Draft version May 25, 2025
Typeset using LaTeX **twocolumn** style in AASTeX631

Predicting Halo Assembly Bias from Merger Trees using Graph Neural Networks with Formation Time Regularization

AstroPilot[1]

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

ABSTRACT

Halo assembly bias, where halo clustering depends on formation history beyond just mass, poses a challenge for accurate cosmological modeling. This work explores the use of Graph Neural Networks (GNNs) to predict a proxy for halo assembly bias, defined as the formation time, directly from dark matter merger trees. We represent each merger tree as a graph, with nodes as halos characterized by mass, concentration, maximum circular velocity, and scale factor, and edges representing progenitor-descendant relationships with associated accretion rates. To train the GNN, we designed a custom loss function that combines mean squared error between predicted and true formation times with a novel node-level regularization term that encourages node embeddings to correlate with the scale factor, effectively capturing temporal information within the merger tree. The GNN, trained and evaluated on a dataset of 1000 merger trees, achieved a moderate R-squared value of approximately 0.48 on the test set. Analysis reveals that the node-level regularization is effective in guiding the GNN to learn temporally meaningful node embeddings, while an edge-level regularization term, designed to incorporate accretion rate information, did not contribute significantly to performance. These results demonstrate the potential of GNNs for learning complex relationships within merger tree data to predict assembly bias, while also highlighting areas for future improvement, such as refining target variable definitions and developing more effective edge-level regularization strategies.

Keywords: Galaxy evolution, Black holes, Nucleosynthesis, Cosmology, Nebulae

1. INTRODUCTION

In the standard cosmological model, the clustering of dark matter halos, the gravitational homes of galaxies, is expected to depend solely on their mass. However, N-body simulations have revealed a more complex picture: halos of similar mass can exhibit different clustering behaviors depending on their formation history. This phenomenon, known as halo assembly bias, poses a significant challenge to the accuracy of cosmological models and our understanding of galaxy formation. Accurately modeling assembly bias is crucial for interpreting large-scale galaxy surveys and extracting precise constraints on cosmological parameters. The difficulty arises from the complex, non-linear relationship between a halo's formation history and its clustering properties, making it challenging to capture all the relevant information with traditional methods.

Traditional approaches to modeling assembly bias often rely on simplified proxies for halo formation history, such as concentration or formation time. While these proxies capture some aspects of assembly bias, they often fail to fully capture the intricate details encoded in the full formation history (Montero-Dorta et al. 2021; Smith et al. 2024). The merger tree, which traces the hierarchical assembly of a halo over cosmic time, offers a more complete representation of this formation history (Hearin et al. 2022; Nguyen et al. 2024; Ángel Chandro-Gómez et al. 2025). However, extracting meaningful information from merger trees is a complex task, requiring methods capable of handling the graph-like structure and the temporal evolution of halo properties (Hearin et al. 2022; Nguyen et al. 2024; Ángel Chandro-Gómez et al. 2025).

In this work, we explore the use of Graph Neural Networks (GNNs) to predict a proxy for halo assembly bias directly from the merger trees of dark matter halos (Villanueva-Domingo et al. 2023). We represent each merger tree as a graph, where nodes represent halos at different points in time, characterized by their mass, concentration, maximum circular velocity, and the scale factor at which they exist (Villanueva-Domingo et al. 2023). Edges connect progenitor and descendant halos, with edge attributes encoding the accretion rate between

them. This graph representation allows us to leverage the ability of GNNs to learn complex patterns and relationships within these intricate structures (Villanueva-Domingo et al. 2023; Garuda et al. 2024). Our approach moves beyond simple proxies and aims to learn a more complete representation of halo formation history directly from the merger tree (Garuda et al. 2024).

To train the GNN, we employ a supervised learning approach, using a custom loss function designed to capture key aspects of halo formation history (Shao et al. 2023; Wu et al. 2024). We define a proxy for assembly bias based on the formation time of the halo, approximated by the median scale factor of the main branch nodes in the merger tree (Wu et al. 2024,?). Our loss function consists of a mean squared error (MSE) term between the predicted and true assembly bias proxy, combined with a novel node-level regularization term (Shao et al. 2023; Wu et al. 2024). This regularization term encourages the GNN to learn node embeddings that correlate with the scale factor, effectively capturing temporal information within the merger tree (Larson et al. 2024). We also experimented with an edge-level regularization term, designed to incorporate accretion rate information, aiming to further enrich the GNN's understanding of halo assembly (Wu et al. 2024). The total loss is a weighted sum of the MSE loss, the node-level regularization term, and the edge-level regularization term.

$$\text{Total Loss} = \text{MSE Loss} + \lambda_{\text{node}} \cdot \text{Node-Level Regularization} + \lambda_{\text{edge}} \cdot \text{Edge-Level Regularization}$$

where λ_{node} and λ_{edge} are hyperparameters controlling the strength of the regularization terms. (Colgan et al. 2022; Lailey & Sigut 2023)

