Zero-Shot Protein–Ligand Binding-Residue Prediction from Protein Sequence and SMILES

Mahdi Pourmirzaei[1,2,*] **Salhuldin Alqarghuli** [1,*] **Kai Chen** [1,*]

Mohammadreza Pourmirzaei[3] **Dong Xu** [1,+]

[1] University of Missouri
{mpngf,saakdr,dc57y,xudong}@missouri.edu
[2] ProGene
[3] Politecnico di Milano, Milan, Italy
mohammadreza.pourmirzaeioliaei@mail.polimi.it
[*] Equal contribution
[+] Corresponding author

Abstract

Accurate identification of protein–ligand binding residues is critical for mechanistic biology and drug discovery, yet performance varies widely across ligand families and data regimes. We present a systematic evaluation framework that stratifies ligands into three settings, *overrepresented* (many examples), *underrepresented* (tens of examples), and *zero-shot* (unseen at training). We develop a three-stage, sequence-based modeling suite that progressively adds ligand conditioning and *zero-shot* capability, and use the evaluation framework to assess the suite. Stage 1 trains per-ligand predictors using a pretrained protein language model (PLM). Stage 2 introduces ligand-aware conditioning via an embedding table, enabling a single multi-ligand model. Stage 3 replaces the table with a pretrained chemical language model (CLM) operating on SMILES, enabling *zero-shot* generalization. We show Stage 2 improves Macro F_1 on the *overrepresented* test set from 0.4769 (Stage 1) to 0.5832 and outperforms sequence- and structure-based baselines. Stage 3 attains *zero-shot* performance ($F_1 = 0.3109$) on 5 612 previously unseen ligands while remaining competitive on represented ligands. Ablations across five PLM scales and multiple CLMs reveal larger PLM backbones consistently increase Macro F_1 across all regimes, whereas scaling the CLM yields modest or inconsistent gains, which need further investigation. Our results demonstrate that *zero-shot* residue-level prediction from sequence and SMILES is feasible and identifies the PLM scale as the dominant lever for further advances.

1 Introduction

Reliable identification of protein-ligand binding residues plays a critical role in understanding fundamental biological processes such as gene expression regulation, signal transduction pathways, and antigen–antibody interactions [1–3]. Additionally, precise determination of protein-ligand interaction sites is fundamental for effective drug discovery and rational therapeutic development [4]. However, pinpointing ligand-binding sites accurately remains challenging, particularly when high-resolution structural data for proteins are unavailable [5]. Although experimental techniques like

Submitted to AI for Science Workshop (NeurIPS 2025).

nuclear magnetic resonance and absorption spectroscopy provide high-quality data, they are expensive and labor-intensive, underscoring the critical need for computational approaches capable of rapid, high-throughput binding site predictions [6].

Despite notable advancements in computational prediction methodologies for protein-ligand interactions, several limitations persist. Many current computational tools lack versatility, restricting their effectiveness to specific ligand categories [1, 7–9]. This constraint is particularly challenging given the vast chemical diversity of ligands encountered biologically. Furthermore, data scarcity significantly impacts model performance, particularly affecting ligand types with limited available training examples, which hampers effective generalization [10–12].

To assess the effects of data scarcity on protein-ligand binding site prediction, we systematically establish an evaluation framework comprising three distinct test scenarios: *overrepresented*, *underrepresented*, and *zero-shot* ligand cases. Initially, we develop a baseline model designed to predict binding sites for a single ligand type. Subsequently, we enhance this foundational architecture to accommodate multiple ligands within a single model and further expand it to support *zero-shot* ligand predictions. Our approach not only achieves state-of-the-art performance but also quantifies how prediction performance shifts across these three evaluation scenarios.

Our work introduces several novel contributions, which are summarized as follows:

1. **A homology-controlled, three-regime benchmark.** We establish a standardized evaluation with ligand-wise splits for *overrepresented*, *underrepresented*, and *zero-shot* ligands, enabling directly comparable assessments across methods.
2. **A progressive, sequence-based modeling suite.** We introduce a three-stage framework: (i) per-ligand baselines, (ii) a single multi-ligand model conditioned by a learned ligand embedding, and (iii) a *zero-shot* model that conditions on SMILES via a pretrained chemical language model (CLM).
3. **Scaling analysis across PLMs and CLMs.** Through controlled ablations, we quantify that enlarging the protein encoder (ESM-2) consistently dominates gains across all regimes, while CLM choice/scale yields smaller or inconsistent improvements.

2 Related Work

State-of-the-art sequence-based predictors of protein–ligand binding residues increasingly center on pretrained protein language models (PLMs). PLMs address two long-standing challenges: (i) modeling long-range dependencies via transformers and (ii) improving sample efficiency through large-scale pretraining. We group recent PLM-driven approaches by how they adapt pretrained representations to binding-site prediction.

A common line of work keeps the PLM frozen (or lightly tuned) and adds lightweight classifiers. BindEmbed21 [13] feeds ProtT5 embeddings into a shallow CNN and combines them with homology-based inference, using class-weighted losses to counter label imbalance. CLAPE-SMB [9] freezes ESM-2 and employs a five-layer MLP trained with class-balanced focal loss plus a triplet-center term to better separate binding from non-binding residues. For metal ions, LMetalSite [6] regularizes PLM embeddings with Gaussian noise, shares transformer layers across tasks, and uses ion-specific MLP heads within a multi-task framework to capture cross-ion commonalities while remaining ligand-aware.

IonPred [14] instead adopts a generator–discriminator setup (a masked language model (MLM)-style generator with an ELECTRA-like discriminator) to learn residue-substitution regularities, which improves overall data efficiency by reducing the need for labeled data during fine-tuning.

Prot2Token [15] uses a sequence-only design in which a pretrained ESM-2 encoder provides context to a causal decoder via cross-attention, framing binding-site prediction as next-token generation over sorted residue indices. Each ligand type is represented by a learned *task token* given to the decoder, and the model is trained multi-task over 41 ligand classes. Because conditioning is done over a closed vocabulary, the approach is effective for a fixed set of ligands but does not generalize *zero-shot* to unseen ligands without adding new tokens and retraining.

LaMPSite [16] augments ESM-2 sequence embeddings with an estimated contact map and pairs them with a GNN over the ligand's 2D molecular graph (plus a fast conformer-derived distance

map). A residue–atom interaction module integrates protein contacts and ligand distances to refine residue-level binding likelihoods, yielding a sequence-anchored yet ligand-aware predictor.

Across PLM-based approaches, common tactics for class imbalance include weighted losses [13], class-balanced focal objectives with metric-learning terms [9], multi-task sharing [6], and decoder pretraining for stability in few-shot regimes [15]. Conditioning on ligand identity—via task tokens [15] or explicit ligand graphs [16]—further improves specificity over purely *ligand-agnostic* sequence models. We de-emphasize earlier families—traditional ML, non-pretrained deep learning, and template-based methods—because they either train from scratch on limited data (hurting sample efficiency and long-range modeling) or depend on close structural templates (limiting coverage and ligand specificity). Brief summaries of such methods appear in Appendix A.

3 Method

We systematically evaluated the impact of ligand representation, specifically *overrepresented*, *underrepresented*, and *zero-shot* ligands, on protein-ligand binding site prediction. We first established an evaluation framework tailored to these categories, then introduced a three-stage modeling framework that progressively enhances a baseline transformer architecture. Stage 1 focuses on single-ligand prediction; subsequent stages extend to multiple ligands and enable *zero-shot* prediction. This structure supports a systematic comparison of architectural modifications across diverse ligand contexts.

3.1 Data Preparation and Evaluation Setup

We utilized BioLiP2 [17] as our primary dataset, a comprehensive database of biologically relevant protein-ligand binding interactions sourced from the Protein Data Bank (PDB). After applying quality filters and data preprocessing steps detailed in Appendix B, we obtained 5 780 ligands and 41 327 protein sequences. To systematically evaluate model performance across different data availability scenarios, we categorized ligands into three distinct groups based on the number of associated protein-ligand pairs (Figure 1 and Table 1).

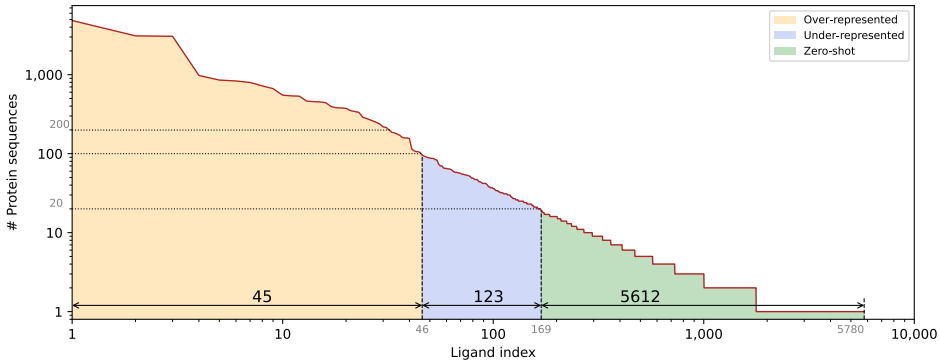


Figure 1: Evaluation strategies showing the distribution of ligands across three categories based on the number of associated protein sequences. Numbers reflect the original dataset before compatibility-based exclusions.

