FROM PAIRS TO NETWORKS: MOTIF GRAPH PROMPT TUNING FOR INDUCTIVE PROTEIN–PROTEIN INTERACTION PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Predicting protein–protein interactions (PPIs) is essential for reconstructing the networks that organize cellular function. Predictors trained on individual pairs can score pairs accurately yet assemble networks with distorted topology, and supplying network context is hardest when both queried proteins are unseen and have no recorded interactions. Our key idea is that topology, although unavailable for the queried proteins, is fully observed during training, so it can serve as supervision rather than as input, at two levels. We propose GraSPPI (GRAph Structural Prompting for PPI), a motif graph prompt tuning method inspired by prompt tuning in language, vision-language, and graph models. At the pair level, it infers from the two sequences a virtual motif graph of the pair's triadic closure, the shared partners that interacting proteins tend to have; this graph is learned against the true motif graphs of the training network and read as a prompt for the interaction decision. At the network level, structural losses regularise training: the decisions on sampled training subgraphs are scored jointly, so that the network they form follows the structure of the training network, while every test pair is still scored independently. Extensive experiments demonstrate that GraSPPI is effective in reconstructing held-out interaction networks, with the lowest clustering discrepancy among the compared methods and the highest complex-pathway recall.

1 INTRODUCTION

Protein–protein interactions (PPIs) organize cellular function, and their network links disease mechanisms to therapeutic targets (Barabási & Oltvai, 2004; Vidal et al., 2011; Menche et al., 2015; Cheng et al., 2018). Experimental maps of this network remain incomplete and concentrate on well-studied proteins (Luck et al., 2020; Rocha et al., 2023), so computational predictors are asked to find the partners of proteins with no recorded interaction. Their predictions are assembled into networks on which downstream analyses of complexes, functional modules, and essential proteins build (Humphreys et al., 2021; Zheng et al., 2025).

Most PPI predictors decide one pair at a time from the two protein sequences (Ko et al., 2024; Liu et al., 2025), and even when their individual decisions are accurate, the networks those decisions assemble can be too dense or distorted in degree and clustering (Zheng et al., 2025). What they lack is the network context of the pair, and proteins without recorded interactions do not have it: link predictors read the observed neighbourhood of a query (Zhang & Chen, 2018; Wang et al., 2024), while graph distillation folds training topology into a sequence-only student that still scores each pair without context (Zhang et al., 2022; Guo et al., 2023). Graph prompt tuning has been brought to PPI prediction to inject an interaction prior into each pair's decision (Gao et al., 2026), but the prompt is supervised only through the pair's label and each decision is trained on its own, so no training signal checks the prompt against the network's observed structure, and nothing asks whether the decisions together form a network with the structure the prior describes. This leaves a central challenge: *given only two unseen proteins, how can a predictor decide their interaction in light of network topology, both around the pair and across the network its decisions form?*

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.

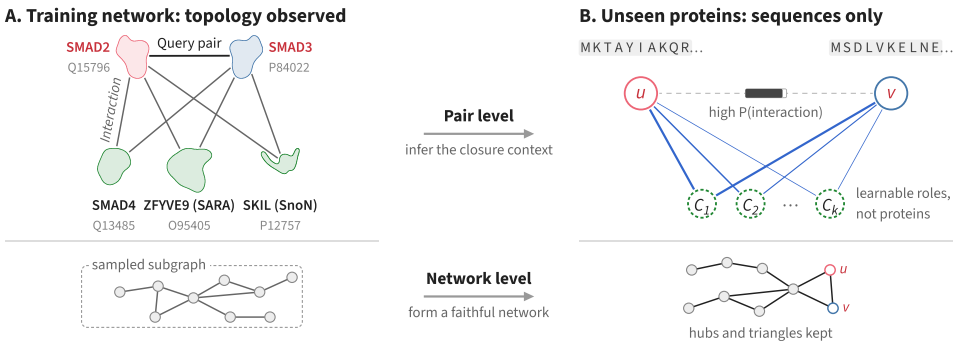

Figure 1: Topology supervises GraSPPI at two levels. **Pair level:** in the training network (A, top), interacting proteins such as SMAD2 and SMAD3 share partners (SMAD4, SARA, SnoN), so their interaction closes triangles (triadic closure). For two unseen proteins u and v (B, top), GraSPPI infers this closure context from the two sequences as a motif graph prompt, a virtual motif graph whose nodes $C_1, \ldots, C_k$ are learnable roles rather than real proteins; the inferred closure (blue) is read as learned evidence for the interaction decision. **Network level:** the decisions on a sampled training subgraph (A, bottom) are trained jointly to reproduce its structure, so that the network assembled from test decisions (B, bottom) is structurally faithful. Interactions in (A, top) are from the STRING physical network (Szklarczyk et al., 2023); the other networks are schematic.

Our key idea is that topology, although unavailable for the queried proteins, is fully observed during training, so it can serve as supervision rather than as input. We use it at two levels (Fig. 1). At the *pair level*, the training network shows the context that surrounds an interacting pair: proteins that share partners tend to interact, because members of one complex bind a common set of partners, a pattern known as triadic closure (Goldberg & Roth, 2003; Spirin & Mirny, 2003). On our benchmarks, closure marks interacting pairs beyond what protein degree explains (Section 4.2). We represent this closure context as a prompt: a small virtual motif graph whose nodes are learnable structural roles rather than real proteins, which the model infers for any pair from its two sequences. This motif graph prompt tuning is inspired by prompt tuning, which steers large language and vision-language models with learnable prompt tokens or attention prefixes instead of fine-tuning their weights (Lester et al., 2021; Li & Liang, 2021; Zhou et al., 2022b), and by graph prompt tuning, which builds such prompts from learnable virtual nodes and prompt graphs (Sun et al., 2023; Tan et al., 2023). At the *network level*, the training network shows what the assembled network should look like, so we train the decisions on sampled training subgraphs jointly rather than one pair at a time.

We instantiate this idea as GraSPPI (GRAph Structural Prompting for PPI), a motif graph prompt tuning method. At the pair level, a fixed closure template gives every pair context of the same form, a set of shared-neighbour paths between its two proteins. The model first learns to read true motif graphs compiled from the training network and then learns to infer them from the two sequences, supervised by the true graphs, so that at test time the predictor reads an inferred graph as the prompt for its decision. At the network level, structural losses score the decisions on a sampled training subgraph together and compare the network they form with the true subgraph. A test pair needs only its two sequences.

The main contributions of this work are:

- We propose motif graph prompt tuning, which infers the triadic-closure context of a protein pair from its two sequences as a virtual motif graph, learned against the true motif graphs of the training network and read by the predictor as a prompt for the interaction decision.
- We introduce network-level structural losses, a training-time regulariser that scores the decisions on sampled training subgraphs jointly, so that the network they form follows the structure of the training network.
- Extensive experiments demonstrate that GraSPPI is effective in reconstructing held-out networks, with the lowest clustering discrepancy among the compared methods, competitive complex recovery, and pair rankings on par with strong baselines; ablations show that each component contributes.

2 Related Work

Protein–protein interaction prediction. Sequence-based PPI prediction has moved from hand-crafted descriptors with classical classifiers (Shen et al., 2007; Guo et al., 2008) to deep networks (Hashemifar et al., 2018; Chen et al., 2019) and protein language models (Lin et al., 2023; Hayes et al., 2025). These models now underpin predictors that infer inter-protein contacts, model residue dependencies with uncertainty, or jointly encode protein pairs (Sledzieski et al., 2021; Ko et al., 2024; Liu et al., 2025). Evaluations that share proteins between training and test, or let node degree act as a shortcut, inflate pairwise accuracy (Park & Marcotte, 2012; Bernett et al., 2024; Aiyappa et al., 2024; Yılmaz et al., 2025), and PRING shows that accurate pairwise predictors can still assemble distorted networks (Zheng et al., 2025). Our predictor also reads only the two queried sequences, but its training supervises both the context of each pair and the network its decisions form.

Link prediction and knowledge distillation. Link predictors encode observed connectivity or link-specific evidence such as enclosing subgraphs and common neighbours (Kipf & Welling, 2016; Zhang & Chen, 2018; Chamberlain et al., 2023; Wang et al., 2024), which two unseen endpoints lack, and graph structure learning repairs the adjacency used for representation learning rather than the network assembled from a predictor's outputs (Franceschi et al., 2019; Jin et al., 2020). Graph distillation instead trains a structure-free student from a teacher that reads training structure (Zhang et al., 2022; Zheng et al., 2022); for links, students match anchor-to-context rankings or topology-aware representations (Guo et al., 2023; Samy et al., 2023; Deng et al., 2026). These students absorb training topology into their parameters and score each pair without structural context, whereas our predictor receives an explicit motif graph per query and is supervised on the network its decisions form.

Prompt tuning and graph prompt tuning. Prompt tuning adapts a pretrained model through learnable prompt tokens or attention prefixes instead of fine-tuning its weights, in language models (Lester et al., 2021; Li & Liang, 2021) and in vision and vision-language models, where the prompt can also be conditioned on each input (Zhou et al., 2022b;a; Jia et al., 2022). Graph prompt tuning carries the idea to graphs through feature prompts, virtual nodes, and prompt graphs (Sun et al., 2022; Fang et al., 2023; Tan et al., 2023; Sun et al., 2023; Huang et al., 2025). L3-PPI brings it to PPI prediction: it gates virtual L3 paths between the two proteins, scores them with a GNN pretrained on the L3 paths of the training network and then frozen, and supervises the prompt only through label-based losses (Gao et al., 2026). Our motif graph prompt tuning likewise infers a per-pair prompt graph, but it encodes triadic closure rather than L3 paths and is learned against the pair's true motif graph rather than the interaction label, while structural losses supervise the network its decisions form across pairs (Section 4.5 isolates each difference). For us the prefixes are an information route rather than a parameter-efficiency device: unlike frozen-backbone prompt tuning, the predictor itself is trained (Section 3.2).

3 Methodology

Task formulation. Given the amino-acid sequences s_u and s_v of a queried protein pair (u, v), protein–protein interaction (PPI) prediction aims to output an interaction probability $\widehat{A}_{uv}$ and a binary decision $\widehat{y}_{uv}$,

$$\widehat{A}_{uv} = f(s_u, s_v) = f(s_v, s_u) \in [0, 1], \qquad \widehat{y}_{uv} = \mathbf{1}\big[\widehat{A}_{uv} \geq \tau\big], \tag{1}$$

where f is the model, symmetric in its two inputs, and τ is one decision threshold shared by all queried pairs. Applied to every pair of a protein set $\mathcal{V}_{\mathrm{te}}$, the decisions form the predicted network $\widehat{\mathcal{E}} = \{\{u, v\} : u, v \in \mathcal{V}_{\mathrm{te}},\ \widehat{y}_{uv} = 1\}$ ($u = v$ allowed), which should recover the true interactions $\mathcal{E}_{\mathrm{te}}$ pair by pair and match their degree, clustering, and spectral structure. The task is inductive: the only known interactions form a network $\mathcal{G}_{\mathrm{tr}} = (\mathcal{V}_{\mathrm{tr}}, \mathcal{E}_{\mathrm{tr}})$ on proteins disjoint from $\mathcal{V}_{\mathrm{te}}$, so no queried protein has a known partner and f decides from the two sequences alone. Inside f, we infer a small motif graph $\widehat{B}_{uv}$ (Section 3.1) that serves only as context for the single decision $\widehat{y}_{uv}$. The training network $\mathcal{G}_{\mathrm{tr}}$ supervises f at two levels (Section 3.2): at the pair level, the inferred motif graph is learned against true motif graphs compiled from $\mathcal{G}_{\mathrm{tr}}$, and at the network level, structural losses score the decisions on sampled training subgraphs together.

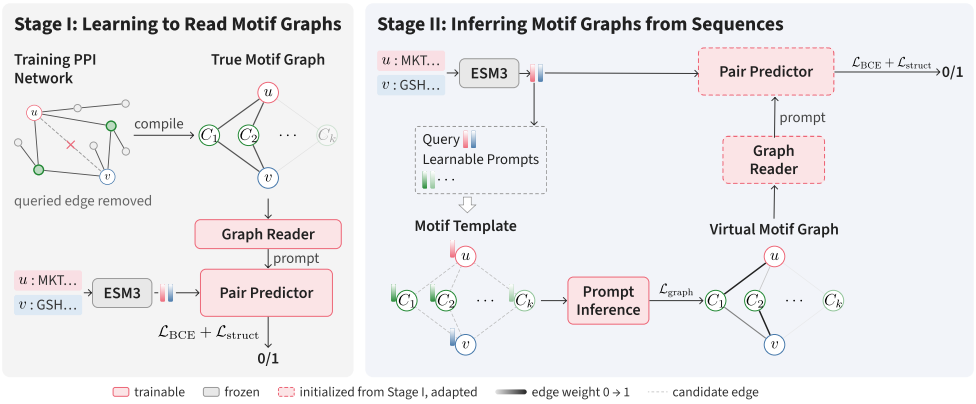

Figure 2: Overview of motif graph prompt tuning. *Stage I:* a true motif graph, compiled from the training network with the queried edge removed, places the pair's shared neighbours in the closure slots of a fixed template; the graph reader turns it into prompts for the pair predictor, trained with pairwise cross-entropy and structural losses on sampled subgraphs. *Stage II:* learnable prompt nodes read the two sequences and infer edge weights on the same template, supervised by the motif-graph loss; the reader and predictor, initialised from Stage I, adapt to the inferred graphs. At test time only the two sequences are needed.

