Inductive link prediction boosts data availability and enables cross-community link prediction in ecological networks

Barry Biton [1], Rami Puzis [2], and Shai Pilosof [1,3,*]

[1] *Department of Life Sciences, Ben-Gurion University of the Negev, Beer-Sheva, Israel*
[2] *Department of Systems Information and Software Engineering, Ben-Gurion University of the Negev, Beer-Sheva, Israel*
[3] *The Goldman Sonnenfeldt School of Sustainability and Climate Change, Ben-Gurion University of the Negev, Be'er Sheva, Israel*
[*] *Corresponding author: pilos@bgu.ac.il*

Abstract

Predicting species interactions (links) within ecological networks is crucial for advancing our understanding of ecosystem functioning and responses of communities to environmental changes. Traditional link prediction models are often constrained by sparse, incomplete data and are typically limited to single networks. Here, we address these issues using an innovative inductive link prediction (ILP) approach. By pooling data across communities and applying transfer learning, our model predicts interactions within and between ecological networks. We evaluated the model performance on 538 networks across four community types: plant-seed disperser, plant-pollinator, host-parasite, and plant-herbivore. ILP models outperform non-ILP models, achieving a mean balanced accuracy of 0.68 compared to 0.57. Furthermore, ILP consistently predicts missing links more effectively across fractions of missing links. Despite its strengths, cross-community prediction efficacy varies, with plant-seed disperser and host-parasite networks showing better performance than plant-pollinator and plant-herbivore networks as both training and test sets. Finally, leveraging ILP's generalizability, we developed a pre-trained model that ecologists can readily use to make instant predictions for their networks. This study highlights ILP's potential to improve prediction of ecological interactions, enabling generalization across diverse ecological contexts and bridging critical data gaps.

Introduction

Ecosystem services, such as pollination, are vital for human societies and arise from interactions among organisms in ecological communities [1]. For instance, crop pollination depends on interactions between wild bees and non-crop flowers [2]. These dependencies can be modeled as networks, where nodes represent species and links represent interactions. Fully sampling these networks would improve our understanding of indirect dependencies between species, but the resources needed to confirm each interaction are immense [3–5]. Consequently, ecological networks are often under-sampled, limiting structural analysis. Predicting probable but unconfirmed links offers a solution [6–12] and aids in anticipating network responses to anthropogenic changes, such as disease host prediction and local-invasive species interactions [11]. However, predicting links is challenging due to network sparsity, unobserved true negatives, and missing links.

Ecological link prediction models can use information on species traits [9,13] and phylogeny [12,14], but these are often difficult to obtain or biased because some taxonomic groups are far more studied than others. Instead, it is possible to rely on the topology of the known part of the network to predict the unknown [7,8]. For instance, a parasite with a broad host range (high node degree) will likely infect a new host. Such approaches that predict missing links based on known properties within the same network are called *transductive* link prediction [15]. While this is the primary approach used in ecology so far [7,8], its performance is hindered in networks where only a few links are known or where some parts are known much better than the others [16,17], two issues prevalent in ecological networks.

To overcome these issues, we take an approach called inductive link prediction (ILP), in which links in one network are predicted by learning the structure of others [17]. ILP harnesses the principle of universality [18–21], reflected in ecological networks by cross-system topological similarities [22,23]. Indeed, despite system differences (e.g., mutualistic vs. antagonistic networks), processes such as spatiotemporal distributions, evolutionary history, and neutral dynamics produce recurring non-random patterns across networks [24–29]. For example, heavy-tail degree distributions, with most nodes having few links and a few having many [30–32] and structures like nestedness and modularity are common in multiple types of ecological networks [22,33–36].

Cross-system similarities in macroscopic patterns provide an opportunity to increase training data by using multiple networks within an ILP framework [37,38]. In addition, it enables transfer learning, where a model trained on data from one domain is applied to predict outcomes in another. The idea of using transfer learning in ecology has been recently proposed [39], but its application has been very limited [13,14]. For example, Caron et al. [13] used transfer learning to predict links in food webs in one area (e.g., Europe) based on knowledge in another (e.g., Serengeti). Using trait-based predictions, they found that pairwise interactions are better predicted using a model trained on the same food web than with models trained on other food webs. However, they did not pool data from different food webs for training. Moreover, no ecological study to date has applied ILP within a machine learning framework, a powerful approach that offers remarkable flexibility and scalability for leveraging data across diverse systems.

Given the structural similarities between networks from different ecological communities, we hypothesize that ILP is more effective than transductive link prediction for predicting links in ecological networks. Further evidence for the plausibility of this hypothesis comes from two recent studies that showed that it is challenging to identify the type of network (e.g., plant-pollinator, host-parasite) based solely on its structure [23,40]. This is because variation in network structure is similar within and between different types of ecological networks. On the one hand, this observation means that it would be challenging to predict the ecological community of a network based on its structure [23]. On the other hand, the structural similarity of different community types can be utilized to improve link prediction by training models on a large dataset of diverse communities. We tested this hypothesis using an ILP model we developed, which further enables cross-community prediction. We find that our model outperforms transductive link prediction models in predicting links, but that prediction accuracy varies by community type. Specifically, plant-pollinator networks weaken cross-community predictions while host-parasite networks enhance them.

Results

Data

We used the data set compiled by Michalska-Smith and Allesina [23], later used by [40]. We used networks with $\geq$ 25 species and with connectance $\geq$ 0.1 (Table S1, Figs. S1, S2). This data set includes 205 plant-seed disperser networks (PSD), 217 plant-pollinator networks (PP), 84 host-parasite networks (HP), and 32 plant-herbivore networks (PH) (538 networks in total). A potential limitation of the data set could be the non-equal number of networks per community. To test for this, we included the type of community as a feature in the model, and it was not an important feature in the prediction. In addition, communities represented by more networks did not necessarily perform better as train or test sets (see results below), indicating that the number of networks is not a limitation. As previously shown for this data set [23,40], variation in network structure was not higher between than within networks (Fig. S3).

Overview of the link prediction pipeline

We developed a pipeline to evaluate the performance of ILP models on multi-network data sets using nested grouped cross-validation (Fig. 1). We split the data into training, validation and test sets, ensuring that instances (links) from the same network are always together in the set. We predicted links based on network properties, which we used as features in the model. Before training, we used feature selection to mitigate correlations, reduce dimensionality, prevent over-fitting, and identify informative features for link prediction (see feature selection in Methods). In the inner loop, we used the training and validation sets to tune hyperparameters and select the best model to perform link prediction on the test set. We performed 5-fold cross-validation in the outer loop. We trained our model to optimize the F1 score, which balances model's ability to predict links and non-links, using popular machine learning models: logistic regression, random forest, and XGboost. The results did not qualitatively change between models (Supplementary note S1.1), and we present results for Random Forest. We considered the significant imbalance between the number of observed and unobserved interactions in our evaluations [41] (Methods).

Evaluation metrics

We assess model performance using several indices that each highlight different aspects of its strengths and weaknesses [11,41]. Our evaluation is based on a confusion matrix containing counts of true

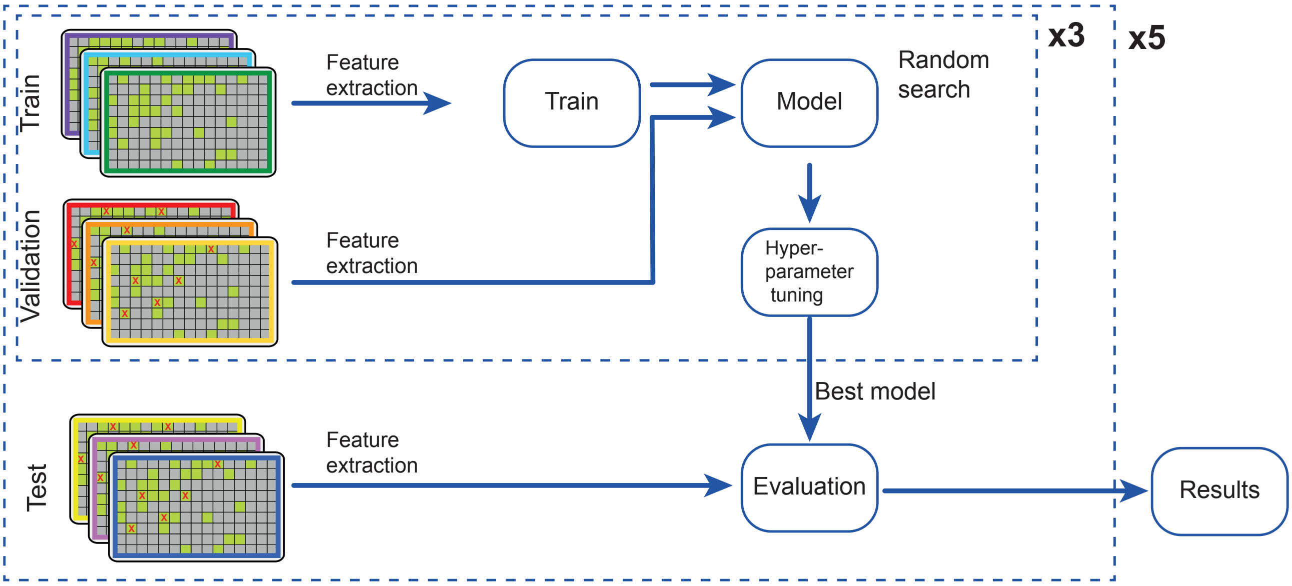

Fig. 1: Pipeline overview. (A) We used nested cross-validation, with 3 and 5 folds in the inner and outer loops, respectively. We split the data into train, validation, and test sets and calculated network properties (feature extraction). The validation and test sets contained withheld links (red x's). We used the inner loop to tune the hyperparameters (using random search) and selected the best model to predict links in the test set in the outer loop. Model evaluation was based on a confusion matrix with TP, TN, FP, and FN values gathered from the outer loop's five folds. When splitting the data, we ensured that each network was included entirely in the set (i.e., we split the data by networks, not by links) and that each network appeared at least once in the test set. See details in Methods.

positives, true negatives, false positives, and false negatives. For imbalanced data, we employed common metrics such as recall $(TP/(TP+FN))$, precision $(TP/(TP+FP))$, the F1 score (the harmonic mean of precision and recall), specificity $(TN/(TN+FP))$, and balanced accuracy (the arithmetic mean of recall and specificity). These metrics require setting a threshold for link probability to decide when a link is considered present, and we followed the standard practice of using a 0.5 threshold. However, links can also be evaluated using alternative thresholds. Consequently, we further examined the precision-recall tradeoff across thresholds and computed the area under the precision-recall curve (precision-recall (PR) AUC). The PR tradeoff indicates the ability of the model to retrieve links (recall) while minimizing false positives (precision) and is adequate for imbalanced data. For more details, see Supplementary Note S1.2.

Inductive link prediction outperforms transductive approaches

To test the hypothesis that collating data across communities would improve link prediction, we compared our model performance to three transductive models that train and test on a single network at a time: the stochastic block model (SBM), connectance, and matching centrality models, which were previously published and are not based on machine learning [8]. While ILP predicts links with similar recall to that of transductive link prediction (TLP) models, it does so with higher precision. In addition, the ILP model recovers non-links with much higher specificity than TLP (Fig. 2). The area under the PR curve was significantly higher in the ILP model. However, the TLP models were not based on machine learning techniques. To ensure a more fair comparison we also trained a transductive machine learning model. The ability of that model to recover links (recall) was much lower than ILP, though it had a better specificity (Fig. 2). Finally, we calculated the overall assessment of the models' accuracy by considering their ability to detect both links and non-links, using the F1 and balanced accuracy measures. Overall, the ILP model outperformed the transductive ones (Fig. 2, Fig. S4).

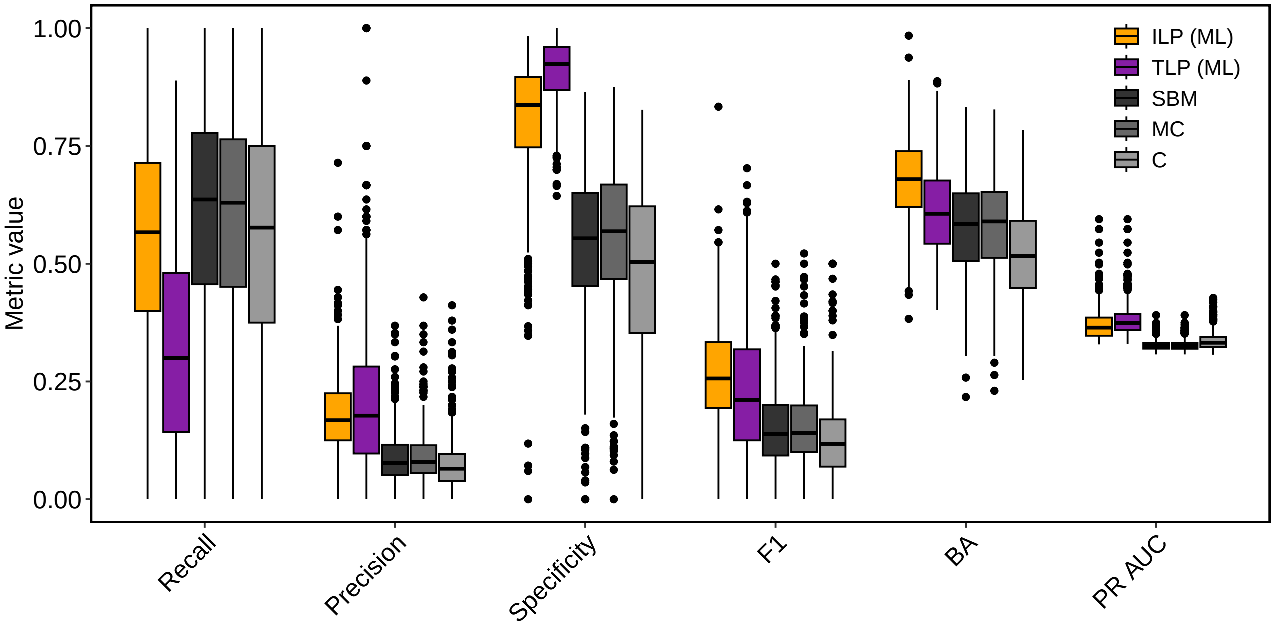

Fig. 2: Comparison of model performance. The inductive link prediction model was trained and tested on networks from all communities. We compared ILP to three transductive previously published models (stochastic block model, connectance (C) and matching centrality (MC) [8]), and to a transductive machine learning model that we trained (TLP). Each boxplot is a distribution of an evaluation metric using a 0.5 classification threshold, bedsides the PR AUC. The boxplots show the interquartile range (box), median (central line), data within 1.5 times the interquartile range (whiskers), and potential outliers (individual points beyond whiskers). Each data point is a network and boxplots contain networks from all the five outer folds. BA is balanced accuracy. See definitions of evaluation metrics in the Methods and additional metrics in (Fig. S4).

To better understand the differences in model performance, we further investigated how model predictions vary with node degree (Fig. 3). Specifically, each link is composed of two species, in the low and high trophic levels, and their degrees are features in the model. We chose node degree because the most influential features were related to degree (Fig. S5). Both the inductive and transductive machine learning models outperformed the non-machine learning models across the range of node degrees. While the transductive machine learning model has high specificity across node degrees, its ability to detect links (F1) lower. This difference between the ILP and TLP models was particularly evident for the high trophic level species (pollinators, seed dispersers, herbivores, and parasites). Therefore, overall, collating data increases our ability to predict positive links regardless of the level of generalism of the species involved, particularly for high-trophic level species.

Prediction ability varies by community type

Network types vary in the ecological and evolutionary processes that determine them. Therefore, we expect variation in our ability to predict links in different communities. We first evaluated prediction within each community type. The PR curves showed that predictive ability varies by community type, with the best predictions performed in host-parasite networks and the worst in plant-pollinator networks (Fig. 4A). Comparing communities across a range of evaluation metrics revealed more nuanced differences (Fig. 4B, Fig. S6, Table S2). The ILP model's ability to predict plant-herbivore and plant-pollinator interactions or host-parasite compared to plant-seed disperser interactions was

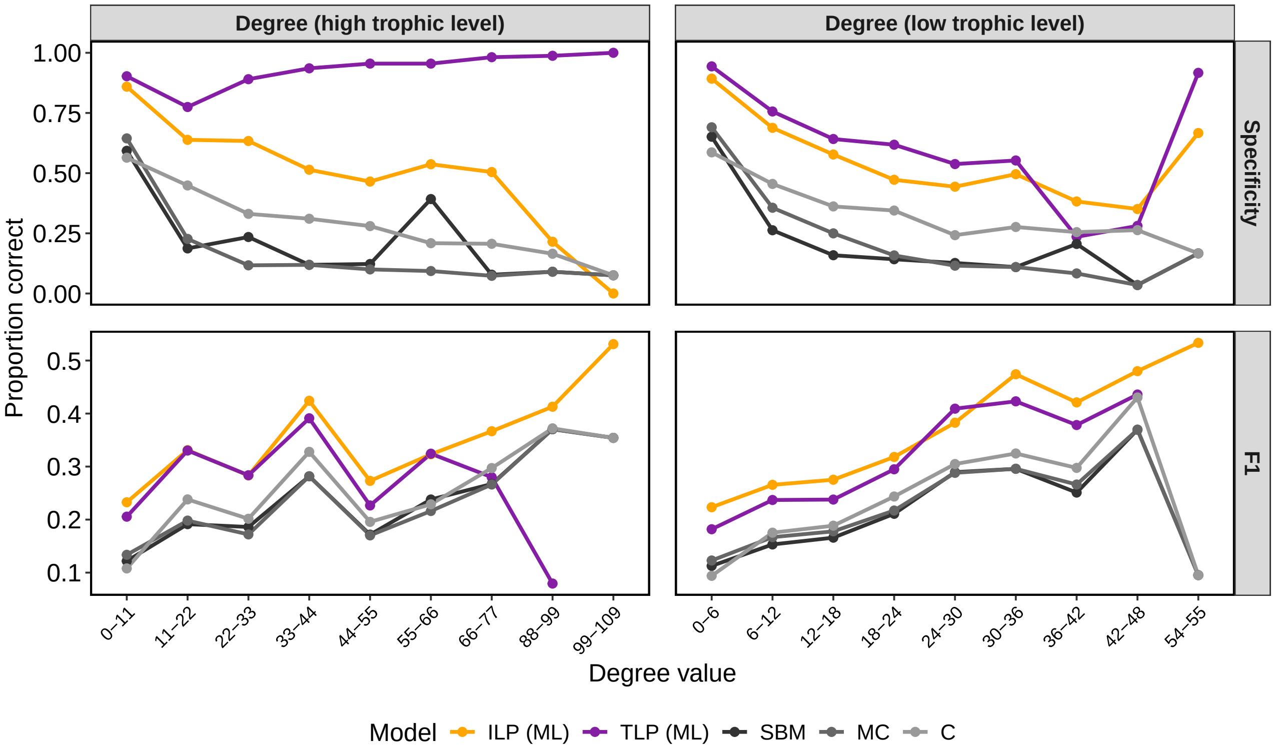

Fig. 3: Comparison of model performance across node degrees. The ILP and TLP models outperform the non-machine learning models across node degrees in detecting positive (F1) and negative (specificity) links. The ILP model is detects positive links better than a TLP model, particularly for nodes with high degree.

the same for all evaluation metrics. In contrast, its ability to predict plant seed-disperser interactions was significantly higher than plant-pollinator interactions.

