SHARED MOTIFS, SPECIFIC MATCHES: LEARNING PROTEIN–GLYCAN COMPATIBILITY

Anonymous authors
Paper under double-blind review

ABSTRACT

Predicting protein–glycan binding for unseen molecules requires recognition patterns that transfer across measured pairs. Recent approaches provide increasingly detailed glycan representations, but models must still learn which local structures matter for each protein from labels assigned to complete molecular pairs. We organize this supervision around recurring motifs: an unseen glycan can share local patterns with measured glycans, while their contribution to binding depends on the protein partner. We introduce **MotifMatch**, which learns one shared parameter vector per motif from measurements on all glycans carrying it. Explicit, linkage-aware motif counts combine these vectors into glycan factors, which sequence-derived protein factors rescale through Hadamard matching. We show that gradient coupling through the shared motif parameters factorizes into motif overlap and the alignment of pair-conditioned responses. Fixed-budget experiments show that measurements on glycans with shared motifs improve prediction for held-out glycans. Controlled comparisons further show that the benefit of Hadamard matching depends on the glycan representation. On GlycanML, MotifMatch achieves the highest mean ranking scores for unseen proteins, unseen glycans, and pairs where both partners are unseen. With protein features cached for all methods, MotifMatch trains $7.4\times$ and infers $5.5\times$ faster than the fastest baseline in each stage.

1 INTRODUCTION

Protein–glycan recognition contributes to cellular communication, immune responses, and host–pathogen interactions. Predicting binding responses from protein sequences and glycan structures can characterize lectin preferences and prioritize candidate binding partners (Lundstrøm et al., 2022). For a protein of interest, useful predictions distinguish its preferences across candidate glycans and guide further measurements. The diversity of both partners creates a vast pairing space, only part of which can be measured experimentally. Models must therefore generalize to unmeasured pairs, including those involving proteins, glycans, or both that are absent from the training measurements.

Recent approaches have expanded the structural information available for this task. LectinOracle combines pretrained protein features with glycan graphs, while higher-order graph modeling and hierarchical atom–monosaccharide representations capture additional structural relationships (Lundstrøm et al., 2022; Joeres & Bojar, 2025; Xu et al., 2025b). These descriptions expose local structure and its wider molecular context, giving models more information with which to distinguish glycans. Supervision, however, remains a response assigned to a complete molecular pair; it does not identify which local structures matter for the measured protein. Learning transferable recognition patterns therefore requires connecting this pair-level evidence across molecules. How should a measurement on one protein–glycan pair inform predictions for another?

Two observations shape our answer. First, an unseen glycan can be a new combination of local motifs found in measured glycans. Measurements on different glycans can train common motif parameters even when their remaining structures differ. A new glycan can then reuse these parameters according to its own composition, drawing on supervision from other molecules. Second, the same motif can contribute differently to the preferences of different proteins. Structural evidence pooled across glycans must therefore be used in the context of the current protein. Together, these observations motivate pooling measurements across glycans through common structural parameters and letting each protein use the learned factors to rank its own candidate glycans (Figure 1).

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.

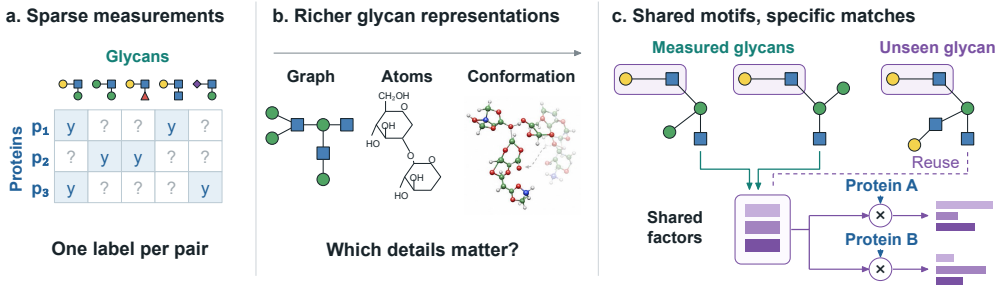


Figure 1: Shared motifs, specific matches. (a) Measurements supervise complete protein–glycan pairs and cover only part of the pairing space. (b) Richer molecular descriptions add structural detail, but which structures matter for each protein must still be inferred from pair labels. (c) Recurring motifs connect observations across glycans through shared factors; protein factors modulate their use for the current pair. Structures and factor values are schematic.

We introduce **MotifMatch**, which builds these two observations into the model. For shared motifs, glycans are represented by counts of linkage-aware motifs with terminal context, reflecting the tendency of lectins to recognize terminal epitopes (Taylor et al., 2017; Bojar et al., 2022); each motif owns one column of a shared projection, so measurements on every glycan carrying it train the same parameters. For specific matches, sequence-derived protein factors rescale the resulting glycan factors through a Hadamard product, so each protein weighs the shared motifs in its own way. This design answers the question above: a measurement on one pair informs another according to their motif overlap and the alignment of their pair-conditioned responses (Proposition 1).

Experiments on GlycanML (Xu et al., 2025a) support both parts of this design. Shared motifs carry measurements across glycans: at a fixed measurement budget, redirecting measurements to other glycans carrying a motif lowers error on held-out glycans carrying it for all twelve motifs tested, and MotifMatch gains the most from this support despite starting with the highest error. Specific matches depend on the shared units: the same Hadamard scorer improves top-candidate selection with motif counts but not with circular neighborhoods, so the units and their protein-specific use must be designed together. Together, they give MotifMatch the best mean ranking scores for unseen proteins, unseen glycans, and both, with faster training and inference than every baseline.

2 Related Work

Early computational approaches to glycan recognition used explicit motifs. Klamer et al. (2017) described lectin specificities using substitution constraints and logical operators, and CCARL combined subtree mining with supervised motif selection to distinguish binding from nonbinding glycans for individual lectins (Coff et al., 2020). GlyNet carried this line into neural prediction, using counts of small monosaccharide subgraphs, terminal monosaccharides, and modifications to predict array responses through one output per predefined protein sample (Carpenter et al., 2022).

More recent work has placed increasing emphasis on richer molecular descriptions for binding prediction. SweetTalk and SweetNet used sequence and graph representations for binding prediction, while glyBERT and GLAMOUR incorporated local and global context or monomer chemistry (Bojar et al., 2021; Burkholz et al., 2021; Dai et al., 2021; Mohapatra et al., 2022). LectinOracle paired learned glycan graphs with pretrained protein language model features to score proteins absent from training (Lundstrøm et al., 2022). Later models for binding prediction introduced higher-order relations in GIFFLAR, hierarchical atom–monosaccharide modeling in GlycanAA, and atom-level fingerprints in MCNet (Joeres & Bojar, 2025; Xu et al., 2025b; Carpenter et al., 2025). MotifMatch revisits explicit motifs to organize how pair-level supervision is shared across glycans. A shared mapping pools supervision from glycans carrying the same motifs, while sequence-derived protein factors modulate the resulting glycan factors for each binding partner.

Interaction models also differ in how they combine the binding partners. LectinOracle and DeepDTA concatenate partner encodings, while MolTrans and DrugBAN model local cross-molecular associations (Lundstrøm et al., 2022; Öztürk et al., 2018; Huang et al., 2021; Bai et al., 2023). Multiplicative fusion combines projected factors efficiently through compact bilinear pooling,

Hadamard pooling, or low-rank multimodal fusion (Fukui et al., 2016; Kim et al., 2017; Liu et al., 2018). MotifMatch applies Hadamard matching to factors learned from explicit motif counts and uses controlled comparisons to examine how its benefit depends on the glycan representation.

3 METHOD

Given measurements $\mathcal{D} = \{(p_i, g_i, y_i)\}_{i=1}^{N}$, MotifMatch learns to predict binding responses from protein sequences and glycan structures. A shared projection turns each glycan's motif counts into factors that are rescaled by sequence-derived protein factors. The matched factors enter a nonlinear readout to give a pair score, which is combined with additive terms for each partner's response tendency. All trainable components learn jointly from the measured pair responses (Figure 2).

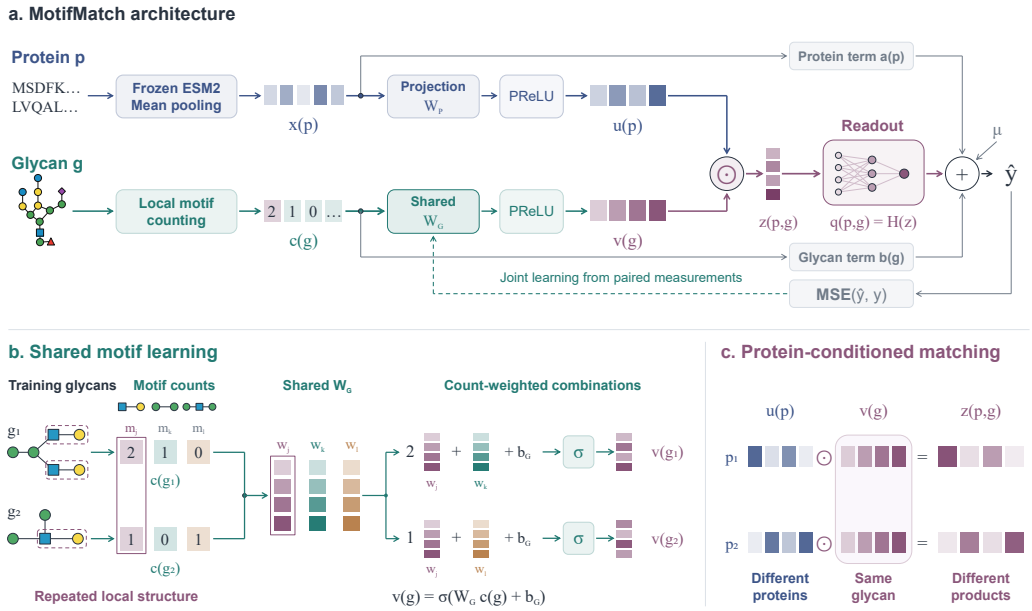


Figure 2: MotifMatch. (a) Protein features and motif counts are projected to factors whose Hadamard product feeds a nonlinear readout. Additive branches capture single-molecule response tendencies. (b) Glycans with a common motif reuse the same mapping column, weighted by their counts. (c) Protein factors modulate glycan factors. Trainable components learn jointly from pair responses.

3.1 SHARED MOTIFS: ONE PARAMETER COLUMN PER MOTIF

We represent each glycan by a count vector $c(g) \in \mathbb{R}^K$, where $c_j(g)$ counts occurrences of motif j. A shared trainable projection combines these counts into glycan factors:

$$v_\theta(g) = \sigma_G(W_G c(g) + b_G) \in \mathbb{R}^d, \qquad W_G c(g) = \sum_{j=1}^{K} c_j(g) w_j, \tag{1}$$

where σ_G is coordinate-wise PReLU and w_j is column j of $W_G \in \mathbb{R}^{d \times K}$. The column w_j is reused whenever motif j appears, so measurements on structurally different glycans contribute to learning the same vector (Figure 2(b)). The projection combines these contributions across factor coordinates, so each factor can draw on several motifs and each motif can contribute to several factors.

