JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

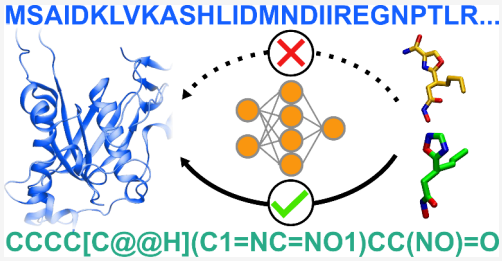
pubs.acs.org/jcim
Review

Natural Language Processing Methods for the Study of Protein–Ligand Interactions

James Michels, Ramya Bandarupalli, Amin Ahangar Akbari, Thai Le, Hong Xiao,* Jing Li,* and Erik F. Y. Hom*

Cite This: J. Chem. Inf. Model. 2025, 65, 2191–2213
Read Online

ABSTRACT: Natural Language Processing (NLP) has revolutionized the way computers are used to study and interact with human languages and is increasingly influential in the study of protein and ligand binding, which is critical for drug discovery and development. This review examines how NLP techniques have been adapted to decode the "language" of proteins and small molecule ligands to predict protein–ligand interactions (PLIs). We discuss how methods such as long short-term memory (LSTM) networks, transformers, and attention mechanisms can leverage different protein and ligand data types to identify potential interaction patterns. Significant challenges are highlighted including the scarcity of high-quality negative data, difficulties in interpreting model decisions, and sampling biases in existing data sets. We argue that focusing on improving data quality, enhancing model robustness, and fostering both collaboration and competition could catalyze future advances in machine-learning-based predictions of PLIs.



1. INTRODUCTION

The study of protein–ligand interactions (PLIs) lies at the heart of cellular function and regulation, orchestrating a complex interplay of molecular processes essential for life. These interactions govern fundamental biological activities, including enzyme catalysis,[1,2] cellular signaling,[3] membrane transport,[4] immune response[5] and transcription factor regulation.[6] At the molecular level, PLIs control cellular homeostasis through metabolic feedback loops,[7] facilitate signal transduction cascades across membranes,[3] mediate immune system recognition of foreign molecules,[5] and regulate gene expression through the control of ligand-dependent activity of transcription factors.[6] The remarkable specificity of these interactions is achieved through a combination of structural complementarity and physicochemical properties and enables precise control of cellular functions.

Understanding PLIs has become instrumental in modern drug discovery development,[8,9] providing a rational framework for designing drugs with maximal efficacy and minimal side effects. Structure-based drug design efforts optimize lead compound development through the strategic modification of chemical groups to enhance binding affinity and specificity.[10] Beyond pharmaceutical applications, protein–ligand engineering efforts are revolutionizing both agricultural biotechnology and industrial bioprocessing. For instance, the engineering of crop proteins has led to improved nutrient utilization efficiency[11] and studies of protein–ligand interactions have been crucial in the development of enzymes with enhanced catalytic activity.[12]

Experimentally, methods like X-ray crystallography and cryo-electron microscopy[13] provide atomic-resolution structural information while biophysical approaches like isothermal titration calorimetry and surface plasmon resonance can provide binding thermodynamic and kinetic data for protein–ligand interactions.[14] Although these experimental methods provide high-quality data benchmarks,[15] they are typically resource- and labor-intensive and thus low-throughput. Computational approaches that simulate the underlying physics and chemistry of PLIs, such as molecular docking[16] or dynamics simulations,[17] can be less resource intensive but nevertheless demand significant computational and time investment.[18]

Recent advancements in machine learning (ML) and deep learning have opened new avenues for effective PLI prediction by leveraging large-scale data sets. ML-based approaches can rapidly assess compound–protein pairs by \learning' from diverse biochemical, topological, and physicochemical properties[19−23] at a pace far quicker than that using traditional methods. ML models have already delivered promising predictive performance for drug–target interaction and binding affinity, supporting early stage target identification

Received: October 22, 2024
Revised: February 5, 2025
Accepted: February 6, 2025
Published: February 24, 2025

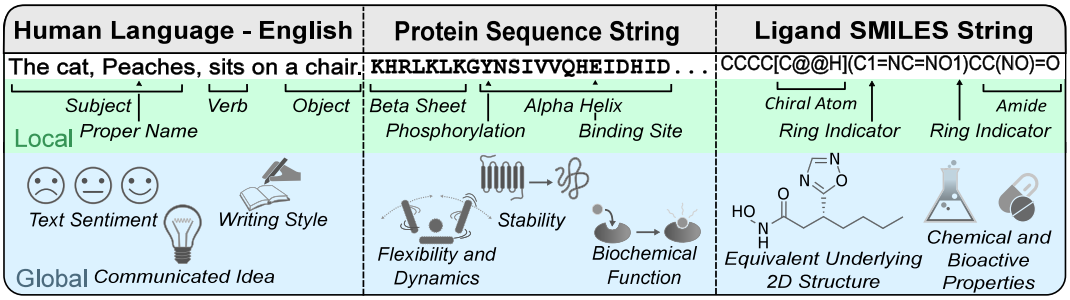


Figure 1. Language of protein sequences and the ligand SMILES representation: NLP methods can be applied to text representations to infer local and global properties of human language, proteins, and molecules alike. Local properties are inferred from subsequences in text: (*left*) for human language, this includes a part of speech or role a word serves; (*middle*) for protein sequences, this includes motifs, functional sites, and domains; and (*right*) for SMILES strings, this can include functional groups and special characters used in SMILES syntax to indicate chemical attributes. Similarly, global properties can theoretically be inferred from a text in its entirety.

and lead optimization.[24,25] As the excitement for ML use in the biological sciences grows, the prediction of protein−ligand interactions appears increasingly possible given recent advances in both ML and Natural Language Processing (NLP),[26,27] the computational study of language.[28]

NLP centers on the computational analysis and manipulation of language constructs[28] to bridge the gap between human communication and computer automation. NLP has experienced significant recent breakthroughs as demonstrated by the proliferation of widely used chatbots such as OpenAI's ChatGPT,[29,30] Anthropic's Claude,[31] and Microsoft's Bing Copilot.[32] NLP has been further used to summarize texts, deduce author sentiment, solve symbolic math problems, and even generate programming code.[33−36] The effectiveness of NLP is predicated on languages having a structured symbolic syntax and set of rules to assemble basic units known as "tokens" (e.g., characters, words, or punctuation) to form higher-order constructs such as sentences or paragraphs.[37] The structured outputs of such a system reflect the grammar, conventions, and styles of the associated language. In NLP, tokens are transformed to encode "meanings" through mathematical vectors such that tokens of similar meaning are positioned closer together in the representational vector space.[38] By analyzing a large collection of data, NLP methods aim to infer emergent relationships between tokens that define the "rules" of a language. Once learned, this inferred set of rules can be used to perform predictive tasks such as separating tokens into categories, translating text from one language to another, and even predicting whether a literary work will be a commercial success.[39−41]

NLP can provide a complementary perspective to a simply biochemical view of biomolecules by treating protein and compound sequences as "languages" composed of amino acid and chemical tokens. Through the use NLP-inspired models, researchers can capture subtle sequence patterns, secondary structural motifs, and functional domains that correlate with ligand-binding specificity and affinity.[26,27] Integrating NLP approaches with ML for PLI prediction has shown early promise, as models pretrained on large protein or compound databases learn contextual embeddings that can enhance pattern recognition for predictions of ligand binding affinity and specificity. By analyzing common surrounding tokens of given amino acids or atoms, biological roles may be inferred, such as whether an amino acid plays an important role in secondary structure.[42,43] Early comparisons with traditional methods have shown encouraging performance improvements, highlighting the potential of NLP-based models to refine both the accuracy and interoperability of predictions and ultimately, help expedite drug discovery. For practical uses, NLP methods have been used for a variety of predictive tasks, including to inferr disease-gene associations,[44] predict tumor gene expression patterns,[45] and assign functional annotations to various protein-coding genes.[21] Despite impressive advances, the creation of these NLP models is associated with a sizable computational burden[46−51] and it remains a challenge to understand what and which specific features of the input sequence data are responsible for predictive success.

In this review, we explain how NLP offers new ways to understand and predict PLIs. We first describe the relationship between common protein and ligand text representations vis-á-vis the characteristics of human language. Next, we present a paradigm of data collection for PLI studies and employ a table of data sources organized loosely by tasks for which they may be best suited. We then introduce and discuss three major NLP-associated methods often employed in machine-learning-based PLI studies: the Recurrent Neural Network (including variants like Long Short-Term Memory (LSTM)), the Transformer, and Attention Mechanisms. We provide several tables to convey how published studies have employed these major architectures to predict PLIs. What is in common between these major methods is their efficacy in capturing long-distance relationships between atoms and/or amino acids that are crucial for binding; we contextualize their use by presenting a conceptual framework for predicting PLIs that is followed by many NLP-PLI studies.

