GCMATCH: PROTEIN POCKET MATCHING WITH GEOMETRIC–CHEMICAL SPATIAL SEMANTIC MODELING AND FINE-GRAINED REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Protein pocket similarity matching supports drug repurposing and pocket function annotation. This task requires similarity in both cavity geometry and chemical environment, i.e., dual similarity. Existing learning-based methods typically encode geometric and chemical information jointly within the same local unit. They produce single-vector pocket representations for matching, enabling effective assessment of overall similarity. However, they do not explicitly model spatial semantic relations between cavity geometry and chemical environment, and the coarse-grained single-vector representation obscures local geometric or chemical differences. Moreover, the datasets used by these methods define similarity labels according to whether pockets bind similar ligands, but ligand similarity is not equivalent to dual similarity. Together, these issues limit the accuracy of dual-similarity pocket matching. We propose GCMatch, a dual-similarity pocket matching method based on spatial semantic modeling and fine-grained representations. First, GCMatch extracts cavity geometric points and chemical functional points from each pocket. It then connects both point types in a single neighborhood relation graph. A graph encoder learns two sets of fine-grained node representations, one for geometry and one for chemistry. Multi-vector matching is applied separately to each set, and product fusion produces the final matching score. GCMatch achieves a test AUC of 0.920, exceeding the best baseline by 5.1 percentage points. We also construct a four-quadrant dataset with separate geometric and chemical similarity labels for each pocket pair. It serves as a unified evaluation benchmark for the field.

1 INTRODUCTION

Protein pocket similarity matching supports drug repurposing and pocket function annotation (Simonovsky & Meyers, 2020; Ehrt et al., 2018; Reim et al., 2024). A pocket is characterized by its cavity geometry and chemical environment. Cavity geometry includes shape, size, and boundary, while the chemical environment includes the types of chemical groups facing the cavity and their spatial distribution (Yeturu & Chandra, 2008; Kahraman et al., 2007; 2010). Pocket matching therefore requires similarity in both aspects, i.e., dual similarity. Classical methods assess dual similarity using manually defined structural and chemical features (Schmitt et al., 2002; Shulman-Peleg et al., 2004; Konc & Janezic, 2010; Gao & Skolnick, 2013; Chartier & Najmanovich, 2015). These methods rely on feature engineering and cannot learn from data, which limits their ability to capture the rich semantic information in pockets.

Learning-based methods can learn from data and capture information that is difficult to represent with handcrafted features. DeeplyTough maps pocket atoms onto a voxel grid and learns voxel representations through 3D convolutions, with cavity geometry reflected only indirectly by the atomic distribution (Simonovsky & Meyers, 2020). BindSiteS-CNN projects chemical properties onto cavity surface points and learns surface-point representations through spherical convolutions, combining geometric and chemical information at the same sampled points (Scott et al., 2022). Methods such as ProFSA and OneProt use atoms or residues as local units and learn spatial relations between them through Transformers or graph neural networks, but cavity geometry itself is not an encoded unit (Zhou et al., 2023; Gao et al., 2024; Flöge et al., 2025). These methods either represent cav-

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.

ity geometry implicitly or combine it with chemistry, without explicitly modeling spatial semantic relations between cavity geometry and chemical environment. Moreover, they all produce a single coarse-grained vector per pocket, obscuring local geometric or chemical differences. At the data level, the datasets these methods use for pocket-matching training and evaluation primarily define labels by whether pockets bind similar ligands (Govindaraj & Brylinski, 2018; Chen et al., 2016). Such pockets may nevertheless differ in geometry or chemistry (Kahraman et al., 2007; Barelier et al., 2015), so these labels are not equivalent to dual similarity. Therefore, existing learning-based methods have limited accuracy in dual-similarity pocket matching.

To address the aforementioned limitations, we propose GCMatch, a pocket matching method based on fine-grained spatial semantic modeling. GCMatch represents local cavity geometry and chemical groups as two types of spatial points, and it builds a neighborhood relation graph to model their spatial semantic relations. A graph encoder learns fine-grained semantic representations of both point types, and multi-vector matching produces separate geometric and chemical similarity scores. The encoder is jointly trained with supervision on the geometric and chemical matching scores and the final similarity. We also construct a four-quadrant dataset of pocket pairs based on geometric and chemical dual similarity. This dataset addresses the reliance on ligand-similarity labels in the datasets used by prior methods, as such labels cannot support training and evaluation targeting dual similarity. Section 4.1 describes its construction and annotation.

Our main contributions are as follows.

1. We propose a geometry–chemistry spatial semantic modeling method for pockets. The method treats local cavity geometry and chemical groups as two types of spatial points. It then builds a neighborhood relation graph from their relative positions and orientations, providing a foundation for fine-grained pocket representations.
2. We propose a method for geometric and chemical dual-similarity pocket matching based on the neighborhood relation graph. A graph encoder learns two sets of fine-grained node representations from the graph, one for geometry and one for chemistry. Multi-vector matching produces separate geometric and chemical similarity scores, which are multiplied to obtain the final similarity and improve the accuracy of dual-similarity pocket matching.
3. We construct a four-quadrant dataset for dual-similarity pocket matching. It includes pocket pairs that are similar in geometry alone, chemistry alone, both geometry and chemistry, or neither. Pairs similar in both geometry and chemistry are treated as positive samples, and the other three categories as negative samples. The dataset provides a benchmark for further research in this field.

2 Related Work

2.1 Classical pocket matching methods

Classical methods fall into two categories: alignment-free and alignment-based. Alignment-free methods compress a pocket into a descriptor and compare descriptors directly; PocketMatch and RAPMAD, for instance, use distributions of pairwise point distances grouped by chemical type as descriptors, losing the orientation and provenance of point pairs (Yeturu & Chandra, 2008; Krotzky et al., 2015). Alignment-based methods first find correspondences between residues or feature points across pockets, then score the paired points. APoc finds residue correspondences via dynamic and integer programming, combining backbone geometry, side-chain orientation, and chemical similarity into a PS-score (Gao & Skolnick, 2013); G-LoSA represents pockets with chemical feature points, establishes correspondences via clique search, and scores them with a distance score weighted by chemical type (Lee & Im, 2016); SiteMotif establishes correspondences at the residue level, giving separate structural matching and BLOSUM substitution scores (Sankar & Chandra, 2022). In these methods, both descriptors and scoring rules are hand-designed, limiting the pocket information they can express and making them hard to optimize against task-specific data.

2.2 Learning-based pocket matching methods

Learning-based methods optimize pocket representations through data-driven training and fall into two categories by representation form. The first represents pockets with regularly sampled voxels

or surface points. For example, DeeplyTough maps pocket atoms and their chemical properties onto a 3D voxel grid, encodes them with 3D convolutions into a fixed-length descriptor, and performs metric learning on pocket pairs defined by ligand similarity, so cavity geometry is only implicitly reflected through atom-free voxels (Simonovsky & Meyers, 2020); BindSiteS-CNN represents geometry with a negative-image surface of the cavity, projects the physicochemical properties of lining residues onto surface points, and learns representations with spherical convolutions, mixing geometry and chemistry within the same sampled points (Scott et al., 2022).

The second category represents pockets with atoms or residues as the unit, so the cavity itself is not the encoded unit. These methods are often pretrained on large-scale structural data and then fine-tuned on pocket-matching datasets. For example, Uni-Mol takes atom types and 3D coordinates as input and learns representations with a Transformer that encodes atom-pair distances. The model is pretrained through masked atom prediction and coordinate denoising (Zhou et al., 2023); ProFSA adopts the Uni-Mol pocket encoder and performs contrastive pretraining by constructing pseudo ligand–pocket pairs from protein fragments and their surrounding environment (Gao et al., 2024); OneProt builds a graph with residues as nodes, encodes spatial relations among residues with a graph neural network, and aligns pocket representations with those of other modalities through sequence-centric contrastive learning (Flöge et al., 2025); CoSP uses atoms as nodes, encodes them with geometric message passing, and performs contrastive learning against ligands (Gao et al., 2023).

None of these learning-based methods explicitly model the spatial semantic relations between cavity geometry and chemical environment. Each of these methods also outputs a single vector for the whole pocket, which obscures local geometric or chemical differences and prevents the similarity of the two aspects from being constrained separately. This limits the accuracy of dual-similarity matching.

