Skepthical review: *QTT-Informed Subgraph Feature Engineering for Merger Tree Regression: A Proof-of-Concept*

Summary

This manuscript proposes a proof-of-concept feature-engineering pipeline for cosmological dark-matter halo merger trees to predict final halo mass at $z = 0$. The approach extracts localized k-hop subgraphs around nodes on the main progenitor branch, forms node-feature matrices (mass, concentration, $v_{\max}$, scale factor), pads/reshapes them to enable Quantized/"Quantum" Tensor Train (QTT/TT) decomposition, and uses the resulting compressed subgraph embeddings (aggregated to a tree-level vector) as inputs to a Random Forest regressor (Sec. 1; Sec. 3.1–3.6). A critical data/implementation issue—invalid main-branch indices in $mask_{\text{main}}$—prevents subgraph extraction for most of the nominal **300** trees, leaving only $N = 5$ usable trees (Sec. 4.2). Consequently, essentially all quantitative results (QTT reconstruction errors, regression metrics, PCA, feature importances; Sec. 4.3–4.6) are in-sample diagnostics on five objects and cannot support performance claims. The paper's value is therefore currently primarily methodological (pipeline outline and encountered pitfalls), but reproducibility and interpretability are limited by under-specified QTT details, unclear evaluation protocol, and several presentation/structure problems (Sec. 2–3; Sec. 4.4; Sec. 5).

Strengths

- Coherent end-to-end concept: k-hop local neighborhoods + tensor-train compression + classical regression provides a potentially lightweight alternative to end-to-end GNNs for merger-tree data (Sec. 1; Sec. 3.2–3.5).
- Transparent acknowledgement that the current experiments are constrained to $N = 5$ due to $mask_{\text{main}}$ issues and are not statistically meaningful, appropriately framing the work as proof-of-concept (Sec. 4.2; Sec. 4.4.3; Sec. 5).
- The pipeline is laid out in a way that could be reusable once data integrity is fixed (preprocessing → subgraph extraction → padding/reshaping → QTT → aggregation → regressor → diagnostics; Sec. 3.1–3.7).
- Including QTT reconstruction MSE across k and ranks is a useful sanity check that the decomposition machinery is at least producing controllable approximations (Sec. 4.3).
- Baseline comparison (simple main-branch summary statistics) is directionally appropriate and should remain as a reference point once a valid evaluation set exists (Sec. 4.4.1).

Major issues

1. **Empirical evaluation is blocked: due to invalid/out-of-bounds $mask_{\text{main}}$ entries, only 5/300 trees yield valid main-branch nodes and k-hop subgraphs, so all reported regression/PCA/importance results are in-sample on $N = 5$ (Sec. 4.2; Sec. 4.3–4.6).** Even with caveats, the volume of tables/figures and specific R^2 comparisons can be misread as evidence of effectiveness.

   Recommendation: Make resolving/working around $\boldsymbol{mask}_{\text{main}}$ the top priority and rerun the full pipeline on a substantially larger, representative sample. Concretely (Sec. 3.2; Sec. 4.2): (i) specify what $\boldsymbol{mask}_{\text{main}}$ is (boolean mask vs integer index list), (ii) add diagnostics per tree (min/max $\boldsymbol{mask}_{\text{main}}$, $\boldsymbol{num}_{\boldsymbol{nodes}}$; count invalid indices; when invalidation occurs), (iii) verify no reindexing/batching step desynchronizes $\boldsymbol{mask}_{\text{main}}$ from $\boldsymbol{data.x/edge_index}$, and (iv) if $\boldsymbol{mask}_{\text{main}}$ cannot be trusted, reconstruct the main branch from the merger-tree topology (algorithmically identify the main progenitor chain) or drop corrupted trees. If a larger valid run is not feasible for this submission, strongly downshift Results: keep only minimal illustrative outputs, remove/appendix most performance/importance/PCA plots, and clearly label remaining numbers as "pipeline output examples on 5 trees," not comparative evidence.

2. **Core methodological ambiguity: the manuscript does not precisely define the QTT/TT object being computed (factorization equation, core shapes, rank definition, algorithm used), nor the exact tensorization (reshape) scheme from a padded $[n_{\text{nodes}}, 4]$ matrix to higher-order modes (Sec. 3.3–3.4).** This prevents reproduction and makes it unclear what structure QTT is intended to capture beyond generic compression.