3.1 Architecture

Overall architecture. In Fig. 2, a pair predictor encodes each residue sequence and then mixes the two through cross-attention layers; a motif graph enters those layers through a graph reader and gated attention prefixes. The design rests on three choices. First, a fixed motif template turns network context into a graph whose nodes are structural roles rather than proteins, so every pair receives context of the same form: the shared-neighbour paths that triadic closure predicts. Second, the graph conditions the predictor only through zero-initialised gated prefixes, so closing the gates recovers the sequence-only predictor. Third, so that the reader learns to use exact graphs before it reads inferred ones, training has two stages (Section 3.2): Stage I trains the reader and predictor on true motif graphs compiled from the training network, and Stage II trains a prompt inference module to predict those graphs from the sequences while the predictor adapts.

Sequence encoder. Each protein sequence s_i of length L_i is first embedded per residue by the pre-trained ESM3 encoder (Hayes et al., 2025), $X_i = E_\eta(s_i) \in \mathbb{R}^{L_i \times d_0}$, whose parameters η stay fixed and whose embedding width is d_0. A residue encoder F_{θ_e} then projects the embeddings linearly to width d and applies L_e transformer self-attention layers within each sequence, with weights shared by the two proteins, giving residue states $H_i = F_{\theta_e}(X_i) \in \mathbb{R}^{L_i \times d}$, which both the pair predictor and prompt inference read. Encoding each sequence on its own adapts the fixed embeddings to the interaction task before the two proteins are compared, and sharing the weights gives a protein the same states in either endpoint position.

Motif template. The template has the query endpoints u, v and k prompt nodes, the closure slots $\{C_c\}_{c=1}^{k}$. Its candidate edges are

$$\mathcal{E}_C = \{(u, C_c), (C_c, v)\}_{c=1}^{k}, \tag{2}$$

so each slot contributes a closure path $u \to C_c \to v$ that stands for one shared neighbour, a triangle the queried edge would close. A motif graph assigns a weight in $[0, 1]$ to each of the $2k$ candidate edges; its symmetric adjacency $B \in [0, 1]^{(k+2)\times(k+2)}$ is zero off the template and never contains the queried edge (B_{uv} names the pair's whole graph, B_{ij} one entry). For a training pair, the *true motif graph* B^*_{uv} is compiled from $\mathcal{G}_{\text{tr}}$ with the queried edge removed: up to k shared neighbours fill the closure slots, each edge weighted by the inverse square root of the neighbour's degree, and unfilled slots stay at zero (Section 6.1). At test time, the *virtual motif graph* $\widehat{B}_{uv}$ is inferred from sequences.

Prompt inference. The prompt inference module P_ϕ maps the residue states to a virtual motif graph and a presence probability, $(\widehat{B}_{uv}, \pi_{uv}) = P_\phi(H_u, H_v)$. Each of the k closure slots carries a

hidden state of dimension d_g, which learnable prompt nodes obtain from the residue states through residual cross-attention: shared queries $\{q_c\}_{c=1}^{k} \subset \mathbb{R}^{d_g}$, one per slot, read each endpoint separately, and the two reads merge symmetrically into slot states $h_{C_c} \in \mathbb{R}^{d_g}$ (Section 6.1). With endpoint states $h_u = W_g \,\mathrm{Mean}(H_u)$ and $h_v = W_g \,\mathrm{Mean}(H_v)$, where $W_g \in \mathbb{R}^{d_g \times d}$ projects mean-pooled residue states to width d_g, all $k+2$ states pass through message-passing layers on the template. From symmetric edge features $e_{ij} = [h_i + h_j, |h_i - h_j|]$, an edge head predicts a logit ξ_{ij} for every candidate edge, and a presence head on the mean edge feature predicts the probability π_{uv} that the pair has a shared neighbour at all. With the logistic function σ, the virtual motif graph of the pair is

$$\widehat{B}_{ij} = \pi_{uv}\, \sigma(\xi_{ij}), \qquad (i,j) \in \mathcal{E}_C, \tag{3}$$

so π_{uv} scales every edge weight at once, and a pair without shared neighbours can shrink its whole graph through π_{uv}. All weights stay continuous, and the two edges of a slot are predicted separately; swapping u and v swaps the two edges of every slot.

Graph reader. A GRIT graph transformer (Ma et al., 2023) R_{θ_r} encodes the motif graph with learned role embeddings for endpoints and closure slots, the edge weights, and relative random-walk features of the weighted template (Section 6.1). It returns endpoint tokens and a relation token, $(t_u, t_v, t_{\mathrm{rel}}) = R_{\theta_r}(B)$, where the relation token is a symmetric function of the reader's two directed pair states (Section 6.1). Swapping u and v exchanges t_u and t_v and leaves t_{rel} unchanged.

Prompt prefixes. A prefix, as in prefix tuning (Li & Liang, 2021), is a set of learned key–value tokens read alongside an attention layer's input tokens. The adapter Π_{θ_p} arranges the reader's tokens into self, partner, and relation fields, $(t_u, t_v, t_{\mathrm{rel}})$ for the u stream and $(t_v, t_u, t_{\mathrm{rel}})$ for the v stream. Role embeddings and layer-specific projections expand each field into n_p prefix keys and values (K^p, V^p) for every cross-attention site of the pair predictor. The prefixes enter through a separately normalised branch,

$$\mathrm{Attn}^p(Q,K,V) = \mathrm{softmax}\Big(\tfrac{QK^\top}{\sqrt{d_h}}\Big)V + \tanh(\alpha_{\ell a h})\,\mathrm{softmax}\Big(\tfrac{Q(K^p)^\top}{\sqrt{d_h}}\Big)V^p, \tag{4}$$

written for one head, where Q holds the site's queries, K, V the keys and values of the residues it attends to, $d_h = d/n_h$ is the head width for the trunk's n_h attention heads, and $\alpha_{\ell a h}$ is a zero-initialised gate for layer ℓ, site a, and head h; the residue attention itself is unchanged.

Pair prediction. The pair classifier C_{θ_c} mixes the residue states of the two proteins through L_c bidirectional cross-attention layers. Each layer has three attention sites: the residues of u attend to those of v, the residues of v attend to those of u, and a classification token attends to both; every site reads the prompt prefixes through Eq. (4). After the last layer, each endpoint's residue states are pooled and combined with the classification token into a pair representation z_{uv} (Section 6.1). With B_{uv} the motif graph given to the reader, the true graph B^*_{uv} in Stage I and the inferred graph $\widehat{B}_{uv}$ in Stage II (Section 3.2),

$$\widehat{A}_{uv} = \sigma\big(\mathrm{MLP}(\max(z_{uv}, z_{vu}))\big), \qquad z_{uv} = C_{\theta_c}\big(H_u, H_v; \Pi_{\theta_p}(R_{\theta_r}(B_{uv}))\big), \tag{5}$$

where z_{vu} is computed in the swapped order from H_v, H_u and the swapped graph B_{vu}, whose two edges of every slot are exchanged; the elementwise maximum over the two orders makes the score symmetric. We write $\theta = (\theta_e, \theta_c, \theta_r, \theta_p)$ for the residue encoder, the classifier with its prediction head, the reader, and the adapter; η stays frozen.

3.2 Two-Stage Training and Inference

Stage I: learning to read motif graphs. We train θ from scratch with $B_{uv} = B^*_{uv}$. Two regularisers keep the predictor from relying on exact true graphs: with probability p_b, a non-self pair's graph is blended toward the training mean, and with probability p_d all prefix gates of a pair are closed, which keeps sequence-only examples in training (Section 6.1). Training reads two streams of pairs. The task stream pairs each training edge with non-edges proposed, with equal probability, uniformly or by replacing one endpoint of a training edge with a degree-proportional draw, which weakens degree as a shortcut (Chatterjee et al., 2025). The structural stream draws one subgraph $\mathcal{S}$ of $\mathcal{G}_{\mathrm{tr}}$ per step and scores all its legal pairs: a BFS ball grown from a random protein, a wedge ball grown around an open wedge or a closed triangle, or a two-ball subgraph grown from two distant roots, each topped up with background proteins (Section 6.1). In each stream, $\mathcal{L}^s_{\mathrm{BCE}}$ is the binary cross-entropy with weight w_+ on positives, normalised by the total weight, and $\mathcal{L}_{\mathrm{BCE}} = \mathcal{L}^{\mathrm{task}}_{\mathrm{BCE}} + \mathcal{L}^{\mathrm{sub}}_{\mathrm{BCE}}$.

Network-level structural losses. Pairwise cross-entropy scores each decision alone and does not constrain the network the decisions form together, so three losses supervise the subgraph's predicted network. They regularise training only: at inference every pair is scored independently by Eq. (1), and the fidelity of the assembled network is assessed empirically (Section 4.3). Let Ω be the legal-pair mask of $\mathcal{S}$ with zero diagonal, Z the logits, $A_{\mathcal{S}}$ the adjacency of $\mathcal{S}$ in $\mathcal{G}_{\mathrm{tr}}$, $P = \Omega \odot \sigma(Z)$ and $A = \Omega \odot A_{\mathcal{S}}$ the predicted and true adjacency, $\mathbf{1}$ the all-ones vector, $\mathcal{N}_i^{\pm}$ the legal partners and non-partners of protein i in $\mathcal{S}$, and ℓ_δ the Huber loss with threshold δ (Huber, 1964). *Neighbour ranking* orders each protein's partners above its non-partners with the pairwise logistic loss of RankNet and BPR (Burges et al., 2005; Rendle et al., 2009), which link predictors use to optimise AUC (Wang et al., 2021):

$$\mathcal{L}_{\mathrm{rank}} = \frac{1}{|\mathcal{I}_r|} \sum_{i \in \mathcal{I}_r} \frac{1}{|\mathcal{N}_i^+|\,|\mathcal{N}_i^-|} \sum_{j \in \mathcal{N}_i^+} \sum_{j' \in \mathcal{N}_i^-} \mathrm{softplus}(Z_{ij'} - Z_{ij} + m), \tag{6}$$

where m is the ranking margin and $\mathcal{I}_r$ holds the proteins with at least one partner and one non-partner in $\mathcal{S}$. *Degree matching* compares, on a log scale, each protein's expected degree under independent edges with probabilities P (Chung & Lu, 2002) with its true degree, so that no protein receives far too many or too few partners:

$$\mathcal{L}_{\mathrm{deg}} = \frac{1}{|\mathcal{I}_d|} \sum_{i \in \mathcal{I}_d} \ell_\delta\Big(\log \frac{1 + [P\mathbf{1}]_i}{1 + [A\mathbf{1}]_i}\Big), \tag{7}$$

where $\mathcal{I}_d$ holds the proteins with at least one legal pair. *Motif matching* supervises two-hop structure, the pattern the closure template encodes. Under independent edges, $M^{\mathrm{c}}(P) = P \odot P^2$ and $M^{\mathrm{o}}(P) = (\mathbf{1}\mathbf{1}^\top - P) \odot P^2$ are the expected numbers of closed and open two-hop paths between two proteins, and edge-independent predictions tend to close too few of them (Seshadhri et al., 2020; Chanpuriya et al., 2021). The legal pairs of $\mathcal{S}$ split by their true paths into $\mathcal{E}_{\mathrm{c}}$ (a closed path), $\mathcal{E}_{\mathrm{o}}$ (only open paths), and $\mathcal{E}_0$ (neither); each class in the set $\mathcal{C}$ of non-empty classes receives equal weight, so the many pairs without two-hop paths do not dominate:

$$\mathcal{L}_{\mathrm{mot}} = \frac{1}{|\mathcal{C}|} \sum_{b \in \mathcal{C}} \frac{1}{|\mathcal{E}_b|} \sum_{(i,j) \in \mathcal{E}_b} \sum_{M \in \{M^{\mathrm{c}}, M^{\mathrm{o}}\}} \ell_\delta\Big(\log \frac{1 + M(P)_{ij}}{1 + M(A)_{ij}}\Big). \tag{8}$$

Stage I minimises $\mathcal{L}_{\mathrm{I}} = \mathcal{L}_{\mathrm{BCE}} + \mathcal{L}_{\mathrm{struct}}$ with $\mathcal{L}_{\mathrm{struct}} = \lambda_{\mathrm{rank}}\mathcal{L}_{\mathrm{rank}} + \lambda_{\mathrm{deg}}\mathcal{L}_{\mathrm{deg}} + \lambda_{\mathrm{mot}}\mathcal{L}_{\mathrm{mot}}$, where the nonnegative weights $\lambda_{\mathrm{rank}}, \lambda_{\mathrm{deg}}, \lambda_{\mathrm{mot}}$ balance the three structural losses. Its checkpoint is selected on validation motif graphs compiled from the validation network and initialises Stage II.

