EPISODIC RELATIONAL CONTEXT LEARNING FOR PROTEIN–PROTEIN INTERACTION PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Protein–protein interaction (PPI) prediction is essential for understanding cellular processes and disease mechanisms. However, existing learning-based methods struggle in the *Neither-Seen* (NS) setting, where neither protein in a test pair appears in the training data. To understand this limitation, we analyze the failure modes of existing baselines under the NS setting. Our analysis shows that, beyond the well-recognized effect of sequence similarity, the absence of PPI network topology for unseen proteins constitutes another key source of performance degradation. In this paper, our core idea is to induce relational context for unseen proteins by linking them to the training PPI network rather than leaving them isolated. To this end, we propose Episodic Relational Context Learning (**ERCL**) to improve PPI prediction generalization, particularly under the NS setting. For each query pair, ERCL connects the retrieved anchors from the PPI network to the query proteins via virtual edges, forming a relational context graph. This graph is fed into a relation-aware graph encoder and classified at the graph level, with the resulting label serving as the PPI prediction. To learn relational context from isolation, we construct each training episode by masking the observed interactions of query proteins, turning seen proteins into pseudo-unseen ones. On the cross-species benchmark with exclusively NS testing pairs, our method achieves state-of-the-art performance on all five testing species, with particularly pronounced gains on species more phylogenetically distant from the human training data. Moreover, our model achieves these gains with 585$\times$ fewer parameters than strong baselines such as PLM-interact, while being up to 29$\times$ faster in both training and inference.

1 INTRODUCTION

Protein–protein interactions (PPIs) are fundamental to cellular processes and disease mechanisms. (Scott et al., 2016; Luck et al., 2020). Experimental mapping remains costly and time-consuming, leaving interaction maps incomplete and many newly identified or understudied proteins without well-characterized interaction partners. (Liu et al., 2025; Luck et al., 2020; Kustatscher et al., 2022). Learning-based approaches offer a scalable complement to experimental assays (Liu et al., 2025; Gao et al., 2026). Learning-based approaches offer a scalable complement to experimental assays, but generalizing to proteins absent from the training data remains a key challenge.

Existing PPI predictors draw on two complementary sources of information. Protein-centered approaches predict pairwise interactions using representations derived from amino-acid sequences, structures, or related molecular features (Hamp & Rost, 2015; Hashemifar et al., 2018; Chen et al., 2019; Sledzieski et al., 2021; Liu et al., 2025). Other approaches incorporate observed PPI networks, exploiting network topology and interaction patterns to provide relational context for prediction (Lv et al., 2021; Xie et al., 2023; Gao et al., 2023; Zhou et al., 2025). For proteins with no observed interaction history, molecular features remain available, but relational context from observed query-side PPI topology is unavailable.

We study this challenge in the stringent *Neither-Seen* (NS) setting, where neither protein in a test pair appears in any training pair. Unlike a random pair-level split, which can retain protein overlap between training and test sets, NS evaluation requires generalization to two unseen endpoints (Szymborski & Emad, 2022; Bernett et al., 2024). In the setting considered here, observed PPI evidence is restricted to the training data, leaving both query proteins without accessible observed PPI neighborhoods

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

at inference.Sequence-based predictors remain applicable , yet prior evaluations report substantial performance degradation when training and test proteins are disjoint (Szymborski & Emad, 2022; Bernett et al., 2024). **To inform our method design, we first examine the factors limiting generalization in the NS setting.**

Reduced sequence similarity between training and test proteins offers one plausible explanation (Hamp & Rost, 2015). However, our comparison across five NS test sets suggests that measured sequence proximity alone does not fully account for variation in PLM-interact's reported AUPR (Liu et al., 2025). A complementary topology-only oracle uses privileged access to each test set's PPI network, with the queried edge removed whenever present. Its strong performance without sequence information indicates predictive signal in network topology that is unavailable under our NS setting.Together, these observations motivate viewing NS PPI prediction not only as a sequence generalization challenge, but also as a relational cold-start problem: unseen proteins lack accessible observed PPI context that could otherwise support prediction.

In this work, we propose Episodic Relational Context Learning (**ERCL**), whose **core idea is to induce relational context for otherwise isolated query proteins**. Specifically, for each query pair, ERCL retrieves relational anchors and uses them to organize observed interactions from the training network into a PPI evidence graph. It then links each query protein to its corresponding anchors via virtual edges, forming a pair-specific relational context graph. A relation-aware graph encoder then supports interaction prediction within the constructed context. To teach the model to exploit this context under relational isolation, we introduce the episodic training component technique. **It trains the model to exploit this constructed context when the query proteins' own interaction history is unavailable.** In each training episode, ERCL masks the query proteins and their observed interactions from the network before retrieval, turning seen proteins into *pseudo-unseen* queries. These episodes reproduce the relational information constraints of NS inference and train the model to use evidence retrieved from the remaining network. We evaluate ERCL on D-SCRIPT's five non-human test sets, all NS relative to the human training data (Sledzieski et al., 2021). ERCL outperforms PLM-interact, our strongest baseline, on all five test sets, improving average AUPR by an absolute 0.031. With precomputed features and assembled context graphs, its downstream predictor uses 0.17% as many parameters as PLM-interact and achieves a $29.3\times$ forward-pass speedup.

2 Related Work

PPI Prediction. Learning-based PPI predictors draw on two complementary sources of evidence: molecular information from proteins and relational information from observed PPI networks (Tang et al., 2023; Zhou et al., 2025). Protein-centered approaches predict PPIs from sequence-derived representations, as in PIPR (Chen et al., 2019), or incorporate structure-derived features, as in TT3D (Sledzieski et al., 2023). Pretrained protein language models (PLMs) are also adapted to jointly encode protein pairs, as in PLM-interact (Liu et al., 2025). Network-based approaches combine molecular features with information from observed PPI networks, as exemplified by HNSPPI (Xie et al., 2023), GNN-PPI (Lv et al., 2021), and HIGH-PPI (Gao et al., 2023), with the latter two focusing on multi-label interaction-type prediction. Under NS, molecular features remain available, but neither query protein has an accessible observed PPI neighborhood. Methods such as RAPPPID, INTREPPPID, and TUnA seek to improve prediction for unseen proteins (Szymborski & Emad, 2022; 2024; Ko et al., 2024), yet 2 generalization remains challenging for sequence-based predictors under protein-disjoint and similarity-controlled evaluations (Bernett et al., 2024; Reim et al., 2025). By contrast, network-based methods that require observed interaction information involving the query proteins are not directly applicable in this NS setting. Other approaches leverage network-derived knowledge: Topsy-Turvy uses topology-derived auxiliary supervision (Singh et al., 2022), while L3-PPI incorporates an L3-based interaction prior through graph prompting (Gao et al., 2026). KSGPPI and LLaPA obtain proxy network representations for query proteins from similar proteins in PPI networks (Hu et al., 2024; Zhou et al., 2025), while PPISearch and Path2PPI retrieve or transfer interaction evidence through homologous proteins (Chen et al., 2009; Philipp et al., 2016). These approaches leave open how to assemble an NS query pair and PPI evidence retrieved from the observed training network into a relational context graph for joint encoding and prediction.

In-Context and Episodic Learning. At a broader methodological level, in-context learning (ICL) conditions predictions on demonstrations provided alongside a query, without test-time parameter

updates (Brown et al., 2020). Liu et al. (2022) construct query-specific prompts by retrieving examples semantically similar to each query. Inspired by this context-conditioning principle, ERCL evaluates each NS pair within a relational context constructed from training network evidence. On the training side, Matching Networks and Prototypical Networks organize learning into support–query episodes that reflect test-time task structure, training models to make predictions conditioned on labeled support sets (Vinyals et al., 2016; Snell et al., 2017). ERCL draws on this training principle to learn to exploit constructed relational context under conditions that reflect the information constraints of NS inference.

3 Problem Setup

Binary PPI prediction. We consider binary PPI prediction over a protein set $\mathcal{P}$, treating interactions as undirected. For each labeled pair $q_{ij} = (p_i, p_j)$, with $p_i, p_j \in \mathcal{P}$, the benchmark label $y_{ij} \in \{0, 1\}$ denotes a recorded positive interaction ($y_{ij} = 1$) or a negative pair constructed for the benchmark ($y_{ij} = 0$). Given a query pair q_{ij} and the evidence $\mathcal{I}_{ij}$ permitted under the setting below, a predictor parameterized by θ outputs an estimated interaction probability $\hat{y}_{ij} = f_\theta(q_{ij}, \mathcal{I}_{ij}) \in [0, 1]$.