   Recommendation: Expand Sec. 3.3–3.4 with a complete, implementation-level specification: (i) explicitly define TT/QTT with an equation (cores $\boldsymbol{G}^{(m)}$, ranks $\boldsymbol{r}_m$, mode sizes), (ii) list for each $\boldsymbol{k}$ used in Sec. 4.3–4.4 the padded matrix shape and the exact tensorization (e.g., node dimension quantized into $\mathbf{2} \times \mathbf{2} \times \ldots$ and how the feature-channel dimension $\mathbf{4}$ is handled—kept as a mode vs also quantized), (iii) state padding location/value (zeros? mean? mask-aware?) and whether a padding indicator is added, (iv) name the library/algorithm (TT-SVD, rounding, tolerance vs fixed rank) and rank-selection rule, and (v) define exactly how cores are converted to a fixed-length vector (flatten/concatenate) and report the resulting dimensionality per configuration.

3. **Permutation (node-order) sensitivity is not addressed: the method applies QTT to an ordered node-feature matrix, but subgraphs are graphs and node ordering can be arbitrary. Without a canonical ordering, isomorphic subgraphs can yield different QTT embeddings purely due to indexing, undermining stability and physical interpretability (Sec. 3.2–3.4).**

Recommendation: In Sec. 3.2–3.4, define and justify a canonical node ordering for each extracted subgraph before forming the matrix (e.g., sort by scale factor/time, then graph distance from the center/main-branch node, with deterministic tie-breaking; or BFS/DFS with fixed tie rules). Add a sensitivity check once N is fixed: permute node order within subgraphs and quantify variation in QTT features and downstream predictions. If sensitivity is high, consider permutation-invariant pre-aggregation (e.g., pooling/histogramming) prior to tensorization, and discuss the trade-off with losing fine-grained structure.

4. **Potential target leakage / trivial prediction setup is not ruled out: the target is final halo mass at $z = 0$ while node features include mass; if subgraphs include late-time nodes near (or at) $z = 0$ on the main branch, predicting $M(z = 0)$ may be nearly direct, making reported performance uninterpretable as "learning from history" (Sec. 1; Sec. 3.1; Sec. 3.2; Sec. 4.4).**

   Recommendation: Clarify the prediction task formally (Sec. 1; Sec. 3.2; Sec. 4.4): specify which snapshots/nodes are included for features, whether the $z = 0$ node (and nodes after a cutoff redshift) are excluded, and whether the goal is forecasting from earlier epochs. Add leakage-aware baselines once data is fixed (e.g., "use last available main-branch mass before cutoff" and "linear model on late-time mass only") to demonstrate the task is non-trivial.

5. **Evaluation protocol is unclear/inconsistent: Sec. 3.5 suggests cross-validation and hyperparameter optimization, but Sec. 4.4 states evaluations are in-sample given $N = 5$, with insufficient detail on splits, tuning, seeds, or how QTT/rank/k choices are selected. This undermines reproducibility even as a proof-of-concept (Sec. 3.5; Sec. 4.4).**

   Recommendation: Rewrite Sec. 3.5 and the opening of Sec. 4.4 to unambiguously specify: (i) split unit (by tree/halo), (ii) train/val/test or k-fold protocol, (iii) hyperparameter search space and selection criterion, (iv) random seeds and number of repeats, and (v) which metrics are reported on which split. For the current $N = 5$ run, label everything explicitly as in-sample debugging output; once N is repaired, report only out-of-sample performance with uncertainty (bootstrap or repeated CV).

6. **Subgraph construction is underspecified in graph-theoretic terms: it is unclear whether k-hop neighborhoods treat edges as directed or undirected (merger trees are DAGs), whether neighborhoods include progenitors, descendants, or both, and whether extraction is from the full tree or constrained around the main branch (Sec. 3.2). These choices materially change the included nodes and thus the learned representation.**

   Recommendation: In Sec. 3.2, define the neighborhood operator precisely: directed vs undirected adjacency; whether hops traverse progenitor→descendant, descendant→progenitor, or both; what the center nodes are (which main-branch nodes, at which

times); and whether off-branch nodes are included. Provide minimal pseudocode (or a PyG snippet) showing how subgraphs are built from $\boldsymbol{edge_index}$, including any reindexing to subgraph-local node IDs.

Minor issues

1. Manuscript structure and formatting are internally inconsistent: an empty/unused "2 Methods" section is followed by "3 Methodology" containing the actual methods; Sec. 3.8 shows markdown artifacts (e.g., "# 3.8.2.", stray "####") and placeholder text like "(?)(?)(?)(?)" (Sec. 2; Sec. 3; Sec. 3.8).

   Recommendation: Consolidate into a single coherent Methods/Methodology section with consistent numbering (e.g., Sec. 2.1–2.8), remove empty headings, delete placeholders, and ensure cross-references point to unique sections.

2. Figure/table cross-references are broken or misleading (e.g., "Figures ??, 4, 5, 6, 7, and 8"), and Sec. 4.6 references "the table in Section 3.2" although the performance table appears as Table 1 in Sec. 4.4.2 (Sec. 4.4.2; Sec. 4.6).