Overrepresented ligands (45 ligands) have more than 100 samples and represent well-studied binding interactions. These were split using standard train-validation-test ratios, with higher training proportions for ligands with abundant data. *Underrepresented* ligands (122 ligands) have 20-99 samples, representing moderately studied interactions. These were split using balanced ratios (34%-33%-33%) to ensure adequate representation across all sets.

Zero-shot ligands (5,612 ligands) have fewer than 20 samples and were reserved exclusively for testing to evaluate the model's ability to generalize to entirely unseen binding interactions. To prevent data leakage between splits, we applied `CD-HIT` [18] clustering with a 40% identity threshold to group similar protein sequences, ensuring that clusters rather than individual sequences were assigned to different sets. This approach maintains the independence of training and evaluation data

Table 1: Dataset split statistics across *overrepresented*, *underrepresented*, and *zero-shot* ligand categories. Numbers marked with † indicate values after compatibility-based ligand exclusions explained in Appendix B.

Category	Unique Ligands	Split ratios (Train-Valid-Test)	Train	Valid	Test
Overrepresented	45	(> 200 pairs) 70%-10%-20% (< 200 pairs) 50%-20%-30%	17 640	2 814	5 492
Underrepresented	122	34%-33%-33%	1 634	1 630	1 658
Zero-shot	5 612 (5 583†)	0%-0%-100%	–	–	10 438 (10 365†)
Total	5 779 (5 750†)	–	19 274	4 444	17 588 (17 515†)

while preserving the biological diversity within each category. Complete details of the clustering procedure and data splitting methodology are provided in Appendix B.

To ensure compatibility across different CLMs, we applied minor modifications to the dataset during integration. Specifically, 29 ligands were excluded from the final evaluation dataset. Complete details of these compatibility adjustments are provided in Appendix B. Although we used the reduced version for our experiments, we release the original unmodified dataset as well to facilitate future research.

3.2 Predictor Architecture

The proposed architecture leverages transformer-based models, specifically utilizing Bert-style pre-trained language models tailored separately for proteins (PLMs) and chemicals (CLMs). PLMs such as ESM-2 [19] provide contextualized residue-level embeddings based solely on protein sequences. CLMs like *MolFormer* [20] are employed to represent ligand structures using their SMILES notation. These pre-trained models provide robust feature representations essential for accurately predicting binding sites.

3.2.1 Stage 1: Baseline Model with Single-Ligand Prediction

Stage 1 establishes our baseline architecture, which predicts binding sites for *one* ligand type at a time. Let p be an input protein sequence of fixed length L. The sequence is tokenized and passed through a *pre-trained* protein language model $G_{\boldsymbol{\theta}}$, whose parameters are denoted by $\boldsymbol{\theta}$. This model outputs contextualised residue-level embeddings $G_{\boldsymbol{\theta}}(p) \in \mathbb{R}^{L\times d}$, where d is the hidden dimension. The embeddings are then fed into a linear predictor (binary classification head) $C_{\boldsymbol{\phi}}$, parameterised by $\boldsymbol{\phi}$, to produce a logit for each residue (Equation 1). At this stage, the architecture operates independently of the ligand-specific context, focusing exclusively on individual ligand predictions without cross-ligand generalization (Figure 4).

$$C_{\boldsymbol{\phi}}\big(G_{\boldsymbol{\theta}}(p)\big) \;\in\; \mathbb{R}^{L\times 2} \tag{1}$$

The calculated loss for a given sample first uses the standard binary cross-entropy (BCE) formulation. To emphasize hard-to-classify residues while correcting for dataset- and class-specific imbalance, we multiply each residue by a sample weight w_i, yielding the weighted loss in (Equation 2). Here $w_i \geq 0$ is the product of a dataset-level weight and, when enabled, a positive-class token weight.

$$\mathcal{L} = -\frac{1}{L}\sum_{i=1}^{L} w_i\Big[\, y_i \log \hat{y}_i + \,(1-y_i)\log\!\big(1-\hat{y}_i\big)\Big] \tag{2}$$

3.2.2 Stage 2: Multi-Ligand Prediction with Ligand-Specific Conditioning

Stage 2 augments the baseline by conditioning on ligand identity, enabling the simultaneous prediction of binding sites for multiple *pre-defined* ligand types. Protein sequences p are encoded by the same $G_{\boldsymbol{\theta}}$ used in Stage 1, producing residue embeddings $G_{\boldsymbol{\theta}}(p) \in \mathbb{R}^{L\times d}$. Ligand information is supplied as an index ℓ that queries a trainable embedding table $E_{\boldsymbol{\eta}}$, parameterized by $\boldsymbol{\eta}$, yielding an

embedding table $E_{\boldsymbol{\eta}}(\ell) \in \mathbb{R}^d$ (analogous to word embeddings, e.g. word2vec [21]). A bidirectional transformer decoder $T_{\boldsymbol{\psi}}$, whose parameters are indicated by $\boldsymbol{\psi}$, integrates protein and ligand representations through cross-attention, after which, similar to stage 1, the linear binary classification head $C_{\boldsymbol{\phi}}$ produces residue level logits (Equation 3). This stage supports all ligand types seen during training, but cannot yet be generalized to novel ligands (Figure 5).

$$C_{\boldsymbol{\phi}}\Big(T_{\boldsymbol{\psi}}\big(G_{\boldsymbol{\theta}}(p),\, E_{\boldsymbol{\eta}}(\ell)\big)\Big) \;\in\; \mathbb{R}^{L\times 2} \tag{3}$$

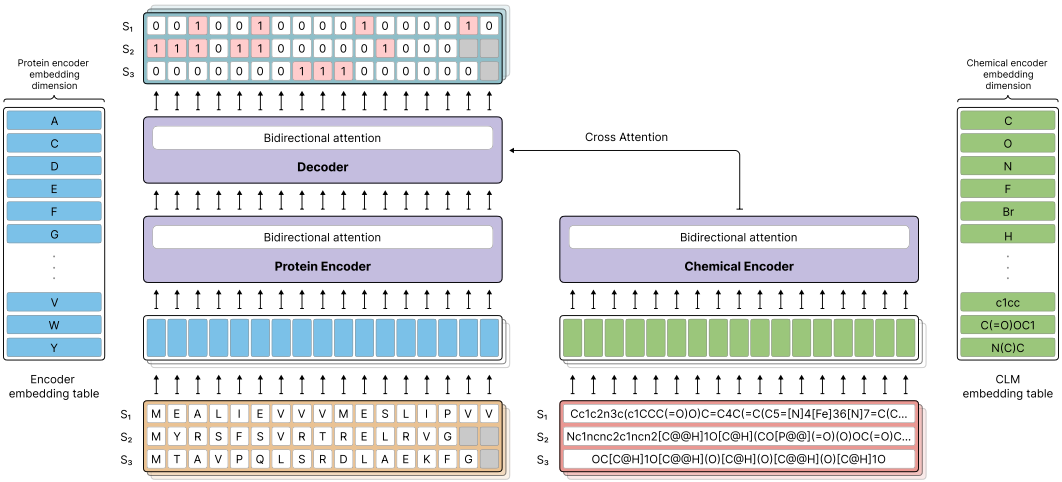


Figure 2: Stage-3 (*zero-shot*) architecture. A protein sequence p of length L is embedded by a pretrained PLM $G_{\boldsymbol{\theta}}$ (left), and the ligand SMILES s with l tokens is embedded by a pretrained CLM $F_{\boldsymbol{\omega}}$ (right). A learnable projector $P_{\boldsymbol{\rho}}$ maps CLM features to the protein feature dimension, and a bidirectional decoder $T_{\boldsymbol{\psi}}$ fuses the two streams via cross-attention; a linear head $C_{\boldsymbol{\phi}}$ then outputs residue-wise binding logits $\hat{\mathbf{y}} \in \mathbb{R}^{L\times 2}$ (Eq. 4). Gray cells indicate padding/masks. Conditioning on SMILES rather than a fixed ligand ID enables inference on unseen ligands (*zero-shot* residue-level prediction).