The Neither-Seen Setting in PPI Prediction. Relative to the training set $\mathcal{D}_{\text{train}}$, a protein is seen if it appears in at least one positive or negative training pair, and unseen otherwise. Test pairs are categorized as Both-Seen (BS), Either-Seen (ES), or Neither-Seen (NS) according to whether they contain two, one, or zero seen endpoints, respectively. In the NS setting considered here, the predictor may use available molecular features of the query proteins. Observed PPI relations available as evidence are restricted to the recorded positive interactions in $\mathcal{D}_{\text{train}}$. Test-pair labels are used only for evaluation.

The recorded positive interactions in $\mathcal{D}_{\text{train}}$ define an undirected training PPI network $\mathcal{G}_{\text{train}} = (\mathcal{V}_{\text{train}}, \mathcal{E}_{\text{train}})$. Its edge set is

$$\mathcal{E}_{\text{train}} = \{\{p_i, p_j\} \mid (q_{ij}, y_{ij}) \in \mathcal{D}_{\text{train}},\ y_{ij} = 1\},$$

and $\mathcal{V}_{\text{train}}$ consists of the endpoints of these edges. Negative training pairs are used for supervision but do not form edges in $\mathcal{G}_{\text{train}}$.

Interaction-type prediction assigns types to recorded PPIs; some network-based NS evaluations use a semi-supervised setting in which query-side PPI topology is available but test interaction-type labels are withheld (Lv et al., 2021). Methods that require this observed query-side topology are therefore not directly applicable to our NS setting.

4 What Limits PPI Prediction for Unseen Proteins?

To better understand the challenges of NS PPI prediction, we examine sequence-based generalization and the potential predictive value of observed query-side PPI network context that is unavailable under this setting. We conduct three complementary diagnostic analyses, with their setups and principal findings summarized in Figure 1. Additional details of the diagnostic analyses are provided in Appendix A.1.

4.1 Sequence-Only Performance by Endpoint Visibility

To compare performance across endpoint-visibility conditions without introducing cross-species variation, we construct training and test splits entirely within the D-SCRIPT benchmark's original human training set (Sledzieski et al., 2021). We use a random pair-level split as the baseline, without imposing constraints on the endpoint visibility of test pairs. By contrast, we partition proteins to construct Either-Seen (ES) and Neither-Seen (NS) splits, whose test pairs contain exactly one and zero seen endpoints, respectively, relative to the corresponding training subset. Across the three splits, we use the same lightweight sequence-only PPI architecture and training configuration, with fixed training and test set sizes and positive-to-negative ratios.

The predictor's mean AUPR decreases from 0.823 on the random split to 0.621 on the ES split and 0.386 on the NS split (Figure 1(a)). These results show that, for the evaluated sequence-only predictor, the difficulty of NS generalization is already evident within a single species, without a species shift.

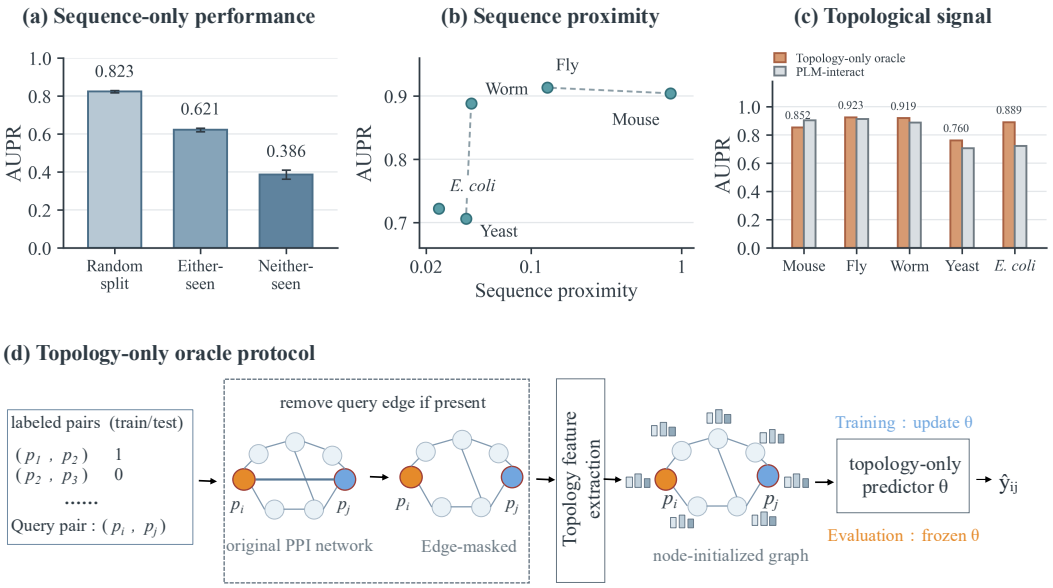

Figure 1: Diagnostic analyses of NS PPI prediction. (a) Sequence-only performance by endpoint visibility. (b) Median sequence proximity to human training proteins (log-scaled x-axis) versus reported PLM-interact AUPR (Liu et al., 2025); dashed lines mark the comparisons discussed in the text. (c) Topology-only oracle AUPR, with reported PLM-interact results for reference. (d) Oracle protocol: training on human data and evaluation with fixed parameters on privileged graphs of recorded test-set PPIs. Bars in (a) and oracle bars in (c) show means over three seeds; error bars in (a) show sample standard deviations.

4.2 Sequence Proximity and NS Performance

Across the benchmark's five non-human test sets, all pairs are NS relative to the human training set, yet PLM-interact's reported AUPR varies substantially (Liu et al., 2025). Motivated by the association between train–test sequence identity and performance reported in the same study, we compare these AUPRs with a test-set-level measure of sequence proximity to human training proteins.

For each test protein with a detected match, we select the highest-bit-score MMseqs2 hit among the human training proteins (Steinegger & Söding, 2017). We divide this bit score by the protein's self-alignment bit score and multiply the result by query coverage. Records failing alignment or self-alignment validity checks are excluded; among the remaining proteins, those without a detected human match are assigned a score of zero. We use the median of the resulting protein-level scores within each test set as its sequence-proximity measure.

As shown in Figure 1(b), sequence proximity broadly aligns with the reported AUPR across the five test sets, but notable discrepancies remain. Worm and yeast have similar sequence-proximity values, yet AUPR is substantially higher for worm than for yeast (0.888 vs. 0.706). Conversely, sequence proximity is substantially higher for mouse than for fly, yet their AUPRs are similar (0.904 vs. 0.913). Together, these comparisons suggest that sequence proximity alone does not fully account for the variation in PLM-interact's reported AUPR across the five NS test sets.

4.3 PPI Topology Carries Predictive Signal

Connectivity patterns in observed PPI networks can inform the prediction of missing interactions (Kovács et al., 2019; Wang et al., 2023). As a complementary diagnostic to the sequence-proximity analysis, we examine whether the topology of test-set PPI networks carries predictive signal for NS pairs using a topology-only oracle.

The oracle uses no sequence-derived information. It is trained only on labeled pairs from the human training set, using the PPI network formed from that set's recorded positive interactions. On each

test set, we keep the learned parameters fixed and provide privileged access to the PPI network formed from that set's recorded positive interactions. For each queried pair, during both training and evaluation, we remove the queried edge whenever present before extracting node- and pair-level topology features and performing message passing on the resulting network. Figure 1(d) illustrates the training and evaluation protocol.

As shown in Figure 1(c), the oracle's mean AUPR ranges from 0.760 to 0.923 across the five test sets, with PLM-interact's reported AUPRs included for reference. These results indicate that the topology of these test-set PPI networks carries substantial predictive signal under the oracle setting, even without sequence-derived information. This topology, however, is unavailable under the NS information constraints defined in Section 3. Together, these diagnostics support viewing NS PPI prediction not only as a challenge of sequence-based generalization but also as a relational cold-start problem, in which query proteins lack accessible observed PPI context.

5 METHOD

5.1 OVERVIEW

Motivated by this relational cold-start perspective, we propose Episodic Relational Context Learning (ERCL) for NS PPI prediction. The core idea is to place each NS query pair in a query-specific relational context constructed from the observed interactions of the training PPI network, so that the pair is evaluated together with relational evidence rather than as two isolated proteins.

Context construction and prediction proceed in four stages. First, ERCL retrieves relational anchors for each query protein from the training PPI network. Second, it uses these anchors to retrieve and organize observed PPI relations from the training network into a PPI evidence graph. Third, it connects each query protein to its anchors via virtual edges, placing the query pair within this relational structure to form a query-specific relational context graph. Finally, it encodes the full graph with a relation-aware graph encoder and performs graph-level classification for interaction prediction.