We conclude with a discussion of the limitations of using NLP in studying PLIs and with the data currently available for training machine learning models. Current approaches have shown promising results, but there are still significant challenges related to data variety, model interpretability, and bias. NLP offers valuable strategies for exploratory analysis and has taken a place in the foundation of such efforts but is not a standalone solution; integrating insights from other disciplines, such as computer vision, and domain-specific knowledge may be crucial for advancing PLI research in the future. We emphasize the need for high-quality, well-balanced data sets and suggest that new strategies, such as high-throughput simulations, could provide a pathway to overcoming current data limitations. Moreover, we emphasize the importance of integrating domain expertise, such as structure-based insights.

https://doi.org/10.1021/acs.jcim.4c01907
J. Chem. Inf. Model. 2025, 65, 2191−2213

2. THE LANGUAGES OF LIFE

Human languages are ever-evolving,[52] often ambiguous,[53] and idiosyncratic, which make them not ideal for computational study given the importance of context.[54] Human languages are generally hierarchical, composed of layers on the order of words, phrases, sentences, etc. by which information is communicated.[55] Human languages also demonstrate complex local behaviors that diverge from a hierarchal perspective, including long-distance dependencies[56] (e.g., subject and pronoun), as well as common substructure constructs like idioms ("raining cats and dogs") or groups of objects that function as a unit ("knife and fork").[57] Recursion is another linguistic aspect of human language that goes beyond simple hierarchy, for example, "I *believe* that you *suppose* that...".[58] In general, these meta-linguistic occurrences are consistent with a view of language in which the linear order of words gives rise to a construct that embodies information.[57] While biochemical texts are distinct from human languages, there is remarkable similarity between the two regarding the hierarchal-and-sequential nature of construction as well as how local and global information is encoded (Figure 1). Nevertheless, the hierarchies of construction are not directly analogous as both protein sequences and molecular texts have significant structural and ontological distinctions that should be accounted for during computational processing. A comparison between human languages and the most common forms of text-based representations of proteins and molecules is presented below.

2.1. The "Language" of Proteins. Protein sequences are akin to human language in that they possess a hierarchical order of construction and embody embedded information. Human language text is inherently ordered with characters of an alphabet assembled linearly and grouped into words, phrases, and sentences that convey an emergent message. Protein sequences similarly obey a hierarchy of assembly, with amino acids (AAs) serving as the alphabet. When AAs are strung together, secondary structural motifs, domains, and quaternary (multidomain-interacting) structures may emerge with properties that contribute to function.[59,60] While external factors such as post-translational modifications and cellular state can play a substantial role in dictating protein three-dimensional structure and function, the AA sequence represents the essential blueprint that ontologically defines the properties of a protein.[61−63] This fact has served as the foundation for bioinformatic analysis of proteins.[64] Individual AAs and common subsequences contribute to the "information" of the overall protein just as words contribute to the meaning of a text.

However, protein sequences are not entirely analogous in their hierarchy as compared to human languages, and "words" are not easily identified or demarcated. In linguistics, a "word" is a complete unit of meaning that a reader can recognize. It would be dubious to assume that AAs are equivalent to "words" because the roles of individual AAs are highly dependent on their context and environment. The meaning of a word may be independent of its surroundings; however, an amino acid carries "meaning" highly dependent on its three-dimensional context. Protein motifs or domains are also not comparable to "words", since not all regions of a protein are independent of one another[65] and motifs and domains are not completely independent units. This lack of word-equivalence for protein sequences has driven "sub-word" identification methods that identify strings that act similarly to words.[66] Protein sequences also differ from human languages in the length scale of interactions and the number of long-distance interactions that contribute to a 3D structure. While human language often features distant dependencies, such as between subject and pronoun or text that foreshadows later content, these relations can be easily deduced by a reader and remain relatively sparse on a per-sentence basis. In contrast, AAs may have numerous distant relationships that are difficult to predict[67−69] without the assistance of computational or experimental tools. These characteristics allow a protein sequence to encode multiple layers of complex information, including 3D structure, structural dynamics, and/or binding interactions.[59,60] In essence, a sequence is not just a static representation, but rather a sophisticated programmatic embodiment that determines both structure and behavior of a protein.

2.2. The "Language" of Ligands. The chemical structures of molecules can be similarly translated into text-based notations and analyzed computationally.[70] However, unlike the elements of human text and protein sequences, the chemical connectivity patterns of molecules are not one-dimensional. Nevertheless, text-based schema has been developed to represent chemical information in a manner convenient for computational analysis,[71] with the Simplified Molecular-Input Line-Entry System (SMILES) format being one of the most widely used.[72]

SMILES strings are text representations constructed over a depth-first traversal of a two-dimensional molecular graph (Figure 1), with atoms, atomic properties, bonds, and structural properties represented by characters following an established set of conversion rules. Given the memory-efficient and somewhat human-readable format of SMILES, it has become a standard in chemical databases and computational tools,[73−75] and the most commonly used text representation in PLI studies. Although SMILES lacks an intuitive way to determine a chemical equivalent of a "word", there is a well-defined grammar to denote properties and substructures of a molecule. Moreover, the same molecule can be represented by multiple different SMILES strings,[72] which is similar to how there could be multiple sentence constructions to convey the same idea in human languages. In NLP applications, incorporating tokens with the same meaning into the training process can yield a robust predictive model.[76] The use of multiple SMILES per molecule has been leveraged to guide ML models to discern which parts of a ligand contribute to drug potency.[77]

The SMILES format is dissimilar from human languages in a similar way as for protein sequences. First, the lengths of SMILES strings could vary far more than in human languages, ranging from listing each atom of a small molecule to those constituting entire proteins. The SMILES format is less practical to use for larger molecules, however, since structural graphs can provide a more compact and accurate representation of atoms in a large three-dimensional structure. A disadvantage of using SMILES in general is that it is difficult to intuitively discern "word" equivalents within the string. Individual branches separated by parentheses could be viewed as words,[78] but this is only practical for small branching groups. Moreover, the handling of nesting parentheses in SMILES for large molecules can be problematic and has become a major limiting factor in ML models designed to generate novel molecules.[79] The sum of these SMILES

https://doi.org/10.1021/acs.jcim.4c01907
J. Chem. Inf. Model. 2025, 65, 2191−2213

Table 1. Data Sets and Databases for PLI Prediction[a]

Data set Name	Year	Proteins	Ligands	Interactions	Protein Category	Ligand Category	Task
Functional Data Available							
Protein Data Bank (PDB)[92]	2000	220,777	–	–	General (Structure)	General (Structure)	C
BRENDA[103]	2002	8,423	38,623	–	Enzymes	General	R, C
PDBBind[b,96]	2004	–	–	23,496	General (Structure)	General	R, C
DrugBank[b,91]	2006	4,944	16,568	19,441	Human Proteome	General	C
BindingDB[104]	2007	2,294	505,009	1,059,214	General	General	R, C
PubChem[73,92]	2009	248,623	119,108,078	250,633	General	General	R, C
Davis[94]	2011	442	68	30,056	Kinases (Sequence)	Kinase Inhibitors (SMILES)	R
PSCDB[105]	2011	–	–	839	Human Proteome	General	R, C
ChEMBL[90]	2012	15,398	2,399,743	20,334,684	General (Protein ID)	General (SMILES)	R, C
DUD-E[106]	2012	102	22,886	2,334,372	General	General	R, C
Iridium Database[b,107]	2012	–	–	233	General	General	R, C
KIBA[95]	2014	467	52,498	246,088	Kinases (Protein ID)	Kinase Inhibitors (SMILES)	R
Natural Ligand Database (NLDB)[b,108]	2016	3,248	–	189,642	Enzymes (Structure)	General	R, C
PDID[109]	2016	3,746	51	1,088,789	Human Proteome	General	R, C
dbHDPLS[b,110]	2019	–	–	8,833	General (Structure)	General	C
CovPDB[b,111]	2022	733	1,501	2,294	General (Structure)	General	C
PSnpBind[b,112]	2022	731	32,261	640,074	General	General	R, C
Protein Binding Atlas[b,112] Portal	2023	1,716	30,360	129,333	Drug Targets	Drug Molecules	R, C
Protein–Ligand Binding Database (PLDB)[b,113]	2023	12	556	1,831	Carbonic Anhydrases, Heat Shock Proteins	General	R
BioLiP2[114]	2023	426,209	–	823,510	General (Structure)	General	R, C
PLAS-20k[b,115]	2024	–	–	20,000	Enzymes	General	R, C
Functional Data Unavailable							
Database of Interacting Proteins[116]	2004	28,850	–	81,923	Various Species	–	C
Protein Small-Molecule Classification Database[b,117]	2009	4,916	8,690	–	General (Structure)	General (Structure)	C
CavitySpace[b,118]	2022	23,391	–	23,391	General (Structure)	General	C

[a]*Note*: Data sets categorized as "General" provide broad information without focusing on specific categories of proteins or ligands. Data types (e.g., sequence, structure), are denoted in parentheses. Categories labeled with "Protein ID" include protein IDs from established databases. Data sets may receive periodic updates. Suggested tasks are denoted as "R" for regression and "C" for classification. "–" indicated that exact information is either not included in the source or is not readily obtainable. [b]Protein–ligand complexes are available with the data set.

shortcomings has led to the development of alternative chemical representations for computational studies such as DeepSMILES and SELFIES.[80,81] Although promising, these alternative forms have rarely been used in ML-based PLI studies to date. The question remains whether a three-dimensional molecule can be truly mapped to a text representation in a way that preserves all relevant structural information for use in predicting PLIs.

