Skepthical review: *QITT-Enhanced Multi-Scale Substructure Analysis with Learned Topological Embeddings for Cosmological Parameter Estimation*

Summary

This paper proposes an end-to-end pipeline to estimate cosmological parameters (Ω_{m}, σ_8) from dark-matter halo merger trees by (i) extracting "significant" progenitor-branch substructures (Sec. 2.2), (ii) describing each substructure using engineered physical features plus a learned 64D GraphSAGE autoencoder embedding (Sec. 2.3), (iii) padding/truncating to a fixed $(60, 74)$ per-tree tensor (Sec. 2.4), and (iv) compressing this tensor via a Tensor-Train/QITT-style decomposition into a compact 202D feature vector used by standard regressors (Sec. 2.5–2.7). Results on 1000 trees from 40 simulations show QITT features substantially improve over naïve flattened high-dimensional substructure baselines, but a simple linear model on global aggregate features remains strongest for Ω_{m} (Sec. 3.3–3.4). The approach is promising, but several internal inconsistencies (substructure definition, GNN architecture, TT/QITT tensorization) and evaluation gaps (cluster-aware significance testing, stronger compression baselines, padding/truncation sensitivity) currently limit reproducibility and make it hard to isolate what component drives gains.

Strengths

- Clear high-level motivation and an end-to-end pipeline that combines substructure extraction, learned graph embeddings, and TT-based compression in a coherent way (Sec. 2–3).
- Good experimental hygiene in splitting at the simulation level to reduce leakage risk across cosmologies (Sec. 2.1.1–2.1.2).
- Novel and potentially useful application of TT/QITT-style compression to structured merger-tree substructure tensors, yielding a compact fixed-length representation (Sec. 2.5, Sec. 3.2).
- Useful qualitative evidence that GraphSAGE embeddings correlate with meaningful structural properties (e.g., subhalo size distributions) (Sec. 3.1).
- Broad baseline coverage compared to many papers in this area (aggregate features, raw substructure tensors, flattened features, multiple regressors), and an appropriately candid discussion that Ω_{m} is already well predicted by simple global aggregates (Sec. 3.3–3.4).
- Reporting of RMSE/R^2 and an attempt at statistical testing of differences between methods (Sec. 2.8, Sec. 3.3.5).

Major issues

1. **The TT/QITT tensorization and decomposition description is internally inconsistent between Methods and Results, making the core contribution difficult to reproduce and interpret (Sec. 2.5.1 vs. Sec. 3.2).** Sec. 2.5.1 describes flattening $\mathbf{60 \times 74 = 4440}$ then reshaping into a 6-mode tensor $\mathbf{(2, 2, 2, 3, 5, 37)}$ (i.e., quantized-style tensorization), while Sec. 3.2 describes reshaping $\mathbf{(60, 74)}$ to a 3rd-order tensor $\mathbf{(60, 2, 37)}$ and applying a 3-core TT with ranks $\mathbf{(1, 2, 2, 1)}$ to obtain 202 features. These are not equivalent constructions and generally yield different cores/features; the reported 202D arithmetic matches the $\mathbf{(60, 2, 37)}$ pathway, not the 6-mode one. It is also unclear which TT algorithm is used (TT-SVD vs. ALS/optimization), whether decomposition is performed independently per tree, and what is meant by "QITT" vs. standard TT in this implementation.

   Recommendation: Unify Sec. 2.5.1–2.5.2 and Sec. 3.2 around one definitive procedure that matches the reported experiments. Explicitly specify: (i) the exact tensor shape used in all results (e.g., $\mathbf{(60, 2, 37)}$ or $\mathbf{(2, 2, 2, 3, 5, 37)}$), (ii) the number of TT cores, (iii) the TT algorithm/implementation (e.g., TT-SVD with a fixed rank tuple; library and version), (iv) whether TT is computed per-sample independently, and (v) why this is termed QITT (if it is not quantized TT in the standard sense, consider renaming to TT). Add a short pseudocode/algorithm box and ensure the stated 202D feature construction follows from the finalized pipeline.

2. **Substructure identification ("significant substructures") is under-specified and inconsistent across sections, and truncation to $\max_{N_{\rm sub}} = 60$ may dominate what information is retained (Sec. 2.2.1 vs. Sec. 3.1; Sec. 2.4.2).** Sec. 2.2.1 refers to selecting the top 10% mass ratios (and mentions concentration/$V_{\rm max}$ deviations) while Sec. 3.1 reports an adaptive 20th percentile threshold on $\log_{10}(M_{\rm sub}/M_{\rm main})$. The treatment of overlapping/nested branches, multiple triggers, and the exact ordering of substructures before truncation are unclear. Since $N_{\rm sub}$ ranges from 2 to 563 (Sec. 3.1), truncation without a clearly defined ordering/selection rule could discard most substructures in complex trees and bias the representation toward specific epochs or merger ratios.

   Recommendation: In Sec. 2.2.1 provide a single, precise algorithmic definition used for all experiments: define the mass-ratio statistic and its percentile direction/value (10% vs 20th percentile), specify numeric thresholds and computation for concentration/$V_{\rm max}$ deviations (or remove if unused), define how branches are segmented, and how overlaps/nesting are resolved. In Sec. 2.4.2 explicitly define the truncation policy (which 60 are kept and in what order—e.g., highest mass ratio, earliest, latest, longest-lived). Add a sensitivity study varying (a) the percentile threshold (e.g., 10/20/30%) and (b) $\mathbf{max}_{N_{\rm sub}}$ (e.g., 20/40/60/100), reporting both substructure-count statistics and $\Omega_{\rm m}/\sigma_8$ performance.