2.3 Pocket matching datasets

Existing benchmarks primarily use ligand relationships to label similarity between different pockets. TOUGH-M1 defines positive and negative samples by the chemical similarity of the ligands bound by two pockets (Govindaraj & Brylinski, 2018); Vertex defines them by whether two targets bind the same ligand (Chen et al., 2016); in ProSPECCTs, the subsets assessing whether different protein pockets are similar likewise take binding the same or similar ligands as positive examples, while the remaining subsets use different definitions, conformations, or artificially mutated versions of the same pocket to test comparison methods' robustness and sensitivity, without addressing similarity judgments between different pockets (Ehrt et al., 2018). However, pockets that bind similar ligands may still differ markedly in cavity shape, chemical environment, and key recognition residues (Kahraman et al., 2007; 2010; Barelier et al., 2015), so such labels are not equivalent to dual similarity. As a result, these datasets do not directly support model training and evaluation for dual-similarity pocket matching.

3 Method

Problem definition. We study geometry–chemistry dual-similarity pocket matching. Given two protein binding pockets A and B, we determine whether they are simultaneously similar in cavity geometry and chemical environment. The geometric label $G(A, B) \in \{0, 1\}$ indicates whether the cavity geometries of the two pockets are similar, and the chemical label $C(A, B) \in \{0, 1\}$ indicates whether their chemical environments are similar. The dual-similarity label is defined as $Y(A, B) = G(A, B)\, C(A, B)$, i.e., a pocket pair is regarded as a match only when both aspects are similar. Our goal is to learn a matching function $s(A, B)$ that assigns high scores to dual-similar pocket pairs and low scores to others.

Overview. The overall framework of GCMatch is shown in Figure 1. First, cavity geometric points and chemical functional points are extracted from the pocket structure (3.1). Next, a neighborhood relation graph is built from the two point types, characterizing the spatial semantic relations between them via inter-point distance and direction (3.2). Finally, a graph encoder learns two sets of fine-grained node representations, one for geometry and one for chemistry. Multi-vector matching is performed separately on each set, and the two similarity scores are combined via product fusion to obtain the pocket pair's final similarity (3.3).

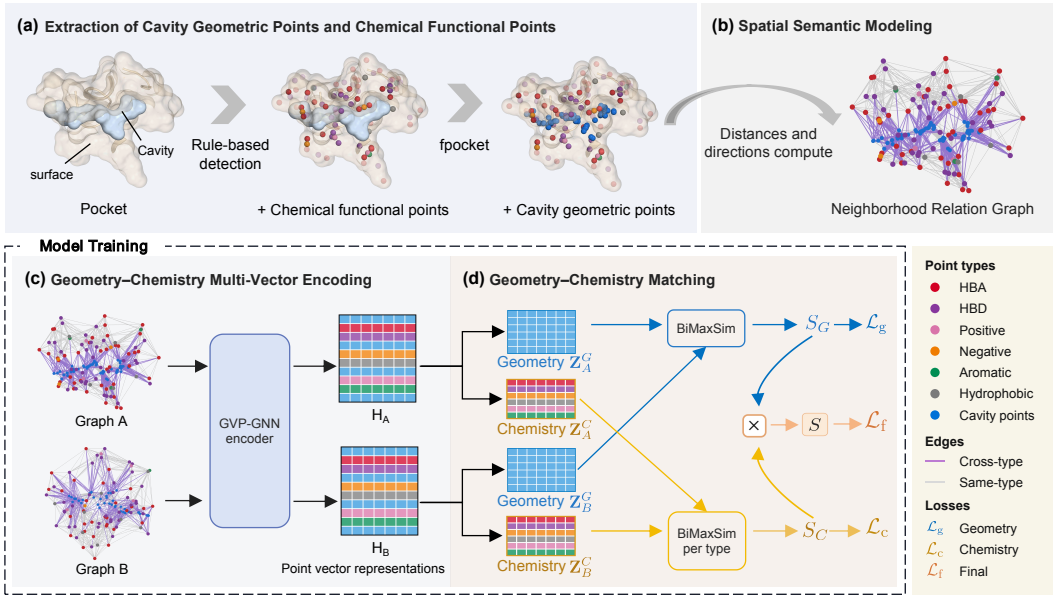

Figure 1: Overall architecture of GCMatch. Each row of the matrices $\mathbf{H}$ and $\mathbf{Z}$ illustrates one node vector.

3.1 Extraction of Cavity Geometric Points and Chemical Functional Points

To build the neighborhood relation graph, we extract a set of chemical functional points $\mathcal{V}_P^C$ and a set of cavity geometric points $\mathcal{V}_P^G$ from pocket P, as shown in Figure 1(a).

Chemical functional points. These points are extracted from residue chemical templates and fall into six categories: hydrogen-bond acceptor (HBA), hydrogen-bond donor (HBD), positive charge, negative charge, aromatic, and hydrophobic. Atom-type functional points use the coordinates of the corresponding atom, group-type functional points use the geometric center of the relevant atoms, and hydrophobic points are represented by the center of the spatially connected cluster of selected side-chain atoms. The detailed construction rules are given in Appendix A.1.

Cavity geometric points. These points come from the alpha spheres in the cavity detected by fpocket. We take the sphere center as the point coordinate and use the sphere radius r_i to describe the local cavity scale at that location (Le Guilloux et al., 2009).

3.2 Spatial Semantic Modeling of Cavity Geometric Points and Chemical Functional Points

To characterize the spatial relations between cavity geometry and chemical environment, we construct a neighborhood relation graph $\mathcal{G}_P = (\mathcal{V}_P, \mathcal{E}_P)$ from the cavity geometric points and chemical functional points, $\mathcal{V}_P = \mathcal{V}_P^G \cup \mathcal{V}_P^C$, as shown in Figure 1(b), where $\mathcal{V}_P$ and $\mathcal{E}_P$ denote the node set and edge set, respectively. Let N be the total number of nodes. Each node is connected to its 16 spatially nearest neighbors, retaining both same-type and cross-type edges.

Node features. Geometric points describe cavity morphology via the sphere radius, the standard deviation of cavity geometric points along the three principal directions, the position relative to the centroid, and the distance to the nearest chemical point. The three principal directions come from principal component analysis (PCA) on the geometric points' coordinates, and the standard deviation reflects the width of their distribution along each direction. Each geometric point also has a directional feature, a unit vector pointing toward the cavity centroid. Chemical points describe chemical properties via a multi-hot encoding of the six functional types and a backbone flag, and are represented directionally by the donor direction, the acceptor direction axis, and the aromatic ring normal vector.

Both node types adopt a unified input format of scalar and vector features. Feature dimensions not defined for a given node type are set to zero, and a node-type flag indicates the point's category. Detailed feature computations are given in Appendix A.2.

Edge features. For an edge $j \to i$, we describe the neighborhood relation via the inter-point distance and the unit relative direction

$$d_{ji} = \|\mathbf{x}_i - \mathbf{x}_j\|_2, \qquad \mathbf{e}_{v,ji} = \frac{\mathbf{x}_i - \mathbf{x}_j}{\max(d_{ji}, 10^{-6})} \tag{1}$$

To help the model learn neighborhood relations across distance ranges, we use 16 Gaussian radial basis functions (RBFs) to smoothly encode the inter-point distance d_{ji} into a 16-dimensional feature vector:

$$[\mathbf{e}_{s,ji}]_r = \exp\left(-\frac{(d_{ji} - \mu_r)^2}{2\sigma^2}\right), \qquad r = 1, \ldots, 16 \tag{2}$$

Here, the centers μ_r are equally spaced within 0–15 Å, with bandwidth $\sigma = 15/16$ Å.

3.3 A GRAPH-NEURAL-NETWORK-BASED METHOD FOR FINE-GRAINED GEOMETRY–CHEMISTRY MATCHING

To preserve fine-grained information, we first learn fine-grained node representations from the neighborhood relation graph, then compute the geometric and chemical matching scores separately, and finally combine them via product fusion to obtain the final similarity.

Graph encoding and fine-grained node representations. As shown in Figure 1(c), we adopt GVP-GNN as the shared encoder for the two pockets. It jointly processes scalar and 3D vector features and maintains geometric consistency of directional information (Jing et al., 2021). The encoder fuses node attributes with the distance and direction of neighboring edges. After three layers of message passing, it outputs a 64-dimensional representation vector for each node, forming $\mathbf{H} \in \mathbb{R}^{N\times 64}$. The representations of geometric and chemical points are extracted from $\mathbf{H}$ by node type, which gives $\mathbf{Z}^G$ and $\mathbf{Z}^C$. Each row corresponds to one node, so fine-grained information is preserved. For pockets A and B, we obtain $(\mathbf{Z}_A^G, \mathbf{Z}_A^C)$ and $(\mathbf{Z}_B^G, \mathbf{Z}_B^C)$, respectively, which serve as the input to multi-vector matching.