Communities vary in their quality as train and test sets

The ILP model's main advantage is in transfer learning, allowing us to perform cross-community link prediction. We trained the model using networks from a given community type and used them to predict links in the same community or in any other. To create unbiased comparisons, we ensured that the number and identity of the networks in the training set were the same across experiments. Because we are primarily interested in predicting missing links accurately, we focus on the F1 metric. We expected prediction to be more robust within community types (the diagonal in Fig. 5). This prediction was confirmed for host-parasite networks only. In other communities, interactions are predicted similarly well or better when trained on communities that are not the predicted ones.

We further hypothesized that pooling data across all networks (All) for training will improve predictions (i.e., that the "All" column will have higher values than within-community predictions). In contrast to our hypothesis, pooling data worsened the predictions. This result can stem from the fact that plant-pollinator networks are highly sparse and consistently under-perform (Fig. 4A), affecting overall predictions. Indeed, removing plant-pollinator networks from the training set (No PP) improved predictions. This may happen because while numerous instances existed in the data set, plant-pollinator networks had the lowest connectance [17]. Low connectance leads to a significant imbalance between existing and non-existing links, which can decrease the precision and recall scores.

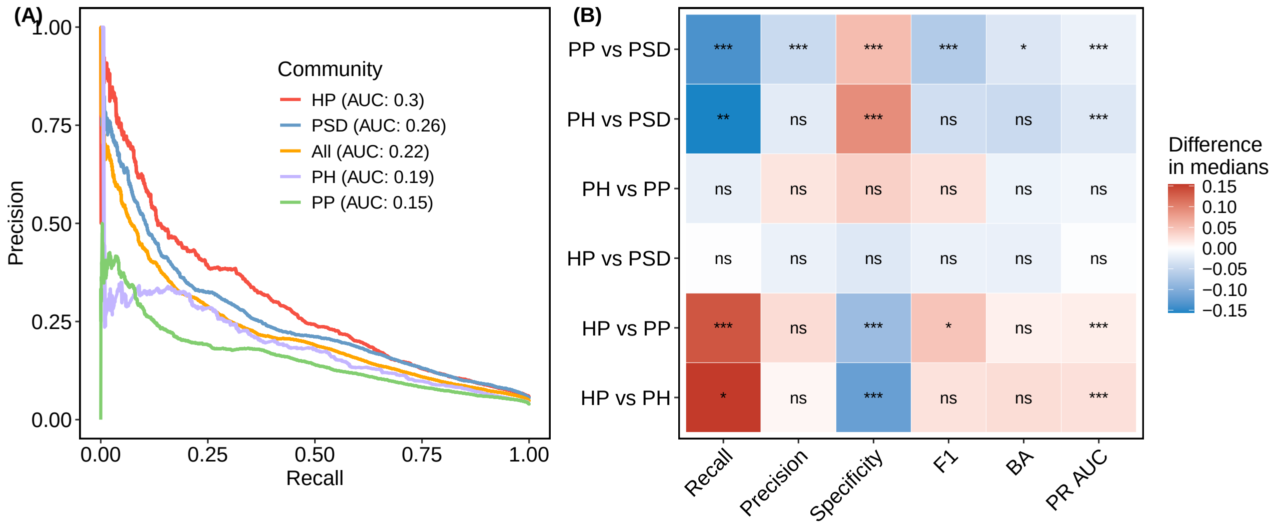

Fig. 4: Prediction within each community types. The model was trained on networks from all communities and tested separately for each community type. **(A)** PR curves across thresholds. The AUC values are noted in the legend. **(B)** Pairwise comparison between communities for multiple evaluation metrics (the comparison between the PR curves is at the right-most column). For each metric, we tested differences in medians across all communities using Kruskal-Wallis, and then performed a Dunn posthoc test with Bonferroni correction for pairwise comparisons. Positive values indicate that the median was higher in the second community (e.g., PSD had higher specificity than PP networks). $* : p - value < 0.05, ** : p - value < 0.01, *** : p - value < 0.001$, ns: not significant. PSD: plant-seed disperser, PP: plant-pollinator, HP: host-parasite, PH: plant-herbivore.

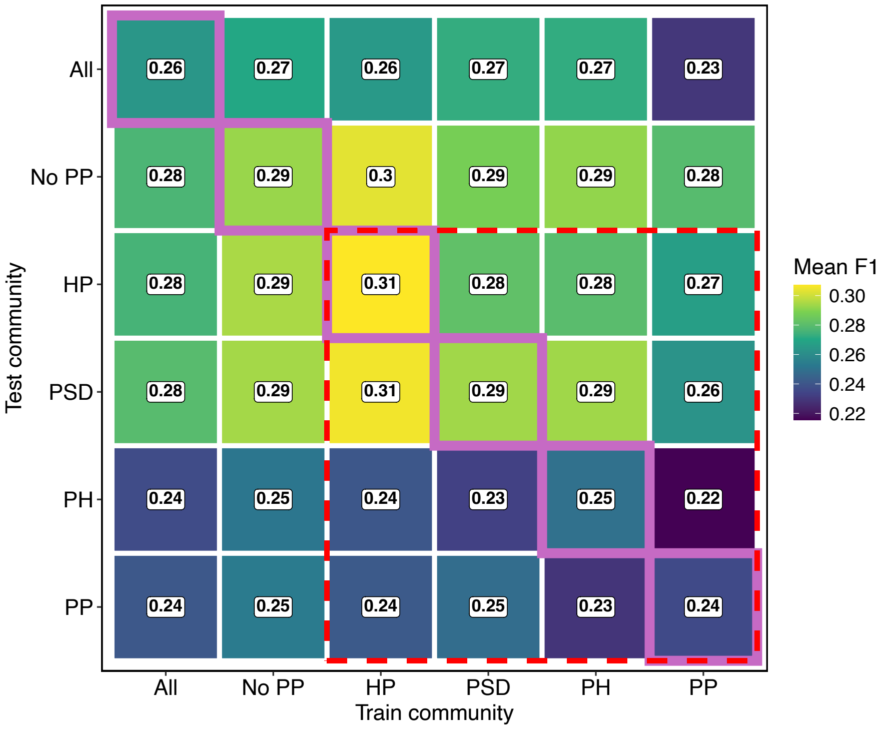

Fig. 5: Link prediction within and between community types. Each cell in the heatmap is the median of F1, calculated across all networks for a model that was trained on data from a certain community (columns) and tested on another (rows). The diagonal depicts model results for models trained and tested on the same community type. The red dashed border encloses models trained and tested using single community types. PSD: plant-seed disperser, PP: plant-pollinator, HP: host-parasite, PH: plant-herbivore. 'All' is a model trained or tested using pooled data from all communities. 'No PP' is the same as 'All' but without plant-pollinator networks.

Estimating the bounds of link prediction performance

One issue common to all link prediction studies is the lack of knowledge of the ground truth. That is, which of the non-existing links are truly missing (if we had this knowledge, we would not need link prediction). Moreover, it is likely that obtaining a fully complete network without setting a threshold may be extremely hard because there is typically a very long tail of very weak interactions. Ideally, link prediction models would guide the field sampling efforts of ecologists who want to complete their networks. In turn, field data can evaluate model predictions. To date, no empirical study has estimated model performance in light of ground truth. Undertaking such effort is necessary but may be extremely time-consuming because if links were missing in the first place, an intensive amount of sampling would likely be needed to sample even a few of them. As a starting point, we take here an alternative computational approach. We calculate the bounds of the ILP and TLP model predictions across a theoretical range of proportion of true missing links. We devised two scenarios. In the best-case scenario, the false positive links of the models were indeed missing links. In the worst-case scenario, the predicted negative links are actually positive in nature (see Methods). F1 is a good measure of our model's predictive ability because it is based solely on instances we withheld, which are observed links. The ILP model outperforms TLP in recovering missing links overall, as indicated by the consistently higher F1 in the best-case and worst-case scenarios. In contrast to ILP, the TLP model will better predict non-links across the range of missing link proportions, as indicated by higher specificity. However, the difference between these models in specificity is much smaller than for F1 (Fig. 6).

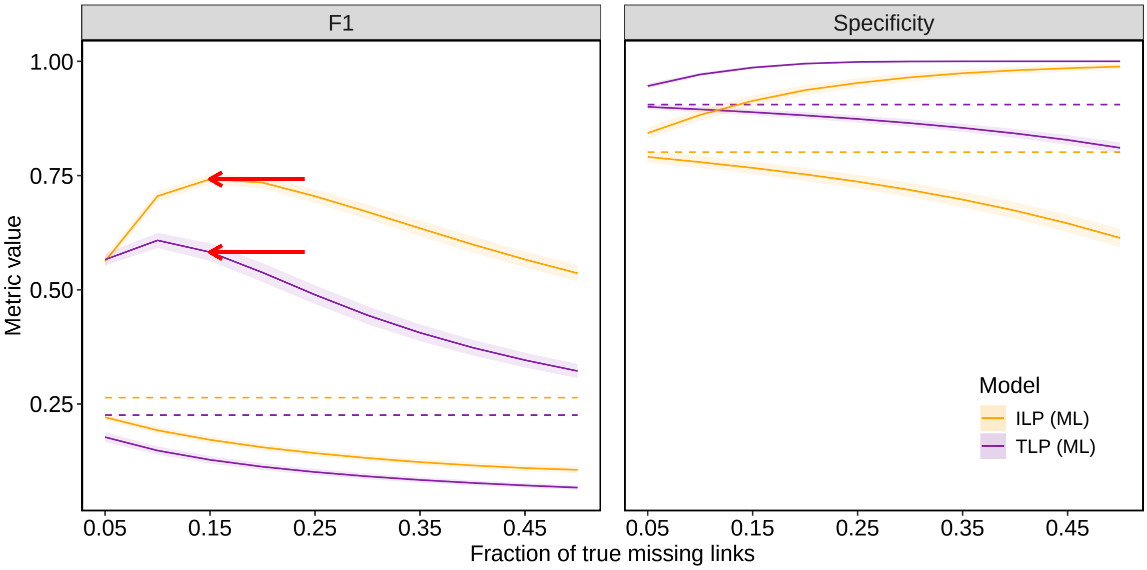

Fig. 6: Bounds of model predictions across proportions of ground-truth missing links. The horizontal dashed lines depict the current models' performance. The upper and lower curves (solid lines) represent the best-case (false positive links of the models were indeed missing links) and worst-case (predicted negative links are actually positive in nature) scenarios, respectively. The x-axis is the proportion of links that are missing (unobserved but exist in nature). An example for figure interpretation for the red arrows: If the true proportion of missing links in nature is 0.15, the ILP and TLP models' best-scenario F1 values would be ≈ 0.74 and ≈ 0.58, respectively. Bounds were calculated as means, and their confidence intervals (light-colored ribbons) were calculated across all networks. The ILP model was one in which the train and test contained networks from all communities.

Model implementation and a case study

Because ILP enables prediction in networks on which the model was not trained, we developed an online implementation that includes a pre-trained model. The application allows users to upload an existing network, and instantly obtain probabilities for the links (Fig. S7). Users can also choose to withhold links and obtain model evaluations (e.g., F1, balanced accuracy, etc). The model is available online (`https://ecomplab-eco-ilp.hf.space`). In addition, we released the model as a Python package `https://github.com/Ecological-Complexity-Lab/eco_ILP/package`.

To demonstrate a use case, we analyze a host-parasite network (that was not part of the training set), representing the infection of 22 small mammalian host species by 56 ectoparasite species during six consecutive summers in Siberia (1982–1987)[42,43]. In each of the six networks, we predicted links (withholding 20% of interactions). The prediction ability did not vary greatly across years (except for recall, Fig. 7A). We therefore aggregated the network (unifying links across years) and predicted links in the aggregated network. We find that F1 increases with the degree of species but faster in parasites (Fig. 7B; see Fig. S8 for yearly correlations). Therefore, for the same degree, it is easier to predict links for parasites than for hosts. This pattern was not affected by host or parasite taxonomy (Fig. S9).

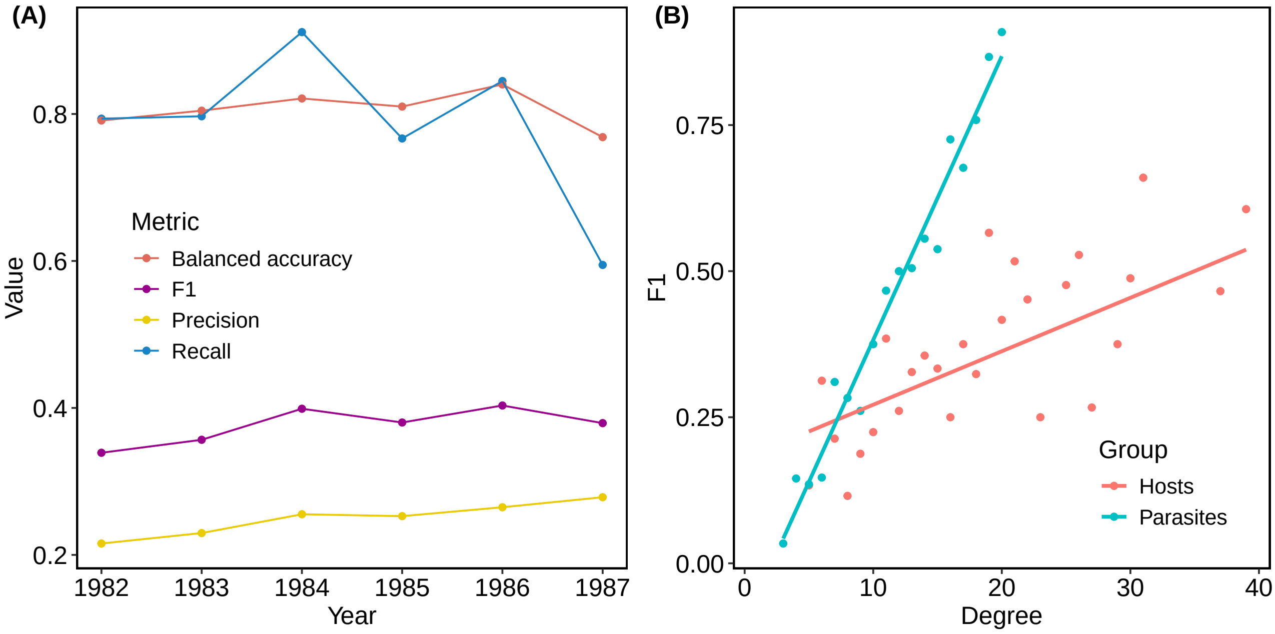

Fig. 7: Case study. We use our model to predict a host-parasite network (that was not part of the training set) representing the infection of 22 small mammalian host species by 56 ectoparasite species during 6 consecutive summers in Siberia (1982–1987). In each prediction we withheld 20% of the links. **(A)** Overall prediction metrics per year. **(B)** A correlation between parasite degree and F1.

Discussion

Species interactions are the backbone of ecosystem functioning. Link prediction helps us improve our knowledge of species interaction structure and gain insights into how interactions would change in response to perturbations. However, training data are often incomplete and biased. In this paper, we take a step forward using ILP, which allowed us to pool data across networks. The ILP approach also enables predicting interactions in entirely new networks (based on their existing structure) via transfer learning, an advantage we leveraged to develop implementations that allow researchers to obtain instant predictions for existing networks.

ILP models overcome the limitations of sparse and incomplete data by pooling information across different systems. Indeed, the model we developed generally outperformed transductive models. Moreover, our theoretical evaluation of model performance showed that ILP would retrieve missing links better than TLP regardless of the true proportion of missing links. However, the TLP machine learning model better predicted non-links, pointing to a trade-off between predicting these two classes. Because typically ecologists are most interested in predicting links, ILP offers a better solution.

Using networks from multiple communities allowed us to perform cross-community predictions, an approach previously tested only on four food webs using trait data [13]. Unlike that study, we found that predictions were not necessarily better within each community. For instance, plant-pollinator and plant-herbivore interactions were predicted equally well regardless of the training community, suggesting that different kinds of ecological networks can share similar assembly patterns or co-evolutionary dynamics [44], leading to comparable structures and constraints on interactions [23]. These findings reinforce that the structural similarities across ecological networks enable effective cross-community link prediction, even amidst taxonomic differences. They further align with our finding that network type was not an important predictor of links. Previous studies also showed that distinguishing between different types of ecological networks can be challenging because structural variation is similar within compared to between network types [23,40].

On the other hand, host-parasite and plant-seed disperser networks had better predictive performance than plant-pollinator and plant-herbivore networks overall, and were better predicted when trained on the same community. These differences between communities in train/test quality are an avenue for future research. An obvious question is whether those differences are solely due to data quality or whether there are ecological mechanisms underlying interaction probabilities. For instance, host-parasite and plant-seed disperser networks exhibit higher connectance, further facilitating predictions (Table S1). Along with that, host-parasite and seed-disperser networks often involve more specialized animal groups, possibly rendering their topology more predictable. In contrast, the diverse and heterogeneous animal species in plant-pollinator networks, each with distinct behaviors and roles, can reduce predictability. Therefore, variation between communities in their quality as training data underscores the need to consider each community type's unique characteristics. Incorporating features specific to community types, such as traits [9,12,13], structural properties [45] or environmental features [46], may aid in addressing such open questions and simultaneously increase model performance. Hence, future studies can build on our work to improve the ILP model by addressing specific data biases and incorporating non-structural features (e.g., phylogeny).

