings, #layers, hidden dims, activations, normalization, dropout/residuals, optimizer/scheduler, learning rate/weight decay, batch construction (trees per batch; transitions per batch), α and τ values, and projection-head architecture. Clarify the contrastive formulation as supervised contrastive learning; specify whether positives are all same-label samples in-batch, how multiple positives are aggregated, and how anchors with no positives are treated. Consider class-balanced batching and/or class-weighted BCE/focal loss to address the observed degeneracy toward class 1 (Sec. 3.4).

7. **Baseline feature specification and "fairness" of the RF vs GNN comparison are unclear: the RF uses hand-engineered environment/merger features (counts/mass of partners, major-merger flags), while the GNN's ability to recover comparable information via message passing is not analyzed, and the RF feature dimensionality is inconsistent across sections (Sec. 2.2, Sec. 2.4, Sec. 3.1–3.2; Table 1; Fig. 6).**

Recommendation: Provide a definitive RF feature list (ordered vector, final dimensionality, units/transforms) and reconcile all counts across Sec. 2.2/Sec. 2.4/Sec. 3 and Table 1. Add an ablation for parity: RF with only intrinsic+cosmology (+$\mathbf{\Delta sf}$) but without engineered environment features, and compare it to the GNN on the same split (Sec. 3.5). Separately, clarify the major-merger definition and justify/cite a conventional threshold. This will make claims about "graph-based learning vs feature engineering" more defensible.

8. **Internal inconsistencies in reported metrics and model specification reduce confidence in results: (i) Table 2 weighted-average recall appears inconsistent with the displayed per-class recalls and supports; (ii) the manuscript states a 2-logit output head but uses BCEWithLogits against a single binary target (major methodological mismatch) (Table 2; Sec. 2.3.3; Sec. 3.4).**

   Recommendation: Audit the evaluation pipeline and correct Table 2 (or explain any non-standard averaging). Make the classification head/loss mathematically consistent: either use a single logit with BCEWithLogits for $Y \in \mathbf{0, 1}$, or use 2 logits with softmax cross-entropy for a class index. State the exact target encoding and loss implementation details.

Minor issues

1. Limited physical/error analysis reduces scientific insight: the paper reports overall metrics and feature importances, but does not quantitatively connect errors (especially concentration decreases) to regimes of mass, redshift/scale factor, $\mathbf{\Delta sf}$, or merger environment (Sec. 3.2–3.5, Conclusion).

   Recommendation: Add stratified performance analyses for both RF and GNN: metrics vs progenitor mass, progenitor concentration, $\mathbf{\Delta sf}$, scale factor/redshift, and merger indicators (major merger presence, number/total mass of partners). Include a small set of qualitative case studies (a few trees/branches) showing correct vs incorrect predictions for decreases and discuss plausible physical drivers (e.g., major mergers, stripping).

2. Contextual baselines and class-imbalance handling are not sufficiently reported, particularly for the reduced GNN subsets (Sec. 3.1–3.4). Without trivial baselines, it is hard to interpret weighted F1/ROC-AUC under class bias.

   Recommendation: Report majority-class and random-prior baselines, and optionally a simple logistic regression on a minimal feature set (e.g., progenitor concentration + mass +$\mathbf{\Delta sf}$). Report class proportions for every split actually used (especially the $\mathbf{10/5/5}$-tree GNN setup) and whether any class weighting or threshold tuning is performed.

3. Figures and tables often lack quantitative annotations needed for standalone interpretability (sample sizes per panel, uncertainty across seeds/subsamples, explicit metric definitions), and some plots (e.g., t-SNE embeddings) are difficult to interpret given undertrained models (Figs. 1–14; Sec. 3.3–3.5).

   Recommendation: Add sample sizes, uncertainty (error bars/CI or mean±std across runs), and explicit metric definitions to captions. For embedding plots, report t-SNE/UMAP hyperparameters (perplexity, seed, normalization) and consider moving them to an appendix unless they are linked to a trained/converged model.

4. Related work and positioning could be more systematic, particularly around (i) GNNs on merger trees/cosmological graphs and (ii) supervised contrastive learning (Introduction, Sec. 2.3.2).

   Recommendation: Expand related work to clearly state what is novel here (transition-level concentration-direction classification; cosmology conditioning; supervised contrastive objective) and how it differs from prior merger-tree ML and supervised-contrastive frameworks. Ensure citations are specific to claims made.