   Recommendation: Run a full reference audit: fix unresolved labels (remove "??"), ensure every figure/table is uniquely numbered and cited correctly, and correct the Sec. 4.6 reference to Table 1 / Sec. 4.4.2 (or renumber consistently). Consider consolidating redundant plots, especially given the proof-of-concept status.

3. Preprocessing description is inconsistent about how normalization statistics are computed ("training set" in Sec. 3.1.2 vs "global statistics from the entire dataset" in Sec. 4.2), and handling of log transforms (zeros/negatives) and the final target scaling/units is not fully specified (Sec. 3.1.1–3.1.2; Sec. 4.2).

   Recommendation: Specify: (i) exact feature list used, (ii) transformations applied ($\log_{10}$? natural log?), (iii) how nonpositive values/missingness are handled, (iv) whether $\boldsymbol{\mu, \sigma}$ are computed on train only (preferred) and then applied to val/test, and (v) the target definition/units (raw mass vs $\log$ mass vs standardized). Make Sec. 3.1 and Sec. 4.2 consistent.

4. Baseline feature computation is not fully formalized in Methods and shows a red flag: variance features being identically zero suggests either only one node per tree, constant features, or a bug—this affects the fairness of baseline vs QTT comparisons (Sec. 4.4.1).

   Recommendation: Move baseline definition into Methods (e.g., new subsection near Sec. 3.4–3.5): define exactly which nodes are included (entire main branch? truncated?), feature ordering, and missing-node handling. Add a brief debug note explaining any zero-variance behavior and verify baseline extraction once $\boldsymbol{mask}_{\text{main}}$ is corrected.

5. Aggregation from subgraph embeddings to a tree-level vector is described abstractly, but it is not always explicit which aggregation is used in each reported configuration, and how aggregation behaves when each tree yields only one valid subgraph under the current failure mode (Sec. 3.4; Sec. 4.3–4.4).

   Recommendation: At the start of Sec. 4.3 (and in Sec. 3.4), state exactly which aggregation operator is used per experiment (mean/max/concat), and report the resulting tree-level feature dimensionality. Note explicitly when aggregation is effectively identity because only one subgraph exists.

6. Interpretation-heavy analyses (feature importances, PCA variance explained) are presented in the main Results despite $N = 5$; impurity-based Random Forest importances are especially unstable and the mapping from QTT feature indices back to interpretable modes/cores is not provided (Sec. 4.5–4.6).

   Recommendation: For the current draft, reframe these as "example pipeline diagnostics" or move to an appendix. Once N is fixed, switch to permutation importance/SHAP with stability across folds and provide an index-to-$(core, mode, original\ variable)$ mapping so importances can be interpreted physically.

7. Terminology/positioning: the manuscript alternates between "Quantum Tensor Train" and the standard numerical-linear-algebra term "Quantized Tensor Train"; novelty relative to PCA/autoencoders/GNNs is not sharply articulated (Sec. 1; Sec. 5).

   Recommendation: Define terminology once (prefer "Quantized Tensor Train (QTT)" unless genuinely quantum), cite core TT/QTT references, and add a short "Why QTT here?" paragraph (Sec. 1 or Sec. 5) clarifying expected advantages/limitations vs PCA, autoencoders, and GNNs.

8. Figures frequently lack precise axis labels/units/transformations (e.g., "Final Halo Mass Property"), and styling/readability is uneven across many plots (Sec. 4.3–4.6).

   Recommendation: Standardize: label targets/features with units and any log/standardization; use consistent parameter notation (k, rank); increase readability (font sizes, top-k bars, better layouts); and ensure captions state evaluation protocol (in-sample vs out-of-sample).

9. Reproducibility metadata is missing: code availability, data access constraints, and compute requirements are not stated (Sec. 1–5).

   Recommendation: Add a brief "Data and Code Availability" note (end of Methods or Sec. 5): dataset source/licensing, what will be released (scripts, configs), and approximate runtime/resources for QTT + RF.

Very minor issues

1. An extraneous/undefined equation involving J_2 appears in Sec. 3.1.1 and seems unrelated to the described feature set; symbol definitions are missing and it may be an artifact.

   Recommendation: Remove the J_2 line if accidental, or explicitly define J_2 (and Q, M, R) and state whether it is used as an input feature.

2. Citation and reference formatting is inconsistent (author-year variants, OCR-like list markers in References, possible duplicate/year-suffix inconsistencies), and the keyword list includes irrelevant terms (e.g., "Solar neutrinos").

   Recommendation: Normalize citations and bibliography to the target style; remove OCR artifacts; deduplicate; and update keywords to match the actual topic (merger trees, dark matter halos, tensor trains, feature engineering, ML).

