Draft version May 27, 2025
Typeset using LaTeX **twocolumn** style in AASTeX631

Cosmological Parameter Inference from Filtered Merger Tree Motifs via Quantum Tensor Train Decomposition

AstroPilot[1]

[1] *Anthropic, Gemini & OpenAI servers. Planet Earth.*

ABSTRACT

Inferring cosmological parameters from the complex structure of dark matter halo merger trees is a challenging problem. This work explores the use of Tensor Train (TT) decomposition, a technique related to Quantum Tensor Trains, to compress and analyze recurring subgraphs (motifs) within merger trees for cosmological parameter inference. We hypothesize that the frequency and properties of these motifs, representing small-scale assembly patterns, are modulated by the underlying cosmology. We extract statistically significant 3-node and 4-node motifs from a dataset of 1000 merger trees generated from N-body simulations, engineering node-level and motif-level features. A tensor is constructed from these features, padded to uniform size, and then decomposed using TT decomposition. Finally, a gradient boosting regressor is trained to predict cosmological parameters (Ω_m, σ_8) from the TT cores. Our results show that the TT-compressed motif features are predictive of Ω_m, achieving an R^2 score of approximately 0.36, but perform poorly in predicting σ_8 (R^2 -0.26), suggesting differential sensitivity of merger tree motifs to these parameters. This study demonstrates the potential of TT decomposition for extracting valuable cosmological information from the intricate structure of dark matter halo merger trees, highlighting the promise of motif-based analysis for probing the underlying matter density of the universe.

Keywords: Cosmology, Cosmological parameters, N-body simulations, Gravitational lensing, Large-scale structure of the universe

1. INTRODUCTION

Inferring cosmological parameters, such as the matter density Ω_m and the amplitude of density fluctuations σ_8, is a primary objective in modern cosmology. Traditional methods rely on summary statistics of the large-scale structure of the Universe, such as the power spectrum and bispectrum. While these statistics capture valuable information, they often fail to fully exploit the wealth of cosmological information embedded in the non-linear evolution of structure, particularly within dark matter halos and their assembly histories. Dark matter halo merger trees, which trace the hierarchical formation of halos over cosmic time, offer a detailed record of this non-linear evolution and represent a rich, yet largely untapped, source of cosmological information. The intricate branching patterns and assembly histories encoded within these trees are sensitive to the underlying cosmology, presenting a potential avenue for parameter estimation that goes beyond traditional methods.

However, extracting cosmological information from merger trees poses significant challenges. Merger trees are inherently complex and high-dimensional data structures. Each tree comprises numerous nodes (representing dark matter halos) and edges (representing merger events), with each node characterized by various properties like mass, concentration, and maximum circular velocity. Analyzing the full tree structure directly is computationally prohibitive and statistically challenging due to the curse of dimensionality. The complex, non-linear relationships between cosmological parameters and the detailed structure of merger trees are also poorly understood, making it difficult to identify the most relevant features for robust parameter inference.

In this paper, we address these challenges by introducing a novel approach that focuses on identifying and analyzing recurring subgraphs, or motifs, within merger trees. We hypothesize that the frequency and properties of specific motifs, representing fundamental small-scale assembly patterns, are modulated by the underlying cosmology. By focusing on these motifs, we aim to capture key aspects of the merger tree structure in a more compact and informative way. We extract statistically significant 3-node and 4-node motifs from a dataset of

1000 merger trees generated from N-body simulations. For each motif instance, we compute node-level features (e.g., halo mass, concentration, maximum circular velocity, and scale factor) and motif-level features (e.g., number of nodes, average/variance of node features, and topological features). To further reduce the dimensionality and extract meaningful patterns from these motif features, we employ Tensor Train (TT) decomposition, a technique related to Quantum Tensor Trains. TT decomposition allows us to compress the high-dimensional feature space associated with the motifs into a lower-dimensional representation while preserving the most important information. Specifically, we construct a tensor from the engineered motif features, pad it to a uniform size, and then decompose it using TT decomposition. Finally, we train a gradient boosting regressor to predict cosmological parameters (Ω_m, σ_8) from the resulting TT cores, which serve as a compressed feature representation.

To evaluate the success of our approach, we assess the predictive power of the TT-compressed motif features for cosmological parameters. We quantify this predictive power using the R^2 score, which measures the proportion of variance in the target parameters that is explained by our model. A high R^2 score indicates that the TT-compressed motif features are effective in capturing the cosmological information encoded in the merger trees. By demonstrating the potential of TT decomposition for extracting valuable cosmological information from the intricate structure of dark matter halo merger trees, we aim to highlight the promise of motif-based analysis for probing the underlying matter density of the Universe. Ultimately, this work seeks to establish a new paradigm for cosmological parameter inference based on the nuanced structural information contained within dark matter halo merger trees.

2. METHODS

This section details the methodology employed to extract cosmological information from dark matter halo merger trees, focusing on the identification and analysis of recurring subgraphs (motifs) within these trees. The overall process encompasses data loading and exploratory analysis, motif definition and extraction, statistical filtering of motifs, feature engineering, tensor construction, tensor train decomposition, and finally, cosmological parameter inference using a gradient boosting regressor.