To simulate the absence of observed query-side PPI relations at NS inference, ERCL uses episodic masking to exclude the training query proteins and all their incident observed PPI edges from the network used for evidence retrieval, creating pseudo-unseen queries. The same context-construction and prediction pipeline is applied to these queries, training the model to use relational context under the information constraints of NS inference.

5.2 PAIR-SPECIFIC RELATIONAL CONTEXT CONSTRUCTION

For each NS query pair $q_{ij} = (p_i, p_j)$, ERCL constructs a query-specific relational context graph using the training PPI network $\mathcal{G}_{\text{train}} = (\mathcal{V}_{\text{train}}, \mathcal{E}_{\text{train}})$ as a source of relational evidence. We use sequence-derived protein representations and pairwise sequence-matching scores to guide the construction of this context.

5.2.1 RELATIONAL ANCHOR RETRIEVAL

To retrieve relational evidence for each query endpoint, we first identify a set of relational anchors. Let $b(u, v)$ denote the sequence-matching score obtained for a protein pair (u, v) with a detected sequence match. In our implementation, $b(u, v)$ is the alignment bit score returned by MMseqs2 (Steinegger & Söding, 2017).

For each query endpoint p_u, $u \in \{i, j\}$, we search the training PPI network for candidate proteins with sequence matches to p_u. Let $\mathcal{C}_u$ denote the resulting candidate set. We retain up to K candidates with the highest matching scores to form the relational anchor set:

$$\mathcal{A}_u = \mathrm{TopK}_{v \in \mathcal{C}_u}\, b(p_u, v), \qquad u \in \{i, j\},$$

where TopK returns the set of selected protein nodes.

The resulting anchor sets $\mathcal{A}_i$ and $\mathcal{A}_j$ serve as entry points for retrieving PPI evidence relevant to the query pair q_{ij}.

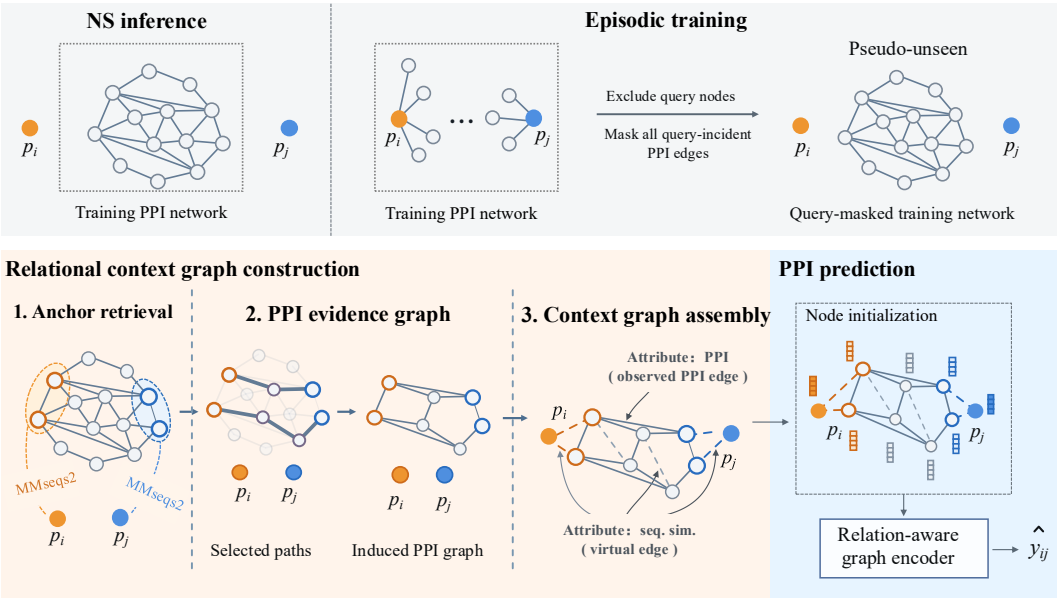

Figure 2: Overview of ERCL. Episodic masking excludes training query proteins and their incident observed PPI edges from the retrieval network, creating pseudo-unseen queries to simulate the relational constraints of NS inference. A shared training and inference pipeline retrieves anchors and PPI evidence, connects queries to anchors via virtual edges, and predicts interactions within the resulting pair-specific context graph using a relation-aware encoder.

5.2.2 PPI Evidence Graph

Given the anchor sets $\mathcal{A}_i$ and $\mathcal{A}_j$, we search the training PPI network $\mathcal{G}_{\text{train}}$ for simple PPI paths of at most L_{path} hops connecting distinct anchors $a \in \mathcal{A}_i$ and $b \in \mathcal{A}_j$. The hop limit may still leave multiple candidate paths; we therefore rank them according to their relevance to the query pair q_{ij}.

Specifically, we score each candidate path by aggregating the relevance of its internal nodes to the two query proteins using sequence-derived representations. Let $\mathbf{z}_x$ denote the sequence-derived representation of protein x. For each internal node v, we define its relevance to the query pair as

$$r_{ij}(v) = \max\left\{\text{sim}(\mathbf{z}_v, \mathbf{z}_{p_i}), \text{sim}(\mathbf{z}_v, \mathbf{z}_{p_j})\right\},$$

where $\text{sim}(\cdot,\cdot)$ denotes cosine similarity.

Let $\text{Int}(\pi)$ denote the internal nodes of a candidate path π, excluding its two anchor endpoints. We define $\text{score}_{ij}(\pi)$ as the mean of $r_{ij}(v)$ over $v \in \text{Int}(\pi)$, assigning a score of 1 when $\text{Int}(\pi) = \varnothing$.

For each anchor pair $(a, b) \in \mathcal{A}_i \times \mathcal{A}_j$ with $a \neq b$, we retain up to B candidate paths with the highest values of $\text{score}_{ij}(\pi)$. The paths retained over all anchor pairs form the collection Π_{ij}^{ret}.

The anchors and internal nodes of the retained paths define the PPI evidence nodes. We preserve all observed training PPI edges among these nodes, forming the evidence graph with

$$\mathcal{R}_{ij} = \mathcal{A}_i \cup \mathcal{A}_j \cup \bigcup_{\pi \in \Pi_{ij}^{\text{ret}}} \text{Int}(\pi),$$

$$\mathcal{E}_{ij}^{\text{ppi}} = \{\{u, v\} \in \mathcal{E}_{\text{train}} \mid u, v \in \mathcal{R}_{ij}\}.$$

Together, $\mathcal{R}_{ij}$ and $\mathcal{E}_{ij}^{\text{ppi}}$ form the query-specific PPI evidence graph retrieved for q_{ij}.

5.2.3 Context Graph Assembly

Given the retrieved PPI evidence, we assemble a query-specific relational context graph for q_{ij} by integrating the query proteins with the retrieved evidence.

The context node set is $\mathcal{V}_{ij}^{\mathrm{ctx}} = \{p_i, p_j\} \cup \mathcal{R}_{ij}$, combining the query proteins with the retrieved evidence nodes. We first add query–anchor virtual edges that connect each query protein p_u to its anchors $a \in \mathcal{A}_u$, thereby linking the query pair to the retrieved PPI evidence.

To incorporate additional sequence matches within the context, we add sequence-derived virtual edges between other pairs of context nodes whose matching scores exceed a fixed threshold.

Together, the query–anchor and supplementary links define $\mathcal{E}_{ij}^{\mathrm{virt}}$, whose edges are weighted by the corresponding sequence-matching scores $b(\cdot, \cdot)$.

Combining the retrieved PPI edges, each assigned unit weight, with $\mathcal{E}_{ij}^{\mathrm{virt}}$ gives the final relational context graph:

$$\mathcal{G}_{ij}^{\mathrm{ctx}} = \left(\mathcal{V}_{ij}^{\mathrm{ctx}}, \{\mathcal{E}_{ij}^{r}\}_{r \in \{\mathrm{ppi,virt}\}}\right).$$

5.3 Relation-Aware Graph Encoding and Prediction

Given the assembled relational context graph $\mathcal{G}_{ij}^{\mathrm{ctx}}$, we encode its full multi-relational structure with a relation-aware graph encoder. Each node is initialized by projecting its sequence-derived representation $\mathbf{z}_v$ into the encoder's hidden space. During message passing, the encoder propagates information over the full graph while distinguishing observed PPI edges from virtual edges and incorporating their corresponding weights into neighborhood aggregation. After L message-passing layers, the resulting node representations $\mathbf{h}_v^{(L)}$ integrate sequence-derived information with the pair-specific relational context.