3. Equation/notation presentation needs cleanup (inconsistent R^2 notation; unnumbered pooling equations; inconsistent math spacing such as $z = 0$ vs $z = 0$; inline equations embedded in long sentences) (Sec. 3.4–3.6).

   Recommendation: Standardize to a single notation (use R^2), number key equations that are referenced later, and format important definitions as display equations with symbols defined on first use.

4. Wording about padding uses "next power of 2," which can be misread as strictly greater even when already a power of two (Sec. 3.3).

   Recommendation: Clarify phrasing to "smallest power of two $\geq$ current size" and state explicitly whether equality is allowed.

Key statements and references

- ✖ **The analysis uses a dataset of 300 cosmological merger trees, each representing the hierarchical assembly history of a dark matter halo, stored in PyTorch Geometric format and based on merger-tree generation methods developed in prior work on dark matter halo merger trees and their accuracy for semi-analytic galaxy formation models.**
- *Reference(s):* Parkinson et al., 2007, Jiang and van den Bosch, 2013, Ángel Chandro-Gómez et al., 2025
- *Justification:* Both references discuss what merger trees are and present/assess algorithms for generating accurate dark-matter halo merger trees for use in semi-analytic models (Parkinson et al., 2007; Jiang and van den Bosch, 2013). However, neither paper mentions any dataset of **300** merger trees, nor any storage in PyTorch Geometric format, nor a specific dataset used in an analysis. Thus the statement's key dataset/format claims are not supported by these papers.

- ⚠ **To reduce the dynamic range and approximate normality, the halo mass and $v_{\max}$ node features are transformed using the natural logarithm before further processing, following prior applications of log-transformations to cosmological density and convergence fields and related large-scale-structure statistics.**
- *Reference(s):* Seo et al., 2011, Greiner and Enßlin, 2014, Rubira and Voivodic, 2021
- *Justification:* The cited works show that logarithmic transforms have been used to Gaussianize and reduce dynamic range in large-scale-structure fields: log-transforming the weak-lensing convergence field makes the one-point PDF more Gaussian and improves covariance properties (Seo et al., 2011), $\log(1+\delta)$ behaves approximately Gaussian and reduces non-linearities in the matter power spectrum (Greiner and Enßlin, 2014), and $A = \ln(1+\delta_R)$ Gaussianizes and linearizes the density field (Rubira and Voivodic, 2021). However, none of the papers discuss applying log transforms to halo mass or $v_{\max}$ node features specifically. Thus, the rationale is supported, but the specific features mentioned are not.
- ✖ **For each merger tree, the main progenitor branch is identified using a mask derived from the TreeFrog merger-tree construction methodology, and k-hop neighborhoods around these main-branch nodes are extracted as localized subgraphs to capture their immediate assembly history.**
- *Reference(s):* Elahi et al., 2019, Jespersen et al., 2022
- *Justification:* Elahi et al., 2019 defines main branches within TreeFrog merger trees and details how progenitor/descendant links are constructed, but does not describe creating a mask for later k-hop subgraph extraction. Jespersen et al., 2022 uses ROCKSTAR + ConsistentTrees merger trees and encodes the entire (pruned) merger tree as a directed graph for a GNN; they do not identify main-branch nodes via TreeFrog, nor extract k-hop neighborhoods around main-branch nodes as localized subgraphs. Instead they retain specific node types (progenitor, pre-/post-merger, final) and perform message passing on the full tree.
- ✖ **Each subgraph's node feature matrix is reshaped (e.g., to a power-of-two size) and decomposed using Quantum Tensor Train (QTT) methods that approximate the matrix as a product of low-rank tensor cores, leveraging recent advances in efficient tensor-train and quantized-tensor-network algorithms for high-dimensional problems.**
- *Reference(s):* Bharadwaj et al., 2024, Erpenbeck et al., 2023, Matveev and Smirnov, 2024
- *Justification:* Neither paper mentions subgraphs, node feature matrices, power-of-two reshaping, or Quantum Tensor Train (QTT). Bharadwaj et al., 2024 develops efficient sampling for standard TT-ALS; Erpenbeck et al., 2023 applies standard tensor

train/cross-interpolation to quantum impurity integrals. While both describe TT as products of low-rank cores, there is no discussion of QTT or graph-specific preprocessing, so the statement is not supported.