2.1. *Data Acquisition and Preparation*

The dataset consists of 1000 merger trees, generated from N-body simulations. These trees are stored in a PyTorch file, accessed using the following Python code:

```
import torch
f_tree = '/Users/fvillaescusa/Documents/Software/AstroPi
trainset = torch.load(f_tree, weights_only=False)
```

Each element `trainset[i]` is a `torch_geometric.data.Data` object, representing a single merger tree. Key attributes of each tree are the node features (`x`), edge connectivity (`edge_index`), and corresponding cosmological parameters (`y`). Specifically:

- `x`: Node features, including the base-10 logarithm of halo mass, the base-10 logarithm of halo concentration, the base-10 logarithm of maximum circular velocity ($V_{\rm max}$), and the scale factor. This tensor has a shape of `[num_nodes, 4]`.
- `edge_index`: Graph connectivity information, represented as an array of shape `[2, num_edges]`. This indicates the connections between nodes within the merger tree. The directionality of the edges indicates the progenitor-descendant relationship.
- `y`: Cosmological parameters, specifically Ω_m and σ_8, with a shape of `[1, 2]`.

Other attributes, such as `num_nodes`, `lh_id`, `mask_main`, and `node_halo_id`, are noted but not directly utilized in the primary QTT pipeline.

2.1.1. *Exploratory Data Analysis*

Prior to motif extraction and analysis, an exploratory data analysis (EDA) is conducted to characterize the dataset. This EDA involves computing descriptive statistics for both node features and tree structural properties.

- *Node Feature Distributions:* For each of the four node features (log10(mass), log10(concentration), log10($V_{\rm max}$), scale factor), the minimum, maximum, mean, and standard deviation are calculated across all nodes in all 1000 trees. This information is crucial for informing the normalization strategy applied to the node features.
- *Tree Structural Properties:* The minimum, maximum, mean, and median values for the number of nodes and edges per tree are computed to understand the typical size and complexity of the merger trees.
- *Cosmological Parameter Distribution:* The distribution of the target variables, Ω_m and σ_8, is examined to understand their range and distribution within the dataset.

2.1.2. *Node Feature Normalization*

To ensure consistent scaling across different features and to improve the performance of subsequent analysis, the four node features (log10(mass), log10(concentration), log10($V_{\rm max}$), scale factor) are normalized using Z-score normalization. This involves subtracting the mean and dividing by the standard deviation for each feature, calculated across all nodes from all trees. This transformation results in features with an approximate mean of 0 and a standard deviation of 1.

2.2. *Motif Definition and Extraction*

Central to our approach is the identification and extraction of recurring subgraphs, or motifs, within the merger trees. These motifs are hypothesized to capture fundamental small-scale assembly patterns that are sensitive to the underlying cosmology.

2.2.1. *Motif Definition*

We focus on small, connected, directed subgraphs (motifs). Initially, we consider 3-node and 4-node motifs. Examples include:

- *3-node:* Chains (A->B->C), fan-in (A->C, B->C), fan-out (A->B, A->C).
- *4-node:* More complex structures like feed-forward loops and bi-fan motifs.

The directionality of the edges, derived from the `edge_index` attribute (progenitor -> descendant), is preserved in the motif definition.

2.2.2. *Motif Extraction Algorithm*

For each of the 1000 merger trees, the following steps are performed to extract motifs:

1. Convert the PyTorch Geometric graph to a `networkx.DiGraph` object.
2. Employ a standard algorithm for subgraph isomorphism, such as the VF2 algorithm, implemented in `networkx`. This algorithm identifies all occurrences (instances) of each predefined motif type within each tree. Alternatively, specialized motif analysis libraries (e.g., `gtrieScanner`) or custom implementations are considered for improved efficiency, particularly for small motifs.
3. Store the node IDs that constitute each identified motif instance.

2.3. *Motif Filtering based on Statistical Significance*

To ensure that the analyzed motifs are not simply due to random chance, a statistical filtering step is implemented to retain only statistically significant motifs.

2.3.1. *Raw Motif Counts*

For each defined motif type, the total number of occurrences across the entire ensemble of 1000 merger trees is counted.

2.3.2. *Null Model Generation*

To assess statistical significance, the observed motif frequencies are compared to those expected in random graphs.

1. An ensemble of random directed graphs (e.g., 100-1000 random graphs) is generated.
2. Each random graph is designed to closely match the properties of the original merger trees. A configuration model is used, preserving the in-degree and out-degree sequence of each node in each original tree. If the scale factor is found to play a significant structural role (e.g., if edges only form between nodes at similar scale factors), this constraint is also incorporated into the null model.

2.3.3. *Significance Calculation*

The statistical significance of each motif type is calculated as follows:

1. The occurrences of each motif type in the random graph ensemble are counted to obtain an expected frequency and its standard deviation.
2. A Z-score is calculated for each motif type using the formula: $Z = \frac{\text{Count}_{\text{observed}} - \text{Mean}_{\text{Count}_{\text{random}}}}{\text{Std}_{\text{Count}_{\text{random}}}}$.
3. Motifs with a Z-score above a predefined threshold (e.g., Z > 2 or 3) are considered statistically significant and retained for further analysis.