Monosaccharide coordinates record composition, while linkage-aware two- and three-monosaccharide patterns record local arrangements. For the multi-residue patterns, we also include terminal-context coordinates that count occurrences with at least one terminal endpoint. These distinguish the same linked pattern at the end of a glycan from one occurring internally, reflecting the tendency of lectins to recognize terminal epitopes (Taylor et al., 2017; Bojar et al., 2022). Counting every occurrence

also preserves how often a pattern is repeated within a glycan:

$$c_j(g) = \sum_{s \in \mathcal{O}(g)} \mathbf{1}\{s \text{ matches motif } j\}. \tag{2}$$

Here, $\mathcal{O}(g)$ lists local occurrences with monosaccharide, linkage, and terminal-context labels.

Figure 2(b) illustrates this reuse with two glycans whose count vectors are $c(g_1) = (2, 1, 0)$ and $c(g_2) = (1, 0, 1)$. Their preactivations, $2w_1 + w_2 + b_G$ and $w_1 + w_3 + b_G$, both depend on w_1, while the other columns distinguish their structural contexts. Measurements on either glycan therefore contribute to training the vector for their shared motif. The same construction lets an unseen glycan use motif parameters trained on other molecules, combining them according to its own counts.

3.2 Specific matches: protein-specific use of shared factors

A glycan's factors can be reused across protein partners. To score a particular pair, MotifMatch rescales them with factors derived from the protein sequence. We mean-pool frozen ESM2 features (Lin et al., 2023) to obtain a sequence representation $x(p) \in \mathbb{R}^m$, then project it into protein factors:

$$u_\theta(p) = \sigma_P(W_P x(p) + b_P) \in \mathbb{R}^d, \tag{3}$$

where $W_P \in \mathbb{R}^{d \times m}$ and σ_P is PReLU. We pass the factors' Hadamard product to the readout:

$$z_\theta(p, g) = u_\theta(p) \odot v_\theta(g), \qquad q_\theta(p, g) = H_\theta(z_\theta(p, g)). \tag{4}$$

Here, H_θ has a single PReLU hidden layer and a linear scalar output.

Through this rescaling, the protein determines how each glycan factor enters the readout. Writing B and b_H for the hidden weights and bias, the readout's preactivation is

$$B z_\theta(p, g) + b_H = B_p v_\theta(g) + b_H, \qquad B_p = B \operatorname{Diag}(u_\theta(p)).$$

The protein vector rescales the columns of B, changing the weight of each glycan factor as it enters the hidden layer. The hidden units combine these factors, and their nonlinear responses let the readout use different combinations for different protein–glycan pairs.

The full prediction combines this score with the response tendencies of each partner:

$$\widehat{F}_\theta(p, g) = \mu_{\text{ref},\theta} + a_\theta(p) + b_\theta(g) + q_\theta(p, g). \tag{5}$$

The protein term $a_\theta(p)$ shifts all candidates for that protein equally, while the glycan term $b_\theta(g)$ contributes the same value across protein partners. The two branches use separate parameters and take $x(p)$ and $c(g)$ as inputs, respectively. We train all branches jointly with Adam and squared loss $\ell_i = e_i^2$ on standardized targets, where $e_i = \widehat{F}_\theta(p_i, g_i) - \widetilde{y}_i$. With molecular features as inputs, the predictor can score pairs with either or both partners unseen in training.

3.3 How one measurement informs another

A measurement on one glycan can affect another pair's prediction by changing motif parameters used by both glycans. To trace this influence, write $q_i = q_\theta(p_i, g_i)$, $c_i = c(g_i)$, and $\eta_i = W_G c_i + b_G$ for a measured pair i. The gradient $\delta_i = \nabla_{\eta_i} q_i$ describes how a change in this projection affects the pair's score. The chain rule then gives the gradients of the score and training loss:

$$\nabla_{W_G} q_i = \delta_i c_i^\top, \qquad \nabla_{w_j} \ell_i = 2 e_i c_j(g_i) \delta_i. \tag{6}$$

For each column, the motif count scales the observation's contribution. The residual indicates whether the prediction is too high or too low, while δ_i describes how the shared motif vector affects the score.

The direction of this learning signal depends on the protein partner as well as the glycan. For $\eta_g = W_G c(g) + b_G$, expanding the sensitivity through the projection and readout gives

$$\delta_\theta(p, g) := \nabla_{\eta_g} q_\theta(p, g) = \sigma_G'(\eta_g) \odot u_\theta(p) \odot \nabla_z H_\theta(z_\theta(p, g)). \tag{7}$$

The protein enters through $u_\theta(p)$, while the glycan's other motifs affect the projection and readout activation states. Measurements on glycans carrying the same motif can therefore provide different learning directions, depending on the protein and the other motifs present (Appendix A.1).

To determine the effect on a target pair, we also need its response to the same parameter change. The inner product of the two score gradients gives the target score's rate of change when the shared parameters move along the support pair's score gradient. For the shared projection, this inner product factorizes into motif overlap and response alignment:

Proposition 1 (Overlap–alignment factorization)**.** *Let J index motif coordinates, W_J contain their columns, and $c_{i,J}$ denote the corresponding count subvector. For support pair i and target pair t, their prediction gradients couple through W_J as follows:*

$$\kappa_J(t,i) := \langle \nabla_{W_J} q_t, \nabla_{W_J} q_i \rangle_F = \left(c_{t,J}^\top c_{i,J}\right)\left(\delta_t^\top \delta_i\right). \tag{8}$$

The count term measures overlap in the selected motifs; the sensitivity term measures alignment of the pair-specific responses. Shared motifs connect observations through common columns, while protein-conditioned scoring determines how changes in those columns affect each pair.

To relate this coupling to prediction error, consider a negative support-loss gradient step on W_J, holding the remaining parameters fixed. The target-loss directional derivative is then $-4e_t e_i \kappa_J(t,i)$ (Appendices A.2 and A.3). When the two pairs share motifs and their residuals have the same sign, aligned sensitivities lower the target loss to first order. Opposite residual signs reverse the effect, so both pairs' errors determine the direction of the loss change.

The experiments examine how this shared supervision is supplied and used. The support experiment changes coverage by other motif-carrying glycans at a fixed measurement budget and measures prediction error on held-out targets (Section 4.3). The controlled comparison keeps the measurements fixed and changes the representation and scoring function (Section 4.4).

4 Experiments

The experiments examine generalization to unseen partners, the use of shared-motif support, and how candidate selection depends on the combination of representation and scoring.

4.1 Prediction task and evaluation

Prediction task. GlycanML (Xu et al., 2025a) contains 564,647 measured pairs, 1,392 protein entries, and 927 glycans. The official split holds out proteins; additional entity-cluster splits hold out glycans or both partners. All models use the same frozen ESM2 features, training-label standardization, and validation-based checkpoint selection. We report mean $\pm$ sample SD over three runs, varying training seeds on the official split and both split and training seeds on the additional splits.

Evaluation. We evaluate response accuracy and candidate selection separately. Normalized mean squared error (NMSE) measures response error on a fixed scale (the training-label variance on the official split); Global Spearman measures rank association across all test pairs. For a protein's candidate list, Protein Spearman measures recovery of its full preference ordering, while NDCG@20 emphasizes strong responses among the first 20 candidates. NDCG uses responses above each protein's minimum as relevance, discounts lower positions logarithmically, and normalizes by the ideal ordering to a zero-to-one scale. Protein-level metrics weight proteins equally, so heavily measured proteins do not dominate. Appendix C.1 details normalization and tie handling.

4.2 Generalization to unseen partners

Generalization to a new protein requires translating its sequence information into preferences over glycans. We test this on the official protein-cold split, where test proteins have no training binding measurements and all methods use the same ESM2 features. Holding out glycans or both partners then tests whether these preferences also transfer to unseen glycan structures.

Comparison methods. The baselines differ in the structural units from which glycan features are built (Table 1). IUPAC n-gram counts local sequence fragments; BiLSTM learns context along the serialized glycan. LectinOracle and SweetNet learn over monosaccharide-level graphs, whereas the GIFFLAR implementation of GNNGLY (Alkuhlani et al., 2023; Joeres & Bojar, 2025) uses atomic connectivity, GlycanAA connects atom and monosaccharide scales, and GIFFLAR introduces higher-order structural relations. PMCf MF3_c uses chirality-aware atom-neighborhood counts from MCNet, adapted to condition on protein features (Carpenter et al., 2025).

Table 1: Prediction for unseen proteins on the official split. All methods use frozen ESM2 protein features. Entries are mean ± sample SD over three training seeds; bold and underline indicate the best and second-best means. Rows are grouped by the glycan representation.

Model	NMSE ↓	Global ρ ↑	Protein ρ ↑	NDCG@20 ↑
IUPAC sequence features				
IUPAC n-gram	0.9656 ± 0.0328	0.2629 ± 0.0040	$\underline{0.3224 \pm 0.0052}$	0.3965 ± 0.0522
BiLSTM	$\underline{0.9220 \pm 0.0515}$	0.2600 ± 0.0124	0.3190 ± 0.0224	0.4440 ± 0.0579
Monosaccharide graph encoders				
LectinOracle	0.9268 ± 0.0239	$\underline{0.2709 \pm 0.0136}$	0.3196 ± 0.0124	0.4377 ± 0.0514
SweetNet	0.9231 ± 0.0083	0.2593 ± 0.0068	0.3119 ± 0.0111	$\underline{0.4463 \pm 0.0079}$
Atomic and higher-order graph encoders				
GNNGLY	0.9858 ± 0.0161	0.1474 ± 0.0084	0.1558 ± 0.0116	0.3562 ± 0.0167
GlycanAA	0.9753 ± 0.0163	0.2566 ± 0.0084	0.3028 ± 0.0076	0.3402 ± 0.0377
GIFFLAR	0.9772 ± 0.0011	0.2215 ± 0.0107	0.2521 ± 0.0078	0.3151 ± 0.0079
Explicit structural features				
PMCf MF3_c	0.9549 ± 0.0050	0.2164 ± 0.0121	0.2505 ± 0.0084	0.3876 ± 0.0159
MotifMatch	$\mathbf{0.9160 \pm 0.0077}$	$\mathbf{0.2998 \pm 0.0090}$	$\mathbf{0.3534 \pm 0.0035}$	$\mathbf{0.4708 \pm 0.0035}$

For a new protein, absolute responses remain hard to predict: even the lowest NMSE, 0.916, is close to the training-label variance (Table 1). Within-protein ranking separates the methods more clearly. MotifMatch has the best mean on all four metrics, raising Protein Spearman and NDCG@20 to 0.353 and 0.471 from the strongest competing values of 0.322 and 0.446. Its advantage is thus clearest in choosing which glycans to test first for an unseen protein.

The baseline comparisons indicate that how structural detail is organized matters as much as how much is supplied. Adding a monosaccharide scale to atomic structure helps markedly: GlycanAA reaches a Protein Spearman of 0.303, close to the monosaccharide graphs, whereas the atom-connectivity GNNGLY reaches 0.156. Higher-order relations in GIFFLAR, by contrast, do not improve on simpler monosaccharide graphs. PMCf shows that the pattern holds for fixed descriptors: its explicit atom-neighborhood counts also rank below the monosaccharide graphs on both protein-level measures.