Stage II: inferring motif graphs from sequences. Stage II initialises θ from Stage I and adds P_ϕ. *Motif-graph supervision*, the pair-level signal, teaches P_ϕ the closure context that the Stage I reader has learned to use. Prompt slots have no fixed identity, so the loss compares sorted statistics rather than individual edges: this is set matching (Carion et al., 2020) in one dimension, where sorting gives the optimal assignment (Peyré & Cuturi, 2019), and a presence term plays the role of the empty class. Let $r(B) \in [0,1]^k$ list the closure path products $B_{uC_c}B_{C_cv}$ in decreasing order, and let $a_{uv} = \mathbf{1}\big[\sum_c B^*_{uC_c}B^*_{C_cv} > 0\big]$ indicate whether the true graph has a shared neighbour. Each step supervises the non-self rows of the two streams $s \in \{\mathrm{task}, \mathrm{sub}\}$; $\mathcal{R}_{s,a}$ holds the rows of stream s with indicator a, $\mathcal{A}_s$ the indicators present in stream s, and $\mathcal{S}_+$ the streams with at least one row. Averaging within each indicator first gives pairs with and without a shared neighbour equal weight:

$$\begin{aligned} \ell_{\mathrm{slot}}(\widehat{B}, B^*) &= \tfrac{1}{2k}\big\|r(\widehat{B}) - r(B^*)\big\|_2^2, \\ \mathcal{L}_{\mathrm{graph}} &= \frac{1}{|\mathcal{S}_+|} \sum_{s \in \mathcal{S}_+} \frac{1}{|\mathcal{A}_s|} \sum_{a \in \mathcal{A}_s} \frac{1}{|\mathcal{R}_{s,a}|} \sum_{(u,v) \in \mathcal{R}_{s,a}} \Big[\ell_{\mathrm{slot}}\big(\widehat{B}_{uv}, B^*_{uv}\big) + \lambda_\pi \,\mathrm{BCE}\big(\pi_{uv}, a_{uv}\big)\Big], \end{aligned} \tag{9}$$

where λ_π weighs the presence term. The loss matches the closure-path profile and the presence of a shared neighbour rather than every edge: how a slot's weight splits between its two edges is left to P_ϕ, and the reader adapts to that split below. In phase A, the first E_w epochs, only ϕ is trained and θ stays frozen. In phase B, θ adapts at reduced learning rates to the inferred graphs, with prompt dropout off, and stop-gradients $\mathrm{sg}(\cdot)$ separate the two objectives: for every training row,

$$(\widehat{B}_{uv}, \pi_{uv}) = P_\phi\big(\mathrm{sg}(H_u), \mathrm{sg}(H_v)\big), \qquad \mathcal{L}_{\mathrm{II}} = \mathcal{L}_{\mathrm{graph}} + \mathcal{L}_{\mathrm{I}}, \tag{10}$$

where both losses are the batch reductions above and the predictor in $\mathcal{L}_{\mathrm{I}}$ reads $\mathrm{sg}(\widehat{B}_{uv})$, so ϕ is trained only by $\mathcal{L}_{\mathrm{graph}}$ and θ only by $\mathcal{L}_{\mathrm{I}}$. The motif-graph loss alone shapes the inferred graph, while the task and structural losses adapt the reader, the prefixes, and the sequence trunk to it. Stage II checkpoints are selected with inferred graphs as the only structural input.

Inference. A test pair needs only its two sequences: the sequence encoder produces H_u, H_v, prompt inference produces $\widehat{B}_{uv}$, and Eq. (5) scores the pair without reading any training graph or neighbourhood. The threshold τ in Eq. (1) is fixed on the validation region and applied unchanged to all test pairs (Section 6.2.4).

4 Experiments

We test the motif graph's closure design (Section 4.2), network reconstruction and pairwise classification (Section 4.3), downstream function analyses (Section 4.4), and each module's contribution (Section 4.5).

4.1 Experimental Setup

Datasets. We evaluate on two leakage-free human PPI benchmarks (Section 6.2.1). PRING (Zheng et al., 2025) evaluates network reconstruction on test subgraphs that breadth-first (BFS), depth-first (DFS), and random-walk (RW) traversals sample from a held-out network of unseen proteins, one split per traversal. It also poses three function-oriented tasks on the BFS test network: recovering the interactions of Reactome complex pathways, preserving the enriched GO terms of its communities, and separating essential from non-essential proteins by network centrality. The gold standard of Bernett et al. (2024) evaluates pairwise classification on disjoint, sequence-dissimilar protein sets. For model selection we hold out a validation region $\mathcal{V}_{\mathrm{val}}$ from each training graph with the split's own traversal and remove every training pair that touches it; every method we run takes its threshold (and, if we train it, its checkpoint) from $\mathcal{V}_{\mathrm{val}}$ (Section 6.2.4).

Evaluation metrics. On PRING, each method's scores are thresholded once, at the $\mathcal{V}_{\mathrm{val}}$ threshold whose relative density is closest to one, into a predicted network on every test subgraph. Graph similarity (GS, the edge-set Dice score) measures the recovered edges and relative density (RD) the edge count; degree, clustering, and spectral MMD (You et al., 2018), each normalised by the discrepancy between two halves of the true subgraphs, measure the network shape (Section 6.2.3). The function-oriented tasks use PRING's metrics for complex recovery (PR, PP, PC), GO enrichment (FA, CR), and essential proteins (P@100, DO) on the whole BFS test network (Section 6.3.5). On Bernett we report AUROC, AUPRC, and accuracy, F1, and MCC at probability 0.5.

Baselines. We compare with sequence models (SPRINT (Li & Ilie, 2017), Topsy-Turvy (Singh et al., 2022)), PLM predictors (TUnA (Ko et al., 2024), PLM-interact (Liu et al., 2025), C3PI (Hosseini et al., 2025)), graph distillation (LLP (Guo et al., 2023), CAZI-MBN (Deng et al., 2026)), and graph prompt tuning (L3-PPI (Gao et al., 2026), UniPrompt (Huang et al., 2025)). Published results, including PRING's function results, keep their authors' protocol; the other rows are our reproductions on the frozen ESM3 embeddings (Hayes et al., 2025) and our training pairs under our selection rules, with L3-PPI, which has no official code, reproduced as a plug-in on our sequence encoder backbone (Section 6.2.2). All other settings are in Section 6.1.

4.2 Why Triadic Closure

Our motif graph gives each pair closure slots only. To check this design, we compare every interacting pair of each split's training graph, measured with its own edge removed, with a non-interacting pair from matched degree bins; no test protein is read (Section 6.3.1). Pairs with a shared neighbour interact in 78–94% of cases, against 19–27% of pairs without one (Fig. 3a), and the shared-neighbour count separates the two within every third of the smaller endpoint degree, where the degree product stays near chance (Fig. 3b). Kovács et al. (2019) argue that closure fails for PPI networks and that paths of length three (L3) predict interactions instead (Yuen & Jansson, 2023); on these graphs the L3 count also ranks pairs well alone, but adds little within bins of the shared-neighbour count (Section 6.3.1).

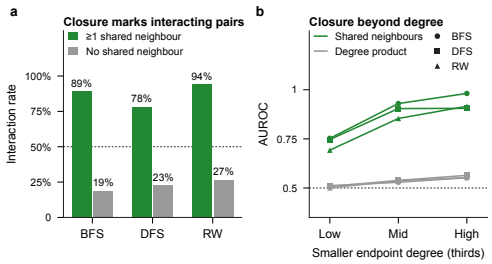

Figure 3: Triadic closure on the three PRING training graphs. Each interacting pair is measured with its own edge removed and matched with a non-interacting pair from the same degree bins, so chance (dashed) is 50% in (a) and 0.5 in (b). (a) Share of pairs that interact, among pairs with and without a shared neighbour. (b) AUROC of the shared-neighbour count and of the product of the two degrees within thirds of the smaller endpoint degree (Section 6.3.1).

Table 1: Network reconstruction on the three PRING splits.[2] Every method takes its threshold, and where we train it its checkpoint, from $\mathcal{V}_{\text{val}}$. Grey lines repeat the published probability-0.5 results of Zheng et al. (2025) for reference; they are excluded from shading and Avg. Rank, and Table 13 lists all published results. We highlight the first, second, and third best results per column; RD is ranked by $|\log \text{RD}|$.

	BFS					DFS					RW					
Method	GS↑	RD→1	Deg.↓	Clus.↓	Spec.↓	GS↑	RD→1	Deg.↓	Clus.↓	Spec.↓	GS↑	RD→1	Deg.↓	Clus.↓	Spec.↓	Avg. Rank
SPRINT†	0.278	3.120	13.210	13.411	12.077	0.260	0.946	118.828	21.813	26.641	0.220	0.757	25.001	14.042	14.579	5.80
as published, 0.5	0.227	1.61	10.9	12.5	11.2	0.145	5.91	316	207	53.8	0.178	1.32	17.9	26	15.1	–
TUnA†	0.399	0.569	9.433	8.200	14.741	0.504	0.979	108.909	27.207	24.756	0.534	1.007	16.617	5.409	8.803	3.00
as published, 0.5	0.342	2.99	47.8	30.2	22.9	0.289	4.44	416	272	46.0	0.450	2.23	55.4	42.5	13.7	–
PLM-interact (35M)†	0.432	1.887	10.574	9.690	7.881	0.398	0.608	195.092	56.568	36.346	0.509	1.465	39.320	10.131	10.589	5.00
as published, 0.5	0.383	2.29	24.4	16.0	12.8	0.322	2.85	209	78.2	36.3	0.429	2.47	52.6	22.6	13.6	–
LLP†	0.349	0.369	19.740	17.010	27.630	0.457	1.285	107.407	32.493	19.446	0.559	1.008	23.502	8.702	11.914	4.67
CAZI-MBN†	0.299	1.113	22.830	15.620	27.568	0.280	1.269	255.814	52.530	42.150	0.402	1.981	44.068	18.592	22.069	6.80
L3-PPI‡	0.305	0.840	6.682	6.076	14.030	0.300	1.046	94.436	58.356	14.354	0.461	1.099	15.496	5.551	7.398	3.40
UniPrompt†	0.374	0.876	10.953	9.230	16.937	0.350	1.837	148.014	62.927	21.738	0.453	1.868	25.032	10.070	11.297	5.47
GraSPPI (Ours)	0.413	0.894	6.777	6.024	8.968	0.510	0.942	89.596	19.150	21.161	0.517	1.058	14.712	4.140	6.527	1.87

4.3 Main Results

Network reconstruction. GraSPPI has the best average rank on PRING (1.87, against 3.00 for TUnA and 3.40 for L3-PPI; Table 1), with the lowest clustering discrepancy on every split and the highest DFS graph similarity. Edge overlap and shape order the methods differently: PLM-interact (35M) takes the BFS and LLP the RW graph-similarity cell, the latter with about three times our BFS discrepancies, and L3-PPI, the other graph prompt tuning method, is third overall with lower graph similarity than ours on every split (0.30–0.46). The grey lines show the same models at their published probability-0.5 operating points, where TUnA and PLM-interact (35M) over-predict density 2.2–4.4× and every discrepancy inflates, so our protocol strengthens rather than weakens the baselines.

Pairwise classification. On Bernett (Table 2), GraSPPI has the highest AUROC and AUPRC (both 0.71, AUROC tied with LLP), with accuracy and MCC within 0.01 of the best; its F1 (0.59) trails the PLM predictors and LLP (0.65–0.67). L3-PPI ranks pairs almost as well (AUROC 0.69) but predicts few interactions at probability 0.5 (recall 0.08).

4.4 Function-Oriented Tasks

Table 3 asks whether the reconstructed networks support PRING's complex, functional-module, and essential-protein analyses: GraSPPI ranks third on average, with the best complex-pathway recall (0.743; 0.588 without the set's 325 self-interactions) and GO consistency ratio, but weaker essential-protein separation, low connectivity (0.423), and no within-complex discrimination (AUROC 0.47; Section 6.3.5).

[2]Unmarked baseline rows of Table 2 are published results. † authors' code on our features, training pairs, and selection rules (PLM-interact from its released checkpoint); ‡ reproduced from the paper on our backbone; –: not reported. Details in Sections 6.2.2, 6.3.6 and 6.3.7.

Table 2: Pairwise classification on the Bernett benchmark.[2] Shading and Avg. Rank as in Table 1; a row's Avg. Rank averages only its reported columns.

Method	AUROC↑	AUPRC↑	Acc.↑	F1↑	MCC↑	Avg. Rank
SPRINT	0.52	0.51	0.52	–	–	8.67
Topsy-Turvy	0.59	0.59	0.56	0.37	0.15	7.20
TUnA	0.70	0.69	0.65	0.65	0.30	2.60
PLM-interact	0.70	0.69	–	0.66	–	3.33
C3PI	0.70	0.70	0.65	0.67	0.29	2.00
LLP†	0.71	0.70	0.65	0.67	0.30	1.20
CAZI-MBN†	0.51	0.51	0.50	0.55	0.01	8.60
L3-PPI‡	0.69	0.68	0.53	0.14	0.15	6.80
UniPrompt†	0.65	0.65	0.59	0.64	0.19	5.80
GraSPPI (Ours)	0.71	0.71	0.64	0.59	0.29	3.00

Table 3: Function-oriented tasks on the PRING BFS test network. Published rows are from Zheng et al. (2025); GraSPPI replays its $\mathcal{V}_{\text{val}}$-frozen threshold. Shown: the Table 1 methods with published function results, PRING's three best-ranked function methods, and the best structure-based method; shading and Avg. Rank computed over all rows of Table 12. PR/PP/PC: complex-pathway recall, precision, and connectivity; FA/CR: GO functional alignment and consistency ratio; P@100/DO: essential-protein precision and centrality overlap (Section 6.3.5).

	Complex pathway			GO enrichment		Essential protein		
Method	PR↑	PP↑	PC↑	FA↑	CR↑	P@100↑	DO↓	Avg. Rank
SPRINT	0.526	0.856	0.752	0.174	0.655	0.54	0.727	6.00
PPITrans	0.583	0.863	0.818	0.250	0.756	0.48	0.886	5.14
TUnA	0.526	0.864	0.794	0.259	0.622	0.57	0.849	6.00
PLM-interact (35M)	0.550	0.847	0.880	0.335	0.785	0.80	0.423	2.86
PLM-interact (650M)	0.549	0.862	0.887	0.368	0.844	0.77	0.440	2.57
TAGPPI	0.487	0.781	0.713	0.168	0.649	0.43	0.828	8.14
GraSPPI (Ours)	0.743	0.872	0.423	0.332	0.875	0.67	0.605	3.00

Table 4: Ablations on the PRING BFS split, each removing one module and replaying its own $\mathcal{V}_{\text{val}}$-frozen threshold. Best per column highlighted; RD ranked by $|\log \text{RD}|$.