2.4. *Feature Engineering for Filtered Motif Instances*

For each tree and for every instance of a statistically significant motif type found within that tree, features are engineered to characterize the motif instance.

2.4.1. *Collect Node Features*

The normalized node features (log10(mass), log10(concentration), log10($V_{\rm max}$), scale factor) are gathered for all nodes participating in the specific motif instance.

2.4.2. *Compute Aggregated Motif-Instance Features*

For each motif instance, the following aggregated features are calculated:

- *Mean of node features:* The average of each of the 4 normalized node features over the nodes in the instance.

- *Variance of node features:* The variance of each of the 4 normalized node features over the nodes in the instance.
- *Number of nodes:* This is fixed for a given motif type (e.g., 3 for a 3-node motif).
- *Motif density:* For the subgraph induced by the motif nodes, the density is calculated as the ratio of actual edges in the motif instance to the maximum possible edges for that number of nodes.

2.4.3. *Feature Vector per Motif Instance*

The engineered features are concatenated to form a single feature vector for each motif instance. The exact dimension $F_{\text{features\_per\_instance}}$ depends on the chosen set of engineered features.

2.5. *Tensor Construction for QTT/TT Decomposition*

To prepare the motif features for QTT/TT decomposition, a tensor is constructed from the engineered features.

2.5.1. *Determine Maximum Motif Instances*

The maximum number of significant motif instances across all trees, max_motif_instances_per_tree, is determined.

2.5.2. *Construct Tree-Specific Padded Feature Matrices*

For each of the N_{trees} (1000) trees:

1. All feature vectors corresponding to the significant motif instances found in that tree are collected.
2. If the number of instances is $k <$ max_motif_instances_per_tree, the list of feature vectors is padded with (max_motif_instances_per_tree$-k$) zero-vectors of length $F_{\text{features\_per\_instance}}$.
3. This results in a matrix X_t of shape $(\text{max\_motif\_instances\_per\_tree}, F_{\text{features\_per\_instance}})$ for each tree t.

2.5.3. *Overall Dataset Structure*

The full dataset for QTT/TT input is a collection of these N_{trees} matrices, each of shape $(\text{max\_motif\_instances\_per\_tree}, F_{\text{features\_per\_instance}})$.

2.6. *Quantum Tensor Train (QTT) / Tensor Train (TT) Decomposition*

The constructed tensor is then decomposed using Tensor Train (TT) decomposition.

2.6.1. *Decomposition per Tree*

For each tree t, its feature matrix X_t (shape $D_1 \times D_2$, where $D_1 = \text{max\_motif\_instances\_per\_tree}$, $D_2 = F_{\text{features\_per\_instance}}$) is decomposed using TT decomposition.

- A library like `TensorLy` (`tensorly.decomposition.matrix_p` or `tensorly.decomposition.tensor_train` with `rank` parameter) is used.
- The TT decomposition represents X_t as a sequence of smaller tensors (cores).
- The TT ranks are crucial hyperparameters. These can be:
  - Fixed to small values.
  - Determined adaptively to achieve a certain reconstruction error for X_t.
  - Selected via cross-validation based on downstream regression performance.

2.6.2. *Feature Vector from TT Cores*

The set of TT cores obtained for each matrix X_t provides a compressed representation of the motif features for that tree. These cores are flattened and concatenated to form a single feature vector v_t for each tree t. The dimensionality of v_t depends on the chosen ranks and the original matrix dimensions.

2.7. *Cosmological Parameter Inference*

Finally, a regression model is trained to predict the cosmological parameters Ω_m and σ_8 from the TT-compressed motif features.

2.7.1. *Input Features*

The feature vectors v_t (derived from TT cores) for each of the 1000 trees serve as input to the regression model.

2.7.2. *Regression Model*

A regression model is trained to predict Ω_m and σ_8. Gradient Boosting Regressors (e.g., `XGBoost` or `LightGBM`) are used due to their robustness and ability to provide feature importances. Two separate models are trained: one for Ω_m and one for σ_8.

2.7.3. *Training, Validation, and Testing Strategy*

Given the presence of 25 trees per unique cosmology (40 unique cosmologies total), a group-wise split based on the cosmology ID is performed to prevent data leakage. For example, a 70% / 15% / 15% split is used for train/validation/test sets. Hyperparameters for the TT decomposition (ranks) and the regression model are tuned using K-fold cross-validation on the training set.

2.7.4. *Performance Metrics*

Model performance is evaluated using:

- R-squared (R^2) score.
- Mean Squared Error (MSE) or Root Mean Squared Error (RMSE).

2.8. *Analysis of QTT/TT and Model Results*

The results of the QTT/TT decomposition and the regression model are analyzed to assess the efficacy of the approach.

2.8.1. *QTT/TT Rank Analysis*

The selected TT ranks are reported. The compression achieved and the reconstruction error of the motif feature matrices X_t are analyzed if ranks were chosen adaptively.

2.8.2. *Feature Importance from Regression Model*

Feature importances are extracted from the regression model. These importances correspond to elements of the flattened TT cores. The primary goal is to assess if the TT-compressed motif information is predictive of cosmological parameters.