3. **Padding with a "null substructure" produced by running the pretrained GraphSAGE encoder on a single-node graph with average features may introduce a non-neutral, systematic signature correlated with the amount of padding (Sec. 2.4.2, Sec. 3.1).** This can inadvertently encode (via the fraction of padded slots) a proxy for substructure count/tree complexity, and TT compression may exploit this in ways that are hard to interpret as representing physical content rather than missingness.

   Recommendation: Ablate padding strategies in Sec. 2.4.2 / Sec. 3.3: compare (i) all-zero 74D padding, (ii) the current single-node-graph embedding padding, (iii) a learned padding token (trained downstream), and/or (iv) an explicit mask feature per substructure (append an is_pad bit or provide a mask to the model). Quantify whether prediction error correlates with padding fraction, and report the effect on performance—especially for trees with $N_{\text{sub}} \ll 60$.

4. **GraphSAGE autoencoder architecture/training details are inconsistent and incomplete, and potential representation-learning leakage is not ruled out (Sec. 2.3.2 vs. Sec. 3.1).** The paper alternates between three GraphSAGE layers and two SAGEConv layers, and does not fully specify hidden sizes, activations, decoder, loss definition (node-feature reconstruction only vs. adjacency/topology), training schedule, validation/early stopping, or regularization. The text also alternates between "a large corpus of generated graphs" and "33,759 substructures from the training set," leaving ambiguity about what data were used and whether any substructures from validation/test simulations were included (transductive leakage).

   Recommendation: Standardize the GNN description between Sec. 2.3.2 and Sec. 3.1 and add a complete specification: encoder/decoder architecture (layer types, widths), embedding dimension, pooling, losses and targets (what is reconstructed), optimizer, LR, batch size, epochs, early stopping/validation, regularization. State explicitly that pretraining uses substructures from training simulations only (or clearly justify otherwise) and that embeddings are frozen for downstream regression unless fine-tuning is performed (in which case describe the protocol). Include a small ablation on embedding dimension and/or a non-topological control to verify the embedding adds information beyond smoothed node attributes.

5. **Baseline set and ablations do not yet isolate the value added by (i) TT/QITT compression vs. generic regularization/compression, and (ii) learned topology vs. physical features; some announced baselines (e.g., graphlet counts) are under-specified or not fully reported (Sec. 2.7.1, Sec. 3.3).** The key comparison "flattened 4440D $\rightarrow$ Linear Regression" is known to fail in high dimension with limited samples; stronger baselines like Ridge/ElasticNet and PCA/PLS are needed to demonstrate TT-specific benefit. Additionally, there is no clear 'physical-only QITT' variant to quantify what the GraphSAGE embeddings contribute once TT is applied.

Recommendation: Expand Sec. 3.3 with targeted ablations/baselines: (i) QITT/TT on physical-only tensors $((60, N_{\text{phys}}) \to \mathbf{TT} \to \mathbf{features})$ vs topology-only vs combined; (ii) Ridge/ElasticNet on flattened 4440D features with CV; (iii) PCA (or PLS) to 202D followed by the same regressors as QITT; (iv) concatenate global aggregate features with QITT features to test complementarity; (v) fully specify and report graphlet baseline metrics (Sec. 2.7.1 and Sec. 3.3) or remove it if not executed. This will make claims about QITT and learned topology much more defensible.

6. **Statistical significance testing likely overstates evidence because it treats the 150 test trees as iid, despite clustering by simulation (6 test simulations $\times$ 25 trees) (Sec. 2.8, Sec. 3.3.5).** Paired t-tests on per-tree squared errors can be pseudo-replication if errors are correlated within a simulation, inflating apparent significance. This is particularly important for claims that QITT_XGBoost significantly outperforms other baselines.

   Recommendation: Redo significance testing with simulation-aware blocks: e.g., compute per-simulation mean error ($n = 6$ paired points) and run paired tests on those, or use a clustered bootstrap/permutation test that resamples at the simulation level. Report both per-tree and per-simulation aggregated performance (mean±std across simulations) to assess robustness.

7. **Core feature dimensionalities are inconsistent, which propagates into tensor sizes and TT setup (Sec. 2.3.1, Sec. 3.1, Sec. 2.4.1).** Sec. 2.3.1 claims a 10D physical feature vector, but the enumerated components can be read as 12D (mass ratio, merger scale factor, two property differences, and mean+std over four properties). Sec. 3.1 additionally reports 'num_halos_in_branch', creating further ambiguity about whether this is an input feature or only descriptive. Since $\mathbf{74 = 10 + 64}$ and $\mathbf{4440 = 60 \times 74}$ are used throughout, this needs to be exact.

   Recommendation: In Sec. 2.3.1 provide an explicit ordered list of the physical features actually used as model inputs, with an unambiguous count, and clarify whether 'num_halos_in_branch' (Sec. 3.1) is included in the tensor or only for reporting. Update all dependent dimensionality statements in Sec. 2.4–2.5 and all related figures/tables to match the true input dimension.