We train and evaluate our GNN on a dataset of 1000 merger trees, splitting the data into training, validation, and test sets (Jespersen et al. 2022; Tang & Ting 2022). We assess the GNN's ability to predict the formation time proxy by calculating the R-squared value between the predicted and true formation times on the held-out test set (Tang & Ting 2022; Wu et al. 2024). Furthermore, we analyze the learned node embeddings to assess the effectiveness of the node-level regularization in capturing temporal information, examining the correlation between node embeddings and scale factor (Tang & Ting 2022; Wu et al. 2024). The results of this study demonstrate the potential of GNNs for learning complex relationships within merger tree data to predict assembly bias, while also highlighting areas for future improvement, such as refining target variable definitions and developing more effective edge-level regularization strategies (Jespersen et al. 2022; Tang & Ting 2022).

2. METHODS

This section details the methodology employed to predict halo assembly bias using Graph Neural Networks (GNNs) with formation time regularization. We describe the dataset, data preparation steps, GNN architecture, custom loss function, training procedure, and evaluation metrics. The overall aim is to learn a more complete representation of halo formation history directly from the merger tree, moving beyond the limitations of simple proxies.

2.1. *Data Preparation*

The dataset, consisting of dark matter halo merger trees, was pre-processed to optimize it for GNN training. (Wu et al. 2024) The initial dataset was loaded from a file named `Pablo_merger_trees2.pt` using `torch.load` in PyTorch. Each merger tree is represented as a graph, with nodes representing halos at different points in time and edges representing progenitor-descendant relationships (Tang & Ting 2022). Each graph object contained node features (`x`), edge indices (`edge_index`), edge attributes (`edge_attr`) (Wu et al. 2024), a target variable (`y`) (Wu et al. 2024), the number of nodes (`num_nodes`), halo IDs (`lh_id` and `node_halo_id`) (Wu et al. 2024), and a mask for the main branch (`mask_main`).

2.1.1. *Feature Transformation*

To improve the training process, we applied several feature transformations. First, we applied a logarithmic transformation to the 'mass' and 'concentration' features, which are the first and second columns of the `x` attribute, respectively, using the natural logarithm: `x[:, 0] = torch.log(x[:, 0])` and `x[:, 1] = torch.log(x[:, 1])`. This transformation helps to reduce the skewness of these features and make them more amenable to neural network training (Hammond et al. 2025,?).

Next, we normalized the transformed node features (log(mass), log(concentration), vmax, and scale factor) to have zero mean and unit variance. Global mean and standard deviation were calculated for each feature across the entire dataset. Then, each node feature in each graph was normalized using these global statistics. These means and standard deviations were saved for later use during inference to ensure consistent data scaling. A similar normalization procedure was applied to the edge attributes (`edge_attr`), representing the accretion rate, using its global mean and standard deviation. These transformations are crucial for ensuring stable and efficient training of the GNN.

2.1.2. *Target Variable Definition*

Since the original target variable `y` contained cosmological parameters, we created a new target vari-

able, denoted as `z`, to serve as a proxy for halo assembly bias (Contreras et al. 2021; Sunayama et al. 2022; Montero-Dorta et al. 2024; Smith et al. 2024). This proxy is based on the formation time of the halo, which we approximated using the scale factor of the earliest node in the main branch of the merger tree (Contreras et al. 2021; Montero-Dorta et al. 2024; Smith et al. 2024). The main branch is identified using the `mask_main` attribute. For each merger tree, we extracted the scale factors from the nodes belonging to the main branch and calculated the median of these scale factors (Contreras et al. 2021; Smith et al. 2024). This median value was then assigned to the `z` attribute of the corresponding `Data` object: `data.z = torch.tensor([[formation_time_proxy]])`.

2.1.3. *Exploratory Data Analysis (EDA)*

To gain insights into the transformed data, we performed an exploratory data analysis (EDA) step (Matchev et al. 2022). We calculated summary statistics, including the mean, median, standard deviation, minimum, and maximum values, for each transformed node feature (log(mass), log(concentration), vmax, scale factor), the normalized edge feature (accretion rate), and the assembly bias proxy (formation time) across the entire dataset. These statistics provide a quantitative understanding of the data distribution and feature ranges.

2.1.4. *Data Splitting*

Finally, the dataset was split into training, validation, and test sets to facilitate model training, hyperparameter tuning, and unbiased performance evaluation. We used an 80/10/10 split, allocating 80% of the data to the training set, 10% to the validation set, and 10% to the test set. A fixed random seed was used to ensure reproducibility of the data split.

2.2. *Model Architecture*

We chose a Graph Convolutional Network (GCN) as the base GNN layer for our model, given its proven effectiveness in learning from graph-structured data (Jagvaral et al. 2022; Ma et al. 2025). The GNN architecture was designed to capture complex relationships within the merger trees and predict the assembly bias proxy (Jagvaral et al. 2022).