Model generalizability and transfer learning make ILP models particularly valuable for various practical applications. For instance, our pretrained model offers researchers an easy-to-use tool for instant predictions, enabling them to explore potential interactions in their networks with minimal effort. By prioritizing interactions most likely to be missing, this tool can guide field sampling efforts, significantly improving the efficiency of data collection. Nevertheless, validating prediction outcomes should be done via field sampling of well-known systems in which missing and forbidden links can be evaluated with high credibility.

Researchers can further use ILP to predict how interactions might change in time [47] or under environmental shifts, such as habitat loss, climate change, or species introductions. This can be done by providing a baseline network and potential changes such as removal/addition of species or interactions under the new conditions (or time points), allowing the model to infer changes in interaction patterns from the baseline to the new scenarios. By forecasting dynamic responses, ILP offers valuable insights for anticipating ecosystem reorganization, guiding conservation efforts, and addressing both theoretical and pressing ecological challenges.

In conclusion, our study demonstrates the potential of ILP models to enhance ecological research. However, our findings also reveal significant variability in the quality of train sets across different ecological communities, emphasizing the need for tailored approaches that consider the unique characteristics of each community. We further provide an online application and a Python package for instant predictions via a pre-trained model that can guide efficient field sampling. Therefore, beyond filling gaps in data, generalizable ILP models, such as ours, provide a scalable solution for predicting links in rapidly changing ecosystems, potentially enabling the identification of interactions involving invasive species and novel disease hosts to support proactive conservation and management strategies.

Methods

Nested cross validation and hyperparameters

Nested cross-validation allows optimization of hyperparameters along with an unbiased estimation of the model's performance. This reduces the risk of over-fitting by ensuring that the model evaluation is conducted on data not seen during the hyperparameter tuning phase [48]. Hyperparameters control the learning process rather than being learned from the data, and are set before the training process begins [49]. The optimal values for the hyperparameters that maximize the machine learning model's performance are selected through hyperparameter tuning during that inner loop of the nested cross-validation (Table S3). We optimized hyperparameters using a random search, which is a computationally efficient tuning technique. The process involved randomly sampling values from a predefined search space, followed by training and evaluating model performance on those values using the training and validation sets. We used the F1-score as the performance metric to evaluate the hyperparameter combinations. The average F1-score across all folds was calculated for each random set of hyperparameters. We repeated this procedure for multiple random sets of hyperparameters, and the set that yields the highest F1 score was chosen as the optimal solution for the current iteration of the outer loop.

Because the data includes multiple networks, we used grouped cross-validation to prevent information leakage between training, validation, and test sets [50]. This method ensures each fold contains entire networks, increasing model robustness by preventing instances from the same network from appearing in the training, validation and test sets simultaneously.

We built and evaluated the machine learning models using the `Scikit-learn` package [51], with the addition of the `xgboost` package [52]. We executed hyperparameter optimization through randomized search using the scikit-learn's RandomizedSearchCV [51]. We used randomized optimization due to its efficiency in exploring the hyperparameter space with fewer iterations compared to traditional grid search method. We optimized the model's hyperparameters using the F1-score.

Link withholding

Because the goal is to predict missing links (possible yet unobserved interactions [53]), link prediction requires generating a ground truth. That is, a lack of link (0 in the matrix), of which we are certain the link actually exists. A common way to do so is by withholding some existing links randomly [8,54]. Hence, to emulate real-world under-sampled networks, each network in the validation and test sets was sub-sampled by randomly withholding 20% of the existing links. We chose this proportion to balance sufficient data for training while ensuring robust evaluation of model performance. However, we ran a sensitivity analysis withholding 5-30% of the links. In addition, we tested three withholding strategies (Supplementary note S1.3).

Withholding creates three types of links:

- *Existing links*: links that exist in the network and were not sub-sampled.
- *Non-existing links*: links that did not exist in the network. These may be missing or forbidden (interactions not possible due to some ecological, morphological or other constraints [53]).
- *Withheld links*: links that existed in the network and are now missing because they were removed. Because we want to predict instances in which non-existing links should have existed, withheld links were relabeled as existing after feature extraction.

Feature extraction

Each instance in the data set, representing an interaction (or lack of an interaction), between each two species, constitutes a vector of topological features. Our features encompassed three levels as follows:

- Network-level features: Defined for a whole network (e.g., nestedness). Hence, all the instances that are related to a network will get the same values for those features. We also included the type of network (e.g., plant-pollinator).
- Link-level features: Defined for a pair of nodes (e.g., preferential attachment: the multiplication of both node degrees).
- Node-level features: Defined for each node (e.g., centrality, degree). Each instance will have two different versions of the feature, one for each node in the pair (higher and lower trophic levels).

Feature extraction is done once on all the data, and is not related to the fold. We rescaled numerical features to a range of [0, 1] to ensure that no particular feature dominates others during the learning process. We calculated feature importance based on the average decrease in Gini impurity across all trees. Features that are more important are used more frequently and result in significant improvements in node purity.

For the analysis of ecological networks and extraction of topological features (network properties), we used the `networkx` package [55] in Python and the `igraph` [56] and `bipartite` [57] R packages. We handled data manipulation through `numpy` [58] and `Pandas` [59,60] in Python. A complete list of the features we used and their descriptions are provided on the GitHub repository accompanying this article (https://github.com/Ecological-Complexity-Lab/eco_ILP/blob/main/results/final/features.csv).

While some of the features we used were inevitably correlated [27], the overall correlation was not high (generally below 0.5). We present a summary of the correlation values in Fig. S10, as well as the detailed correlation matrices between features at the network (Fig. S11), node (Figs. S12, S13) and link (Fig. S14) levels.

Feature selection

Prior to training, we explored correlations using a correlation matrix. While some of the features we used were inevitably correlated [27], the overall correlation was not high (generally below 0.5, see feature extraction in Methods). Moreover, given that the study's goal was to perform a large-scale pattern detection rather than to identify features that drive specific patterns, we did not a-priori remove features. In addition, we mitigated correlations by feature selection, which is a common process in machine learning. Feature selection also allowed us to reduce dimensionality, mitigate over-fitting, and identify the most informative features for link prediction.

We implemented feature selection as an integral part of the modeling process using a pipeline-based approach, employing mutual information as the selection criterion to capture both linear and non-linear relationships between features and the target variable (link presence/absence). We used scikit-learn's SelectKBest with mutual_info_classif as the scoring function. This method evaluates the mutual dependence between each feature and the target variable without assuming a specific relationship type. We chose mutual information over alternative metrics (e.g., correlation coefficients or chi-squared tests) for its robustness in capturing non-linear dependencies. The number of selected features was treated as a hyperparameter, optimized through cross-validation alongside other model parameters using RandomizedSearchCV. We tested several feature set sizes ($k \in \{10, 20, 30, 40, 50\}$), allowing the cross-validation process to determine the optimal number of features based on the specific dataset and model configuration.

In the inductive model, feature selection was performed once per each outer fold, and the selected features were consistently applied to all predictions of that fold. Cross-validation ensured the stability of feature selection across different data splits. In contrast, the transductive model applied feature selection independently for each network, enabling the model to adapt its feature set to the unique characteristics of each network. This approach integrated feature selection directly into the prediction task, ensuring flexibility and precision across different ecological contexts.

Dealing with class imbalance

In machine learning, class imbalance is a situation in which the distribution of classes in the training data is highly skewed, where one or more classes have considerably fewer samples compared to others. This imbalance can significantly impact the training and evaluation of prediction models [41]. Classifiers developed with such skewed data tend to favor the majority class, which can lead to subpar performance when identifying instances of the minority class. This issue is particularly prevalent in ecological networks, characterized by their sparsity (i.e., low connectance). In binary classification tasks, this sparsity creates a disparity between the small number of existing links (positive class) and the much larger set of non-existing links (negative class) [11,41]. To overcome this problem we incorporated cost-sensitive learning (Supplementary note S1.4) [51]. Specifically, to make the importance of both classes equal, we computed their weights inversely proportionally to the frequency of the respective classes [61,62].

Comparison to transductive models

We compared our model to four transductive link prediction models: The SBM, connectance, and matching-centrality, which were previously published [8] and are not based on machine learning, and to a transductive machine learning model we developed (TLP). All four models train and test on each network separately and do not separate links between the train and test set.

In *SBM*, nodes are partitioned into blocks or groups, and the probability of a link between any two nodes depends on the blocks to which they belong [63]. The probability of a link between two nodes is higher if they belong to the same block, reflecting community structure. The stochastic block model is a degree corrected bipartite SBM algorithm, which accounts for the degree heterogeneity within the blocks [64]. The *connectance model* is a model used to describe the pattern of interactions in ecological networks. It assigns a connectivity value to each species, reflecting its propensity to interact with others. The connectivity values are estimated by using maximum likelihood optimization, which adjusts the parameters to best fit the observed interaction data [8]. The *matching-centrality model* combines a trait-based framework with connectivity values to predict interactions in ecological networks. Each species is assigned a latent trait value, representing an unmeasured characteristic influencing its interaction propensity. The model calculates the probability of an interaction between two species as a function of their trait similarity, with higher probabilities assigned to species pairs with similar traits. Additionally, it incorporates connectivity values to account for species' interaction frequencies. Parameters are estimated through maximum likelihood optimization, balancing both trait similarity and species connectivity to fit observed interaction data [8]. We used the R package cassandRa [65] to predict links using these two models. The TLP model we developed uses random forest with a standard 3-fold cross validation with hyper-parameters tuning (maximizing f1 score). The model balances the weights of the classes by computing weights inversely proportionally to the frequency of the respective classes.

Estimating the bounds of link prediction performance

We evaluate the models' ability to predict links under a range of true missing link values. For a theoretical proportion p (range 0.05-0.5) of ground-truth missing links, the **number** of missing links is calculated as $L_m = p \times L_{ne}$, where L_{ne} is number of zeros (non-links) in the original matrix. In the *best case scenario*, we choose L_m FP links and change them to TP. This simulates a scenario in which the model indeed predicted links that were not observed. If $L_{FP} < L_m$ (L_{FP} is the number of FP links) we convert a remainder of $L_m - L_{FP}$ TN links to FN to simulate the scenario in which the links actually existed in nature. In the *worst-case scenario*, we choose L_m TN links and convert their class to FN. This simulates a scenario in which the links actually existed in nature but the model failed to retrieve them. If $L_{TN} < L_m$ (L_{TN} is the number of TN links) we convert a remainder of $L_m - L_{FN}$ FP links to TP. After the conversion of links, we recalculate each evaluation metric (e.g., F1, recall) for each of the two scenarios, to form the upper and lower bounds of model performance per network.

Code and data availability

The data are available in the repository set up in original publication https://osf.io/my9tv/. The full code and technical descriptions on how to run the pipeline are available on the GitHub repository https://github.com/Ecological-Complexity-Lab/eco_ILP.

Acknowledgments

We thank Prof. Peter Mucha for comments and suggestions.

Funding

This work was supported by grants to SP from the Israeli Data Science Initiative, Israel Science Foundation (grant 1281/20), the Human Frontiers Science Program Organization (grant number RGY0064/2022) and the Binational Science Foundation (NSF-BSF grant 2022721)

References

1. Windsor, F. M. *et al.* Network science: Applications for sustainable agroecosystems and food security. *Perspectives in Ecology and Conservation* **20,** 79–90. doi:10.1016/j.pecon.2022.03.001 (2022).
2. Magrach, A. *et al.* Plant-pollinator networks in semi-natural grasslands are resistant to the loss of pollinators during blooming of mass-flowering crops. *Ecography* **41,** 62–74. doi:10.1111/ecog.02847 (2018).
3. Falcão, J. C. F., Dáttilo, W. & Rico-Gray, V. Sampling effort differences can lead to biased conclusions on the architecture of ant–plant interaction networks. *Ecol. Complex.* **25,** 44–52. doi:10.1016/j.ecocom.2016.01.001 (2016).
4. Jordano, P. Sampling networks of ecological interactions. *Funct. Ecol.* **30,** 1883–1893. doi:10.1111/1365-2435.12763 (2016).
5. Dallas, T., Park, A. W. & Drake, J. M. Predicting cryptic links in host-parasite networks. *PLoS Comput. Biol.* **13,** e1005557. doi:10.1371/journal.pcbi.1005557 (2017).
6. Rohr, R. P., Naisbit, R. E., Mazza, C. & Bersier, L.-F. Matching-centrality decomposition and the forecasting of new links in networks. *Proc. Biol. Sci.* **283.** doi:10.1098/rspb.2015.2702 (2016).
7. Desjardins-Proulx, P., Laigle, I., Poisot, T. & Gravel, D. Ecological interactions and the Netflix problem. *PeerJ* **5,** e3644. doi:10.7717/peerj.3644 (2017).
8. Terry, J. C. D. & Lewis, O. T. Finding missing links in interaction networks. *Ecology* **101,** e03047. doi:10.1002/ecy.3047 (2020).
9. Pichler, M., Boreux, V., Klein, A., Schleuning, M. & Hartig, F. Machine learning algorithms to infer trait-matching and predict species interactions in ecological networks. *Methods Ecol. Evol.* **11,** 281–293. doi:10.1111/2041-210x.13329 (2020).
10. Stock, M. *et al.* Pairwise learning for predicting pollination interactions based on traits and phylogeny. *Ecol. Modell.* **451,** 109508. doi:10.1016/j.ecolmodel.2021.109508 (2021).
11. Strydom, T. *et al.* A roadmap towards predicting species interaction networks (across space and time). *Philos. Trans. R. Soc. Lond. B Biol. Sci.* **376,** 20210063. doi:10.1098/rstb.2021.0063 (2021).
12. Benadi, G., Dormann, C. F., Fründ, J., Stephan, R. & Vázquez, D. P. Quantitative Prediction of Interactions in Bipartite Networks Based on Traits, Abundances, and Phylogeny. *Am. Nat.* **199,** 841–854. doi:10.1086/714420 (2022).

13. Caron, D. *et al.* Trait-matching models predict pairwise interactions across regions, not food web properties. *Glob. Ecol. Biogeogr.* **33.** doi:10.1111/geb.13807 (2024).

14. Nunes Martinez, A. & Mistretta Pires, M. Estimated missing interactions change the structure and alter species roles in one of the world's largest seed-dispersal networks. *Oikos,* e10521. doi:10.1111/oik.10521 (2024).

15. Ahmadi, Z. *et al.* Inductive and transductive link prediction for criminal network analysis. *J. Comput. Sci.* **72,** 102063. doi:10.2139/ssrn.4331130 (2023).

16. Fire, M., Puzis, R. & Elovici, Y. in *Handbook of Computational Approaches to Counterterrorism* (ed Subrahmanian, V. S.) 283–300 (Springer New York, New York, NY, 2013). doi:10.1007/978-1-4614-5311-6_14.

17. Galkin, M., Berrendorf, M. & Hoyt, C. T. An Open Challenge for Inductive Link Prediction on Knowledge Graphs. *arXiv [cs.LG]* (2022).

18. Watts, D. J. & Strogatz, S. H. Collective dynamics of 'small-world' networks. *Nature* **393,** 440–442 (1998).

19. Barabási, A.-L. *Network Science* (2021).

20. Barabási, A.-L. & Albert, R. Emergence of scaling in random networks. *Science* **286,** 509–512. doi:10.1126/science.286.5439.509 (1999).

21. Milo, R *et al.* Network motifs: simple building blocks of complex networks. *Science* **298,** 824–827. doi:10.1126/science.298.5594.824 (2002).

22. Fortuna, M. A. *et al.* Nestedness versus modularity in ecological networks: two sides of the same coin? *J. Anim. Ecol.* **79,** 811–817. doi:10.1111/j.1365-2656.2010.01688.x (2010).

23. Michalska-Smith, M. J. & Allesina, S. Telling ecological networks apart by their structure: A computational challenge. *PLoS Comput. Biol.* **15,** e1007076. doi:10.1371/journal.pcbi.1007076 (2019).

24. Bascompte, J. & Stouffer, D. B. The assembly and disassembly of ecological networks. *Philos. Trans. R. Soc. Lond. B Biol. Sci.* **364,** 1781–1787. doi:10.1098/rstb.2008.0226 (2009).

25. Vázquez, D. P., Blüthgen, N., Cagnolo, L. & Chacoff, N. P. Uniting pattern and process in plant–animal mutualistic networks: a review. *Ann. Bot.* **103,** 1445–1457. doi:10.1093/aob/mcp057 (2009).

26. Bascompte, J. & Jordano, P. *Mutualistic networks* (Princeton University Press, Princeton, 2014).

27. Delmas, E. *et al.* Analysing ecological networks of species interactions: Analyzing ecological networks. *Biol Rev* **94,** 16–36. doi:10.1111/brv.12433 (2019).

28. Segar, S. T. *et al.* The Role of Evolution in Shaping Ecological Networks. *Trends Ecol. Evol.* **35,** 454–466. doi:10.1016/j.tree.2020.01.004 (2020).

29. Guimarães, P. R. The Structure of Ecological Networks Across Levels of Organization. *Annu. Rev. Ecol. Evol. Syst.* doi:10.1146/annurev-ecolsys-012220-120819 (2020).

30. Jordano, P., Bascompte, J. & Olesen, J. M. Invariant properties in coevolutionary networks of plant–animal interactions. *Ecol. Lett.* **6,** 69–81. doi:10.1046/j.1461-0248.2003.00403.x (2003).

31. Vázquez, D. P., Poulin, R., Krasnov, B. R. & Shenbrot, G. I. Species abundance and the distribution of specialization in host–parasite interaction networks. *J. Anim. Ecol.* **74,** 946–955 (2005).

32. Williams, R. J. Biology, methodology or chance? The degree distributions of bipartite ecological networks. *PLoS One* **6,** e17645. doi:10.1371/journal.pone.0017645 (2011).

33. Bascompte, J., Jordano, P., Melián, C. J. & Olesen, J. M. The nested assembly of plant–animal mutualistic networks. *Proc. Natl. Acad. Sci. U. S. A.* **100,** 9383–9387. doi:10.1073/pnas.1633576100 (2003).

34. Olesen, J. M., Bascompte, J., Dupont, Y. L. & Jordano, P. The modularity of pollination networks. *Proc. Natl. Acad. Sci. U. S. A.* **104,** 19891–198916. doi:10.1073/pnas.0706375104 (2007).

35. Krasnov, B. R. *et al.* Phylogenetic signal in module composition and species connectivity in compartmentalized host-parasite networks. *Am. Nat.* **179,** 501–511. doi:10.1086/664612 (2012).

36. Dallas, T. & Cornelius, E. Co-extinction in a host-parasite network: identifying key hosts for network stability. *Sci. Rep.* **5,** 13185. doi:10.1038/srep13185 (2015).