3.2.3 Stage 3: Zero-Shot Generalization with a Pre-Trained Chemical Language Model

Stage 3 equips the model with *zero-shot* capability by replacing the fixed ligand-embedding mechanism with a pre-trained chemical language model. Protein sequences p are encoded by the same $G_{\boldsymbol{\theta}}$ as previous stages. Each ligand is supplied directly as a SMILES string s and embedded by the chemical language model $F_{\boldsymbol{\omega}}$, whose parameters are denoted by $\boldsymbol{\omega}$, yielding an embedding $F_{\boldsymbol{\omega}}(s) \in \mathbb{R}^{L\times c}$ whose dimensionality c generally differs from the protein feature dimension d. To align these feature spaces we introduce a learnable *linear projector* $P_{\boldsymbol{\rho}} : \mathbb{R}^{l\times c} \to \mathbb{R}^{L\times d}$, parameterised by $\boldsymbol{\rho}$, producing $P_{\boldsymbol{\rho}}\big(F_{\boldsymbol{\omega}}(s)\big) \in \mathbb{R}^{L\times d}$ for the cross-attention layers of the same decoder $T_{\boldsymbol{\psi}}$. The fused representation is then passed to the binary classifier head $C_{\boldsymbol{\phi}}$. Because $F_{\boldsymbol{\psi}}$ is pre-trained on large chemical corpora, the architecture (Equation 4) can technically predict binding sites for entirely new ligand-protein combinations, including those that have not been encountered during training (Figure 2).

$$C_{\boldsymbol{\phi}}\Big(T_{\boldsymbol{\psi}}\big(G_{\boldsymbol{\theta}}(p),\, P_{\boldsymbol{\rho}}\big(F_{\boldsymbol{\omega}}(s)\big)\big)\Big) \;\in\; \mathbb{R}^{L\times 2} \tag{4}$$

4 Experiments

We constructed a base architecture and systematically extended it in two progressive stages to enable comprehensive evaluation across all ligand representation categories. These categories include *overrepresented*, *underrepresented*, and *zero-shot* ligands. Throughout our experiments, we evaluated multiple combinations of pre-trained language models: specifically, PLMs including different scales of ESM-2, and CLMs including *MolFormer* and *UniMol-2* [22].

Optimization procedures employed the AdamW optimizer [23], configured with a weight decay of 0.01, beta-1 and beta-2 parameters set to $\beta_1 = 0.9$, $\beta_2 = 0.98$, respectively, and $\epsilon = 1e - 7$. We implemented a cosine annealing learning rate schedule [24], without incorporating an initial warm-up phase, wherein the learning rate starts from 5e-5 and ends at 0. All experiments were implemented using PyTorch 2.6.0 framework [25], and employed mixed-precision BF16 training [26, 27] on an NVIDIA A100 GPU with 80GB of memory. Detailed hyperparameter configurations are documented in Appendix D.

4.1 Stage 1: Single Ligand

In Stage 1, we trained individual models for each ligand type without using any ligand-specific embeddings or chemical representations. We focused this stage on the 45 *overrepresented* ligands in our dataset and maintained the same hyperparameter setup for each model. The result of each ligand was obtained from a single training run. Training was monitored using a validation set, with the best checkpoint from each run selected based on the peak validation F_1 score. Final evaluations were conducted on the held-out *overrepresented* test set. Need to mention that we only used ESM-2 650m for the PLM of this stage. The results, summarized in Table 7 provide a performance baseline for residue-level binary classification. The overall results are provided in Tables 2 and 4.

Despite uniform training procedures, we observed considerable variation in performance across ligand types. While several ligands such as `FAD` and `ZN` achieved high F_1 scores, others, such as `BMA` and `MAN`, achieved zero or near-zero predictive accuracy. These discrepancies may reflect differences in sample size, but also suggest that certain ligands have inherently more complex binding behaviors.

4.2 Stage 2: Multiple Ligands

For stage 2, we tested the effects of the extended architecture and exposure to multiple ligand types on the performance of the model. The introduction of ligand-specific information, implemented through the embedding table, enabled a single model to be trained on a compiled dataset consisting of multiple ligands. We evaluated this architecture under two training conditions: first, using only the 45 *overrepresented* ligands from Stage 1, and second, using an expanded training set that combined these *overrepresented* ligands with the set of *underrepresented* ligands. For both scenarios, results were obtained by averaging the performance across three training runs using random seeds, reported in Table 2. In both settings, the Stage 2 model showed significantly improved predictive performance over the Stage 1 baseline with a Macro F_1 score of 0.4769, demonstrating the benefits of using multi-task learning. The comparison with Prot2Token on the *overrepresented* test set is shown in Table 4. Need to mention that we only used ESM-2 650m PLM for this stage (Table 5).

Fair benchmarking across sequence- and structure-based predictors is hampered primarily by data contamination: models pretrained or tuned on PDB-derived sequences/complexes can overlap (via homology or near-duplicates) with evaluation sets and inflate reported scores. We mitigate within-dataset leakage by `CD-HIT` clustering at 40% identity (Appendix B), but cross-corpus contamination remains a caveat when comparing to external tools. With that in mind, we include Boltz-2x [28]—a recent, high-performing open-source protein–ligand complex predictor—as a structure-based baseline alongside sequence-only methods. We ran Boltz-2x without multiple sequence alignments (MSAs) and truncated protein sequences to 1280, using only single protein sequences as input, so that its predictions were not advantaged by evolutionary context unavailable to our models. To evaluate the prediction of Boltz-2x at the residue level, we converted each predicted complex into binding-site labels by enumerating intermolecular atomic contacts: a contact is any protein–ligand atom pair whose distance is less than or equal the sum of their van der Waals radii + 0.5 Å; a residue is designated binding if it forms more than 2 such contacts with the ligand. This yields residue-level labels directly comparable to our model's outputs.

Table 2: Benchmarking the three Stages on the test sets. Macro F_1 scores are based on the average of F_1 for each type of ligand.

Method	Training Dataset	Overrepresented		Underrepresented		Zero-shot		
		Accuracy	Macro F_1	Accuracy	Macro F_1	Accuracy	Macro F_1	F_1
Stage 1	Overrepresented (separated)	98.92	0.4769	-	-	-	-	-
Stage 2	Overrepresented	99.33 ± 0.02	0.5826 ± 0.0035	-	-	-	-	-
Stage 2	Overrepresented + Underrepresented	99.35 ± 0.00	0.5832 ± 0.0014	98.87 ± 0.02	0.3752 ± 0.0049	-	-	-
Stage 3	Overrepresented + Underrepresented	99.48 ± 0.00	0.5526 ± 0.0012	99.12 ± 0.01	0.3603 ± 0.0029	99.02 ± 0.01	0.2338 ± 0.0051	0.3109 ± 0.0087

Table 3: Comparison of our method's best performance for each ligand with other available methods on selected ligands in the *overrepresented* test set based on F_1 score. The main values are taken from the original papers, and * indicates methods evaluated on our test sets.

Ligand	Metrics	Stage 2 (Our)	Prot2Token [29]	TargetS [5]	LMetalSite [6]	ZinCap [30]	MIB2 [31]	Boltz-2x [28]
Ca^{2+}	F_1	0.6958 ± 0.0011	0.6566*	0.392*	0.526 (0.7370*)	-	-	0.380*
	MCC	0.6912 ± 0.0014	-	0.320 (0.431*)	0.542 (0.7342*)	-	-	0.406*
	Acc	0.9895 ± 0.0001	-	0.984 (0.977*)	0.9884*	-	0.941	0.982*
Mg^{2+}	F_1	0.5637 ± 0.0036	0.4603*	0.433*	0.367 (0.5560*)	-	-	0.339*
	MCC	0.5640 ± 0.0004	-	0.383 (0.450*)	0.419 (0.5773*)	-	-	0.336*
	Acc	0.9935 ± 0.0001	-	0.990 (0.992*)	0.9949*	-	0.946	0.991*
Zn^{2+}	F_1	0.8180 ± 0.0017	0.7594*	0.660*	0.760 (0.8299*)	0.451*	-	0.557*
	MCC	0.8151 ± 0.0018	-	0.557 (0.660*)	0.761 (0.8275*)	0.540 (0.480*)	-	0.566*
	Acc	0.9943 ± 0.0000	-	0.989 (0.989*)	0.9953*	0.870 (0.970*)	0.948	0.990*
Mn^{2+}	F_1	0.7663 ± 0.0113	0.7376*	0.579*	0.662 (0.8048*)	-	-	0.419*
	MCC	0.7644 ± 0.0114	-	0.445 (0.574*)	0.661 (0.8024*)	-	-	0.417*
	Acc	0.9945 ± 0.0003	-	0.987 (0.989*)	0.995*	-	0.950	0.990*

4.3 Stage 3: Zero-Shot Ligands

In Stage 3, we evaluated the effect of using the MolFormer CLM to generate chemical embeddings, replacing the ligand embedding table from Stage 2 and abstracting ligands through the use of SMILES strings. We trained this architecture using the combined set of *overrepresented* and *underrepresented* training sets. Evaluation was conducted on three test sets and reported in Table 8. The reported results for this stage were obtained by averaging performance across three independent training rounds with randomized seeds. The comparison to Prot2Token on the *overrepresented* test set is shown in Table 4.

Table 4: Comparison of Prot2Token with our approach on the top 40 ligands in the *overrepresented* test set. †: values are reported from the original paper.