Branch multi-vector matching. As shown in Figure 1(d), the geometric branch compares all geometric points, while the chemical branch compares only points sharing the same functional type. A point possessing multiple types participates in the matching of each type it belongs to. The similarity matrices of the two branches are

$$\mathbf{M}_G = \mathrm{ReLU}\big(\mathbf{Z}_A^G(\mathbf{Z}_B^G)^\top\big), \qquad \mathbf{M}_{C,t} = \mathrm{ReLU}\big(\mathbf{Z}_{A,t}^C(\mathbf{Z}_{B,t}^C)^\top\big) \tag{3}$$

Here, t denotes one of the six chemical functional types from Section 3.1, and $\mathbf{Z}_{A,t}^C$ and $\mathbf{Z}_{B,t}^C$ are the type-t chemical-point vector matrices for pockets A and B, respectively. The rows and columns of the matrix correspond to the nodes of A and B, respectively. Since the node representations are L_2-normalized, their inner product is the cosine similarity. Zeroing out negative values with ReLU confines the local similarity to $[0, 1]$.

From these similarity matrices, we adopt MaxSim for multi-vector matching, finding each node's best match on the other side and aggregating the matching scores (Khattab & Zaharia, 2020). In a single direction, a high score can be obtained if most nodes of pocket A match well to a subset of nodes in B, while the remaining nodes in B may have only low matching similarity. We therefore adopt bidirectional MaxSim (BiMaxSim), which computes the mean best matching score in both the A-to-B and B-to-A directions and takes the smaller of the two, so that the direction with poorer matching constrains the branch score. For $\mathbf{M} \in [0, 1]^{n\times m}$, we define

$$\mathrm{BiMaxSim}(\mathbf{M}) = \min\left\{ \frac{1}{n}\sum_{i=1}^{n} \max_{1\le j\le m} M_{ij}, \quad \frac{1}{m}\sum_{j=1}^{m} \max_{1\le i\le n} M_{ij} \right\} \tag{4}$$

Here, n and m are the numbers of participating nodes on the two sides. This operation allows many-to-one matching, and the score is zero when either side has no nodes. The geometric and chemical matching scores are, respectively,

$$S_G = \mathrm{BiMaxSim}(\mathbf{M}_G), \qquad S_C = \frac{1}{|\mathcal{T}_{AB}|}\sum_{t\in\mathcal{T}_{AB}} \mathrm{BiMaxSim}(\mathbf{M}_{C,t}) \tag{5}$$

Here, $\mathcal{T}_{AB}$ is the set of chemical types present in at least one of the two pockets. A type present in only one pocket contributes a score of zero to the average, while types absent from both sides are excluded from the average.

Product fusion. To jointly constrain the final matching score by geometric and chemical similarity, we multiply the two branch scores to define the matching function $s(A, B)$. Since $S_G(A, B), S_C(A, B) \in [0, 1]$, we have

$$s(A, B) = S_G(A, B)\, S_C(A, B) \le \min\{S_G(A, B), S_C(A, B)\} \tag{6}$$

Hence the final score cannot exceed the smaller branch score, which limits how much one branch can compensate for the other. This scoring satisfies $s(A, B) = s(B, A)$.

Branch supervision and joint training. To constrain the two branches separately as well as the final dual-similarity decision, we supervise the geometric, chemical, and final scores simultaneously. For each training pair (A, B), the labels $G(A, B)$, $C(A, B)$, and $Y(A, B)$ supervise S_G, S_C, and s through $\mathcal{L}_{\mathrm{g}} = \mathrm{BCE}(S_G, G)$, $\mathcal{L}_{\mathrm{c}} = \mathrm{BCE}(S_C, C)$, and $\mathcal{L}_{\mathrm{f}} = \mathrm{BCE}(s, Y)$, where BCE is binary cross-entropy. The three losses are summed with equal weight, giving the joint objective

$$\mathcal{L}(\theta) = \mathbb{E}_{(A,B)\sim\mathcal{D}_{\mathrm{tr}}}\left[\mathcal{L}_{\mathrm{f}} + \mathcal{L}_{\mathrm{g}} + \mathcal{L}_{\mathrm{c}}\right] \tag{7}$$

where θ denotes the encoder parameters and $\mathcal{D}_{\mathrm{tr}}$ is the training set. The three losses jointly optimize the shared encoder. At matching time, only the final similarity s is output. Gradients reach the encoder through both the local matching and the product fusion.

4 Experiments

To evaluate GCMatch's geometry–chemistry dual-similarity matching capability, we compare it against classical and learning-based methods on the four-quadrant dataset and examine each model component's role through six ablations. We also compare end-to-end matching time of each method on a batch pocket-matching task.

4.1 Dataset construction

To evaluate geometry–chemistry dual-similarity pocket matching, we construct a four-quadrant pocket-pair dataset based on the PLINDER database (Durairaj et al., 2024).

First, a pocket is defined jointly by residues within a 6 Å neighborhood of the ligand and interacting residues identified by PLIP (Salentin et al., 2015; Adasme et al., 2021). Related structures are then grouped by sequence homology (Steinegger & Söding, 2017) and PDB origin, and training, validation, and test sets are split at the group level to prevent related structures from appearing across sets.

Next, within each split we annotate geometric and chemical labels for candidate pocket pairs separately. Geometric labels come from PPS-align matching results (Hu et al., n.d.). As no general tool directly annotates pocket chemical-environment similarity and interaction-based comparison

depends strongly on the bound ligand (Ehrt et al., 2018), chemical labels use ECFP4 Tanimoto similarity of the bound ligands as a proxy (Rogers & Hahn, 2010). Unlike benchmarks that define pocket similarity entirely by ligand similarity, this proxy labels only the chemical aspect, while geometric similarity comes independently from structural alignment. Using preset thresholds, we keep only pocket pairs for which both aspects give an unambiguous judgment and obtain four candidate categories: geometrically and chemically similar (POS), chemically similar only (HARD-G), geometrically similar only (HARD-C), and similar in neither (EASY). POS pairs are positive examples, and the other three categories are negative. Criteria and thresholds are given in Appendix B.1.

We further sample from the four candidate categories to reduce the influence of other differences on evaluation. Since POS and HARD-G are both chemically similar, we match their chemical-similarity distributions as closely as possible to highlight geometric differences; likewise, we match the geometric-similarity distributions of POS and HARD-C to highlight chemical differences. Pocket size is also controlled during sampling. This yields a final set of 75,200 pairs, with 60,000, 7,200, and 8,000 in the training, validation, and test sets, respectively, each containing equal numbers of the four categories. Sampling strategies and controlled metrics are given in Appendix B.2.

Finally, to check whether the two hard-negative types show the expected differences under other representations, we recompute similarity using three ligand chemical fingerprints and two pocket backbone geometric representations not used in annotation or sampling. Across all three splits, HARD-G similarity is consistently higher than HARD-C under the chemical representations, and HARD-C similarity is consistently higher than HARD-G under the geometric representations, consistent with label expectations. Definitions of each representation and complete results are given in Appendix B.3.

4.2 Experimental setup

Baseline methods. Using their official codebases, we run three classical methods, namely APoc, SiteMotif, and G-LoSA, and four learning-based methods, namely DeeplyTough, BindSiteS-CNN, ProFSA, and OneProt. DeeplyTough and BindSiteS-CNN are trained from scratch; ProFSA and OneProt are each fine-tuned.

Evaluation metrics. AUC is the primary metric. Taking POS as the positive class and the other three categories as the negative class, we compute Full-AUC on the complete test set to assess overall discriminative ability. We also form three evaluation subsets, each consisting of POS and one negative category (EASY, HARD-G, or HARD-C), with POS as the positive class. The corresponding AUCs are denoted Easy, G, and C. G and C evaluate the model's ability to discriminate geometrically and chemically dissimilar hard negatives, respectively, and their mean summarizes performance on the two hard-negative types.

Training and evaluation protocol. Each learning-based method is trained for 50 epochs on the four-quadrant dataset, with the best weights selected by validation-set Full-AUC and fixed for test-set evaluation; classical methods are evaluated directly on the test set. GCMatch and all ablation variants are optimized with AdamW (Loshchilov & Hutter, 2019) at a constant learning rate of 10^{-3} and a weight decay of 10^{-4}, using dynamic batching with a budget of 9,000 nodes per batch. Baseline settings are given in Appendix C. Matching time is reported as mean and sample standard deviation over three cold-start runs; the timing protocol is given in Appendix D.

4.3 Main results

Table 1 compares GCMatch with the baseline methods in terms of AUC on the complete test set and the three evaluation subsets.

GCMatch achieves the best result in every setting. Its Full-AUC reaches 0.920, which is 5.1 percentage points higher than the 0.869 of the strongest baseline, OneProt. When distinguishing POS from HARD-G, whose pairs are dissimilar only in geometry, and from HARD-C, whose pairs are dissimilar only in chemistry, GCMatch attains AUCs of 0.914 and 0.895, respectively. Its G/C mean AUC of 0.905 is higher than that of all baselines, and the second-best method, OneProt, reaches 0.854. The three classical methods achieve AUCs of 0.898–0.911 on HARD-G, close to GCMatch, but only 0.619–0.652 on HARD-C. This indicates that their discrimination relies more heavily on geometric information and is not sufficiently sensitive to chemical differences, which causes a marked

Table 1: AUC of each method on the test set.