Minor issues

1. Model-selection protocol (TT rank selection and regressor hyperparameters) is described in a fragmented and sometimes contradictory way across sections (Sec. 2.5.2, Sec. 2.6.2, Sec. 2.8, Sec. 3.2, Sec. 3.3). It is unclear whether ranks are tuned once using Ridge and then fixed, or tuned jointly per downstream regressor, and whether CV occurs on train only or on train+validation.

   Recommendation: Consolidate the full experimental protocol into one place (ideally Sec. 2.6.2) and cross-reference it elsewhere. Specify: simulation-level split; what CV is run on which subset; the objective for rank selection; when ranks are frozen; hyperpa-

rameter search spaces for RF/XGBoost; whether Ω_m and σ_8 are tuned jointly or separately; and confirm test simulations are untouched until final reporting. Add a table of final chosen ranks and regressor hyperparameters.

2. Discussion of TT-core magnitude ranges risks over-interpretation because TT representations are not unique and permit rescaling ("gauge freedom") across cores (Sec. 3.2). Comparing raw core magnitudes can be misleading without a normalization convention.

   Recommendation: Either remove/soften claims based on core magnitude comparisons, or adopt an invariant diagnostic (e.g., singular values from TT-SVD steps, normalized cores under a specified gauge) and explicitly state the normalization used before interpreting core scales.

3. Claims and framing occasionally overreach relative to results, especially for Ω_m where global aggregate features perform best and QITT does not clearly exceed them (Sec. 3.4, Sec. 4). The "unlocking predictive power" narrative can be read as outperforming simpler approaches, which is not consistently supported.

   Recommendation: Recalibrate Sec. 3.4 and Sec. 4 to emphasize what is clearly demonstrated: TT/QITT provides compact summaries of substructure tensors and beats naïve high-dimensional substructure baselines; global aggregates remain very strong for Ω_m; σ_8 remains challenging. Where improvements are claimed, quantify them against the strongest aggregate baselines and (after updating) cluster-aware statistical tests.

4. Figures and captions often omit essential methodological details needed for interpretation and reproduction (e.g., Figure 1 directionality/time; Figure 2 t-SNE parameters and whether only training points are shown; definitions of feature-importance metrics; mapping from feature IDs to physical meaning) (Sec. 3.1–3.3; Figures 1–11).

   Recommendation: Augment figure captions with key parameters/definitions: add time/flow direction and consistent colorbar limits for Figure 1; report t-SNE hyperparameters and show validation/test embeddings or a quantitative embedding-quality check for Figure 2; define importance metrics and target variables in Figures 3–11; and provide a supplementary table mapping feature IDs (including QITT indices) back to interpretable components.

5. Graphlet-count baseline is introduced but not fully specified and/or not fully reported in Results (Sec. 2.7.1 vs. Sec. 3.3).

   Recommendation: Either fully specify and report it (graphlets counted, normalization, dimensionality, model and metrics) with a consistent label across Sec. 2.7.1 and Sec. 3.3, or explicitly mark it as future work and remove it from baseline-comparison claims.

6. Computational cost/scalability and generalization beyond the limited simulation grid are only briefly discussed (Sec. 1–4).

   Recommendation: Add a short paragraph (Sec. 4 or end of Sec. 1) with approximate compute costs (GNN pretraining time, TT feature extraction time per tree) and discuss expected scaling to larger suites, and limitations of training on a 40-simulation Ω_{m}–σ_8 grid (including the risk of shortcut learning via mass-function differences).

Very minor issues

1. Typographical/formatting inconsistencies and minor notation errors reduce polish (e.g., broken words, inconsistent section header formatting like "# 2.8.", typos such as "recostruction", inconsistent R^2 notation, and minor symbol mistakes like $\mathbf{max}_{N_{\mathrm{cub}}}$ vs $\mathbf{max}_{N_{\mathrm{sub}}}$) (Intro, Sec. 2.7.1, Sec. 4, References).

   Recommendation: Do a full proofreading/formatting pass: fix broken words and typos, standardize section numbering style, correct minor symbol errors, and use consistent mathematical notation for Ω_{m}, σ_8, and R^2 throughout.

2. Reference list contains inconsistent formatting and possibly incomplete entries (e.g., malformed DOIs/placeholders; inconsistent numbering/bracketing styles) (References).

   Recommendation: Verify all citations against authoritative sources, fill missing DOIs/metadata, and standardize bibliography formatting to the target venue's style.

3. TT notation in Sec. 2.5.1 uses informal "×" without specifying contraction indices, which can confuse readers unfamiliar with TT.

   Recommendation: Add the standard TT elementwise/index definition (or explicitly define "×" as contraction over TT-rank indices) and clearly state the ordering of modes used in the finalized tensorization.

Key statements and references

- ✖ **The dataset used in this work consists of 1000 dark matter halo merger trees originating from 40 distinct cosmological simulations, with 25 trees per simulation and each simulation corresponding to a unique pair of cosmological parameters (Ω_{m}, σ_8), constructed following the merger-tree generation methodology of Jiang \& van den Bosch and based on simulations characterized in Yung et al. and Nguyen et al.**
- *Reference(s):* Jiang and van den Bosch, 2013, Yung et al., 2024, Nguyen et al., 2025
- *Justification:* None of the papers describe a dataset of 1000 merger trees from 40 distinct cosmological simulations (25 per simulation) with unique (Ω_{m}, σ_8) pairs. Yung et al., 2024 presents four simulations (same Planck-like cosmology), not 40 or varied cosmologies. Nguyen et al., 2025 trains on VSMDPL (single cosmology) and optionally GUREFT, using tens to hundreds of thousands of trees, not 1000, and not partitioned

per simulation. The merger trees in Yung et al., 2024 and Nguyen et al., 2025 are built with Rockstar/Consistent-Trees from N-body outputs, not using the Jiang and van den Bosch, 2013 EPS-based generation methodology. Hence the dataset description is not supported.