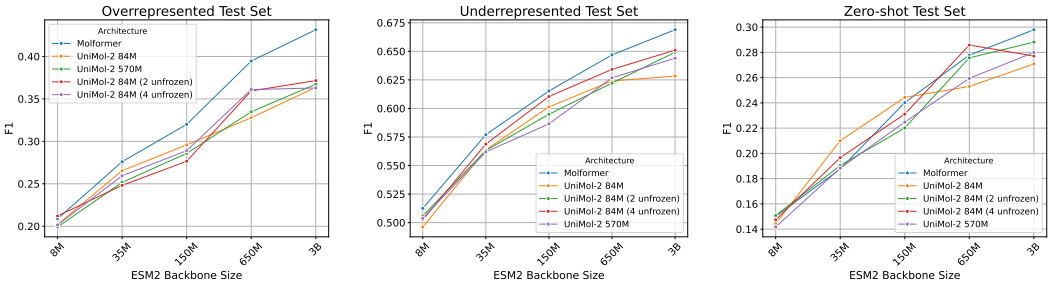
Ligands	Stage 1 (Macro F_1)	Stage 2 (Macro F_1)	Stage 3 (Macro F_1)	Prot2Token (Macro F_1)
Top 10 ligands	0.6696	0.7018 ± 0.0021	0.6657 ± 0.0065	0.6130†
Top 20 ligands	0.6435	0.7165 ± 0.0014	0.6835 ± 0.0049	0.6192†
Top 30 ligands	0.5702	0.6602 ± 0.0016	0.6257 ± 0.0027	0.5615†
Top 40 ligands	0.5113	0.6144 ± 0.0014	0.5828 ± 0.0033	0.5114†

When evaluating with the same set of hyperparameters used for the prior stages, we found that the Stage 3 model underperformed on both the *overrepresented* (0.5526 vs. 0.5832 Macro F_1) and *underrepresented* (0.3603 vs. 0.3752 Macro F_1) test sets compared to Stage 2.

Nevertheless, Stage 3 offered a distinct advantage in its ability to perform *zero-shot* inference, achieving a F_1 score of 0.3109 on the test set of completely new ligands.

4.4 Scaling comparison

To better understand the impact of model scale on protein-ligand binding site prediction, we conducted a series of ablation experiments varying both the size of the PLM backbone and the choice of CLM on the Stage 3 architecture. Specifically, we evaluated model performance across five ESM2 sizes (8M, 35M, 150M, 650M, and 3B), paired with different CLM configurations to assess how scaling affects predictive capability. In addition to varying the backbone size, we compared multiple CLMs, including MolFormer, UniMol-2 (both 84M and 570M parameter variants), and partially unfrozen versions of the 84M UniMol-2 model.



(a) *Overrepresented* test set (b) *Underrepresented* test set (c) *Zero-shot* test set

Figure 3: Results of ablation studies on the three ligand test sets based on F_1: (a) overrepresented, (b) *underrepresented*, and (c) *zero-shot* test sets.

To ensure a fair comparison across architectures, we held the training recipe fixed: similar optimization hyperparameters and the same architecture configurations (Tables 5 and 6). Each configuration was trained with three random seeds, and we report the mean results. As shown in Figure 3, scaling the *protein* encoder dominates performance: larger ESM-2 backbones consistently yield higher Macro F_1 across *overrepresented*, *underrepresented*, and *zero-shot* test sets, largely independent of the paired CLM. In contrast, increasing the *chemical* encoder's scale (e.g., UniMol-2 84M→570M) produces negligible or inconsistent gains. Among CLMs, MolFormer is the most reliable—typically matching or exceeding the UniMol-2 variants with lower variability.

5 Discussion

In this study we asked whether residue-level binding prediction can be made both ligand-aware and *zero-shot* from sequence and SMILES alone. The three-stage progression answers *yes*, and clarifies where the gains come from. Moving from per-ligand models to a single multi-ligand model (Stage 1→Stage 2) delivers a large improvement on *overrepresented* ligands (Macro F_1 0.4769→0.5832; Table 2), while remaining competitive or better than strong sequence baselines across many ions and cofactors (e.g., Zn^{2+} $F_1 = 0.818$ vs. 0.557 for Boltz-2x under residue-label conversion; Table 3). Because our approach avoids 3D inference, it is also highly efficient: empirically, we observe $\geq 100\times$ higher throughput than structure-based pipelines such as Boltz-2x under our evaluation setting, making it well-suited as a front-end, high-throughput virtual screening stage prior to more expensive 3D modeling or docking.

Replacing the lookup embedding with a CLM over SMILES (Stage 3) unlocks *zero-shot* generalization: on the held-out set of unseen ligands the model reaches $F_1 = 0.3109$ (Macro $F_1 = 0.2338$; Table 2). The price is a modest drop on represented ligands relative to Stage 2 (*overrepresented* Macro F_1 0.5526 vs. 0.5832; *underrepresented* 0.3603 vs. 0.3752). Together with the ablation in Fig. 3, which shows consistent gains from scaling the protein encoder (ESM-2) but only small or inconsistent effects from scaling the CLM, these results suggest that richer protein context is the dominant driver of binding-site discrimination, while current cross-modal fusion likely leaves accuracy on the table for seen ligands.

Two factors likely limit headroom. First, label imbalance and noise (99%+ accuracy with far lower F_1) make optimization sensitive to calibration and thresholds, especially for sugars and small polar ligands where Stage 1 underperforms (e.g., `BMA`, `MAN`; Table 7). Second, CLM–PLM alignment is learned through a shallow projector; stronger fusion could both recover the small Stage 3 degradation on seen ligands and actively elevate the importance of chemical representations, allowing the CLM to contribute more decisively (and potentially make CLM scale matter) without sacrificing *zero-shot* ability.

References

[1] Jingtian Zhao, Yang Cao, and Le Zhang. Exploring the computational methods for protein-ligand binding site prediction. *Computational and structural biotechnology journal*, 18:417–

426, 2020.

[2] Ashwin Dhakal, Cole McKay, John J Tanner, and Jianlin Cheng. Artificial intelligence in the prediction of protein–ligand interactions: recent advances and future directions. *Briefings in Bioinformatics*, 23(1):bbab476, 2022.

[3] Ying Xia, Xiaoyong Pan, and Hong-Bin Shen. A comprehensive survey on protein-ligand binding site prediction. *Current Opinion in Structural Biology*, 86:102793, 2024.

[4] Chun-Qiu Xia, Xiaoyong Pan, and Hong-Bin Shen. Protein–ligand binding residue prediction enhancement through hybrid deep heterogeneous learning of sequence and structure data. *Bioinformatics*, 36(10):3018–3027, 2020.

[5] Dong-Jun Yu, Jun Hu, Jing Yang, Hong-Bin Shen, Jinhui Tang, and Jing-Yu Yang. Designing template-free predictor for targeting protein-ligand binding sites with classifier ensemble and spatial clustering. *IEEE/ACM transactions on computational biology and bioinformatics*, 10 (4):994–1008, 2013.

[6] Qianmu Yuan, Sheng Chen, Yu Wang, Huiying Zhao, and Yuedong Yang. Alignment-free metal ion-binding site prediction from protein sequence through pretrained language model and multi-task learning. *Briefings in bioinformatics*, 23(6):bbac444, 2022.

[7] Ying Xia, Chunqiu Xia, Xiaoyong Pan, and Hong-Bin Shen. Bindweb: a web server for ligand binding residue and pocket prediction from protein structures. *Protein Science*, 31(12):e4462, 2022.

[8] Yitian Fang, Yi Jiang, Leyi Wei, Qin Ma, Zhixiang Ren, Qianmu Yuan, and Dong-Qing Wei. Deepprosite: structure-aware protein binding site prediction using esmfold and pretrained language model. *Bioinformatics*, 39(12):btad718, 2023.

[9] Jue Wang, Yufan Liu, and Boxue Tian. Protein-small molecule binding site prediction based on a pre-trained protein language model with contrastive learning. *Journal of cheminformatics*, 16(1):125, 2024.

[10] Gelany Aly Abdelkader and Jeong-Dong Kim. Advances in protein-ligand binding affinity prediction via deep learning: A comprehensive study of datasets, data preprocessing techniques, and model architectures. *Current drug targets*, 25(15):1041–1065, 2024.

[11] Amit Gangwal, Azim Ansari, Iqrar Ahmad, Abul Kalam Azad, and Wan Mohd Azizi Wan Sulaiman. Current strategies to address data scarcity in artificial intelligence-based drug discovery: A comprehensive review. *Computers in Biology and Medicine*, 179:108734, 2024.

[12] Tobias Harren, Torben Gutermuth, Christoph Grebner, Gerhard Hessler, and Matthias Rarey. Modern machine-learning for binding affinity estimation of protein–ligand complexes: Progress, opportunities, and challenges. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 14(3):e1716, 2024.