3. PROTEIN–LIGAND INTERACTION DATA AND DATA SETS

Protein–ligand binding is a complex process dictated by many factors including protein states, hydrophobicity/hydrophilicity, and conformational flexibility.[82] The question of *how* to represent a protein and ligand in a computational space is critical and multifaceted. A wealth of information has been collected experimentally and generated through simulation studies on the properties of proteins and ligands, but these data are highly variable with regard to type, quality, and quantity. This section catalogs several primary data representations used in PLI studies. We also discuss the availability, selection, and curation of available data for machine-learning-based training and evaluation.

Protein and ligand representations are typically sequence- or structure-based. Unlike sequence-based text formats, structure-based information can appear in multiple forms, e.g., atomic coordinates of protein–ligand complexes or contact maps. Some structural information can be artificially reconstructed from sequence-based formats through algorithms such as AlphaFold for proteins[83] and RDKit for ligands.[84] PLI studies using machine-learning methods will typically select either sequence-based or structure-based inputs, although there is a growing use of mixed input data types.[85,86] For example, a mixed-data study may represent proteins by AA sequences but ligands by atomic coordinates, a choice based in part on the fact that highly accurate 3D chemical structures are easier to obtain than those of proteins and that full-atom representations of ligands are not memory intensive.

Other data can also be incorporated to augment ground-truth information about PLIs. For example, molecular weights, polarity, and bioactive properties can be incorporated into models to further improve the prediction of PLIs.[87,88] Studies have included molecular weights, ligand polar surface area, and protein aromaticity,[87] or bioactive properties of chemical and clinical relevance[88] have resulted in improved predictions of binding affinity. Leveraging multiple-sequence alignment or phylogenetic information to identify coevolutionary trends among AAs and sites of covalent modification has been shown to dramatically improve the accuracy of structural predictions of protein–ligand complexes.[89] The use of non-sequence/non-structural data can enable models to yield better predictive

https://doi.org/10.1021/acs.jcim.4c01907
J. Chem. Inf. Model. 2025, 65, 2191–2213

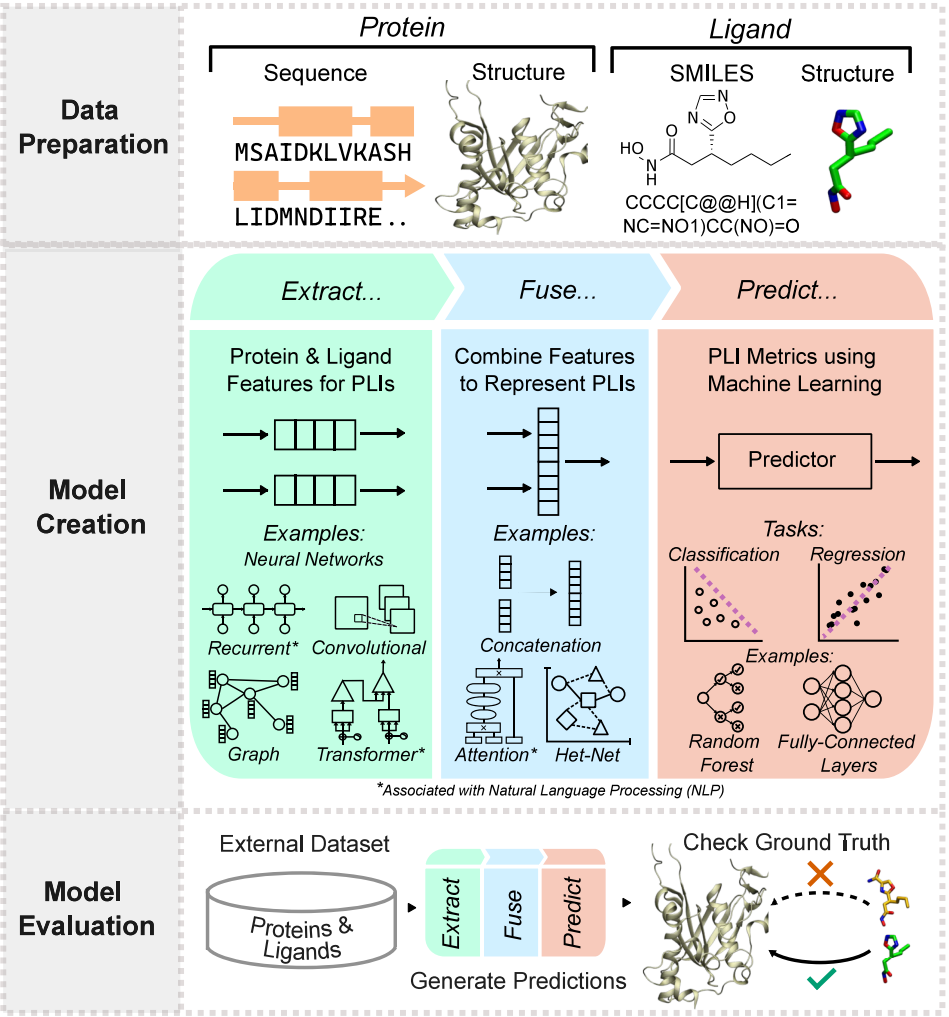


Figure 2. Summary of the data preparation, model creation, and model evaluation workflow. Model Creation for PLI studies follows an Extract-Fuse-Predict Framework: input protein and ligand data are extracted and embedded, combined, and passed into a machine learning model to generate predictions.

performance for characterizing protein and ligand and their interactions.[87]

Data for the study of PLIs can be manually curated by domain experts or sourced from existing data sets. Widely used public databases such as ChEMBL,[90] PubChem,[73] and DrugBank[91] play a critical role in the development and evaluation of drug−protein interaction models. These databases contain a wealth of information on ligands, proteins, and their interactions, supporting various predictive tasks. For instance, PubChem contains over 119 million ligands and is a cornerstone resource for general-purpose regression and classification models. Similarly, DrugBank focuses on the human proteome and offers curated data tailored to drug discovery, while ChEMBL provides comprehensive data on protein−ligand interactions, including SMILES-based ligand information.

Many databases are also inherently interconnected. For example, data sets involving structural information often reference available structures in the Protein Data Bank.[92] Similarly, sequence-based data sets frequently link back to UniProt[93] for protein sequence data. This interconnectedness emphasizes the importance of selecting a data set with the intended predictive task in mind. For tasks requiring high-quality, targeted data—such as predicting kinase activity—specialized data sets like Davis[94] or KIBA[95] are preferable. These data sets offer focused, curated information that aligns with specific biological questions. Conversely, general data sources like ChEMBL or PubChem are more suitable for deriving models aimed at uncovering generalizable underlying rules.

Given a protein−ligand representation, several predictive tasks are possible. *Classification* studies seek to categorize PLIs into distinct groups, for example, whether a protein−ligand pair binds or not. These models are relatively simple and allow for input from various sources. *Regression* studies use a continuous functional metric to characterize PLIs such as a binding affinity/dissociation constant (K_d) or inhibition constant ($IC50$). Continuous target variables allow for the involvement of numerical values derived directly from 'ground-truth' experimental data in both training and evaluation. Databases like PDBBind[96] contain functional metrics such as K_d and $IC50$ but not all protein and ligand pairings cataloged have such metrics available, for example, complexes identified from X-ray crystallography, Cryo-EM, or NMR screening studies.[13,97] Since regression studies require quantitative PLI data and not merely whether a protein and ligand interact,

https://doi.org/10.1021/acs.jcim.4c01907
J. Chem. Inf. Model. 2025, 65, 2191−2213

Table 2. Sequence-Based PLI Prediction Models[a]

Model Name	Extraction		Fusion	Prediction
	Protein Extractor	Ligand Extractor		
LSTM				
Affinity2Vec[140]	ProtVec	Seq2Seq	Heterogeneous Network	Gradient-Boosting Trees (R)
DeepLPI[141]	ResNet	ResNet	Concatenation with LSTM	FCN (C, R)
FusionDTA[142]	BiLSTM	BiLSTM	Concatenation with Linear Attention	FCN (R)
Transformer				
Shin et al.[181]	CNN	Transformer	Concatenation	FCN (R)
MolTrans[182]	Transformer	Transformer	Interaction Matrix[b] with CNN	FCN (C)
ELECTRA-DTA[180]	CNN with Squeeze-and-Excite Mechanism	CNN with Squeeze-and-Excite Mechanism	Concatenation	FCN (R)
MGPLI[184]	Transformer, CNN	Transformer, CNN	Concatenation	FCN (C)
SVSBI[183]	Transformer, LSTM, and AutoEncoder	Transformer, LSTM, and AutoEncoder	k-embedding fusion[c]	FCN, Gradient-Boosting Trees[d] (R)
Non-Transformer Attention				
DeepCDA[121]	CNN with LSTM	CNN with LSTM	Two-Sided Attention[d]	FCN (R)
HyperAttention-DTI[151]	CNN	CNN	Cross-Attention, Concatenation	FCN (C)
ICAN[150]	Various	Various	Cross-Attention, Concatenation	1D CNN (C)
Other NLP Methods				
GANsDTA[202]	GAN Discriminator	GAN Discriminator	Concatenation	1D CNN (R)
Multi-PLI[203]	CNN	CNN	Concatenation	FCN (C, R)
ChemBoost[124]	Various	SMILESVec	Concatenation	Gradient-Boosting Trees (R)

[a]*Note:* A model's task of Classification (C) and/or Regression (R) is denoted beside the "Prediction" column entries in parentheses. Definitions for specific terms may be found in the Glossary (Table 6). *Terms Defined by the Cited Authors*: [b]**Interaction Matrix**: Output from dot product operations to measure interactions between protein subsequence and ligand substructure pairs. [c]**k-embedding fusion**: The use of machine learning to find an optimal combination of lower-order embeddings via different integrating operations. [d]**Two-sided Attention**: Attention mechanism that computes scores using the products of both pairs of protein/ligand fragments and protein/ligand feature vectors.

relevant data set sizes may be smaller than those for classification. However, gathering such data is a laborious process in terms of both time and laboratory resources. Additionally, while functional metrics associated with regression studies can be used to predict exact values, the same data can support classification tasks, such as predicting binding versus non-binding rather than a specific binding affinity value.