Method	Full ↑	Easy ↑	G ↑	C ↑	G/C mean ↑
ProFSA	0.758	0.869	0.669	0.736	0.703
BindSiteS-CNN	0.773	0.843	0.729	0.747	0.738
DeeplyTough	0.775	0.835	0.706	0.784	0.745
G-LoSA	0.817	0.915	0.898	0.639	0.769
SiteMotif	0.822	0.915	0.898	0.652	0.775
APoc	0.822	0.935	0.911	0.619	0.765
OneProt	0.869	0.898	0.878	0.830	0.854
GCMatch	**0.920**	**0.950**	**0.914**	**0.895**	**0.905**

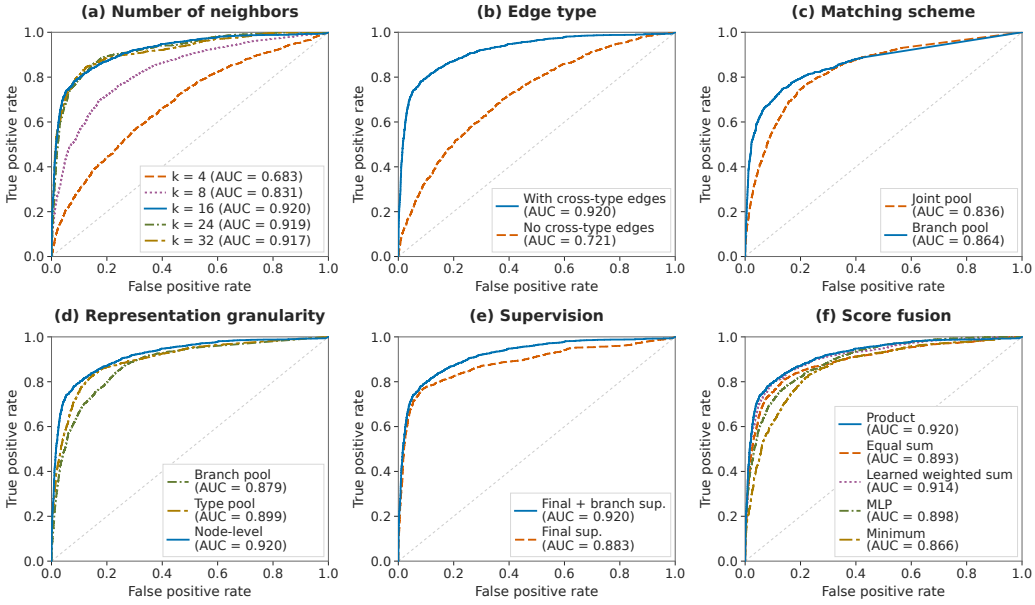

Figure 2: Test-set ROC curves for the six ablation experiments. The legend reports Full-AUC. The solid blue line marks the configuration with the highest Full-AUC in each subplot, and the dashed gray line marks the random baseline. "Final sup." denotes final-similarity supervision, and "branch sup." denotes geometric and chemical branch supervision. In (c), both configurations use only final supervision; in (d), all three use both final and branch supervision.

drop on HARD-C. Beyond AUC, GCMatch also achieves the best overall performance on the other classification metrics. Complete results are given in Table 9 of Appendix E.

4.4 Ablation Experiments

We examine the effects of the number of neighbors, edge type, matching scheme, representation granularity, supervision scheme, and fusion scheme on matching performance. Each configuration is trained separately. Test-set ROC curves are shown in Figure 2, with complete results in Appendix F.

Number of neighbors. We construct graphs with $k \in \{4, 8, 16, 24, 32\}$. As shown in Figure 2(a), as k increases from 4 to 8 and 16, Full-AUC rises from 0.683 to 0.831 and 0.920; increasing k further to 24 and 32 yields 0.919 and 0.917, respectively. This shows that too few neighbors harm matching accuracy, while increasing beyond 16 brings no further gains.

Edge type. We remove all cross-type edges from the neighborhood relation graph, retaining only same-type edges without adding other connections. In Figure 2(b), Full-AUC drops from 0.920 to 0.721, indicating that connections between the two point types are important for dual-similarity discrimination.

Table 2: End-to-end matching time of each method (mean ± sample standard deviation). Methods are ordered from slowest to fastest, column by column.

Method	Time (s) ↓	Method	Time (s) ↓
SiteMotif	3,849.13 ± 46.90	ProFSA	33.96 ± 1.05
G-LoSA	76.39 ± 2.18	APoc	30.06 ± 0.27
DeeplyTough	56.82 ± 0.65	**GCMatch**	23.76 ± 0.82
OneProt	42.59 ± 2.00	BindSiteS-CNN	**9.22 ± 1.15**

Matching scheme. Under final-loss-only supervision, we compare joint matching with branch matching. The former pools all nodes into a single vector before matching, producing one score; the latter pools geometric and chemical nodes separately, matches each independently, and combines the two branch scores through product fusion. As shown in Figure 2(c), switching to branch matching raises Full-AUC from 0.836 to 0.864, which shows that matching the two types of information separately and then fusing the two scores works better than merging them into a single score.

Representation granularity. We compare three representations: branch-level, type-level, and node-level. The branch-level variant pools geometric and chemical node representations into one vector each. The type-level variant pools geometric node representations into one vector and chemical node representations separately by functional type. The node-level variant retains one vector per node. All three configurations use both final and branch supervision, combining geometric and chemical matching scores via product fusion. As shown in Figure 2(d), the Full-AUCs of the three are 0.879, 0.899, and 0.920, respectively. Node-level achieves the highest Full-AUC, indicating that retaining fine-grained node representations improves geometry–chemistry dual-similarity discrimination.

Supervision scheme. To examine the effectiveness of the geometric and chemical losses during training, we remove them and retain only final supervision. In Figure 2(e), Full-AUC drops from 0.920 to 0.883, showing that branch supervision helps identify dual-similar samples.

Fusion scheme. To examine the effectiveness of product fusion, we test product, equal-weight summation, learned weighted summation, MLP, and minimum fusion configurations. As shown in Figure 2(f), product fusion achieves the highest Full-AUC of the five fusion schemes.

4.5 Matching efficiency

We compare end-to-end matching time on the complete test set of 8,000 pocket pairs. Timing covers all stages from input structures to output scores, and all methods run on the same hardware. As Table 2 shows, GCMatch takes 23.76 ± 0.82 seconds, second only to BindSiteS-CNN (9.22 ± 1.15 seconds) among the eight methods, and its runtime is 21.0%, 30.0%, and 44.2% lower than that of APoc, ProFSA, and OneProt, respectively. Combined with Table 1, GCMatch achieves the highest Full-AUC among the compared methods at the second-lowest matching time.

5 Conclusion

We propose GCMatch, a dual-similarity pocket matching method. It treats local cavities and chemical groups as two types of spatial points and builds a neighborhood relation graph that characterizes their spatial semantic relations through inter-point distance and direction. A graph encoder learns fine-grained node representations for the two point types. Multi-vector matching is applied to each type separately to obtain geometric and chemical similarities, and product fusion combines them into a final similarity that constrains both aspects. We also construct a four-quadrant dataset of pocket pairs annotated separately for geometric and chemical similarity, which supports training and evaluating models for dual-similarity pocket matching. Experiments show that the overall discriminative accuracy of GCMatch exceeds that of the classical and learning-based methods compared. Its matching efficiency is acceptable but remains lower than that of the fastest method, BindSiteS-CNN. Reducing matching overhead is a direction for future research. The chemical labels are currently assigned indirectly through ligand similarity, and direct annotations of pocket chemical environments would further strengthen the benchmark.

AI USE STATEMENT

In this work, we used generative AI tools to assist with writing, including polishing and editing the text, drafting sentences from author-provided content, condensing the text to meet the page limit, and checking the consistency of numerical values, citations, and terminology across the main text, tables, and appendix. We have not used generative AI tools for research ideation, method design, implementation, experiments, data analysis, or figure preparation. We have reviewed all AI-assisted text against the original manuscript and our experimental records, and all reported numbers were verified by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Section 3 describes the method, and Section 4.2 gives the training and evaluation protocol, including the optimization settings of GCMatch. Appendix A details point extraction, including the chemical functional point templates (Table 3), and the computation of node features. Appendix B describes the dataset construction, including data sources, splits, homology grouping, label thresholds, and the independent validation of the labels. Appendix C gives the scoring and training settings of the baselines, Appendix D the timing protocol and hardware, Appendix E the additional classification metrics, and Appendix F the complete ablation configurations and results. The code and the four-quadrant dataset are available at `https://github.com/Anonymousymous/GCMatch`.