[13] Maria Littmann, Michael Heinzinger, Christian Dallago, Konstantin Weissenow, and Burkhard Rost. Protein embeddings and deep learning predict binding residues for various ligand classes. *Scientific reports*, 11(1):23916, 2021.

[14] Clement Essien, Lei Jiang, Duolin Wang, and Dong Xu. Prediction of protein ion–ligand binding sites with electra. *Molecules*, 28(19):6793, 2023.

[15] Mahdi Pourmirzaei, Salhuldin Alqarghuli, Farzaneh Esmaili, Mohammadreza Pourmirzaei, Mohsen Rezaei, and Dong Xu. Using autoregressive-transformer model for protein-ligand binding site prediction. *bioRxiv*, pages 2025–03, 2025.

[16] Shuo Zhang and Lei Xie. Protein language model-powered 3d ligand binding site prediction from protein sequence. *arXiv preprint arXiv:2312.03016*, 2023.

[17] Chengxin Zhang, Xi Zhang, Peter L Freddolino, and Yang Zhang. Biolip2: an updated structure database for biologically relevant ligand–protein interactions. *Nucleic Acids Research*, 52 (D1):D404–D412, 2024.

[18] Weizhong Li and Adam Godzik. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, 22(13):1658–1659, 2006.

[19] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

[20] Fang Wu, Dragomir Radev, and Stan Z Li. Molformer: Motif-based transformer on 3d heterogeneous molecular graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 5312–5320, 2023.

[21] Kenneth Ward Church. Word2vec. *Natural Language Engineering*, 23(1):155–162, 2017.

[22] Xiaohong Ji, Zhen Wang, Zhifeng Gao, Hang Zheng, Linfeng Zhang, Guolin Ke, et al. Unimol2: Exploring molecular pretraining model at scale. *arXiv preprint arXiv:2406.14969*, 2024.

[23] I Loshchilov. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[24] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[25] Jason Ansel, Edward Yang, Horace He, Natalia Gimelshein, Animesh Jain, Michael Voznesensky, Bin Bao, Peter Bell, David Berard, Evgeni Burovski, et al. Pytorch 2: Faster machine learning through dynamic python bytecode transformation and graph compilation. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, pages 929–947, 2024.

[26] Dhiraj Kalamkar, Dheevatsa Mudigere, Naveen Mellempudi, Dipankar Das, Kunal Banerjee, Sasikanth Avancha, Dharma Teja Vooturi, Nataraj Jammalamadaka, Jianyu Huang, Hector Yuen, et al. A study of bfloat16 for deep learning training. *arXiv preprint arXiv:1905.12322*, 2019.

[27] Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, et al. Mixed precision training. *arXiv preprint arXiv:1710.03740*, 2017.

[28] Saro Passaro, Gabriele Corso, Jeremy Wohlwend, Mateo Reveiz, Stephan Thaler, Vignesh Ram Somnath, Noah Getz, Tally Portnoi, Julien Roy, Hannes Stark, et al. Boltz-2: Towards accurate and efficient binding affinity prediction. *BioRxiv*, pages 2025–06, 2025.

[29] Mahdi Pourmirzaei, Farzaneh Esmaili, Salhuldin Alqarghuli, Mohammadreza Pourmirzaei, Ye Han, Kai Chen, Mohsen Rezaei, Duolin Wang, and Dong Xu. Prot2token: A unified framework for protein modeling via next-token prediction. *arXiv preprint arXiv:2505.20589*, 2025.

[30] Clement Essien, Duolin Wang, and Dong Xu. Capsule network for predicting zinc binding sites in metalloproteins. In *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 2337–2341. IEEE, 2019.

[31] Chih-Hao Lu, Chih-Chieh Chen, Chin-Sheng Yu, Yen-Yi Liu, Jia-Jun Liu, Sung-Tai Wei, and Yu-Feng Lin. Mib2: metal ion-binding site prediction and modeling server. *Bioinformatics*, 38(18):4428–4429, 2022.

[32] Daniel Barry Roche, Danielle Allison Brackenridge, and Liam James McGuffin. Proteins and their interacting partners: An introduction to protein–ligand binding site prediction methods. *International journal of molecular sciences*, 16(12):29829–29842, 2015.

[33] Jianyi Yang, Ambrish Roy, and Yang Zhang. Protein–ligand binding site recognition using complementary binding-specific substructure comparison and sequence profile alignment. *Bioinformatics*, 29(20):2588–2595, 2013.

[34] Yifeng Cui, Qiwen Dong, Daocheng Hong, and Xikun Wang. Predicting protein-ligand binding residues with deep convolutional neural networks. *BMC bioinformatics*, 20:1–12, 2019.

[35] Xiuzhen Hu, Qiwen Dong, Jianyi Yang, and Yang Zhang. Recognizing metal and acid radical ion-binding sites by integrating ab initio modeling with template-based transferals. *Bioinformatics*, 32(21):3260–3269, 2016.

[36] Jun Hu, Yang Li, Yang Zhang, and Dong-Jun Yu. Atpbind: accurate protein–atp binding site prediction by combining sequence-profiling and structure-based comparisons. *Journal of chemical information and modeling*, 58(2):501–510, 2018.

[37] Yu-Feng Lin, Chih-Wen Cheng, Chung-Shiuan Shih, Jenn-Kang Hwang, Chin-Sheng Yu, and Chih-Hao Lu. Mib: metal ion-binding site prediction and docking server. *Journal of chemical information and modeling*, 56(12):2287–2291, 2016.

[38] Ying Xia, Chun-Qiu Xia, Xiaoyong Pan, and Hong-Bin Shen. Graphbind: protein structural context embedded rules learned by hierarchical graph neural networks for recognizing nucleic-acid-binding residues. *Nucleic acids research*, 49(9):e51–e51, 2021.

[39] Clement Essien, Ning Wang, Yang Yu, Salhuldin Alqarghuli, Yongfang Qin, Negin Manshour, Fei He, and Dong Xu. Predicting the location of coordinated metal ion-ligand binding sites using geometry-aware graph neural networks. *Computational and Structural Biotechnology Journal*, 27:137–148, 2025.

[40] Vineeth R Chelur and U Deva Priyakumar. Birds-binding residue detection from protein sequences using deep resnets. *Journal of Chemical Information and Modeling*, 62(8):1809–1818, 2022.

[41] Mark L Benson, Richard D Smith, Nickolay A Khazanov, Brandon Dimcheff, John Beaver, Peter Dresslar, Jason Nerothin, and Heather A Carlson. Binding moad, a high-quality protein–ligand database. *Nucleic acids research*, 36(suppl_1):D674–D678, 2007.

[42] Tiqing Liu, Yuhmei Lin, Xin Wen, Robert N Jorissen, and Michael K Gilson. Bindingdb: a web-accessible database of experimentally determined protein–ligand binding affinities. *Nucleic acids research*, 35(suppl_1):D198–D201, 2007.

[43] Helen M Berman, John Westbrook, Zukang Feng, Gary Gilliland, Talapady N Bhat, Helge Weissig, Ilya N Shindyalov, and Philip E Bourne. The protein data bank. *Nucleic acids research*, 28(1):235–242, 2000.

[44] Paul CD Hawkins, A Geoffrey Skillman, and Anthony Nicholls. Comparison of shape-matching and docking as virtual screening tools. *Journal of medicinal chemistry*, 50(1):74–82, 2007.

A Extended Related Work

In this section, we categorize existing protein–ligand binding site prediction studies into distinct groups. For each study, we briefly describe two key aspects: the structure of the proposed method and the strategy used to address the class imbalance between binding and non-binding sites, which remains one of the main challenges in developing effective deep learning models for this task. While some previous works have grouped prediction methods into sequence and structure-based approaches [32, 3], our focus is primarily on sequence-based methods. We further classify these studies into three groups based on the techniques used and the order in which they were developed. The earliest approaches applied traditional machine learning methods [5, 33], followed by deep learning-based methods [34, 4], and more recently, pretrained models-based methods [6, 16]. We also include some studies that combine sequence and structure-based methods due to their novelty in extracting sequence features or addressing the class imbalance problem, as these provide valuable insights and broaden the scope for the reader.

A.1 Traditional Machine Learning Methods

Yu et al. [5] proposed TargetS, which integrates multiple features for binding residue prediction, including position-specific scoring matrices (PSSMs) to capture protein evolutionary information, predicted secondary structures obtained using PSIPRED, and ligand-specific binding propensities. The ligand-specific propensities are calculated based on the frequencies of amino acids among binding residues for each ligand type. A 17-dimensional window centered at the target residue is used to capture not only the residue's own propensity but also the influence of its local environment. To enhance prediction accuracy, the authors employed a modified AdaBoost (MAdaBoost) ensemble scheme that combines multiple classifiers. To address the class imbalance between binding and non-binding residues, they applied a random undersampling strategy within the AdaBoost framework.

