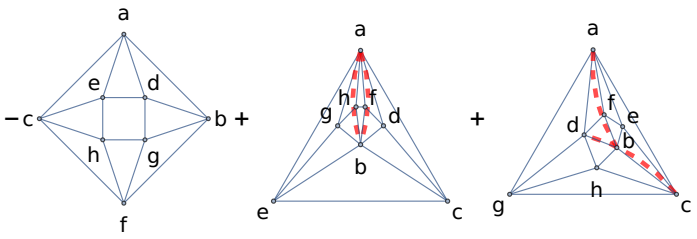
Predicting Vanishing Coefficients of f-Graphs with Machine Learning

1 Setup and Motivation

We consider the set of graph called f-graphs contributing to the four-point correlator at fixed loop order in planar $\mathcal{N} = 4$ supersymmetric Yang–Mills. Each f-graph is a has two types of edges which are graphically represented with solid and dashed lines. F-graphs are graphs with the following properties

- The solid lines form a planar graph.
- The solid lines cannot have multiple edges connecting a pair of vertices.
- For each vertex the number of solid edges minus the number of dashed edges is equal to 4.

The n-point correlator is given by a linear combination with rational coefficients of f-graphs. For example the 8-point correlator is given by



The space of f-graphs at 6 and 7 points is one dimensional. From 8-point the space is non-trivial and at 9-point the first f-graphs with a zero coefficient start appearing. At 14-point graphs with zero coefficient become the majority and this percentage keeps growing with the number of points. Our goal is to train machine learning models capable of predicting whether the coefficient associated with a given f-graph is zero or non-zero.

f-graph			
Loop	**Total**	**#1**	**#0**
7	220	127	93
8	2,709	1,060	1,649
9	43,017	10,525	32,492
10	153,252	51,263	101,989

Table 1: Summary of f-graph data across loops.

Solid graph			
Loop	**Total**	**#1**	**#0**
7	164	107	57
8	1,432	729	703
9	13,972	5,661	8,311
10	153,252	51,263	101,989

Table 2: Summary of solid graph data across loops.

2 Random forest for f-graphs

For each f-graph, we construct two main sets of features:

- **Solid line spectrum:** We compute the eigenvalue spectrum of the adjacency matrix associated with the subgraph defined by the solid lines. This captures global structural information about the propagator topology.
- **Difference spectrum:** Since different f-graphs can share the same denominator graph but still have different coefficients, we introduce a second matrix: the difference between the adjacency matrices of the solid line and the dashed lines. We compute the spectrum of this matrix as well.

Our classification target is the column `COEFFICIENT_DEN`, which takes binary values: 1 if the coefficient is non-zero, and 0 otherwise.

We evaluate each model using an 80/20 train-test split. That is, 80% of the full dataset is used for training, and the remaining 20% is reserved for testing the model's ability to classify unseen examples.

We also perform initial experiments on the full dataset, as our goal is to determine whether a perfect classifier exists. In such a context, overfitting

is not a concern: we are not interested in generalization to unseen examples, but in whether a model can in principle learn the structure of the data.

Loop	Accuracy	Class 0 (Prec/Rec)	Class 1 (Prec/Rec)	Perfect Class.
7	0.89	0.89 / 0.84	0.88 / 0.92	yes
8	0.83	0.88 / 0.85	0.77 / 0.82	yes
9	0.87	0.89 / 0.94	0.78 / 0.65	yes
10	0.89	0.91 / 0.97	0.76 / 0.45	no

Table 3: Random Forest results for loops 7 to 10 using 0.8 training set and corresponding full set accuracies.

At 10-loop a perfect classification is not possible. The misclassified graphs indices are: [4724, 7075, 19810, 281029, 290977, 291224, 291226, 672010, 824379]

There are 3308 cospectral graphs. 4 out of 9 of the mismatches comes from the co-spectral graphs that have different coefficients, there are

$$(4656, 4724), (290977, 290994), (672010, 672028), (280495, 281029)$$

3 Models and Methodology

We experiment with a range of classification models, including:

- Logistic Regression
- Support Vector Machines (SVM)
- Decision Trees
- Random Forests

Our classification target is the column `COEFFICIENT_DEN`, which takes binary values: 1 if the coefficient is non-zero, and 0 otherwise.

We perform initial experiments on the full dataset (which contains all possible twelve-point f-graphs at this loop order), as our goal is to determine whether a perfect classifier exists. In such a context, overfitting is not a concern: we are not interested in generalization to unseen examples, but in whether a model can in principle learn the structure of the data.

4 8-loop result

We evaluate each model using an 80/20 train-test split. That is, 80% of the full dataset is used for training, and the remaining 20% is reserved for testing the model's ability to classify unseen examples.

4.1 Logistic Regression

- **Accuracy:** 0.8007
- **Precision/Recall (Class 0):** 0.84 / 0.83
- **Precision/Recall (Class 1):** 0.74 / 0.76

Logistic regression performs reasonably well, with an overall accuracy of 80% and fairly balanced precision and recall. This suggests that while the problem is not linearly separable, the spectral features contain enough structure for a linear model to capture a significant portion of the signal.