REFERENCES

Melissa F. Adasme, Katja L. Linnemann, Sarah Naomi Bolz, Florian Kaiser, Sebastian Salentin, V. Joachim Haupt, and Michael Schroeder. PLIP 2021: expanding the scope of the protein–ligand interaction profiler to DNA and RNA. *Nucleic Acids Research*, 49(W1):W530–W534, 2021. doi: 10.1093/nar/gkab294.

Pedro J. Ballester and W. Graham Richards. Ultrafast shape recognition to search compound databases for similar molecular shapes. *Journal of Computational Chemistry*, 28(10):1711–1723, 2007. doi: 10.1002/jcc.20681.

S. Barelier, T. Sterling, M. J. O'Meara, and B. K. Shoichet. The recognition of identical ligands by unrelated proteins. *ACS Chemical Biology*, 10(12):2772–2784, 2015. doi: 10.1021/acschembio.5b00683.

Raymond E. Carhart, Dennis H. Smith, and R. Venkataraghavan. Atom pairs as molecular features in structure-activity studies: definition and applications. *Journal of Chemical Information and Computer Sciences*, 25(2):64–73, 1985. doi: 10.1021/ci00046a002.

M. Chartier and R. Najmanovich. Detection of binding site molecular interaction field similarities. *Journal of Chemical Information and Modeling*, 55(8):1600–1615, 2015. doi: 10.1021/acs.jcim.5b00333.

Y.-C. Chen, R. Tolbert, A. M. Aronov, G. McGaughey, W. P. Walters, and L. Meireles. Prediction of protein pairs sharing common active ligands using protein sequence, structure, and ligand similarity. *Journal of Chemical Information and Modeling*, 56(9):1734–1745, 2016. doi: 10.1021/acs.jcim.6b00118.

Janani Durairaj, Yusuf Adeshina, Zhonglin Cao, Xuejin Zhang, Vladas Oleinikovas, Thomas Duignan, Zachary McClure, Xavier Robin, Gabriel Studer, Daniel Kovtun, Emanuele Rossi, Guoqing Zhou, Srimukh Veccham, Clemens Isert, Yuxing Peng, Prabindh Sundareson, Mehmet Akdel, Gabriele Corso, Hannes Stärk, Gerardo Tauriello, Zachary Carpenter, Michael Bronstein, Emine Kucukbenli, Torsten Schwede, and Luca Naef. PLINDER: The protein-ligand interactions dataset and evaluation resource. *bioRxiv*, 2024. doi: 10.1101/2024.07.17.603955.

Joseph L. Durant, Burton A. Leland, Douglas R. Henry, and James G. Nourse. Reoptimization of MDL keys for use in drug discovery. *Journal of Chemical Information and Computer Sciences*, 42(6):1273–1280, 2002. doi: 10.1021/ci010132r.

C. Ehrt, T. Brinkjost, and O. Koch. A benchmark driven guide to binding site comparison: An exhaustive evaluation using tailor-made data sets (ProSPECCTs). *PLOS Computational Biology*, 14(11):e1006483, 2018. doi: 10.1371/journal.pcbi.1006483.

K. Flöge, S. Udayakumar, J. Sommer, M. Piraud, S. Kesselheim, V. Fortuin, S. Günnemann, K. J. van der Weg, H. Gohlke, E. Merdivan, and A. Bazarova. OneProt: Towards multi-modal protein foundation models via latent space alignment of sequence, structure, binding sites and text encoders. *PLOS Computational Biology*, 21(11):e1013679, 2025. doi: 10.1371/journal.pcbi.1013679.

Bowen Gao, Yinjun Jia, YuanLe Mo, Yuyan Ni, Wei-Ying Ma, Zhi-Ming Ma, and Yanyan Lan. Self-supervised pocket pretraining via protein fragment-surroundings alignment. In *International Conference on Learning Representations (ICLR)*, 2024.

M. Gao and J. Skolnick. APoc: large-scale identification of similar protein pockets. *Bioinformatics*, 29(5):597–604, 2013. doi: 10.1093/bioinformatics/btt024.

Zhangyang Gao, Cheng Tan, Jun Xia, and Stan Z. Li. Co-supervised pre-training of pocket and ligand. In *Machine Learning and Knowledge Discovery in Databases: Research Track (ECML PKDD)*, volume 14169 of *Lecture Notes in Computer Science*, pp. 405–421, Cham, 2023. Springer Nature Switzerland. doi: 10.1007/978-3-031-43412-9_24.

Peter Gedeck, Bernhard Rohde, and Christian Bartels. QSAR – how good is it in practice? Comparison of descriptor sets on an unbiased cross section of corporate data sets. *Journal of Chemical Information and Modeling*, 46(5):1924–1936, 2006. doi: 10.1021/ci050413p.

Alberto Gobbi and Dieter Poppinger. Genetic optimization of combinatorial libraries. *Biotechnology and Bioengineering*, 61(1):47–54, 1998. doi: 10.1002/(sici)1097-0290(199824)61:1<47::aid-bit9>3.0.co;2-z.

R. G. Govindaraj and M. Brylinski. Comparative assessment of strategies to identify similar ligand-binding pockets in proteins. *BMC Bioinformatics*, 19(1):91, 2018. doi: 10.1186/s12859-018-2109-2.

Jun Hu, Xiaogen Zhou, Dongjun Yu, Guijun Zhang, and Yang Zhang. PPS-align: an user-friendly pocket structural alignment server. `https://zhanggroup.org/PPS-align/`, n.d. Accessed September 2026.

Bowen Jing, Stephan Eismann, Patricia Suriana, Raphael J. L. Townshend, and Ron Dror. Learning from protein structure with geometric vector perceptrons. In *International Conference on Learning Representations (ICLR)*, 2021.

A. Kahraman, R. J. Morris, R. A. Laskowski, and J. M. Thornton. Shape variation in protein binding pockets and their ligands. *Journal of Molecular Biology*, 368(1):283–301, 2007. doi: 10.1016/j.jmb.2007.01.086.

A. Kahraman, R. J. Morris, R. A. Laskowski, A. D. Favia, and J. M. Thornton. On the diversity of physicochemical environments experienced by identical ligands in binding pockets of unrelated proteins. *Proteins: Structure, Function, and Bioinformatics*, 78(5):1120–1136, 2010. doi: 10.1002/prot.22633.

Omar Khattab and Matei Zaharia. ColBERT: Efficient and effective passage search via contextualized late interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 39–48, 2020. doi: 10.1145/3397271.3401075.

J. Konc and D. Janezic. ProBiS algorithm for detection of structurally similar protein binding sites by local structural alignment. *Bioinformatics*, 26(9):1160–1168, 2010. doi: 10.1093/bioinformatics/btq100.

Timo Krotzky, Christian Grunwald, Ute Egerland, and Gerhard Klebe. Large-scale mining for similar protein binding pockets: with RAPMAD retrieval on the fly becomes real. *Journal of Chemical Information and Modeling*, 55(1):165–179, 2015. doi: 10.1021/ci5005898.

Vincent Le Guilloux, Peter Schmidtke, and Pierre Tuffery. Fpocket: an open source platform for ligand pocket detection. *BMC Bioinformatics*, 10(1):168, 2009. doi: 10.1186/1471-2105-10-168.

H. S. Lee and W. Im. G-LoSA: An efficient computational tool for local structure-centric biological studies and drug design. *Protein Science*, 25(4):865–876, 2016. doi: 10.1002/pro.2890.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

Ramaswamy Nilakantan, Norman Bauman, J. Scott Dixon, and R. Venkataraghavan. Topological torsion: a new molecular descriptor for SAR applications. Comparison with other descriptors. *Journal of Chemical Information and Computer Sciences*, 27(2):82–85, 1987. doi: 10.1021/ci00054a008.

Art B. Owen. The pigeonhole bootstrap. *The Annals of Applied Statistics*, 1(2):386–411, 2007. doi: 10.1214/07-aoas122.

RDKit. RDKit: Open-source cheminformatics. `https://www.rdkit.org`, n.d. Accessed September 2026.

T. Reim, C. Ehrt, J. Graef, S. Günther, A. Meents, and M. Rarey. SiteMine: Large-scale binding site similarity searching in protein structure databases. *Archiv der Pharmazie*, 357(5):e2300661, 2024. doi: 10.1002/ardp.202300661.

David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, 50(5):742–754, 2010. doi: 10.1021/ci100050t.

Sebastian Salentin, Sven Schreiber, V. Joachim Haupt, Melissa F. Adasme, and Michael Schroeder. PLIP: fully automated protein–ligand interaction profiler. *Nucleic Acids Research*, 43(W1): W443–W447, 2015. doi: 10.1093/nar/gkv315.