Yang et al. [33] proposed TM-SITE and S-SITE, two template-based methods that leverage structural and sequence similarity, respectively, to predict ligand-binding residues. TM-SITE performs structural comparisons by aligning the subsequence spanning from the first to the last binding residue (SSFL) of the query protein against a pre-calculated SSFL library using TM-align. Proteins with the highest alignment scores are selected as putative templates. A secondary scan is then conducted using the entire query structure against the BioLiP database. Final predictions are obtained through a clustering and voting scheme. S-SITE, in contrast, relies on sequence-based alignment. PSI-BLAST is used to generate multiple sequence alignments for the query protein, from which a position-specific frequency matrix (PSFM) is derived. Template profiles in the BioLiP library, represented as position-specific scoring matrices (PSSMs), are precomputed in advance. The PSFM of the query is then compared to these template PSSMs using sequence alignment techniques, and a voting scheme is applied to determine the final binding residues.

Hu et al. [35] introduced two ligand-specific methods for predicting ion-binding sites. IonSeq is a sequence-based approach that employs sequence profiles together with a modified AdaBoost algorithm to distinguish between binding and non-binding residues. IonCom is a composite method that combines IonSeq with multiple template-based predictors, including COFACTOR, TM-SITE, S-SITE, and COACH, to improve prediction accuracy and robustness. To address class imbalance and reduce the risk of overfitting, both methods use a modified AdaBoost framework that applies selective random sampling to negative (non-binding) residues while fully utilizing all positive (binding) residues in every training round.

Hu et al. [36] proposed ATPbind, a method that combines sequence-based features, including position-specific scoring matrices (PSSMs), predicted secondary structure, and solvent accessibility, with the outputs of S-SITE and TM-SITE, which are sequence- and structure-template-based predictors, respectively. To address the imbalance between binding and non-binding sites, ATPbind employs multiple support vector machines (SVMs) trained with a random undersampling strategy and integrates their outputs using a mean-ensemble approach.

Lin et al. [37] presents the MIB webserver, which utilizes available templates from the Protein Data Bank (PDB) to identify binding sites for 12 types of metal ions. The method compares the structure of query proteins with these templates without any data training. Metal ion–binding residue templates are extracted from the PDB, and homologous proteins are filtered out to ensure diversity.

The server aligns the query protein with the templates and assigns a binding score based on sequence and structure similarity. MIB also predicts the docking positions of metal ions in protein structures.

A.2 Deep Learning-Based Methods

Cui et al. [34] introduced DeepCSeqSite (DCS-SI), a novel sequence-based approach for predicting protein–ligand binding residues. DCS-SI is built on a deep convolutional neural network with an encoder–decoder architecture. The encoder transforms entire amino acid sequences into hierarchical representations that capture both local and long-range dependencies between residues. The decoder then uses these features to predict binding residues. To address the class imbalance issue, the authors did not apply explicit oversampling or undersampling techniques. Instead, they leveraged the strong representation capability of their deep convolutional network and processed entire amino acid sequences as input units during mini-batch grouping. This approach preserved the natural proportion of positive (binding) and negative (non-binding) samples within each mini-batch, reducing the risk of batches dominated by negative samples and enabling the model to learn from a more representative distribution.

Xia et al. [4] proposed DELIA, a hybrid deep learning model that combines 1D sequence-based features with a 2D structure-based distance matrix representing the Euclidean distances between residues in a protein structure. The model is composed of three main modules: a feature extractor, a ResNet module, and a BiLSTM module. The feature extractor utilizes various tools such as PSI-BLAST, HHblits, SCRATCH-1D, and S-SITE to generate features, which are then concatenated. The ResNet and BiLSTM modules are employed to extract high-level representations, and their outputs are concatenated and passed through a softmax layer to generate the final predictions. To address the class imbalance issue, the authors applied a combination of random undersampling and oversampling. For undersampling, they constructed multiple training subsets, each containing all positive samples and 20% of the original negative samples. For oversampling, they ensured that each negative sample was used only once per epoch, while positive samples were randomly selected and reused multiple times across mini-batches.

Essien et al. [30] utilized a Capsule Network (CapsNet) for predicting zinc-binding sites. The model processes protein sequences using a sliding window approach, where each window consists of 25 amino acids. These fixed-length segments are extracted from longer sequences and processed individually. Convolutional layers extract increasingly abstract features, while the PrimaryCaps layer captures higher-level representations. Dynamic routing further refines these features, and the output layer uses them to predict zinc-binding residues. To address class imbalance, the authors applied a bootstrapping strategy. Specifically, in each iteration, a deep learning classifier was trained on a balanced subset of the data, containing all positive samples and a portion of negative samples. This process was repeated multiple times to ensure that all negative samples were used across different iterations, resulting in several independent classifiers. The final prediction was obtained by averaging the outputs of these classifiers.

Xia et al. [38] proposed GraphBind, a model that leverages both structure-based and sequence-based features to identify binding residues. GraphBind consists of three main modules: feature extraction, structural context extraction, and graph construction. In the feature extraction module, both sequence and structure-based features are collected. Sequence features include position-specific scoring matrices (PSSMs) and hidden Markov models (HMMs), generated using PSI-BLAST and HHblits. Structure features include atomic properties and secondary structure (SS) information derived from DSSP analysis. The structural context extraction module defines a local environment around each target residue using a sliding sphere, creating pseudo-positions that represent the 3D context of the residue. In the graph construction module, each residue is treated as a node. Node feature vectors, a distance matrix, an adjacency matrix, and edge feature vectors are used to build the graph. The resulting graph-based representations are then passed to a classifier that determines whether a residue is part of a binding site. To address the class imbalance issue, the authors first applied BL2SEQ and TM-align to assess sequence identity and structural similarity between protein chain pairs. Chains with sequence identity > 0.8 and TM-score > 0.5 were clustered together. Binding site annotations from chains within each cluster were transferred to the chain with the largest number of residues. Finally, `CD-HIT` was used to remove redundant sequences, reducing sequence identity in the training set to below 30%.

Xia et al. [7] presented BindWeb, a model that integrates predictions from GraphBind and DELIA, using mean shift clustering to identify binding residues. The binding scores predicted by GraphBind and DELIA are averaged to generate the final prediction. For each residue, the averaged binding score from both methods is compared to an averaged threshold (derived from the individual thresholds of GraphBind and DELIA) to classify the residue as binding or non-binding. To address the data imbalance issue, DELIA employs a random undersampling-based ensemble strategy.

Essien et al. [39] proposed GPred, a model with four main modules for predicting the location of coordinated metal ion–ligand binding sites in proteins. The Point Neighborhood Grouper finds the nearby atoms around each atom in the protein's point cloud. The Point Transformer combines information about atomic properties, shapes, and evolutionary data, using self-attention and position encoding. The Residual Pooler changes the information from the atom level to the residue level using a MaxPooling layer, helping the model see how atoms work together in the protein. The Classifier then decides if each residue is a binding site or not. The authors used a weighted binary cross-entropy (BCE) loss function to reduce the effect of data imbalance.

Chelur and Priyakumar [40] proposed BiRDS, a deep Residual Neural Network designed to predict a protein's most active binding site using only sequence information. It extracts features such as token embeddings, positional embeddings, segment embeddings, Position-Specific Scoring Matrix, information content, secondary structure, and solvent accessibility, which are combined through simple concatenation to form the input feature vector. This vector is fed into the BiRDS model for classification, and to address the imbalance between binding and non-binding residues, the model is trained using a weighted binary cross-entropy loss function.

A.3 Pretrained Model-Based Methods

Yuan et al. [6] proposed LMetalSite, a method in which the protein sequence is input into a pretrained protein language model to generate embedding representations. To prevent overfitting during training, Gaussian noise is added to these embeddings. The noise-augmented embeddings are then passed through shared transformer networks composed of multiple layers. The output from these shared layers is fed into ion-specific multilayer perceptrons (MLPs), each tailored to predict the binding patterns of one of four different metal ions. To address the limited availability of training data and to capture shared patterns among different ions, the method adopts a multi-task learning framework.

Essien et al. [14] proposed IonPred, a two-part framework for ion-binding site prediction. The first component is a MLM generator, which takes protein sequences with certain amino acids masked and predicts the original residues based on the contextual information from surrounding residues. The second component is an ELECTRA-based discriminator, which evaluates the full sequence and determines whether each amino acid is original or replaced by the generator. To generate training samples, the frequency distribution of all 20 amino acids was computed for each ligand type to identify candidate residues. A sliding window centered on each candidate residue was then used to define positive and negative samples.

Pourmirzaei et al. [15] proposed Prot2Token, a model that combines a protein language model with an autoregressive transformer. The model takes protein sequences and task tokens (representing ligand types) as input to predict the residue indices of binding sites. The protein encoder processes the input sequence, and its output is combined with learnable positional embeddings before being projected. This projected context, together with task-specific token embeddings, is used by the decoder to generate autoregressive outputs for binding site prediction. The decoder is pre-trained in a self-supervised way to provide initial weights, which is especially helpful when the number of training samples is small.