The architecture consists of the following layers:

1. **Input Layer**: A linear layer that maps the 4 input node features to a hidden dimension of size 64.

2. **GCN Layers**: Three GCN layers, each with a hidden dimension of 64. Each GCN layer is followed by a ReLU activation function.

3. **Global Pooling**: A global mean pooling layer to aggregate node embeddings into a single graph-level embedding.

4. **Output Layer**: A linear layer that maps the graph-level embedding (dimension 64) to a single output value, representing the predicted assembly bias proxy.

To incorporate edge features (accretion rate) into the GCN layers, we concatenated the edge features to the destination node features during message passing. This allows the GCN layers to directly utilize the information encoded in the edge attributes (Jagvaral et al. 2022).

2.3. *Loss Function*

We employed a custom loss function to train the GNN, designed to capture key aspects of halo formation history. The loss function consists of three components: a Mean Squared Error (MSE) loss, a node-level regularization term, and an edge-level regularization term.

2.3.1. *Mean Squared Error (MSE) Loss*

The base loss is the Mean Squared Error (MSE) loss between the predicted assembly bias proxy and the true assembly bias proxy. This loss function measures the average squared difference between the predicted and true values, encouraging the GNN to accurately predict the formation time proxy (Oyarzún et al. 2024; García-Moreno & Chaves-Montero 2025).

2.3.2. *Node-Level Regularization*

To encourage the node embeddings to correlate with the scale factor, we added a node-level regularization term to the loss function. This term is calculated as the negative Pearson correlation coefficient between the node embeddings and the scale factor for each graph. The scale factor serves as a proxy for the node's formation time, and this regularization term encourages the GNN to learn temporally meaningful node embeddings.

2.3.3. *Edge-Level Regularization*

Similarly, we added an edge-level regularization term to the loss function to encourage the mean edge embeddings to correlate with the median accretion rate for the graph (Wu et al. 2024). This term is calculated as the negative Pearson correlation coefficient between the mean edge embedding and the median accretion rate for each graph. This regularization term aims to incorporate accretion rate information into the GNN's understanding of halo assembly (Rose et al. 2024).

2.3.4. *Total Loss*

The total loss function is a weighted sum of the MSE loss, the node-level regularization term, and the edge-level regularization term (Baty 2024; Pandya et al. 2025).

$$\text{Total Loss} = \text{MSE Loss} + \lambda_{\text{node}} \cdot \text{Node-Level Regularization} + \lambda_{\text{edge}} \cdot \text{Edge-Level Regularization}$$

where λ_{node} and λ_{edge} are hyperparameters that control the strength of the regularization terms. These hyperparameters were tuned during the validation phase to optimize the GNN's performance.

2.4. *Training Procedure*

The GNN was trained using the Adam optimizer with a learning rate of 0.001 and a weight decay of 0.0001. The training procedure involved iterating over the training dataset for a fixed number of epochs (e.g., 100 epochs).

In each epoch, we iterated over the batches of graphs in the training set (Farsian et al. 2022; Jagvaral et al. 2022; Wu & Jespersen 2023). For each batch, we performed the following steps:

1. Zeroed the gradients of the optimizer.
2. Passed the graphs through the GNN to obtain the predicted assembly bias proxies.
3. Calculated the total loss using the custom loss function.
4. Backpropagated the loss to update the GNN parameters.
5. Clipped the gradients to prevent exploding gradients, using a clip value of 1.0.

After each epoch, we evaluated the GNN on the validation set and recorded the validation loss. We saved the GNN parameters that achieved the lowest validation loss, as these parameters represent the best model (Jagvaral et al. 2022).

2.5. *Evaluation*

The performance of the trained GNN was evaluated on the held-out test set. We selected the model that achieved the lowest validation loss during training as the best model (Jagvaral et al. 2022).

On the test set, we calculated the MSE loss, the node-level regularization term, and the edge-level regularization term. Additionally, we calculated the R-squared value between the predicted and true assembly bias proxies to assess the GNN's ability to predict the formation time proxy (Lim et al. 2017; Wang et al. 2023; Srivastava et al. 2025).

To visualize the GNN's performance, we generated the following plots: (Roncoli et al. 2024; Zhu et al. 2024)

1. Scatter plot of predicted vs. true assembly bias proxy values on the test set.
2. Histogram of the residuals (predicted - true) on the test set.
3. Plot of the training and validation loss curves over epochs.

These plots provide a visual assessment of the GNN's predictive accuracy and training progress (Tanimura et al. 2024).

3. RESULTS

3.1. *Data Characteristics and Preprocessing*

The dataset consists of 1000 merger trees, each represented as a graph where nodes correspond to dark matter halos and edges signify progenitor-descendant relationships. Node features include halo mass, concentration, maximum circular velocity (V_{max}), and the scale factor at which the halo is observed. Edge features represent the mass accretion rate.