A graph-level readout aggregates the encoded node representations into a context representation $\mathbf{g}_{ij}$. We evaluate the query pair within the constructed relational context using its encoded endpoint representations together with $\mathbf{g}_{ij}$:

$$\hat{y}_{ij} = f_\psi\left(\mathbf{h}_{p_i}^{(L)}, \mathbf{h}_{p_j}^{(L)}, \mathbf{g}_{ij}\right),$$

where f_ψ denotes the prediction head, which outputs an interaction probability and is invariant to exchanging the two endpoint inputs. Additional implementation details are provided in Appendix A.2.

5.4 Episodic Training and Optimization

To simulate the relational information constraints of NS inference, ERCL constructs a masked episode for each labeled training pair. For a training query pair $q_{ij} = (p_i, p_j)$, we exclude both query proteins and their incident PPI edges from the network used for evidence retrieval:

$$\mathcal{G}_{\mathrm{train}}^{(-ij)} = \mathcal{G}_{\mathrm{train}}\left[\mathcal{V}_{\mathrm{train}} \setminus \{p_i, p_j\}\right],$$

where the brackets denote the subgraph induced by the specified node set. The query proteins are thus treated as pseudo-unseen with respect to observed PPI topology, while their sequence information remains available. We then apply the context construction and graph prediction procedures described in Sections 5.2 and 5.3, using $\mathcal{G}_{\mathrm{train}}^{(-ij)}$ as the source of relational evidence.

For each masked training episode, we apply binary focal loss (Lin et al., 2017) to the predicted interaction probability $\hat{y}_{ij}$ and the corresponding pair label y_{ij}. We minimize the mean loss over training pairs to jointly optimize the trainable components of ERCL, while keeping the sequence-derived representations and matching scores fixed.

6 Experiments

6.1 Experimental Setup

We evaluate ERCL for NS PPI prediction on the D-SCRIPT human-to-five-species benchmark (Sledzieski et al., 2021). We train on the released human training split, use the released human test split for validation, and evaluate on the mouse, fly, worm, yeast, and *E. coli* test sets. Relative to the human training split, the validation pairs are Both-Seen (BS), whereas all pairs in the five test sets are NS. All splits use a 1:10 positive-to-negative sampling ratio.

Table 1: NS prediction results (AUPR). Bold marks the best score in each column; PLM-interact scores are underlined, and parentheses show absolute gains over them.

Method	Mouse	Fly	Worm	Yeast	*E. coli*	AVG.
Previously reported benchmark results						
PIPR	0.526	0.278	0.346	0.230	0.308	0.338
D-SCRIPT	0.580	0.552	0.548	0.405	0.571	0.531
Topsy-Turvy	0.735	0.713	0.700	0.534	0.556	0.648
TT3D	0.779	0.757	0.738	0.553	0.605	0.686
TUnA	0.885	0.855	0.834	0.641	0.677	0.778
X-PAIR	0.892	0.877	0.854	0.682	0.715	0.804
PLM-interact	<u>0.904</u>	<u>0.913</u>	<u>0.888</u>	<u>0.706</u>	<u>0.722</u>	<u>0.827</u>
Adapted network-informed baselines						
GNN-PPI (adaptation)	0.168	0.154	0.260	0.152	0.090	0.165
KSGPPI-style	0.902	0.901	0.843	0.619	0.468	0.747
L3-PPI-style	0.772	0.778	0.737	0.543	0.622	0.690
ERCL	**0.930** (+0.026)	**0.941** (+0.028)	**0.914** (+0.026)	**0.740** (+0.034)	**0.763** (+0.041)	**0.858** (+0.031)

We compare ERCL with seven representative PPI predictors using their reported benchmark results: PIPR, D-SCRIPT, Topsy-Turvy, TT3D, TUnA, X-PAIR, and PLM-interact. Reported species-level AUPRs are taken from Liu et al. (2025), except those for X-PAIR, which are from Rescalli & Carbone (2026). We use PLM-interact as our primary reference, as it has the highest reported average AUPR among these methods. To compare different ways of using network knowledge, we evaluate three adapted baselines: GNN-PPI, KSGPPI-style, and L3-PPI-style. All three baselines are evaluated under our NS setting, with adaptation details provided in Appendix A.3.

For ERCL, we initialize node features with representations from a frozen ESM-2-650M model (Lin et al., 2023). We use the same graph construction configuration across all five test sets. We use AUPR, computed as average precision, as the primary evaluation metric. We train ERCL, its variants, and the adapted baselines with three random seeds and select the checkpoint with the highest human validation AUPR in each run. For each run, average AUPR is computed as the unweighted mean of the AUPRs across the five NS test sets. For models trained in this study, we report means across runs in the main-text results tables and provide sample standard deviations in Table A.7.

6.2 Main Results

As shown in Table 1, ERCL outperforms the previously reported baselines on all five NS test sets. Compared with PLM-interact, ERCL achieves an absolute gain of 0.031 in average AUPR. The two largest species-level gains occur on yeast and *E. coli*, the two test sets with the lowest PLM-interact AUPR. These gains support using the training network as a source of context for NS prediction.

Among the baselines, the GNN-PPI adaptation performs poorly on NS pairs. The KSGPPI-style and L3-PPI-style adaptations obtain stronger results, but both remain below ERCL across all five test sets. These comparisons highlight the importance of how network knowledge is organized and used for NS pairs. ERCL places each unseen pair within a relational context constructed from PPI evidence in the training network and learns to predict the interaction within that context.

6.3 Relational Context Analysis

We compare ERCL with an endpoint-only sequence predictor that uses the same frozen protein representations but no retrieved context (Table 2). ERCL achieves higher AUPR on all five NS test sets, with an absolute gain of 0.036 in average AUPR. These gains suggest that evaluating NS pairs within the constructed relational context adds predictive value beyond endpoint representations alone.

We further evaluate weighted RGCN and RGAT configurations using the same context graphs and node features as ERCL. Both configurations outperform Endpoint-only on all five NS test sets, achieving average AUPRs of 0.829 and 0.843, respectively, compared with 0.827 for PLM-interact.

These results suggest that the constructed context can provide useful information for NS prediction under a simpler graph-based configuration.

Table 2: Relational context analysis: AUPR on the human BS validation set and five NS test sets.

	BS validation	NS test					
Configuration	Human	Mouse	Fly	Worm	Yeast	*E. coli*	AVG.
PLM-interact	–	0.904	0.913	0.888	0.706	0.722	0.827
Endpoint-only	0.823	0.901	0.895	0.871	0.701	0.742	0.822
ERCL (weighted RGCN)	0.930	0.908	0.900	0.878	0.713	0.747	0.829
ERCL (RGAT)	0.933	0.921	0.916	0.890	0.732	0.758	0.843
ERCL	0.930	0.930	0.941	0.914	0.740	0.763	0.858

6.4 Efficiency Analysis

We compare the prediction-stage cost of ERCL and PLM-interact across the five NS test sets (Figure 3). For ERCL, the reported costs cover the downstream predictor using precomputed protein features and assembled context graphs; measurement details are provided in Appendix A.5. ERCL achieves higher AUPR on all five test sets while using only 0.17% as many predictor parameters and achieving a $29.3\times$ speedup in the measured forward pass. Together with the relational context analysis, these findings suggest that the constructed relational context can support effective NS prediction with a compact downstream model.

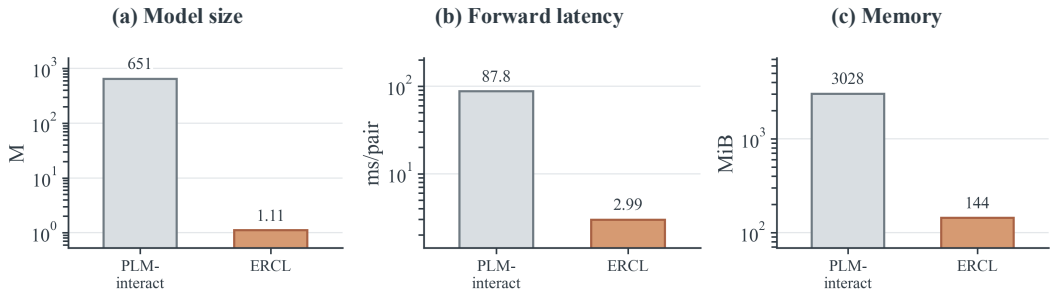

Figure 3: Prediction-stage model size and execution cost across the five NS test sets.