Littmann et al. [13] proposed bindEmbed21, which has two main components: bindEmbed21DL (deep learning) and bindEmbed21HBI (homology-based inference). bindEmbed21DL uses embeddings from the Transformer-based protein language model ProtT5 as input, with a relatively shallow two-layer convolutional neural network (CNN) as its core. bindEmbed21HBI is based on the idea that proteins with high sequence similarity are often evolutionarily related and therefore share similar functions, including binding sites. The final bindEmbed21 method combines the strengths of both bindEmbed21DL and bindEmbed21HBI. To address class imbalance, a weighted cross-entropy loss function was used, with individual weights assigned to each ligand class.

Wang et al. [9] proposed CLAPE-SMB, a method designed to predict the probability of general small molecule binding sites on a protein, rather than binding sites for a specific ligand. The model uses ESM-2 to extract informative features relevant to binding sites from the input protein sequence. The weights of ESM-2 are kept fixed during training to avoid the high computational cost of fine-tuning and to prevent catastrophic forgetting. The backbone of CLAPE-SMB is a multi-layer perceptron (MLP) composed of five fully connected layers, which uses the features extracted by ESM-2 to classify residues as binding or non-binding for small molecules. To address class imbalance, the model employs a combined loss function that integrates class-balanced focal loss and triplet center loss.

Zhang and Xie [16] proposed LaMPSite, which takes a protein's amino acid sequence and a ligand's 2D molecular graph as input. The protein sequence is processed by the ESM-2 protein language model to generate residue embeddings and a contact map indicating which residues are likely close in 3D space. The ligand's molecular graph is processed by a graph neural network (GNN) to produce atom embeddings and a distance map estimating atom–atom distances from a quick 3D conformer. Each residue vector is then paired with each atom vector to create an interaction map, which is refined by an interaction module using both the protein contact map and ligand distance map. For each residue, interaction scores are averaged across all ligand atoms to estimate binding likelihood, and nearby high-scoring residues are grouped based on the protein contact map into predicted binding pockets.

B Data Pre-Processing

BioLiP2 [17] was utilized in this work as a comprehensive and curated database for biologically relevant protein–ligand binding interactions. It primarily sources data from the Protein Data Bank (PDB), supplemented with annotations from literature and other specialized databases such as Binding MOAD [41] and BindingDB [42]. Three main files were used: `BioLiP_nr.txt.gz`, which provides annotations for each ligand–protein interaction site, including binding site residues with PDB residue numbering, binding site residues re-numbered starting from 1, and additional information such as structure resolution and binding affinity; `protein_nr.fasta.gz`, which contains protein receptor sequences clustered at a 90% identity cutoff, formatted in FASTA where lines starting with "$>$" represent sequence IDs (e.g., $>$1hv2A) followed by amino acid sequences; and `ligand.tsv.gz`, which provides a ligand summary including the Chemical Component Dictionary (CCD), chemical formula, SMILES strings, and other identifiers. These resources were used to retrieve protein sequences, ligand SMILES representations, and binding site information for subsequent analysis.

We removed any sequences containing fewer than 50 residues and excluded ligands without SMILES representations. Binary labels representing binding and non-binding sites were generated based on the `binding_residues_renum` column, which contains the positions of binding sites. For each sequence, an equal number of zeros was initially assigned, and then the positions corresponding to binding sites were set to ones, representing binding residues, while the remaining positions retained zeros to indicate non-binding sites.

First, the ligands were categorized based on the number of associated samples:

- **Overrepresented**: ligands with more than 100 samples
- **Underrepresented**: ligands with fewer than 100 samples
- **Zero-shot**: ligands with fewer than 20 samples

After categorizing the ligands and their corresponding samples, we generated a separate FASTA file for each ligand. Each FASTA file contained all sequences that bind to that ligand, with the sequence ID followed by the amino acid sequence.

Next, we applied `CD-HIT` to each ligand's FASTA file individually, clustering the sequences into groups of similar sequences using a 40% identity threshold.

The resulting clusters were then assigned to the training, validation, and testing sets according to the predefined split ratios. Finally, the *overrepresented* and *underrepresented* training sets were combined into a single training set, and their validation sets were merged into one validation set.

For testing, we maintained three separate test sets: the *overrepresented* test set, the *underrepresented* test set, and the *zero-shot* test set.

The Chemical Component Dictionary (CCD) was used to retrieve SMILES strings for each ligand from `ligand.tsv.gz`. This file contains multiple SMILES variants, although the BioLiP2 documentation does not specify what each variant represents. For consistency, we used the first SMILES string provided. A comparison with RCSB PDB [43] confirmed that this first entry corresponds to the standard SMILES and Canonical SMILES generated by OpenEye OEToolkits.

We ended up with 5 780 ligands and 41 327 sequences. In order to have a systematic evaluation, we divided them into three sets. The first category includes *overrepresented* ligands, we broke it into two parts:

1) Defined as those that bind to at least 200 protein sequences. This contained 32 ligands, which were split into training, validation, and testing sets using a 70%, 10%, and 20% ratio, respectively.

2) Defined as those that bind to fewer than 200 sequences but at least 100 sequences. This category contained 13 ligands, and their data were split into training, validation, and testing sets using a 50%, 20%, and 30% ratio, respectively.

The second category includes *underrepresented* ligands, defined as those that bind to fewer than 100 sequences but at least 20 sequences. This category contained 123 ligands, and their data were split into training, validation, and testing sets using a 34%, 33%, and 33% ratio, respectively.

`CD-HIT` clusters sequences into groups based on similarity, ensuring that all sequences within a cluster are similar up to a specified threshold, and no sequence in one cluster is similar (beyond that threshold) to sequences in another cluster. After clustering, the next step is to assign clusters to the training, validation, and testing sets according to the predefined splitting ratios. However, for ligands with a small number of highly similar sequences, `CD-HIT` may produce fewer than three clusters. In such cases, there are not enough clusters to split across training, validation, and testing sets properly. This occurred for only one ligand in the *underrepresented* category (`CL0`), which is why it was excluded. For the *zero-shot* category, we did not use `CD-HIT` because this category was reserved exclusively for the testing phase.

The third category is the *zero-shot* category, which includes ligands that bind to fewer than 20 sequences but at least one. For this category, 100% of the sequences were used for testing, resulting in 5 612 ligands.

During integration with the UniMol-2 CLMs, we applied minor changes to the dataset to ensure compatibility. UniMol-2's built-in SMILES parser imposed strict parsing restraints compared to the MolFormer CLM. To standardize ligand representation in different CLMs, we attempted to use a standard canonical SMILES generated by OpenEye OEToolkits [44]. However, some SMILES representations under this format were rejected by the UniMol-2 parser. In such cases, we attempted to substitute alternative SMILES variants. These alternatives were selected from the additional SMILES strings listed in the `ligand.tsv.gz` file. If no alternatives were accepted or available, the ligand was excluded from the dataset. A ligand, `TRP`, appeared in the *underrepresented* sets, required SMILES modification to be parsed successfully. In the *zero-shot* test set, 88 ligands required SMILES substitutions in order to be parsed, while 29 ligands had no valid alternatives and were excluded from the *zero-shot* test set at the end.

The excluded ligands from the *zero-shot* test set included `08T`, `7RZ`, `8MO`, `AOH`, `B1M`, `COB`, `D6N`, `DVT`, `GCR`, `ICS`, `K6G`, `KEG`, `MO7`, `NFV`, `OEX`, `OZN`, `PQJ`, `R5Q`, `S5Q`, `S9F`, `SIW`, `TEW`, `UOO`, `UZC`, `V9G`, `WJS`, `WO2`, `X3P`, and `ZRW`.

C Architecture

Table 5: Key hyperparameters for Stage 2 and Stage 3 architectures across different model scales.

Protein encoder	ESM-2 8m	ESM-2 35m	ESM-2 150m	ESM-2 650m	ESM-2 3b
Protein encoder dimension	320	480	640	1280	2560
Decoder dimension	320	480	640	1280	2560
Decoder blocks			4		
Decoder heads			8		
Decoder FF size			2048		
Decoder activation			GELU		

Table 6: Key hyperparameters for Stage 3 architectures across different CLM scales.