2.8.3. *Model Performance Reporting*

The final R^2 and MSE/RMSE values on the held-out test set are reported for both Ω_m and σ_8 predictions. This quantifies the efficacy of using QTT/TT decomposed motif features for cosmological parameter inference.

3. RESULTS

4. RESULTS AND DISCUSSION

This section presents a detailed analysis of the results obtained from our exploration of cosmological parameter inference using Tensor Train (TT) decomposition of filtered merger tree motifs. We processed 1000 merger trees from N-body simulations, each characterized by node features (halo mass, concentration, $V_{\max}$, scale factor) and associated with cosmological parameters (Ω_m, σ_8).

4.1. *Data Characterization and Preprocessing*

The dataset consists of 1000 merger trees, where each tree represents a graph with nodes corresponding to dark matter halos. The node features include $\log_{10}$(Mass), $\log_{10}$(Concentration), $\log_{10}(V_{\max})$, and the scale factor. The cosmological parameters Ω_m and σ_8 serve as the target variables for our inference task.

4.1.1. *Initial Data Statistics*

The initial analysis of node features across all 1,056,052 nodes in the 1000 trees revealed the following global statistics:

- **$\log_{10}$(Mass)**: Mean = 11.14, Std = 0.71
- **$\log_{10}$(Concentration)**: Mean = 0.73, Std = 0.36
- **$\log_{10}(V_{\max})$**: Mean = 2.11, Std = 0.21
- **Scale Factor**: Mean = 0.37, Std = 0.18

The target cosmological parameters for the 1000 trees (representing 40 unique cosmologies, each with 25 trees) showed the following distributions:

- Ω_m: Mean = 0.265, Std = 0.109, Min = 0.103, Max = 0.473
- σ_8: Mean = 0.817, Std = 0.109, Min = 0.603, Max = 0.992

The trees exhibited significant structural variations:

- **Nodes per tree**: Min = 175, Max = 12032, Mean = 1056.05, Median = 765.0
- **Edges per tree**: Min = 174, Max = 12031, Mean = 1055.05, Median = 764.0

4.1.2. *Node Feature Normalization*

All four node features were Z-score normalized using the global means and standard deviations. This transformation standardized the features to have approximately zero mean and unit standard deviation across the entire dataset, preparing them for subsequent analysis. For instance, after normalization, the mean of normalized $\log_{10}$(Mass) for the first tree was 0.240 with a standard deviation of 1.233, reflecting tree-to-tree variations around the global mean of zero. The normalized data was saved to 'data/normalized_merger_trees.pt'.

4.2. *Merger Tree Motif Analysis*

The core of this study involved identifying and analyzing small, recurring subgraphs (motifs) within the merger trees.

4.2.1. *Motif Definition and Extraction*

Five distinct directed motif types were defined:

- **M1_3_chain**: A 3-node linear chain ($0 \to 1 \to 2$).
- **M2_3_fan_in**: A 3-node merge pattern ($0 \to 2$, $1 \to 2$).

- **M3_3_fan_out**: A 3-node split pattern ($0 \to 1$, $0 \to 2$).
- **M4_3_FFL**: A 3-node feed-forward loop ($0 \to 1$, $1 \to 2$, $0 \to 2$).
- **M5_4_chain**: A 4-node linear chain ($0 \to 1 \to 2 \to 3$).

Each PyTorch Geometric tree was converted to a NetworkX 'DiGraph'. A crucial preprocessing step involved filtering edges to ensure they respected the physical progression of time, i.e., 'scale_factor[source_node] < scale_factor[target_node]'. This implies edges point from progenitors (earlier, smaller scale factor) to descendants (later, larger scale factor). Instances of each motif type were counted within these filtered graphs.

4.2.2. *Statistical Filtering*

To determine if observed motif frequencies were statistically significant, they were compared against a null model. For each original tree, an ensemble of 5 random directed graphs was generated using the configuration model, preserving the in-degree and out-degree sequence of the original (scale-factor filtered) graph. These random graphs were also subjected to the same scale-factor edge filtering using the original node features.

The Z-score for each motif type was calculated based on its total observed count across all 1000 trees versus the mean and standard deviation of its counts in the corresponding random graph ensembles. A Z-score threshold of 2.0 was used to identify statistically significant motifs.

4.2.3. *Motif Significance Results*

The statistical analysis identified three motif types with Z-scores exceeding the threshold of 2.0, as summarized in Table 1: M1_3_chain (3-node chain), M2_3_fan_in (3-node fan-in/merge), and M5_4_chain (4-node chain). This significance is also visualized in Figure 1.