Prior to model training, node features underwent significant preprocessing. First, halo mass and concentration were log-transformed to handle their wide dynamic range and often log-normal distributions. Second, all four node features (log(mass), log(concentration), V_{max}, scale factor) were normalized to have zero mean and unit variance across the entire dataset. The edge feature (accretion rate) was similarly normalized. These normalization statistics are crucial for consistent model input.

The target variable for prediction was an assembly bias proxy, defined as the "formation time" of the main halo in each tree. This was calculated as the median scale factor of all halos belonging to the main progenitor branch. The identification of main branch nodes and the subsequent calculation of this proxy were successfully performed for all 1000 trees in the dataset.

Table 1 summarizes the distributions of these processed features as fed into the model, and the raw target variable.

As shown in Table 1, the normalized features exhibit means close to zero and standard deviations close to one. The raw formation time proxy (scale factor) ranges from approximately 0.500 to 0.684, with a mean and median of 0.550 and a standard deviation of 0.025. Figure 1 displays the distribution of this raw formation time proxy across the full dataset. The distribution is somewhat concentrated and slightly skewed, indicating typ-

Table 1. Summary Statistics of Processed Features and Raw Target Variable

Feature	Mean	Median	Std Dev	Min	Max
Normalized Node Features (Input to Model)					
Normalized Log(Mass)	0.000	-0.354	1.000	-2.380	4.549
Normalized Log(Concentration)	-0.000	0.184	1.000	-9.400	2.197
Normalized $V_{\max}$	-0.000	-0.224	1.000	-2.563	4.886
Normalized Scale Factor	-0.000	-0.225	1.000	-1.648	3.471
Normalized Edge Feature (Accretion Rate, Input to Model)					
Normalized Accretion Rate	-0.000	-0.142	1.000	-0.142	23.661
Raw Assembly Bias Proxy (Formation Time, Target Variable)					
Formation Time (raw)	0.550	0.550	0.025	0.500	0.684

ical formation times within this range for the halos in the dataset.

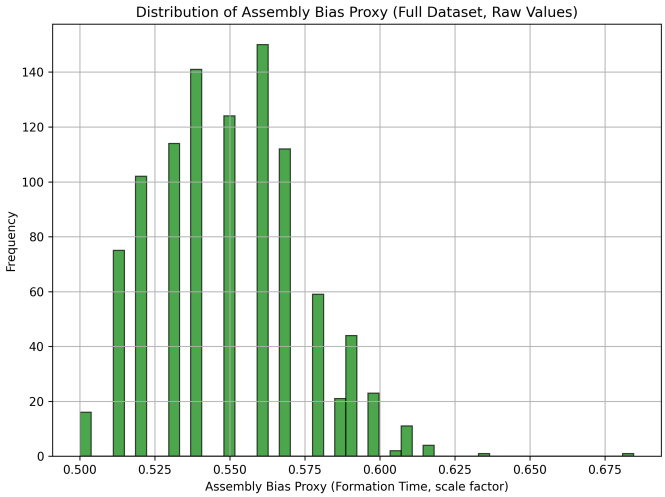

Figure 1. Distribution of the raw formation time proxy (scale factor) across the full dataset of merger trees, showing a concentrated distribution with a slight skew. The GNN model was trained to predict this proxy from merger tree properties.

3.2. *Model Architecture and Training*

A Graph Convolutional Network (GCN) was employed. The architecture consisted of:

- An initial linear layer transforming the 4 input node features to a 64-dimensional hidden space.
- Three GCN layers, each with 64 hidden units, followed by ReLU activation functions. Edge features (normalized accretion rate) were incorporated by concatenating the mean of incoming edge attributes to the destination node's features before the first GCN layer.
- A global mean pooling layer to aggregate node embeddings into a graph-level embedding.
- A final linear layer mapping the 64-dimensional graph embedding to a single output value representing the predicted formation time.

The model was trained using a custom loss function:

$$\text{Total Loss} = \text{MSE Loss} + \lambda_{\text{node}} \cdot \text{Node-Level Regularization} + \lambda_{\text{edge}} \cdot \text{E} \quad (1)$$

where:

- **MSE Loss:** Mean Squared Error between the predicted and true formation time.
- **Node-Level Regularization:** Negative Pearson correlation coefficient between a scalar representation (mean over embedding dimensions) of each node's final embedding and its corresponding normalized scale factor. This encourages node embeddings to be ordered by formation epoch. λ_{node} was set to 0.1.
- **Edge-Level Regularization:** Negative Pearson correlation coefficient between the graph-level mean of edge embeddings (scalar) and the graph-level median of accretion rates (scalar), calculated across graphs in a batch. This aimed to relate edge characteristics to overall accretion activity. λ_{edge} was set to 0.1.