37. Gao, M. *et al.* Inductive Link Prediction via Interactive Learning Across Relations in Multiplex Networks. *IEEE Transactions on Computational Social Systems* **PP,** 1–13. doi:10.1109/TCSS.2022.3176928 (2022).

38. Hao, Y., Cao, X., Fang, Y., Xie, X. & Wang, S. Inductive Link Prediction for Nodes Having Only Attribute Information. *arXiv [cs.LG]* (2020).

39. Strydom, T *et al.* Graph embedding and transfer learning can help predict potential species interaction networks despite data limitations. *Methods Ecol. Evol.* **14,** 2917–2930. doi:10.1111/2041-210X.14228 (2023).

40. Brimacombe, C., Bodner, K., Michalska-Smith, M., Poisot, T. & Fortin, M.-J. Shortcomings of reusing species interaction networks created by different sets of researchers. *PLoS Biol.* **21,** e3002068. doi:10.1371/journal.pbio.3002068 (2023).

41. Poisot, T. Guidelines for the prediction of species interactions through binary classification. *Methods Ecol. Evol.* **14,** 1333–1345. doi:10.1111/2041-210x.14071 (2023).

42. Pilosof, S., Fortuna, M. A., Vinarski, M. V., Korallo-Vinarskaya, N. P. & Krasnov, B. R. Temporal dynamics of direct reciprocal and indirect effects in a host-parasite network. *J. Anim. Ecol.* **82,** 987–996. doi:10.1111/1365-2656.12090 (2013).

43. Pilosof, S., Porter, M. A., Pascual, M. & Kéfi, S. The multilayer nature of ecological networks. *Nat Ecol Evol* **1,** 0101. doi:10.1038/s41559-017-0101 (2017).

44. Ponisio, L. C. & M'Gonigle, L. K. Coevolution leaves a weak signal on ecological networks. *Ecosphere* **8,** e01798. doi:10.1002/ecs2.1798 (2017).

45. Pichon, B., Le Goff, R., Morlon, H. & Perez-Lamarque, B. Telling mutualistic and antagonistic ecological networks apart by learning their multiscale structure. *Methods Ecol Evol* **15,** 1113–1128. doi:10.1111/2041-210X.14328 (2024).

46. Song, C. & Saavedra, S. Telling ecological networks apart by their structure: An environment-dependent approach. *PLoS Comput. Biol.* **16,** e1007787. doi:10.1371/journal.pcbi.1007787 (2020).

47. CaraDonna, P. J. *et al.* Seeing through the static: the temporal dimension of plant-animal mutualistic interactions. *Ecol. Lett.* **24,** 149–161. doi:10.1111/ele.13623 (2021).

48. Yates, L. A., Aandahl, Z., Richards, S. A. & Brook, B. W. Cross validation for model selection: A review with examples from ecology. *Ecol. Monogr.* **93.** doi:10.1002/ecm.1557 (2023).

49. Bergstra, J. & Bengio, Y. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.* **13** (2012).

50. Roberts, D. R. *et al.* Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure. *Ecography* **40,** 913–929. doi:10.1111/ecog.02881 (2017).

51. Pedregosa, F *et al.* Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* (2011).

52. Chen, T. & Guestrin, C. *XGBoost: A Scalable Tree Boosting System* in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* San Francisco, California, USA (Association for Computing Machinery, New York, NY, USA, 2016), 785–794. doi:10.1145/2939672.2939785.

53. Olesen, J. M. *et al.* Missing and forbidden links in mutualistic networks. *Proc. Biol. Sci.* **278,** 725–732. doi:10.1098/rspb.2010.1371 (2011).

54. Lü, L. & Zhou, T. Link prediction in complex networks: A survey. *Physica A: Statistical Mechanics and its Applications* **390,** 1150–1170. doi:10.1016/j.physa.2010.11.027 (2011).

55. Hagberg, A., Swart, P. J. & Schult, D. A. *Exploring network structure, dynamics, and function using NetworkX* research rep. LA-UR-08-05495; LA-UR-08-5495 (Los Alamos National Laboratory (LANL), Los Alamos, NM (United States), 2008).

56. Csárdi, G & Nepusz, T. The igraph software package for complex network research. *InterJournal Complex Systems* **1695,** 1695 (2006).

57. Dormann, C. F., Fründ, J., Blüthgen, N. & Gruber, B. Indices, graphs and null models: analyzing bipartite ecological networks. *The Open Ecology Journal* **2,** 7–24. doi:10.2174/1874213000902010007 (2009).

58. Harris, C. R. *et al.* Array programming with NumPy. *Nature* **585,** 357–362. doi:10.1038/s41586-020-2649-2 (2020).

59. Team, T. P. D. Pandas development Pandas-dev/pandas: Pandas. *Zenodo* **21,** 1–9 (2020).

60. McKinney, W. *Data Structures for Statistical Computing in Python* in *Proceedings of the 9th Python in Science Conference* Python in Science ConferenceAustin, Texas (SciPy, 2010). doi:10.25080/majora-92bf1922-00a.

61. Thai-Nghe, N., Gantner, Z. & Schmidt-Thieme, L. *Cost-sensitive learning methods for imbalanced data* in *The 2010 International Joint Conference on Neural Networks (IJCNN)* (IEEE, 2010), 1–8. doi:10.1109/IJCNN.2010.5596486.

62. Yang, Y., Lichtenwalter, R. N. & Chawla, N. V. Evaluating link prediction methods. *Knowl. Inf. Syst.* **45,** 751–782. doi:10.1007/s10115-014-0789-0 (2015).

63. Holland, P. W., Laskey, K. B. & Leinhardt, S. Stochastic blockmodels: First steps. *Soc. Networks* **5,** 109–137. doi:10.1016/0378-8733(83)90021-7 (1983).

64. Larremore, D. B., Clauset, A. & Jacobs, A. Z. Efficiently inferring community structure in bipartite networks. *Phys. Rev. E Stat. Nonlin. Soft Matter Phys.* **90,** 012805. doi:10.1103/PhysRevE.90.012805 (2014).

65. Terry, C. *CassandRa: Finds missing links and metric confidence intervals in ecological bipartite networks (v0.2.0)* comp. software. Version 0.2.0. 2024. doi:10.32614/CRAN.package.cassandRa.

Supplementary Information

Supplementary results

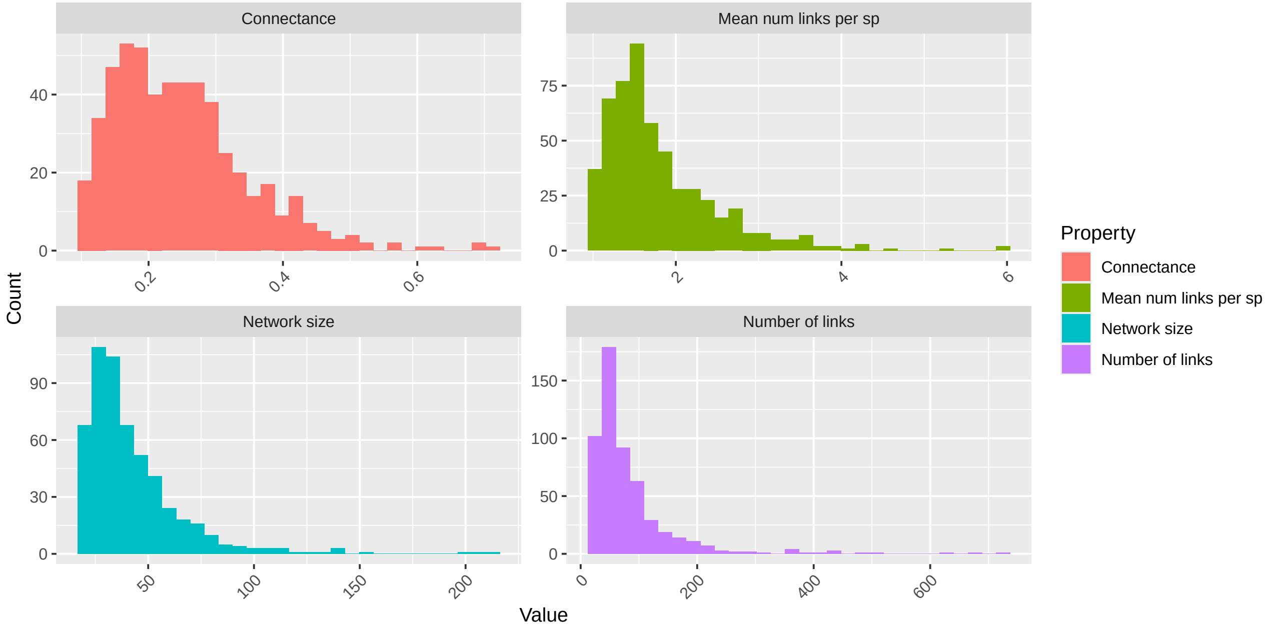

Fig. S1: Distributions of basic network properties of the networks included in the analysis. Number of links is the average number of links per species.

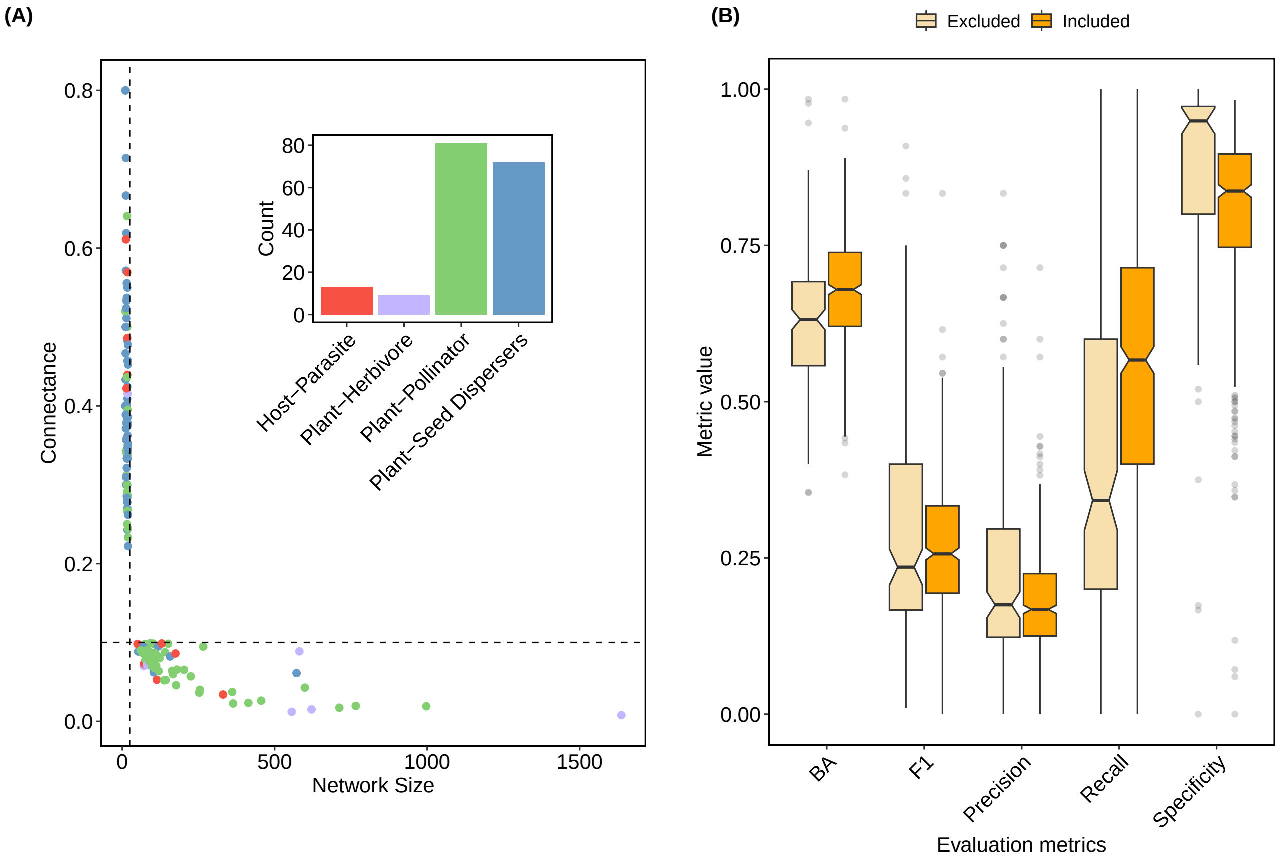

Fig. S2: Networks excluded from analysis. (A) The relationship between network size (x-axis) and connectance (y-axis) for networks (data points) excluded from the training set. Networks were excluded if their connectance was < 0.1 (below the horizontal line) or their size was < 25 species (to the left of the vertical line). The inset summarizes the number of excluded networks. Including the large amount of plant-pollinator networks would reduce the model's performance due to their sparseness, as demonstrated in the manuscript. **(B)** Comparison of test performance between included and excluded networks. Data for included networks are as presented in Fig. 2 of the main text. Each data point underlying the boxplots represents an individual network. A significant difference in medians is indicated by non-overlapping notches. Overall, the excluded networks exhibit poorer test performance, with higher specificity attributed to their extreme sparsity, facilitating the detection of true negatives (0's).

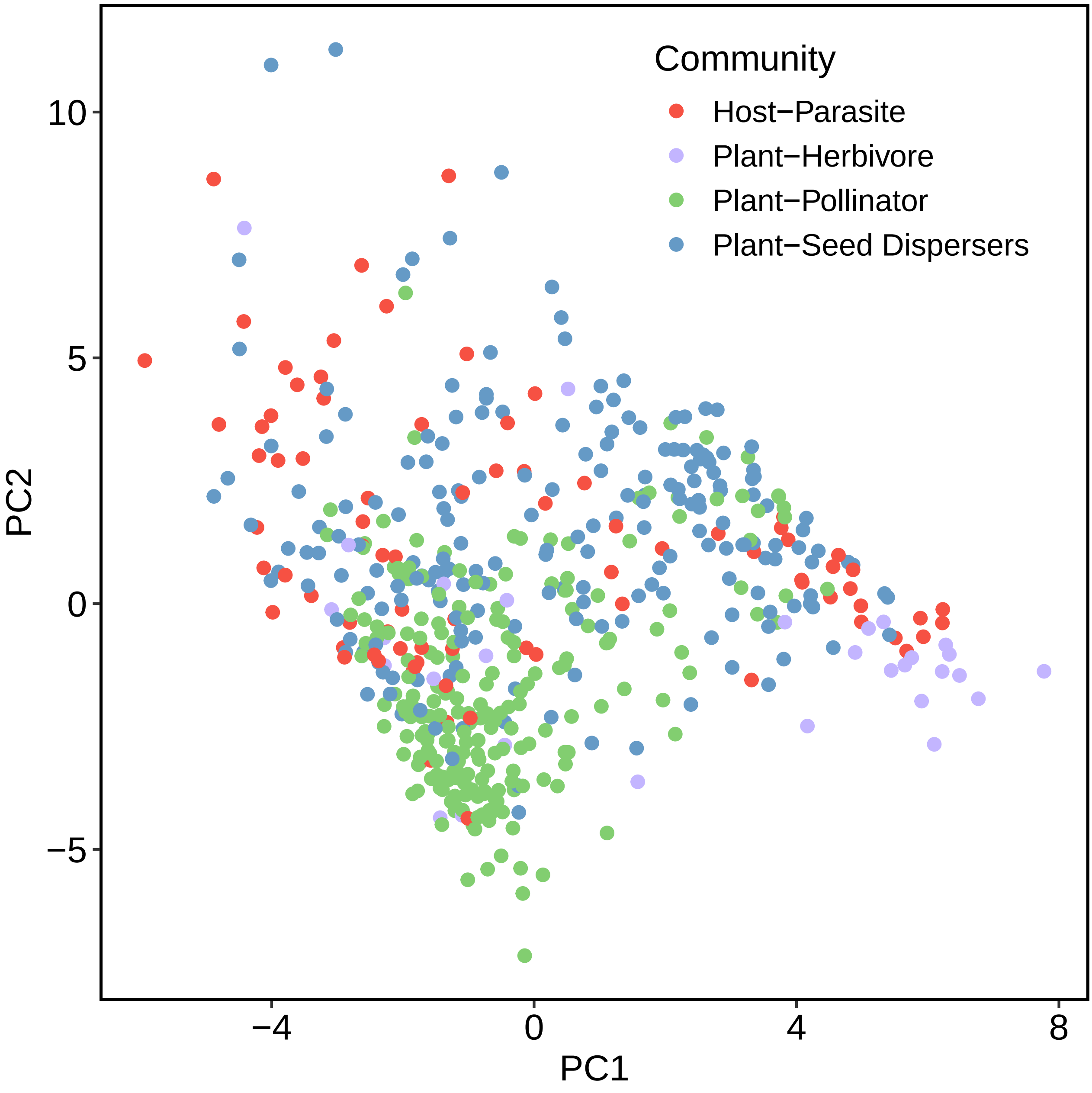

Fig. S3: Variation in network structure. The PCA shows no clear subgrouping of networks by their class, reflecting previous studies using the same data set [1,2]. The PCs were calculated using the topological features we used in the link prediction model.