- ✖ **Aggregated QTT-based feature vectors are used as inputs to Random Forest regressors, whose hyperparameters (number of trees, maximum depth, and minimum samples per split) are tuned via cross-validation, following established practice for probabilistic and ensemble Random Forest methods in astronomical regression tasks and for baseline halo-mass prediction from traditional features.**

- *Reference(s):* Reis et al., 2018, Fujita and Aung, 2019, Larson et al., 2024
- *Justification:* None of the papers mention QTT-based features or using them with Random Forest regressors. Larson et al. (2024) do use Random Forests for baseline halo-mass prediction from traditional features (stellar mass, morphology, overdensity), but they fix 100 estimators and do not tune hyperparameters such as number of trees, max depth, or min samples per split via cross-validation. Reis et al. (2018) presents a Probabilistic Random Forest focused on classification, not demonstrating cross-validated hyperparameter tuning for regression, and Fujita and Aung (2019) does not address machine learning. Therefore the statement is not supported by the attached papers.

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 number of explicit mathematical formulas (standardization and regression metrics) and mostly qualitative descriptions of the QTT decomposition/feature construction. The central mathematical mechanism (QTT factorization) is described narratively without formal equations, limiting the ability to audit the core method symbolically. Several internal consistency issues exist in preprocessing definitions and cross-references, plus an apparent stray undefined equation fragment.

Checked items

1. ✔ **Log-transform definitions (mass, $v_{\max}$)** (Sec. 3.1.1, p.2)
    - **Claim:** Mass and $v_{\max}$ are log-transformed using the natural logarithm: $Mass_{transformed} = \log(Mass)$, $v_{\max}^{transformed} = \log(v_{\max})$.
    - **Checks:** definition consistency, notation sanity
    - **Verdict:** PASS; confidence: high; impact: minor
    - **Assumptions/inputs:** log denotes the natural logarithm as stated in text
    - **Notes:** Definitions are syntactically correct. (Separately, an unrelated stray equation appears nearby and is flagged in another item.)

2. ✖ **Extraneous undefined equation fragment** (Sec. 3.1.1, p.2 (between log-transform lines))
    - **Claim:** The PDF text includes: "$J_2 = QMR^2$".
    - **Checks:** symbol definition consistency, internal relevance
    - **Verdict:** FAIL; confidence: medium; impact: moderate
    - **Notes:** Symbols J_2, Q, M, R are not defined anywhere in the paper and the expression is unrelated to the stated methodology. This reads like an accidental paste/typesetting corruption; if so it should be removed/fixed.
3. ✔ **Standardization equation** (Eq. (1), Sec. 3.1.2, p.2)
    - **Claim:** Features are standardized by $x_{\text{normalized}} = \frac{x-\mu}{\sigma}$, with μ and σ computed from the training set.
    - **Checks:** algebra, definition consistency
    - **Verdict:** PASS; confidence: high; impact: minor
    - **Assumptions/inputs:** $\sigma \neq 0$ for each standardized feature
    - **Notes:** Equation is mathematically correct for z-scoring; variable meanings are stated.
4. ✖ **Normalization-statistics source inconsistency** (Sec. 3.1.2, p.2 vs Sec. 4.2, p.5)
    - **Claim:** Sec. 3.1.2: μ, σ computed from training set; Sec. 4.2: standardization based on global statistics from the entire dataset.
    - **Checks:** definition consistency, pipeline consistency
    - **Verdict:** FAIL; confidence: high; impact: moderate
    - **Assumptions/inputs:** A training set exists (even if evaluation is in-sample later)
    - **Notes:** These two statements are mutually inconsistent definitions of the preprocessing transform. This affects what $x_{\text{normalized}}$ means (even without checking any numbers).
5. ✔ **Mean pooling aggregation formula** (Sec. 3.4.1, p.3)
    - **Claim:** Tree feature vector is the mean of subgraph QTT vectors: $\frac{1}{N}\sum_{i=1}^{N} QTTVector_i$.
    - **Checks:** algebra, shape/compatibility
    - **Verdict:** PASS; confidence: medium; impact: minor
    - **Assumptions/inputs:** All QTT vectors have identical length for a given experiment
    - **Notes:** Averaging vectors is consistent provided vector lengths match; the paper implies fixed-length vectors but does not formally prove it.
6. ✔ **Max pooling aggregation definition** (Sec. 3.4.2, p.3)
    - **Claim:** Tree feature vector is the element-wise maximum over QTT vectors.

- **Checks:** definition consistency, shape/compatibility
- **Verdict:** PASS; confidence: high; impact: minor
- **Assumptions/inputs:** All QTT vectors have identical length, Max is taken element-wise
- **Notes:** Definition is coherent; element-wise max requires consistent vector dimension.