- ✖ **Each node in the merger trees is represented by a 4-dimensional feature vector comprising $\log_{10}$(mass), $\log_{10}$(concentration), $\log_{10}(V_{\max})$, and scale_factor, defined in accordance with standard merger-tree constructions such as Parkinson et al. and Nguyen et al., and these node features are globally normalized using the FeatureNorm-style procedure advocated by Yang et al. and Skryagin et al. for graph neural network inputs.**
- *Reference(s):* Parkinson et al., 2007, Nguyen et al., 2025, Yang et al., 2021
- *Justification:* Nguyen et al., 2025 represents each halo with a 2D feature vector (mass and concentration) and conditions on redshift/scale factor; it does not include $V_{\max}$, a 4D vector, or $\log_{10}$ transforms. Parkinson et al., 2007 describes an EPS-based merger-tree algorithm but does not define such node feature vectors. Yang et al., 2021 proposes L_2 FeatureNorm for dynamic graphs, yet neither Nguyen et al., 2025 nor Parkinson et al., 2007 reports using this global normalization for merger-tree node features. Therefore, the specific 4D feature set and FeatureNorm-style normalization are not supported by the attached papers.
- ✖ **Significant substructures within each merger tree are defined and extracted by traversing from the main root halo and identifying progenitor branches associated with merger events or substantial changes in intrinsic halo properties, following criteria and practices developed in prior merger-tree and substructure studies such as Rangel et al., Jung et al., and Ángel Chandro-Gómez et al., with each identified substructure then represented as an independent graph as in Robles et al.**
- *Reference(s):* Rangel et al., 2020, Jung et al., 2024, Ángel Chandro-Gómez et al., 2025
- *Justification:* Jung et al., 2024 use main-branch merger-tree histories to match halos across simulations but do not define or extract 'significant substructures' within trees, nor represent branches as independent graphs. Ángel Chandro-Gómez et al., 2025 analyze main-branch mass accretion histories and quantify artefacts (mass-swapping, massive transients), but they likewise do not propose a procedure to extract progenitor branches as substructures or to represent them as independent graphs. Neither paper references or implements methods akin to those attributed to Rangel et al. or Robles et al. Thus the stated methodology is not supported by the attached papers.
- ✖ **For each identified substructure, a 10-dimensional physical feature vector is engineered following feature-engineering principles from Owens et al. and Sen et al., and interpreted in the context of substructure and cosmic-web analyses such as Hunde et al. and Bahe \& Jablonka, including mass ratio at merger, merger scale factor, differences in normalized concentra-**

tion and $V_{\max}$ at merger, and the mean and standard deviation of normalized $\log_{10}$(mass), $\log_{10}$(concentration), $\log_{10}(V_{\max})$, and scale_factor across all halos in the substructure.

- *Reference(s):* Owens et al., 2024, Sen et al., 2025, Bahe and Jablonka, 2025
- *Justification:* Owens et al., 2024 and Sen et al., 2025 discuss general feature-engineering workflows (for grain boundaries and images, respectively) but do not define a 10-D astrophysical vector or the listed merger/halo features. Bahe and Jablonka, 2025 analyze cosmic-web filaments (e.g., widths, lengths, densities, temperature profiles, substructure fractions) but do not construct per-substructure feature vectors or use merger mass ratios, merger scale factors, concentration or $V_{\max}$ statistics across halos. The specific 10-D vector and its components are not supported by the attached papers.

- **✖ Topological information for each substructure is captured using a GraphSAGE-based autoencoder, motivated by prior work on topological regularization and embeddings in graph neural networks (Song et al., Tola et al., Li et al.), where the encoder maps 4-dimensional node features to 64-dimensional node embeddings that are globally mean-pooled to yield a 64-dimensional graph-level topological embedding per substructure, in line with cosmological graph-embedding approaches such as Villanueva-Domingo \& Villaescusa-Navarro and Kvasiuk et al.**

- *Reference(s):* Song et al., 2021, Tola et al., 2024, Villanueva-Domingo and Villaescusa-Navarro, 2023
- *Justification:* The papers do not describe a GraphSAGE-based autoencoder that maps 4-D node features to 64-D embeddings and mean-pools them into a 64-D graph-level topological embedding. Song et al., 2021 introduces topological regularization using Node2Vec features and dual GNNs (not a GraphSAGE autoencoder). Tola et al., 2024 (TopER) provides 2-D topology-inspired graph summaries via filtrations and regression, not GNN embeddings. Villanueva-Domingo and Villaescusa-Navarro, 2023 employs message-passing GNNs on galaxy graphs (with optional 4 node features) and multi-pooling aggregation to predict cosmological quantities, but not GraphSAGE, not an autoencoder, and not a topological embedding. References to Li et al. and Kvasiuk et al. are not present in the attached papers.