Variant	GS↑	RD→1	Deg.↓	Clus.↓	Spec.↓
GraSPPI (Ours)	0.413	0.894	6.777	6.024	8.968
without motif graph prompt	0.427	0.600	17.394	13.568	22.870
without $\mathcal{L}_{\text{graph}}$	0.428	0.612	9.548	8.163	13.670
closure template → L3 template	0.357	0.536	7.387	7.442	11.554
without all structural losses	0.313	0.483	12.823	11.894	18.690
without $\mathcal{L}_{\text{rank}}$	0.428	0.614	7.524	6.494	11.813
without $\mathcal{L}_{\text{deg}}$	0.404	0.791	7.444	6.952	10.263
without $\mathcal{L}_{\text{mot}}$	0.409	0.727	8.108	7.499	11.280

4.5 Ablation Study

Table 4 removes GraSPPI's core modules one at a time on the BFS split: every removal worsens RD and all three discrepancies, by 2.3–2.6× without the motif graph prompt and 1.9–2.1× without the structural losses (where graph similarity also falls to 0.313), though three removals trade a small graph-similarity gain (0.427–0.428 against 0.413). Each removal also moves the operating point, so Section 6.3.4 re-thresholds GraSPPI and the variant without structural losses to matched test densities (Fig. 5): the discrepancy gap closes but GraSPPI recovers more edges at every density, and its test-blind threshold generalizes to near the true density (RD 0.894 against 0.483). Removing any one loss or swapping in an L3 template worsens every discrepancy; the inferred motif graph tracks shared neighbours on unseen test proteins ($\rho = 0.41$; Section 6.3.2), and swapping it between test pairs worsens every metric (Section 6.3.3).

5 Conclusion

We propose GraSPPI, a motif graph prompt tuning method that turns the training network into two-level supervision for inductive PPI prediction: a triadic-closure motif graph inferred from the two sequences prompts each decision, and structural losses regularise training so that the assembled network follows the training structure. It leads the compared predictors in average rank, clustering discrepancy, and complex-pathway recall on the three reconstruction splits, though its complexes are less connected than the densest predictors'; ablations show that both levels contribute, with edge recovery improved at matched density, a threshold that transfers to test, and an inferred closure that tracks shared neighbours on test proteins. On Bernett it matches the strongest baselines' pair ranking but trails in F1; the motif graph is coarse and both benchmarks are human, so richer templates and other interactomes are next steps.

AI USE STATEMENT

In this work, we used generative AI tools to assist with implementing and debugging code, literature search and citation checking, and drafting and polishing the manuscript. We have not used generative AI tools to propose or refine hypotheses, design or provide feedback on research methodology or experiments, clean or reformat datasets, or interpret results. The remaining required-disclosure activities—generating synthetic datasets, developing theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, translation, and qualitative or thematic data analysis—are not applicable to this work.

All AI-assisted outputs were reviewed and verified by the authors. Code generated or modified with AI assistance was inspected and validated through our experiments, references and citations were independently checked against the original sources, and AI-assisted manuscript text was reviewed and revised by the authors.

We take responsibility for the final content of this work, including text, claims, code, and other artifacts produced with the aid of generative AI.

REFERENCES

Rachith Aiyappa, Xin Wang, Munjung Kim, Ozgur Can Seckin, Jisung Yoon, Yong-Yeol Ahn, and Sadamori Kojaku. Implicit degree bias in the link prediction task, 2024.

Albert-László Barabási and Zoltán N. Oltvai. Network biology: understanding the cell's functional organization. *Nature Reviews Genetics*, 5(2):101–113, 2004.

Mayank Baranwal, Abram Magner, Jacob Saldinger, Emine S Turali-Emre, Paolo Elvati, Shivani Kozarekar, J Scott VanEpps, Nicholas A Kotov, Angela Violi, and Alfred O Hero. Struct2Graph: a graph attention network for structure based predictions of protein–protein interactions. *BMC Bioinformatics*, 23(1):370, 2022. doi: 10.1186/s12859-022-04910-9.

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.

Christopher J. C. Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Gregory N. Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd International Conference on Machine Learning (ICML)*, pp. 89–96, 2005.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Computer Vision – ECCV 2020*, volume 12346 of *Lecture Notes in Computer Science*, pp. 213–229, 2020.

Benjamin Paul Chamberlain, Sergey Shirobokov, Emanuele Rossi, Fabrizio Frasca, Thomas Markovich, Nils Hammerla, Michael M. Bronstein, and Max Hansmire. Graph neural networks for link prediction with subgraph sketching. In *The Eleventh International Conference on Learning Representations (ICLR 2023)*, 2023.

Sudhanshu Chanpuriya, Cameron Musco, Konstantinos Sotiropoulos, and Charalampos E. Tsourakakis. On the power of edge independent graph models. In *Advances in Neural Information Processing Systems 34*, 2021.

Ayan Chatterjee, Babak Ravandi, Parham Haddadi, Naomi H Philip, Mario Abdelmessih, William R Mowrey, Piero Ricchiuto, Yupu Liang, Wei Ding, Juan Carlos Mobarec, and Tina Eliassi-Rad. Topology-driven negative sampling enhances generalizability in protein–protein interaction prediction. *Bioinformatics*, 41(5):btaf148, April 2025. ISSN 1367-4811.

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, July 2019. ISSN 1367-4811.

Feixiong Cheng, Rishi J. Desai, Diane E. Handy, Ruisheng Wang, Sebastian Schneeweiss, Albert-László Barabási, and Joseph Loscalzo. Network-based approach to prediction and population-based validation of in silico drug repurposing. *Nature Communications*, 9(1):2691, 2018.

Fan Chung and Linyuan Lu. Connected components in random graphs with given expected degree sequences. *Annals of Combinatorics*, 6(2):125–145, 2002. doi: 10.1007/PL00012580.

Alana Deng, Sugitha Janarthanan, Yan Sun, Zihao Jing, and Pingzhao Hu. Distilling and adapting: A topology-aware framework for zero-shot interaction prediction in multiplex biological networks. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 18426–18461, 2026.

Taoran Fang, Yunchao Zhang, Yang Yang, Chunping Wang, and Lei Chen. Universal Prompt Tuning for Graph Neural Networks. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 52464–52489. Curran Associates, Inc., 2023.

Luca Franceschi, Mathias Niepert, Massimiliano Pontil, and Xiao He. Learning discrete structures for graph neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1972–1982. PMLR, June 2019.

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.

Debra S. Goldberg and Frederick P. Roth. Assessing experimentally derived interactions in a small world. *Proceedings of the National Academy of Sciences*, 100(8):4372–4376, 2003.

Yanzhi Guo, Lezheng Yu, Zhining Wen, and Menglong Li. Using support vector machine combined with auto covariance to predict protein–protein interactions from protein sequences. *Nucleic Acids Research*, 36(9):3025–3030, April 2008. ISSN 0305-1048.

Zhichun Guo, William Shiao, Shichang Zhang, Yozen Liu, Nitesh V Chawla, Neil Shah, and Tong Zhao. Linkless Link Prediction via Relational Distillation. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 12012–12033. PMLR, 23–29 Jul 2023.

Somaye Hashemifar, Behnam Neyshabur, Aly A Khan, and Jinbo Xu. Predicting protein–protein interactions through sequence-based deep learning. *Bioinformatics*, 34(17):i802–i810, September 2018. ISSN 1367-4811.

Thomas Hayes, Roshan Rao, Halil Akin, Nicholas J. Sofroniew, Deniz Oktay, Zeming Lin, Robert Verkuil, Vincent Q. Tran, Jonathan Deaton, Marius Wiggert, Rohil Badkundri, Irhum Shafkat, Jun Gong, Alexander Derry, Raul S. Molina, Neil Thomas, Yousuf A. Khan, Chetan Mishra, Carolyn Kim, Liam J. Bartie, Matthew Nemeth, Patrick D. Hsu, Tom Sercu, Salvatore Candido, and Alexander Rives. Simulating 500 million years of evolution with a language model. *Science*, 387(6736):850–858, 2025.

SeyedMohsen Hosseini, G Brian Golding, and Lucian Ilie. Component puzzle protein–protein interaction prediction. *Briefings in Bioinformatics*, 26(6):bbaf685, 2025. doi: 10.1093/bib/bbaf685.

Yongqi Huang, Jitao Zhao, Dongxiao He, Xiaobao Wang, Yawen Li, Yuxiao Huang, Di Jin, and Zhiyong Feng. One Prompt Fits All: Universal Graph Adaptation for Pretrained Models. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 85497–85530. Curran Associates, Inc., 2025.

Peter J. Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, 1964.

Ian R. Humphreys, Jimin Pei, Minkyung Baek, Aditya Krishnakumar, Ivan Anishchenko, Sergey Ovchinnikov, Jing Zhang, Travis J. Ness, Sudeep Banjade, Saket R. Bagde, Viktoriya G. Stancheva, Xiao-Han Li, Kaixian Liu, Zhi Zheng, Daniel J. Barrero, Upasana Roy, Jochen Kuper, Israel S. Fernández, Barnabas Szakal, Dana Branzei, Josep Rizo, Caroline Kisker, Eric C. Greene, Sue Biggins, Scott Keeney, Elizabeth A. Miller, J. Christopher Fromme, Tamara L. Hendrickson, Qian Cong, and David Baker. Computed structures of core eukaryotic protein complexes. *Science*, 374(6573):eabm4805, 2021.

Ian R. Humphreys, Jing Zhang, Minkyung Baek, Yaxi Wang, Aditya Krishnakumar, Jimin Pei, Ivan Anishchenko, Catherine A. Tower, Blake A. Jackson, Thulasi Warrier, Deborah T. Hung, S. Brook Peterson, Joseph D. Mougous, Qian Cong, and David Baker. Protein interactions in human pathogens revealed through deep learning. *Nature Microbiology*, 9(10):2642–2652, 2024. doi: 10.1038/s41564-024-01791-x.

Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *Computer Vision – ECCV 2022*, volume 13693 of *Lecture Notes in Computer Science*, pp. 709–727. Springer Nature Switzerland, 2022. doi: 10.1007/978-3-031-19827-4_41.

Wei Jin, Yao Ma, Xiaorui Liu, Xianfeng Tang, Suhang Wang, and Jiliang Tang. Graph structure learning for robust graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, pp. 66–74. ACM, August 2020.

Thomas N. Kipf and Max Welling. Variational graph auto-encoders, 2016. Bayesian Deep Learning Workshop, NIPS 2016.

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, July 2024. ISSN 1477-4054.

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.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3045–3059. Association for Computational Linguistics, 2021.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4582–4597. Association for Computational Linguistics, 2021.

Yiwei Li and Lucian Ilie. SPRINT: ultrafast protein-protein interaction prediction of the entire human interactome. *BMC Bioinformatics*, 18(1):485, 2017.

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, March 2023. ISSN 1095-9203.

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, October 2025. ISSN 2041-1723.

Katja Luck, Dae-Kyum Kim, Luke Lambourne, Kerstin Spirohn, Bridget E. Begg, Wenting Bian, Ruth Brignall, Tiziana Cafarelli, Francisco J. Campos-Laborie, Benoit Charloteaux, Dongsic Choi, Atina G. Coté, Meaghan Daley, Steven Deimling, Alice Desbuleux, Amélie Dricot, Marinella Gebbia, Madeleine F. Hardy, Nishka Kishore, Jennifer J. Knapp, István A. Kovács, Irma Lemmens, Miles W. Mee, Joseph C. Mellor, Carl Pollis, Carles Pons, Aaron D. Richardson, Sadie Schlabach, Bridget Teeking, Anupama Yadav, Mariana Babor, Dawit Balcha, Omer Basha, Christian Bowman-Colin, Suet-Feung Chin, Soon Gang Choi, Claudia Colabella, Georges Coppin, Cassandra D'Amata, David De Ridder, Steffi De Rouck, Miquel Duran-Frigola, Hanane Ennajdaoui, Florian Goebels, Liana Goehring, Anjali Gopal, Ghazal Haddad, Elodie Hatchi, Mohamed Helmy, Yves Jacob, Yoseph Kassa, Serena Landini, Roujia Li, Natascha van Lieshout, Andrew MacWilliams, Dylan Markey, Joseph N. Paulson, Sudharshan Rangarajan, John Rasla, Ashyad Rayhan, Thomas Rolland, Adriana San-Miguel, Yun Shen, Dayag Sheykhkarimli, Gloria M. Sheynkman, Eyal Simonovsky, Murat Taşan, Alexander Tejeda, Vincent Tropepe, Jean-Claude Twizere, Yang Wang, Robert J. Weatheritt, Jochen Weile, Yu Xia, Xinping Yang, Esti Yeger-Lotem, Quan Zhong, Patrick Aloy, Gary D. Bader, Javier De Las Rivas, Suzanne Gaudet, Tong Hao, Janusz Rak, Jan Tavernier, David E. Hill, Marc Vidal, Frederick P. Roth, and Michael A. Calderwood. A reference map of the human binary protein interactome. *Nature*, 580(7803): 402–408, 2020.

Liheng Ma, Chen Lin, Derek Lim, Adriana Romero-Soriano, Puneet K. Dokania, Mark Coates, Philip Torr, and Ser-Nam Lim. Graph inductive biases in transformers without message passing. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 23321–23337. PMLR, 2023.

Jörg Menche, Amitabh Sharma, Maksim Kitsak, Susan Dina Ghiassian, Marc Vidal, Joseph Loscalzo, and Albert-László Barabási. Uncovering disease-disease relationships through the incomplete interactome. *Science*, 347(6224):1257601, 2015.