7. ✔ **Concatenation aggregation and padding** (Sec. 3.4.3, p.3)
   - **Claim:** Concatenate QTT vectors from last n nodes; if fewer than n nodes, pad with zeros.
   - **Checks:** shape/compatibility, definition consistency
   - **Verdict:** PASS; confidence: medium; impact: minor
   - **Assumptions/inputs:** A fixed ordering of nodes along the main progenitor branch exists, Per-node QTT vectors have identical length
   - **Notes:** Mechanically consistent; however the paper does not specify the exact ordering rule or how ties/branch ambiguities are handled (not a mathematical contradiction, but a clarity gap).
8. ✔ **Baseline feature dimension count** (Sec. 4.4.1, p.6)
   - **Claim:** Baseline features use mean, max, variance of 4 node features → 12-dimensional vector.
   - **Checks:** counting/dimensionality
   - **Verdict:** PASS; confidence: high; impact: minor
   - **Assumptions/inputs:** Exactly four node features are used (mass, concentration, $v_{\max}$, scale factor)
   - **Notes:** 4 features $\times \mathbf{3}$ statistics $= \mathbf{12}$; internally consistent.
9. ✔ **MSE metric formula** (Sec. 3.6.1, p.4)
   - **Claim:** $\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$.
   - **Checks:** algebra, definition consistency
   - **Verdict:** PASS; confidence: high; impact: minor
   - **Assumptions/inputs:** N is the number of samples
   - **Notes:** Matches the textual definition.
10. ✔ **MAE metric formula** (Sec. 3.6.2, p.4)
    - **Claim:** $\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$.
    - **Checks:** algebra, definition consistency
    - **Verdict:** PASS; confidence: high; impact: minor
    - **Assumptions/inputs:** N is the number of samples
    - **Notes:** Matches the textual definition.
11. ✔ **R^2 metric formula** (Sec. 3.6.3, p.4)

- **Claim:** $R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}$.
- **Checks:** algebra, definition consistency
- **Verdict:** PASS; confidence: high; impact: minor
- **Assumptions/inputs:** Denominator $\sum(y_i - \bar{y})^2 \neq 0$
- **Notes:** Standard definition; consistent with text.

12. ✔ **Example reshape for QTT ($k = 1$)** (Sec. 4.3, p.6)

    - **Claim:** A padded node-feature matrix of shape $[8, 4]$ is reshaped to $[2, 2, 2, 4]$ before QTT decomposition.
    - **Checks:** dimensionality/shape consistency
    - **Verdict:** PASS; confidence: high; impact: minor
    - **Assumptions/inputs:** Reshape only splits the node dimension: $8 = 2 \cdot 2 \cdot 2$
    - **Notes:** Shape arithmetic is consistent: $2 \times 2 \times 2 \times 4 = 8 \times 4$ elements.

13. ⚠ **Fixed-length QTT feature vector claim (flatten+concatenate cores)** (Sec. 3.3, p.3 (and referenced in Sec. 4.3, p.6))

    - **Claim:** QTT produces tensor cores that can be flattened and concatenated into a fixed-length feature vector per subgraph.
    - **Checks:** shape/compatibility, missing derivation/formal definition check
    - **Verdict:** UNCERTAIN; confidence: medium; impact: critical
    - **Assumptions/inputs:** Padding/reshaping scheme is fixed across subgraphs, QTT rank is fixed per experiment, Core dimensions depend only on mode sizes and ranks
    - **Notes:** The paper does not provide the formal TT/QTT core-shape equations (as functions of mode sizes and ranks), nor the exact definition of 'QTT rank' used. Without these, it cannot be verified that concatenation yields a fixed-length vector under all cases (especially with variable subgraph sizes/padding/truncation).

14. ✖ **Cross-reference to table/section** (Sec. 4.6, p.10)

    - **Claim:** Text says 'Analyzing the table in Section 3.2' while the performance table appears as Table 1 in Sec. 4.4.2 and Sec. 3.2 is subgraph extraction.
    - **Checks:** cross-reference consistency
    - **Verdict:** FAIL; confidence: high; impact: minor
    - **Notes:** This is a document-internal referencing inconsistency (not affecting formulas directly, but it impedes traceability of claims).

Limitations

- The paper provides very limited formal mathematics for QTT; most of the core method is described qualitatively, so a detailed symbolic derivation audit is not possible.

- Some problematic strings (e.g., the '$J_2 = QMR^{2}$' fragment) could be typesetting/OCR corruption; this audit treats them as part of the provided PDF text because no corrected source is available.
- No explicit notation is given for tensor core indices, TT-ranks across modes, or vectorization order; therefore, checks involving QTT feature dimensionality and invariance are necessarily marked UNCERTAIN.

Numerical results audit

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

All executed internal arithmetic/logic checks on the provided numeric statements passed. Verified items include: dataset reduction logic (300 to effective $N = 5$), equality of "5 unique subgraphs" and "5 distinct trees," averages lying within stated min/max ranges for $k = 1..3$, power-of-two padding examples relative to max node counts, reshape element-count preservation, baseline feature dimensionality (3 aggregations $\times$ 4 node features $= 12$), identification of best QTT configuration by MSE and R^2 from the provided table values, baseline vs best-QTT R^2 difference (0.048), PCA two-component explained-variance totals, and monotonic trends of R^2 and MSE with k (rank 2) plus reconstruction-MSE improvement from rank 2 to rank 3.