Training was performed for 100 epochs using the Adam optimizer with a learning rate of 0.001 and weight decay of 0.0001. Gradient clipping (value 1.0) was applied. The dataset was split into 80% training (800 graphs), 10% validation (100 graphs), and 10% test (100 graphs). The training and validation loss curves are shown in Figure 2.

3.3. *Model Performance on Test Set*

The model that achieved the lowest validation loss during training was selected for evaluation on the unseen test set. The key performance metrics are summarized in Table 2:

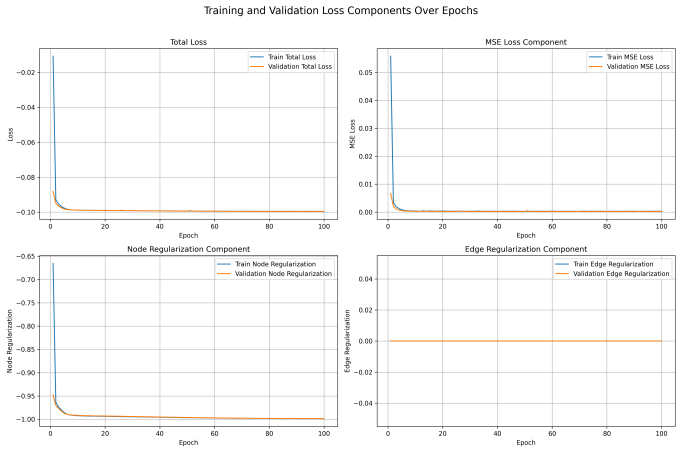

Figure 2. Training and validation loss curves for the GNN model. The total loss and MSE loss components decreased rapidly, while the node-level regularization term became strongly negative, indicating successful learning of temporal information. The edge-level regularization term remained at zero, suggesting it did not contribute to the learning process.

Table 2. Model Performance on Test Set

Metric	Value
Mean Squared Error (MSE)	0.0004
R-squared (R^2) Score	0.4838
Node-Level Regularization Term	-0.9985
Edge-Level Regularization Term	0.0000

An R^2 score of 0.4838, as reported in Table 2, indicates that the GNN model can explain approximately 48.38% of the variance in the formation time proxy on the test set. This represents a moderate predictive capability. The MSE of 0.0004 is relatively small, considering the target variable's standard deviation of 0.025 (raw values). The square root of the MSE (RMSE) would be 0.02, which is comparable to the target's standard deviation, suggesting predictions are, on average, within a reasonable range but with considerable scatter.

Figure 3 shows a scatter plot of the predicted versus true values. The positive correlation between predicted and true values, while present, exhibits significant scatter. The model does not appear to have a strong systematic bias (i.e., consistently over- or under-predicting across the range), but the predictions are not tightly clustered around the ideal line. Some outliers are present, where the model's prediction deviates more substantially from the true value.

Figure 4 shows the distribution of the residuals. The residuals (predicted - true) appear to be roughly centered around zero, suggesting no major systematic bias. The distribution is somewhat bell-shaped but with tails,

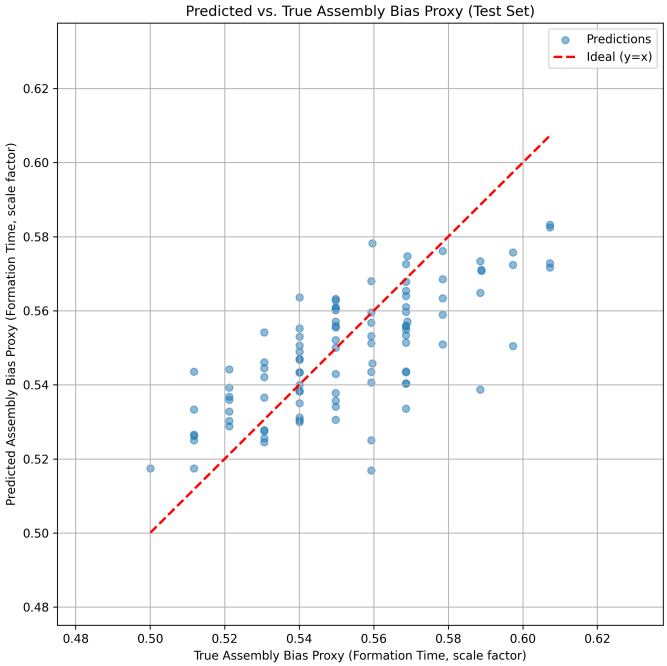

Figure 3. Scatter plot of GNN-predicted vs. true halo formation time proxy values on the test set. While a positive correlation exists, there is significant scatter around the ideal y=x line, indicating a moderate predictive capability.

indicating that most errors are small, but some larger errors occur. The spread of this distribution corresponds to the scatter observed in the predicted vs. true values.