7 Conclusion

The broader implication of this work is that relational cold start in NS PPI prediction concerns not only what evidence is available, but also how that evidence is organized and how a model learns to use it. ERCL connects these questions through ICL-inspired construction of query-specific relational context and episodic masking that creates pseudo-unseen training queries. Context construction organizes available training-network evidence around each query pair, while masking trains the model to use this context without access to the query proteins' own observed interactions. Our findings support this joint view of context construction and learning under relational information constraints as a complementary direction to improving transferable protein representations.

AI use statement

In preparing this work, we used generative AI tools to assist with English translation, code review, and reviewing experimental procedures. We also used these tools for manuscript editing, LaTeX formatting, and selecting color schemes for figures. The authors take responsibility for the final content of this work, including its text, claims, code, and figures.

REFERENCES

Judith Bernett, David B. Blumenthal, and Markus List. Cracking the black box of deep sequence-based protein–protein interaction prediction. *Briefings in Bioinformatics*, 25(2):bbae076, 2024. doi: 10.1093/bib/bbae076.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901, 2020.

Chun-Chen Chen, Chun-Yu Lin, Yu-Shu Lo, and Jinn-Moon Yang. PPISearch: A web server for searching homologous protein–protein interactions across multiple species. *Nucleic Acids Research*, 37(suppl_2):W369–W375, 2009. doi: 10.1093/nar/gkp309.

Muhao Chen, Chelsea J.-T. Ju, Guangyu Zhou, Xuelu Chen, Tianran Zhang, Kai-Wei Chang, Carlo Zaniolo, and Wei Wang. Multifaceted protein–protein interaction prediction based on siamese residual RCNN. *Bioinformatics*, 35(14):i305–i314, 2019. doi: 10.1093/bioinformatics/btz328.

Ziqi Gao, Chenran Jiang, Jiawen Zhang, Xiaosen Jiang, Lanqing Li, Peilin Zhao, Huanming Yang, Yong Huang, and Jia Li. Hierarchical graph learning for protein–protein interaction. *Nature Communications*, 14(1):1093, 2023. doi: 10.1038/s41467-023-36736-1.

Ziqi Gao, Chenyi Zi, Zijing Liu, Ziqiao Meng, Yu Li, and Jia Li. Learning the interaction prior for protein–protein interaction prediction: A model-agnostic approach. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL `https://arxiv.org/abs/2605.09964`.

Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 855–864, 2016. doi: 10.1145/2939672.2939754.

Tobias Hamp and Burkhard Rost. More challenges for machine-learning protein interactions. *Bioinformatics*, 31(10):1521–1525, 2015. doi: 10.1093/bioinformatics/btu857.

Somaye Hashemifar, Behnam Neyshabur, Aly A. Khan, and Jinbo Xu. Predicting protein–protein interactions through sequence-based deep learning. *Bioinformatics*, 34(17):i802–i810, 2018. doi: 10.1093/bioinformatics/bty573.

Jun Hu, Zhe Li, Bing Rao, Maha A. Thafar, and Muhammad Arif. Improving protein–protein interaction prediction using protein language model and protein network features. *Analytical Biochemistry*, 693:115550, 2024. doi: 10.1016/j.ab.2024.115550.

Young Su Ko, Jonathan Parkinson, Cong Liu, and Wei Wang. TUnA: An uncertainty-aware transformer model for sequence-based protein–protein interaction prediction. *Briefings in Bioinformatics*, 25(5):bbae359, 2024. doi: 10.1093/bib/bbae359.

István A. Kovács, Katja Luck, Kerstin Spirohn, Yang Wang, Carl Pollis, Sadie Schlabach, Wenting Bian, Dae-Kyum Kim, Nishka Kishore, Tong Hao, Michael A. Calderwood, Marc Vidal, and Albert-László Barabási. Network-based prediction of protein interactions. *Nature Communications*, 10(1):1240, 2019. doi: 10.1038/s41467-019-09177-y.

Georg Kustatscher, Tom Collins, Anne-Claude Gingras, Tiannan Guo, Henning Hermjakob, Trey Ideker, Kathryn S. Lilley, Emma Lundberg, Edward M. Marcotte, Markus Ralser, and Juri Rappsilber. Understudied proteins: Opportunities and challenges for functional proteomics. *Nature Methods*, 19(7):774–779, 2022. doi: 10.1038/s41592-022-01454-x.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2980–2988, 2017. doi: 10.1109/ICCV.2017.324.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Salvatore Candido, and Alexander Rives. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023. doi: 10.1126/science.ade2574. URL https://doi.org/10.1126/science.ade2574.

Dan Liu, Francesca Young, Kieran D. Lamb, Adalberto Claudio Quiros, Alexandrina Pancheva, Crispin J. Miller, Craig Macdonald, David L. Robertson, and Ke Yuan. PLM-interact: Extending protein language models to predict protein–protein interactions. *Nature Communications*, 16(1): 9012, 2025. doi: 10.1038/s41467-025-64512-w.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pp. 100–114, 2022. doi: 10.18653/v1/2022.deelio-1.10.

Katja Luck, Dae-Kyum Kim, Luke Lambourne, Kerstin Spirohn, et al. A reference map of the human binary protein interactome. *Nature*, 580(7803):402–408, 2020. doi: 10.1038/s41586-020-2188-x.

Guofeng Lv, Zhiqiang Hu, Yanguang Bi, and Shaoting Zhang. Learning unknown from correlations: Graph neural network for inter-novel-protein interaction prediction. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, pp. 3677–3683. International Joint Conferences on Artificial Intelligence Organization, 2021. doi: 10.24963/ijcai.2021/506.

Oliver Philipp, Heinz D. Osiewacz, and Ina Koch. Path2PPI: An R package to predict protein–protein interaction networks for a set of proteins. *Bioinformatics*, 32(9):1427–1429, 2016. doi: 10.1093/bioinformatics/btv765.

Timo Reim, Anne Hartebrodt, David B. Blumenthal, Judith Bernett, and Markus List. Deep learning models for unbiased sequence-based PPI prediction plateau at an accuracy of 0.65. *Bioinformatics*, 41(Supplement 1):i590–i598, 2025. doi: 10.1093/bioinformatics/btaf192.

Sara Rescalli and Alessandra Carbone. X-PAIR: An ultrafast multitask framework for proteome-scale reconstruction of PPI networks and partner-specific interfaces from sequence. *bioRxiv*, 2026. doi: 10.64898/2026.07.20.739596. Preprint.

Duncan E. Scott, Andrew R. Bayly, Chris Abell, and John Skidmore. Small molecules, big targets: Drug discovery faces the protein–protein interaction challenge. *Nature Reviews Drug Discovery*, 15(8):533–550, 2016. doi: 10.1038/nrd.2016.29.

Rohit Singh, Kapil Devkota, Samuel Sledzieski, Bonnie Berger, and Lenore Cowen. Topsy-Turvy: Integrating a global view into sequence-based PPI prediction. *Bioinformatics*, 38(Supplement_1): i264–i272, 2022. doi: 10.1093/bioinformatics/btac258.

Samuel Sledzieski, Rohit Singh, Lenore Cowen, and Bonnie Berger. D-SCRIPT translates genome to phenome with sequence-based, structure-aware, genome-scale predictions of protein–protein interactions. *Cell Systems*, 12(10):969–982.e6, 2021. doi: 10.1016/j.cels.2021.08.010.

Samuel Sledzieski, Kapil Devkota, Rohit Singh, Lenore Cowen, and Bonnie Berger. TT3D: Leveraging precomputed protein 3D sequence models to predict protein–protein interactions. *Bioinformatics*, 39(11):btad663, 2023. doi: 10.1093/bioinformatics/btad663.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 30, pp. 4077–4087, 2017.

Martin Steinegger and Johannes Söding. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature Biotechnology*, 35(11):1026–1028, 2017. doi: 10.1038/nbt.3988.

Joseph Szymborski and Amin Emad. RAPPPID: Towards generalizable protein interaction prediction with AWD-LSTM twin networks. *Bioinformatics*, 38(16):3958–3967, 2022. doi: 10.1093/bioinformatics/btac429.

Joseph Szymborski and Amin Emad. INTREPPPID—an orthologue-informed quintuplet network for cross-species prediction of protein–protein interaction. *Briefings in Bioinformatics*, 25(5):bbae405, 2024. doi: 10.1093/bib/bbae405.

Tao Tang, Xiaocai Zhang, Yuansheng Liu, Hui Peng, Binshuang Zheng, Yanlin Yin, and Xiangxiang Zeng. Machine learning on protein–protein interaction prediction: models, challenges and trends. *Briefings in Bioinformatics*, 24(2):bbad076, 2023. doi: 10.1093/bib/bbad076. URL https://doi.org/10.1093/bib/bbad076.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, volume 29, pp. 3630–3638, 2016.