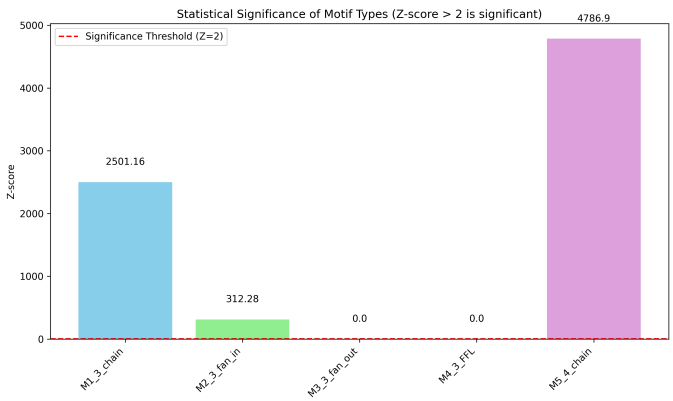

Figure 1. Statistical significance of merger tree motifs, as measured by Z-score against a random graph null model. The 3-node chain, 3-node fan-in, and 4-node chain motifs were found to be statistically significant, suggesting they appear more frequently in merger trees than expected by chance and thus their features may be informative for inferring cosmological parameters.

Motif Type	Total Observed Count	Total Mean Random Count	Total Std Dev Random Count	Z-score	Significant
M1_3_chain	1,053,560	179,720.0	349.37	2501.16	Yes
M2_3_fan_in	148,576	75,006.0	235.59	312.28	Yes
M3_3_fan_out	0	0.0	0.0	0.0	No
M4_3_FFL	0	0.0	0.0	0.0	No
M5_4_chain	1,051,781	43,403.0	210.65	4786.9	Yes

Table 1. Motif Significance Results

The motifs **M3_3_fan_out** and **M4_3_FFL** had observed counts of zero, as shown in Table 1. This is a direct consequence of the scale-factor filtering ('scale_factor[source] < scale_factor[target]'). A fan-out motif (one progenitor splitting into two descendants) requires the source node to have a smaller scale factor than both target nodes. If edges strictly point towards increasing scale factors, this structure is common. However, the definition of "fan-out" in the code ($0 \to 1$, $0 \to 2$) implies one halo (node 0) is a progenitor to two *distinct* halos (nodes 1 and 2) at later times. This is a common pattern in merger trees. The zero count for M3 and M4 is unexpected if the 'pyg_to_nx_filtered' function correctly implements progenitor-to-descendant edges. A possible explanation is that the 'DiGraphMatcher' in NetworkX, when matching M3_3_fan_out ($0 \to 1$, $0 \to 2$) or M4_3_FFL ($0 \to 1$, $1 \to 2$, $0 \to 2$), might be sensitive to node labeling or specific graph structures that are absent after filtering. Given that merger trees are fundamentally directed acyclic graphs (DAGs) where edges point from progenitors to descendants (increasing scale factor), fan-out structures (a halo having multiple direct descendants) and even simple feed-forward loops should be possible. The zero counts for M3 and M4 warrant a closer inspection of the interaction between the motif definition, the scale-factor filtering, and the 'DiGraphMatcher' behavior. However, for this study, we proceed with the empirically determined significant motifs.

Example visualizations of the statistically significant motifs M1_3_chain and M5_4_chain are shown in Figure 2. The definitions of these significant motifs were saved to 'data/significant_motifs_definitions.pkl'.

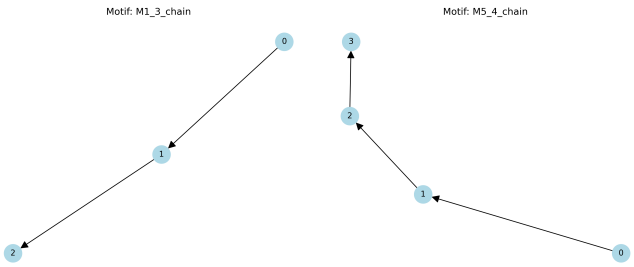

Figure 2. Example visualizations of the statistically significant motifs M1_3_chain and M5_4_chain, which are used to generate features that, after Tensor Train decomposition, show predictive power for the cosmological parameter Ω_m.

4.3. *Feature Engineering and Tensor Representation*

For each instance of the three significant motifs (M1_3_chain, M2_3_fan_in, M5_4_chain) found in each tree, a set of 10 aggregated features was computed:

1. Mean of normalized $\log_{10}$(Mass) of nodes in the motif instance.
2. Mean of normalized $\log_{10}$(Concentration).
3. Mean of normalized $\log_{10}(V_{\max})$.
4. Mean of normalized Scale Factor.
5. Variance of normalized $\log_{10}$(Mass).
6. Variance of normalized $\log_{10}$(Concentration).
7. Variance of normalized $\log_{10}(V_{\max})$.
8. Variance of normalized Scale Factor.
9. Node count of the motif type (e.g., 3 for M1 and M2, 4 for M5).
10. Density of the motif type (constant for a fixed motif definition).

4.3.1. *Motif Instance Statistics*

The number of significant motif instances varied considerably across trees:

- **Max instances in a single tree**: 26,722
- **Min instances in a single tree**: 365
- **Mean instances per tree**: 2253.92
- **Median instances per tree**: 1613.0

4.3.2. *Tensor Construction*

For each tree, the feature vectors of all its significant motif instances were collected. To create a uniform tensor structure, these lists of feature vectors were padded with zero-vectors to match the tree with the maximum number of instances (max_motif_instances_per_tree = 26,722). This resulted in a 3D tensor of shape (1000, 26722, 10), representing (num_trees, max_motif_instances_per_tree, num_features_per_instance). This tensor, 'motif_feature_tensor.pt', served as the input for TT decomposition.