Counting local fragments already recovers much of what learned encoders capture: IUPAC n-gram matches BiLSTM, LectinOracle, and SweetNet in Protein Spearman, though not in NDCG@20. MotifMatch also counts, but over linkage-aware patterns with terminal context, and its stronger shortlist is consistent with these units providing more useful local evidence.

Table 2: Response error and within-protein ranking on the additional cold splits. Subscripts give sample SD over three paired split and training seeds. †GIFFLAR uses a fixed unknown graph for 22 unparsed structures; full metrics appear in Appendix C.3.

	Glycan-cold			Double-cold		
Model	NMSE ↓	Protein ρ ↑	NDCG@20 ↑	NMSE ↓	Protein ρ ↑	NDCG@20 ↑
IUPAC n-gram	$0.9443_{\pm.0068}$	$0.1816_{\pm.0190}$	$0.5504_{\pm.0089}$	$1.0060_{\pm.0077}$	$0.0854_{\pm.0268}$	$0.4533_{\pm.0136}$
BiLSTM	$0.9329_{\pm.0160}$	$0.1782_{\pm.0400}$	$0.5357_{\pm.0160}$	$\mathbf{0.9971_{\pm.0054}}$	$0.0945_{\pm.0481}$	$0.4486_{\pm.0086}$
LectinOracle	$0.9291_{\pm.0170}$	$0.1902_{\pm.0205}$	$0.5702_{\pm.0057}$	$1.0276_{\pm.0108}$	$0.0919_{\pm.0254}$	$0.4702_{\pm.0121}$
SweetNet	$0.9206_{\pm.0141}$	$0.1887_{\pm.0189}$	$0.5685_{\pm.0125}$	$1.0215_{\pm.0159}$	$0.0894_{\pm.0277}$	$0.4615_{\pm.0198}$
GNNGLY	$0.9254_{\pm.0160}$	$0.1808_{\pm.0195}$	$0.5520_{\pm.0082}$	$1.0033_{\pm.0022}$	$0.0823_{\pm.0245}$	$0.4667_{\pm.0117}$
GlycanAA	$0.9423_{\pm.0054}$	$0.1753_{\pm.0201}$	$0.5379_{\pm.0062}$	$1.0033_{\pm.0059}$	$0.0890_{\pm.0307}$	$0.4594_{\pm.0176}$
GIFFLAR†	$0.9384_{\pm.0163}$	$0.1967_{\pm.0262}$	$0.5649_{\pm.0144}$	$1.0303_{\pm.0377}$	$0.0809_{\pm.0234}$	$0.4554_{\pm.0169}$
PMCf MF3_c	$0.9643_{\pm.0249}$	$0.1505_{\pm.0296}$	$0.5186_{\pm.0072}$	$0.9986_{\pm.0137}$	$0.0846_{\pm.0546}$	$0.4657_{\pm.0175}$
MotifMatch	$\mathbf{0.9098_{\pm.0325}}$	$\mathbf{0.1971_{\pm.0328}}$	$\mathbf{0.5723_{\pm.0138}}$	$1.0030_{\pm.0007}$	$\mathbf{0.0991_{\pm.0488}}$	$\mathbf{0.4751_{\pm.0039}}$

The additional cold splits reinforce the distinction between predicting response magnitudes and selecting candidates (Table 2). With glycans held out, predictions must reuse structures from other glycans; MotifMatch leads on all mean metrics, with monosaccharide graphs close in top-candidate selection. With both partners unseen, all NMSEs lie near the constant-mean reference; BiLSTM has the lowest error, but MotifMatch is best on every ranking metric.

4.3 SHARED MOTIFS CARRY MEASUREMENTS ACROSS GLYCANS

An unseen glycan can still share local motifs with measured glycans. To test whether this overlap supplies useful supervision, we vary the number of measured glycans carrying a target motif while keeping the measurement budget fixed. Four variants cross two 1,264-dimensional representations with two scoring functions. *Motif counts* records occurrences of the linkage-aware, terminal-context patterns in Section 3.1; *Circular binary* records the presence of typed circular neighborhoods of radii 0–3. Concat feeds concatenated factors to an MLP; Hadamard feeds their product to a readout of the same hidden width, forming MotifMatch when paired with motif counts.

Changing shared-motif support at a fixed measurement budget. All measurements on the motif-carrying target glycans are held out. Other glycans carrying the motif supply training support, while motif-negative test glycans serve as controls. Support k counts distinct glycans, measuring structural breadth rather than repeated measurements of one molecule. Support observations replace motif-negative training observations within the same proteins, preserving the total budget and each protein's observation count. For three initial motifs, all four variants are evaluated at $k = 0, 1, \ldots, 16$.

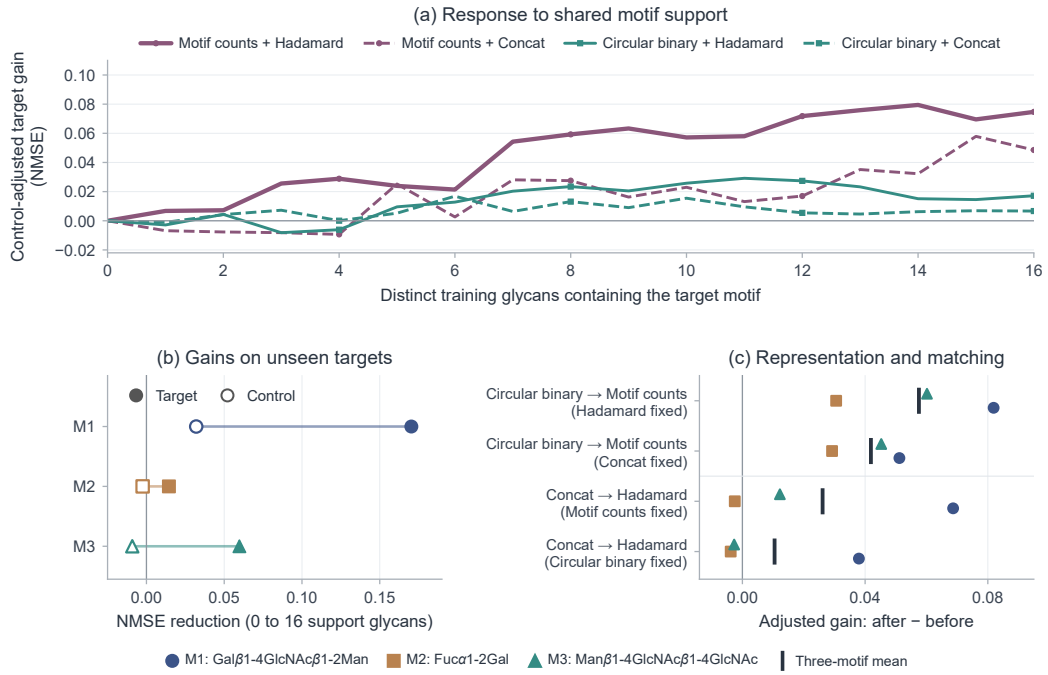


Figure 3: How shared observations become useful at a fixed budget. (a) Control-adjusted target gain, averaged over three motifs and three seeds. (b) MotifMatch's target and control gains from $k = 0$ to 16, averaged over three seeds. (c) Adjusted-gain changes for the replacements shown by arrows. Colored points are motif-specific seed means; black marks are their three-motif means. Table 3 summarizes the nine-motif extension; Appendix C.4 gives individual results and seed variation.

Measuring where the improvement occurs. Target NMSE measures prediction on the held-out motif-containing glycans. Protein-matched, motif-negative controls measure how the same training change affects other glycans. The target and control errors, $E_T(k)$ and $E_C(k)$, define the gains

$$G_T(k) = E_T(0) - E_T(k), \qquad G_{\text{adj}}(k) = G_T(k) - \big[E_C(0) - E_C(k)\big]. \tag{9}$$

Positive G_T means lower target error; positive G_{adj} means a greater reduction in normalized error on targets than on controls. Evaluation sets and their NMSE scales stay fixed across k, with gains measured from each model's zero-support condition.

Measurements on one glycan can help predict another that shares the supported motif (Figure 3b). Increasing support from zero to 16 glycans lowers MotifMatch's mean target error for all three initial motifs, while motif-negative controls change less.

Changing the structural units yields more consistent gains than changing the scorer across the three initial motifs (Figure 3a,c). Replacing Circular binary with motif counts raises the adjusted gain at $k = 16$ for each initial motif under both scorers. Replacing Concat with Hadamard has a more motif-dependent effect, with the largest benefit for M1.

The gains differ across the first three motifs, so we next ask whether the benefit extends to a wider range of structures. We repeat the fixed-budget protocol for nine additional motifs at $k = 0, 4, 16$, comparing their final target errors as well as their gains.

Table 3: Nine-motif support extension: target NMSE and gains from $k = 0$ to 16. Entries are mean $\pm$ sample SD over three seeds, with motifs averaged equally within each seed.

Model	Target, $k = 0$	Target, $k = 16$	$G_T \uparrow$	$G_{\mathrm{adj}} \uparrow$
Circular binary + Concat	0.9660 ± 0.0045	0.9477 ± 0.0666	0.0183 ± 0.0649	0.0112 ± 0.0591
Circular binary + Hadamard	0.9610 ± 0.0066	0.9465 ± 0.0883	0.0145 ± 0.0847	0.0139 ± 0.0857
Motif counts + Concat	0.9491 ± 0.0471	0.9442 ± 0.0687	0.0049 ± 0.0522	0.0275 ± 0.0460
Motif counts + Hadamard	0.9830 ± 0.0663	$\mathbf{0.9199 \pm 0.0536}$	$\mathbf{0.0631 \pm 0.1082}$	$\mathbf{0.0652 \pm 0.0910}$

Added support changes which representation–scoring combination predicts the targets most accurately (Table 3). MotifMatch starts with a higher mean target error than the other variants at zero support, but finishes with the lowest error at 16 supporting glycans. Its mean control error changes little, so the adjusted gain largely reflects improvement on the targets themselves; with motif counts and Concat, part of the gain instead comes from higher control error. Across both experiments, MotifMatch reduces mean target error for all twelve motifs tested.

4.4 Structural representation and protein-conditioned scoring

The support gains vary with both the glycan representation and the scorer. We now fix the measurements available to every model and ask how these choices affect candidate ranking. Table 4 compares the four core variants with additive, bilinear, and multiplicity controls. *Circular counts* records neighborhood multiplicities in the same vocabulary as Circular binary.

Table 4: Representation and scoring controls on the official split, reported as mean $\pm$ sample SD over three seeds. All inputs have 1,264 coordinates, with factor widths and training protocol fixed. Joint only retains Hadamard scoring and removes both single-molecule branches.