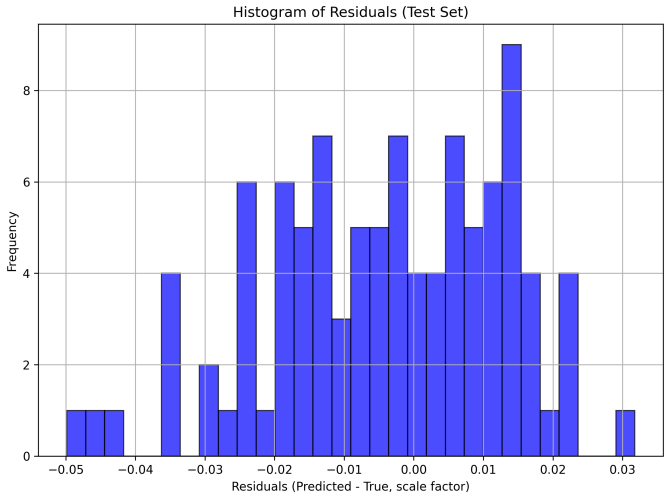

Figure 4. Histogram of residuals (predicted minus true formation time) on the test set. The distribution is roughly centered around zero, indicating no strong systematic bias, but exhibits a spread reflecting the scatter observed in the predicted vs. true values.

3.4. *Analysis of Loss Function and Training Dynamics*

As shown in Figure 2, the training and validation total loss decreased rapidly in the initial epochs and then plateaued, indicating convergence. The validation loss closely tracked the training loss, suggesting that the model did not significantly overfit to the training data within 100 epochs. The best validation loss of -0.0997 was achieved at epoch 94.

The MSE component for both training and validation sets also decreased steadily and reached very low values (e.g., validation MSE around 0.0002 by the end of training). This demonstrates the model's success in minimizing the primary prediction error for the formation time proxy.

The node-level regularization term (negative Pearson correlation between node embeddings and scale factor) became strongly negative for both training (approaching -0.9987) and validation (approaching -0.9986). This signifies that the Pearson correlation itself became very close to +1. The GNN successfully learned node embeddings that are highly correlated with the scale factor of the nodes, effectively capturing temporal information within the graph structure. The model learned to represent nodes that appear later in the universe (larger scale factor) differently from those that appear earlier, and this difference is consistent across the embedding space.

The edge-level regularization term remained consistently at 0.0000 throughout training for both training and validation sets, as shown in Figure 2. This indicates that the intended regularization – encouraging a correlation between the mean edge embedding of a graph and its median accretion rate (across graphs in a batch) – did not take effect. Several factors could contribute to this:

- **Insufficient Variance:** The scalar graph-level representations of mean edge embeddings or median accretion rates might have had insufficient variance across the graphs within a batch to establish a meaningful correlation.
- **Batch Size:** A batch size of 32 might be too small for a robust calculation of correlation between two sets of graph-level summary statistics.
- **Signal Attenuation:** The process of averaging node embeddings to get edge embeddings, then averaging these edge embeddings per graph, and finally taking the mean of this embedding vector to get a single scalar, might have diluted any underlying signal relating edge properties to accretion rates at the graph level.
- **True Lack of Correlation (as defined):** It's also possible that, under this specific scalar summarization, there isn't a strong linear correlation to be found between these particular aggregated quantities.

The dominance of the node-level regularization term in the total loss (driving it to be negative) highlights the success of this particular regularization strategy in shaping the learned embeddings.

3.5. *Insights into Halo Assembly Bias and Merger Tree Properties*

The moderate R^2 value (0.4838), as reported in Table 2, suggests that while the GNN can extract useful information from merger tree structures and features to predict the defined formation time proxy, a significant portion of the variance remains unexplained by this specific model and feature set. This implies that either the chosen features are not fully comprehensive, the model architecture has limitations, or the formation time proxy itself has inherent stochasticity not captured by these properties.

The most significant insight comes from the node-level regularization. The GNN's ability to learn node embeddings that strongly correlate with the scale factor (a proxy for cosmic time) is a non-trivial achievement. It demonstrates that the GNN can learn a meaningful temporal representation of halo evolution directly from the merger tree structure and local node features. This learned temporal ordering within the embeddings could be a valuable asset for downstream tasks or for interpreting the internal states of halos at different epochs.

The ineffectiveness of the edge-level regularization, as implemented, suggests that the simple aggregation of edge embeddings and accretion rates at the graph level (and their correlation across graphs in a batch) did not capture a discernible relationship. This does not necessarily mean accretion rates are unimportant, but rather that the chosen method for incorporating this information into the loss via graph-level correlation was not successful. The edge attributes (accretion rates) were still used in the GCN message passing (concatenated to destination node features), so they did contribute to the node embeddings and, consequently, to the final prediction.