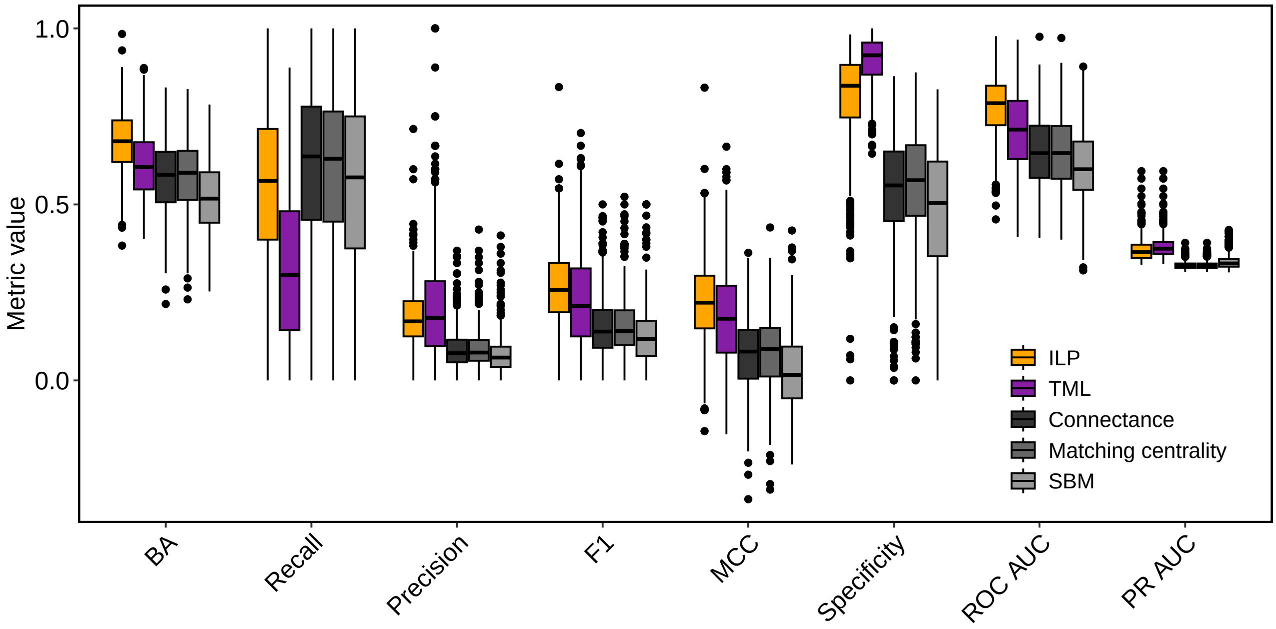

Fig. S4: Comparison of model performance. The inductive link prediction model was trained and tested on networks from all communities. We compared ILP to three transductive previously published models (stochastic block model, connectance (C) and matching centrality (MC) [3]), and to a transductive machine learning model that we trained (TLP). Each boxplot is a distribution of an evaluation metric using a 0.5 classification threshold, bedsides the PR AUC. The boxplots show the interquartile range (box), median (central line), data within 1.5 times the interquartile range (whiskers), and potential outliers (individual points beyond whiskers). Each data point is a network and boxplots contain networks from all the five outer folds. BA is balanced accuracy. See definitions of evaluation metrics in the Methods.

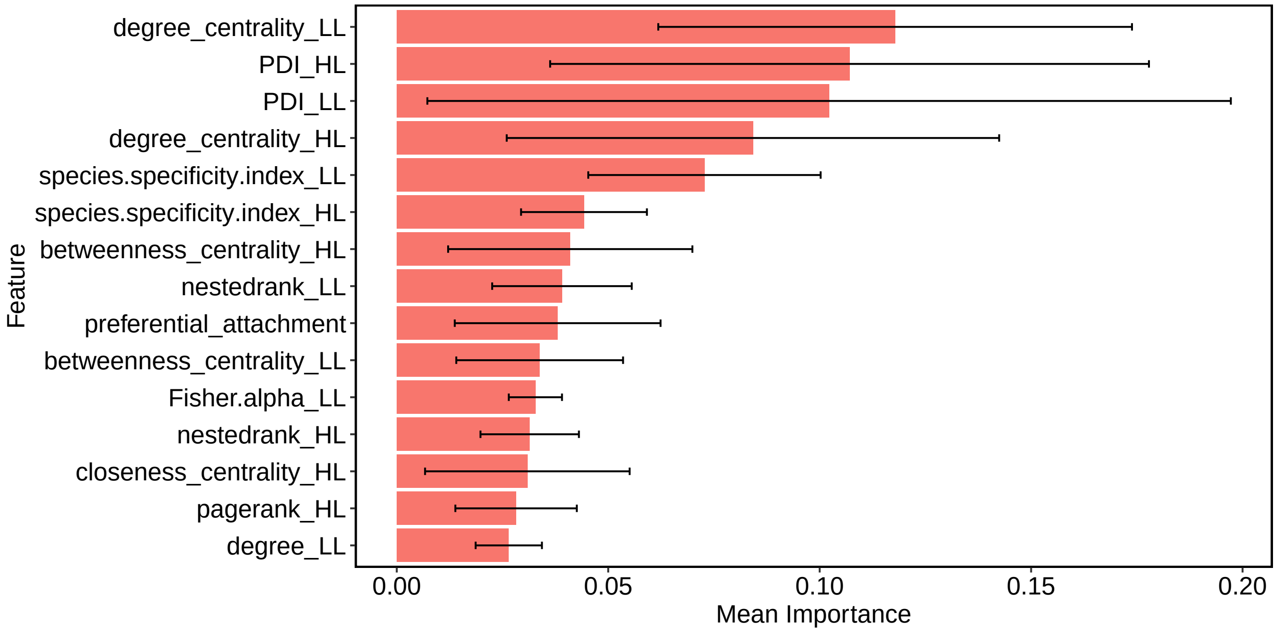

Fig. S5: Feature importance. Each bar presents the average value of feature importance across the five folds for the random forest model. Error bars depict minimum and maximum. HL are higher-level trophic species that consume resources (pollinators, parasites, seed dispersers, herbivores). LL are lower-level trophic species (resources) (plants, hosts, seeds). Features are invariably related to degree. The link-level feature preferential attachment (the multiplication of the two interacting species' degrees). Feature importance was computed based on the average decrease in Gini impurity across all trees.

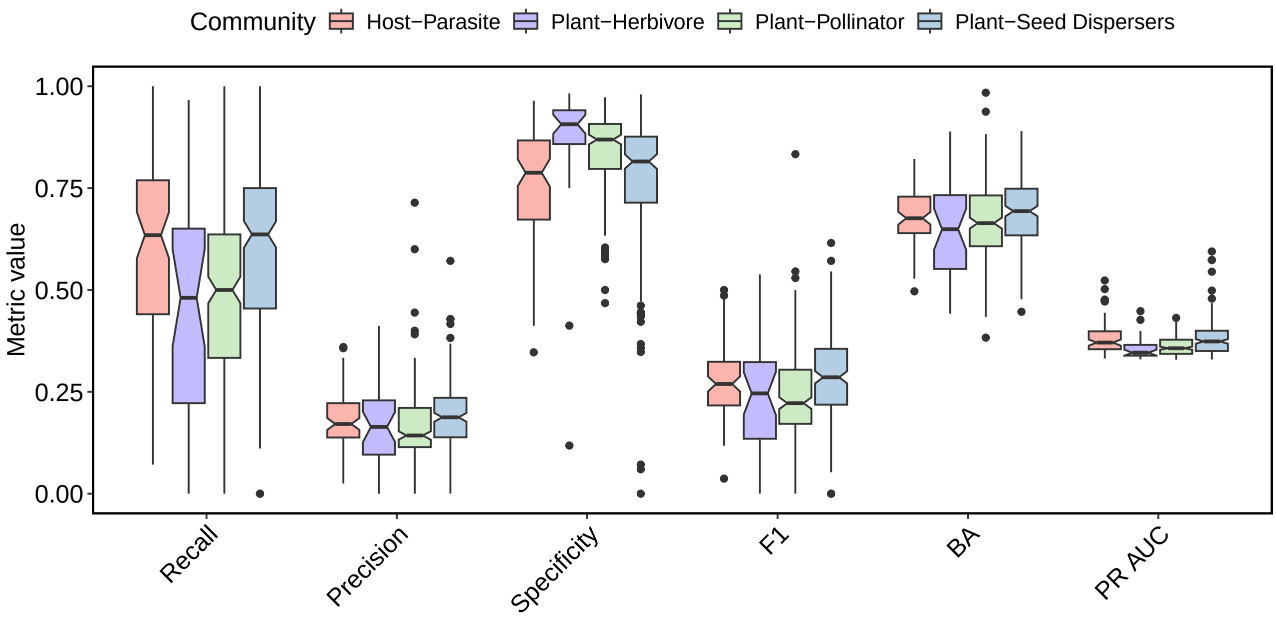

Fig. S6: Distributions of evaluation metrics per test community type. The model was trained on networks from all communities, and tested separately on each community type. Each data point is a network and box plots contain networks from all the five outer folds. The boxplots show the interquartile range (box), median (central line), data within 1.5 times the interquartile range (whiskers), and potential outliers (individual points beyond whiskers)

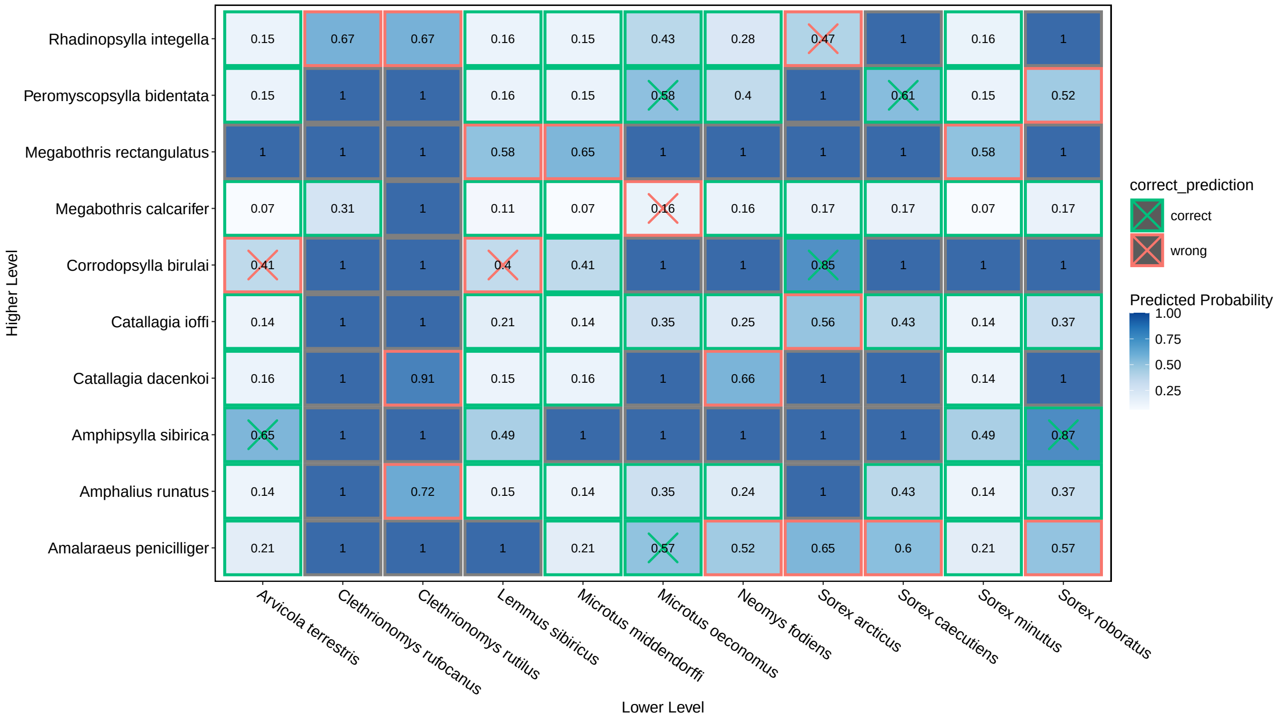

Fig. S7: Link prediction example for a host-parasite network. Values inside matrix cells are the predicted link probabilities. The sub-sampled links are marked with X. Cells above the 0.5 threshold are classified as links. Correct and wrong classifications are colored green and red, respectively. Therefore, true positives are green with X, true negatives are green, false positives are red (no X) and false negatives are red with X.

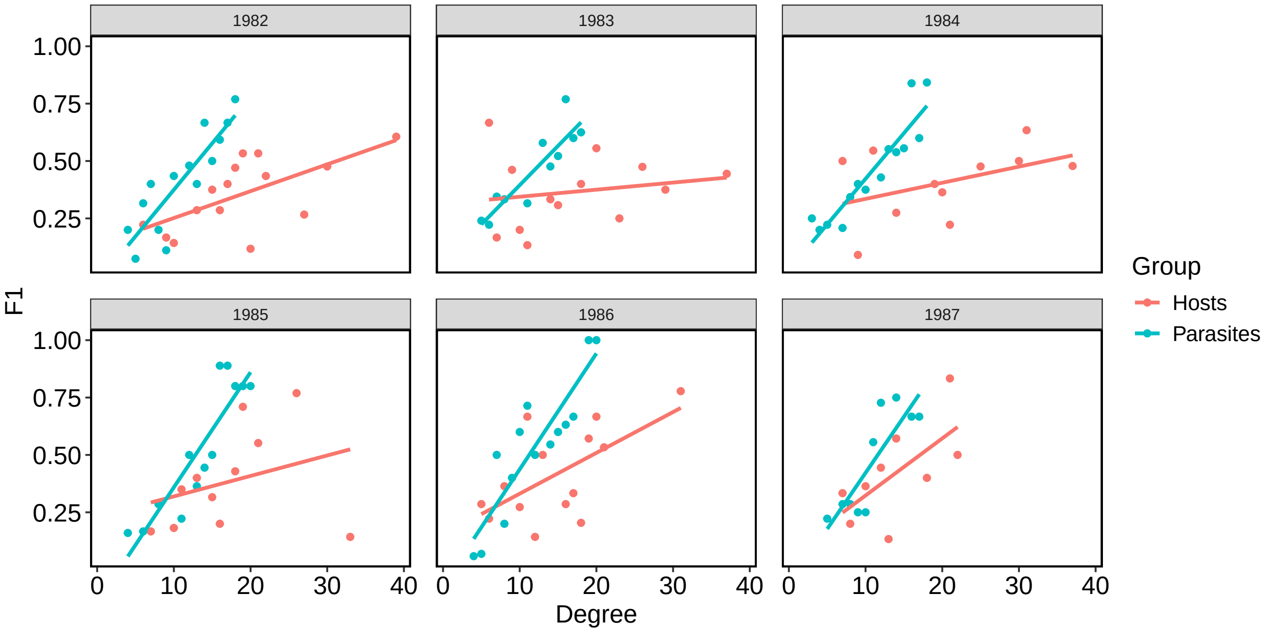

Fig. S8: Correlation between degree and F1 peer year. The plots reflect the results in Fig 7.

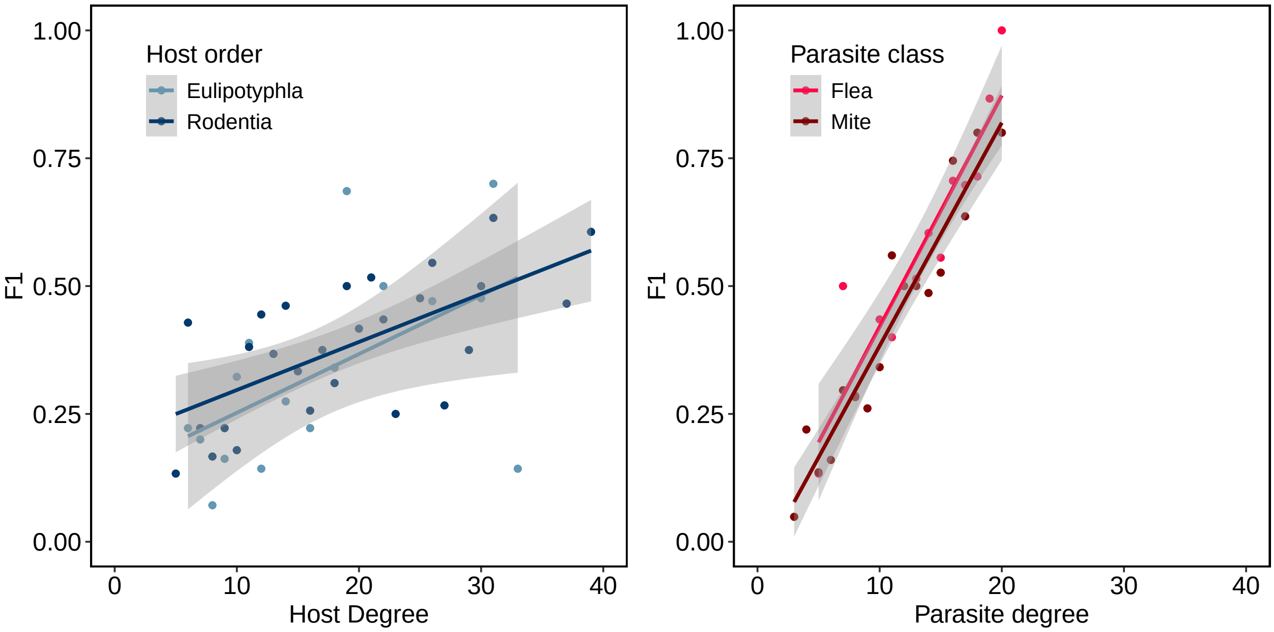

Fig. S9: Correlation between degree and F1. The plots reflect the results in Fig 7, separately for host and parasite groups.

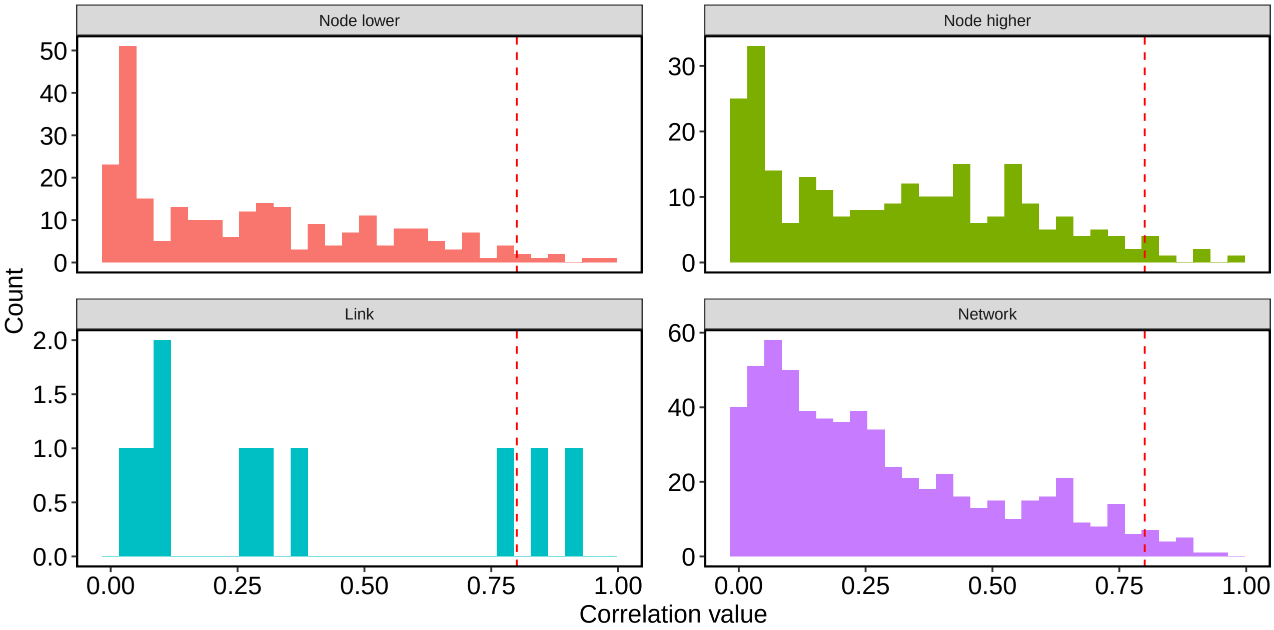

Fig. S10: A histogram of feature correlation values. We only kept one of the features when the correlation was above the 0.8 threshold (indicated with a dashed red line. Overall, correlations were below 0.5.