- **✔ The Quantum-Inspired Tensor Train (QITT) step applies Tensor Train decomposition, as formulated in tensor-network literature (Diniz; Phan et al.; Chen et al.; Wang et al.; Matsuura et al.; Sander et al.), to the reshaped substructure feature tensor, with TT-ranks treated as hyperparameters and optimized via cross-validation, yielding TT-cores that are flattened and concatenated into a 202-dimensional feature vector per merger tree that serves as a compressed representation for downstream regression models.**

- *Reference(s):* Diniz, 2021, Phan et al., 2016, Wang et al., 2025

- *Justification:* Verification failed with gpt-5: Error code: 400 - {'error': {'message': 'Your input exceeds the context window of this model. Please adjust your input and try again.', 'type': 'invalid_request_error', 'param': 'input', 'code': 'context_length_exceeded'}}

Mathematical consistency audit

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

Maths relevance: substantial

The paper's core analytic structure is a feature-dimension pipeline that constructs a fixed $(60, 74)$ tensor per tree and compresses it via TT/QITT into a 202-dimensional vector. The audit focused on dimensional consistency (feature counts, tensor reshapes, TT-ranks/core sizes) and definition consistency (substructure thresholds, padding/truncation). While several arithmetic checks pass (4440 and 202 computations), the main mathematical inconsistency is a contradiction between the Methods' 6-mode TT/QITT reshape and the Results' 3-mode reshape, as well as an internal mismatch in the claimed 10-dimensional physical feature vector.

Checked items

1. ✔ **Node feature normalization formula** (Sec. 2.1.1, p.3)
    - **Claim:** Each node feature is normalized as $x_{\text{normalized}} = (x - \mu)/\sigma$ using global mean and standard deviation from training nodes.
    - **Checks:** algebra, definition consistency
    - **Verdict:** PASS; confidence: high; impact: minor
    - **Assumptions/inputs:** $\sigma \neq 0$ for each feature, μ, σ computed per-feature over training nodes
    - **Notes:** Formula is standard and internally consistent with the described preprocessing.
2. ⚠ **Mass accretion / mass ratio definition** (Sec. 2.2.1, p.3)
    - **Claim:** Relative mass accretion rate is quantified as $\log_{10}(M_{\text{progenitor}}/M_{\text{descendant}})$.
    - **Checks:** notation consistency, sanity/limiting case
    - **Verdict:** UNCERTAIN; confidence: medium; impact: minor
    - **Assumptions/inputs:** Masses are positive, The ratio refers to progenitor vs descendant halo mass at the merger interface
    - **Notes:** Mathematically well-formed, but later Results use different symbols ($M_{\text{sub,progenitor}}/M_{\text{main,progenitor}}$) and a different percentile rule; the paper does not fully reconcile which ratio is actually used.
3. ✖ **Physical feature vector dimensionality** (Sec. 2.3.1, p.3–4)

- **Claim:** A 10-dimensional physical feature vector is engineered for each substructure, comprising the listed items.
- **Checks:** dimension counting, definition consistency
- **Verdict:** FAIL; confidence: high; impact: critical
- **Assumptions/inputs:** Each bullet corresponds to one or more scalar features exactly as counted
- **Notes:** Counting the described components gives 12 features: mass ratio (1) + merger scale factor (1) + differences in concentration and $V_{\max}$ (2) + mean and std for 4 properties (8) = 12, not 10. This propagates to the claimed 74-dimensional concatenated substructure vector and 4440 total features.

4. ✖ **Results mention of additional physical feature** (Sec. 3.1, p.6)
   - **Claim:** Physical features include 'num_halos_in_branch' with reported mean/std.
   - **Checks:** definition consistency, dimension counting
   - **Verdict:** FAIL; confidence: high; impact: moderate
   - **Assumptions/inputs:** The described feature is part of the 10-dimensional vector
   - **Notes:** Introducing 'num_halos_in_branch' as a physical feature contradicts the earlier fixed 10-feature claim unless another feature is removed. No reconciled list is given.

5. ⚠ **Concatenated substructure feature dimension (physical+topological)** (Sec. 2.4.1, p.4)
   - **Claim:** Concatenating 10 physical and 64 topological features yields 74 features per substructure.
   - **Checks:** dimension counting, dependency check
   - **Verdict:** UNCERTAIN; confidence: high; impact: critical
   - **Assumptions/inputs:** Physical vector truly has length 10, Topological embedding truly has length 64
   - **Notes:** The $\mathbf{10 + 64 = 74}$ arithmetic is correct, but the physical feature length is inconsistent elsewhere (appears to be 12+). Therefore the claimed 74 is not verifiable.

6. ⚠ **Fixed tensor shape and total feature count** (Sec. 2.4.2, p.4; also Abstract p.1)
   - **Claim:** Setting $\mathbf{max}\, N_{\text{sub}} = \mathbf{60}$ yields a fixed tensor $\mathbf{(60, 74)}$ with 4440 features $\mathbf{(60 \times 74)}$.
   - **Checks:** dimension counting
   - **Verdict:** UNCERTAIN; confidence: high; impact: critical
   - **Assumptions/inputs:** Per-substructure vector length is 74, Exactly 60 substructures are kept after padding/truncation

- **Notes:** $60 \times 74 = 4440$ is correct, but depends on the disputed 74-dimensional substructure vector and on an under-specified truncation policy.