5. Several presentation/readability issues in figures (small fonts, dense layouts, color reliance) reduce accessibility (Figs. 1–14).

   Recommendation: Increase font sizes, use colorblind-safe palettes and/or markers/patterns, avoid relying on color alone, and standardize axis labels/units and legend terminology.

Very minor issues

1. Notation and variable naming sometimes blur raw physical quantities vs transformed/standardized features (e.g., conc$*wusedforx[u,1]wherex[u,1]is\log*$(concentration) and possibly standardized; ambiguous linear vs log handling in environmental mass features) (Sec. 2.1.3, Sec. 2.2, Sec. 2.3).

   Recommendation: Disambiguate notation: use c for linear concentration, $\ell c = \log_{10}(c)$, and $\tilde{\ell c}$ for standardized values; define $Y_{\text{transition}}$ using one explicitly. Similarly, explicitly define whether masses in sums/ratios are linear or log-transformed and apply conversions consistently.

2. Minor formatting/copy-editing inconsistencies: duplicated years in citations, inconsistent quoting/monospace for filenames and variables, inconsistent hyphenation/capitalization, inconsistent symbol naming for α (Sec. 2–3; References).

   Recommendation: Proofread for consistency in citations, file/variable formatting, hyphenation, capitalization, and parameter notation (e.g., $\alpha_{\text{contrastive}}$).

3. Section heading formatting appears inconsistent (e.g., stray markdown delimiters like "####", inconsistent numbering styles) (Sec. 2.5.2, Sec. 3, Sec. 4).

   Recommendation: Standardize section/subsection heading formatting and numbering throughout to improve polish and navigability.

Key statements and references

- • **The concentration of a dark matter halo undergoes complex, non-monotonic evolution over cosmic time, driven by continuous accretion, stochastic merger events, and the cosmological environment, such that a halo's concentration can either increase or decrease over a given interval depending on its mass accretion rate, merger type and timing, and cosmic epoch, which makes it difficult for traditional analytical models and empirical relations to capture the full range of evolutionary pathways and often leads to oversimplifications in galaxy formation simulations (Okoli, 2017; Wang et al., 2020).**

- *Reference(s):* Okoli, 2017, Wang et al., 2020
- • **Merger trees are well-established as ideal data structures for representing dark matter halo assembly histories because they inherently encode hierarchical relationships between progenitors and descendants and capture merger events over cosmic time, and multiple algorithms and comparisons for generating such trees have been developed and analyzed in the literature (Parkinson et al., 2007; Jiang and van den Bosch, 2013; Yung et al., 2024).**
- *Reference(s):* Parkinson et al., 2007, Jiang and van den Bosch, 2013, Yung et al., 2024
- • **The CAMELS-SAM simulations constitute a suite specifically designed to explore the interplay between cosmology and galaxy formation, providing merger trees and halo assembly histories across a diverse range of cosmological parameters, and have been proposed as a multiscale, multiview, multitask benchmark for geometric deep learning in cosmology (Ramakrishnan and Velmani, 2022; Lovell et al., 2024; Huang et al., 2025).**
- *Reference(s):* Ramakrishnan and Velmani, 2022, Lovell et al., 2024, Huang et al., 2025
- • **Global cosmological parameters such as the present-day matter density parameter Ω_m and the amplitude of matter fluctuations σ_8 are known to influence halo evolution and substructure, and their variation has been studied in simulations to quantify effects on halo properties and clustering (Dooley et al., 2014; Ishiyama et al., 2025; Wu et al., 2024).**
- *Reference(s):* Dooley et al., 2014, Ishiyama et al., 2025, Wu et al., 2024
- • **The Normalized Temperature-scaled Cross-Entropy (NT-Xent) loss is a standard contrastive learning objective that, for an anchor embedding and a positive partner, computes a log-softmax over cosine similarities scaled by a temperature parameter τ, and has been theoretically analyzed (including upper bounds) and empirically applied in metric learning and multimodal retrieval contexts (Ågren, 2022; Bleeker and de Rijke, 2022; Fahim et al., 2024; Steck et al., 2024).**
- *Reference(s):* Ågren, 2022, Bleeker and de Rijke, 2022, Fahim et al., 2024
- • **Feature scaling and standardization (subtracting the mean and dividing by the standard deviation computed on the training set) are known to significantly affect classification performance by preventing features with larger magnitudes from dominating learning, and systematic studies have shown that the choice of scaling technique can materially impact both regression and classification outcomes (de Amorim et al., 2022; Islam, 2024; Pinheiro et al., 2025).**
- *Reference(s):* de Amorim et al., 2022, Islam, 2024, Pinheiro et al., 2025