Table 1 provides a comprehensive overview of existing PLI data sets and databases, summarizing their characteristics and suitability for various predictive tasks. Preassembled data sets are appealing for their convenience, though aligning the data set's scope and quality with one's modeling goals and the nature of the scientific inquiry is essential. Such a task-driven approach ensures robust model performance and meaningful predictions.

A secondary but still crucial consideration is the splitting of the data into training, validation, and test sets for use in a model. The training set constitutes the majority of the data from which a model's parameters are learned;the validation set is used to tune the model's configuration (controled by "hyperparameters");[98] and the test set is a separate set of data points used to determine model performance.[98] There are several ways to create data splits aside from the simple option of randomly dividing the data. For example, a model may be designed with data splits that ensure different proteins or ligands are included in the training/validation/test sets such that proteins and ligands are not shared between them.[99] Evaluating a PLI prediction model on thes sets would then provide data on a model's performance on unknown proteins and ligands that are outside of its training set. Competitions often provide a specific, well-designed test set data split as a benchmark, an approach used for other predictive challenges such as the Critical Assessment of Structural Prediction (CASP)[100] and the Critical Assessment of Prediction of Interactions (CAPRI).[101,102]

4. MACHINE LEARNING AND NLP FOR PLIS

Machine learning is a field of study where algorithms are used to uncover hidden patterns from data sets without explicit rule-based programming. Desired outcomes of specific processes are referred to as tasks (e.g., classification, regression, etc.), and depending upon the tasks, a suitable machine learning *model* is chosen, which includes decision trees, support vector machines, neural networks (NNs), and deep learning architectures.[98] NLP tasks often rely on deep learning and neural network architectures, which can both process the immense amounts of language-related data available and model the complex and often conflicting rules of human languages..[119] Due to the parallels between the representation of language constructs and those of proteins and ligands, NLP-oriented machine learning approaches will be the focal point of this review article.

The general workflow for any ML-based study can be broadly characterized into three stages: data preparation, model creation, and model evaluation (Figure 2). For PLI studies, data preparation typically entails selecting the types and formats of protein and ligand data (e.g., sequence and/or structural). ML model creation may involve the following three tasks, although the boundary between these tasks could be fuzzy at times: (i) *Extract*: the "extraction" of vector "embeddings" from the protein and ligand input data, which

https://doi.org/10.1021/acs.jcim.4c01907
J. Chem. Inf. Model. 2025, 65, 2191−2213

Table 3. Structure-Based PLI Prediction Models[a]

Model Name	Extraction		Fusion	Prediction
	Protein Extractor	Ligand Extractor		
Transformer				
UniMol[122]	Transformer-Based Encoder	Transformer-Based Encoder	Concatenation	Transformer-Based Decoder (R)
Other Attention				
Lim et al.[160]	GNN	GNN	Attention	FCN (C)
Jiang et al.[152]	GCN	GCN	Concatenation	FCN (R)
GEFA[153]	GCN	GCN	Concatenation	FCN (R)
Knutson et al.[155]	GAT	GAT	Concatenation	FCN (C, R)
AttentionSite-DTI[158]	GCN with Attention	GCN with Attention	Concatenation, Self-Attention	FCN (C, R)
HAC-Net[156]	GCN with Attention Aggregation	GCN with Attention	Combined Graph Representation	FCN (R)
BindingSite-AugmentedDTI[157]	GCN with Attention	GCN with Attention	Concatenation, Self-Attention	Various (R)
PBCNet[154]	GCN	Message-Passing NN	Attention	FCN (R)

[a]*Note*: A model's task of Classification (C) and/or Regression (R) is denoted beside the "Prediction" column entries in parentheses. Definitions for specific terms may be found in the Glossary (Table 6).

Table 4. Mixed Representation PLI Prediction Models[a]

Model Name	Input Type	Extraction		Fusion	Prediction
		Protein	Ligand		
LSTM					
Zheng et al.[204]	P: Struct. L: Seq	Dynamic CNN[b] with Attention	BiLSTM with Attention	Concatenation	FCN (C)
DeepGLSTM[85]	P: Seq L: Struct.	BiLSTM with FCN	GCN	Concatenation	FCN (R)
Transformer					
Transformer-CPI[86]	P: Seq L: Struct.	Transformer Encoder	GCN	Transformer Decoder	FCN (C)
DeepPurpose[201]	P: Seq L: Either	4 Various Encoders	5 Various Encoders	Concatenation	FCN (C, R)
CAT-CPI[185]	P: Seq L: Image	Transformer Encoder	Transformer Encoder	Concatenation	CNN and FCN (C)
Non-Transformer Attention					
Tsubaki et al.[205]	P: Seq L: Struct.	CNN	GNN	Attention and Concatenation	FCN (C)
DeepAffinity[206]	P: Seq L: Struct.	RNN-CNN with Attention	RNN-CNN with Attention	Concatenation	FCN (R)
MONN[207]	P: Seq L: Struct.	CNN	GCN	Pairwise Interaction Matrix,[c] Attention	Linear Regression (C, R)
GraphDTA[197]	P: Seq L: Struct.	CNN	4 GNN Variants	Concatenation	FCN (R)
CPGL[208]	P: Seq L: Struct.	LSTM	GAT with Attention	Two-Sided Attention,[d] Concatenation	Logistic Regression (C)
CAPLA[161]	P: Both L: Struct.	Dilated Convolutional Block	Dilated Convolutional Block with Cross-Attention to Binding Pocket	Cross-Attention, Concatenation	FCN (R)

[a]*Note*: A model's task of Classification (C) and/or Regression (R) is denoted beside the "Prediction" column entries in parentheses. Definitions for specific terms may be found in the Glossary (Table 6). The input representations for sequence and structure are abbreviated for brevity. *Terms Defined by the Cited Authors*: [b]**Dynamic CNN**: ResNet-based CNN modified to handle inputs of variable lengths by padding the sides of the input with zeroes. [c]**Pairwise Interaction Matrix**: A [number of atoms]-by-[number of residues] matrix in which each element is a binary value indicating if the corresponding atom-residue pair has an interaction.[207] [d]**Two-sided Attention**: Attention mechanism that uses dot product operations between protein AA and ligand atom pairs, while taking matrices of learned weights into account.

can be used in computational operations (described in Section 4.2), (ii) *Fuse*: the fusion of protein and ligand vector embeddings, and (iii) *Predict*: the prediction of a PLI target property as a model's output. The predictive capability of the model would be ideally validated against results from other studies and/or real-world measurements in a model evaluation stage. While data preparation and extraction steps have typically been the focus of most research efforts, every component of the workflow is crucial to successful PLI prediction.

4.1. The Extract-Fuse-Predict Framework. A variety of models for PLI prediction have been constructed in recent years, and these models tend to fall into four general categories: (1) *sequence-based*, where protein sequences and SMILES are used to represent protein and ligand, respectively; (2) *structure-based*, where structural information is included in the representation of both protein and ligand; (3) *mixed representations*, where both structural and sequence information are involved; and (4) *sequence-structure-plus*, which substantially incorporates other ground-truth information

https://doi.org/10.1021/acs.jcim.4c01907
J. Chem. Inf. Model. 2025, 65, 2191−2213

Table 5. Sequence-Structure-Plus PLI Prediction Models[a]

	Extraction				
Model Name	Protein Extractor	Ligand Extractor	Additional Features Used	Fusion	Prediction
LSTM					
HGDTI[209]	BiLSTM	BiLSTM	Disease and Side Effect Information	Concatenation	FCN (C)
ResBiGAAT[87]	Bidirectional GRU with Attention	Bidirectional GRU with Attention	Global Protein Features	Concatenation	FCN (R)
Transformer					
Gaspar et al.[125]	Transformer or LSTM	ECFC4 Fingerprints	Multiple Sequence Alignment Information	Concatenation	Random Forest (C)
HoTS[210]	CNN	FCN	Binding Region	Transformer Block	FCN (C, R)
PLA-MoRe[88]	Transformer	GIN and AutoEncoder	Bioactive Properties	Concatenation	FCN (R)
AlphaFold 3[89]	Attention-Based Encoder[b]	Attention-Based Encoder[b]	Post-Translational Modifications, Multiple Sequence Alignment Information	Attention	Diffusion Transformer[c]
Other NLP Methods					
MultiDTI[123]	CNN with FCN	CNN with FCN	Disease and Side Effect Information	Heterogeneous Network	FCN (C)

[a]*Note*: A model's task of Classification (C) and/or Regression (R) is denoted beside the "Prediction" column entries in parentheses. Definitions for specific terms may be found in the Glossary (Table 6). *Terms Defined by the Cited Authors*: [b]**Atom Attention Encoder**: An attention-based encoder that uses cross-attention to capture local atom features. [c]**Diffusion Transformer**: A transformer-based model that aims to remove noise from predicted atomic coordinates until a suitable final structure is output.