Representation	Joint score	NMSE $\downarrow$	Protein $\rho \uparrow$	NDCG@20 $\uparrow$
Representation and scoring comparison				
Circular binary	Concat	0.9341 ± 0.0025	0.3519 ± 0.0019	0.4259 ± 0.0243
Circular binary	Hadamard	0.9331 ± 0.0175	0.3514 ± 0.0066	0.4195 ± 0.0179
Motif counts	Concat	0.9366 ± 0.0059	0.3491 ± 0.0079	0.4330 ± 0.0075
Motif counts	Hadamard	$\mathbf{0.9160 \pm 0.0077}$	$\mathbf{0.3534 \pm 0.0035}$	$\mathbf{0.4708 \pm 0.0035}$
Joint-score controls with motif counts				
Motif counts	None (additive only)	1.0516 ± 0.0006	0.2880 ± 0.0006	0.1951 ± 0.0048
Motif counts	Hadamard, joint only	0.9165 ± 0.0119	0.3467 ± 0.0045	0.4664 ± 0.0263
Motif counts	Bilinear	0.9255 ± 0.0162	0.3266 ± 0.0242	0.4525 ± 0.0225
Multiplicity control: circular counts				
Circular counts	Concat	0.9363 ± 0.0028	0.3527 ± 0.0041	0.4233 ± 0.0135
Circular counts	Hadamard	0.9520 ± 0.0025	0.3477 ± 0.0063	0.4220 ± 0.0037

The same scorer change affects motif counts and circular neighborhoods differently (Table 4). Under Concat, the two representations give similar full-list correlations and top-candidate performance. Hadamard raises NDCG@20 from 0.433 to 0.471 with motif counts but brings no corresponding gain to Circular binary. Circular counts restores multiplicity in the same vocabulary without reproducing this gain, so what matters is which structures share a coordinate, not only how often they occur.

Without the joint branch, additive terms rank a common glycan set identically for every protein: shared tendencies still yield a positive Protein Spearman, but NDCG@20 falls to 0.195. The joint branch alone retains nearly all of the full model's accuracy and ranking quality. A scalar bilinear

score recovers much of the shortlist quality (NDCG@20 of 0.453) but lowers Protein Spearman to 0.327; keeping the matched vector for a nonlinear readout preserves both.

4.5 THE COST OF SHARING SUPERVISION

MotifMatch is the fastest method in both training and inference (Table 5). It trains $7.4\times$ faster than BiLSTM, the fastest training baseline, and infers $5.5\times$ faster than PMCf, the fastest inference baseline. All methods run on a single L20 GPU with cached ESM2 features; training time includes validation and early stopping, and inference covers 63,364 test pairs. Because BiLSTM and PMCf already avoid message passing over graphs, the remaining gap reflects how MotifMatch separates entity encoding from pair scoring. Each entity is encoded once and its factors are reused across partners, leaving only compact matching and readout for each pair.

Table 5: Runtime on the official split. Underlines mark the fastest baseline for each stage.

Model	Train (s)	Infer (s)
MotifMatch	**25.63**	**0.291**
BiLSTM	<u>189.26</u>	5.685
IUPAC n-gram	244.39	1.683
PMCf MF3_c	785.43	<u>1.607</u>
LectinOracle	1 252.04	5.865
GlycanAA	8 018.75	20.199
GNNGLY	18 644.19	296.400
SweetNet	21 704.77	288.931
GIFFLAR	35 965.35	358.596

Parameter sharing keeps cross-glycan supervision inexpensive. Reusing columns of W_G produces the coupling in Proposition 1, so ordinary updates affect glycans sharing a motif without comparing them pairwise. With n_P proteins, n_G glycans, n_{pair} candidate pairs, and readout width h, the joint branch has inference cost $O(n_P md + n_G Kd + n_{\text{pair}} dh)$.

5 DISCUSSION AND CONCLUSION

MotifMatch pools measurements across glycans through shared motif parameters and lets each protein use this evidence through protein-conditioned matching, achieving the best mean ranking scores across three cold-start settings. While glycan learning has largely advanced toward richer molecular descriptions, our results suggest that in cold-start settings with sparse pair-level supervision, the bottleneck may lie less in descriptive capacity than in how evidence is shared across molecules. A representation does more than describe individual molecules; it also implicitly specifies which measurements should inform one another, because the coordinates two glycans share determine how far a measurement on one trains the parameters used by the other. Explicit motifs and learned encoders are thus complementary; learned context built on shared motif coordinates could add information that motifs cannot express while preserving this coupling.

This view also has consequences for evaluation. A single fixed split misses how a representation uses new measurements: MotifMatch starts with the highest target error at zero support but finishes with the lowest at 16 supporting glycans. Comparing encoders under one concatenation head can likewise undervalue representations that benefit from multiplicative matching, as two representations similar under Concat separate under Hadamard. Support curves and comparisons across scoring functions would complement fixed-split accuracy in benchmarks.

The simplicity of counts also brings clear limitations. They retain linkage and terminal context but not motif arrangement, so glycans with identical counts but patterns on different branches or at different spacings are indistinguishable, although such arrangements can affect multivalent and branch-specific recognition. Motifs absent from the training measurements also lack direct supervision. Coordinates for topology between motifs and motif parameters composed from monosaccharide and linkage embeddings could address both issues while keeping one shared column per coordinate.

The fixed-budget results further suggest that, because shared coordinates determine which measurements support one another, the representation itself can guide which glycans to measure next. Because our experiment reassigns existing observations and the loss-change analysis uses unknown target labels, a prospective study will need to estimate support value before measurement and compare the resulting strategy with random and plain motif-coverage selection. More broadly, when measurements are scarce, what a pair-prediction model learns from each measurement depends not only on how it describes molecules but also on which molecules it lets share evidence.

Reproducibility Statement

Section 3 specifies the model and objective, and Appendix A provides the assumptions and derivations underlying the theoretical claims. Appendix B describes feature extraction, architecture, optimization, and computing hardware. Appendices C.1, C.3, and C.4 document evaluation metrics, data splits, random seeds, and support-transfer protocols; Appendix C.5 specifies the timing procedure. The GlycanML data are accessible through the official benchmark repository (Xu et al., 2025a). MotifMatch source code, dependency versions, data-preparation instructions, and training and evaluation commands are available at `https://anonymous.4open.science/r/Motifmatch-DFC2/`. The release provides the method implementation and tests; benchmark data and pretrained ESM2 weights are obtained from their original providers.

Ethics Statement

This study uses publicly available GlycanML data for computational research on protein–glycan binding prediction. It involves no new experiments with human participants or animals and uses no personally identifiable data. We acknowledge the original data and software providers and direct users to their resources and applicable terms. The predictions concern benchmark fluorescence responses, not validated physical affinities or clinical outcomes. Their reliability outside the evaluated setting depends on how well training measurements cover the relevant motifs, and experimental validation is needed before biological or biomedical use. Section 5 discusses what motif counts do not capture.

AI Use Statement

Generative AI tools were used to assist with method implementation and code writing, data processing, manuscript drafting and editing, and reference checking. They were not used to formulate research ideas or hypotheses, design the experiments, develop theoretical models, formulate or prove mathematical claims, interpret the results, or translate the manuscript. AI-assisted code was checked using automated tests and training and evaluation smoke runs, and bibliographic information was checked against source publications. The authors take full responsibility for the final manuscript, code, data processing, and scientific claims, including material produced with AI assistance.

References

Alhasan Alkuhlani, Walaa Gad, Mohamed Roushdy, and Abdel-Badeeh M. Salem. GNNGLY: Graph Neural Networks for Glycan Classification. *IEEE Access*, 11:51838–51847, 2023. doi: 10.1109/ACCESS.2023.3280123. URL `https://doi.org/10.1109/ACCESS.2023.3280123`.

Peizhen Bai, Filip Miljković, Bino John, and Haiping Lu. Interpretable bilinear attention network with domain adaptation improves drug-target prediction. *Nature Machine Intelligence*, 5(2): 126–136, 2023. doi: 10.1038/s42256-022-00605-1. URL `https://doi.org/10.1038/s42256-022-00605-1`.

Daniel Bojar, Rani K. Powers, Diogo M. Camacho, and James J. Collins. Deep-Learning Resources for Studying Glycan-Mediated Host-Microbe Interactions. *Cell Host & Microbe*, 29(1): 132–144.e3, 2021. doi: 10.1016/j.chom.2020.10.004. URL `https://doi.org/10.1016/j.chom.2020.10.004`.

Daniel Bojar, Lawrence Meche, Guanmin Meng, William Eng, David F. Smith, Richard D. Cummings, and Lara K. Mahal. A Useful Guide to Lectin Binding: Machine-Learning Directed Annotation of 57 Unique Lectin Specificities. *ACS Chemical Biology*, 17(11):2993–3012, 2022. doi: 10.1021/acschembio.1c00689. URL `https://doi.org/10.1021/acschembio.1c00689`.

Rebekka Burkholz, John Quackenbush, and Daniel Bojar. Using graph convolutional neural networks to learn a representation for glycans. *Cell Reports*, 35(11):109251, 2021. doi: 10.1016/j.celrep.2021.109251. URL `https://doi.org/10.1016/j.celrep.2021.109251`.

Eric J. Carpenter, Shaurya Seth, Noel Yue, Russell Greiner, and Ratmir Derda. GlyNet: a multi-task neural network for predicting protein–glycan interactions. *Chemical Science*, 13(22):6669–6686, 2022. doi: 10.1039/D1SC05681F. URL https://pubs.rsc.org/en/content/articlelanding/2022/sc/d1sc05681f.

Eric J. Carpenter, Chuanhao Peng, Simatsidk Haregu, Nicholas Twells, Logan Woudstra, Amika Sood, Jonathan Cartmell, Robert J. Woods, Lara K. Mahal, Sheng-Kai Wang, Russell Greiner, and Ratmir Derda. Atom-level machine learning of protein-glycan interactions and cross-chiral recognition in glycobiology. *Science Advances*, 11(49):eadx6373, 2025. doi: 10.1126/sciadv.adx6373. URL https://doi.org/10.1126/sciadv.adx6373.

Lachlan Coff, Jeffrey Chan, Paul A. Ramsland, and Andrew J. Guy. Identifying glycan motifs using a novel subtree mining approach. *BMC Bioinformatics*, 21:42, 2020. doi: 10.1186/s12859-020-3374-4. URL https://doi.org/10.1186/s12859-020-3374-4.

Bowen Dai, Daniel E. Mattox, and Chris Bailey-Kellogg. Attention please: modeling global and local context in glycan structure-function relationships. *bioRxiv*, 2021. doi: 10.1101/2021.10.15.464532. URL https://doi.org/10.1101/2021.10.15.464532.

Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. Multimodal Compact Bilinear Pooling for Visual Question Answering and Visual Grounding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 457–468, 2016. doi: 10.18653/v1/D16-1044. URL https://aclanthology.org/D16-1044/.

Kexin Huang, Cao Xiao, Lucas M. Glass, and Jimeng Sun. MolTrans: Molecular Interaction Transformer for drug–target interaction prediction. *Bioinformatics*, 37(6):830–836, 2021. doi: 10.1093/bioinformatics/btaa880. URL https://doi.org/10.1093/bioinformatics/btaa880.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural Tangent Kernel: Convergence and Generalization in Neural Networks. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://papers.neurips.cc/paper_files/paper/2018/hash/5a4be1fa34e62bb8a6ec6b91d2462f5a-Abstract.html.