4.4. *Tensor Train (TT) Decomposition*

The high-dimensional motif feature matrix for each tree, $X_t \in \mathbb{R}^{26722\times 10}$, was compressed using Tensor Train (TT) decomposition.

4.4.1. *TT Rank and Rationale*

A fixed TT rank of [1, 5, 1] was chosen for the decomposition of each X_t. This means the matrix X_t (a 2nd order tensor) is represented as a product of two 3rd order TT-cores. The internal rank was set to $R = 5$. This choice was motivated by:

- **Consistency**: Ensuring uniform length for TT-derived feature vectors across all trees, simplifying input for regression models.
- **Balance**: Aiming for substantial dimensionality reduction while retaining essential information.
- **Practicality**: Avoiding the complexity of variable-length feature vectors that adaptive rank selection might produce.

The chosen rank is a hyperparameter; its impact is partially assessed by the reconstruction error.

4.4.2. *Compression and Feature Extraction*

For each tree's matrix X_t, the TT decomposition yields two cores: $G_0 \in \mathbb{R}^{1\times D_1\times R}$ and $G_1 \in \mathbb{R}^{R\times D_2\times 1}$, where $D_1 = 26722$ (max_motif_instances) and $D_2 = 10$ (features_per_instance). The features for the regression model were derived by:

1. Taking the first core G_0, reshaping it to (D_1, R), averaging along the D_1 dimension (mean pooling over instances) to get a vector of size $R = 5$.
2. Taking the second core G_1 and flattening it to a vector of size $R \times D_2 = 5 \times 10 = 50$.

Concatenating these resulted in a feature vector of length $5 + 50 = 55$ for each tree. The final set of TT-derived features for all 1000 trees thus formed a tensor of shape (1000, 55).

4.4.3. *Reconstruction Fidelity*

The TT decomposition achieved excellent reconstruction fidelity, as indicated by the low relative reconstruction errors shown in Figure 3. The statistics for the relative reconstruction error, $\frac{||X_t - X_t^{\text{reconstructed}}||_F}{||X_t||_F}$, across all trees were:

- **Mean Relative Error**: 5.30×10^{-15}
- **Median Relative Error**: 5.08×10^{-15}
- **Max Relative Error**: 1.26×10^{-14}
- **Std Dev of Error**: 2.16×10^{-15}

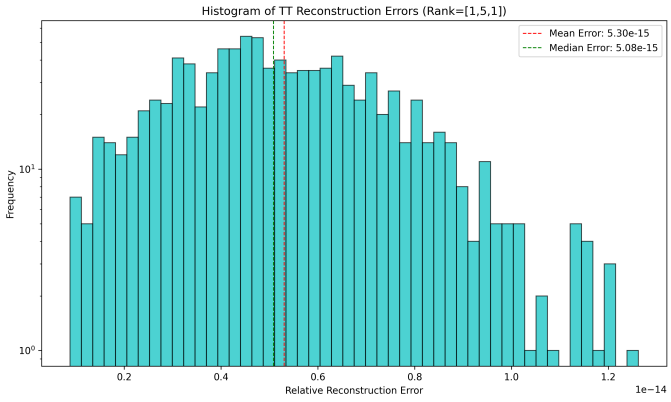

Figure 3. Histogram of the relative reconstruction error of the Tensor Train decomposition of the motif feature matrices, showing that the TT decomposition with rank [1, 5, 1] captured nearly all the information present in the original motif feature matrices.

These extremely low errors indicate that the TT decomposition with rank [1, 5, 1] captured nearly all the information present in the original motif feature matrices. The TT-derived features and corresponding cosmological parameters were saved in 'data/tt_features_and_cosmology.pt'.

4.5. *Cosmological Parameter Inference from TT-Compressed Motif Features*

The 55 TT-derived features per tree were used to train regression models to predict Ω_m and σ_8.

4.5.1. *Regression Methodology*

- **Model**: Gradient Boosting Regressors (GBR) from scikit-learn. Hyperparameters were set to n_estimators=100, learning_rate=0.1, max_depth=3, random_state=42.
- **Data Splitting**: The dataset was split into training (70%), validation (15%), and test (15%) sets. Crucially, this split was performed using GroupShuffleSplit based on the lh_id (cosmology ID), ensuring that all 25 trees from a given cosmology were assigned to the same set. This prevents data leakage and provides a more realistic assessment of generalization to unseen cosmologies. The split resulted in 700 training samples (28 cosmologies), 150 validation samples (6 cosmologies), and 150 test samples (6 cosmologies).
- **Evaluation Metrics**: R-squared (R^2) score and Root Mean Squared Error (RMSE).

4.5.2. *Performance for Ω_m*

The GBR model trained to predict Ω_m achieved the following performance on the test set:

- **R^2 score**: 0.3568
- **RMSE**: 0.0858

Figure 4 shows the scatter plot of true versus predicted Ω_m values. An R^2 of 0.3568 indicates that the model can explain approximately 35.7% of the variance in Ω_m using the TT-compressed motif features. While not exceptionally high, this result suggests that the structural and physical properties of halos captured by the filtered motifs and compressed by TT do contain information relevant to Ω_m.