Xu-Wen Wang, Lorenzo Madeddu, Kerstin Spirohn, Leonardo Martini, Adriano Fazzone, Luca Becchetti, Thomas P. Wytock, István A. Kovács, Olivér M. Balogh, Bettina Benczik, Mátyás Pétervári, Bence Ágg, Péter Ferdinandy, Loan Vulliard, Jörg Menche, Stefania Colonnese, Manuela Petti, Gaetano Scarano, Francesca Cuomo, Tong Hao, Florent Laval, Luc Willems, Jean-Claude Twizere, Marc Vidal, Michael A. Calderwood, Enrico Petrillo, Albert-László Barabási, Edwin K. Silverman, Joseph Loscalzo, Paola Velardi, and Yang-Yu Liu. Assessment of community efforts to advance network-based prediction of protein–protein interactions. *Nature Communications*, 14(1): 1582, 2023. doi: 10.1038/s41467-023-37079-7.

Shijie Xie, Xiaojun Xie, Xin Zhao, Fei Liu, Yiming Wang, Jihui Ping, and Zhiwei Ji. HNSPPI: a hybrid computational model combing network and sequence information for predicting protein–protein interaction. *Briefings in Bioinformatics*, 24(5):bbad261, 2023. doi: 10.1093/bib/bbad261.

Peng Zhou, Pengsen Ma, Jianmin Wang, Xibao Cai, Haitao Huang, Wei Liu, Longyue Wang, Lai Hou Tim, and Xiangxiang Zeng. Large language and protein assistant for protein–protein interactions prediction. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11312–11327, Vienna, Austria, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.554. URL https://aclanthology.org/2025.acl-long.554/.

A Appendix

A.1 Diagnostic Analysis Details

Endpoint-visibility splits. For the sequence-only diagnostic in Section 4.1, we construct all splits within the released human training split of the D-SCRIPT benchmark (Sledzieski et al., 2021). This design avoids introducing cross-species variation when comparing endpoint-visibility conditions. The random split is constructed at the pair level from the full human training-pair pool, without imposing endpoint-visibility constraints.

For the Single-Seen and Neither-Seen (NS) splits, we stratify proteins into those occurring in at least one positive pair and those occurring only in negative pairs. Within each group, proteins are sorted by positive-pair count or total-pair count, respectively, and divided into five approximately equal-sized bins. Within each bin, proteins are randomly assigned to seen and unseen groups at an 80:20 ratio using seed 42. We then classify candidate pairs into seen–seen, seen–unseen, and unseen–unseen pools and sample test pairs from the latter two pools, respectively. In both cold-start splits, training and validation pairs are sampled from the seen–seen pool. Across all three endpoint-visibility conditions, we keep the split sizes and positive-to-negative ratio fixed as shown in Table A.1. We use a lightweight sequence-only predictor comprising an amino-acid embedding layer, a CNN–BiGRU encoder, masked mean pooling, and an MLP pair classifier. The architecture and training configuration are identical across the three conditions, and checkpoints are selected by validation AUPR.

Table A.1: Split sizes used for each endpoint-visibility condition in the sequence-only diagnostic.

Split	Positive pairs	Negative pairs	Total pairs
Train	20,000	180,000	200,000
Validation	1,000	9,000	10,000
Test	1,691	15,219	16,910

Sequence proximity. For the sequence-proximity diagnostic in Section 4.2, we define a protein-level proximity score for proteins in each non-human test set. For a test protein p with a detected human-training match, we use MMseqs2 (Steinegger & Söding, 2017) to identify its highest-bit-score match $h^\star(p)$ among proteins in the human training split. We normalize this bit score by the self-alignment bit score of p and weight it by the alignment-length-to-query-length ratio:

$$s(p) = \frac{\mathrm{bits}(p, h^\star(p))}{\mathrm{bits}(p, p)} \cdot \mathrm{qcov}(p, h^\star(p)).$$

Here, $\mathrm{qcov}(p, h^\star(p))$ is the MMseqs2 alignment length divided by the query protein length. A detected human-training match is considered valid only if its target is in the human reference and its bit score and query and target lengths pass the record checks. For normalization, we require a full-length, 100%-identity self-alignment from the default MMseqs2 search, with a valid bit score and sequence lengths consistent with the input sequence. Proteins failing these checks are excluded. Among the remaining proteins, those without a detected human-training match are assigned a score of zero. For each test set, we use the median of the resulting protein-level scores as its test-set-level sequence-proximity measure. Table A.2 reports the values used in Figure 1(b), together with the reported PLM-interact AUPR (Liu et al., 2025).

Table A.2: Test-set median sequence proximity and reported PLM-interact AUPR used in Figure 1(b).

Test set	Sequence proximity	PLM-interact AUPR
Mouse	0.842	0.904
Fly	0.128	0.913
Worm	0.040	0.888
Yeast	0.037	0.706
E. coli	0.024	0.722

Topology-only oracle. For the diagnostic in Section 4.3, we construct separate PPI base graphs from the recorded positive interactions in the human training split, the human validation split, and each non-human test set. For a positive query pair, its edge is removed from the corresponding base graph before computing node- and pair-level topology features and performing message passing. For a negative query pair, the edge is absent, so the base graph is used unchanged.

The predictor uses 18-dimensional node-level topology features and 10-dimensional pair-level topology features. A two-layer GIN with 128-dimensional hidden representations encodes the nodes; the two endpoint representations are combined by elementwise multiplication and absolute difference, then fused with the pair-level features for prediction. No sequence-derived features or PLM representations are used. We train on human training pairs, select the checkpoint by human validation AUPR under the same query-edge-masking protocol, and evaluate on each non-human test set with the model parameters fixed. Because evaluation uses a graph constructed from the target test set's positive interactions, this is an oracle diagnostic of topological signal, not a predictor deployable under the strictly inductive NS setting.

A.2 Implementation Details

Protein representations and sequence matching. We use frozen ESM-2-650M (Lin et al., 2023) to precompute one sequence-derived representation per protein. Specifically, we mean-pool its final-layer residue representations over at most the first 1,022 residues; the resulting protein-level vectors remain fixed during ERCL training. In each relational context graph, these vectors are projected

into the graph encoder hidden space for node initialization. They are also used to compute cosine similarities for ranking candidate PPI paths. Separately, we use precomputed MMseqs2 alignment bit scores (Steinegger & Söding, 2017) to retrieve homologous anchors, weight query–anchor virtual edges, and add supplementary sequence-derived virtual edges when the bit score exceeds the fixed threshold.

Context construction configuration. We use the same context-construction settings across all five NS test sets. The retrieval graph is the largest connected component of the human training PPI network constructed from recorded positive interactions. For each query pair, we select up to K homologous anchors for each endpoint using MMseqs2 alignment bit scores. For each pair of anchors from the two endpoints, we search for simple PPI paths subject to a fixed hop limit and a cap on the number of candidate paths enumerated. Among the paths enumerated for each anchor pair, we retain up to B with the highest query-specific path scores defined in Section 5.2.2. The selected anchors and internal nodes of the retained paths define the retrieved protein set; all PPI edges induced by this set in the retrieval graph are included as PPI evidence.

We connect each query protein to its selected anchors with virtual edges, independently of the supplementary-edge threshold. After the context node set has been fixed—comprising the two query proteins and the retrieved human proteins—we consider other pairs of nodes within this set with detected MMseqs2 matches and add supplementary sequence-derived virtual edges when their bit scores exceed the fixed threshold. Both types of virtual edge are weighted by the corresponding bit scores. Table A.3 summarizes the context-construction settings used in our experiments.

Table A.3: Context-construction configuration used by ERCL.