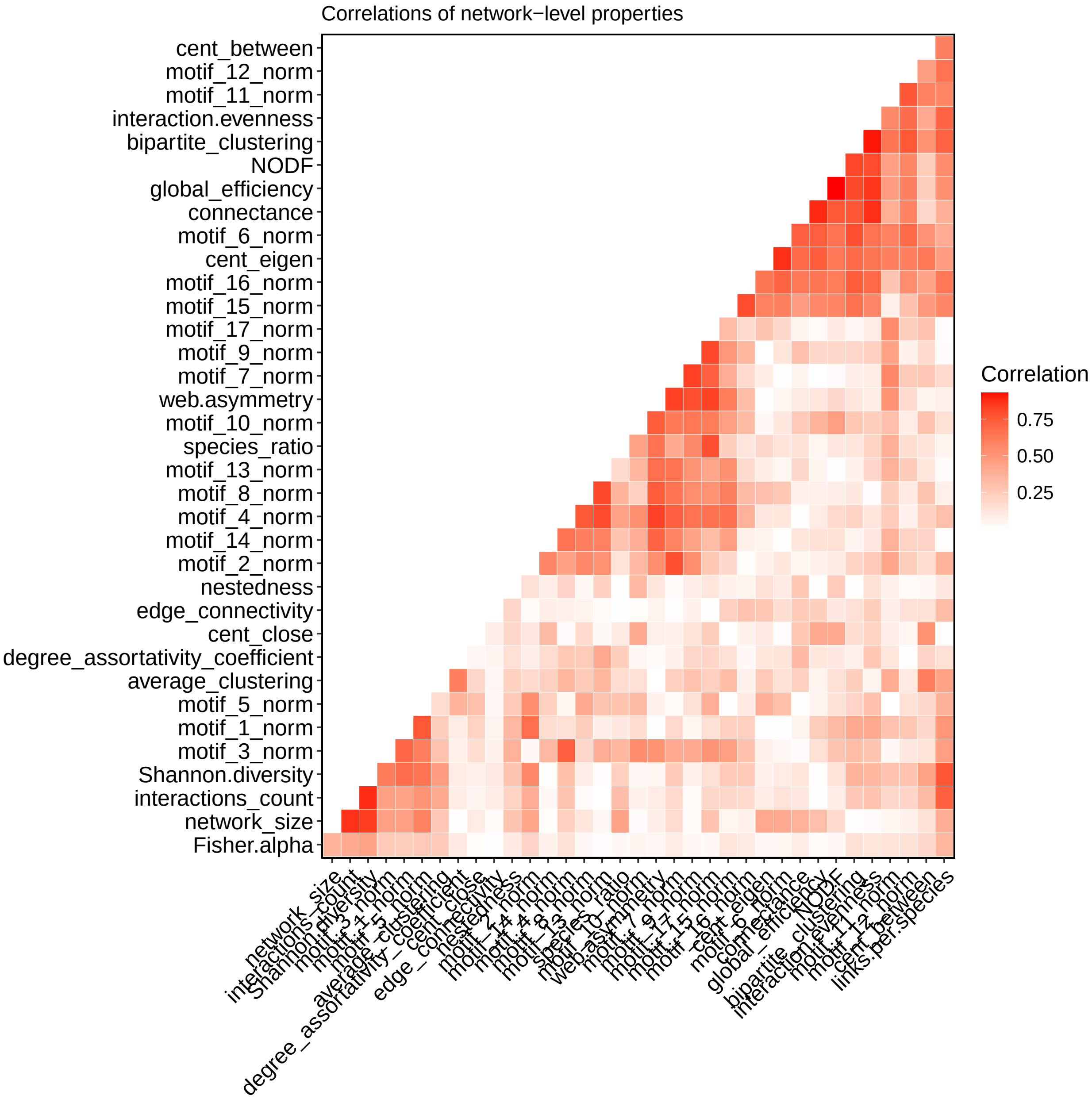

Fig. S11: Correlation between features at the network level.

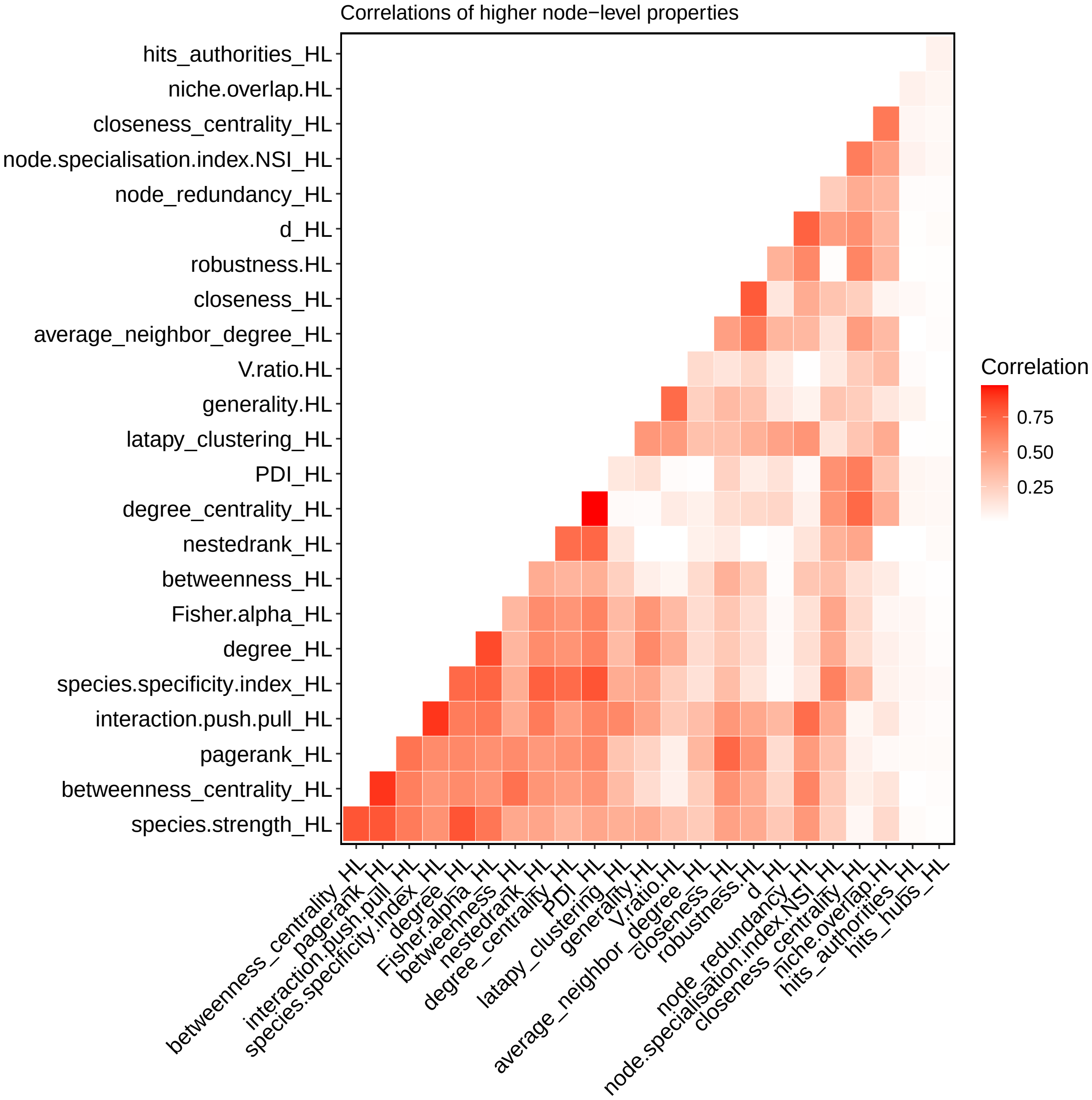

Fig. S12: Correlation between features at the node level for the higher-trophic level (e.g., parasites, pollinators.

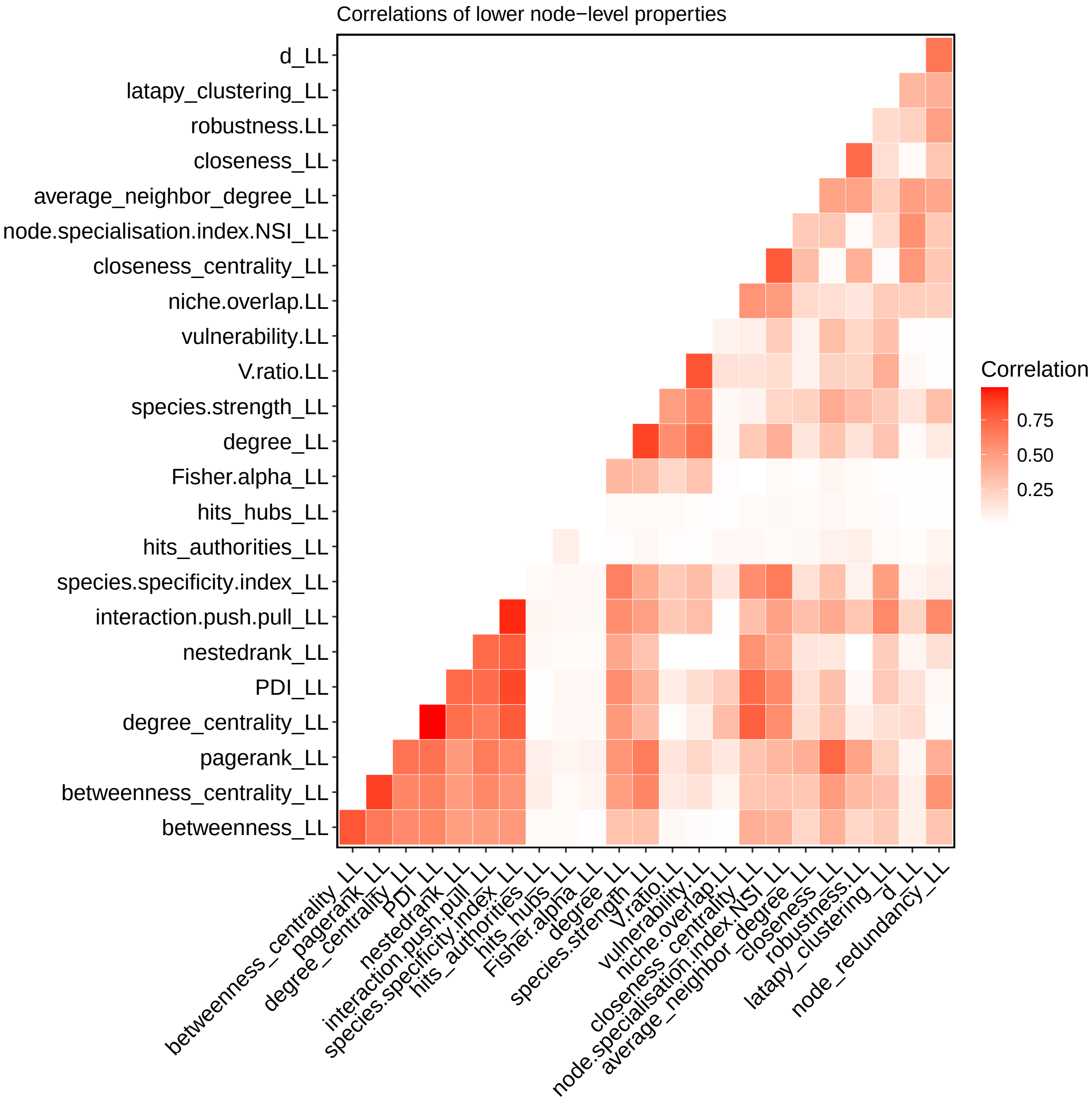

Fig. S13: Correlation between features at the node level for the lower-trophic level (e.g., hosts, plants.

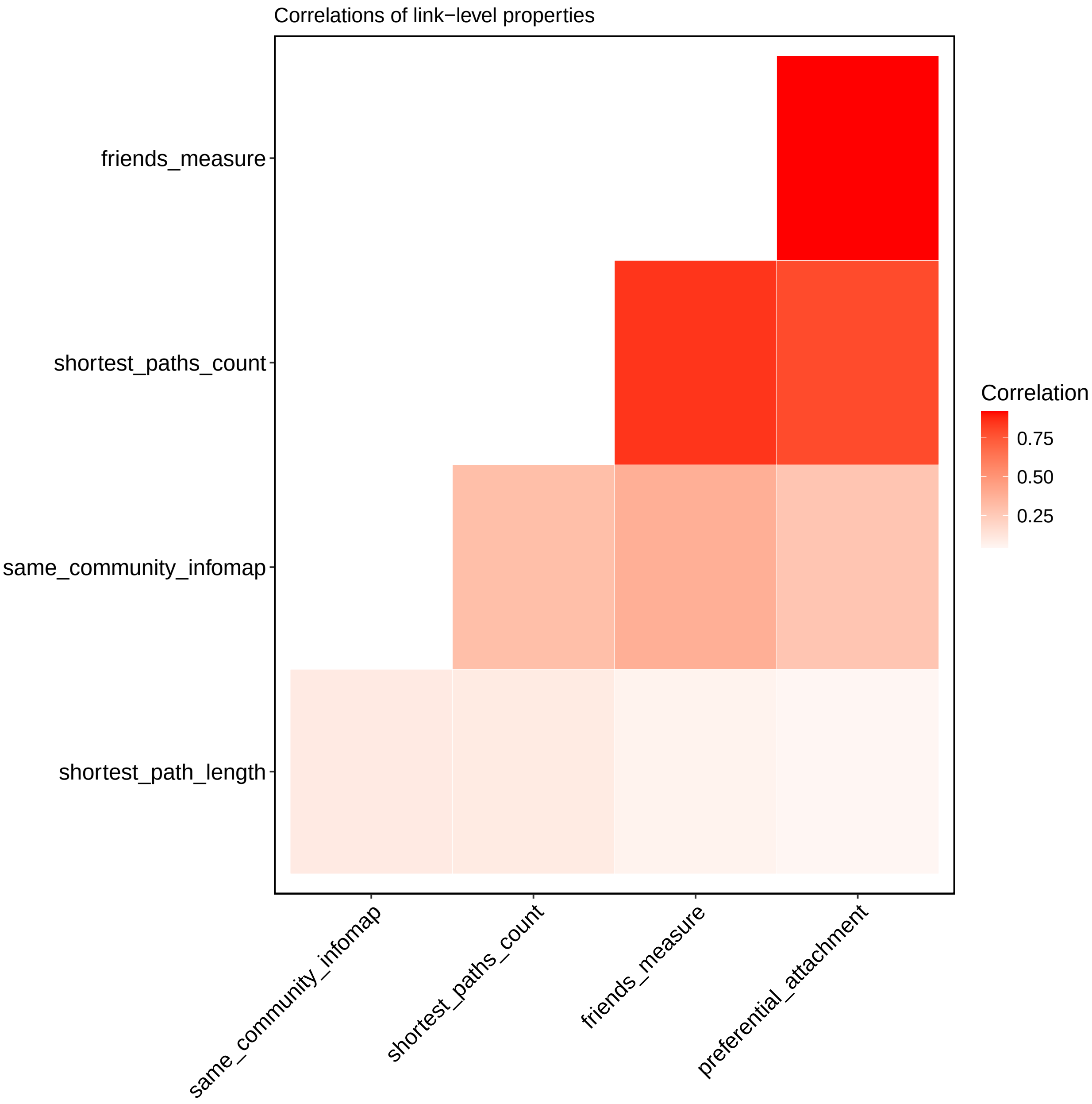

Fig. S14: Correlation between features at the link level.

Table S1: Summary of network properties. The table provides an overview of the network properties for different ecological communities, detailing each variable's mean, median, and range values. N: number of nodes; L_e: number of existing links; L_{ne}: number of nonexisting links; C: connectance, $C = L_e/(L_e{+}L_{ne})$

Variable	Community	Mean	Median	Range
N	Host-Parasite	37.58	33	20-137
	Plant-Herbivore	42.53	37	21-78
	Plant-Pollinator	43.36	36	20-205
	Plant-Seed Dispersers	43.96	36	20-213
L_e	Host-Parasite	82.86	63	19-490
	Plant-Herbivore	67.69	42.5	22-294
	Plant-Pollinator	72.94	52	21-631
	Plant-Seed Dispersers	98.59	72	21-720
L_{ne}	Host-Parasite	1622.08	991.5	349-18279
	Plant-Herbivore	1997.22	1332.5	409-5996
	Plant-Pollinator	2394.52	1260	363-41600
	Plant-Seed Dispersers	2585.2	1238	362-44687
C	Host-Parasite	0.28	0.25	0.11-0.61
	Plant-Herbivore	0.2	0.165	0.1-0.47
	Plant-Pollinator	0.22	0.2	0.1-0.42
	Plant-Seed Dispersers	0.28	0.27	0.1-0.71

Table S2: Kruskal-Wallis and Dunn's test results for pairwise comparisons of metrics between communities. The KW test was performed for each metric separately, with communities as factor levels. All Comparisons were significant, and we therefore performed a Dunn post-hoc test (p value adjusted using Bonferroni correction) for all of them. Note that the median values are close for some comparisons despite being statistically significant. A visualization of the values and their distributions is in the main text, in Fig. 4B.