Mathematical consistency audit

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

Maths relevance: light

The paper contains a small set of central mathematical definitions: (i) feature definitions (log-transformed halo properties and scale factor), (ii) a binary target defined via an inequality along directed progenitor→descendant edges, (iii) a generic message-passing update for the GNN, (iv) an NT-Xent-style contrastive loss using cosine similarity, and (v) a combined loss with a weighting parameter. The main internal-consistency problems are mismatches between stated tensor shapes and stated losses (2-logit output vs BCE) and inconsistent bookkeeping of the baseline feature vector dimensionality and composition.

Checked items

1. ✔ **Scale factor definition** (Sec. 2.1.1, p.3)
    - **Claim:** Scale factor is defined as $a = 1/(1+z)$.
    - **Checks:** definition consistency, dimensional/units sanity
    - **Verdict:** PASS; confidence: high; impact: minor
    - **Assumptions/inputs:** z denotes cosmological redshift
    - **Notes:** Definition is self-contained and dimensionless as expected.
2. ✔ **Node feature vector content and indexing for concentration** (Sec. 2.1.1 and Sec. 2.1.3, p.3)
    - **Claim:** Node features $\boldsymbol{x}$ are $[\log_{10}(\mathbf{mass}), \log_{10}(\mathbf{concentration}), \log_{10}(V_{\mathrm{max}}), \mathbf{scale_factor}]$, and conc_u is taken as $\boldsymbol{x}[u, 1]$.
    - **Checks:** symbol/definition consistency, sanity check
    - **Verdict:** PASS; confidence: medium; impact: minor
    - **Assumptions/inputs:** $\boldsymbol{x}[u, 1]$ uses 0-based indexing as in Python/PyTorch
    - **Notes:** Indexing is consistent with the stated ordering. However, the symbol conc$$u denotes \log(concentration)(and possibly standardized), not raw concentration; the inequality-based label remains invariant under log/standardization if std>0$, but naming is potentially misleading.
3. ✔ **Transition definition on directed edges** (Sec. 2.1.3, p.3)
    - **Claim:** A transition sample is defined for each directed edge (u, v) with $\mathbf{sf}_v > \mathbf{sf}_u$.
    - **Checks:** logical consistency, definition consistency
    - **Verdict:** PASS; confidence: high; impact: minor
    - **Assumptions/inputs:** scale factors increase along forward cosmic time
    - **Notes:** Condition $\mathbf{sf}_v > \mathbf{sf}_u$ ensures time-forward transitions; consistent with stated goal.

4. ✔ **Binary target definition for concentration direction** (Sec. 2.1.3, p.3; reiterated Sec. 3.1, p.6)

   - **Claim:** $Y_{\mathrm{transition}} = 1$ if $\mathrm{conc}_v > \mathrm{conc}_u$ else 0 (including equality in class 0).
   - **Checks:** logical consistency, sanity cases
   - **Verdict:** PASS; confidence: high; impact: moderate
   - **Assumptions/inputs:** conc_u and conc_v are comparable scalars for progenitor and descendant on the chosen scale (raw/log/standardized)
   - **Notes:** The binary split is well-defined. Equality assigned to class 0 is explicit.

5. ⚠ **Baseline environmental feature: total mass of other partners** (Sec. 2.2, p.4)

   - **Claim:** Total mass of other progenitors is computed as the sum of $10^{\log_{10}(\mathrm{mass})}$ over progenitors of v excluding u.
   - **Checks:** dimensional/units consistency, symbol/definition consistency
   - **Verdict:** UNCERTAIN; confidence: medium; impact: minor
   - **Assumptions/inputs:** $\log_{10}(\mathbf{mass})$ refers to base-10 log of mass in solar masses, the summation is intended in linear mass units
   - **Notes:** Using $10^{\log_{10}(\mathrm{mass})}$ correctly maps a log-mass feature back to linear mass for summation. But later ratios use mass_u without explicitly applying the same inverse-log transform, so it is unclear whether mass_u is linear or log-space in those formulas.