Roman Joeres and Daniel Bojar. Higher-Order Message Passing for Glycan Representation Learning. *arXiv preprint arXiv:2409.13467v3*, 2025. URL https://arxiv.org/abs/2409.13467v3.

Jin-Hwa Kim, Kyoung-Woon On, Woosang Lim, Jeonghee Kim, Jung-Woo Ha, and Byoung-Tak Zhang. Hadamard Product for Low-rank Bilinear Pooling. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=r1rhWnZkg.

Zachary Klamer, Ben Staal, Anthony R. Prudden, Lin Liu, David F. Smith, Geert-Jan Boons, and Brian Haab. Mining High-Complexity Motifs in Glycans: A New Language To Uncover the Fine Specificities of Lectins and Glycosidases. *Analytical Chemistry*, 89(22):12342–12350, 2017. doi: 10.1021/acs.analchem.7b04293. URL https://doi.org/10.1021/acs.analchem.7b04293.

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

Zhun Liu, Ying Shen, Varun Bharadhwaj Lakshminarasimhan, Paul Pu Liang, AmirAli Bagher Zadeh, and Louis-Philippe Morency. Efficient Low-rank Multimodal Fusion With Modality-Specific Factors. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2247–2256, 2018. doi: 10.18653/v1/P18-1209. URL https://aclanthology.org/P18-1209/.

Jon Lundstrøm, Emma Korhonen, Frédérique Lisacek, and Daniel Bojar. LectinOracle: A Generalizable Deep Learning Model for Lectin–Glycan Binding Prediction. *Advanced Science*, 9

(1):2103807, 2022. doi: 10.1002/advs.202103807. URL https://doi.org/10.1002/advs.202103807.

Somesh Mohapatra, Joyce An, and Rafael Gómez-Bombarelli. Chemistry-informed macromolecule graph representation for similarity computation, unsupervised and supervised learning. *Machine Learning: Science and Technology*, 3(1):015028, 2022. doi: 10.1088/2632-2153/ac545e. URL https://doi.org/10.1088/2632-2153/ac545e.

Hakime Öztürk, Arzucan Özgür, and Elif Ozkirimli. DeepDTA: deep drug–target binding affinity prediction. *Bioinformatics*, 34(17):i821–i829, 2018. doi: 10.1093/bioinformatics/bty593. URL https://doi.org/10.1093/bioinformatics/bty593.

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating Training Data Influence by Tracing Gradient Descent. In *Advances in Neural Information Processing Systems*, volume 33, pp. 19920–19930, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/e6385d39ec9394f2f3a354d9d2b88eec-Abstract.html.

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

Maureen E. Taylor, Kurt Drickamer, Ronald L. Schnaar, Marilynn E. Etzler, and Ajit Varki. Discovery and Classification of Glycan-Binding Proteins. In Ajit Varki et al. (eds.), *Essentials of Glycobiology*, chapter 28. Cold Spring Harbor Laboratory Press, Cold Spring Harbor, NY, third edition, 2017. URL https://www.ncbi.nlm.nih.gov/books/NBK453061/.

Luc Thomès, Rebekka Burkholz, and Daniel Bojar. Glycowork: A Python package for glycan data science and machine learning. *Glycobiology*, 31(10):1240–1244, 2021. doi: 10.1093/glycob/cwab067. URL https://doi.org/10.1093/glycob/cwab067.

Minghao Xu, Yunteng Geng, Yihang Zhang, Ling Yang, Jian Tang, and Wentao Zhang. GlycanML: A Multi-Task and Multi-Structure Benchmark for Glycan Machine Learning. In *International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=owEQ0FTfVj.

Minghao Xu, Jiaze Song, Keming Wu, Xiangxin Zhou, Bin Cui, and Wentao Zhang. Modeling All-Atom Glycan Structures via Hierarchical Message Passing and Multi-Scale Pre-training. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 69675–69688, 2025b. URL https://proceedings.mlr.press/v267/xu25z.html.

A DERIVATIONS

We analyze the finite network defined in Section 3 at its current parameters, using deterministic predictions with dropout disabled. Derivatives are taken at differentiable points, and input sensitivities use the network's continuous extension in the count coordinates. The parameter block W_J selects motif columns only; the appended size and branching statistics are outside J.

A.1 PROTEIN-CONDITIONED STRUCTURAL RESPONSES

For a pair (p, g), write $c = c(g)$, $u = u_\theta(p)$, $\eta = W_G c + b_G$, and $z = u \odot \sigma_G(\eta)$. Let $h_z = \nabla_z H_\theta(z)$. The differential through the glycan projection and matching operation is

$$\mathrm{d}q = h_z^\top \operatorname{Diag}(u) \operatorname{Diag}(\sigma_G'(\eta))\, \mathrm{d}\eta = \delta^\top \mathrm{d}\eta, \tag{10}$$

where $\delta := \nabla_\eta q = \sigma_G'(\eta) \odot u \odot h_z$ is the joint-score sensitivity defined in Equation (7). For a perturbation of the count vector with parameters fixed, $\mathrm{d}\eta = W_G \mathrm{d}c$, giving

$$\nabla_c q = W_G^\top \delta, \qquad \frac{\partial q}{\partial c_j} = w_j^\top \delta. \tag{11}$$

For a perturbation of W_G with the input fixed, $\mathrm{d}\eta = (\mathrm{d}W_G)c$, so

$$\mathrm{d}q = \delta^\top (\mathrm{d}W_G)c = \langle \delta c^\top, \mathrm{d}W_G \rangle_F, \qquad \nabla_{W_G} q = \delta c^\top. \tag{12}$$

Joint-score sensitivity controls responses to motif counts and learning on their shared columns.

With dropout disabled, the readout takes the following explicit form:

$$H_\theta(z) = a_H^\top \sigma_H(Bz + b_H) + b_0. \tag{13}$$

Writing $v = \sigma_G(\eta)$, the hidden preactivation is equivalently $B_p v + b_H$, with $B_p = B \operatorname{Diag}(u)$ as in Section 3.2. Let $D_H(z)$ be the diagonal matrix of PReLU slopes at $Bz + b_H$. Then

$$h_z = B^\top D_H(z) a_H. \tag{14}$$

Equivalently, $\nabla_v q = B_p^\top D_H(z) a_H$, and $\delta = \sigma_G'(\eta) \odot \nabla_v q$. The hidden preactivations combine matched coordinates, so changing the match can change $D_H(z)$ and the effective readout weights. For training-time derivatives with a fixed dropout mask m, the same identities hold with a_H replaced by $a_H \odot m$, where m includes the dropout scaling.

A.2 CROSS-GLYCAN GRADIENT COUPLING

Parameter-gradient inner products underlie neural tangent kernels and gradient-based training-data influence analysis (Jacot et al., 2018; Pruthi et al., 2020). Here we evaluate this inner product on the shared motif parameter block W_J of the current finite network. For support i and target t, their shared-block gradients follow from Equation (12):

$$\nabla_{W_J} q_i = \delta_i c_{i,J}^\top, \qquad \nabla_{W_J} q_t = \delta_t c_{t,J}^\top. \tag{15}$$

Proof of Proposition 1. Using $\langle A, B \rangle_F = \operatorname{tr}(A^\top B)$,

$$\begin{aligned} \kappa_J(t, i) &= \operatorname{tr}\big(c_{t,J} \delta_t^\top \delta_i c_{i,J}^\top\big) \\ &= (\delta_t^\top \delta_i) \operatorname{tr}\big(c_{t,J} c_{i,J}^\top\big) \\ &= (c_{t,J}^\top c_{i,J})(\delta_t^\top \delta_i). \quad \square \end{aligned} \tag{16}$$

For $J = \{j\}$, the coupling through column w_j is $\kappa_j(t, i) = c_j(g_t) c_j(g_i) \delta_t^\top \delta_i$. Since motif counts are nonnegative, the structural term is positive exactly when the two glycans share at least one motif in J. Pairs without such overlap have zero coupling through W_J; among overlapping pairs, pair-conditioned response alignment sets the sign of the coupling.

The outer-product gradient follows from the shared count projection. For a general differentiable joint score $\widetilde{H}(u, v)$, it takes the form $\widetilde{\delta}_i c_i^\top$, where $\widetilde{\delta}_i = \sigma_G'(\eta_i) \odot \nabla_v \widetilde{H}(u_i, v_i)$. In MotifMatch, Hadamard matching specializes this sensitivity to $\delta_i = \sigma_G'(\eta_i) \odot u_i \odot \nabla_z H(z_i)$. The count projection specifies structural overlap; matching affects pair-conditioned response alignment.

A.3 Local loss change and fixed-budget replacement

The additive branches and intercept have parameters separate from W_G, so $\nabla_{W_J}\widehat{F}_i = \nabla_{W_J} q_i$. With $e_i = \widehat{F}_i - \widetilde{y}_i$ and $\ell_i = e_i^2$,

$$\nabla_{W_J}\ell_i = 2e_i\delta_i c_{i,J}^\top. \tag{17}$$

Consider the path $W_J(\epsilon) = W_J - \epsilon\nabla_{W_J}\ell_i$, with its direction evaluated at $\epsilon = 0$ and all other parameters held fixed. Its effect on the target prediction is

$$\left.\frac{\mathrm{d}\widehat{F}_t(W_J(\epsilon))}{\mathrm{d}\epsilon}\right|_0 = -\left\langle \nabla_{W_J} q_t, \nabla_{W_J}\ell_i\right\rangle_F = -2e_i\kappa_J(t,i). \tag{18}$$

Applying the chain rule to the squared target loss $\ell_t = e_t^2$ then gives

$$\left.\frac{\mathrm{d}\ell_t(W_J(\epsilon))}{\mathrm{d}\epsilon}\right|_0 = -\left\langle \nabla_{W_J}\ell_t, \nabla_{W_J}\ell_i\right\rangle_F = -4e_t e_i\kappa_J(t,i). \tag{19}$$

This gives the local descent condition used in Section 3.3. A strictly negative directional derivative ensures a loss decrease for all sufficiently small positive ϵ. If the target's PReLU activation states remain fixed over the step, its score is affine in W_J along this path, and the loss change is exactly

$$\ell_t(W_J(\epsilon)) - \ell_t(W_J) = -4\epsilon e_t e_i\kappa_J(t,i) + 4\epsilon^2 e_i^2\kappa_J(t,i)^2. \tag{20}$$

The first-order term specifies improvement alignment; the second-order term accounts for step size.