Table 6. Glossary of Terms That Appear in the Tables

Term	Definition
AutoEncoder	A neural network tasked with compressing and reconstructing input data, often used for feature learning.[262]
BiLSTM	Bidirectional Long Short-Term Memory, a variant of LSTM where two passes are made over the input sequence, one reading in forward order, and one in reverse order.
CNN	Convolutional Neural Network, a type of neural network that processes grid-like data, such as images, through a gradually-optimized filter that slides across input data to discern important features.
Dilated Convolutional Block	Convolutional Neural Network operations with defined gaps between kernels, which can capture larger receptive fields with fewer parameters.
ECFC4 Fingerprint	A molecular fingerprint that encodes information about the presence of specific substructures within a diameter of 4 bonds from each atom.[263]
FCN	Fully-Connected Network, a feedforward Neural Network where each neuron in one layer connects to every layer in the next. FCNs can also be referred to as Multi-Layer Perceptrons.
GAN Discriminator	An NN part of Generative Adversarial Networks (GAN) that learns important features to distinguish between real and artificial data.
GAT	Graph Attention Network, a type of Graph Neural Network that uses attention mechanisms to deciding the value of neighboring nodes to a given node when updating a node's information.[264]
GCN	Graph Convolutional Network, a type of Graph Neural Network that aggregates neighboring node features through a first-order approximation on a local filter of the graph.[265]
GIN	Graph Isomorphism Network, a type of Graph Neural Network that uses a series of functions to ensure embeddings are the same no matter what order nodes are presented in.[266]
Gradient-Boosting Trees	A machine learning technique where many decision trees are trained in order, such that the next tree learns from the misclassified samples of the previous tree. All trees are then used to "vote" on results of each input.
GRU	Gated Recurrent Unit, a simplified version of Long Short-Term Memory that similarly uses a gating mechanism to retain and forget information, but is less complex than Long Short-Term Memory.[137]
Heterogeneous Network	A graph where nodes and edges represent different types of information, often used to convey complex relationships in biological systems (e.g., drug, target, side-effect, etc.).
Message-Passing NN	Type of Graph Neural Network that computes individual messages to be passed between nodes so that representations for each node contain information from its neighbors.[267]
ProtVec	A method for representing protein sequences as dense vectors using skip-gram neural networks.[268]
Random Forest	A machine learning method where many decision trees are constructed, and the result of the ensemble is the mode of the individual tree predictions.
ResNet	Short for Residual Network. A neural network architecture that speeds up training by learning functions to substitute for layer operations, allowing for the "skipping" of layers and faster training.[269]
Seq2Seq	A machine learning method used for language translation in NLP, featuring an encoder-decoder structure.[266]
SMILESVec	Previous work from authors. 8-character ligand SMILES fragments are assigned a vector through a single-layer neural network, and an input SMILES string's vector is equal to the mean of fragment vectors present in that input SMILES.[270]
Squeeze-And-Excite Mechanism	Mechanism for Convolutional Neural Networks that uses global information to adapt the model to emphasize more important features.[271]

beyond sequence and structural data (such as molecular weights or polar surface area[87]). Tables 2, 3, 4, and 5 summarize several representative NLP-based PLI prediction studies across these categories over the past five years. Although PLI studies could be categorized in other ways—for example by the ML model used (neural network, decision tree, etc.) or by the predictive task type (classification vs. regression)—we have chosen to emphasize a categorization

https://doi.org/10.1021/acs.jcim.4c01907
J. Chem. Inf. Model. 2025, 65, 2191−2213

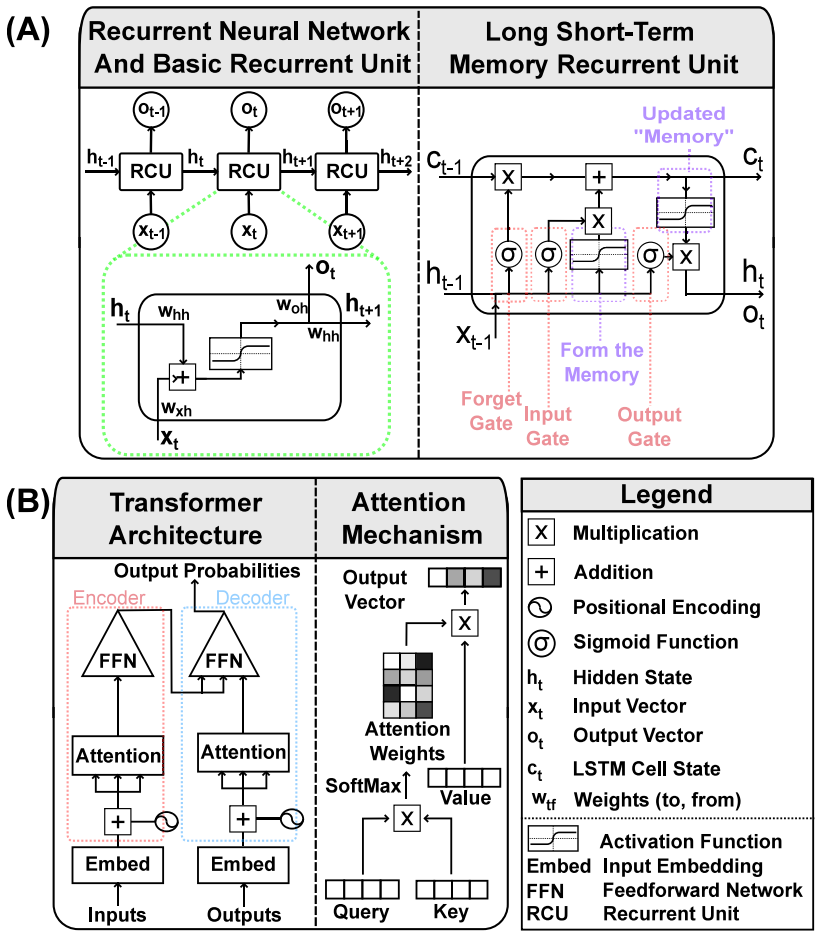


Figure 3. Framework diagrams for RNN (and its variant LSTM), transformer, and attention with arrows representing a flow of information. (**A**) The "unrolled" structure of an RNN and the recurrent units, where hidden states propagate across time steps. The recurrent unit takes the current token X_t as input, combines it with the value of the current hidden state h_t, and computes their weighted sum before generating the response O_t and an updated hidden state h_{t+1}. Weighted sums depend upon the associated network weights W_{xh}, W_{hh}, or W_{oh}, which connect input to hidden state, hidden state to hidden state, and hidden state to output, respectively. LSTM differs in that a memory state is updated during each iteration, facilitating long-term dependency learning. (**B**) A simplified framework of a transformer's encoder-decoder architecture, and associated attention mechanism. A scaled product of the Query and Key vectors yields attention weights that can provide interpretability, with the new embedding vector (or the output vector) updated based on this specific key.

based on input data type since the computational methods used for sequence text and structural data comprise a major difference.

4.2. Extraction of Embeddings. NLP approaches deconstruct text into individual tokens or "units of meaning" for use in computational operations and inferences via a process referred to as "tokenization".[37] Schema for tokenization, aside from character-based and word-based, can also be subword-based. Subword-based tokenization breaks down text into units smaller than words to create a wider vocabulary; it is commonly selected when the definition of a "word" is unclear, as subwords can be used as a means to discover "words".[66,120] Common ways to assemble subwords include methods such as "n-grams", where each subword has a select fixed-length value n (e.g., "Sma", "mar", "art", etc. for n = 3 and the word "Smart"). While subword tokenization has been attempted in PLI studies for both protein (e.g., amino acid k-mers such as "KHR", "LKL", "KGY") and ligand (e.g., "CCCC","[C@@H]"),[121−125] the current trend is to use amino acids and/or individual atoms directly as tokens.