Leslie O'Bray, Max Horn, Bastian Rieck, and Karsten Borgwardt. Evaluation metrics for graph generative models: Problems, pitfalls, and practical solutions. In *International Conference on Learning Representations*, 2022.

Yungki Park and Edward M Marcotte. Flaws in evaluation schemes for pair-input computational predictions. *Nature Methods*, 9(12):1134–1136, 2012.

Gabriel Peyré and Marco Cuturi. Computational optimal transport: With applications to data science. *Foundations and Trends in Machine Learning*, 11(5–6):355–607, 2019. doi: 10.1561/2200000073.

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 452–461, 2009.

Florian Richoux, Charlène Servantie, Cynthia Borès, and Stéphane Téletchéa. Comparing two deep learning sequence-based models for protein-protein interaction prediction. arXiv preprint arXiv:1901.06268, 2019.

João J. Rocha, Satish Arcot Jayaram, Tim J. Stevens, Nadine Muschalik, Rajen D. Shah, Sahar Emran, Cristina Robles, Matthew Freeman, and Sean Munro. Functional unknomics: Systematic screening of conserved genes of unknown function. *PLOS Biology*, 21(8):e3002222, 2023.

Ahmed E. Samy, Zekarias T. Kefato, and Sarunas Girdzijauskas. Graph2Feat: Inductive link prediction via knowledge distillation. In *Companion Proceedings of the ACM Web Conference 2023*, pp. 805–812, 2023.

C. Seshadhri, Aneesh Sharma, Andrew Stolman, and Ashish Goel. The impossibility of low-rank representations for triangle-rich complex networks. *Proceedings of the National Academy of Sciences*, 117(11):5631–5637, March 2020. ISSN 1091-6490.

Juwen Shen, Jian Zhang, Xiaomin Luo, Weiliang Zhu, Kunqian Yu, Kaixian Chen, Yixue Li, and Hualiang Jiang. Predicting protein–protein interactions based only on sequences information. *Proceedings of the National Academy of Sciences*, 104(11):4337–4341, March 2007. ISSN 1091-6490.

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, June 2022. ISSN 1367-4811.

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, October 2021. ISSN 2405-4712.

Samuel Sledzieski, Meghana Kshirsagar, Minkyung Baek, Rahul Dodhia, Juan Lavista Ferres, and Bonnie Berger. Democratizing protein language models with parameter-efficient fine-tuning. *Proceedings of the National Academy of Sciences*, 121(26):e2405840121, 2024. doi: 10.1073/pnas.2405840121.

Bosheng Song, Xiaoyan Luo, Xiaoli Luo, Yuansheng Liu, Zhangming Niu, and Xiangxiang Zeng. Learning spatial structures of proteins improves protein–protein interaction prediction. *Briefings in Bioinformatics*, 23(2):bbab558, 2022. doi: 10.1093/bib/bbab558.

Victor Spirin and Leonid A. Mirny. Protein complexes and functional modules in molecular networks. *Proceedings of the National Academy of Sciences*, 100(21):12123–12128, 2003.

Mingchen Sun, Kaixiong Zhou, Xin He, Ying Wang, and Xin Wang. GPPT: Graph Pre-training and Prompt Tuning to Generalize Graph Neural Networks. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, pp. 1717–1727. ACM, August 2022.

Xiangguo Sun, Hong Cheng, Jia Li, Bo Liu, and Jihong Guan. All in One: Multi-Task Prompting for Graph Neural Networks. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, pp. 2120–2131. ACM, August 2023.

Damian Szklarczyk, Rebecca Kirsch, Mikaela Koutrouli, Katerina Nastou, Farrokh Mehryary, Radja Hachilif, Annika L. Gable, Tao Fang, Nadezhda T. Doncheva, Sampo Pyysalo, Peer Bork, Lars J. Jensen, and Christian von Mering. The STRING database in 2023: protein–protein association networks and functional enrichment analyses for any sequenced genome of interest. *Nucleic Acids Research*, 51(D1):D638–D646, 2023.

Zhen Tan, Ruocheng Guo, Kaize Ding, and Huan Liu. Virtual node tuning for few-shot node classification. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2177–2188. ACM, 2023. doi: 10.1145/3580305.3599541.

Varun Ullanat, Bowen Jing, Samuel Sledzieski, and Bonnie Berger. Learning the language of protein-protein interactions. *Nature Communications*, 17(1):1199, 2026.

Marc Vidal, Michael E. Cusick, and Albert-László Barabási. Interactome networks and human disease. *Cell*, 144(6):986–998, 2011.

Clement Vignac, Igor Krawczuk, Antoine Siraudin, Bohan Wang, Volkan Cevher, and Pascal Frossard. DiGress: Discrete denoising diffusion for graph generation. In *International Conference on Learning Representations*, 2023.

Xiyuan Wang, Haotong Yang, and Muhan Zhang. Neural common neighbor with completion for link prediction. In *The Twelfth International Conference on Learning Representations (ICLR 2024)*, 2024.

Zhitao Wang, Yong Zhou, Litao Hong, Yuanhang Zou, Hanjing Su, and Shouzhi Chen. Pairwise learning for neural link prediction. arXiv preprint arXiv:2112.02936, 2021.

Sen Yang, Peng Cheng, Yang Liu, Dawei Feng, and Shengqi Wang. Exploring the knowledge of an outstanding protein to protein interaction transformer. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 21(5):1287–1298, 2024. doi: 10.1109/TCBB.2024.3381825.

Yu Yao, Xiuquan Du, Yanyu Diao, and Huaixu Zhu. An integration of deep learning with feature embedding for protein–protein interaction prediction. *PeerJ*, 7:e7126, 2019. doi: 10.7717/peerj.7126.

Jiaxuan You, Rex Ying, Xiang Ren, William Hamilton, and Jure Leskovec. GraphRNN: Generating realistic graphs with deep auto-regressive models. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5708–5717. PMLR, July 2018.

Ho Yin Yuen and Jesper Jansson. Normalized L3-based link prediction in protein–protein interaction networks. *BMC Bioinformatics*, 24(1):59, 2023.

Serhan Yılmaz, Kaan Yorgancioglu, and Mehmet Koyutürk. Bias-aware training and evaluation of link prediction algorithms in network biology. *Proceedings of the National Academy of Sciences*, 122(24):e2416646122, 2025.

Muhan Zhang and Yixin Chen. Link Prediction Based on Graph Neural Networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Shichang Zhang, Yozen Liu, Yizhou Sun, and Neil Shah. Graph-less neural networks: Teaching old MLPs new tricks via distillation. In *The Tenth International Conference on Learning Representations (ICLR 2022)*, 2022.

Wenqing Zheng, Edward W Huang, Nikhil Rao, Sumeet Katariya, Zhangyang Wang, and Karthik Subbian. Cold brew: Distilling graph node representations with incomplete or missing neighborhoods. In *The Tenth International Conference on Learning Representations (ICLR 2022)*, 2022.

Xinzhe Zheng, Hao Du, Fanding Xu, Jinzhe Li, Zhiyuan Liu, Wenkang Wang, Tao Chen, Wanli Ouyang, Stan Z. Li, Yan Lu, Nanqing Dong, and Yang Zhang. PRING: Rethinking Protein-Protein Interaction Prediction from Pairs to Graphs. In *Advances in Neural Information Processing Systems 38, Datasets and Benchmarks Track*, NeurIPS 2025, pp. 103882–103926. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2025.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16795–16804, 2022a. doi: 10.1109/CVPR52688.2022.01631.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022b. doi: 10.1007/s11263-022-01653-1.

Tao Zhou, Linyuan Lü, and Yi-Cheng Zhang. Predicting missing links via local information. *The European Physical Journal B*, 71(4):623–630, October 2009. ISSN 1434-6036.

6 APPENDIX

This appendix complements the main text in three parts. Section 6.1 gives more method details: software and hardware, features and precision, the architecture, training, hyperparameters, and the Bernett protocol. Section 6.2 details the experimental setup: the two benchmarks, the baseline reproduction, the evaluation metrics, and the validation region with its model-selection rules. Section 6.3 reports more experimental results: network pattern statistics, the generalization of inferred motif graphs, test-time prompt interventions, the function-oriented tasks, and all published results on PRING and on the Bernett benchmark.

6.1 MORE METHOD DETAILS

6.1.1 SOFTWARE AND HARDWARE

We implement all models in PyTorch 2.10 (CUDA 12.8) with PyTorch Geometric 2.8, use the official GRIT implementation for the reader (Ma et al., 2023), and train with distributed data parallelism through Accelerate 1.13. Every run of our method uses four NVIDIA H20 GPUs (96 GB each). Stage I trains in 3.8 h and Stage II in 3.2 h of wall-clock time, and held-out scoring of one checkpoint takes about 26 min.

6.1.2 FEATURES AND PRECISION

Each protein is represented by frozen per-residue ESM3 embeddings (`esm3-sm-open-v1`, 1,536 dimensions, at most 1,024 tokens) (Hayes et al., 2025), stored in bfloat16 and standardised with training-protein statistics. Training and scoring use bfloat16 autocast, except that the prompt edge weights and presence probability of Eq. (3) and the random-walk features of the reader run in float32. Micro-batches are packed to at most 196,608 residue tokens and 1,536 pairs per GPU; the structural stream is scored in chunks of 24,576 tokens with activation recomputation, so every legal pair of a sampled subgraph is still scored exactly once.

6.1.3 ARCHITECTURE

Pair predictor. The residue encoder projects the 1,536-dimensional features to width $d = 512$ and applies $L_e = 3$ transformer layers; the classifier has $L_c = 3$ bidirectional cross-attention layers; the sequence trunk uses $n_h = 8$ attention heads (the reader uses four). The pair readout pools each endpoint's residue states by mean, by maximum, and by attention conditioned on the partner's mean and max pools, where each token's attention score is a two-layer MLP of the token state and the partner summary; each of the three branches forms $[a, b, |a-b|, a\odot b] \in \mathbb{R}^{4d}$ from the pooled vectors a and b of the two endpoints, a softmax gate over the three concatenated branches mixes them, and the mix is projected and concatenated with the classification token into $z_{uv} \in \mathbb{R}^d$. The prediction head is an MLP with hidden widths 512, 256, and 128, GELU activations, layer normalisation, and dropout 0.2. The trunk uses dropout, token dropout, cross-attention dropout, and stochastic depth of 0.1 each.

Motif template and true motif graphs. We use $k = 8$ closure slots, so a motif graph has $k+2 = 10$ nodes and $2k = 16$ candidate edges. Closure slots take up to k shared neighbours in increasing order of degree, with ties broken by a swap-invariant hash of the neighbour and the queried pair, where degrees are measured with the queried edge removed. Both edges of a filled slot carry the inverse square root of the slotted neighbour's degree. The compiler reads loopless graphs, so self-interactions enter no neighbourhood set or degree. Task-stream templates are compiled once with a fixed random slot order per pair; structural-stream templates keep the compiled order. Self-pairs receive an empty graph and no motif-graph supervision; prompt inference still predicts a graph for them.

Prompt inference. Queries and slot states have width $d_g = 96$; each read attends over the residue states projected to width d_g and excludes padding and the sequence boundary tokens. Writing Read for residual cross-attention, each closure slot merges its two endpoint reads $w_{i,c} = \mathrm{Read}(q_c, H_i)$ symmetrically,

$$h_{C_c} = \mathrm{MLP}\big([\, w_{u,c} + w_{v,c},\ |w_{u,c} - w_{v,c}|\,]\big). \tag{11}$$

With $\tilde{A}$ the row-normalised, unweighted incidence of the sixteen candidate edges (no self-loops), each of two message-passing layers, without shared parameters, updates $h \leftarrow h + W_2\,\mathrm{GELU}\big(W_1\,\mathrm{LN}(\tilde{A}\,W_{\mathrm{rel}}h) + b_1\big) + b_2$, with W_{rel} shared across nodes within a layer. From the symmetric edge feature $e_{ij} \in \mathbb{R}^{2d_g}$ of Section 3.1, an LN–Lin–GELU–Lin head with hidden width d_g predicts one logit per candidate edge; the presence head has the same architecture with separate parameters and reads the mean edge feature over the template. The edge-head bias starts at the mean weight of a filled closure edge in the training templates and the presence-head bias at the rate of pairs with a shared neighbour, both clipped to $[0.01, 0.99]$ before the logit.

Graph reader and prefixes. The GRIT reader has three layers of width 96 with four heads and attends over all ordered node pairs of the ten-node graph, including the absent queried edge. Node inputs concatenate a role embedding (endpoint or closure slot) with the diagonal of the relative random-walk features, and pair inputs concatenate those features with the edge weight; the weighted log-degree enters GRIT's degree scaling. The relative random-walk features are $[I, T, T^2, T^3]$ ($K_{\mathrm{rw}} = 4$) with $T_{ij} = B_{ij}/\max(D_{ii}, 10^{-6})$ and $D_{ii} = \sum_j B_{ij}$, so an isolated node has zero rows in every channel but the identity. With reader node states h_i^R, directed pair states h_{ij}^R, learned projections W_n, W_e, and layer normalisation LN, the tokens are $t_u = \mathrm{LN}(W_n h_u^R)$, $t_v = \mathrm{LN}(W_n h_v^R)$, and $t_{\mathrm{rel}} = \mathrm{LN}\big(W_e(h_{uv}^R + h_{vu}^R)\big)$, each of width 128. For each layer, a linear map expands each field into $n_p = 2$ prefix rows of model width 512, plus learned row offsets, which gives six rows at each of the nine cross-attention sites (three layers, three sites each). Each site passes the rows through its own key, value, and output projections, split into eight heads of width 64; the classification-token site reads the view of the endpoint that comes first in the current order. Table 5 lists parameter counts.