For an observation set $\mathcal{B}$ of size n, let $\mathcal{L}_\mathcal{B} = n^{-1}\sum_{i\in\mathcal{B}}\ell_i$. The same calculation gives

$$\left.\frac{\mathrm{d}}{\mathrm{d}\epsilon}\ell_t(W_J - \epsilon\nabla_{W_J}\mathcal{L}_\mathcal{B})\right|_0 = -\frac{4e_t}{n}\sum_{i\in\mathcal{B}} e_i\kappa_J(t,i). \tag{21}$$

To isolate a fixed-budget replacement at the same parameters, replace removed observations $\mathcal{N}$ by an equally sized support set $\mathcal{S}$. The change in the negative-gradient direction on W_J is

$$\Delta_J = -\frac{1}{n}\left(\sum_{i\in\mathcal{S}}\nabla_{W_J}\ell_i - \sum_{r\in\mathcal{N}}\nabla_{W_J}\ell_r\right). \tag{22}$$

The corresponding difference in the target loss's directional derivative is

$$\left\langle \nabla_{W_J}\ell_t, \Delta_J\right\rangle_F = -\frac{4e_t}{n}\left(\sum_{i\in\mathcal{S}} e_i\kappa_J(t,i) - \sum_{r\in\mathcal{N}} e_r\kappa_J(t,r)\right). \tag{23}$$

For a single motif j, removed observations on glycans lacking that motif have $c_j(g_r) = 0$ and contribute zero to the second sum. Added support contributes through its motif counts, residuals, and pair-conditioned response alignment with the target. Equation (23) expresses the change in the target-loss direction induced by the observation replacement used in Section 4.3.

A.4 Local bilinear structure of the joint score

Fix an activation region $\mathcal{R}$ for the two factor projections and the deterministic readout in Equation (13). Let D_P, D_G, and D_H denote the diagonal matrices of their PReLU slopes on this region. Then

$$\begin{aligned} u &= D_P(W_P x + b_P),\\ v &= D_G(W_G c + b_G),\\ H(z) &= a_H^\top D_H(Bz + b_H) + b_0. \end{aligned} \tag{24}$$

Define $\omega_\mathcal{R} = B^\top D_H a_H$, $Q_\mathcal{R} = D_P\,\mathrm{Diag}(\omega_\mathcal{R})D_G$, and $\tau_\mathcal{R} = a_H^\top D_H b_H + b_0$. Expanding this product separates the cross-input term from the linear and constant terms:

$$\begin{aligned} q(x,c) &= (W_P x + b_P)^\top Q_\mathcal{R}(W_G c + b_G) + \tau_\mathcal{R}\\ &= x^\top M_\mathcal{R} c + r_\mathcal{R}^\top x + s_\mathcal{R}^\top c + t_\mathcal{R}. \end{aligned} \tag{25}$$

The interaction matrix and the linear and constant coefficients are

$$\begin{aligned} M_{\mathcal{R}} &= W_P^\top Q_{\mathcal{R}} W_G, \quad r_{\mathcal{R}} = W_P^\top Q_{\mathcal{R}} b_G, \\ s_{\mathcal{R}} &= W_G^\top Q_{\mathcal{R}}^\top b_P, \quad t_{\mathcal{R}} = b_P^\top Q_{\mathcal{R}} b_G + \tau_{\mathcal{R}}. \end{aligned} \tag{26}$$

Since $Q_{\mathcal{R}} \in \mathbb{R}^{d\times d}$, $\operatorname{rank}(M_{\mathcal{R}}) \le d$. This is the local form of the implemented PReLU joint branch, extending the factorized bilinear form of multiplicative pooling (Kim et al., 2017). Protein, glycan, and readout states define the region and can alter the effective interaction matrix.

For the Concat ablation described in Appendix C.2, the deterministic joint score is

$$q_{\mathrm{cat}}(x, c) = o_C^\top \sigma_C(C_P u(x) + C_G v(c) + \beta_C) + o_{C0}. \tag{27}$$

Fixing the projection and readout activation states gives

$$q_{\mathrm{cat}}(x, c) = \alpha_{\mathcal{R}}^\top x + \beta_{\mathcal{R}}^\top c + \gamma_{\mathcal{R}}. \tag{28}$$

The mixed derivatives therefore differ within each fixed activation region:

$$\nabla_x \nabla_c^\top q = M_{\mathcal{R}}, \qquad \nabla_x \nabla_c^\top q_{\mathrm{cat}} = 0. \tag{29}$$

Concat forms cross-input dependence only through changes in activation states, whereas Hadamard matching also supplies an explicit cross-input term within a region. The nonlinear readout allows the effective weights on these factor products to change across regions.

B Implementation details

Computing environment. Experiments run on Ubuntu 22.04.5 LTS with two Intel Xeon Platinum 8336C CPUs at 2.30 GHz, totaling 64 physical cores and 128 logical threads, and approximately 256 GiB of system-visible RAM. The server has four NVIDIA L20 GPUs with 46,068 MiB per card; each training and timing run uses one of these GPUs.

Protein representation. We use frozen `esm2_t33_650M_UR50D` embeddings of dimension 1,280. Sequences exceeding the context limit use 1,022-residue windows with a 128-residue overlap. We omit special tokens, average overlapping features, and mean-pool the reconstructed sequence.

Glycan representation. We parse IUPAC strings with `glycowork` (Thomès et al., 2021) and enumerate monosaccharides, linked pairs, and local three-monosaccharide patterns formed by two linkages incident to a common monosaccharide. The three-monosaccharide patterns include chains and V-shaped configurations. Motif identities retain monosaccharide and linkage labels; chains use descendant-to-parent order where defined, and other local paths use a deterministic ordering of their two orientations. A separate terminal-context count is incremented when either endpoint is terminal. We append the monosaccharide count and the number of enumerated three-monosaccharide patterns whose central monosaccharide is branching. Raw counts form a 1,264-dimensional vector. The vocabulary is constructed from benchmark glycan structures without binding labels and covers all 927 glycans.

Network and centering. Both factor functions use a linear projection to $d = 256$ followed by PReLU. The joint readout uses a $256 \to 128$ linear layer, PReLU, dropout 0.2, and a scalar linear output. Each single-molecule branch uses input layer normalization, a linear projection to 128 dimensions, GELU, dropout 0.15, and a bias-free scalar readout. If the raw outputs are $a_{\mathrm{raw},\theta}$ and $b_{\mathrm{raw},\theta}$, centering over their respective training entities gives

$$\begin{aligned} a_\theta &= a_{\mathrm{raw},\theta} - \bar{a}_\theta, \qquad b_\theta = b_{\mathrm{raw},\theta} - \bar{b}_\theta, \\ \mu_{\mathrm{ref},\theta} &= \mu_{0,\theta} + \bar{a}_\theta + \bar{b}_\theta. \end{aligned} \tag{30}$$

The reference distributions for these averages are uniform over unique training proteins and unique training glycans. Centering preserves every total prediction. Beyond the frozen ESM2 encoder, the complete model contains 1,016,037 trainable parameters across its branches.

Optimization and feature reuse. Targets are standardized using training-set statistics. We optimize all branches jointly with Adam, learning rate 10^{-3}, weight decay 10^{-5}, and batches grouped by 32 proteins. Training runs for up to 50 epochs, with validation-NMSE early stopping and patience 7. Predictions are transformed to the original response scale for evaluation. Fixed ESM2 features and glycan motif vectors are cached before training; learned entity outputs are recomputed per optimization step and shared by the pairs evaluated in that step. At inference, entity factors can be computed once and reused across all associated pairs. Given the factors, constructing their product costs $O(d)$, followed by $O(dh)$ for a readout of hidden width h.

C Supplementary experiments

C.1 Evaluation protocols

The official benchmark clusters proteins by sequence similarity and allocates protein clusters in an 8:1:1 ratio; pairs inherit their protein partition. We use the supplied pair lists, run seeds 0, 1, and 2, and select checkpoints on validation data. All methods use frozen ESM2 final-layer features, cached after mean-pooling across the residues of each protein sequence.

For an evaluation set $\mathcal{E}$, let $\mathrm{MSE}_{\mathcal{E}} = |\mathcal{E}|^{-1} \sum_{i\in\mathcal{E}} (\widehat{y}_i - y_i)^2$ on the original response scale. On the official split, NMSE is $\mathrm{MSE}_{\mathcal{E}}/s^2_{\mathrm{train}}$, where s_{train} is the training-label standard deviation used for target normalization. On the additional entity-cluster splits, NMSE is

$$\frac{\mathrm{MSE}_{\mathcal{E}}}{|\mathcal{E}|^{-1} \sum_{i\in\mathcal{E}} (y_i - \bar{y}_{\mathrm{train}})^2},$$

where $\bar{y}_{\mathrm{train}}$ is the mean label of the Seen–Seen training set.

Global Spearman pools all evaluation pairs; per-protein Spearman averages within-protein correlations, using average ranks for ties. Protein Spearman and NDCG weight proteins equally. Groups with fewer than two observations or constant true or predicted responses are excluded from this correlation average; a dash denotes an undefined aggregate. For protein p, let $\mathcal{E}_p$ be its evaluated candidates, $T_p = \min(20, |\mathcal{E}_p|)$, and $r_{p,g} = y_{p,g} - \min_{g'\in\mathcal{E}_p} y_{p,g'}$. Let π_p and π^*_p order candidates by decreasing predicted and measured response, respectively. Then

$$\mathrm{DCG}_{20}(p,\pi) = \sum_{j=1}^{T_p} \frac{r_{p,\pi(j)}}{\log_2(j+1)}, \qquad \mathrm{NDCG}_{20}(p) = \frac{\mathrm{DCG}_{20}(p,\pi_p)}{\mathrm{DCG}_{20}(p,\pi^*_p)}.$$

We average over proteins with at least two observations and nonconstant measured responses. The shift makes relevance nonnegative, linear gains retain measured response differences, and the discount places more weight on earlier candidates. Normalizing by the ideal score makes each protein contribute on the same zero-to-one scale. For tied predictions, NDCG is averaged over the possible orderings, so a constant predictor generally has a positive NDCG; the constant-predictor values in Appendix C.3 provide the corresponding reference.

C.2 Comparison models and structural ablations

PMCf concatenates ESM2 features with the 1,385-dimensional MF3_c fingerprint from MCNet's public vocabulary, counting chirality-aware Morgan environments at radii 0–3. The two input blocks are separately L2-normalized, and the predictor uses two width-100 hidden layers with LeakyReLU and a linear scalar output. This adaptation uses protein features and omits concentration.

Additive only removes the joint branch; Joint branch only removes the two single-molecule branches. On the official split, Joint branch only reaches NMSE 0.9165 ± 0.0119, Global ρ 0.2940 ± 0.0057, Protein ρ 0.3467 ± 0.0045, and NDCG@20 0.4664 ± 0.0263, close to the full model; the additive terms reduce the seed SD of NDCG@20 from 0.026 to 0.004. Concat and Bilinear retain the two single-molecule branches and 256-dimensional factor projections, replacing the computation of the joint score. Concat applies a width-128 linear layer, PReLU, dropout 0.2, and a scalar linear output to the concatenated 512-dimensional vector. Bilinear uses a learned 256×256 matrix Λ and scalar bias λ_0 to produce $u(p)^\top \Lambda v(g) + \lambda_0$.

The representation–matching comparisons use the same protein features, input dimension, factor dimensions, optimization protocol, and observation splits across variants. Motif counts + Hadamard is MotifMatch itself, so Tables 1 and 4 report the same official benchmark runs for it. The two circular encodings share one vocabulary of typed neighborhoods at radii 0–3, with 1,264 coordinates from the unlabeled benchmark structures. Circular binary records whether each neighborhood occurs; Circular counts records its number of occurrences. The support intervention uses Circular binary; Circular counts controls multiplicity in fixed-data comparisons.