To be processed computationally, tokens must be translated into a numerical form through a process known as "embedding". There are many types of token embedding, but they are generally designed to capture either a particular token meaning, frequency, or both[126,127] and represented by a multidimensional vector. The direction of a token's vector embedding effectively represents its "meaning" and its magnitude represents the strength by which that meaning is conveyed. In isolation, each token could possess multiple meanings (e.g., the word "run" has multiple meanings[128]), and so context may be necessary to impart an intended meaning. NLP methods have been demonstrated to be highly effective at extracting patterns that convey context-dependent meanings from a large corpus of text. Embeddings that capture semantic meaning and relationships can then be used for many other tasks aside from predicting whether a protein interacts with a ligand, such as predicting protein and ligand solubilities.[129,130]

Token embedding is typically accomplished using a neural network (NN) architecture that approximates nonlinear relationships between the "inputs" of the network (the data) and its "outputs" (the predictions).[131] Neurons in an artificial NN receive, integrate, and transmit signals to other neurons through a nonlinear response function and are arranged in layers. Information is passed from an input layer through one

https://doi.org/10.1021/acs.jcim.4c01907
J. Chem. Inf. Model. 2025, 65, 2191−2213

or more intermediate "hidden" layers to an output layer.[98] Interconnection weights that govern the strength of influence of one neuron on another are crucial parameters of an NN. A wide variety of NNs have been applied to studying PLIs although not all are commonly used in NLP. Nevertheless, two types of NNs commonly associated with NLP are Recurrent Neural Networks (RNNs)[132,133] and attention-based NN models.[134] Below, we highlight the details necessary to understand how RNNs, attention, and other non-NLP-driven NNs have been used to glean global patterns essential for PLI predictive tasks. For reference, Figure 3 presents simplified framework diagrams of RNN, transformer, and attention operations.

4.2.1. Recurrent Neural Networks. RNNs[132] are specialized in processing sequential data in which the order of the data is significant. Consider an input data sequence x_1, x_2, ..., x_{t-1}, x_t, x_{t+1}, ... in which individual tokens x_t are ordered by a time-step t, and the input sequence embodies a particular yet unknown pattern over the length of the sequence. In traditional NNs, information flows from the input layer to the output in a single pass, making it difficult to decipher any interdependencies between earlier and subsequent tokens. To remedy this, the RNN architecture introduces recurrent units through which the processing of the input sequence at the current time-step will also update "hidden states" that serve as memory, nonlinearly capturing the information of all input tokens up to the current time-step. The recurrent unit derives its name from the fact that the hidden state participates in the computation both as an input and as an output for each input in the sequence.

In other words, given the network weights, the ordered sequence of input tokens will determine a network output sequence O_1, O_2, ..., O_{t-1}, O_t, O_{t+1}, ..., and the hidden states h_1, h_2, ..., h_{t-1}, h_t, h_{t+1}... Thus, the hidden states are functionally equivalent to the hidden layers of traditional NNs but differ by updating *recurrently*, where information is carried over from previous time-steps to the current time-step. Consequently, the dependencies between tokens of the sequential inputs can be captured *implicitly* by the hidden state.

RNNs can be represented in an unfolded, or "unrolled" state (see Figure 3 A). In this representation, an input sequence can be considered as a mapping between preceding input values and values of subsequent elements in the same sequence, due to the inherent patterns existing within all elements.[135] For example, given a protein sequence for which each AA is a token, an RNN would process the sequence of AAs one at a time to create and maintain a mapping for the next AA in the sequence accounting for all input tokens seen so far. The mapping, encoded in the network weights of RNN, may be found via the backpropagation process, through which the shared weights are adjusted so that the "errors" between the computed outputs of the RNN and the expected outputs as presented in the input sequence are calculated and minimized.[136] The process of using backpropagation to adjust the network weights so that the desired outputs of an NN are achieved is the so-called *training* process in machine learning, with the resulting collection of weights being called a *model*.

A good example of an RNN applied to the study of PLIs is provided by Abdelkader et al.'s ResBiGAAT model,[87] which was designed to use a variant of bidirectional RNN layers to embed input strings (protein sequences or SMILES). ResBiGAAT's bidirectional RNN, which processed the input sequence of tokens both forwards and backwards in different passes, enables it to identify relations between a given token and both its previous and subsequent tokens. While effective in many NLP tasks, early RNNs commonly suffered diminishing returns with increasing text length. This was due to a simplistic network architecture in which there was systematic and non-discriminatory retention of information from *all tokens*, including outlier tokens that contribute little informationally to the underlying pattern. A variant of an RNN was chosen in ResBiGAAT that features a gating mechanism to specifically update and forget information from previous time steps;[137] the RNN used was also modified to include residual connections that enable information to be transmitted directly between layers without the need for calculating intermediate layers. This enabled several RNN layers to be stacked together with a relatively insignificant increase in convergence time. This use of RNN, alongside several other changes, allowed ResBiGAAT to outperform a selection of baselines at the time of publication in 2023.

To address the diminishing returns of early RNNs, gating mechanisms were developed to control the flow of information into the hidden state. The primary example of this is Long Short-Term Memory (LSTM) networks,[138] a popular variant of RNNs in which three gates are introduced into each recurrent unit: input gate, forget gate, and output gate (Figure 3A). The signature component of LSTMs is the forget gate, which selectively inhibits information not concordant with previously learned patterns found from processing prior tokens.[138] In addition, the input gate controls the level of input information added to the cell state, and the output gate governs the amount of information output at each step. Combined, the gating mechanisms selectively handle memory functionality, enabling effective encoding of long-term dependencies. For example, in the task of predicting protein secondary structures, LSTM has been shown to attenuate the contribution of AAs that do not correlate with any defined secondary structural element, yielding a small but definitively improved performance over the then state-of-the-art.[42,43] Unlike human languages where sentence structures possess distinct temporal orders, sequence-based representation of proteins and ligands may exhibit temporal or spatial symmetry, leading to researchers utilizing bidirectional LSTMs (BiLSTMs) to capture both preceding and subsequent tokens in a sequence string by applying an LSTM to text in both original and reverse order, and concatenating each of the resulting embeddings end-to-end.[139]

LSTMs and BiLSTMs are promising embedding approaches for predicting binding affinities of proteins and ligands.[140−142] However, their effectiveness is constrained by the computational inefficiency of the LSTM/BiLSTM architectures when processing large-scale data sets. Most successful applications of LSTM to date have been applied to only relatively small training data sets, on the order of a few thousand proteins and ligand pairs. This limitation mainly arises from the inherently non-parallel design where the tokens are being processed step-by-step, which makes training on large data sets slow and computationally expensive. Thus, NN architectures that leverage parallelization will be important to ensure reasonable training and prediction runtimes.

4.2.2. Attention-Based Architectures. Protein lengths can vary dramatically, from Insulin with 51-AAs to "giant proteins" that can exceed 85,000 AAs.[143] To use large amounts of sequence data to effectively process and predict PLIs for which long-distance interactions may be impactful, several alter-

https://doi.org/10.1021/acs.jcim.4c01907
J. Chem. Inf. Model. 2025, 65, 2191−2213

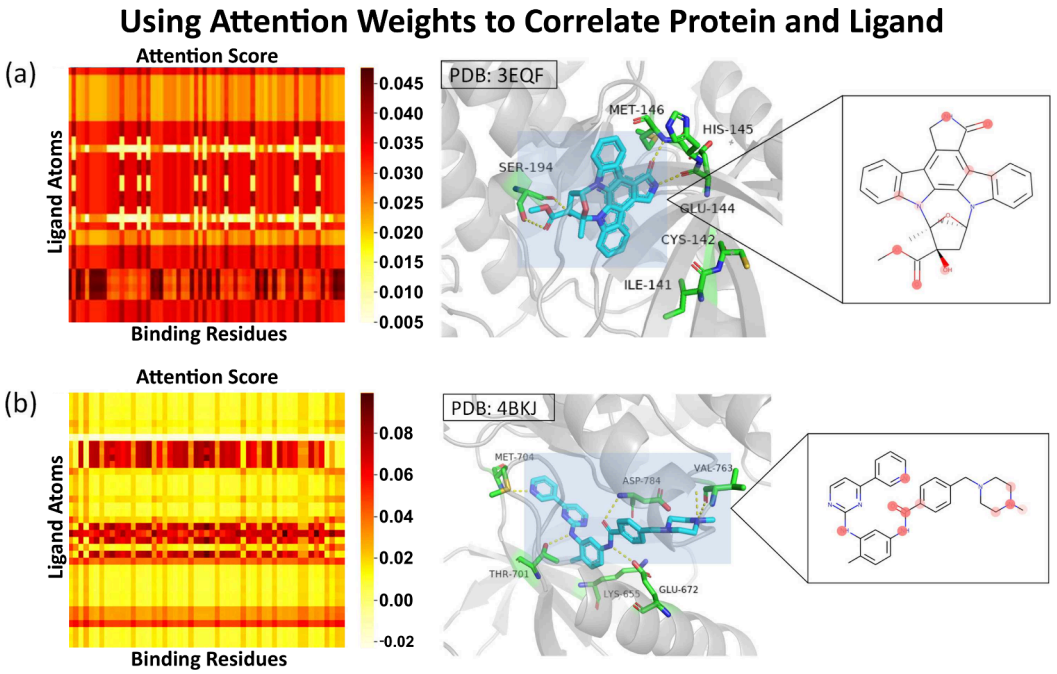


Figure 4. Sample attention weights for relating protein and ligand. The heatmaps on the left help visualize the weighted importance of select protein residues and ligand atoms in a PLI. Structural views of the protein−ligand binding pocket are shown in the middle, with insets of the 2D ligand structures on the right. The colored residues and red color highlights indicate AAs in the protein binding pocket and ligand atoms with high attention scores. Reproduced with permission from Figure 7 of Wu et al.[148] Used with permission under license CC BY 4.0. Copyright 2023 The Author(s). Published by Elsevier Ltd.

natives to RNN have been proposed. The "neural attention"—or simply "attention"—mechanism is an important recent breakthrough by which "attention weights" are dynamically calculated to quantify the relative contribution of different input tokens or elements to a predictive end goal.[134]