7. ✔ **Prime-factor reshape of 4440 into 6-mode tensor** (Sec. 2.5.1, p.5)
   - **Claim:** The 4440 features are reshaped into a 6-mode tensor of dimensions $(2, 2, 2, 3, 5, 37)$ corresponding to prime factors of 4440.
   - **Checks:** arithmetic, dimension consistency
   - **Verdict:** PASS; confidence: high; impact: moderate
   - **Assumptions/inputs:** 4440 is exactly factorized as $2 \times 2 \times 2 \times 3 \times 5 \times 37$
   - **Notes:** Multiplication yields 4440, and the factor list matches the stated intent.

8. ⚠ **TT decomposition expression and core count** (Sec. 2.5.1, p.5)
   - **Claim:** After reshaping to 6 modes, the tensor is approximated by a product of TT-cores: $T \approx G_1 \times G_2 \times \cdots \times G_D$.
   - **Checks:** notation consistency, structural consistency
   - **Verdict:** UNCERTAIN; confidence: medium; impact: moderate
   - **Assumptions/inputs:** D equals the tensor order (implied 6 here), $\times$ indicates appropriate TT contractions
   - **Notes:** The expression is not incorrect but is under-defined (no contraction/index notation). More importantly, later Results switch to a 3-mode TT, creating an internal inconsistency about D and the actual decomposition performed.

9. ✔ **Results reshape of 74 into (2,37) and 3-mode tensor** (Sec. 3.2, p.7)
   - **Claim:** Before decomposition, the $(60, 74)$ tensor is reshaped into a 3rd-order tensor $(60, 2, 37)$.
   - **Checks:** arithmetic, dimension consistency
   - **Verdict:** PASS; confidence: high; impact: moderate
   - **Assumptions/inputs:** $74 = 2 \times 37$
   - **Notes:** $74 = 2 \times 37$ and $60 \times 74 = 60 \times 2 \times 37$, so the reshape is dimensionally consistent.

10. ✖ **Contradiction between 6-mode Methods and 3-mode Results reshapes** (Sec. 2.5.1 (p.5) vs Sec. 3.2 (p.7))
    - **Claim:** The paper describes a single QITT pipeline but uses incompatible tensor orders (6-mode vs 3-mode) and correspondingly different TT-core counts.
    - **Checks:** definition consistency, pipeline consistency
    - **Verdict:** FAIL; confidence: high; impact: critical
    - **Assumptions/inputs:** Methods and Results refer to the same experiment/pipeline producing the reported 202 features

- **Notes:** Methods: flatten 4440 and reshape to $(2, 2, 2, 3, 5, 37)$ (6 modes) $\rightarrow$ implies 6 TT cores. Results: reshape $(60, 74)$ to $(60, 2, 37)$ (3 modes) $\rightarrow$ 3 TT cores. The reported 202 feature length is derived using the 3-core setup, not the 6-core setup, so the central mathematical pipeline is inconsistent.

11. ✔ **TT ranks and core shapes for 3-mode tensor** (Sec. 3.2, p.7; Fig. 3 caption p.8)
    - **Claim:** For tensor dimensions $(60, 2, 37)$ and ranks $(1, 2, 2, 1)$, the TT-core shapes are $(1, 60, 2)$, $(2, 2, 2)$, $(2, 37, 1)$.
    - **Checks:** dimension consistency
    - **Verdict:** PASS; confidence: high; impact: moderate
    - **Assumptions/inputs:** Standard TT core shape convention: (r_{k-1}, n_k, r_k)
    - **Notes:** Shapes match the stated ranks and mode sizes.

12. ✔ **202-dimensional QITT feature length calculation** (Sec. 3.2, p.7)
    - **Claim:** Flattening and concatenating the 3 TT-cores yields 202 features: $1 \times 60 \times 2 + 2 \times 2 \times 2 + 2 \times 37 \times 1 = 202$.
    - **Checks:** arithmetic, dimension counting
    - **Verdict:** PASS; confidence: high; impact: moderate
    - **Assumptions/inputs:** All core entries are included in the feature vector without further reduction
    - **Notes:** Arithmetic is correct: $120 + 8 + 74 = 202$.

13. ✔ **Simulation-level split arithmetic** (Sec. 2.1.2, p.3)
    - **Claim:** 40 simulations split into 28/6/6 simulations yields 700/150/150 trees given 25 trees per simulation.
    - **Checks:** arithmetic consistency
    - **Verdict:** PASS; confidence: high; impact: minor
    - **Assumptions/inputs:** Exactly 25 trees per simulation
    - **Notes:** $28 \times 25 = 700$, $6 \times 25 = 150$, totals match 1000.

14. ⚠ **Padding/truncation completeness relative to observed N_{sub} range** (Sec. 2.4.2, p.4 vs Sec. 3.1, p.6–7)
    - **Claim:** A fixed length of 60 substructures is used despite some trees having up to 563 substructures.
    - **Checks:** definition completeness, consistency
    - **Verdict:** UNCERTAIN; confidence: high; impact: moderate
    - **Assumptions/inputs:** Trees with $N_{\text{sub}} > 60$ are truncated deterministically or by a defined rule

- **Notes:** Results explicitly mention truncation but Methods do not define it. Without a truncation rule, the ordering/selection of substructures in the tensor is not mathematically specified, which affects the reproducibility of the TT input tensor.