C.3 Entity-cluster splits and complete generalization results

Split construction. The official protein-cluster split has 442,396/58,887/63,364 training/validation/test pairs. The additional splits use all 564,647 pairs, with 1,388 unique protein sequences and 927 glycans. Protein grouping uses the MMseqs2 (Steinegger & Söding, 2017) hit graph with coverage 0.8 and coverage mode 0, connecting sequences for each retained alignment with positive sequence identity. Glycan grouping uses typed circular graph features of radii 0–3, connecting glycans at Jaccard similarity at least 0.80. Each connected component is allocated intact to training, validation, or test, using separate 70/15/15 allocations for protein and glycan groups. Edges between seen entities are further split 80/10/10, and models learn only from Seen–Seen training labels. Normalization is fitted on that training set, and early stopping uses macro NMSE across the four validation regimes. Split and model seeds are paired as $(0, 0)$, $(1, 1)$, and $(2, 2)$, so the tabulated standard deviations reflect both split and training variation. The motif vocabulary is fixed in advance from all 927 unlabeled glycan structures. Graph coverage is 927/927 for GlycanAA and 905/927 for GIFFLAR; the remaining 22 GIFFLAR structures use a fixed unknown-graph representation to retain the complete test set.

Results across regimes. The two reference models show what structural information contributes in each regime (Tables 6–8). Between seen entities, a two-way entity bias already raises NDCG@20 from 0.524 to 0.655, so much of the ranking signal comes from each partner's response tendency; structural models build on it to reach 0.70–0.77. Under glycan-cold evaluation, unseen glycans have no learned bias and the entity-bias model ranks no better than the global mean, so the rise in NDCG@20 from 0.333 to 0.52–0.57 comes entirely from glycan structure. MotifMatch has the highest Protein Spearman and NDCG@20 in all three regimes and the lowest NMSE in Seen–Seen and glycan-cold. In these two regimes, the atom-neighborhood fingerprint PMCf also ranks below the monosaccharide-level models, consistent with Table 1.

When both partners are unseen, every model's NMSE lies near the constant-mean reference, yet rankings stay above it: MotifMatch raises NDCG@20 from 0.393 to 0.475, with the smallest seed variation in both NMSE and NDCG@20. Models therefore recover part of each protein's candidate ordering even when response magnitudes for new pairs are hard to predict, supporting the emphasis on candidate ranking in the main text.

Table 6: Complete Seen–Seen reference results on the additional splits. †GIFFLAR uses a fixed unknown-graph representation for the 22 structures it cannot parse.

Model	NMSE ↓	Global ρ ↑	Protein ρ ↑	NDCG@20 ↑
Global mean	1.0000 ± 0.0000	–	–	0.5235 ± 0.0043
Two-way entity bias	0.9521 ± 0.0011	0.2013 ± 0.0060	0.2415 ± 0.0117	0.6554 ± 0.0014
IUPAC n-gram	0.7617 ± 0.0224	0.3441 ± 0.0186	0.3113 ± 0.0162	0.7551 ± 0.0168
BiLSTM	0.7951 ± 0.0158	0.3048 ± 0.0002	0.2966 ± 0.0101	0.7349 ± 0.0038
LectinOracle	0.7646 ± 0.0174	0.3353 ± 0.0081	0.3148 ± 0.0059	0.7637 ± 0.0062
GlycanAA	0.7898 ± 0.0313	0.3233 ± 0.0066	0.3074 ± 0.0009	0.7513 ± 0.0078
GIFFLAR†	0.7901 ± 0.0136	0.3146 ± 0.0155	0.3019 ± 0.0090	0.7540 ± 0.0059
PMCf MF3_c	0.8812 ± 0.0062	0.2238 ± 0.0045	0.2232 ± 0.0053	0.6967 ± 0.0034
MotifMatch	$\mathbf{0.7161 \pm 0.0168}$	$\mathbf{0.3561 \pm 0.0069}$	$\mathbf{0.3417 \pm 0.0044}$	$\mathbf{0.7742 \pm 0.0016}$

Table 7: Complete glycan-cold results on the additional splits. †GIFFLAR uses a fixed unknown-graph representation for the 22 structures it cannot parse.

Model	NMSE ↓	Global ρ ↑	Protein ρ ↑	NDCG@20 ↑
Global mean	1.0000 ± 0.0000	–	–	0.3329 ± 0.0124
Two-way entity bias	1.0132 ± 0.0018	-0.0711 ± 0.0213	–	0.3329 ± 0.0124
IUPAC n-gram	0.9443 ± 0.0068	0.1625 ± 0.0296	0.1816 ± 0.0190	0.5504 ± 0.0089
BiLSTM	0.9329 ± 0.0160	0.1781 ± 0.0316	0.1782 ± 0.0400	0.5357 ± 0.0160
LectinOracle	0.9291 ± 0.0170	0.1876 ± 0.0094	0.1902 ± 0.0205	0.5702 ± 0.0057
SweetNet	0.9206 ± 0.0141	0.1892 ± 0.0091	0.1887 ± 0.0189	0.5685 ± 0.0125
GNNGLY	0.9254 ± 0.0160	0.1841 ± 0.0250	0.1808 ± 0.0195	0.5520 ± 0.0082
GlycanAA	0.9423 ± 0.0054	0.1834 ± 0.0160	0.1753 ± 0.0201	0.5379 ± 0.0062
GIFFLAR†	0.9384 ± 0.0163	0.1935 ± 0.0149	0.1967 ± 0.0262	0.5649 ± 0.0144
PMCf MF3_c	0.9643 ± 0.0249	0.1406 ± 0.0194	0.1505 ± 0.0296	0.5186 ± 0.0072
MotifMatch	$\mathbf{0.9098 \pm 0.0325}$	$\mathbf{0.1969 \pm 0.0210}$	$\mathbf{0.1971 \pm 0.0328}$	$\mathbf{0.5723 \pm 0.0138}$

Table 8: Complete double-cold results on the additional splits. †GIFFLAR uses a fixed unknown-graph representation for the 22 structures it cannot parse.

Model	NMSE ↓	Global ρ ↑	Protein ρ ↑	NDCG@20 ↑
Global mean	1.0000 ± 0.0000	–	–	0.3928 ± 0.0201
Two-way entity bias	1.0000 ± 0.0000	–	–	0.3928 ± 0.0201
IUPAC n-gram	1.0060 ± 0.0077	0.0761 ± 0.0223	0.0854 ± 0.0268	0.4533 ± 0.0136
BiLSTM	$\mathbf{0.9971 \pm 0.0054}$	0.0712 ± 0.0346	0.0945 ± 0.0481	0.4486 ± 0.0086
LectinOracle	1.0276 ± 0.0108	0.0830 ± 0.0262	0.0919 ± 0.0254	0.4702 ± 0.0121
SweetNet	1.0215 ± 0.0159	0.0708 ± 0.0263	0.0894 ± 0.0277	0.4615 ± 0.0198
GNNGLY	1.0033 ± 0.0022	0.0649 ± 0.0347	0.0823 ± 0.0245	0.4667 ± 0.0117
GlycanAA	1.0033 ± 0.0059	0.0726 ± 0.0288	0.0890 ± 0.0307	0.4594 ± 0.0176
GIFFLAR†	1.0303 ± 0.0377	0.0694 ± 0.0228	0.0809 ± 0.0234	0.4554 ± 0.0169
PMCf MF3_c	0.9986 ± 0.0137	0.0629 ± 0.0459	0.0846 ± 0.0546	0.4657 ± 0.0175
MotifMatch	1.0030 ± 0.0007	$\mathbf{0.0848 \pm 0.0443}$	$\mathbf{0.0991 \pm 0.0488}$	$\mathbf{0.4751 \pm 0.0039}$

C.4 Support-transfer protocol and individual motifs

Protocol. All target-glycan observations are excluded from training. Support counts distinct training glycans containing the selected motif, with typed circular-feature Jaccard similarity below 0.80 to the targets. Support observations replace motif-negative observations within the same proteins; test controls are also protein-matched. Circular binary uses presence of typed neighborhoods at radii 0–3, retaining the 1,264 most frequent coordinates across all 927 unlabeled benchmark glycans.

For each motif and seed, target and control sets are fixed across support levels. Training-label normalization and the constant-mean reference use the zero-support training set. Target and control NMSE have separate denominators computed on their respective evaluation sets, as in Appendix C.1; these denominators remain fixed across k. Equation (9) uses each model's zero-support baseline.

Initial three-motif cohort. The initial experiment uses three motifs selected without binding labels and with sufficient cross-glycan support: Galβ1-4GlcNAcβ1-2Man (M1), Fucα1-2Gal (M2), and Manβ1-4GlcNAcβ1-4GlcNAc (M3). Four model variants, seventeen integer support levels from 0 to 16, and three seeds yield 612 training runs. Table 9 summarizes this cohort; Figure 3 shows the support curves averaged across motifs and the endpoint contrasts for each motif. The endpoint gains mirror the fixed-data comparison in Section 4.4. Replacing Circular binary with motif counts raises both gains under either scorer, and Hadamard more than doubles the target gain of motif counts, from 0.036 to 0.082, while leaving that of Circular binary essentially unchanged. How much Hadamard matching gains from support thus also depends on the structural units it matches.

Nine-motif extension. The extension evaluates nine additional motifs at $k = 0, 4, 16$ with seeds 0, 1, and 2. Across all twelve motifs, MotifMatch reduces mean target error from $k = 0$ to 16. For

Table 9: Initial three-motif experiment: endpoint gains from $k = 0$ to 16. Entries report mean $\pm$ sample SD across all nine combinations of three motifs and three training seeds.

Model	Target gain $G_T \uparrow$	Adjusted gain $G_{\text{adj}} \uparrow$
Motif counts + Hadamard	0.0815 ± 0.1030	0.0747 ± 0.1173
Motif counts + Concat	0.0359 ± 0.0689	0.0486 ± 0.0751
Circular binary + Hadamard	0.0153 ± 0.0799	0.0172 ± 0.0842
Circular binary + Concat	0.0155 ± 0.0491	0.0067 ± 0.0505

Table 3, we first average the nine motifs equally within each seed, then calculate the mean and sample SD over the three seeds. Table 11 summarizes three seeds per motif.

The extension distinguishes response to support from final prediction quality. Motif counts + Hadamard reaches the lowest mean target error at $k = 16$, and its target and adjusted gains are nearly equal, so its error reduction falls on the targets. For Motif counts + Concat, by contrast, the adjusted gain exceeds the target gain largely because control error increases. The advantage extends to ranking: at $k = 16$, Motif counts + Hadamard is best on all four metrics, with a Protein Spearman of 0.218 against at most 0.178 for the other variants (Table 10). Support therefore improves not only response error but also how the held-out glycans are ranked for each protein.

Table 10: Prediction at $k = 16$ on the nine additional motifs. For each model and metric, values average the nine per-motif means with equal weights across motifs.