Metric	KW statistic	Community 1	Community 2	Median 1	Median 2	P value (adjusted)
Recall	39.03	HP	PH	0.63	0.48	0.019
Recall	39.03	HP	PP	0.63	0.50	0.001
Recall	39.03	HP	PSD	0.63	0.64	1.000
Recall	39.03	PH	PP	0.48	0.50	1.000
Recall	39.03	PH	PSD	0.48	0.64	0.004
Recall	39.03	PP	PSD	0.50	0.64	0.000
Precision	20.54	HP	PH	0.17	0.16	1.000
Precision	20.54	HP	PP	0.17	0.14	0.060
Precision	20.54	HP	PSD	0.17	0.19	1.000
Precision	20.54	PH	PP	0.16	0.14	1.000
Precision	20.54	PH	PSD	0.16	0.19	0.415
Precision	20.54	PP	PSD	0.14	0.19	0.000
Specificity	53.43	HP	PH	0.79	0.91	0.000
Specificity	53.43	HP	PP	0.79	0.87	0.000
Specificity	53.43	HP	PSD	0.79	0.82	1.000
Specificity	53.43	PH	PP	0.91	0.87	0.088
Specificity	53.43	PH	PSD	0.91	0.82	0.000
Specificity	53.43	PP	PSD	0.87	0.82	0.000
F1	27.82	HP	PH	0.27	0.25	0.611
F1	27.82	HP	PP	0.27	0.22	0.023
F1	27.82	HP	PSD	0.27	0.29	1.000
F1	27.82	PH	PP	0.25	0.22	1.000
F1	27.82	PH	PSD	0.25	0.29	0.099
F1	27.82	PP	PSD	0.22	0.29	0.000
BA	9.12	HP	PH	0.68	0.65	1.000
BA	9.12	HP	PP	0.68	0.66	1.000
BA	9.12	HP	PSD	0.68	0.69	1.000
BA	9.12	PH	PP	0.65	0.66	1.000
BA	9.12	PH	PSD	0.65	0.69	0.275
BA	9.12	PP	PSD	0.66	0.69	0.040
PR AUC	47.86	HP	PH	0.37	0.35	0.000
PR AUC	47.86	HP	PP	0.37	0.36	0.000
PR AUC	47.86	HP	PSD	0.37	0.37	1.000
PR AUC	47.86	PH	PP	0.35	0.36	0.820
PR AUC	47.86	PH	PSD	0.35	0.37	0.000
PR AUC	47.86	PP	PSD	0.36	0.37	0.000

S1 Supplementary notes on methods

S1.1 Models applied

Selecting a machine learning model a priori can be challenging [4]. There is no one-size-fits-all solution; different models have different strengths and weaknesses, and knowing which model will perform better is often impossible. Some models may perform well on particular data types, while others may perform better on different ones. Hence, model selection often requires some experimentation. In practice, trying multiple models and comparing their performance is often a good idea before deciding on a final model. Furthermore, using a model ensemble—a technique in which multiple models are combined—can often lead to better results than a single model [5]. In this study, we tried multiple models and their ensemble. For the ensemble, we averaged the probabilities for each prediction. The models performed similarly overall, and their ensemble did not improve the results (Fig. S15). Therefore, in the main text, we present results for random forest. In this section, we describe the models we used. We present model hyperparameters in Table S3.

To understand how these models work, it is necessary to first explain the terms bagging and boosting [6], two popular ensemble learning techniques. The key difference between bagging and boosting lies in how they combine multiple models to make predictions. Bagging, which stands for bootstrap aggregating, is a parallel ensemble technique. In bagging, multiple base models (multiple instances of the same algorithm) are trained independently on different random subsets of the training data using bootstrap sampling (sampling with replacement). The outputs of the individual models are then combined, often through a voting mechanism or by taking the average of their predictions, to produce a single output. Bagging is often used with decision trees as it has been repetitively shown to outperform other models. Boosting, on the other hand, is a sequential ensemble technique. Boosting works by iteratively training multiple weak models on modified versions of the training data, with each subsequent model trying to correct the errors of the previous models, focusing on the examples that the previous models misclassified. The training data is re-weighted at each iteration so that the misclassified examples receive higher weights and are given more importance in subsequent iterations. The final prediction of the boosted model is a weighted combination of the predictions of all the individual models, with the weights determined by the performance of each model on the training data. More weight is given to models that achieved higher performance.

Logistic regression. This generalized linear model uses a logistic function to model a binary dependent variable. The logistic function maps the linear combination of input variables to a discrete binary value between 0 and 1, which is interpreted as the probability of the input belonging to a particular class. During training, the logistic regression algorithm finds the input variables' best parameters (coefficients) by minimizing the error between the predicted probabilities and the actual class labels in the training data. Logistic regression is easy to interpret and implement without necessarily compromising performance [7].

Random Forest. [8,9] . This is a specific version of a bagging method with decision trees. A random forest model creates a collection of decision trees and combines their predictions to produce a final output. Decision tree models create a tree-like model of decisions and their possible consequences, with each internal node representing a feature, each branch representing a decision rule, and each leaf node representing the outcome class label. The algorithm starts at the root of the tree. It recursively splits the data into subsets using a feature that provides the most information gain at that stage until it reaches a leaf node, representing the predicted class label for the input data. Each tree in the random forest ensemble is trained on a different random subset of the data and features, and the final output is the combination of the predictions of all the decision trees, usually made by either averaging the results for regression tasks or by taking a majority vote for classification tasks.

XGBoost. This gradient-boosting tree-based algorithm is designed for speed and performance [10]. Gradient boosting [11] differs from other boosting algorithms in the way it calculates the weights of the models. Specifically, gradient boosting uses the gradient descent optimization algorithm to minimize the model's loss function. At each iteration, the algorithm calculates the (negative) gradient of the loss function with respect to the predictions of the previous ensemble. It uses this gradient to adjust the weights of the new model. The negative gradient represents the direction of the steepest descent for the loss function, which is the direction that will reduce the loss the most. By fitting a new tree to the negative gradient, gradient boosting focuses on the examples misclassified in the previous ensemble and attempts to correct those errors.

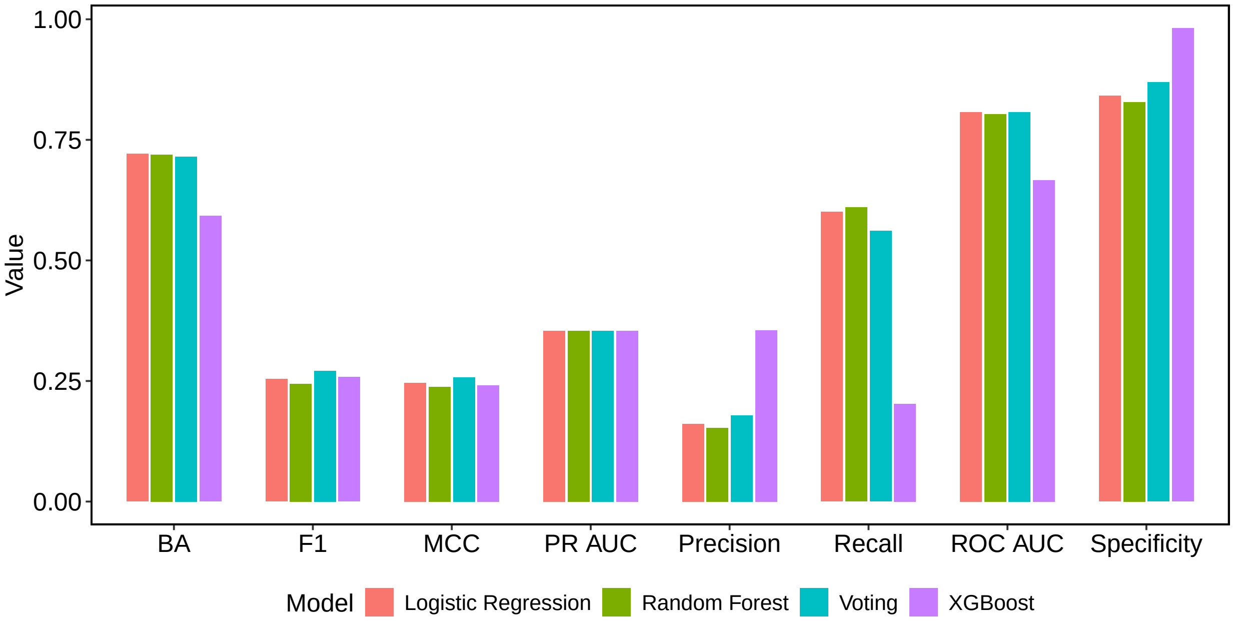

Fig. S15: Model comparison. We evaluated link prediction using various models. Average is a model ensemble. Overall, model choice did not qualitatively affect the results. Therefore, we present results of random forest throughout the main text.

Table S3: Summary of model hyperparameters. The table provides an overview of the best values for different parameters across five folds for each model. We optimized hyperparameters using a random search. We used the F1-score as the performance metric to evaluate the hyperparameter combinations. RFC: RandomForestClassifier; LR: LogisticRegression; XGB: XGBClassifier.

Model	Parameter	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Range
RFC	n_estimators	15	10	15	50	15	[10, 15, 20, 50, 100, 300]
RFC	min_samples_split	5	1	10	1	5	[1, 2, 3, 4, 5, 10]
RFC	min_samples_leaf	12	4	3	10	7	[3, 4, 5, 6, 7, 8, 9, 10, 12, 15, 20]
RFC	max_samples	0.6	0.1	0.4	0.7	0.4	[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]
RFC	max_leaf_nodes	128	16	16	16	32	[2, 4, 8, 16, 32, 64, 128]
RFC	max_features	log2	sqrt	log2	log2	log2	[’sqrt’, ’log2’]
RFC	max_depth	30	60	20	None	3	[3, 5, 7, 10, 20, 30, 40, 50, 60, None]
RFC	criterion	gini	entropy	entropy	gini	gini	[’gini’, ’entropy’]
RFC	bootstrap	True	True	True	True	True	[True]
RFC	feature_selector_k	70	40	70	50	70	[10, 20, 30, 40, 50, 70]
LR	solver	saga	saga	saga	saga	saga	[’newton-cg’, ’lbfgs’, ’sag’, ’saga’, ’liblinear’]
LR	penalty	l2	l2	l2	l2	l1	[’l1’, ’l2’, ’elasticnet’, ’none’]
LR	max_iter	300	300	300	300	300	[300]
LR	C	0.001	0.001	0.001	0.001	0.001	[100, 10, 1.0, 0.7, 0.5, 0.3, 0.1, 0.01, 0.001]
LR	feature_selector_k	50	20	40	40	50	[10, 20, 30, 40, 50, 70]
XGB	tree_method	hist	hist	hist	hist	hist	[’hist’]
XGB	subsample	0.5	0.2	0.5	0.9	0.8	[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0]
XGB	reg_lambda	100	0.1	100	0	1e-05	[0, 1e-05, 0.01, 0.1, 1, 100]
XGB	reg_alpha	100	0	1	100	1	[0, 1e-05, 0.01, 0.1, 1, 100]
XGB	objective	binary:logistic	binary:logistic	binary:logistic	binary:logistic	binary:logistic	[’binary:logistic’]
XGB	n_estimators	30	50	60	60	10	[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
XGB	max_depth	7	11	15	None	5	[1, 3, 5, 7, 9, 11, 13, 15, 17, 19, None]
XGB	learning_rate	0.001	0.001	0.1	0.01	0.01	[0.001, 0.01, 0.05, 0.1, 0.2, 0.3]
XGB	gamma	0	0.1	0.3	1	0.5	[0, 0.1, 0.3, 0.5, 1]
XGB	colsample_bytree	0.2	0.4	0.2	0.1	0.3	[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0]
XGB	booster	gbtree	gbtree	gbtree	gbtree	gbtree	[’gbtree’]
XGB	feature_selector_k	30	30	70	40	50	[10, 20, 30, 40, 50, 70]

S1.2 Exploration of model evaluation

In binary classification, the terms "positive" and "negative" refer to the two possible outcomes, where the positive class is typically used to denote the class of interest. Here, the positive class is the presence of a link. There are four possible prediction outcomes defined for binary classification models, computed by comparing the predicted and actual values of the model's outputs (Fig. S7):

- ***True Positives (TP)***: instances correctly predicted as the positive class. That is, sub-sampled links that were correctly predicted as existing links.
- ***True Negatives (TN)***: instances correctly predicted as the negative class. That is, non-existing links that were correctly predicted as non-existing links.
- ***False Positives (FP)***: instances falsely predicted as the positive class. That is, non-existing links that were incorrectly predicted as existing links. This is also known as a Type I error.
- ***False Negatives (FN)***: instances falsely predicted as the negative class. That is, sub-sampled links that were incorrectly predicted as non-existing links. This is also known as a Type II error.

The output of machine learning models is a probability of a link (Fig. S7). This continuous output is transformed into a categorical (positive / negative) prediction via a threshold. The threshold is typically 0.5, with values > 0.5 considered a link (and values below as a no-link). Therefore, classification into the four categories above depends on the threshold applied. The relationship between true positives, false positives, false negatives, and true negatives are summarized in a *confusion matrix* from which the following metrics are calculated.

Recall or sensitivity: The proportion of correctly predicted positive instances out of all actual positive instances. Recall is important when false negatives are costly. For example, not predicting missing links can have consequences for conservation.

$$\frac{TP}{TP + FN} \tag{S1}$$

Precision: Recall is typically complemented by precision. Precision is the proportion of correctly predicted positive instances out of all positive predictions:

$$\frac{TP}{TP + FP} \tag{S2}$$

Precision is important when false positives are costly. For example, falsely predicting an interaction can lead ecologists to spend research efforts trying to validate a forbidden link in nature.

PR AUC: There is a tradeoff between precision and recall, which depends on the prediction threshold. To evaluate this tradeoff, the area under the PR curve provides a single number that summarizes the overall performance of a model across all possible classification thresholds. Like the ROC-AUC curve, the PR curve is calculated across all thresholds.

F1-score: Another way to evaluate the precision-recall tradeoff is via a balanced measure of performance that reflects the classifier's ability to identify true positive instances while avoiding false positives. The F1-score is defined as the harmonic mean between precision and recall:

$$F1 = \frac{2 * precision * recall}{precision + recall} \tag{S3}$$

Specificity: The proportion of correctly predicted negative instances out of all actual negative instances:

$$\frac{TN}{TN + FP} \tag{S4}$$

While recall and precision focus on true positives, specificity focuses on the retrieval of true negatives.

Balanced accuracy: Balanced accuracy aims to balance the prediction ability for links and no-links, particularly in situations where the classes are imbalanced. It is the average of the true positive rate (sensitivity or recall) and the true negative rate (specificity):

$$\frac{recall + specificity}{2} \tag{S5}$$

MCC: The Matthews Correlation Coefficient (MCC) takes into account the balance ratios of the four confusion matrix categories (TP, TN, FP, FN), and is, therefore, a balanced measure that can be used even if the classes are of very different sizes [12].

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{S6}$$

MCC values range from -1 to $+1$. A coefficient of $+1$ indicates a perfect prediction, 0 indicates no better than a random prediction, and -1 indicates total disagreement between prediction and observation.

Given ecological networks' imbalanced and noisy nature, we explored how the classification threshold affects the PR tradeoff. As a first step, we plotted the link probabilities generated by the model. The non-existing links were generally correctly classified ($\approx 0.83\%$ out of 183K non-links; Fig. S16A). This is expected in imbalanced data sets. However, for the withheld links, only $\approx 0.63\%$ out of 9K were classified correctly (Fig. S16B). This mismatch underlies the tradeoff between precision and recall. Further exploration of the precision-recall tradeoff (Fig. S17) indicated that there is no apparent threshold to choose from, and so we decided to use the common value of 0.5. In a more detailed examination of each community separately, as shown in Fig. S18, we observed a pronounced right-skewed distribution of the sub-sampled links. Specifically, there was a more apparent separation between the two link types in both host-parasite and plant-seed disperser networks.

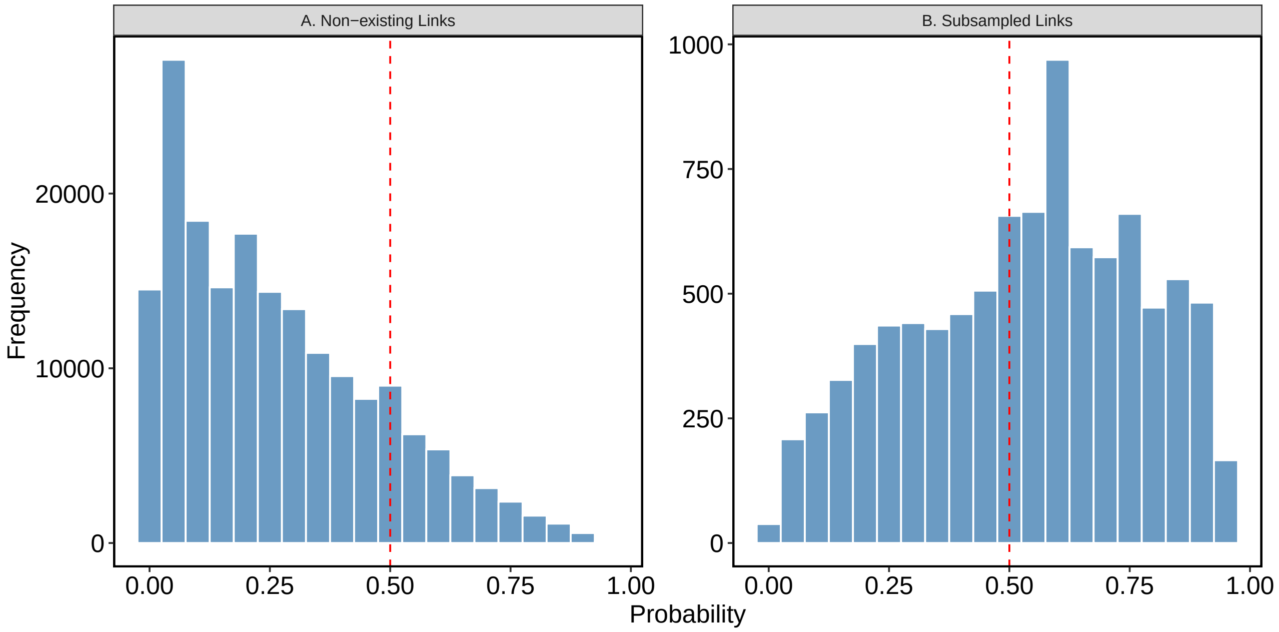

Fig. S16: Distribution of link probabilities obtained from the model. The histograms depict the distribution of the predicted probabilities for our binary classification task, representing the two classes in the test set: non-existing links (A) and sub-sampled links (B). The x-axis represents the predicted probabilities ranging from 0 to 1, while the y-axis represents the frequency of the observations. The dashed red line represents the decision threshold of 0.5.