Checked items

1. ✔ **C1_dataset_reduction_300_to_5** (p.5 (Sec. 4.2 Data Preprocessing and Subgraph Extraction Yield); also Abstract p.1)
    - **Claim:** Initial dataset comprised 300 merger trees, but only 5 distinct trees could be successfully processed (effective sample size $N = 5$).
    - **Checks:** internal_consistency_repeated_constants
    - **Verdict:** PASS
    - **Notes:** Effective sample size is not greater than initial dataset size. Cross-section constant-matching cannot be verified without full text; checked provided values only.
2. ✔ **C2_subgraphs_unique_equals_trees_processed** (p.5 (Sec. 4.2))
    - **Claim:** Across all 300 trees and $k = 1, 2, 3$, only a total of 5 unique subgraphs originating from 5 distinct trees could be successfully processed.
    - **Checks:** internal_consistency_equality_of_counts
    - **Verdict:** PASS
    - **Notes:** Equality check on provided counts. Cross-check to later '$N = 5$' cannot be performed without additional text beyond provided values.
3. ✔ **C3_k1_nodes_avg_with_min_max** (p.5 (Sec. 4.2))

- **Claim:** For $k = 1$, extracted subgraphs had an average of **4.0** nodes (min: **3**, max: 8).
- **Checks:** range_check_average_between_min_max
- **Verdict:** PASS
- **Notes:** Average lies within [min, max].

4. ✔ **C4_k2_nodes_avg_with_min_max** (p.5 (Sec. 4.2))
   - **Claim:** For $k = 2$, average was **8.4** nodes (min: **5**, max: **19**).
   - **Checks:** range_check_average_between_min_max
   - **Verdict:** PASS
   - **Notes:** Average lies within [min, max].
5. ✔ **C5_k3_nodes_avg_with_min_max** (p.5 (Sec. 4.2))
   - **Claim:** For $k = 3$, average was **13.4** nodes (min: **7**, max: **32**).
   - **Checks:** range_check_average_between_min_max
   - **Verdict:** PASS
   - **Notes:** Average lies within [min, max].
6. ✔ **C6_padding_next_power_of_two_k1** (p.5 (Sec. 4.2))
   - **Claim:** Subgraphs' node feature matrices were padded to the next power of **2** in the node dimension (e.g., 8 for $k = 1$).
   - **Checks:** power_of_two_and_next_ge_max
   - **Verdict:** PASS
   - **Notes:** Pad size is a power of two and equals max; consistent if 'next power of two' interpreted as $\geq$ (not strictly $>$).
7. ✔ **C7_padding_next_power_of_two_k2** (p.5 (Sec. 4.2))
   - **Claim:** Padded to next power of **2** (e.g., **32** for $k = 2$).
   - **Checks:** power_of_two_and_next_ge_max
   - **Verdict:** PASS
   - **Notes:** Pad size matches the next power-of-two $\geq$ max.
8. ✔ **C8_padding_next_power_of_two_k3** (p.5 (Sec. 4.2))
   - **Claim:** Padded to next power of **2** (e.g., **32** for $k = 3$).
   - **Checks:** power_of_two_and_next_ge_max
   - **Verdict:** PASS
   - **Notes:** Pad size is a power of two and equals max; consistent if 'next power of two' interpreted as $\geq$ (not strictly $>$).
9. ✔ **C9_k1_tensor_reshape_product** (p.6 (Sec. 4.3))

- **Claim:** For $k = 1$: padded matrix shape $[8, 4]$ reshaped into tensor $[2, 2, 2, 4]$.
- **Checks:** shape_product_invariance
- **Verdict:** PASS
- **Notes:** Element-count preserved under reshape.