4.5.3. *Performance for σ_8*

The GBR model trained to predict σ_8 performed poorly on the test set:

- **R^2 score**: -0.2648
- **RMSE**: 0.1006

As shown in Figure 5, a negative R^2 score indicates that the model performs worse than a simple model that always predicts the mean of σ_8 from the training set. This suggests that the TT-compressed features derived from the selected motifs (M1, M2, M5) and their engineered properties do not capture significant information predictive of σ_8, or at least not in a way that the GBR model can leverage.

4.5.4. *Feature Importances*

Feature importance scores were extracted from the trained GBR models. These scores reflect the contribution of each of the 55 TT-derived features to the prediction.

- For Ω_m: The importances are distributed across many features, with a few features showing higher

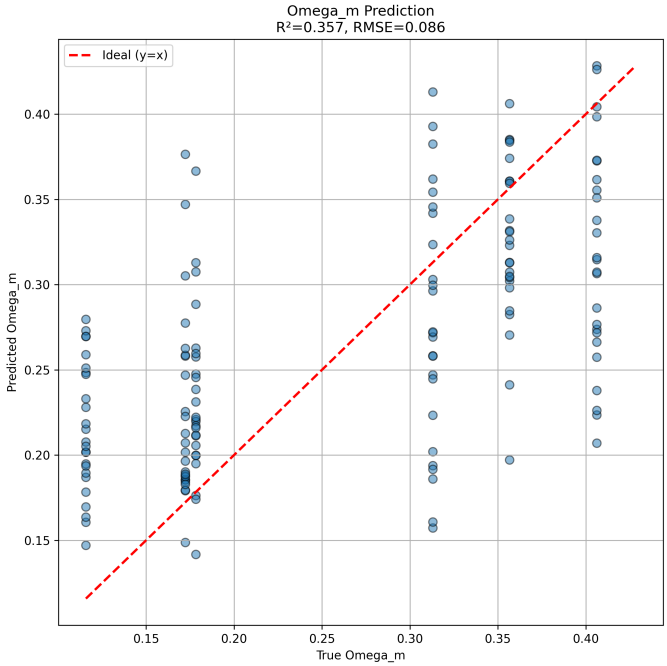

Figure 4. Scatter plot of true versus predicted Ω_m values using TT-compressed motif features, showing a positive correlation with an R^2 score of 0.357 and RMSE of 0.086, which suggests that the frequency and properties of small-scale assembly patterns within merger trees are modulated by the underlying matter density of the universe.

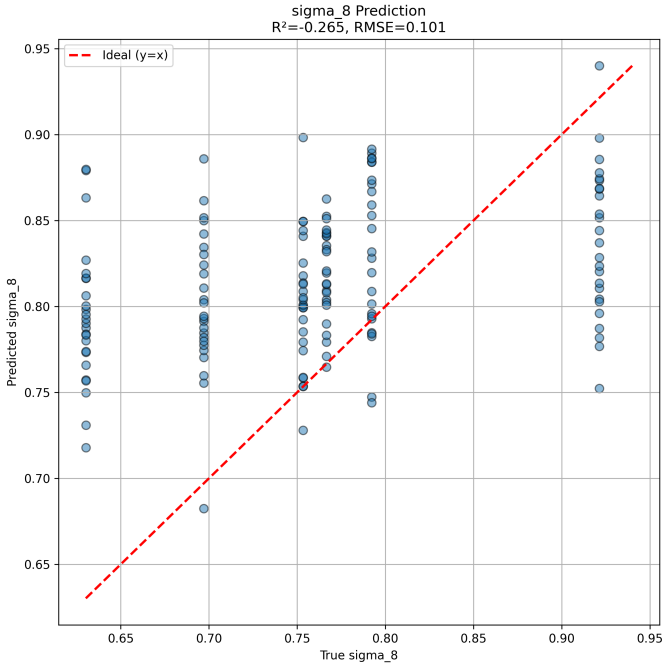

Figure 5. Scatter plot of true vs. predicted σ_8 values using Gradient Boosting Regression on TT-compressed motif features, with a poor R^2 score of -0.2648, indicating the model performs worse than a simple mean predictor and that the selected motifs do not capture information predictive of σ_8.

importance than others, as shown in Figure 6. The first 5 features correspond to the pooled first TT core, and features 5 through 54 correspond to the flattened second TT core.

- For σ_8: Given the poor R^2 score, these feature importances are less meaningful but are provided for completeness and shown in Figure 7.

Directly mapping these importances back to specific physical properties of original motifs (e.g., "mean mass of 3-node chains") is non-trivial due to the nature of TT decomposition and the pooling/flattening process. However, the non-zero importances for Ω_m confirm that the TT representation is being utilized by the model.