Limitations

- The provided PDF text contains very few explicit, step-by-step derivations; many key operations (TT decomposition, rank tuning, feature ordering) are described procedurally without formal mathematical definitions, limiting verifiability beyond dimension checks.
- No equation numbering is present in the extracted text; locations are referenced by section and page only.
- Several central quantities (exact list of physical features, exact substructure selection/truncation ordering) are not fully specified, preventing complete internal verification of downstream tensor dimensions and the QITT feature construction.

Numerical results audit

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

Most internal arithmetic relationships (counts, percentages, dimensionality products, reshapes, and rounding of reported performance metrics) are consistent. Two internal consistency problems were detected: (i) a mismatch in the stated substructure selection threshold (10% vs 20th percentile) across sections, and (ii) an inconsistency in the stated TT core shapes versus the standard TT shape definition given the stated modes and ranks.

Checked items

1. ✔ **C01_dataset_simulation_counts** (Sec. 2.1, p.2-3)
    - **Claim:** Dataset comprises 1000 merger trees from 40 simulations, with 25 trees generated per simulation.
    - **Checks:** product_equals_total
    - **Verdict:** PASS
    - **Notes:** 40×25 computed as 1000; matches stated total.
2. ✔ **C02_split_simulation_level_counts** (Sec. 2.1.2 Data Splitting, p.3)
    - **Claim:** Out of 40 simulations: 28 train (700 trees), 6 val (150 trees), 6 test (150 trees).
    - **Checks:** parts_sum_to_total_and_implied_multiples
    - **Verdict:** PASS
    - **Notes:** $28 + 6 + 6 = 40$; $28 \times 25 = 700$; $6 \times 25 = 150$ for val and test; $700 + 150 + 150 = 1000$.

3. ✔ **C03_split_percentages_match_counts** (Sec. 2.1.2 Data Splitting, p.3)
    - **Claim:** A 70-15-15 split is used on 1000 trees, yielding 700/150/150 trees.
    - **Checks:** percent_to_count_consistency
    - **Verdict:** PASS
    - **Notes:** $0.70 \times 1000 = 700$; $0.15 \times 1000 = 150$ (val/test); $70 + 15 + 15 = 100$.
4. ✔ **C04_substructure_feature_dim_addition** (Sec. 2.4.1, p.4; also Abstract p.1)
    - **Claim:** 10-dimensional physical features concatenated with 64-dimensional topological embedding gives 74-dimensional combined feature vector.
    - **Checks:** dimension_addition
    - **Verdict:** PASS
    - **Notes:** $10 + 64 = 74$.
5. ✔ **C05_max_substructures_from_feature_count** (Sec. 2.4.2, p.4)
    - **Claim:** Maximum number of substructures $\max N_{\text{sub}}$ set to 60 as indicated by 4440 features $= 60 \times 74$.
    - **Checks:** product_equals_total
    - **Verdict:** PASS
    - **Notes:** $60 \times 74 = 4440$.
6. ✔ **C06_tensor_shape_implies_flat_length** (Sec. 2.5.1, p.4-5)
    - **Claim:** A $(60, 74)$ tensor is flattened into a 1D vector of length $60 \times 74 = 4440$.
    - **Checks:** shape_to_size_consistency
    - **Verdict:** PASS
    - **Notes:** $60 \times 74 = 4440$.
7. ✔ **C07_prime_factor_reshape_product** (Sec. 2.5.1, p.5)
    - **Claim:** 4440 features were reshaped into a 6-mode tensor with dimensions $(2, 2, 2, 3, 5, 37)$, reflecting prime factors of 4440.
    - **Checks:** factor_product_equals_total
    - **Verdict:** PASS
    - **Notes:** $2 \times 2 \times 2 \times 3 \times 5 \times 37 = 4440$.
8. ✔ **C08_baseline_B2_dimensionality** (Sec. 2.7.1 Baseline Models, p.5)
    - **Claim:** Raw physical substructure features baseline: $60 \times 10 = 600$-dimensional feature vector per tree.
    - **Checks:** product_equals_total
    - **Verdict:** PASS
    - **Notes:** $60 \times 10 = 600$.

9. ✔ **C09_baseline_B4_dimensionality** (Sec. 2.7.1 Baseline Models, p.5)
    - **Claim:** Flattened combined features baseline: $\mathbf{60 \times 74 = 4440}$-dimensional feature vector per tree.
    - **Checks:** product_equals_total
    - **Verdict:** PASS
    - **Notes:** $60 \times 74 = 4440$.
10. ✔ **C10_training_nodes_normalization_claim_zero_mean_unit_std** (Sec. 3.1, p.6)
    - **Claim:** Node features were normalized to zero mean and unit standard deviation based on global statistics derived from the 700 training trees; example means: $\mathbf{\log_{10}(mass) = 11.14}$, scale_factor= $\mathbf{0.37}$.
    - **Checks:** internal_text_consistency_about_means
    - **Verdict:** PASS
    - **Notes:** Only cross-checked that the stated training-set size (700 trees) matches the split; means/std were not recomputed.
11. ✖ **C11_substructure_threshold_percentile_mismatch** (Sec. 2.2.1, p.3 vs Sec. 3.1, p.6)
    - **Claim:** Substructure significance threshold described as 'top 10% of mass ratios within each tree' (Methods) but later as '20th percentile' (Results).
    - **Checks:** repeated_constant_consistency
    - **Verdict:** FAIL
    - **Notes:** Direct cross-section equality check failed (10 vs 20).
12. ✔ **C12_pretraining_substructures_count_vs_average** (Sec. 3.1, p.6)
    - **Claim:** GNN pre-trained on 33,759 substructures from the training set; average 47.45 substructures per tree across 1000 trees is also reported.
    - **Checks:** order_of_magnitude_cross_check
    - **Verdict:** PASS
    - **Notes:** $\mathbf{47.45 \times 700 = 33,215}$ vs 33,759 reported ($\approx \mathbf{1.6\%}$ relative difference).
13. ✔ **C13_padding_truncation_vs_reported_max_range** (Sec. 3.1, p.6-7; Sec. 2.4.2, p.4)
    - **Claim:** Substructures per tree ranged from 2 to 563, but tensors are padded/truncated to fixed length 60 substructures.
    - **Checks:** range_vs_cap_consistency
    - **Verdict:** PASS
    - **Notes:** Logical implication holds: $\mathbf{563 > 60}$, so truncation must occur for some trees.