Chemical encoder	MolFormer (47m)	UniMol-2 (84m)	UniMol-2 (570m)
Encoder dimension	768	768	1536

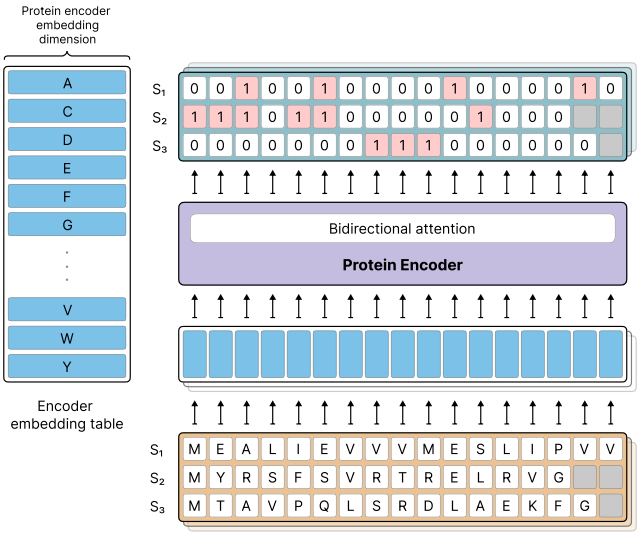


Figure 4: Stage-1 (single-ligand, sequence-only) baseline. A protein sequence p is tokenized and encoded by a pretrained PLM $G_{\boldsymbol{\theta}}$ to yield residue embeddings $G_{\boldsymbol{\theta}}(p) \in \mathbb{R}^{L \times d}$. A linear head $C_{\boldsymbol{\phi}}$ produces residue-wise binding logits $\hat{\mathbf{y}} \in \mathbb{R}^{L \times 2}$ (Eq. 1). No ligand conditioning is used; a separate model is trained per ligand type. Gray cells indicate padding/masks.

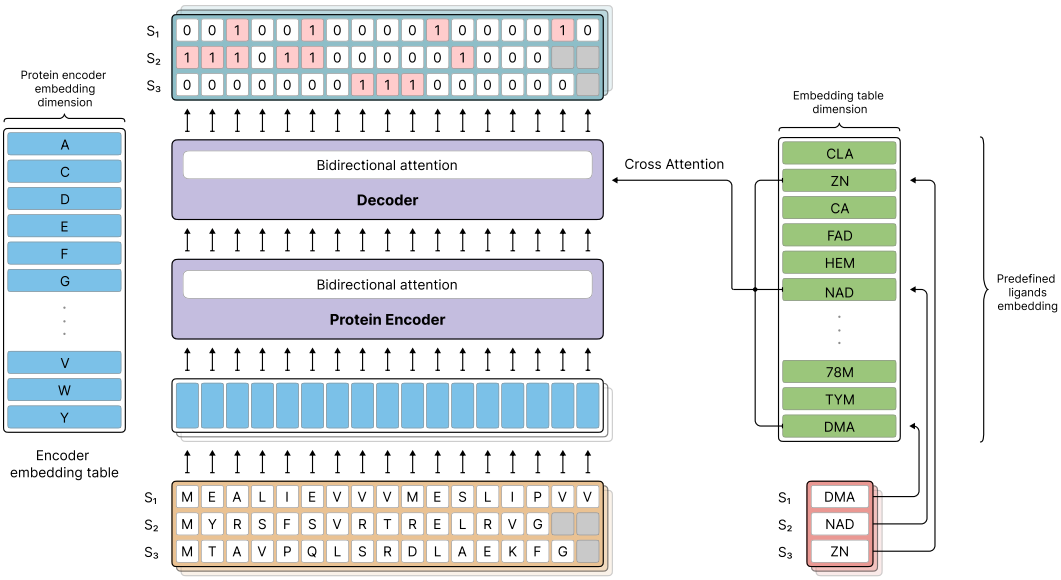


Figure 5: Stage-2 (multi-ligand with learned ligand embedding). The protein is encoded by $G_{\boldsymbol{\theta}}$ as in Stage 1. Ligand identity ℓ indexes a trainable embedding table $E_{\boldsymbol{\eta}}(\ell) \in \mathbb{R}^d$. A bidirectional decoder $T_{\boldsymbol{\psi}}$ fuses protein and ligand signals via cross-attention, and the classifier $C_{\boldsymbol{\phi}}$ outputs residue-wise logits (Eq. 3). This design supports all ligands observed during training but does not generalize to novel ligands. Gray cells indicate padding/masks; the right panel sketches the predefined ligand set.

D Experiments

Table 7: Benchmark of stage 1 architecture on the ligands of the *overrepresented* test set.

Ligand	F_1	**Accuracy**
ACO	0.5290	0.9878
ADP	0.6351	0.9902
AGS	0.5198	0.9893
AMP	0.2971	0.9871
ANP	0.5012	0.9834
ATP	0.5672	0.9864
BGC	0.3015	0.9902
BMA	0.0000	0.9954
Ca^{2+}	0.6834	0.9951
CHL	0.3651	0.9753
CLA	0.5615	0.9701
CLR	0.0000	0.9941
CO	0.5175	0.9953
COA	0.3712	0.9844
Cu^{2+}	0.7131	0.9960
FAD	0.7604	0.9853
Fe^{3+}	0.6908	0.9959
Fe^{2+}	0.6226	0.9954
FES	0.6481	0.9919
FMN	0.7252	0.9878
FUC	0.0000	0.9957
GAL	0.3359	0.9904
GDP	0.7140	0.9902
GLC	0.3052	0.9896
GOL	0.0219	0.9922
GSH	0.6271	0.9901
GTP	0.5530	0.9886
HEC	0.6713	0.9722
HEM	0.6675	0.9865
MAN	0.0171	0.9950
Mg^{2+}	0.5906	0.9971
Mn^{2+}	0.6044	0.9942
NAD	0.6904	0.9856
NAP	0.7151	0.9852
NDP	0.6444	0.9781
PLP	0.7432	0.9921
PO_4^{3-}	0.3733	0.9928
SAH	0.6334	0.9867
SAM	0.5323	0.9837
SF4	0.6321	0.9902
SIA	0.1011	0.9931
SO_4^{2-}	0.0000	0.9930
UDP	0.4515	0.9869
Y01	0.0000	0.9897
Zn^{2+}	0.8245	0.9978

Table 8: Ligand-based F_1 comparison of Prot2Token with our three approaches on the *overrepresented* test set. All values of Prot2Token are reported from the original paper.

Ligands	**Stage 1**	**Stage 2**	**Stage 3**	**Prot2Token**
Zn^{2+}	0.8245	0.8180	0.7747	0.7575
Ca^{2+}	0.6834	0.6948	0.6239	0.6474
CLA	0.5615	0.5740	0.5252	0.4762
FAD	0.7604	0.7833	0.7667	0.6537
HEM	0.6675	0.6922	0.6859	0.6796
NAD	0.6904	0.7515	0.7118	0.6952
ADP	0.6351	0.6930	0.6801	0.5834
Mg^{2+}	0.5906	0.5637	0.4985	0.4575
NAP	0.7151	0.7696	0.7424	0.6746
ATP	0.5672	0.6775	0.6478	0.5050
Average (top 10)	**0.6696**	**0.7018**	**0.6657**	**0.6130**
HEC	0.6713	0.7525	0.7137	0.6537
SF4	0.6321	0.7750	0.7086	0.5685
FMN	0.7252	0.7966	0.8153	0.6945
SAH	0.6334	0.7553	0.7134	0.6503
NDP	0.6444	0.8076	0.7859	0.6979
ANP	0.5012	0.7078	0.6777	0.6217
GDP	0.7140	0.8035	0.7862	0.6465
GLC	0.3052	0.3166	0.3058	0.2214
PLP	0.7432	0.8311	0.8184	0.7620
Mn^{2+}	0.6044	0.7663	0.6876	0.7376
Average (top 20)	**0.6435**	**0.7165**	**0.6835**	**0.6192**
COA	0.3712	0.5134	0.4969	0.4011
SAM	0.5323	0.7022	0.6680	0.6252
AMP	0.2971	0.5306	0.5011	0.4432
BGC	0.3015	0.3514	0.3223	0.1932
Fe^{3+}	0.6908	0.7993	0.7551	0.6606
MAN	0.0171	0.1538	0.0822	0.1216
FES	0.6481	0.7693	0.7167	0.7018
PO_4^{3-}	0.3733	0.3970	0.3178	0.2278
GTP	0.5530	0.6445	0.6412	0.5461
UDP	0.4515	0.6158	0.5994	0.5391
Average (top 30)	**0.5702**	**0.6602**	**0.6257**	**0.5615**
Cu^{2+}	0.7131	0.6893	0.6328	0.5607
GSH	0.6271	0.7759	0.7555	0.6924
AGS	0.5198	0.6785	0.6914	0.5301
ACO	0.5290	0.6391	0.6034	0.5026
GAL	0.3359	0.4616	0.4649	0.2762
SO_4^{2-}	0.0000	0.1807	0.2049	0.1386
CLR	0.0000	0.0568	0.0449	0.0373
Y01	0.0000	0.0981	0.0377	0.0419
BMA	0.0000	0.2927	0.2857	0.2273
Fe^{2+}	0.6226	0.8965	0.8213	0.6033
CO	0.5175	0.6510	0.7048	0.5170
Average (all)	**0.5113**	**0.6153**	**0.5828**	**0.5115**