4.6. *Summary of Results*

This study explored the use of TT decomposition to compress and analyze merger tree motifs for cosmological parameter inference. Statistically significant chain and fan-in motifs were identified. The TT decomposition achieved excellent reconstruction fidelity, compressing the motif feature matrices with minimal loss of information. The TT-compressed features showed some predictive power for Ω_m ($R^2 \approx 0.36$) but performed poorly for σ_8 ($R^2 \approx$ -0.26). These results suggest that merger

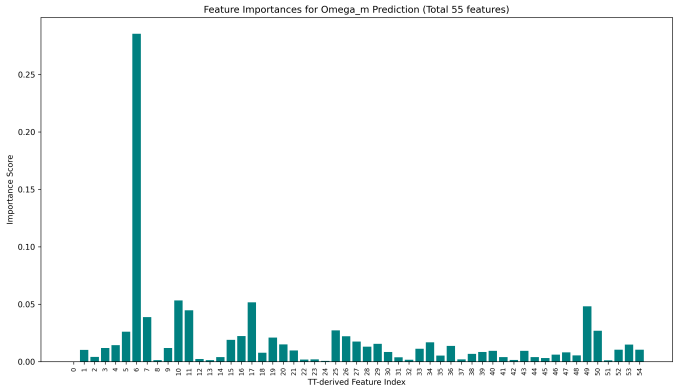

Figure 6. Feature importances for predicting Ω_m using Gradient Boosting Regressor with 55 TT-derived features, showing that a subset of features contributes more to the prediction of this cosmological parameter.

tree motifs are sensitive to the underlying matter density (Ω_m), but the specific motifs and features used in this study were not informative for predicting the amplitude of matter fluctuations (σ_8). The differential sensitivity to these parameters highlights the potential of motif-based analysis for probing different aspects of cosmology.

5. CONCLUSIONS

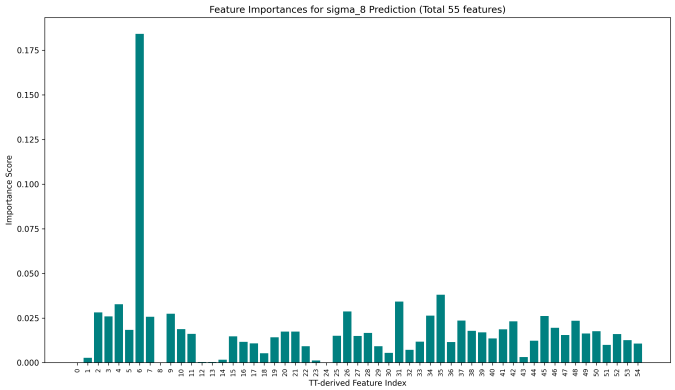

Figure 7. Bar plot showing the feature importances for σ_8 prediction using a gradient boosting regressor trained on 55 TT-derived features, which resulted in a poor R^2 score, indicating that these features do not effectively predict σ_8.

This paper addressed the challenge of extracting cosmological information from the complex structure of dark matter halo merger trees, specifically focusing on the inference of cosmological parameters Ω_m and σ_8. We hypothesized that recurring subgraphs (motifs) within merger trees, representing small-scale assembly patterns, are modulated by the underlying cosmology. To test this hypothesis, we developed a novel approach that combines motif extraction, statistical filtering, feature engineering, Tensor Train (TT) decomposition, and gradient boosting regression.

We analyzed a dataset of 1000 merger trees generated from N-body simulations. After defining and extracting several 3-node and 4-node motifs, we employed a statistical filtering step to identify motifs that occurred significantly more often than expected in random graphs with similar degree sequences. This analysis revealed that 3-node chain, 3-node fan-in, and 4-node chain motifs were statistically significant. For each instance of these motifs, we engineered node-level and motif-level features, which were then used to construct a tensor representation of the merger trees. To reduce the dimensionality of this tensor and extract meaningful patterns, we applied TT decomposition, a technique related to Quantum Tensor Trains. Finally, we trained a gradient boosting regressor to predict cosmological parameters (Ω_m, σ_8) from the TT cores.

Our results demonstrated that the TT-compressed motif features are predictive of Ω_m, achieving an R^2 score of approximately 0.36 on the test set. This suggests that the frequency and properties of these motifs are indeed sensitive to the underlying matter density of the universe. However, the model performed poorly in predicting σ_8 ($R^2 \approx$ -0.26), indicating that the selected motifs and their engineered features were not informative for estimating the amplitude of matter fluctuations. The differential sensitivity of merger tree motifs to these parameters suggests that different aspects of the merger tree structure may encode information about different cosmological parameters.

From this study, we learned that:

- Merger tree motifs can be used to infer cosmological parameters.
- TT decomposition is an effective technique for compressing and analyzing the high-dimensional feature space associated with merger tree motifs.
- The frequency and properties of specific motifs are more sensitive to some cosmological parameters than others.
- The specific motifs and features chosen in this study were more informative for predicting Ω_m than for predicting σ_8.

This work demonstrates the potential of TT decomposition for extracting valuable cosmological information from the intricate structure of dark matter halo merger trees. Future research could explore a wider range of motif types, more sophisticated feature engineering techniques, and alternative machine learning models to further improve the accuracy of cosmological parameter inference from merger trees. Additionally, investigating the reasons for the differential sensitivity to Ω_m and σ_8 could provide valuable insights into the connection between small-scale assembly patterns and the underlying cosmology.