6. ⚠ **Baseline environmental feature: mass ratio definition** (Sec. 2.2, p.4)

   - **Claim:** Mass ratio is $\mathrm{mass}_u/(\mathbf{total_mass_of_other_merging_partners} + \epsilon)$.
   - **Checks:** dimensional/units consistency, symbol consistency
   - **Verdict:** UNCERTAIN; confidence: medium; impact: minor
   - **Assumptions/inputs:** mass_u and total_mass_of_other_merging_partners are in the same units (linear mass)
   - **Notes:** Ratio is dimensionless if both masses are linear. The text does not explicitly define mass$uaslinearmassvs \log_u)$, creating ambiguity.}(\mathrm{mass

7. ✖ **Baseline feature count and composition** (Sec. 2.2 (p.4), Sec. 2.4 (p.5), Sec. 3.1–3.2 (pp.6–8))

   - **Claim:** The Random Forest uses a fixed-size engineered feature vector, described as 12 features (Sec. 2.4) and also as 10 features (Secs. 3.1–3.2), with varying inclusion of descendant $\log_{10}(\mathbf{mass}_v)$.
   - **Checks:** definition consistency, bookkeeping consistency
   - **Verdict:** FAIL; confidence: high; impact: moderate
   - **Assumptions/inputs:** Feature lists in different sections are intended to describe the same model
   - **Notes:** Feature-count arithmetic conflicts across the paper: Sec. 2.2 enumerates 4 progenitor + 2 descendant + 1 temporal + 2 cosmo + 4 environmental = $\mathbf{13}$; Sec. 2.4 states $\mathbf{12}$; Sec. 3.1 claims $\mathbf{10}$ but lists components summing to $\mathbf{12}$; Sec. 3.2 also

claims 10 while listing 12 components. This prevents verifying what vector is actually used.

8. ✔ **Generic message passing update equation** (Sec. 2.3.1, p.4)

    - **Claim:** Node embeddings are updated via $h_u^{(\ell+1)} = \mathrm{UPDATE}(h_u^{(\ell)}, \mathrm{AGGREGATE}(h_v^{(\ell)} | v \in \mathcal{N}(u)))$.
    - **Checks:** notation consistency, shape/typing sanity
    - **Verdict:** PASS; confidence: high; impact: minor
    - **Assumptions/inputs:** UPDATE and AGGREGATE are well-defined functions per layer
    - **Notes:** Abstract formulation is self-consistent and type-correct at the level presented.

9. ✔ **Concatenation of cosmological parameters into node features** (Sec. 2.3.1, p.4; reiterated Sec. 3.3, p.9)

    - **Claim:** Two graph-level parameters (Ω_m, σ_8) are concatenated to each node's 4 intrinsic features, yielding 6-dimensional node inputs.
    - **Checks:** dimension consistency, definition consistency
    - **Verdict:** PASS; confidence: high; impact: minor
    - **Assumptions/inputs:** Ω_m and σ_8 are constant across all nodes in a tree
    - **Notes:** $4 + 2 = 6$ is consistent; treating global parameters as per-node constants is coherent.

10. ✔ **Cosine similarity definition** (Sec. 2.3.2, p.5)

    - **Claim:** $\mathrm{sim}(a, b) = \frac{a \cdot b}{|a| \cdot |b|}$.
    - **Checks:** algebra/definition check, domain conditions
    - **Verdict:** PASS; confidence: high; impact: minor
    - **Assumptions/inputs:** a and b are nonzero vectors in the same embedding space
    - **Notes:** Definition is standard; implicit requirement $|a|, |b| \neq 0$ is not stated but typical.

11. ✔ **NT-Xent contrastive loss expression** (Sec. 2.3.2, p.5)

    - **Claim:** For anchor z_i and positive z_p: $\mathcal{L} = -\log\left(\frac{\exp(\mathrm{sim}(z_i, z_p)/\tau)}{\sum_{k \in \mathrm{batch}, k \neq i} \exp(\mathrm{sim}(z_i, z_k)/\tau)}\right)$.
    - **Checks:** algebraic form, normalization/sanity
    - **Verdict:** PASS; confidence: medium; impact: minor
    - **Assumptions/inputs:** $\tau > 0$, Batch contains at least one other sample besides i
    - **Notes:** Expression is internally consistent. However, the paper defines positives/negatives by labels and likely allows multiple positives; it does not specify how z_p is chosen or how multiple positives are aggregated, so the exact batch objective is underspecified.