14. ✔ **C14_74_reshaped_to_2x37** (Sec. 3.2, p.7)
    - **Claim:** 74-dimensional feature space per substructure reshaped into two factors $(2, 37)$.
    - **Checks:** factor_product_equals_total
    - **Verdict:** PASS
    - **Notes:** $2 \times 37 = 74$.
15. ✔ **C15_tensor_shape_after_reshape_consistency** (Sec. 3.2, p.7)
    - **Claim:** Reshaping transforms original $(60, 74)$ tensor into 3rd-order tensor of shape $(60, 2, 37)$.
    - **Checks:** shape_product_preservation
    - **Verdict:** PASS
    - **Notes:** Element count preserved: $60 \times 74 = 60 \times 2 \times 37 = 4440$.
16. ✔ **C16_TT_rank_tuple_and_core_sizes_to_202** (Sec. 3.2, p.7)
    - **Claim:** With dimensions $(60, 2, 37)$ and TT-ranks $(1, 2, 2, 1)$, flattened core sizes sum to 202: $1 \times 60 \times 2 + 2 \times 2 \times 2 + 2 \times 37 \times 1 = 120 + 8 + 74 = 202$.
    - **Checks:** derived_dimension_from_components
    - **Verdict:** PASS
    - **Notes:** Core element counts 120, 8, and 74 sum to 202 as stated.
17. ✖ **C17_TT_core_shapes_match_ranks_and_modes** (Fig. 3 caption / Sec. 3.2, p.7-8)
    - **Claim:** TT core shapes are Core0 $(1, 60, 2)$, Core1 $(2, 2, 2)$, Core2 $(2, 37, 1)$ given ranks $(1, 2, 2, 1)$ and modes $(60, 2, 37)$.
    - **Checks:** shape_consistency_with_TT_definition
    - **Verdict:** FAIL
    - **Notes:** Computed expected TT core shapes are $(1, 60, 2)$, $(2, 2, 2)$, $(2, 37, 1)$, but the reported core_shapes captured by the check include an extra leading index (e.g., [0,1,60,2]).
18. ✔ **C18_R2_consistency_abstract_vs_results** (Abstract p.1 vs Sec. 3.3.2 p.9 and Conclusions p.12)
    - **Claim:** QITT-based Linear Regression achieves $R^2 \approx 0.923$ for Ω_{m} and ≈ 0.621 for σ_8 (more precisely 0.9231 and 0.6206 in Results).
    - **Checks:** rounded_value_consistency
    - **Verdict:** PASS
    - **Notes:** Results values agree with abstract within 3-decimal rounding tolerance.

19. ✔ **C19_best_baseline_R2_Om_rounding** (Abstract p.1 and Conclusions p.12 vs Sec. 3.3.3 p.9)
    - **Claim:** Aggregate baseline achieves highest R^2 for Ω_{m} reported as 0.970 (Abstract/Conclusions) and 0.9696 (Results).
    - **Checks:** rounded_value_consistency
    - **Verdict:** PASS
    - **Notes:** $\mathbf{0.9696}$ rounds to $\mathbf{0.970}$ at 3 decimals.
20. ✔ **C20_pvalue_threshold_statement_vs_specific_pvalues** (Sec. 2.8 p.6; Sec. 3.3.5 p.10-11)
    - **Claim:** Significance threshold $p < \mathbf{0.05}$; reported p-values include 0.9537, 0.1734 (not significant) and $\mathbf{1.8866 \times 10^{-8}}$, $\mathbf{2.8041 \times 10^{-5}}$, $\mathbf{0.0104}$, $\mathbf{0.0014}$ (significant).
    - **Checks:** threshold_classification_consistency
    - **Verdict:** PASS
    - **Notes:** All listed p-values are correctly classified relative to $\alpha = \mathbf{0.05}$ (two non-significant; four significant).

Limitations

- Only parsed PDF text was available; no underlying datasets (merger trees, extracted substructures, predictions, targets) are provided, preventing recomputation of statistics like means, RMSE, R^2, and t-tests.
- Checks were limited to internal arithmetic/logical consistency that can be verified from explicit numbers stated in the PDF (products, sums, rounding, thresholds, and dimensionality).
- No values were extracted from plotted figures (bar charts/scatter plots) because that would require reading pixel/graphic data rather than explicit numeric statements.