Table 5: Parameter counts of the deployed model.

Component	Parameters
Sequence trunk and prediction head	29.09M
Prompt inference P_ϕ	0.26M
GRIT reader R_{θ_r}	0.34M
Prefix adapter and gates Π_{θ_p}	0.41M
Total	30.10M

6.1.4 Training

Losses. The task stream redraws a 1:5 positive-to-negative sample every epoch. Each negative is proposed, with probability one half each, uniformly (including self-pairs at the training universe's self-pair rate) or by replacing one endpoint of a training edge with a degree-proportional draw; proposals that are positives or duplicates are rejected. In both streams, $\mathcal{L}_{\mathrm{BCE}}^{\mathrm{s}}$ weights positives by $w_+ = 5$ and negatives by one and divides by the sum of the weights. Structural subgraphs have 40 proteins, 32 local and eight background, drawn as a BFS ball, a wedge ball, or a two-ball subgraph with probabilities 0.5, 0.25, and 0.25. A BFS ball grows by breadth-first search from a random protein with at least one partner; a wedge ball grows from an open wedge or a closed triangle, chosen with probability one half each and falling back to a wedge after 32 failed triangle draws; a two-ball subgraph grows two balls of 16 proteins from two roots, drawn up to 32 times to find roots at distance at least three. Balls that stop short are filled from new roots. Each ball grows by first-in-first-out breadth-first search from its seeds, visiting every popped protein's neighbours in seeded random order; wedge seeds accept a centre of degree at least two with two non-adjacent neighbours within 32 tries, triangle seeds draw a training edge and a uniform common neighbour, and the two balls of a two-ball subgraph stay disjoint. One subgraph is drawn per optimizer step from a step-seeded generator and shared by every GPU: its pairs are striped across ranks and the logits all-reduced, so all ranks minimise the same loss. Subgraphs are drawn from the training graph alone, so they contain no validation protein and no protein without features; the legal pairs are all $\binom{40}{2}$ unordered non-self pairs, each scored once, mirrored into the symmetric Z, and labelled by the induced training adjacency. The two streams share their positives, since every edge of a subgraph is a training edge that the task stream also visits once per epoch, but not their negatives: the subgraph negatives are non-edges near or between sampled neighbourhoods, which the task stream draws only by chance, and self-pairs appear only in the task stream. Stage I blends the graphs of non-self rows with probability p_b, replacing B_{uv}^* by $(1-\lambda)B_{uv}^* + \lambda\bar{B}$ with $\lambda \sim \mathcal{U}[0,1]$ and $\bar{B}$ the mean

motif graph of the first epoch's task rows, and applies prompt dropout with probability p_d, using one mask for all sites and both endpoint orders. The structural losses use margin $m = 0.1$ and Huber threshold $\delta = 1$; empty reductions contribute zero. In $\mathcal{L}_{\text{graph}}$, ℓ_{slot} is implemented as the Huber loss at $\delta = 1$, which is quadratic because path products lie in $[0, 1]$; the presence probability is clipped to $[10^{-6}, 1 - 10^{-6}]$, and the row counts are summed over all GPUs at each step.

Optimisation. Both stages train for 8 epochs with AdamW, gradient clipping at one, and a one-cycle schedule with 10% warm-up, initial and final learning rates of $1/25$ and $1/(25 \times 10^4)$ of the maximum, and cosine annealing, without early stopping. Weight decay is 0.05 in Stage I and 0.01 in Stage II. The schedule length counts the task-stream batches of the eight epochs; the structural stream adds no optimizer steps. Stage I trains every parameter group at a maximum learning rate of 10^{-4}; Stage II gives its three groups separate maxima on one shared schedule – 5×10^{-5} for prompt inference, 10^{-5} for the reader and adapter, and 5×10^{-6} for the trunk: phase A (the first E_w epochs) trains only prompt inference, and in phase B the reader, adapter, and trunk join the schedule where it stands, without a new warm-up, with prompt dropout off. All locally trained models use seed 0.

Ablation variants. Each row of Table 4 is a separately trained chain. *Without motif graph prompt* trains the sequence trunk on the structural subgraph stream alone – subgraph cross-entropy plus the rank, degree, and motif losses, for 25 epochs in one stage – with no motif prompt, no motif-graph supervision, and no 1:5 task-pair stream; in Table 4 it trades network shape for edge overlap, a higher edge-set graph similarity (0.427 against 0.413) at a worse relative density (0.600 against 0.894) and 2.3–2.6× higher MMD ratios. *Without all structural losses* removes the structural stream entirely, subgraph cross-entropy included, while keeping the prompt and its graph supervision on task rows; the single-loss rows set one weight λ to zero. *Without* $\mathcal{L}_{\text{graph}}$ sets the slot weight to zero in Stage II: under the stop-gradients of Eq. (10) no gradient then reaches prompt inference, so this row reads an inferred graph frozen at its initialisation, whose edge and presence biases sit at the training base rates, and measures the trained prompt against that near-constant one.

6.1.5 HYPERPARAMETERS

We search the hyperparameters of Section 3 over the values in Table 6 and choose the best on the validation protein set $\mathcal{V}_{\text{val}}$ (Section 6.2.4). On the Bernett benchmark the same values are kept, except $w_+ = 1$: training reads the release's fixed 1:1 Intra-1 pairs, the structural stream samples Intra-1 subgraphs, and checkpoints are selected by Intra-0 AUPRC. We also try four settings of the Stage II peak learning rates, written as (prompt inference; reader and adapter, with the trunk at half the latter): $(5 \times 10^{-5}; 1.0 \times 10^{-5})$, the PRING setting, $(4.7 \times 10^{-5}; 2.1 \times 10^{-5})$, $(1.3 \times 10^{-4}; 1.5 \times 10^{-5})$, and $(2.0 \times 10^{-5}; 5.2 \times 10^{-6})$, and keep the PRING setting, which has the highest Intra-0 AUPRC.

Table 6: Hyperparameters of our method; braces list the values searched, bold the value chosen on $\mathcal{V}_{\text{val}}$.

Hyperparameter	Symbol	Values
Model width	d	{256, **512**, 768}
Residue-encoder layers	L_e	3
Sequence-trunk attention heads	n_h	8
Slot state dimension	d_g	96
Cross-attention layers	L_c	{2, **3**}
Closure slots	k	8
Prefix rows per field	n_p	2
Template blending probability	p_b	0.5
Prompt dropout	p_d	0.25
Positive weight	w_+	{1, **5**}
Ranking weight	λ_{rank}	{0.1, 0.3, **1**, 3}
Ranking margin	m	0.1
Degree weight	λ_{deg}	{0.03, **0.1**, 0.3, 1}
Motif weight	λ_{mot}	{0.03, **0.1**, 0.3, 1}
Huber threshold	δ	1
Presence weight	λ_π	1
Warm-up epochs	E_w	{2, **4**, 8}

6.2 More Experimental Setup

6.2.1 Benchmarks

We evaluate on two complementary human PPI benchmarks. PRING (Zheng et al., 2025) targets network reconstruction: predictions are assembled over subgraphs that breadth-first, depth-first and random-walk traversals sample from held-out proteins, and we use it for all topology metrics. For pairwise classification we additionally use the leakage-free gold standard of Bernett et al. (2024), whose leakage analysis motivates PRING's protein-disjoint splits. Its proteins are partitioned with KaHIP so that no protein occurs in more than one split and sequence similarity across splits is minimized, and CD-HIT at 40% identity further reduces redundancy. The benchmark is balanced and has been adopted by recent sequence-based predictors (Ko et al., 2024; Liu et al., 2025). Because our validation protocol and hyperparameters were developed on PRING and, apart from the Stage II learning rates, are kept fixed on the gold standard, it tests whether edge-level gains transfer to an independently constructed split rather than to the split used for model development. The gold standard is built from HIPPIE human interactions and PRING from UniProt, Reactome, IntAct and STRING, so the second benchmark changes the interaction source as well as the split construction and the development set.

6.2.2 Baseline Reproduction

Rows marked † have no published result on the benchmark in question, so we reproduce them with the authors' official code and three adaptations. First, every method reads the same frozen ESM3 residue embeddings (Section 6.1) in place of its original input features: TUnA its ESM-2 embeddings, and the graph methods LLP, CAZI-MBN, and UniPrompt their node features, for which they receive mean-pooled ESM3 embeddings; SPRINT, which compares sequences directly, keeps its own similarity search over the training sequences. Second, all train on our training data: on PRING the training graph with $\mathcal{V}_{\mathrm{val}}$ removed, and on Bernett the release's Intra-1 pairs. Third, checkpoint and threshold follow our selection rules (Section 6.2.4): on PRING, the five-metric checkpoint rank and one topology threshold frozen on $\mathcal{V}_{\mathrm{val}}$; on Bernett, Intra-0 AUPRC selection and probability 0.5, except for SPRINT, whose scores are not probabilities (Section 6.2.3). PLM-interact (35M) is not retrained: we score the authors' released HumanV11 checkpoint and apply the same threshold rules. L3-PPI (‡) is a model-agnostic plug-in without official code, so we reproduce it from the paper on our own sequence encoder backbone, the pair predictor of Section 3.1 without prompts, and replace its prediction head with the L3 graph head; it therefore shares our backbone architecture and features, and follows the same selection rules.

6.2.3 Evaluation Metrics

Assembled subgraphs. Each PRING split provides sampled test node sets $S_{n,i}$ for sizes $n \in \mathcal{N} = \{20, 40, \ldots, 200\}$ and $i = 1, \ldots, 50$. Let $E_{n,i}$ be the true interactions among $S_{n,i}$, including self-interactions. Given scores $\hat{p}_{uv}$ and a threshold τ, the predicted edge set is $\hat{E}_{n,i} = \{\{u, v\} \subseteq S_{n,i} : \hat{p}_{uv} \geq \tau\}$, where $u = v$ is allowed. The per-subgraph scores are

$$\mathrm{GS}_{n,i} = \frac{2\,|E_{n,i} \cap \hat{E}_{n,i}|}{|E_{n,i}| + |\hat{E}_{n,i}|}, \qquad \mathrm{RD}_{n,i} = \frac{|\hat{E}_{n,i}|}{|E_{n,i}|}, \tag{12}$$

with $\mathrm{GS}_{n,i} = 1$ when both sets are empty; RD is the ratio of predicted to true edge density, because both graphs share the node set. GS equals $1 - |E_{n,i} \triangle \hat{E}_{n,i}|/(|E_{n,i}| + |\hat{E}_{n,i}|)$, the edge-set reading of PRING's graph similarity. Reported values are macro-averages over sizes, $\mathrm{GS} = |\mathcal{N}|^{-1} \sum_n \frac{1}{50} \sum_i \mathrm{GS}_{n,i}$, and likewise for RD.

MMD ratios. For each subgraph we compute three descriptors, each normalised to unit mass: the degree histogram, the histogram of local clustering coefficients over 100 bins on $[0, 1]$, and the histogram of normalised-Laplacian eigenvalues over 200 bins on $[0, 2]$, following the MMD evaluation of graph generative models (You et al., 2018; O'Bray et al., 2022; Vignac et al., 2023) that PRING adopts (Zheng et al., 2025). Descriptors h, h' are compared with the kernel $k(h, h') = \exp\big(-\mathrm{TV}(h, h')^2/2\sigma^2\big)$, where $\mathrm{TV}(h, h') = \frac{1}{2}\|h - h'\|_1$ and $\sigma = 1$. For descriptor sets A and B,

the biased squared MMD is

$$\mathrm{MMD}^2(A,B) = \frac{1}{|A|^2}\sum_{a,a'\in A} k(a,a') + \frac{1}{|B|^2}\sum_{b,b'\in B} k(b,b') - \frac{2}{|A||B|}\sum_{a\in A,\, b\in B} k(a,b). \tag{13}$$

Let $\hat{H}_n$ and H_n be the descriptors of the predicted and true subgraphs of size n, and H_n^{odd}, H_n^{even} the true descriptors with odd and even sample index. The reported ratio for each descriptor is

$$\text{MMD ratio} = \frac{\sum_{n\in\mathcal{N}} \mathrm{MMD}^2(\hat{H}_n, H_n)}{\sum_{n\in\mathcal{N}} \mathrm{MMD}^2(H_n^{\mathrm{odd}}, H_n^{\mathrm{even}})}. \tag{14}$$

The denominator is the discrepancy between two halves of the reference sample, so a ratio of one means the predicted subgraphs are as far from the true ones as true subgraphs are from each other.

Classification metrics. AUROC is the area under the ROC curve and AUPRC the average precision, both on raw scores. The classification threshold is probability 0.5, except for SPRINT, whose similarity scores are not probabilities; its threshold is the maximum-F1 point on the validation split (Intra-0). With true and false positives and negatives $\mathrm{TP}, \mathrm{FP}, \mathrm{TN}, \mathrm{FN}$ at that threshold, accuracy is $(\mathrm{TP}+\mathrm{TN})/(\mathrm{TP}+\mathrm{FP}+\mathrm{TN}+\mathrm{FN})$, F1 is $2\mathrm{TP}/(2\mathrm{TP}+\mathrm{FP}+\mathrm{FN})$, and

$$\mathrm{MCC} = \frac{\mathrm{TP}\cdot\mathrm{TN} - \mathrm{FP}\cdot\mathrm{FN}}{\sqrt{(\mathrm{TP}+\mathrm{FP})(\mathrm{TP}+\mathrm{FN})(\mathrm{TN}+\mathrm{FP})(\mathrm{TN}+\mathrm{FN})}}. \tag{15}$$

6.2.4 Validation Region and Model Selection

Validation region. PRING's training side forms one substrate graph of 8,072 proteins. For each split we draw a root uniformly, with seed 42, among substrate proteins that have exactly five interaction partners and lie in a component of at least 200 proteins. From that root, the split's own traversal (BFS, DFS, or random walk) admits proteins into $\mathcal{V}_{\mathrm{val}}$ until the next protein would exceed 10% of the substrate's positive pairs, counting self-interactions once. $\mathcal{V}_{\mathrm{val}}$ then leaves the training set: every pair with an endpoint in it, whether internal or crossing the boundary and whether positive or negative, is removed from training, mirroring the train/test boundary. Validation subgraphs are sampled inside $\mathcal{V}_{\mathrm{val}}$ with the same traversal, 50 per size for sizes $20, 40, \ldots, 200$. Table 7 lists the resulting sizes.