12. ✔ **Combined loss definition** (Sec. 2.3.3, p.5)

    - **Claim:** Total loss is $\mathcal{L}{\rm total} = \mathcal{L}$.} + \alpha \cdot \mathcal{L}_{\rm contrastive

- **Checks:** symbol consistency, typing/shape sanity
- **Verdict:** PASS; confidence: high; impact: minor
- **Assumptions/inputs:** Both terms are scalar losses per batch (or averaged per sample) on compatible scales
- **Notes:** Additive scalar objective is consistent as written.

13. ✖ **Classification head output vs stated classification loss** (Sec. 2.3.3, p.5)

    - **Claim:** MLP outputs a **2**-dimensional logit vector for two classes; classification loss is Binary Cross-Entropy (BCE) with logits comparing logits against a binary target $Y_{\text{transition}}$.
    - **Checks:** shape/typing consistency, loss-target consistency
    - **Verdict:** FAIL; confidence: high; impact: critical
    - **Assumptions/inputs:** $Y_{\text{transition}}$ is a scalar in $\mathbf{0, 1}$ as defined in Sec. 2.1.3
    - **Notes:** As stated, there is a shape/definition mismatch: a **2**-logit (two-class) output is naturally paired with a categorical cross-entropy against a class index (or with a one-hot target of matching dimension). BCE-with-logits is naturally paired with a single logit for binary classification (or multi-label with multi-dimensional targets). The paper does not define a **2**D target encoding, so the stated loss and output are not mathematically consistent.

Limitations

- The PDF does not provide the explicit standardization formula (though it is standard); audit assumes mean-subtract and divide-by-std with positive std.
- Implementation choices (e.g., whether a one-hot target was used, whether a single logit was actually produced, or how positives are sampled for contrastive loss) are not specified in math; where such details are required to verify consistency, items are marked UNCERTAIN/FAIL based only on the written description.
- No explicit equations are numbered; locations are referenced by section and page.

Numerical results audit

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

Executed **20** numeric consistency checks: **19** PASS and **1** FAIL. The single failure is a substantial mismatch in Table **2** weighted recall versus the value implied by per-class recalls and supports. All other tested identities (dataset/support totals, percent↔decimal conversions, rounding consistency, feature-dimension arithmetic, and several macro/weighted metric recomputations) are consistent within stated tolerances.

Checked items

1. ✔ **C1** (Page 3, Section 2.1 (dataset files))

    - **Claim:** The merger trees are organized into three files with counts: train **14,997**; val **5,099**; test **4,900**.
    - **Checks:** sum_of_parts

- **Verdict:** PASS
- **Notes:** Computed total trees = **24,996** from **14,997** + **5,099** + **4,900**; no explicit total stated to compare against.

2. ✔ **C2** (Page 6, Section 3.1 (development subset target distribution))
   - **Claim:** From **110,838** transitions, there are **51,707** class **0** and **59,131** class **1** instances.
   - **Checks:** parts_equal_total
   - **Verdict:** PASS
   - **Notes:** **51,707** + **59,131** = **110,838** (exact).
3. ✔ **C3** (Page 8, Section 3.2 + Table 1 (validation subset size))
   - **Claim:** Validation subset has **109,866** transitions; Table **1** support is **51,259** (class **0**) and **58,607** (class **1**), total **109,866**.
   - **Checks:** parts_equal_total
   - **Verdict:** PASS
   - **Notes:** **51,259** + **58,607** = **109,866** (exact), consistent between narrative and table.
4. ✔ **C4** (Page 8, Section 3.2 (baseline training subset))
   - **Claim:** Baseline hyperparameter tuning used **200** training trees with **221,770** transitions.
   - **Checks:** rate_per_tree_sanity
   - **Verdict:** PASS
   - **Notes:** Derived average transitions per tree = **221,770**/**200** = **1,108.85**.
5. ✔ **C5** (Page 6, Section 3.1 (development subset transitions vs trees))
   - **Claim:** From a subset of **100** trees, **110,838** transitions were extracted.
   - **Checks:** rate_per_tree_sanity
   - **Verdict:** PASS
   - **Notes:** Derived average transitions per tree = **110,838**/**100** = **1,108.38**.
6. ✔ **C6** (Page 9, Section 3.3 (GNN reduced dataset sizes))
   - **Claim:** GNN reduced dataset: **10** training trees (**13,162** transitions), **5** validation trees (**4,570** transitions), **5** test trees (**3,723** transitions).
   - **Checks:** rate_per_tree_sanity_and_sums
   - **Verdict:** PASS
   - **Notes:** Derived avgs transitions/tree: train **1,316.2**; val **914.0**; test **744.6**. Total reduced transitions = **21,455**.
7. ✔ **C7** (Page 10, Table 2 (reduced test set supports))
   - **Claim:** Table **2** support: **1,674** (class **0**) and **2,049** (class **1**), total **3,723**.
   - **Checks:** parts_equal_total
   - **Verdict:** PASS