S. Sankar and N. Chandra. SiteMotif: A graph-based algorithm for deriving structural motifs in protein ligand binding sites. *PLOS Computational Biology*, 18(2):e1009901, 2022. doi: 10.1371/journal.pcbi.1009901.

S. Schmitt, D. Kuhn, and G. Klebe. A new method to detect related function among proteins independent of sequence and fold homology. *Journal of Molecular Biology*, 323(2):387–406, 2002. doi: 10.1016/s0022-2836(02)00811-2.

O. B. Scott, J. Gu, and A. W. E. Chan. Classification of protein-binding sites using a spherical convolutional neural network. *Journal of Chemical Information and Modeling*, 62(22):5383–5396, 2022. doi: 10.1021/acs.jcim.2c00832.

Alexandra Shulman-Peleg, Ruth Nussinov, and Haim J. Wolfson. Recognition of functional sites in protein structures. *Journal of Molecular Biology*, 339(3):607–633, 2004. doi: 10.1016/j.jmb.2004.04.012.

M. Simonovsky and J. Meyers. DeeplyTough: Learning structural comparison of protein binding sites. *Journal of Chemical Information and Modeling*, 60(4):2356–2366, 2020. doi: 10.1021/acs.jcim.9b00554.

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.

K. Yeturu and N. Chandra. PocketMatch: a new algorithm to compare binding sites in protein structures. *BMC Bioinformatics*, 9(1):543, 2008. doi: 10.1186/1471-2105-9-543.

Gengmo Zhou, Zhifeng Gao, Qiankun Ding, Hang Zheng, Hongteng Xu, Zhewei Wei, Linfeng Zhang, and Guolin Ke. Uni-Mol: A universal 3D molecular representation learning framework. In *International Conference on Learning Representations (ICLR)*, 2023.

A Point Extraction and Node Features

A.1 Point Extraction

Geometric points. We run fpocket on the cropped pocket structure and take its top-ranked non-empty candidate cavity. The alpha-sphere centers of this cavity give the coordinates of the geometric points, and the sphere radii give r_i.

Chemical functional point templates. Chemical functional points are generated from the fixed residue templates in Table 3. Atom-type points take the coordinates of the corresponding atom, and group-type points take the geometric center of the listed atoms. A point may belong to several functional types, which are encoded as a multi-hot vector.

Table 3: Chemical functional point templates.

Source	Point position	Functional type and condition	Direction reference atom
Backbone	N	HBD; not for PRO	CA
Backbone	O	HBA	C
ASP	One point each at OD1 and OD2; one at their center	Atom points: HBA; center point: negative charge; requires both oxygens	Atom points: CG
GLU	One point each at OE1 and OE2; one at their center	Atom points: HBA; center point: negative charge; requires both oxygens	Atom points: CD
SER, THR, TYR	OG, OG1, and OH, respectively	The same point is both HBA and HBD	CB, CB, and CZ, respectively
ASN	OD1, ND2	HBA and HBD, respectively; each generated if the atom is present	CG
GLN	OE1, NE2	HBA and HBD, respectively; each generated if the atom is present	CD
LYS	NZ	The same point is both HBD and positive charge	CE
ARG	One point each at NE, NH1, and NH2; one at the center of the available atoms	Atom points: HBD; positive-charge center point generated when at least two of these atoms are present	NE: CD; NH1, NH2: CZ
TRP	NE1	HBD	CD1
HIS	One point each at ND1 and NE2; one at their center	Each atom point is both HBA and HBD; positive-charge center point generated when both atoms are present	ND1: CG; NE2: CD2
PHE, TYR ring	Center of CG, CD1, CD2, CE1, CE2, CZ	Aromatic	Ring normal
HIS ring	Center of CG, ND1, CD2, CE1, NE2	Aromatic	Ring normal
TRP ring 1	Center of CG, CD1, NE1, CE2, CD2	Aromatic	Ring normal
TRP ring 2	Center of CD2, CE2, CE3, CZ2, CZ3, CH2	Aromatic	Ring normal
Hydrophobic cluster	Center of all candidate atoms in a connected cluster	Hydrophobic	—

An aromatic ring requires at least three available ring atoms, and ring atoms are listed in order. Charge-center points and hydrophobic points have no direction.

Backbone indicator. The indicator is one for chemical functional points generated from backbone N or O atoms and zero otherwise.

Hydrophobic points. Hydrophobic candidate atoms come from ALA, VAL, LEU, ILE, MET, PHE, PRO, TRP, and CYS: after excluding backbone C, CA, N, and O, we keep side-chain atoms whose

names begin with C, plus SD of MET. We construct a distance graph over these atoms, connecting two atoms when their Euclidean distance is at most 4 Å. Each connected component produces one hydrophobic point at the arithmetic mean of its atomic coordinates. Connections can pass through intermediate atoms, so the distance between two atoms in the same component can exceed 4 Å.

A.2 Node Feature Computation

Cavity position and spatial extent. Let $\mathbf{x}_1, \ldots, \mathbf{x}_{N_G}$ denote the coordinates of the N_G geometric points. For $N_G \geq 3$, we compute the cavity centroid and coordinate covariance as

$$\boldsymbol{\mu} = \frac{1}{N_G} \sum_{i=1}^{N_G} \mathbf{x}_i, \qquad \boldsymbol{\Sigma} = \frac{1}{N_G} \sum_{i=1}^{N_G} (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^\top \tag{8}$$

Let $\lambda_1, \lambda_2, \lambda_3$ be the covariance eigenvalues in descending order and $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3$ their unit eigenvectors. The projection onto axis ℓ is $p_{i\ell} = (\mathbf{x}_i - \boldsymbol{\mu})^\top \mathbf{u}_\ell$, for $\ell \in \{1, 2, 3\}$.

We set the sign of each axis using the sum of cubed projections: $a_\ell = -1$ when this sum is negative and $a_\ell = 1$ otherwise. The resulting projection is $q_{i\ell} = a_\ell p_{i\ell}$.

Each geometric point uses its centroid distance $d_i^{\text{cen}} = \|\mathbf{x}_i - \boldsymbol{\mu}\|_2$, its three principal-axis projections, and the three standard deviations $\sigma_\ell = \sqrt{\max(\lambda_\ell, 0)}$. All geometric points in a pocket share these standard deviations. When $N_G < 3$, these seven scalar features are set to zero.

Local spatial context. Let $\mathbf{y}_1, \ldots, \mathbf{y}_{N_C}$ denote the chemical functional point coordinates. The distance to the nearest chemical functional point, its proximity indicator, and the local geometric point count are

$$d_i^{\text{chem}} = \min_{1 \leq j \leq N_C} \|\mathbf{x}_i - \mathbf{y}_j\|_2, \qquad b_i = \mathbb{1}[d_i^{\text{chem}} \leq 4\,\text{Å}], \qquad \rho_i = \sum_{j \neq i} \mathbb{1}[\|\mathbf{x}_i - \mathbf{x}_j\|_2 \leq 8\,\text{Å}] \tag{9}$$

The local count includes other geometric points. If a pocket contains no chemical functional points, we set $d_i^{\text{chem}} = +\infty$ and $b_i = 0$, replacing this distance with a finite value of 50 Å before it enters the network.

Direction computation. The centroid direction for a geometric point is $(\boldsymbol{\mu} - \mathbf{x}_i)/\max(d_i^{\text{cen}}, 10^{-6})$, set to zero when $N_G < 3$. For a functional atom a and its reference atom b listed in Table 3, the donor direction or acceptor axis is $(\mathbf{x}_a - \mathbf{x}_b)/\|\mathbf{x}_a - \mathbf{x}_b\|_2$. The backbone donor direction points from CA to N, and the backbone acceptor direction from C to O. The aromatic ring normal is the direction associated with the smallest singular value of the centered ring coordinates. Its sign is set by the cross product of the first three noncollinear ring atoms in the order listed in Table 3.

Input dimensions. The scalar input $\mathbf{s}_i \in \mathbb{R}^{23}$ contains 11 geometric features, 10 chemical features, and two node type indicators. The geometric scalars comprise four scale features, four position features, and three neighborhood features. The chemical scalars contain six chemical type indicators, one backbone indicator, and three direction validity indicators. The vector input $\mathbf{V}_i \in \mathbb{R}^{4\times 3}$ stores the donor direction, acceptor axis, aromatic ring normal, and geometric point centroid direction, in that order.

B Dataset Construction and Validation

B.1 Source Data, Splits, and Labels

Source version and eligibility. We use PLINDER release 2024-06/v2 and select holo systems with exactly one ligand chain satisfying `ligand_is_proper=True`. This yields 434,320 systems, from which 432,995 pockets are successfully extracted.