In the context of attention,[134] the input sequence of data is tokenized and represented (or embedded) as key-value pairs. A specific, previous section of the input (or a key) is said to be "attended to" when the model gives it a heavier weight in the process of updating the representation (i.e., the query) with each new input token. The attention weight is stored in a matrix, and is determined via a normalizing function and a similarity comparison between the key and the query, the latter of which may change dynamically as the representation of the input stream is updated. The attention mechanism is highly general, and can be applied to inputs such assequences and images, with the keys being potentially any embedding that is relevant to the current task.[144−146] In many NN architectures, attention can also incorporate hidden states into the calculation, allowing a more sophisticated mechanism for capturing longer-range correlations in deeper layers.[134,146]

Attention mechanisms have proven highly compatible with traditional protein sequence analysis approaches in identifying long-distance interactions between AAs of a protein.[147] In PLI studies, attention mechanisms can dynamically adjust the contribution of specific AAs or ligand atoms to a predictive outcome by amplifying interaction sites with higher attention scores and downplaying less relevant ones (Figure 4). This process mirrors the biological intuition that certain residues and atoms are more critical for binding in a protein−ligand complex than others. The use of attention mechanisms has enabled the identification of AAs in proteins and atoms in a ligand that are highly cross-correlated and appear to physically interact (Figure 4),[148,149] although the degree of success in identifying interacting sites remains to be assessed. Attention has also provided an effective way to "fuse" protein and ligand representations in binding prediction models.[86,121,142,150,151]

Attention is a versatile mechanism that can also be applied to structural information such as the spatial coordinates of individual atoms or contact maps of protein−ligand complexes.[152−154] The structural information of proteins and ligands can be well-represented by a graph with nodes representing AAs or atoms, and edges representing chemical bonds or amino acid contacts. Edges may also represent other predefined relationships or constraints between nodes. Integrating attention mechanisms into Graph Neural Networks (GNNs), a class of NNs specialized for processing graphs, has been increasingly used for the study of PLIs.[155−158] GNNs use "message-passing" whereby each node's embedding is updated iteratively based on information from connected nodes.[159] Each connection can be assigned a weight that quantifies the likelihood of interdependence between connected nodes. For example, a cysteine residue may have a higher weight for a nearby cysteine than a nearby glycine due to the potential to form a disulfide bond between cysteines. GNNs are often augmented further, for example, by the addition of an attention mechanism to prioritize connected nodes during message-passing.[152,153,156−158,160] An example of attention's application to PLI studies is Jin et al.'s CAPLA model,[161] which used a "cross-attention" mechanism to directly correlate tokens within the protein and ligand to one another. The resulting attention weights can display the degree by which each unit relates to one another in order to provide a degree of interpretability, as determined by posthoc evaluation of the attention mechanism.

4.2.3. Transformers. While attention mechanisms have been quite beneficial for the predictive success of NLP methods, the "transformer" architecture pioneered in 2017 has also been instrumental in advancing these capabilities.[134] Transformers

https://doi.org/10.1021/acs.jcim.4c01907
J. Chem. Inf. Model. 2025, 65, 2191−2213

are a type of NN architecture that divides attention mechanisms into multiple parallel operations, each applying a different set of weights to the input data sequence. Several relationships between tokens are captured and processed simultaneously, dramatically improving the efficiency with which human text can be processed. The transformer architecture is the foundation of popular large language models such as ChatGPT[30] and was a key component of DeepMind's AlphaFold system.[83,162] Transformers have become widely used in bioinformatics, for DNA, RNA, and protein sequence analysis, as well as gene-based disease predictions and PLI predictions.[163]

Transformers are designed to solve the problem of "sequence transduction" or the conversion of an input sequence of ordinal data into a predicted output sequence, such as a translated text or a vector representation.[164] In NLP, this is called machine translation, whereby the input sequence, for example, could be a sentence in English and the output sequence is its French counterpart. The transformer is an extension of the so-called "encoder-decoder" architecture (Figure 3B), a state-of-the-art sequence-transduction method commonly used today.[134,137,165,166] The premise of encoder-decoders is that sequentially ordered input data (e.g., English text, protein sequences, SMILES) can be "compressed" or encoded by a lower-dimensional fixed-length vector with minimal information loss. "Encoding" is the process of compressing informative features into a reduced vector representation, effectively capturing implicit rules or structures contained within the data. Typically, in this reduced representation (called the "latent" space), inputs with similarly informative characteristics appear close to one another. These compressed vectors can subsequently be "decoded" or expanded to an output representation of choice to complete the transduction task. These transduction tasks naturally align with the goal of text translation from one language to another.[137,167] Importantly, transformers differ from traditional encoder-decoder models by incorporating the attention mechanism.[134] Attention allows latent representations to vary in length, thus eliminating a fundamental constraint of encoder-decoder models: that every input sequence, regardless of length, be represented by a fixed-length vector in the latent space. Transformers are widely used today[27,163,168] (especially for long input sequences) given their inherent parallel architecture, which makes processing data sets with billions of items feasible. As compared to LSTMs, transformers are architecturally more complex and tend to achieve better performance.[169−172] Even so, transformers may not be the most effective approach, particularly when dealing with small data sets on the order of thousands of items.[173−175] In the biological domain, transformers have been applied to the prediction of protein−protein binding affinities,[176] post-translational modifications,[177] and quantum chemical properties of small molecules.[178]

Early applications of transformers for the study of PLIs involved simply retraining existing models designed for human language inputs;[168,179] surprisingly, these transformers surpassed existing state-of-the-art models for predicting binding affinities.[180,181] As new transformers were developed specifically to handle protein sequence data, predictive performance for PLIs improved.[182−184] These developments included preemptively dividing the texts into subsequences to determine which amino acids contribute to binding and merging embeddings from different transformers to provide multiple representational perspectives. Transformers have been further modified for use with additional data types, such as protein structures and images, as well as for predicting PLI properties beyond binding affinity, e.g., binding poses.[122,185] One such example leverages algebraic topology[186] by converting protein−ligand complex structures into unique one-dimensional sequences.[187] This novel approach was notably able to synthesize embeddings directly for the complex itself and demonstrates space for innovation in further developing the transformer architecture for PLI problems.

So far, transformer-based models have demonstrated mastery at manipulating language constructs for tasks involving reasoning, coding, vision, and mathematics at a level that mirrors human performance.[188] This success has also been extended to molecular biology with the advent of Protein Language Models (PLMs).[20,177,189] Through discerning the probabilities of amino acid appearances given a location and surrounding context, PLMs infer a notion of syntax and semantics for proteins from data sets of protein sequences on the order of millions.[190,191] Once a PLM is trained, the embeddings outputed from the last hidden layers can be transferred to any protein-related prediction task. While the embedding vectors are not fully explainable as to *what* information is contained within, the inferred semantic information is sufficiently preserved in the vector for PLMs to be highly effective in protein-related tasks. PLMs have demonstrated greater efficacy than sequence-based RNNs or LSTMs in predicting specific protein properties, such as structure, function, and cellular localization.[192,193] PLMs also present an opportunity to draw conclusions about small protein families that may not have enough evolutionary information available to perform traditional MSA-based approaches.[194] Although PLMs have not been spotlighted as much as breakthrough structure prediction projects such as AlphaFold,[83] they do see practical use for highly specialized tasks. Such cases include predicting if amino acid variations may preclude genetic disease[195] or identifying cellular sublocalization of peroxisomal proteins.[196]

An example of a transformer encoder is Qian et al.'s CAT-CPI model,[185] which applies a transformer to extract features from protein sequences and images of molecules. For protein sequences, Qian et al. experimented with several different tokenization strategies to be used in conjunction with the transformer to assemble protein subsequences based on frequency among the total corpus of protein sequences. A second transformer encoder was applied to discern long-distance relationships between pixels in the input images of molecules, gathering a different type of information entirely. The use of transformers for two different formats of input demonstrated the variety of use cases for an architecture as versatile as the transformer.

4.3. Fusion of Protein−Ligand Representations: Concatenation or Cross-Attention. Once candidate interacting protein and ligand embeddings are extracted, they need to be fused for an interaction pattern to emerge. Methods for extracting embeddings from protein and ligand sequence data have been the primary focus of the field to date such that approaches for fusion have been somewhat neglected untilrecently. A naive method for fusion is to simply concatenate protein and ligand embedding vectors end-to-end. More refined approaches, though, could involve advanced data structures like graphs, whereby information such as coordinates of protein and ligand is used not only to build a

https://doi.org/10.1021/acs.jcim.4c01907
J. Chem. Inf. Model. 2025, 65, 2191−2213

graph representation but is also incorporated into an attention mechanism to account for local factors such as polarity or size.[154,156,197] A mechanism of "cross-attention" could be incorporated into the fusion approach whereby the importance between the different *token representations* of the protein and ligand are directly calculated[150,151,161] in an attempt to mirror the underlying interaction of a protein with a ligand.[155] Cross-attention has been shown to be at least as competitive in predictive PLI tasks as other fusion methods,[197] and an improvement over the use of separate, independent attention mechanisms for both protein and ligand.[198]

While fusion appears to be a natural and important component for NLP studies of PLIs, some models circumvent the idea of fusion altogether and use protein-only or ligand-only representations explicitly. For example, Wang et al.'s CSConv2D algorithm only embeds ligand information.[199] An individual model is trained separately for each protein to predict that protein's compatible ligands, resulting in the creation of hundreds of models. Although the task was to predict PLIs, protein information was only incorporated indirectly by labeling ligands during model training as either binding to a given protein or not. Nonetheless, protein-only or ligand-only models are rare, with most contemporary NLP-PLI models considering both protein and ligand together through a fusion step.