- **Notes:** 1,674 + 2,049 = 3,723 (exact).

8. ✔ **C8** (Page 10, Table 2 (macro/weighted F1 from per-class + support))
   - **Claim:** In Table 2, per-class F1 are 0.23 (class 0) and 0.69 (class 1); macro avg F1 is 0.46; weighted avg F1 is 0.48 given supports 1,674 and 2,049.
   - **Checks:** recompute_macro_and_weighted_f1
   - **Verdict:** PASS
   - **Notes:** Macro F1 recomputes to 0.46 exactly (within floating error). Weighted F1 recomputes to 0.4832 vs reported 0.48, consistent with 2-decimal rounding.
9. ✔ **C9** (Page 8, Table 1 (macro/weighted F1 from per-class + support))
   - **Claim:** In Table 1, per-class F1 are 0.59 (class 0) and 0.66 (class 1); macro avg F1 is 0.62; weighted avg F1 is 0.63 given supports 51,259 and 58,607.
   - **Checks:** recompute_macro_and_weighted_f1
   - **Verdict:** PASS
   - **Notes:** Macro F1 recomputes to 0.625 (rounds to 0.62). Weighted F1 recomputes to 0.6273 (rounds to 0.63).
10. ✖ **C10** (Page 10, Table 2 (weighted recall consistency))
    - **Claim:** Table 2 shows recall 0.15 (class 0), 0.90 (class 1), weighted avg recall 0.48, supports 1,674 and 2,049.
    - **Checks:** recompute_weighted_average
    - **Verdict:** FAIL
    - **Notes:** Support-weighted recall recomputes to 0.5628, not 0.48.
11. ✔ **C11** (Page 8, Table 1 (weighted recall and precision consistency))
    - **Claim:** Table 1 shows precision 0.60/0.65 with weighted avg 0.63 and recall 0.58/0.67 with weighted avg 0.63; supports 51,259 and 58,607.
    - **Checks:** recompute_weighted_averages
    - **Verdict:** PASS
    - **Notes:** Weighted precision recomputes to 0.6267 vs 0.63; weighted recall recomputes to 0.6280 vs 0.63; both consistent with 2-decimal rounding.
12. ✔ **C12** (Page 8, Section 3.2 (accuracy reported as 63%))
    - **Claim:** Baseline model achieved overall accuracy of 63% (Overall Accuracy: 0.63 in Table 1).
    - **Checks:** percent_decimal_consistency
    - **Verdict:** PASS
    - **Notes:** 0.63 equals 63%/100 (exact).
13. ✔ **C13** (Page 10, Table 2 vs Section 3.3 narrative)
    - **Claim:** GNN achieved overall accuracy 56.2% and Table 2 lists Overall Accuracy: 0.562.
    - **Checks:** percent_decimal_consistency

- **Verdict:** PASS
- **Notes:** 0.562 equals 56.2%/100 (exact).