Model	NMSE ↓	Global $\rho \uparrow$	Protein $\rho \uparrow$	NDCG@20 ↑
Circular binary + Concat	0.9477	0.1871	0.1650	0.7869
Circular binary + Hadamard	0.9465	0.1968	0.1578	0.7859
Motif counts + Concat	0.9442	0.2050	0.1776	0.7853
Motif counts + Hadamard	**0.9199**	**0.2377**	**0.2178**	**0.7990**

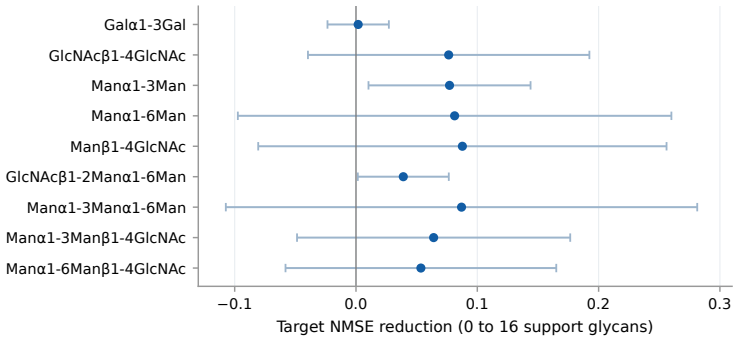


Figure 4: Target prediction gains on the nine additional motifs for Motif counts + Hadamard. Points are means and bars are sample SDs over three seeds. This cohort uses $k = 0, 4, 16$; the plotted gain is the reduction in target NMSE from $k = 0$ to 16.

Table 11: Motif counts + Hadamard on the nine additional motifs: target and control-adjusted gains from $k = 0$ to 16, reported as mean $\pm$ sample SD over three seeds.

Motif	Target gain $G_T \uparrow$	Adjusted gain $G_{\text{adj}} \uparrow$
Galα1-3Gal	0.0018 ± 0.0253	-0.0113 ± 0.0124
GlcNAcβ1-4GlcNAc	0.0764 ± 0.1160	0.0631 ± 0.1103
Manα1-3Man	0.0771 ± 0.0668	0.1000 ± 0.0377
Manα1-6Man	0.0813 ± 0.1787	0.1086 ± 0.1471
Manβ1-4GlcNAc	0.0877 ± 0.1683	0.0874 ± 0.1667
GlcNAcβ1-2Manα1-6Man	0.0390 ± 0.0375	0.0000 ± 0.0525
Manα1-3Manα1-6Man	0.0870 ± 0.1943	0.1192 ± 0.1694
Manα1-3Manβ1-4GlcNAc	0.0640 ± 0.1126	0.0819 ± 0.0852
Manα1-6Manβ1-4GlcNAc	0.0535 ± 0.1116	0.0376 ± 0.1662

Per-motif variation. Gains are positive for most individual motifs (Figure 4, Table 11). Eight of the nine motifs have target gains between 0.039 and 0.088; Galα1-3Gal is the only exception, with a gain near zero. The largest adjusted gains occur for mannose-containing core motifs such as Manα1-3Manα1-6Man, Manα1-6Man, and Manα1-3Man. Because every motif receives the same amount of support, the spread in gains is consistent with Proposition 1: transfer depends on response alignment between support and targets as well as on structural overlap.

C.5 Complete efficiency measurements

Timing uses one NVIDIA L20 on the Ubuntu server described in Appendix B, with identical resources and precached ESM2 features for all methods. Reported times cover training and prediction after protein feature extraction. Training time is wall-clock time including validation and early stopping, and inference covers all 63,364 test pairs. Parameter counts include trainable parameters only, and GPU memory is peak allocated memory recorded by PyTorch.

The speed advantage does not come from a smaller model (Table 12). MotifMatch has 1.02M trainable parameters, comparable to IUPAC n-gram and BiLSTM and more than LectinOracle and PMCf, yet it trains and infers faster than all of them, with peak memory below 0.4 GiB in both stages. The difference lies in how computation is organized: motif counts are extracted once and entity factors are reused across pairs, whereas graph encoders repeat message passing over glycan graphs in every batch.

Table 12: Complete efficiency results. Parameters are in millions; memory is in GiB.

Model	Params	Train (s)	Infer (s)	Train mem.	Infer mem.
IUPAC n-gram	1.017	244.39	1.683	0.235	0.158
BiLSTM	1.157	189.26	5.685	0.787	–
LectinOracle	0.691	1 252.04	5.865	0.300	0.041
SweetNet	5.010	21 704.77	288.931	0.308	0.099
GNNGLY	2.768	18 644.19	296.400	2.886	0.379
GlycanAA	15.927	8 018.75	20.199	1.332	0.694
GIFFLAR	4.514	35 965.35	358.596	2.926	0.375
PMCf MF3_c	0.277	785.43	1.607	0.248	0.191
MotifMatch	1.016	**25.63**	**0.291**	0.370	0.293

C.6 Complete representation–matching comparisons

These comparisons hold the runner and observation protocol fixed while varying the structural representation and joint-score formulation. Official protein-cold uses a fixed split and three training seeds; the additional regimes pair split and training seeds as in Appendix C.3. Motif counts + Hadamard is the complete MotifMatch model and serves as the reference for these comparisons.

Across all four regimes, Motif counts + Hadamard has the best mean on every metric among the seven variants (Table 13). The effect of Hadamard on circular features varies with the regime: it does not improve ranking with Circular binary on official protein-cold, but it raises the Protein Spearman of Circular binary from 0.171 to 0.186 under glycan-cold and from 0.072 to 0.080 under double-cold, and lowers its Seen–Seen NMSE from 0.774 to 0.747. In every regime, motif counts gain more from Hadamard than Circular binary does; multiplicative matching is thus broadly useful, while the size of its benefit depends on the representation it matches, consistent with Section 4.4.

Multiplicity alone does not explain this pattern: Circular counts and Circular binary perform similarly across regimes, without a consistent advantage from counting. The scalar Bilinear scorer approaches MotifMatch under double-cold evaluation but trails it on the official split, suggesting that an explicit cross-input term is especially useful when both partners are new, while a nonlinear readout over the matched vector keeps the benefit consistent across regimes.

Table 13: Matched representation–matching comparisons in all four regimes. Entries report mean ± sample SD over three runs; bold marks the best mean for each metric within a regime.

Representation	Matching	NMSE ↓	Global ρ ↑	Protein ρ ↑	NDCG@20 ↑
Official protein-cold					
Circular counts	Concat	0.9363 ± 0.0028	0.2913 ± 0.0022	0.3527 ± 0.0041	0.4233 ± 0.0135
Circular counts	Hadamard	0.9520 ± 0.0025	0.2922 ± 0.0085	0.3477 ± 0.0063	0.4220 ± 0.0037
Circular binary	Concat	0.9341 ± 0.0025	0.2920 ± 0.0016	0.3519 ± 0.0019	0.4259 ± 0.0243
Circular binary	Hadamard	0.9331 ± 0.0175	0.2975 ± 0.0029	0.3514 ± 0.0066	0.4195 ± 0.0179
Motif counts	Bilinear	0.9255 ± 0.0162	0.2796 ± 0.0115	0.3266 ± 0.0242	0.4525 ± 0.0225
Motif counts	Concat	0.9366 ± 0.0059	0.2930 ± 0.0083	0.3491 ± 0.0079	0.4330 ± 0.0075
Motif counts	Hadamard	$\mathbf{0.9160 \pm 0.0077}$	$\mathbf{0.2998 \pm 0.0090}$	$\mathbf{0.3534 \pm 0.0035}$	$\mathbf{0.4708 \pm 0.0035}$
Glycan-cold					
Circular counts	Concat	0.9244 ± 0.0113	0.1749 ± 0.0158	0.1549 ± 0.0230	0.5656 ± 0.0121
Circular counts	Hadamard	0.9145 ± 0.0072	0.1891 ± 0.0119	0.1802 ± 0.0138	0.5687 ± 0.0159
Circular binary	Concat	0.9243 ± 0.0092	0.1787 ± 0.0138	0.1708 ± 0.0162	0.5714 ± 0.0099
Circular binary	Hadamard	0.9116 ± 0.0107	0.1866 ± 0.0272	0.1863 ± 0.0314	0.5709 ± 0.0034
Motif counts	Bilinear	0.9236 ± 0.0240	0.1889 ± 0.0264	0.1932 ± 0.0301	0.5702 ± 0.0155
Motif counts	Concat	0.9230 ± 0.0218	0.1867 ± 0.0212	0.1763 ± 0.0293	0.5664 ± 0.0206
Motif counts	Hadamard	$\mathbf{0.9098 \pm 0.0325}$	$\mathbf{0.1969 \pm 0.0210}$	$\mathbf{0.1971 \pm 0.0328}$	$\mathbf{0.5723 \pm 0.0138}$
Double-cold					
Circular counts	Concat	1.0248 ± 0.0155	0.0461 ± 0.0260	0.0589 ± 0.0262	0.4491 ± 0.0251
Circular counts	Hadamard	1.0158 ± 0.0158	0.0650 ± 0.0180	0.0789 ± 0.0126	0.4531 ± 0.0215
Circular binary	Concat	1.0226 ± 0.0186	0.0534 ± 0.0178	0.0719 ± 0.0182	0.4528 ± 0.0241
Circular binary	Hadamard	1.0068 ± 0.0159	0.0717 ± 0.0251	0.0803 ± 0.0264	0.4615 ± 0.0242
Motif counts	Bilinear	1.0080 ± 0.0072	0.0775 ± 0.0443	0.0949 ± 0.0447	0.4658 ± 0.0049
Motif counts	Concat	1.0126 ± 0.0165	0.0650 ± 0.0459	0.0762 ± 0.0502	0.4700 ± 0.0095
Motif counts	Hadamard	$\mathbf{1.0030 \pm 0.0007}$	$\mathbf{0.0848 \pm 0.0443}$	$\mathbf{0.0991 \pm 0.0488}$	$\mathbf{0.4751 \pm 0.0039}$
Seen–Seen					
Circular counts	Concat	0.7714 ± 0.0180	0.3530 ± 0.0058	0.3333 ± 0.0029	0.7644 ± 0.0006
Circular counts	Hadamard	0.7317 ± 0.0199	0.3485 ± 0.0068	0.3404 ± 0.0052	0.7678 ± 0.0007
Circular binary	Concat	0.7738 ± 0.0190	0.3521 ± 0.0052	0.3344 ± 0.0039	0.7656 ± 0.0027
Circular binary	Hadamard	0.7468 ± 0.0364	0.3469 ± 0.0208	0.3352 ± 0.0163	0.7690 ± 0.0109
Motif counts	Bilinear	0.7599 ± 0.0105	0.3332 ± 0.0108	0.3199 ± 0.0089	0.7666 ± 0.0065
Motif counts	Concat	0.7782 ± 0.0263	0.3479 ± 0.0103	0.3327 ± 0.0044	0.7610 ± 0.0022
Motif counts	Hadamard	$\mathbf{0.7161 \pm 0.0168}$	$\mathbf{0.3561 \pm 0.0069}$	$\mathbf{0.3417 \pm 0.0044}$	$\mathbf{0.7742 \pm 0.0016}$