10. ✔ **C10_baseline_feature_dimensionality_12** (p.6 (Sec. 4.4.1 Baseline Model))
    - **Claim:** Baseline features: mean, maximum, and variance of the four preprocessed node features → 12-dimensional feature vector per tree.
    - **Checks:** cheap_recomputation_dimension_count
    - **Verdict:** PASS
    - **Notes:** Recomputed dimensionality matches.
11. ✔ **C11_table1_best_mse_identification** (p.6 (Table 1) and p.6-8 (Fig.9 caption text))
    - **Claim:** Claim: QTT model with $k = 1$ and rank $= 2$ achieves the lowest MSE among QTT configurations (and in Fig.9, 'achieves the lowest MSE').
    - **Checks:** min_selection_from_table
    - **Verdict:** PASS
    - **Notes:** Target is the minimum among provided QTT MSEs.
12. ✔ **C12_table1_best_r2_identification** (p.6 (Sec. 4.4.3) and Table 1)
    - **Claim:** Claim: QTT model with $k = 1$, rank $= 2$ showed the best performance ($R^2 = 0.845$) among all models; Table 1 lists R^2 values for QTT configurations.
    - **Checks:** max_selection_from_table
    - **Verdict:** PASS
    - **Notes:** Target is the maximum among provided QTT R^2s (note: 'among all models' would require baseline/other models too).
13. ✔ **C13_baseline_vs_best_qtt_r2_comparison** (p.6 (Sec. 4.4.1 and 4.4.3); p.12 (Conclusions))
    - **Claim:** Baseline $R^2 = 0.797$; best QTT model $R^2 = 0.845$; claim that best QTT slightly outperforms baseline.
    - **Checks:** difference_sign_check
    - **Verdict:** PASS
    - **Notes:** Best QTT exceeds baseline and delta matches within tolerance.
14. ✔ **C14_pca_baseline_explained_variance_sum** (p.8 (Sec. 4.5.2 Dimensionality Reduction (PCA)))
    - **Claim:** Baseline PCA: first two components explain 75.7% and 23.6% (total $\sim 99.3\%$).

- **Checks:** percentage_sum_to_total
- **Verdict:** PASS
- **Notes:** Sum matches stated total within tolerance.

15. ✔ **C15_pca_qtt_explained_variance_sum** (p.8 (Sec. 4.5.2 Dimensionality Reduction (PCA)))
    - **Claim:** QTT ($k = 1$, rank $= 2$) PCA: first two components explain 71.3% and 24.0% (total $\sim 95.3\%$).
    - **Checks:** percentage_sum_to_total
    - **Verdict:** PASS
    - **Notes:** Sum matches stated total within tolerance.
16. ✔ **C16_rank2_r2_trend_with_k** (p.10 (Sec. 4.6.1 Impact of k) referencing Table 1)
    - **Claim:** For rank 2, performance (R^2) decreased as k increased: $k = 1 : 0.845$, $k = 2 : 0.834$, $k = 3 : 0.798$.
    - **Checks:** monotonicity_check
    - **Verdict:** PASS
    - **Notes:** Trend holds.
17. ✔ **C17_rank2_mse_trend_with_k** (p.6 (Table 1))
    - **Claim:** For rank 2, MSE increases with k ($k = 1 : 0.00151$, $k = 2 : 0.00161$, $k = 3 : 0.00196$).
    - **Checks:** monotonicity_check
    - **Verdict:** PASS
    - **Notes:** Trend holds.
18. ✔ **C18_qtt_recon_mse_rank_improves_k1** (p.6 (Sec. 4.3))
    - **Claim:** Reconstruction MSE for $k = 1$: rank 2 average ≈ 0.032; rank 3 reduces to 0.0053.
    - **Checks:** directional_improvement_check
    - **Verdict:** PASS
    - **Notes:** Rank 3 improves (lower MSE) vs rank 2.
19. ✔ **C19_qtt_recon_mse_rank_improves_k2** (p.6 (Sec. 4.3))
    - **Claim:** Reconstruction MSE for $k = 2$: rank 2 MSE 0.033; rank 3 MSE 0.016.
    - **Checks:** directional_improvement_check
    - **Verdict:** PASS
    - **Notes:** Rank 3 improves (lower MSE) vs rank 2.
20. ✔ **C20_qtt_recon_mse_rank_improves_k3** (p.6 (Sec. 4.3))

- **Claim:** Reconstruction MSE for $k = 3$: rank 2 MSE 0.057; rank 3 MSE 0.027.
- **Checks:** directional_improvement_check
- **Verdict:** PASS
- **Notes:** Rank 3 improves (lower MSE) vs rank 2.

Limitations

- Only the provided PDF text was used; no underlying data, code, or supplementary materials were available to recompute model metrics, PCA, or standardization statistics.
- Numeric checks were limited to internal arithmetic/logic (e.g., sums, minima/maxima, monotonic trends, shape product invariance) that can be validated from explicit numbers present in the document.
- Figures were not mined for numeric values (no pixel/plot-value extraction); only numbers explicitly written in captions/body/table were used.
- Cross-section constant-matching (e.g., verifying that effective sample size $N = 5$ and other repeated constants match everywhere they are referenced) could not be fully validated beyond the provided extracted values.
- Items requiring underlying data or intermediate outputs (standardization statistics, raw value ranges, regression metrics recomputation, PCA explained-variance computation, and derived feature counts) remain unverified for the reasons listed in the unverified items.