4.2 Decision Tree

- **Accuracy:** 0.7472
- **Precision/Recall (Class 0):** 0.79 / 0.80
- **Precision/Recall (Class 1):** 0.68 / 0.67

The decision tree model achieves slightly lower performance than logistic regression. While it captures some nonlinear structure, its tendency to overfit on the training data may reduce its generalization capability. The precision and recall for class 1 (non-zero coefficients) are notably lower than for class 0.

4.3 Random Forest

- **Accuracy:** 0.8339
- **Precision/Recall (Class 0):** 0.88 / 0.85
- **Precision/Recall (Class 1):** 0.77 / 0.82

Random Forests outperform both logistic regression and the single decision tree, achieving the best overall accuracy and a good balance between precision and recall across both classes. This indicates that the relationship between features and the coefficient label is likely nonlinear and combinatorial in nature. The ensemble method is better suited to capturing such structure, and the results suggest it is able to learn meaningful decision boundaries from the spectral input.

4.4 Perfect Classification and Feature Importance

To investigate whether a perfect classifier exists for this problem, we trained various models on the *entire* dataset, bypassing the train/test split. The aim of this experiment is not to measure generalization, but rather to determine whether the provided features contain enough information to reconstruct the coefficient labels exactly.

Model	**Accuracy on Full Dataset**
Logistic Regression	0.8007
Decision Tree	0.7472
SVM (polynomial kernel)	0.8600
Random Forest (all features)	**1.0000**
Random Forest (without denominator features)	0.9978

Table 4: Classification accuracy of different models trained on the full dataset. Random Forest achieves perfect accuracy, indicating that the coefficient labels are in principle learnable from the given features. Removing the denominator features causes a small but non-negligible drop in performance.

A Random Forest classifier trained on the full dataset was able to achieve **perfect accuracy**. This suggests that the structure of the data is, in principle, learnable: the classification boundary exists and can be captured by a sufficiently flexible model. This aligns with the expectation that the relation between the spectrum-based features and the coefficient label is combinatorial in nature and well-suited to decision-tree based models.

We analyzed the feature importances extracted from the perfect Random Forest model. The top five features are shown below:

Feature	Importance
FGRAPH_EIGEN11	0.113
DEN_EIGEN11	0.111
FGRAPH_EIGEN10	0.101
FGRAPH_EIGEN7	0.060
FGRAPH_EIGEN9	0.052

Interestingly, the most important features are a mix of eigenvalues of the full f-graph (i.e., the numerator-denominator difference spectrum) and of the denominator graph alone. This suggests that both spectra are independently informative, and the model relies on their interaction to successfully separate the classes.

To assess the necessity of the denominator graph spectrum, we repeated the experiment after removing all features derived from it. In this reduced-feature scenario, the Random Forest accuracy on the full dataset dropped slightly to 0.9978. While still very high, this indicates that the denominator spectrum plays a meaningful role in distinguishing subtle cases.

The fact that perfect classification is no longer achieved when these features are removed confirms that they carry essential structural information, even if not sufficient alone to separate all cases.

The missclassified graphs in this case are 6 out of 2709, and have indices:

$$[151, 244, 921, 1467, 1484, 1486]$$

5 10-loop result

Class	Precision	Recall	F1-score	Support
0	0.91	0.97	0.94	152,742
1	0.76	0.45	0.56	27,287
Accuracy			0.89	180,029
Macro Avg	0.83	0.71	0.75	180,029
Weighted Avg	0.89	0.89	0.88	180,029

Table 5: Random Forest classification report. Accuracy: 0.8949. Computational time 11 min 30 sec.

Misclassified indices: [4724, 7075, 19810, 281029, 290977, 291224, 291226, 672010, 824379]

There are 3308 cospectral graphs. 4 out of 9 of the mismatches comes from the co-spectral graphs that have different coefficients, there are

$$(4656, 4724), (290977, 290994), (672010, 672028), (280495, 281029)$$

6 Denominator Graph Classification

We now focus exclusively on the structure of the denominator graph. Rather than predicting the coefficient of each individual f-graph, we group all f-graphs that share the same denominator and ask a coarser question: *does at least one f-graph in the family contribute to the correlator?*

We assign a binary label to each denominator graph:

- **1** if at least one f-graph with this denominator has a non-zero coefficient,
- **0** if all f-graphs with this denominator have vanishing coefficients.

This shifts the focus from fine-grained coefficient prediction to identifying which denominator topologies are relevant for the correlator.

We trained a Random Forest classifier using only the spectral features of the denominator graph, and evaluated performance across three loop orders: 8, 9, and 10 loops. For each loop order, we report results using an 80/20 train-test split. For the 9-loop case, we also tested on the full dataset to assess whether perfect classification is possible.

Results Summary

Loop Order	Accuracy (80/20 split)	Macro F1-score	Accuracy (full)
7	0.7879	0.74	1
8	0.7770	0.78	1
9	0.7635	0.75	0.9998
10	0.7959	0.76	0.99986

Table 6: Random Forest classification performance using only denominator graph features. The macro F1-score averages performance across the two classes (0 and 1).

At 8 and 10 loops, the model achieves around 78–80% accuracy, indicating that the denominator structure is informative but not fully predictive of whether the family contributes. At 9 loops, a model trained on the full

dataset achieves nearly perfect classification, suggesting that the signal is in principle learnable from the denominator graph alone in that case.