14. ✔ **C14** (Page 9-10, Section 3.3 + Table 2 (AUC rounding))
    - **Claim:** AUC is reported as **0.5924** in text and **0.59** in the ROC figure caption.
    - **Checks:** rounding_consistency
    - **Verdict:** PASS
    - **Notes:** round(**0.5924**, **2**) = **0.59**, consistent.
15. ✔ **C15** (Page 8, Table 1 vs Figure 5 caption text)
    - **Claim:** Figure 5 caption claims **30,000** correct class-0 and **39,114** correct class-1 on validation; Table 1 supports are **51,259** and **58,607**, total **109,866**.
    - **Checks:** implied_accuracy_from_correct_counts
    - **Verdict:** PASS
    - **Notes:** Implied accuracy from caption counts = **(30,000 + 39,114)/109,866 = 0.6291** vs reported **0.63**; consistent given rounding in '**30,000**'.
16. ✔ **C16** (Page 10, Figure 9 (GNN confusion matrix numbers) vs Table 2 supports/recalls)
    - **Claim:** Figure 9 confusion matrix shows **243** (true0 pred0), **1,431** (true0 pred1), **201** (true1 pred0), **1,848** (true1 pred1); Table **2** supports are **1,674** and **2,049** and recalls are **0.15** and **0.90**; overall accuracy **0.562**.
    - **Checks:** confusion_matrix_consistency
    - **Verdict:** PASS
    - **Notes:** Row sums match supports exactly (**1,674** and **2,049**). Accuracy from CM = **0.5616** vs **0.562** (within tolerance). Recall0 from CM = **0.1452** vs **0.15** (abs diff **0.00484**); Recall1 from CM = **0.9019** vs **0.90** (within tolerance).
17. ✔ **C17** (Page 6, Section 3.1 (node feature means/stds))
    - **Claim:** Node feature means are $[\mathbf{11.13}, \mathbf{0.74}, \mathbf{2.11}, \mathbf{0.38}]$ and stds are $[\mathbf{0.70}, \mathbf{0.36}, \mathbf{0.21}, \mathbf{0.18}]$ for $\log_{10}$(mass), $\log_{10}$(concentration), $\log_{10}(V_{\max})$, scale factor.
    - **Checks:** vector_length_consistency
    - **Verdict:** PASS
    - **Notes:** Means length = **4** and stds length = **4**, consistent with four node features.
18. ✔ **C18** (Page 6, Section 3.1 (graph feature means/stds))
    - **Claim:** Graph-level cosmological parameter means are $[\mathbf{0.30}, \mathbf{0.80}]$ and stds are $[\mathbf{0.12}, \mathbf{0.11}]$ for Ω_m and σ_8.
    - **Checks:** vector_length_consistency
    - **Verdict:** PASS
    - **Notes:** Means length = **2** and stds length = **2**, consistent with two graph-level features.
19. ✔ **C19** (Page 9, Section 3.3 (input feature dimension arithmetic))

- **Claim:** Each node's input features: four intrinsic properties + two global cosmological parameters = **6**-dimensional vector.
- **Checks:** dimension_arithmetic
- **Verdict:** PASS
- **Notes:** $4 + 2 = 6$ (exact).

20. ✔ **C20** (Page 12, Table 3 (Figure 14 rounded values vs Table 3 exact values))
    - **Claim:** Table **3** reports F1 (Cls **1**): reference **0.6937**, no cosmo **0.6317**, no contrastive **0.7106**; Figure 14 caption/labels show $\sim$ **0.694**, **0.632**, **0.711**.
    - **Checks:** rounding_consistency
    - **Verdict:** PASS
    - **Notes:** All three figure values match rounding of table values to **3** decimals (**0.6937** $\rightarrow$ **0.694**; **0.6317** $\rightarrow$ **0.632**; **0.7106** $\rightarrow$ **0.711**).

Limitations

- Audit is limited to the provided parsed-text content of the PDF; numeric values embedded only in plot/table images are not extracted or used for candidates (per instruction to avoid plot-pixel extraction).
- Many ML metrics (precision/recall/F1) cannot be fully recomputed without full confusion matrices or raw prediction counts; only macro/weighted-average arithmetic checks from reported values are proposed.
- Several statements are qualitative or depend on external data (CAMELS-SAM contents, training code outputs) and cannot be verified from the PDF alone.
- Full verification would require underlying confusion matrix counts (TP/FP/FN per class) for the validation subset; the PDF does not provide exact off-diagonal counts in text in a machine-readable way (Figure 5 is an image), so we can only recompute weighted/macro from the already-reported per-class metrics and supports.
- Exact verification of precision/F1 from counts would require full confusion matrix numbers; Figure 9 is an image and not to be read via pixel extraction. We can still cross-check consistency if counts are provided elsewhere textually; only partial checks are feasible.
- Requires access to the underlying standardized data arrays; only approximate means/stds are given pre-standardization and no post-standardization summary statistics are provided numerically.