Transitive grouping of structures. Sequence homology is obtained by an all-vs-all MMseqs2 (18.8cc5c) search over all deduplicated sequences in the FASTA files of PLINDER systems (56,732

sequences), with `--min-seq-id 0.20`, `-c 0.8`, `--cov-mode 0`, `-e 0.01`, and `-s 5.7` (Steinegger & Söding, 2017). Alignments with sequence identity (nident/alnlen) of at least 0.30, query and target coverage of at least 0.80, and an E-value of at most 0.01 are treated as homologous. Systems with identical sequence strings are also grouped directly. We take the transitive closure of these sequence relations and same-PDB-entry relations, assigning directly or indirectly related structures to the same group.

Calibration and candidate separation. Within each split, entire closures are assigned to calibration (CAL), audit (AUD), or production (PROD). PROD pockets used for candidate generation must come from X-ray structures, pass structural quality filtering, and have available pocket coordinates and ECFP4 fingerprints.

Candidate generation and size strata. Within each split, we generate candidate pairs through background sampling, chemical nearest neighbors, and pairs sharing the same ligand. Self-pairs and pairs from the same PDB entry are excluded. We then compute PPS-align scores and their corresponding p values in both directions. To account for pocket size, we divide endpoints into ten size strata according to their numbers of C_α atoms. The two endpoints define one of 55 unordered combinations of size strata, which are used for calibration and sample selection. The stratum boundaries are frozen deciles of the C_α atom count, giving ten strata of $\leq$10, 11–16, 17–20, 21–23, 24–25, 26–28, 29–30, 31–34, 35–39, and $\geq$40 atoms.

Chemical thresholds. ECFP4 fingerprints use radius 2 and 2,048 bits. Tanimoto similarities of at least 0.70 receive $C = 1$, and similarities of at most 0.40 receive $C = 0$. Candidates in the intermediate range are excluded.

Geometric thresholds. For each size-stratum combination, we pool the CAL data of the three splits to determine the positive and negative PPS-score thresholds during CAL, which are then frozen. Calibration status is checked separately for each split and size-stratum combination, and PROD candidates use the corresponding stratified thresholds. Background pairs are CAL pocket pairs from the same split with different PDB entries and different closures. Let z^+ and z^- be the smaller and larger of the two directional PPS-scores. The positive threshold is the 99% empirical quantile of z^+ over the background, and the negative threshold is the 50% empirical quantile of z^-. Calibration requires at least 20,000 valid background pairs per stratum and a difference of at least 0.02 between the 95% and 50% quantiles of z^+. We assign $G = 1$ when both directional scores reach their positive thresholds and both p values are below 0.01. We assign $G = 0$ when both scores are at or below their negative thresholds and both p values are at least 0.05. All other candidates are excluded.

Candidate validity. We exclude pairs with invalid Gumbel calibration, invalid PPS scores, or endpoints in the same closure.

Candidate counts. Table 4 reports counts before joint selection. After downsampling EASY, Train uses all eligible candidates for joint selection. Validation and Test further retain candidates within the common support of the four classes in the selection features.

Table 4: Numbers of pockets and pairs at each construction stage.

Stage	Train	Validation	Test
Eligible PROD pockets	40,214	12,411	10,588
Candidate pairs scored by PPS-align	38,909,147	41,385,316	55,139,789
Candidate pairs after EASY downsampling	653,637	496,457	462,363
Input pairs for joint selection	653,637	14,108	18,592

Integrity checks after freezing. We find no duplicate or reversed duplicate pairs, label violations, or non-finite descriptor values. Pocket pairs, endpoints, and closures do not overlap across splits.

B.2 Joint Sample Selection and Distribution Control

Selection features. We use 14 fields describing chemical similarity, geometric similarity, and pocket size. The eight chemical fields are five similarities (ECFP4 (Rogers & Hahn, 2010), MACCS (Durant et al., 2002), AtomPair (Carhart et al., 1985), TT (Nilakantan et al., 1987), and Pharm2D (Gobbi & Poppinger, 1998)), the physicochemical property distance, and indicators for the same ligand

and the same scaffold. The three geometric fields are the PPS-score margin relative to the stratified threshold (zmargin), USR-heavy computed from heavy-atom coordinates, and USR-backbone computed from backbone coordinates (Ballester & Richards, 2007). The three size fields are the minimum, maximum, and absolute difference of the two pocket sizes.

Selection procedure. We first identify the common support of the four candidate classes in the selection features and assign sample quotas to size-stratum combinations. POS is selected while accounting for diversity in pockets, closures, and ligand scaffolds. In addition to the distribution controls between POS and the hard negatives described in the main text, we match the chemical similarity distribution of EASY to HARD-C and its geometric similarity distribution to HARD-G. The four classes have identical counts within each size-stratum combination in a split.

For Train, we fit propensity scores between the relevant candidate classes, bin candidates within size-stratum combinations, and combine maximum flow with nearest-neighbor matching. For Validation and Test, we exchange candidates within fixed size quotas to directly control univariate AUC.

Distribution control. We treat the geometric and chemical labels as two axes and call the axis on which the two compared groups share the same label the shared axis. For each descriptor on this axis, we calculate its AUC for distinguishing the two sample groups and remove score-direction dependence using

$$\mathrm{AUC}^{\star} = \max\big(\mathrm{AUC},\, 1 - \mathrm{AUC}\big) \tag{10}$$

A value closer to 0.5 indicates that the descriptor has less ability to distinguish the groups. For each comparison, we take the maximum $\mathrm{AUC}^{\star}$ over descriptors on the shared axis. The construction target is a point estimate of at most 0.55.

Table 5: Distribution control on shared axes for descriptors used in sample selection.

Split	Comparison	Shared axis	Descriptor attaining the maximum	$\mathrm{AUC}^{\star}$ [95% CI]
Train	POS / HARD-G	Chemical	AtomPair	0.524 [0.501, 0.609]
Train	POS / HARD-C	Geometric	zmargin	0.515 [0.501, 0.590]
Train	HARD-C / EASY	Chemical	TT	0.526 [0.502, 0.629]
Train	HARD-G / EASY	Geometric	USR-heavy	0.540 [0.507, 0.578]
Validation	POS / HARD-G	Chemical	Same-ligand indicator	0.522 [0.502, 0.622]
Validation	POS / HARD-C	Geometric	zmargin	0.525 [0.503, 0.709]
Validation	HARD-C / EASY	Chemical	TT	0.525 [0.502, 0.646]
Validation	HARD-G / EASY	Geometric	USR-backbone	0.525 [0.502, 0.617]
Test	POS / HARD-G	Chemical	Physicochemical property distance	0.514 [0.502, 0.636]
Test	POS / HARD-C	Geometric	zmargin	0.525 [0.503, 0.749]
Test	HARD-C / EASY	Chemical	AtomPair	0.525 [0.502, 0.703]
Test	HARD-G / EASY	Geometric	USR-heavy	0.525 [0.502, 0.653]

Each row reports the largest direction-independent AUC among the descriptors on the corresponding shared axis and its 95% confidence interval. Appendix B.3 describes interval estimation.

B.3 Independent Validation after Dataset Freezing

Chemical probes. Pattern2048, RDK-path2048, and Avalon512 (Gedeck et al., 2006) use ligand fingerprints of 2,048, 2,048, and 512 bits, respectively. All three are computed with RDKit (RDKit, n.d.) from the canonical SMILES of the central ligand, and the pair score is the Tanimoto similarity between the two ligand fingerprints.

Geometric probes. Both representations use the coordinates of pocket backbone atoms N, CA, C, and O. BB-D2 divides all backbone interatomic distances by their root mean square and takes the 19 quantiles from 5% to 95% in steps of 5%. Its pair score is $1/(1+\bar{d})$, where $\bar{d}$ is the mean absolute difference between the two vectors. BB-shape spectrum consists of the three eigenvalues of the coordinate covariance matrix normalized by their sum and the nine quantiles, from 10% to 90% in steps of 10%, of atomic radial distances normalized by the radius of gyration. Its pair score is $1/(1+\|\mathbf{v}_A-\mathbf{v}_B\|_2)$. Neither probe requires ligand information or structural alignment, and neither uses PPS-align or USR.

Statistical computation. AUC calculation accounts for tied scores. The 95% confidence intervals in Tables 5–7 are estimated with a two-ended pigeonhole bootstrap (Owen, 2007) that resamples closures, with 1,000 replicates. Each replicate resamples closures with replacement and weights each pocket pair by the product of the numbers of times its two endpoint closures are drawn, preserving dependence due to shared structures.

Table 6: Independent probe validation on HARD-G versus HARD-C.