Table 7: Validation regions and effective training graphs of the three PRING splits. Positive counts include self-interactions.

Split	$\lvert\mathcal{V}_{\mathrm{val}}\rvert$	Validation positives	Training proteins	Training positives
BFS	869	5,347	7,203	36,857
DFS	1,109	7,086	6,963	39,581
RW	887	4,408	7,185	27,692

Threshold and checkpoint selection. For each checkpoint we scan the distinct score boundaries on the union of validation-subgraph pairs and keep

$$\tau^\star = \arg\min_{\tau} \Big| \sum_{n\in\mathcal{N}} \sum_{i=1}^{50} \log \mathrm{RD}_{n,i}(\tau) \Big|, \tag{16}$$

the threshold whose geometric-mean relative density over validation subgraphs is closest to one in log space (Section 6.2.3); the deployed threshold τ of Eq. (1) is $\tau^\star$. The candidates are every distinct score on that pair union plus one boundary above the maximum; a threshold that empties any subgraph makes the objective infinite and is never selected, and exact ties break by macro GS, then by the geometric mean of the three MMD ratios, then by the higher boundary. That threshold is frozen and replayed unchanged on every test subgraph. We do not binarise at probability 0.5, which would conflate network topology with score calibration: methods whose scores are shifted or scaled would then predict networks of very different density. Validation AUPRC is computed on a separate classification set: every $\mathcal{V}_{\mathrm{val}}$-induced positive, self-pairs included, plus an equal number of endpoint-frequency negatives drawn once without replacement. Checkpoints are ranked by the mean of their ranks on validation AUPRC and on GS and the three MMD ratios, the last four computed

at the checkpoint's own threshold; ties are broken by GS, then by the geometric mean of the MMD ratios, then by the earlier epoch. RD is reported but not ranked, since the threshold already targets it. Our method scores validation classification every epoch and the validation topology on even epochs and the last, and ranks only the epochs with a topology pass. On the Bernett benchmark, checkpoints are selected by Intra-0 AUPRC alone. No selection step reads test proteins.

6.3 More Experimental Results

6.3.1 Network Pattern Statistics

These statistics support Section 4.2. Each split's training graph excludes $\mathcal{V}_{\text{val}}$ and has 31,623 (BFS), 34,378 (DFS), and 22,372 (RW) interactions without self-loops. Every interacting pair is measured with its own edge removed. Degree bins are $\lfloor \log_2(d+1) \rfloor$, and each interacting pair is matched with one non-interacting pair whose two proteins fall in the bins of its two proteins. A pair is closed when it has a shared neighbour; its L3 paths are the edges joining a neighbour of one protein to a neighbour of the other (Kovács et al., 2019). Beyond the shared-neighbour count, we report resource allocation $\sum_w 1/d_w$ over shared neighbours w (Zhou et al., 2009), which discounts neighbours that are hubs. Table 8 splits pairs into thirds by the smaller endpoint degree and gives the values plotted in Fig. 3b; Table 9 compares each graph with ten degree-preserving rewirings ($10|E|$ double edge swaps). Read alone, the L3 count ranks interacting against non-interacting pairs at AUROC 0.83 (BFS), 0.75 (DFS), and 0.82 (RW). To measure each count given the other, both are binned (shared neighbours 0, 1, 2–3, 4–7, 8+; L3 paths 0, 1–3, 4–15, 16–63, 64+); the AUROC of one count is computed within each bin of the other, with ties counted half, and averaged over the bins weighted by their number of pairs, skipping bins that hold only one class. With the shared-neighbour count held fixed, the L3 count reaches 0.44–0.53; with the L3 count held fixed, the shared-neighbour count reaches 0.66–0.73. In the degree-preserving rewirings, 39–54% of edges are reached only by L3 paths and 7–38% are closed (Table 9).

Table 8: AUROC of closure scores against degree-matched non-interacting pairs, overall and within thirds of the smaller endpoint degree (ranges in brackets). The degree product sees only degrees and sits near chance by construction.

Split	Score	All	Low	Mid	High
BFS	Shared neighbours	0.87	0.75	0.93	0.98
	Resource allocation	0.88	0.75	0.93	0.99
	Degree product	0.51	0.51	0.53	0.55
	Min. degree		[0–7]	[8–24]	[25–120]
DFS	Shared neighbours	0.80	0.75	0.90	0.91
	Resource allocation	0.83	0.75	0.91	0.95
	Degree product	0.51	0.51	0.54	0.56
	Min. degree		[0–8]	[9–27]	[28–158]
RW	Shared neighbours	0.81	0.69	0.85	0.92
	Resource allocation	0.81	0.69	0.85	0.92
	Degree product	0.52	0.50	0.54	0.55
	Min. degree		[0–4]	[5–10]	[11–85]

Table 9: Each training graph against ten degree-preserving rewirings: triangle count and transitivity relative to the rewiring mean, and the share of edges that are closed or reached only by L3 paths, each with its own edge removed.

	BFS	DFS	RW
Triangles	32.7×	19.4×	51.2×
Transitivity	32.7×	19.4×	51.2×
Closed edges, real	79%	77%	65%
Closed edges, rewired	29%	38%	7%
L3-only edges, real	7%	11%	14%
L3-only edges, rewired	54%	50%	39%

6.3.2 Motif Graph Generalization

No true graph exists at test time, so we ask whether the closure mass of an inferred motif graph tracks the number of shared neighbours it stands for (Fig. 4), following Gao et al. (2026), with

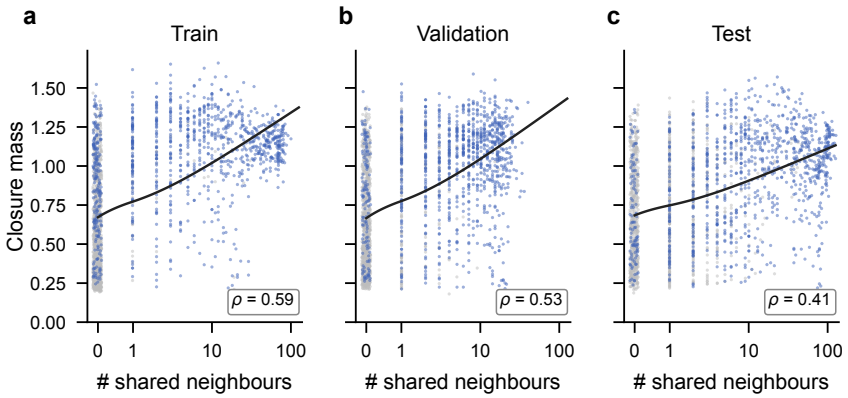

Figure 4: Closure mass (the sum of the first endpoint's eight inferred closure edge weights) against the number of shared neighbours on PRING BFS, for balanced (a) training, (b) validation, and (c) test pairs with the queried edge removed; blue: interacting, grey: non-interacting; ρ: Spearman's correlation.

the BFS checkpoint of Table 1. The correlation holds on test proteins never seen in training ($\rho = 0.41$, against 0.59 on training pairs and 0.53 on validation pairs), and its moderate size marks the inferred graph as a coarse summary of the real neighbourhood. The three universes of Fig. 4 hold 63,246 training pairs, 9,406 validation pairs, and 60,256 test pairs, all non-self and each balanced by subsampling the larger class; the plotted dots are 1,000 pairs per class, while ρ is taken over every pair. Slots of the template are exchangeable, so only the permutation-invariant closure mass is compared: the sum of the first endpoint's eight closure edge weights, presence-scaled as the reader receives them; it summarises, not equals, the sorted path products that $\mathcal{L}_{\text{graph}}$ supervises. Table 10 evaluates the supervised quantity itself on the same balanced pairs: the summed closure path product $\sum_c \widehat{B}_{uC_c}\widehat{B}_{C_cv}$ – the quantity whose sorted profile Eq. (9) matches, summed so that slot order does not matter – is compared by Spearman's ρ with that of the compiled true graph B^*_{uv}, and the presence probability π_{uv} is scored by AUROC against the true indicator a_{uv}. The supervised quantity transfers to unseen proteins and holds within both classes, weaker but positive among non-interacting pairs, and degree does not explain it: the partial ρ, the Spearman correlation of the rank residuals after regressing out both endpoints' log-degree ranks, moves the test correlation only from 0.400 to 0.384, and the presence AUROC averaged within endpoint-degree quintiles stays at 0.669 against 0.693 overall. Endpoint degrees are taken from each universe's true network with the queried edge removed, a diagnostic only; test truth is never used for selection.

Table 10: Agreement of the inferred motif graphs with the supervised quantities of Eq. (9) on PRING BFS, over the balanced non-self pairs of Fig. 4 (63,246 training, 9,406 validation, and 60,256 test pairs).

	Train	Validation	Test
Path-product Spearman ρ, all pairs	0.602	0.530	0.400
interacting pairs only	0.328	0.279	0.357
non-interacting pairs only	0.224	0.270	0.169
partial ρ, controlling both endpoint degrees	0.596	0.529	0.384
Presence AUROC of π_{uv}, all pairs	0.876	0.793	0.693
interacting pairs only	0.799	0.739	0.733
non-interacting pairs only	0.809	0.697	0.600
averaged within endpoint-degree quintiles	0.848	0.782	0.669

6.3.3 Test-Time Prompt Interventions

The ablations of Table 4 retrain the model, so we also intervene on the trained BFS checkpoint of Table 1 at scoring time, replacing the motif graph the predictor reads for every test pair while its $\mathcal{V}_{\text{val}}$-frozen threshold replays unchanged (Table 11). The transplant row gives every pair the inferred graph of another test pair, drawn by one seeded permutation of the test universe, so the prompts keep their marginal distribution and only their pairing with the queried sequences breaks: every metric worsens, with graph similarity falling from 0.413 to 0.372, relative density from 0.894 to 0.540, and the three MMD ratios rising by 26–58%. The topology gain therefore comes from the pair-specific

closure signal the inferred graph carries, not from the prompt's presence or its average graph: giving every pair the training-mean motif graph, or closing the prompt gates, removes the prompt outright and collapses the operating point (RD 0.200 and 0.271).

Table 11: Test-time prompt interventions on the PRING BFS split with the checkpoint of Table 1, each replaying GraSPPI's $\mathcal{V}_{\text{val}}$-frozen threshold on the test subgraphs.

Test-time prompt	GS↑	RD→1	Deg.↓	Clus.↓	Spec.↓
Inferred motif graph (GraSPPI)	0.413	0.894	6.777	6.024	8.968
Another test pair's inferred graph	0.372	0.540	8.543	9.488	12.855
Training-mean motif graph for every pair	0.310	0.200	42.077	32.497	54.296
Prompt gates closed	0.360	0.271	29.387	23.198	39.487

6.3.4 Density-Matched Comparison

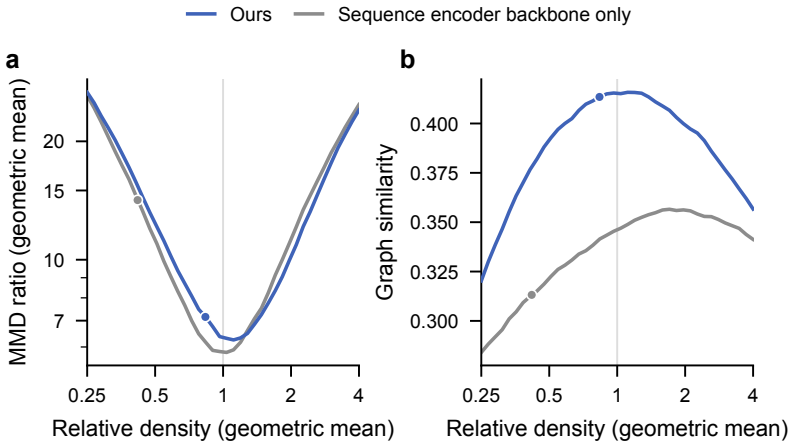

Figure 5: GraSPPI and the variant without all structural losses on the PRING BFS test networks against relative density, each re-thresholded on the test subgraphs to 40 target densities from 0.25 to 4 (a diagnostic, since these thresholds read test labels). (a) Geometric mean of the degree, clustering, and spectral MMD ratios. (b) Graph similarity. Dots mark each model's threshold frozen on $\mathcal{V}_{\text{val}}$; the vertical line marks the true density.