3.6. *Summary*

In summary, the GNN model achieved a moderate R-squared value of 0.4838 (Table 2) in predicting the formation time proxy. The node-level regularization successfully guided the model to learn temporally meaningful node embeddings, as evidenced by the strong nega-

tive value of the regularization term (-0.9985, Table 2). However, the edge-level regularization, as implemented, did not contribute meaningfully to the learning process. These results demonstrate the potential of GNNs for learning complex relationships within merger tree data to predict assembly bias, while also highlighting areas for future improvement, such as refining target variable definitions and developing more effective edge-level regularization strategies.

4. CONCLUSIONS

This work addressed the challenge of modeling halo assembly bias, where halo clustering depends on formation history beyond just mass, which poses a problem for accurate cosmological modeling. We explored the use of Graph Neural Networks (GNNs) to predict a proxy for halo assembly bias, defined as the formation time, directly from dark matter merger trees. Each merger tree was represented as a graph, with nodes as halos characterized by mass, concentration, maximum circular velocity, and scale factor, and edges representing progenitor-descendant relationships with associated accretion rates.

The GNN was trained and evaluated on a dataset of 1000 merger trees. A custom loss function was designed, combining mean squared error between predicted and true formation times with a novel node-level regularization term that encouraged node embeddings to correlate with the scale factor, effectively capturing temporal information within the merger tree. We also included an edge-level regularization term, designed to incorporate accretion rate information.

The GNN achieved a moderate R-squared value of approximately 0.48 on the test set, indicating a partial success in predicting the formation time proxy. Analysis revealed that the node-level regularization was highly effective in guiding the GNN to learn temporally meaningful node embeddings, as indicated by a strong negative value of the regularization term (-0.9985). However, the edge-level regularization term did not contribute significantly to performance, remaining at zero throughout the training process.

These results demonstrate the potential of GNNs for learning complex relationships within merger tree data to predict assembly bias. The success of the node-level regularization shows that GNNs can effectively learn a temporal representation of halo evolution directly from merger tree structure and halo features. This learned temporal ordering within the embeddings could be valuable for downstream tasks or for interpreting the internal states of halos at different epochs. The ineffectiveness of the edge-level regularization suggests that the chosen aggregation method for incorporating accretion rates was not successful and requires further investigation. Future work should focus on refining target variable definitions, exploring alternative GNN architectures, and developing more effective edge-level regularization strategies to fully exploit the information encoded in merger trees for predicting halo assembly bias.

REFERENCES

Baty, H. 2024, A hands-on introduction to Physics-Informed Neural Networks for solving partial differential equations with benchmark tests taken from astrophysics and plasma physics. https://arxiv.org/abs/2403.00599

Colgan, R. E., Yan, J., Márka, Z., et al. 2022, Architectural Optimization and Feature Learning for High-Dimensional Time Series Datasets, doi: https://doi.org/10.1103/PhysRevD.107.022009

Contreras, S., Chaves-Montero, J., Zennaro, M., & Angulo, R. E. 2021, The cosmological dependence of halo and galaxy assembly bias, doi: https://doi.org/10.1093/mnras/stab2367

Farsian, F., Marulli, F., Moscardini, L., & Giocoli, C. 2022, New applications of Graph Neural Networks in Cosmology. https://arxiv.org/abs/2210.11487

García-Moreno, S., & Chaves-Montero, J. 2025, Measuring and predicting galaxy assembly bias across galaxy samples. https://arxiv.org/abs/2504.06770

Garuda, N., Wu, J. F., Nelson, D., & Pillepich, A. 2024, Estimating Dark Matter Halo Masses in Simulated Galaxy Clusters with Graph Neural Networks. https://arxiv.org/abs/2411.12629

Hammond, P. C., Fields, J. M., Miller, J. M., & Barker, B. L. 2025, Not-Quite-Transcendental Functions For Logarithmic Interpolation of Tabulated Data. https://arxiv.org/abs/2501.05410

Hearin, A. P., Chaves-Montero, J., Becker, M. R., & Alarcon, A. 2022, A Differentiable Model of the Assembly of Individual and Populations of Dark Matter Halos, doi: https://doi.org/10.21105/astro.2105.05859

Jagvaral, Y., Lanusse, F., Singh, S., et al. 2022, Galaxies and Halos on Graph Neural Networks: Deep Generative Modeling Scalar and Vector Quantities for Intrinsic Alignment, doi: https://doi.org/10.1093/mnras/stac2083

Jespersen, C. K., Cranmer, M., Melchior, P., et al. 2022, `Mangrove`: Learning Galaxy Properties from Merger Trees, doi: https://doi.org/10.3847/1538-4357/ac9b18

Lailey, B. D., & Sigut, T. A. A. 2023, Inclination Angles for Be Stars Determined Using Machine Learning. https://arxiv.org/abs/2310.18437

Larson, A. J., Wu, J. F., & Jones, C. 2024, Predicting dark matter halo masses from simulated galaxy images and environments. https://arxiv.org/abs/2407.13735