Independent probe	Train	Validation	Test
Pattern2048	0.991 [0.971, 0.996]	0.996 [0.983, 1.000]	0.999 [0.998, 1.000]
RDK-path2048	0.999 [0.997, 1.000]	1.000 [0.999, 1.000]	1.000 [1.000, 1.000]
Avalon512	0.999 [0.998, 1.000]	1.000 [1.000, 1.000]	1.000 [0.999, 1.000]
BB-D2	0.219 [0.190, 0.279]	0.277 [0.184, 0.396]	0.321 [0.178, 0.499]
BB-shape spectrum	0.184 [0.156, 0.239]	0.267 [0.169, 0.394]	0.228 [0.124, 0.377]

Values are AUC [95% CI] without score-direction reversal, with HARD-G assigned to class 1 and HARD-C to class 0 for this comparison. Chemical AUCs are expected to exceed 0.5 and geometric AUCs to fall below 0.5. Both groups remain negative in the pocket matching task.

Table 7: Independent probe validation on POS versus EASY.

Independent probe	Train	Validation	Test
Pattern2048	0.994 [0.989, 0.997]	0.998 [0.993, 1.000]	1.000 [0.998, 1.000]
RDK-path2048	1.000 [0.998, 1.000]	1.000 [0.999, 1.000]	1.000 [1.000, 1.000]
Avalon512	0.999 [0.999, 1.000]	1.000 [1.000, 1.000]	1.000 [0.999, 1.000]
BB-D2	0.691 [0.649, 0.725]	0.637 [0.509, 0.760]	0.686 [0.612, 0.775]
BB-shape spectrum	0.694 [0.652, 0.730]	0.634 [0.545, 0.726]	0.666 [0.574, 0.762]

Values are AUC [95% CI], with POS as the positive class and EASY as the negative class. Both chemical and geometric AUCs are expected to exceed 0.5. This validation is performed after the dataset is frozen.

C Baseline Settings

Classical methods involve no training and use their official matching scores. For learning-based methods, apart from the unified 50 training epochs, optimization settings follow their official implementations or original papers, as listed in Table 8.

Table 8: Scoring and training settings of the baseline methods.

Method	Settings
APoc	Official PS-score (Gao & Skolnick, 2013)
SiteMotif	Official M-dist$_{\max}$ score (Sankar & Chandra, 2022)
G-LoSA	Official GA-score with default parameters `-b 0 -f 1 -iter 3 -itercf 0 -n 1 -ss 0 -surf 0` (Lee & Im, 2016)
DeeplyTough	Adam, learning rate 10^{-3}, weight decay 5×10^{-4}, batch size 16, contrastive margin 1.0, following the official training code (Simonovsky & Meyers, 2020)
BindSiteS-CNN	SGD (momentum 0.9), learning rate 5×10^{-4}, batch size 128, contrastive margin 1.25, following the official TOUGH-M1 example (Scott et al., 2022)
ProFSA	Adam, learning rate 10^{-5}, effective batch size 256, 10% warm-up followed by polynomial decay; fine-tuning is implemented on top of the official codebase following Appendix E.3 of the original paper (Gao et al., 2024)
OneProt	AdamW, learning rate 10^{-3}, weight decay 0.01, batch size 32, ReduceLROnPlateau (factor 0.1, patience 10), following the official configuration and training code (Flöge et al., 2025)

D Matching Timing Protocol

Runs and caching. The complete test set involves 3,375 distinct pockets. Each cold start launches a new process and builds the caches required by the method. Computed pocket representations can

be reused within a run; representation caches are not reused across runs. Methods are measured in interleaved order within the same measurement session.

Timing boundary. Wall-clock time is measured outside the subprocess. It includes process startup, module imports, structure parsing, feature preparation, encoding where applicable, cache reads and writes, matching, and result output. Inputs are cropped pocket structures. Pocket localization and cropping from complete proteins occur before the supplied input and are excluded from timing.

Hardware and model. All methods are allocated the same 12 Intel Xeon Platinum 8481C CPU cores. The GPU components of GCMatch and the learning-based methods additionally use one NVIDIA RTX 4090 D, with their CPU-side computation confined to the same 12 cores, and the classical methods run on the CPU only. The OneProt timing in Table 2 uses the OneProt-5 pretrained model.

E Additional Classification Metrics

Threshold selection. Each method selects the threshold that maximizes F1 on the validation set. The threshold is then frozen to calculate test Accuracy, Precision, Recall, Specificity, and MCC.

Table 9: Test classification metrics with thresholds frozen on the validation set.

Method	Accuracy ↑	Precision ↑	Recall ↑	Specificity ↑	MCC ↑
ProFSA	0.665	0.410	0.776	0.628	0.350
BindSiteS-CNN	0.699	0.437	0.703	0.698	0.354
DeeplyTough	0.679	0.422	0.765	0.651	0.361
G-LoSA	0.815	0.601	0.770	0.830	0.556
SiteMotif	0.810	0.586	0.815	0.809	0.567
APoc	0.761	0.513	**0.874**	0.723	0.523
OneProt	0.827	0.620	0.798	0.837	0.588
GCMatch	**0.849**	**0.654**	0.840	**0.852**	**0.641**

Higher values are better for all metrics. Bold values indicate the maximum in each column.

F Complete Ablation Results

Tables 10–15 supplement Figures 2(a)–(f) with complete AUC results. Configurations are described in Section 4.4 and metrics in Section 4.2. Bold values indicate the maximum in each column of each table.

F.1 Number of Neighbors

The value k is the number of neighbors selected for each node.

Table 10: Complete test results for different numbers of neighbors.

k	Full ↑	Easy ↑	G ↑	C ↑	G/C mean ↑
4	0.683	0.733	0.643	0.674	0.658
8	0.831	0.866	0.793	0.835	0.814
32	0.917	0.949	0.907	0.894	0.900
24	0.919	**0.955**	0.907	0.894	0.900
16	**0.920**	0.950	**0.914**	**0.895**	**0.905**

F.2 EDGE TYPE

Table 11: Complete test results for the cross-type edge ablation.

Graph structure	Full ↑	Easy ↑	G ↑	C ↑
Without cross-type edges	0.721	0.779	0.666	0.719
Full neighborhood graph	**0.920**	**0.950**	**0.914**	**0.895**

F.3 MATCHING SCHEME

Joint matching averages all node representations, and branch matching averages the representations of geometric and chemical nodes separately. The mean vectors are L_2-normalized, and the matching score is their ReLU-clipped cosine similarity. The single score of joint matching is used directly as the final score, and the two scores of branch matching are fused by product.

Table 12: Complete test results for matching strategies under final supervision.

Configuration	Full ↑	Easy ↑	G ↑	C ↑
Joint matching	0.836	0.870	0.830	0.809
Branch matching	**0.864**	**0.886**	**0.850**	**0.854**

F.4 REPRESENTATION GRANULARITY

Branch-level representations are computed as in branch matching (Appendix F.3). Type-level representations average all geometric nodes and, within each chemical functional type, the nodes of that type, with multi-hot points entering every type they belong to. The mean vectors are L_2-normalized and scored by ReLU-clipped cosine similarity, and the chemical score is averaged over types as in Section 3.3.

Table 13: Complete test results for representation granularity under final and branch supervision.

Configuration	Full ↑	Easy ↑	G ↑	C ↑
Branch representations	0.879	0.914	0.880	0.841
Type representations	0.899	0.932	0.893	0.871
Node representations	**0.920**	**0.950**	**0.914**	**0.895**

F.5 SUPERVISION SCHEME

Table 14: Complete test results for supervision settings.

Configuration	Full ↑	Easy ↑	G ↑	C ↑
Final sup.	0.883	0.912	0.859	0.878
Final + branch sup.	**0.920**	**0.950**	**0.914**	**0.895**

F.6 FUSION SCHEME

All configurations use the same graph, multi-vector representations, matching within chemical types, and final and branch supervision. In the learned weighted sum, $\alpha = \mathrm{Sigmoid}(\theta)$ with a learnable scalar θ. The MLP takes only the geometric and chemical scores as input and consists of an 8-dimensional hidden layer with SiLU activation and a one-dimensional output, followed by a sigmoid that produces the final score.

Table 15: Complete test results for fusion methods.

Fusion scheme	Final score	Full ↑	Easy ↑	G ↑	C ↑
Minimum	$\min(S_G, S_C)$	0.866	0.909	0.852	0.836
Equal-weight sum	$(S_G + S_C)/2$	0.893	0.933	0.867	0.880
MLP	$\mathrm{MLP}(S_G, S_C)$	0.898	0.930	0.881	0.883
Learned weighted sum	$\alpha S_G + (1-\alpha) S_C$	0.914	**0.954**	0.890	**0.898**
Product	$S_G S_C$	**0.920**	0.950	**0.914**	0.895