The rows of Table 4 differ both in how their scores rank pairs and in the density their $\mathcal{V}_{\text{val}}$-frozen thresholds select. To separate the two, we re-threshold GraSPPI and the variant without all structural losses – which keeps the motif prompt and its graph supervision but trains without the structural stream – to matched densities on the test subgraphs (Fig. 5). At matched density the discrepancy gap closes: the two geometric-mean MMD ratios stay within 16% of each other at every density (6.34 against 5.83 at the true density), and GraSPPI's is the lower one only above a relative density of 1.25. The structural stream shows instead in edge recovery: GraSPPI's graph similarity is 0.016–0.074 above the variant's at every density (0.415 against 0.346 at the true density). The 1.9–2.1$\times$ lower discrepancies of GraSPPI in Table 4 therefore come mainly from its operating point, a $\mathcal{V}_{\text{val}}$-selected density much closer to the true one (RD 0.894 against 0.483). Both thresholds are frozen by the same rule without any test information, so the operating point is itself learned: the structural stream shapes score distributions whose validation-selected threshold generalizes to the density of the held-out networks.

6.3.5 Function-Oriented Tasks

We port PRING's evaluation scripts (Zheng et al., 2025) to our score files and evaluate GraSPPI at the operating point of Table 1. *Complex pathway* scores the 1,579 released complex pairs over 397 proteins and reads the predicted subgraph induced by each complex: pathway recall and precision against the complex's true interactions, and connectivity, the share of connected true complexes whose predicted subgraph is also connected. *GO enrichment* scores all pairs of the 2,018 BFS test proteins, partitions the predicted and true networks with Louvain (resolution 1, seed 42), and

matches each predicted community to its best true community (Jaccard at least 0.2); the functional alignment is the mean Jaccard of the two communities' enriched GO biological-process terms from g:Profiler, and the consistency ratio is the mean within-community GO Jaccard of the predicted communities over that of the true ones. *Essential proteins* ranks PRING's 100 essential and 100 non-essential BFS test proteins by edge-clustering centrality in the predicted network, and reports precision@100 against the top 100 of the true network and the kernel-density overlap of the two classes' centrality distributions. Three properties of the released complex set qualify the complex columns: 325 of its 1,180 complex interactions are self-interactions (without them GraSPPI's PR is 0.588; the published recalls are not recomputed on this subset); 58 of the 397 complex proteins are PRING-BFS training proteins, although no complex interaction is in the training graph; and every complex negative is a non-interacting pair inside the same complex (GraSPPI does not rank them below the interactions: AUROC 0.47 over all complex pairs, 0.44 over non-self pairs). Avg. Rank is the mean per-column rank over the rows with all seven metrics, with tied values sharing the best rank; PRING's printed integer ranks follow no tie rule we could reproduce, so we recompute them for every row.

Table 12: All published function-oriented results on PRING (Zheng et al., 2025) with GraSPPI. Shading marks the first, second, and third best value per column; /: not evaluated.

		Complex pathway			GO enrichment		Essential protein		
Category	Method	PR↑	PP↑	PC↑	FA↑	CR↑	P@100↑	DO↓	Avg. Rank
Seq. sim.	SPRINT (Li & Ilie, 2017)	0.526	0.856	0.752	0.174	0.655	0.54	0.727	6.00
Sequence	PIPR (Chen et al., 2019)	0.160	0.588	0.323	0.182	0.654	0.52	0.834	8.71
	D-SCRIPT (Sledzieski et al., 2021)	0.225	0.466	0.316	0.175	0.678	0.47	0.629	8.43
	Topsy-Turvy (Singh et al., 2022)	0.240	0.613	0.338	0.232	0.635	0.43	0.755	8.71
PLM	PPITrans (Yang et al., 2024)	0.583	0.863	0.818	0.250	0.756	0.48	0.886	5.14
	TUnA (Ko et al., 2024)	0.526	0.864	0.794	0.259	0.622	0.57	0.849	6.00
	PLM-interact (35M) (Liu et al., 2025)	0.550	0.847	0.880	0.335	0.785	0.80	0.423	2.86
	PLM-interact (650M) (Liu et al., 2025)	0.549	0.862	0.887	0.368	0.844	0.77	0.440	2.57
Structure	Struct2Graph (Baranwal et al., 2022)	0.413	0.742	0.608	0.154	0.649	0.45	0.841	8.86
	TAGPPI (Song et al., 2022)	0.487	0.781	0.713	0.168	0.649	0.43	0.828	8.14
	RF2-Lite (Humphreys et al., 2024)	0.554	0.874	0.135	/	/	/	/	/
Ours	GraSPPI	0.743	0.872	0.423	0.332	0.875	0.67	0.605	3.00

6.3.6 Published Results on PRING

Table 13 lists every network reconstruction result that Zheng et al. (2025) print for the three Human test splits, as printed, below our own runs. PRING trains its baselines on 1:1 pairs from the full training graph (80% training, 20% validation), thresholds every model at probability 0.5, evaluates PLM-interact from its released weights, and scores RF2-Lite on only 10% of the test subgraphs. Our runs instead exclude the validation region $\mathcal{V}_{\text{val}}$ from training and replay one threshold frozen on it (Section 6.2.4). The two protocols share the test subgraphs and the five metrics, but not the operating point: at probability 0.5 every published method except RF2-Lite over-predicts density (RD 1.13–12.3), which also inflates its discrepancies, so the published and our rows of SPRINT, TUnA, and PLM-interact (35M) differ in all five metrics, most in RD and the three MMD ratios.

6.3.7 Published Results on the Bernett Benchmark

Table 14 lists every result we found that the authors of a method printed for the full Bernett test set (Intra-2, 52,048 balanced pairs), rounded to two decimals, below our own runs. Each value is taken from the table, text, or supplement of the paper that ran the method; tables that only copy these values are not listed again. We leave out two later studies whose models are scored after removing long proteins from the test set, and one graph-based method whose test-time graph is not specified. Deep models of Bernett et al. (2024) train on Intra-1 and validate on Intra-0; its methods without tunable hyperparameters train on both. Published thresholds are 0.5 or unstated.

PLM-interact trains on 80% of Intra-1 because of a pair-length limit, and its F1 is read from the bar chart of its Figure 4a. Our SPRINT run trains on Intra-1 alone and takes the maximum-F1 threshold on Intra-0, where it predicts almost every pair positive; our PLM-interact (35M) run scores the authors' HumanV11 checkpoint, which was not trained on this benchmark.

Table 13: All network reconstruction results on the three PRING splits. Top: our runs, each at one threshold frozen on $\mathcal{V}_{\text{val}}$ († authors' official code, adapted to our ESM3 features and training pairs, with our checkpoint and threshold selection; ‡ model-agnostic plug-in reproduced from the paper on our sequence encoder backbone, with the same selection; Section 6.2.2). Below: results published by Zheng et al. (2025) (their Table 3), at probability 0.5 and with the precision printed there.

	BFS					DFS					RW				
Method	GS↑	RD→1	Deg.↓	Clus.↓	Spec.↓	GS↑	RD→1	Deg.↓	Clus.↓	Spec.↓	GS↑	RD→1	Deg.↓	Clus.↓	Spec.↓
Our runs															
GraSPPI (Ours)	0.413	0.894	6.777	6.024	8.968	0.510	0.942	89.596	19.150	21.161	0.517	1.058	14.712	4.140	6.527
SPRINT† (Li & Ilie, 2017)	0.278	3.120	13.210	13.411	12.077	0.260	0.946	118.828	21.813	26.641	0.220	0.757	25.001	14.042	14.579
TUnA† (Ko et al., 2024)	0.399	0.569	9.433	8.200	14.741	0.504	0.979	108.909	27.207	24.756	0.534	1.007	16.617	5.409	8.803
PLM-interact (35M)† (Liu et al., 2025)	0.432	1.887	10.574	9.690	7.881	0.398	0.608	195.092	56.568	36.346	0.509	1.465	39.320	10.131	10.589
LLP† (Guo et al., 2023)	0.349	0.369	19.740	17.010	27.630	0.457	1.285	107.407	32.493	19.446	0.559	1.008	23.502	8.702	11.914
CAZI-MBN† (Deng et al., 2026)	0.299	1.113	22.830	15.620	27.568	0.280	1.269	255.814	52.530	42.150	0.402	1.981	44.068	18.592	22.069
UniPrompt† (Huang et al., 2025)	0.374	0.876	10.953	9.230	16.937	0.350	1.837	148.014	62.927	21.738	0.453	1.868	25.032	10.070	11.297
L3-PPI‡ (Gao et al., 2026)	0.305	0.840	6.682	6.076	14.030	0.300	1.046	94.436	58.356	14.354	0.461	1.099	15.496	5.551	7.398
Published in Zheng et al. (2025), Table 3															
SPRINT (Li & Ilie, 2017)	0.227	1.61	10.9	12.5	11.2	0.145	5.91	316	207	53.8	0.178	1.32	17.9	26	15.1
PIPR (Chen et al., 2019)	0.209	4.39	81.7	40.5	36.2	0.0749	12.3	518	335	77.0	0.196	5.75	165	107	41.1
D-SCRIPT (Sledzieski et al., 2021)	0.215	1.13	15.4	18.8	29.7	0.123	6.40	302	237	67.0	0.209	5.82	134	64.8	52
Topsy-Turvy (Singh et al., 2022)	0.183	1.74	17.3	11.4	14.6	0.130	3.22	180	201	65.9	0.232	4.88	104	64.8	36.7
PPITrans (Yang et al., 2024)	0.362	3.39	52.2	29.1	26.3	0.314	4.54	418	270	46.6	0.449	2.57	71.9	46.1	19.6
TUnA (Ko et al., 2024)	0.342	2.99	47.8	30.2	22.9	0.289	4.44	416	272	46.0	0.450	2.23	55.4	42.5	13.7
PLM-interact (35M) (Liu et al., 2025)	0.383	2.29	24.4	16.0	12.8	0.322	2.85	209	78.2	36.3	0.429	2.47	52.6	22.6	13.6
PLM-interact (650M) (Liu et al., 2025)	0.396	1.64	30.9	14.8	16.5	0.350	2.03	236	63.6	42.2	0.491	1.76	50.3	21.2	15.4
Struct2Graph (Baranwal et al., 2022)	0.27	5.58	95.1	50.5	39.2	0.132	10.8	508	321	67.6	0.237	5.83	155	103	38.3
TAGPPI (Song et al., 2022)	0.283	4.56	70.8	36.2	29.5	0.143	9.94	491	309	65.3	0.256	4.87	127	74.2	28.6
RF2-Lite (Humphreys et al., 2024) (10% of test)	0.227	0.524	43.8	31.7	34.9	0.243	1.10	226	39.3	61.8	0.171	0.514	81.7	73.9	60.9

Table 14: All pairwise classification results on the Bernett test set. Top: our runs († authors' official code, adapted to our ESM3 features and training pairs, with our checkpoint and threshold selection; ‡ reproduced from the paper on our backbone; Section 6.2.2). Below: published results, grouped by the paper that reports them. * Balanced accuracy. –: not reported.

Method	AUROC	AUPRC	Acc.	F1	MCC
Our runs					
GraSPPI (Ours)	0.71	0.71	0.64	0.59	0.29
LLP† (Guo et al., 2023)	0.71	0.70	0.65	0.67	0.30
CAZI-MBN† (Deng et al., 2026)	0.51	0.51	0.50	0.55	0.01
L3-PPI‡ (Gao et al., 2026)	0.69	0.68	0.53	0.14	0.15
UniPrompt† (Huang et al., 2025)	0.65	0.65	0.59	0.64	0.19
SPRINT† (Li & Ilie, 2017)	0.52	0.52	0.50	0.67	0.02
PLM-interact (35M)† (Liu et al., 2025)	0.61	0.64	0.58	0.41	0.19
Published in Bernett et al. (2024), Supplementary Table S5 and Figures S16–S28					
Richoux-FC (Richoux et al., 2019)	0.52	0.51	0.52	0.46	0.05
Richoux-LSTM (Richoux et al., 2019)	0.51	0.51	0.52	0.56	0.03
DeepFE (Yao et al., 2019)	0.52	0.51	0.52	0.49	0.04
PIPR (Chen et al., 2019)	0.53	0.53	0.52	0.49	0.04
D-SCRIPT (Sledzieski et al., 2021)	0.50	0.51	0.50	0.28	0.01
Topsy-Turvy (Singh et al., 2022)	0.59	0.59	0.56	0.37	0.15
SPRINT (Li & Ilie, 2017)	0.52	0.51	0.52	–	–
RF-PCA	0.52	0.51	0.52	0.34	0.04
SVM-PCA	0.51	0.51	0.51	0.52	0.03
RF-MDS	0.51	0.50	0.51	0.25	0.02
SVM-MDS	0.50	0.50	0.50	0.56	−0.00
RF-node2vec	0.51	0.51	0.51	0.31	0.03
SVM-node2vec	0.51	0.51	0.52	0.57	0.03
Global/local consistency	0.50	0.50	0.50	0.67	0.00
Published in Ko et al. (2024), Figure 2b					
ESM2-MLP	0.70	0.68	0.64*	0.64	0.28
ESM2-GP	0.69	0.68	0.63*	0.62	0.26
T-FC	0.69	0.68	0.63*	0.65	0.27
TUnA-ProSE	0.70	0.68	0.65*	0.65	0.29
TUnA	0.70	0.69	0.65*	0.65	0.30
Published in Sledzieski et al. (2024), Table 2 (ESM-2, at most 1,024 residues)					
Frozen ESM-2 + MLP	–	0.68	0.63	0.63	0.26
ESM-2 + LoRA	–	0.60	0.61	0.67	0.23
ESM-2 fine-tuned	–	0.62	0.60	0.63	0.21
Other published results					
PLM-interact (Liu et al., 2025), Fig. 4a	0.70	0.69	–	0.66	–
C3PI (Hosseini et al., 2025), Table 3	0.70	0.70	0.65	0.67	0.29
MINT (Ullanat et al., 2026), Supp. Table 1	–	0.69	–	–	–