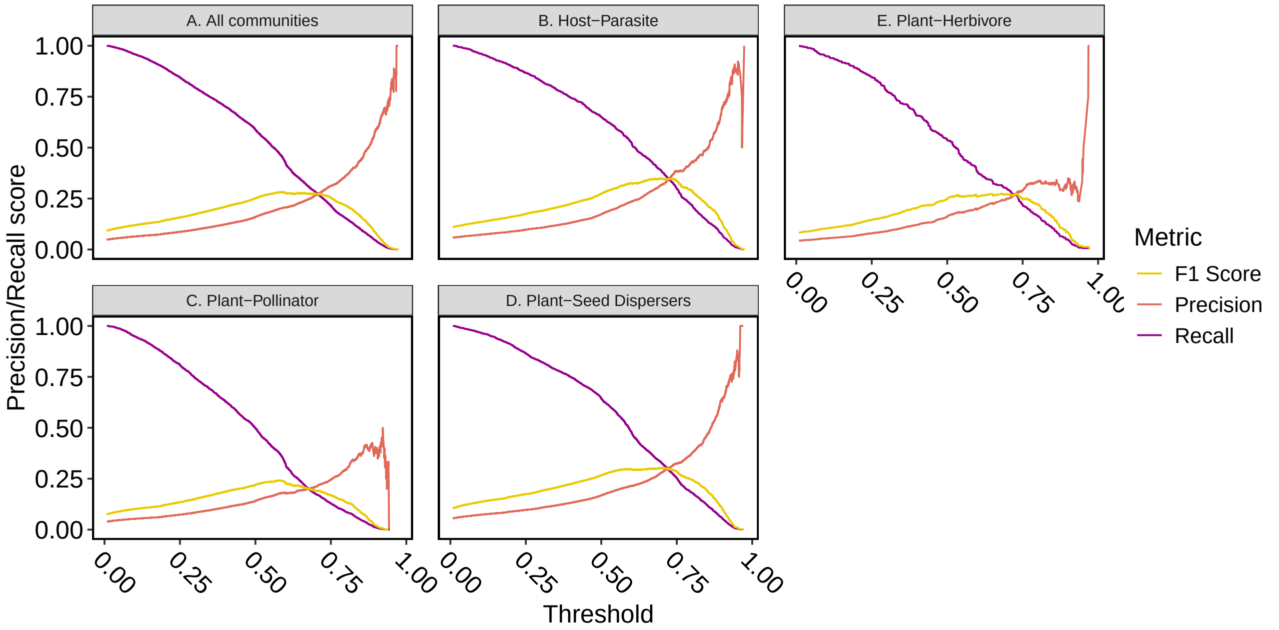

Fig. S17: The precision-recall tradeoff as a function of classification threshold. The tradeoff is presented when testing on all communities (A) or per community type (B-D). Each data point on the curve corresponds to a distinct cutoff threshold value (x-axis).

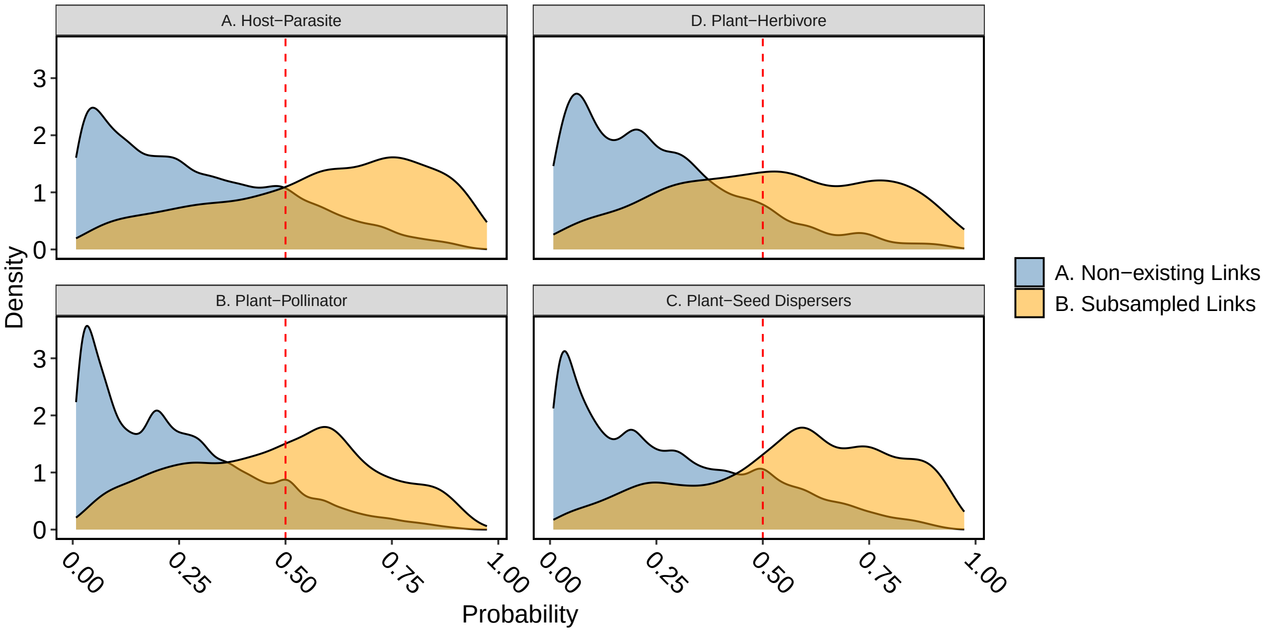

Fig. S18: Distribution of link probabilities across different ecological communities. The kernel density estimate (KDE) curves of the three communities: (A) Host-parasite networks, (B) Plant-pollinator networks, and (C) plant-speed disperser networks. The distributions for non-existing links and sub-sampled links are depicted in blue and orange, respectively. The x-axis represents the predicted probabilities ranging from 0 to 1, while the y-axis denotes the density estimation of the observations. A dashed red line marks the decision threshold of 0.5. The overlap between the blue and orange distributions represents areas of prediction ambiguity. In regions where non-existing links (blue) surpass the decision threshold, false positives emerge, negatively influencing precision. Conversely, in the region where sub-sampled links (orange) fall below the threshold, it results in false negatives, negatively influencing recall.

S1.3 Exploration of link withholding

Link removal strategies could affect evaluation outcomes. Due to the binary nature of our data we tested three link removal scenarios: (1) *Uniform removal*: This scenario assumes equal probability of removal for all interactions, as implemented in the initial submission. (2) *High-degree biased sampling*: This scenario preferentially retains interactions involving species with many partners. It reflects ecological survey biases where interactions of abundant or generalist species are more likely to be observed due to greater detectability or sampling effort. (3) *Low-degree biased sampling:* This scenario preferentially retains interactions involving species with fewer partners, simulating sampling efforts that might focus on rare or specialized interactions. The high-degree biased sampling has lower performance than the other two scenarios because it has a lower recall (Fig. S19). Therefore, when missing links are those of generalist species, the model will have lower prediction abilities. This result is logical because those interactions contain most of the information in the network. However, such scenario is unlikely because typically generalist species are those that are more easily observed. Therefore, our model does perform well under the typical low-degree biased sampling. These additional analyses provide a more comprehensive assessment of our model's performance.

A detailed description of these scenarios is as follows:

Uniform sampling: Our baseline approach implements random uniform sampling, where each link has an equal probability of being included in the sampled network. This strategy assumes that missing interactions are randomly distributed throughout the network, which may occur when sampling effort is uniformly distributed across species pairs but insufficient to detect all interactions.

Low-degree bias: The low-degree biased sampling preferentially retains interactions involving species with fewer partners. For a bipartite network with species sets A and B, the probability of selecting species $i \in A$ and $j \in B$ is: $P(i) \propto \frac{1}{k_i+1} \quad \text{for } i \in A$, and $P(j) \propto \frac{1}{k_j+1} \quad \text{for } j \in B$, where k_i and k_j are the degrees of species i and j respectively. The probability of sampling an interaction between species i and j is then: $P(i,j) \propto P(i) \times P(j)$. This strategy simulates scenarios where rare or specialized interactions are more likely to be observed, which might occur when sampling efforts focus on less common species or when common interactions are overlooked.

High-degree bias: Conversely, the high-degree biased sampling preferentially retains interactions involving species with many partners. The sampling probabilities are proportional to the degrees: $P(i) \propto k_i \quad \text{for } i \in A$, and $P(j) \propto k_j \quad \text{for } j \in B$, with the interaction probability again being: $P(i,j) \propto P(i) \times P(j)$.

This approach simulates scenarios where common or generalist species and their interactions are more likely to be observed, which often occurs in ecological surveys due to the greater detectability of abundant species or sampling bias toward easily observable interactions.

Implementation For each sampling strategy, we: (1) Calculate the target number of interactions to retain based on the desired sampling fraction f. (2) For degree-biased sampling: Calculate degrees for species in each level of the network, compute normalized sampling probabilities for each species based on their degrees and then iteratively sample species pairs using their respective probabilities until reaching the target number of interactions. (3) For uniform sampling we directly sample the target number of interactions with uniform probability.

Withholding proportion For the uniform strategy, which we present in the main text, we also conducted sensitivity analysis, removing 5-30% of the links (Fig. S20). In the main text we present results for 20%, which balances the imbalance in non-links and links with choosing a higher proportion that would artificially improve the results.

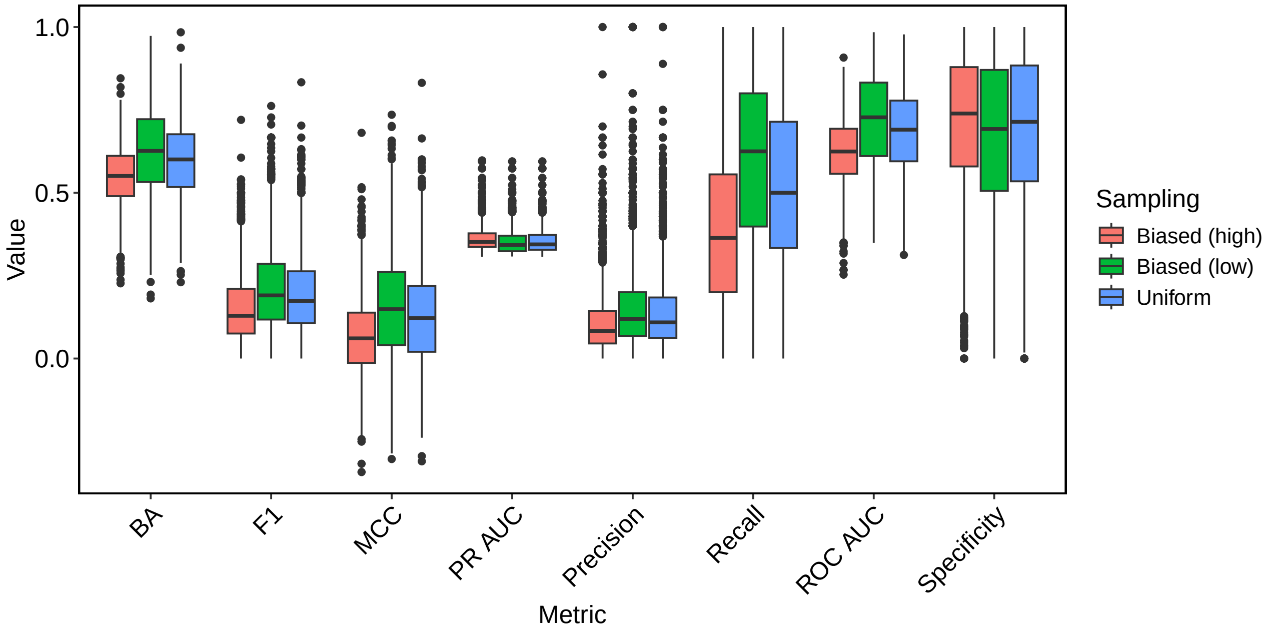

Fig. S19: Comparison of sampling strategies. *Uniform removal*: This scenario assumes equal probability of removal for all interactions. (2) *High-degree biased sampling*: This scenario preferentially retains interactions involving species with many partners. It reflects ecological survey biases where interactions of abundant or generalist species are more likely to be observed due to greater detectability or sampling effort. (3) *Low-degree biased sampling:* This scenario preferentially retains interactions involving species with fewer partners, simulating sampling efforts that might focus on rare or specialized interactions.

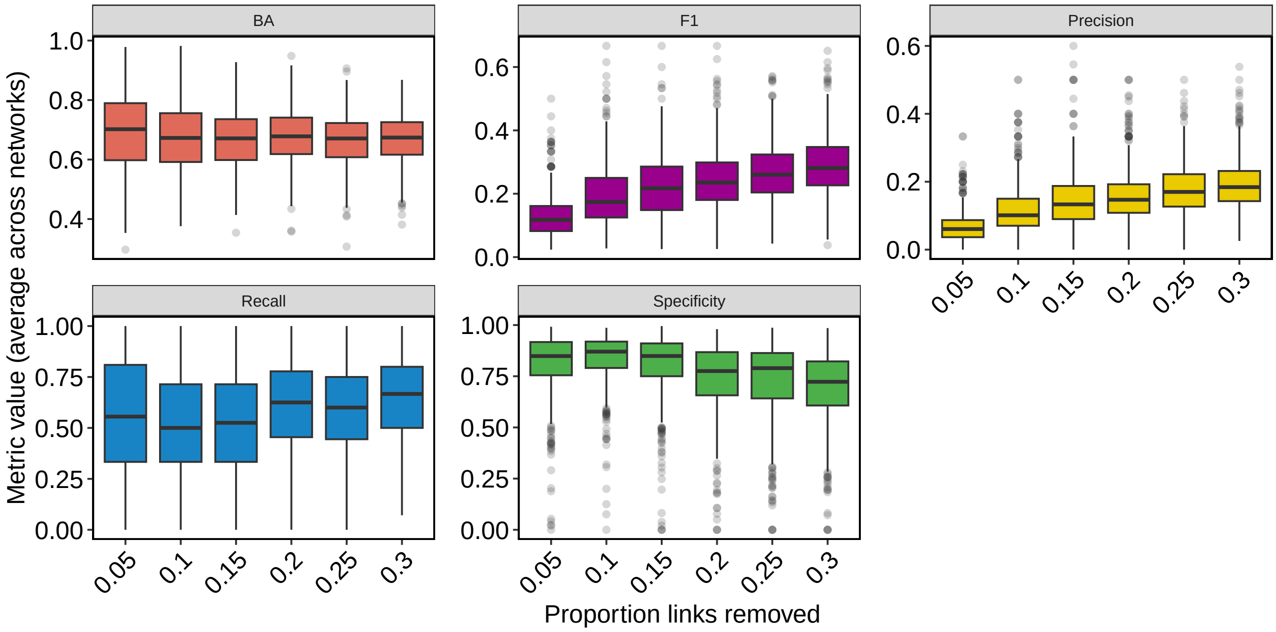

Fig. S20: Link removal sensitivity analysis. Balanced accuracy does not change with withheld proportion (only its variation). F1 generally increases with withheld proportion because increasing the proportion of sub-sampled links reduces the imbalance in the train and test sets, improving precision. Therefore, the choice of 20% balances imbalance with choosing a higher proportion that would artificially improve the results.

S1.4 Cost-sensitive learning

Cost-sensitive learning uses a parameter that the algorithm considers during the learning process. The penalty is the cost, which is minimized during the training process. This approach allows dealing with imbalance while ignoring less samples [13]. The scikit-learn package [14] has a built-in support for cost-sensitive learning. This implementation provides custom weights to a model's classes or samples during training. Depending on the learning algorithm, the loss function is modified to penalize mistakes according to the provided weights, such that higher weights lead to higher penalizing. Because in our study, the minority class is the existing links, these are assigned higher importance by using higher weights. Specifically, each class automatically gets a weight of 1 but can be set with a higher weight. For instance, a class with a weight 2 will have double importance. We aimed for 'balanced' weights, which are proportional to the classes' frequency. For instance, if the training set contains 20 more times non-existing than existing links, then existing links will get 20 times more importance.

Appendix References

1. Michalska-Smith, M. J. & Allesina, S. Telling ecological networks apart by their structure: A computational challenge. *PLoS Comput. Biol.* **15,** e1007076. doi:10.1371/journal.pcbi.1007076 (2019).

2. Brimacombe, C., Bodner, K., Michalska-Smith, M., Poisot, T. & Fortin, M.-J. Shortcomings of reusing species interaction networks created by different sets of researchers. *PLoS Biol.* **21,** e3002068. doi:10.1371/journal.pbio.3002068 (2023).

3. Terry, J. C. D. & Lewis, O. T. Finding missing links in interaction networks. *Ecology* **101,** e03047. doi:10.1002/ecy.3047 (2020).

4. Raschka, S. Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning. *arXiv [cs.LG]* (2018).

5. Kotsiantis, S. B., Zaharakis, I. D. & Pintelas, P. E. Machine learning: a review of classification and combining techniques. *Artificial Intelligence Review* **26,** 159–190. doi:10.1007/s10462-007-9052-3 (2006).

6. Sagi, O. & Rokach, L. Ensemble learning: A survey. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **8,** e1249. doi:10.1002/widm.1249 (2018).

7. Levy, J. J. & O'Malley, A. J. Don't dismiss logistic regression: the case for sensible extraction of interactions in the era of machine learning. *BMC Med. Res. Methodol.* **20,** 171. doi:10.1186/s12874-020-01046-3 (2020).

8. Breiman, L. Random Forests. *Mach. Learn.* **45,** 5–32. doi:10.1023/A:1010933404324 (2001).

9. Biau, G. & Scornet, E. A random forest guided tour. *Test* **25,** 197–227. doi:10.1007/s11749-016-0481-7 (2016).

10. Chen, T. & Guestrin, C. *XGBoost: A Scalable Tree Boosting System* in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* San Francisco, California, USA (Association for Computing Machinery, New York, NY, USA, 2016), 785–794. doi:10.1145/2939672.2939785.

11. Friedman, J. H. Greedy function approximation: A gradient boosting machine. *Ann. Stat.* **29,** 1189–1232. doi:10.1214/aos/1013203451 (2001).

12. Poisot, T. Guidelines for the prediction of species interactions through binary classification. *Methods Ecol. Evol.* **14,** 1333–1345. doi:10.1111/2041-210x.14071 (2023).

13. López, V., Fernández, A., Moreno-Torres, J. G. & Herrera, F. Analysis of preprocessing vs. cost-sensitive learning for imbalanced classification. Open problems on intrinsic data characteristics. *Expert Syst. Appl.* **39,** 6585–6608. doi:10.1016/j.eswa.2011.12.043 (2012).

14. Pedregosa, F *et al.* Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* (2011).