Configuration	Value
Anchors per query protein K	5
Maximum PPI path length L_{path}	4 hops
Retained paths per anchor pair B	200
Candidate-path enumeration cap per anchor pair	3,000
Supplementary virtual-edge threshold	50.0
PPI retrieval graph	Largest connected component of human training PPI network
PPI edge weight	1.0
Sequence-derived virtual-edge weight	MMseqs2 alignment bit score

Graph encoding and prediction. Each message-passing layer assigns a learnable embedding to each of the two relation types, shared by all edges of that type. For a virtual edge with alignment bit score w_{uv}, we normalize its weight as $\tilde{w}_{uv} = \mathrm{clip}_{[0,1]}\big(\log(1+w_{uv})/\log(1+1650)\big)$; observed PPI edges receive a normalized weight of 1. An RBF expansion ϕ, with fixed, evenly spaced centers over $[0,1]$ and a learnable width, maps the normalized weight to an edge feature vector. Within each attention head, the absolute difference between the projected representations of the edge's source and target nodes, together with the relation embedding and encoded edge weight, determines the attention score. Omitting layer and head indices for clarity,

$$e_{uv} = \mathrm{MLP}_{\mathrm{att}}([|f_{\mathrm{s}}(\mathbf{h}_u) - f_{\mathrm{t}}(\mathbf{h}_v)| \,\|\, \mathbf{r}_{t_{uv}} \,\|\, \phi(\tilde{w}_{uv})]) \,,$$
$$\mathbf{m}_v = \sum_{u \in \mathcal{N}(v)} \alpha_{uv} f_{\mathrm{m}}(\mathbf{h}_u).$$

Here, t_{uv} denotes the edge relation type, $f_{\mathrm{s}}, f_{\mathrm{t}}, f_{\mathrm{m}}$ are learned projections, and $\|$ denotes concatenation. The attention coefficients α_{uv} are obtained by applying softmax to the scores over the incoming neighbors $\mathcal{N}(v)$, jointly across both relation types. The outputs of all attention heads are concatenated, followed by an ELU activation, a residual connection, and layer normalization to obtain the next-layer node representation.

After message passing, we use the two query endpoints to condition an attention readout over all nodes in the context graph, including the query nodes. Let $\mathcal{V} = V(\mathcal{G}_{ij}^{\mathrm{ctx}})$ and write $\mathbf{h}_v = \mathbf{h}_v^{(L)}$. We form the symmetric endpoint representation $\mathbf{s}_{ij} = [\mathbf{h}_{p_i} \odot \mathbf{h}_{p_j} \| |\mathbf{h}_{p_i} - \mathbf{h}_{p_j}|]$, where $\odot$ denotes

elementwise multiplication. A linear projection followed by GELU and layer normalization maps $\mathbf{s}_{ij}$ to a readout query $\mathbf{q}_{ij}$. A shared scoring MLP then assigns attention weights to the nodes, and a separate layer normalization is applied to their weighted sum:

$$\beta_v = \mathrm{softmax}_{v\in\mathcal{V}}(\mathrm{MLP}_{\mathrm{read}}([\mathbf{h}_v \| \mathbf{q}_{ij}]))\,,$$
$$\mathbf{g}_{ij} = \mathrm{LN}\left(\sum_{v\in\mathcal{V}} \beta_v \mathbf{h}_v\right).$$

The softmax is taken over all nodes of the individual context graph, and LN denotes layer normalization. The concatenated vector $[\mathbf{s}_{ij}\|\mathbf{g}_{ij}]$ is passed to an MLP classifier that produces a scalar logit, which is converted to an interaction probability by sigmoid. For a fixed encoded context graph, this readout and prediction are invariant to exchanging the two query endpoints. Table A.4 summarizes the architecture settings.

Table A.4: Architecture settings of the graph encoder and predictor.

Setting	Value
Message-passing layers	3
Hidden dimension	256
Attention heads per layer	4
Edge feature dimensions	32-dimensional relation embeddings; 32 RBF centers
Edge-attention MLP	$128 \to 64 \to 1$; LeakyReLU with negative slope 0.2
Readout query projection	Linear $512 \to 256$, followed by GELU and LayerNorm
Node-scoring MLP	$512 \to 128 \to 1$; GELU between linear layers
Classifier	$768 \to 256 \to 128 \to 1$; GELU between linear layers; dropout 0.2

Note. The weights connecting the graph-level context vector to the first classifier layer are initialized to zero and remain trainable.

Training configuration. After constructing the context from the query-masked source graph defined in Section 5.4, we exclude any recorded PPI edges incident to either query protein that appear in the assembled training graph from message passing. The optimization settings and checkpoint-selection rule are summarized in Table A.5.

Table A.5: Training configuration used for ERCL.

Configuration	Value
Optimizer	AdamW
Learning rate	1×10^{-4}
Weight decay	1×10^{-4}
Training batch size	64
Loss	Binary focal loss ($\alpha_{\mathrm{pos}} = 0.9,\ \gamma = 2.0$)
Learning-rate scheduler	ReduceLROnPlateau
Scheduler factor and patience	0.5; 3 epochs
Sequence-derived virtual-edge dropout	0.15 (training only)
Checkpoint selection	Highest human validation AUPR within first 10 epochs
Random seeds	42, 123, 2026

Configurations for relational context analysis. For the comparisons in Section 6.3, Endpoint-only applies the same input projection to the two query proteins' frozen ESM-2 representations and feeds their elementwise product and absolute difference to an MLP classifier, without graph message passing. The weighted RGCN and RGAT configurations use the same context graphs and node features as ERCL. The former uses relation-specific linear messages with edge-weighted mean aggregation; the latter uses relation-specific attention, normalized over incoming neighbors within each relation, with normalized edge weights incorporated into both attention scores and messages. Both graph-based configurations share ERCL's message-passing depth, hidden dimension, edge-weight normalization, and virtual-edge dropout. All three configurations classify the symmetric endpoint representation

without the graph-level attention readout. They use the loss, optimizer, learning rate, weight decay, learning-rate schedule, training batch size, random seeds, and checkpoint-selection rule in Table A.5.

A.3 Adapted Network-Informed Baselines

For all three adaptations, observed PPI information comes from the human training network constructed from recorded positive interactions; none uses target-species PPI edges at NS evaluation. All three use the same frozen, precomputed ESM-2-650M protein representations as ERCL.

GNN-PPI adaptation. Our GNN-PPI adaptation (Lv et al., 2021) uses a PPI graph constructed from the recorded positive interactions in the human training split; the same graph is used for training and human validation. For positive training queries, the queried edge is masked before message passing, so the interaction being predicted is not supplied as graph input. At NS evaluation, target-species proteins are added to the human training graph as isolated nodes with their sequence-derived features. No target-species PPI edges are added, leaving these proteins without observed neighbors from which to aggregate PPI information.

The original GNN-PPI uses a trainable CNN–BiGRU sequence encoder to obtain protein features before GIN-based network encoding, and predicts multiple interaction types. In our adaptation, the shared frozen ESM-2-650M protein representations replace that sequence encoder and are passed through a trainable projection to the GIN input space. The GIN encoder produces node representations from the available graph, and a binary head scores each query pair (u, v) as

$$\hat{p}_{uv} = \sigma\big(\mathbf{w}^\top(\mathbf{h}_u \odot \mathbf{h}_v) + b\big)\,,$$

where $\mathbf{h}_u$ and $\mathbf{h}_v$ are the encoded endpoint representations and $\odot$ denotes elementwise multiplication. This retains GNN-PPI's GIN-based network encoding and endpoint combination while using the same protein representations as ERCL and predicting the binary PPI label required by our evaluation.

KSGPPI-style adaptation. KSGPPI (Hu et al., 2024) supplies a query protein with network information through a sequence-matched reference protein. In our adaptation, we learn Node2Vec representations (Grover & Leskovec, 2016) from the positive-interaction graph of the human training split and keep them fixed during predictor training. For each protein, the highest-bit-score valid MMseqs2 match among human reference proteins supplies the reference protein's Node2Vec vector. Human training and validation proteins without a valid match use their own Node2Vec vectors; non-human proteins without a valid match receive a zero vector as input to the network branch. Thus, NS test proteins obtain network-derived features by transfer, without using target-species PPI edges.

At each endpoint, the query protein's ESM-2-650M representation and its assigned network vector are transformed separately, concatenated, and projected to a fused representation $\mathbf{z}_p$. For a query pair (u, v), a binary classifier receives the concatenation of the elementwise product $\mathbf{z}_u \odot \mathbf{z}_v$ and absolute difference $|\mathbf{z}_u - \mathbf{z}_v|$. Relative to the original KSGPPI, which uses a STRING reference network, NW-align matching, and an ESM-2/CKSAAP-based CNN sequence branch, our adaptation retains the combination of sequence and transferred network information while using the reference graph, matching procedure, and protein features described above for strictly inductive binary PPI prediction.

L3-PPI-style adaptation. We implement an L3-PPI-style adaptation (Gao et al., 2026) using the recorded positive interactions in the human training split as the only source of real PPI paths. From this graph, we enumerate simple three-hop paths and retain a deterministic uniform sample without class balancing. Each four-node path graph contains the three path edges and fixed ESM-2-650M node features; its label indicates whether the two endpoints interact in the human training graph. These examples are used to train a GIN graph classifier, which is then frozen as the surrogate.