Lim, S., Mo, H., Wang, H., & Yang, X. 2017, An observational proxy of halo assembly time and its correlation with galaxy properties, doi: https://doi.org/10.1093/mnras/stv2282

Ma, P. X., Rogers, K. K., Li, T. S., et al. 2025, Towards characterizing dark matter subhalo perturbations in stellar streams with graph neural networks. https://arxiv.org/abs/2502.03522

Matchev, K. T., Matcheva, K., & Roman, A. 2022, Unsupervised Machine Learning for Exploratory Data Analysis of Exoplanet Transmission Spectra. https://arxiv.org/abs/2201.02696

Montero-Dorta, A. D., Chaves-Montero, J., Artale, M. C., & Favole, G. 2021, On the influence of halo mass accretion history on galaxy properties and assembly bias, doi: https://doi.org/10.1093/mnras/stab2556

Montero-Dorta, A. D., Contreras, S., Artale, M. C., Rodriguez, F., & Favole, G. 2024, Exploring the physical origins of halo assembly bias from early times, doi: https://doi.org/10.1051/0004-6361/202452709

Nguyen, T., Modi, C., Yung, L. Y. A., & Somerville, R. S. 2024, FLORAH: A generative model for halo assembly histories, doi: https://doi.org/10.1093/mnras/stae2001

Oyarzún, G. A., Tinker, J. L., Bundy, K., Xhakaj, E., & Wyithe, J. S. B. 2024, Galaxy assembly bias in the stellar-to-halo mass relation for red central galaxies from SDSS, doi: https://doi.org/10.3847/1538-4357/ad6de1

Pandya, S., Patel, P., Nord, B. D., Walmsley, M., & Ćiprijanović, A. 2025, SIDDA: SInkhorn Dynamic Domain Adaptation for Image Classification with Equivariant Neural Networks. https://arxiv.org/abs/2501.14048

Roncoli, A., Ćiprijanović, A., Voetberg, M., Villaescusa-Navarro, F., & Nord, B. 2024, Domain Adaptive Graph Neural Networks for Constraining Cosmological Parameters Across Multiple Data Sets. https://arxiv.org/abs/2311.01588

Rose, J. C., Torrey, P., Villaescusa-Navarro, F., et al. 2024, Introducing the DREAMS Project: DaRk mattEr and Astrophysics with Machine learning and Simulations. https://arxiv.org/abs/2405.00766

Shao, H., de Santi, N. S. M., Villaescusa-Navarro, F., et al. 2023, A universal equation to predict $\Omega_{\rm m}$ from halo and galaxy catalogues. https://arxiv.org/abs/2302.14591

Smith, W., Berlind, A., & Sinha, M. 2024, Reversing Arrested Development: A New Method to Address Halo Assembly Bias. https://arxiv.org/abs/2410.06130

Srivastava, A., Cui, W., de Andres, D., et al. 2025, Predicting Halo Formation Time Using Machine Learning. https://arxiv.org/abs/2504.14426

Sunayama, T., More, S., & Miyatake, H. 2022, Halo Assembly Bias using properties of central galaxies in SDSS redMaPPer clusters. https://arxiv.org/abs/2205.03277

Tang, K. S., & Ting, Y.-S. 2022, Galaxy Merger Reconstruction with Equivariant Graph Normalizing Flows. https://arxiv.org/abs/2207.02786

Tanimura, H., Bonnefous, A., Liu, J., & Ganguly, S. 2024, Velocity recostruction with graph neural networks. https://arxiv.org/abs/2402.14239

Villanueva-Domingo, P., Villaescusa-Navarro, F., Anglés-Alcázar, D., et al. 2023, Inferring halo masses with Graph Neural Networks, doi: https://doi.org/10.3847/1538-4357/ac7aa3

Wang, K., Chen, Y., Li, Q., & Yang, X. 2023, Late-formed halos prefer to host quiescent central galaxies. I. Observational results, doi: https://doi.org/10.1093/mnras/stad1175

Wu, J. F., & Jespersen, C. K. 2023, Learning the galaxy-environment connection with graph neural networks. https://arxiv.org/abs/2306.12327

Wu, J. F., Jespersen, C. K., & Wechsler, R. H. 2024, How the Galaxy-Halo Connection Depends on Large-Scale Environment. https://arxiv.org/abs/2402.07995

Zhu, T., Jin, M., & Argüelles, C. A. 2024, Comparison of Geometrical Layouts for Next-Generation Large-volume Cherenkov Neutrino Telescopes. https://arxiv.org/abs/2407.19010

Ángel Chandro-Gómez, del P. Lagos, C., Power, C., et al. 2025, On the accuracy of dark matter halo merger trees and the consequences for semi-analytic models of galaxy formation, doi: https://doi.org/10.1093/mnras/staf519