For each query pair, the two endpoint representations and shared learnable prompt nodes form candidate virtual L3 paths. Each path connects the endpoints through an intermediate prompt node and a shared prompt hub. A path gate controls the candidate paths, and the frozen surrogate scores the resulting edge-weighted prompt graph. At NS evaluation, no target-species PPI edges are used, and no real path from the human training graph is retrieved for the query pair.

Training first optimizes the prompt embeddings with all candidate paths active. After this stage converges, the prompts and gate are optimized jointly using a binary classification loss and a path-number regularizer that favors more active paths for positive pairs and fewer for negative pairs. The

final predictor checkpoint is selected by human validation AUPR. The original L3-PPI is designed as a plug-and-play head for protein embeddings from an existing predictor; our adaptation uses these fixed representations for both surrogate pretraining and query-pair prediction, with real-path examples drawn only from the human training graph.

Table A.6 summarizes the key implementation settings of the three adapted baselines.

Table A.6: Key implementation settings for the adapted network-informed baselines.

Setting	Value
GNN-PPI adaptation	
Input projection and GIN	$1{,}280 \to 256$ input projection; one GIN layer with 512 hidden dimensions
Predictor optimization	Adam; learning rate 1×10^{-3}; weight decay 5×10^{-4}; batch size 64
Predictor loss	Binary focal loss ($\alpha = 0.9, \gamma = 2.0$)
KSGPPI-style adaptation	
Node2Vec representation	100 dimensions
Node2Vec walks	Walk length 20; context size 10; 10 walks per node; $p = q = 1$
Node2Vec optimization	SparseAdam; learning rate 0.01; 10 epochs
Endpoint representation widths	192-dimensional sequence branch; 64-dimensional network branch; 256-dimensional fused representation
Predictor optimization	AdamW; learning rate 1×10^{-4}; weight decay 1×10^{-4}; batch size 64
Predictor loss	Binary focal loss ($\alpha = 0.9, \gamma = 2.0$)
L3-PPI-style adaptation	
Real-path sample caps	200,000 training paths; 50,000 internal-validation paths
Surrogate architecture	Two-layer weighted GIN; 128 hidden dimensions
Virtual paths and gate	16 candidate paths; two-layer gate with 128 hidden dimensions
Surrogate optimization	Adam; learning rate 1×10^{-4}; batch size 64
Prompt/joint-stage optimization	Adam; learning rate 1×10^{-4}; batch size 64
Classification loss	Binary cross-entropy with logits for surrogate and predictor
Joint-stage path regularization	$\gamma = 2.0$; loss weight 0.3
Prompt/joint-stage convergence	Training-loss range below 0.05 over the last 10 epochs of each stage

Note. The L3 surrogate checkpoint is selected by AUPR on an internal validation split of human-training paths. The final checkpoint for each adapted predictor is selected by human validation AUPR.

A.4 Variability Across Random Seeds

Table A.7: Sample standard deviations of AUPR across three random seeds for models trained in this study. Corresponding means are reported in the main results tables.

	BS validation	NS test					
Configuration	Human	Mouse	Fly	Worm	Yeast	*E. coli*	AVG.
GNN-PPI adaptation	0.001	0.020	0.021	0.033	0.011	0.012	0.019
KSGPPI-style	0.002	0.004	0.006	0.003	0.018	0.019	0.010
L3-PPI-style	0.018	0.032	0.047	0.061	0.074	0.041	0.050
Endpoint-only	0.006	0.002	0.005	0.004	0.012	0.007	0.004
ERCL (weighted RGCN)	0.001	0.001	0.001	0.004	0.006	0.017	0.003
ERCL (RGAT)	0.002	0.002	0.003	0.002	0.003	0.016	0.004
ERCL	0.001	0.002	0.002	0.004	0.007	0.008	0.003

Note. AVG. is the sample standard deviation of the unweighted mean AUPR across the five NS test species, computed separately within each seed; it is not the average of the five species-level standard deviations. Dashes indicate that the corresponding human BS validation result is not reported in the main tables.

A.5 Efficiency Measurement Details

Benchmark scope. The prediction-stage comparison in Figure 3 evaluates ERCL and PLM-interact across the five NS test sets. For each species, we sample 1,000 test pairs and 20 disjoint warm-up pairs using sampling seed 42. Both models use the same measured pairs within each species, with batch size one. The timer covers each model's complete forward pass after its inputs have been prepared and transferred to the GPU. ERCL receives precomputed features and assembled relational context graphs; PLM-interact receives tokenized protein pairs.

Measurement and aggregation. The models are measured in separate processes on the same GPU using float32. We make three timed passes over the fixed pairs. For each species, forward latency is the mean of the CUDA-event measurements across pairs and passes; Figure 3 reports the unweighted mean of the five species-level latencies. Peak GPU memory is the maximum allocated CUDA memory, including the loaded model, observed across all measured pairs, passes, and species. Parameter counts cover the loaded prediction models and exclude ERCL's separate frozen feature extractor. The timed passes are repeated measurements rather than independent training runs. Table A.8 reports the species-level latency and peak-memory measurements.

Table A.8: Prediction-stage forward latency and peak GPU memory on the five NS test sets.

	Forward latency (ms/pair)		Peak GPU memory (MiB)	
Test set	PLM-interact	ERCL	PLM-interact	ERCL
Mouse	99.076	2.986	3023.49	132.51
Fly	96.724	2.993	3028.13	144.33
Worm	86.922	3.107	3015.27	133.28
Yeast	86.587	2.932	2997.36	135.62
E. coli	69.487	2.939	2982.80	127.61

Note. Measurements use prepared inputs with batch size 1. Forward latency is averaged over the measured pairs and three timed passes for each species. Peak GPU memory is the maximum allocated CUDA memory, including the loaded model. Figure 3 reports the unweighted mean of the five species-level latencies and the maximum of the five species-level memory peaks.

A.6 Retrieval Coverage and Shortest-Path Distributions

Analysis setup. We analyze the availability and structure of PPI paths connecting the relational anchors of NS query pairs. For each endpoint, we use up to five retrieved anchors and examine paths

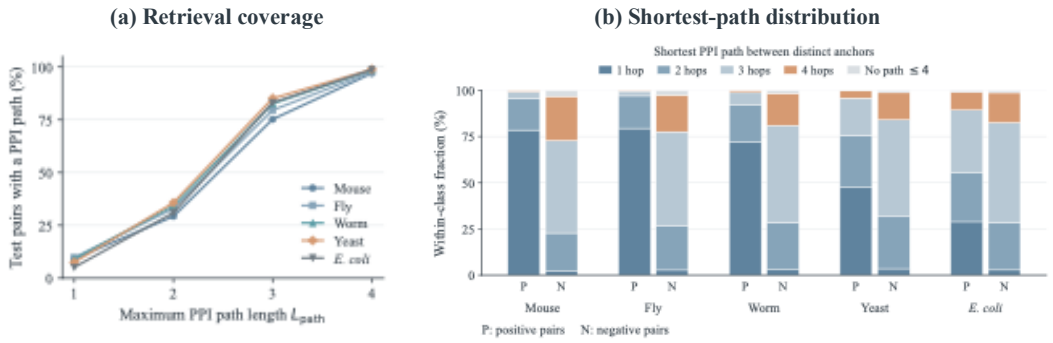

Figure A.1: Retrieval coverage and shortest-path distributions across the five NS test sets. (a) Fraction of query pairs with at least one PPI path connecting distinct anchors within the specified hop limit. (b) Distribution of the shortest connecting path for positive (P) and negative (N) pairs, normalized separately within each class. "No path ≤ 4" denotes pairs without a connecting path of at most four hops.

between distinct anchors from the two endpoints in the human training PPI network. Path length counts only observed PPI edges, excluding query–anchor virtual edges. The analysis includes all test pairs: 55,000 each for mouse, fly, worm, and yeast, and 22,000 for *E. coli*, with a positive-to-negative ratio of 1:10 in each test set. Labels are used only to group the resulting statistics.

Retrieval coverage. A query pair is counted as covered when at least one path connects its two anchor sets within the specified hop limit. As shown in Figure A.1(a), increasing the maximum path length from three to four hops raises coverage from 75.2–85.1% to 97.0–98.9% across the five test sets. Thus, four-hop retrieval makes PPI path evidence available for most NS query pairs.

Class-conditional shortest-path distributions. Figure A.1(b) groups query pairs by the shortest path connecting distinct anchors, with fractions normalized separately within each class. Positive pairs more frequently have directly connected anchors: the one-hop fraction ranges from 28.9% to 79.4%, compared with 2.2–3.5% for negative pairs. Negative pairs most commonly have a shortest connecting path of three hops across all five test sets. These descriptive statistics show that high overall path coverage coexists with differences in anchor connectivity between positive and negative pairs.