Mixed-data approaches aimed at *combining different data types* for protein and/or ligand (e.g., sequence + structure; sequence + image;[185] or both sequence and structure for protein + structure for ligand[161]) have further spurred study into which input formats are best for protein and ligand. Mixed-data models may use a variety of architectures such as an LSTM or transformer for a protein sequence and a GNN for ligand structures.[85,86] Combining multiple state-of-the-art embeddings for both sequence and structure has outperformed sequence-only baselines.[86] Despite the increased complexity involved in handling sequence and structural data simultaneously, mixed-data models are advantageous for both the ease-of-use of protein sequences and the completeness of ligand structural representations.

Although underexplored, combining multiple embeddings for each protein and ligand input in the fusion process may be beneficial. It has been suggested that different protein encoders for extracting features may gather different but relevant information to improve predictive outcomes.[200] In the DeepPurpose algorithm, Huang et al. pursued a library approach that offered 15 different protein and ligand embeddings (including transformer and RNN) to be combined and fed into a small NN to generate binary binding and/or continuous binding affinity predictions.[201] This menu-option system enables users to compare feature extractors and find the best protein and ligand embeddings for their research. Another approach is to combine multiple embeddings through operations such as component-wise multiplication or component-wise difference, as each embedding could represent a different set of features.[183,200] Shen et al.'s SVSBI algorithm[183] demonstrated how a higher-order embedding, by concatenating three different transformer embeddings, could outperform several state-of-the-art baselines (including those based on individual transformers alone) in the prediction of binding affinity.

4.4. Prediction of Target Variables. Ultimately, specific research questions must motivate the relevant PLI target variables that will be predicted by constructed ML models. These models often consist of one or more fully connected layers with relatively fewer parameters than the NNs used for feature extraction or fusion. The purpose of these layers is to utilize the latent protein and ligand features to predict an output target variable such as binding affinity or a binary indication of whether a pairing interacts. Thus, the fused protein and ligand embeddings are passed through these final layers to compute the prediction. Embeddings that effectively capture important underlying features can also be applied to predict other useful properties beyond binding affinity such as protein and ligand solubility.[129,130]

4.5. Evaluation. Evaluation is typically performed by comparing statistical metrics between models on the same test data sets. Evaluation metrics vary by task: classification predictions can be assessed via metrics such as precision, recall, and F1 score metrics whereas regression predictions are often evaluated relative to the ground-truth test data via concordance index and mean square error metrics.[98,211] Premade data sets such as PDBBind[96] are frequently bundled with both training and test data sets to enable fair comparisons with other established models. Models aiming to be generalizable across several types of PLIs should ideally be evaluated on several different sets of proteins and ligands.

While ML models can be assessed through the aforementioned statistical metrics, the practical utility of PLI predictive models and their predictive accuracy in real-world cases is best determined by domain experts.[212] For example, if a model is designed to predict binding affinities, a set of predictions generated *in silico* would be best confirmed through *in vitro* experimentation. PLI prediction models could also gain credibility if predictions are validated through physics-based simulation techniques such as molecular docking and molecular dynamics simulations.[213,214] For instance, Chatterjee et al.'s AI-Bind predicted interactions between SARS-CoV-2 viral proteins and human targets, used molecular docking and in vitro/clinical results to confirm these predictions in agreement with existing literature.[214] Similarly, Kalakoti et al.'s TransDTI employed a transformer-based architecture and corroborated predictions for MAP2k and TGF-β inhibitors with molecular dynamics simulations.[213] These methods confirm the accuracy of predicted interactions and align with existing biological knowledge, demonstrating both predictive reliability and practical relevance. Such experimental and simulation-based validation can justify a model's use in the setting where it can be most effective and create opportunities for future interdisciplinary collaboration between ML practitioners and domain experts in computational and experimental biology.

5. CHALLENGES AND FUTURE DIRECTIONS

Advances in generative AI and NLP have revolutionized how we tackle tasks related to human language. Early successes of NLP methods in discerning the "rules" of protein structure (as exemplified by AlphaFold[83]) suggest significant potential for NLP to transform our approach to studying PLIs. While many innovations in the NLP computational toolkit for PLIs have emerged in recent years, several practical hurdles remain, limiting the impact and potential insights derivable from the ML approaches. This section presents an overview of the many challenges confronting the PLI field and suggests various avenues to address them.

5.1. Lack of "True Negatives". A common challenge in today's data-driven ML paradigm is the limited availability of

https://doi.org/10.1021/acs.jcim.4c01907
J. Chem. Inf. Model. 2025, 65, 2191−2213

abundant, high-quality, and labeled data.[215] In PLI studies, there is a particular lack of bona fide "negative examples", i.e., data for ligand-like molecules that do not bind a protein of interest that are critical for model training. If a model is trained on only positive data without any means to adjust for it, there would consequently be a sizable bias toward labeling all test data as positive. For instance, this could be an enzyme paired with a molecule that is obviously not a compatible substrate. In "supervised" ML,[216] models are trained on data with labels of whether a protein−ligand pair is binding or non-binding, and protein−ligand data spanning the full spectrum of interaction/no-interaction are necessary for models to 'learn'. When a similar situation is encountered in other ML tasks, a common approach is to select random data points not explicitly labeled as "positive" and assume them as "negative". However, given the complexity and specificity of PLIs, these are often *trivial* negative examples, since molecules that do not interact with a protein of interest *and* are dissimilar to the "true" ligands embody little information from which ML models can learn. Manually curating protein−ligand pairs that display weak interaction or lower binding affinity is an option for addressing this problem, although this is time-consuming and labor-intensive.

Unfortunately, the availability of informative negative PLI data requires deliberate efforts of domain experts who recognize the importance of generating, curating, and reporting such data, which are rarely publicized or emphasized in the literature regardless of data type.[217−219] This scarcity of negative examples has been observed in several fields.[220] Learning from positive data only or from a mix of positive and unlabeled data is an active field of study, with attempts to apply "unsupervised" and "semi-supervised" methods[221] (see[202,222] for examples related to PLI prediction). Compared with supervised models, un/semi-supervised models typically require larger data sets of tens to hundreds of thousands of PLIs and are more computationally intensive.[202] In cases where negative data does exist albeit at a significantly reduced quantity, classification studies of PLIs can adjust the distribution of ligands to ensure *equal proportions* of positive and negative examples; this has been shown to mitigate over representational bias of positive data.[223] Future studies should resolve the lack of readily available non-interacting protein−ligand pairs, perhaps through mining the scientific literature for meaningful non-binding pairs.

5.2. Diversity Bias in PLI Data Sets. Many PLI data sets display an underlying bias concerning either the diversity or types of proteins and ligands, hindering the effectiveness of ML algorithms. Training with *insufficiently different* data points can lead to poor predictive performance when a model is deployed for real-world examples. For example, binding affinity predictors trained on the popular PDBBind data set[96] with both protein and ligand information represented performed no better than those trained on only protein or only ligand information as inputs,[99] suggesting that some implicit non-informative patterns within the proteins and ligands of the PDBBind data set were learned rather than information concerning the mechanics of binding. The commonly used DUD-E[106] data set of bioactive compounds and respective protein targets demonstrates a similar problem: classification models that appeared highly accurate were found to differentiate binders/non-binders based primarily on their different shape classes and not the embedding of any relevant information about the protein−ligand interface.[99,224] Existing literature suggests that this is a problem of quality over quantity, as memorization-related biases in PLI models are *not* alleviated by merely increasing the data set size or removing overrepresented items.[225] The presence of bias is understandable, given how idiosyncratic research interests in biological or pharmaceutical fields shape the particular proteins and subsets of ligands studied and the type of PLI data generated and made available.

Given that models trained on biased data often fail in practical, real-world prediction tasks, the creation of high-quality, well-balanced, and unbiased PLI data sets is essential to the future of ML-based PLI studies. One way around the experimental challenges of generating sufficient protein−ligand data may be through high-throughput molecular dynamics simulations and/or docking studies using AlphaFold-predicted[83] protein structures. In particular, methods that can accurately estimate binding affinities, such as free-energy perturbation[226] or umbrella sampling,[227,228] appear promising. Although current simulation methods remain time-intensive, advancements in high-performance computing and the growing availability of GPU-based resources are making this approach increasingly feasible[229] and the benefits may be worth investing in this pursuit. These approaches, unrestricted by experimental technical limitations, could be systematically deployed at scale to generate protein−ligand complex structures and binding information, particularly for historically understudied protein classes and ligand categories. These procedures could also be automated, requiring far less human intervention than laboratory experiments, to yield valuable binding pocket information for improved structure-based ML predictions.

5.3. Interpretable and Generalizable Design in PLI Predictions. The open-data movement and the broad accessibility of machine-learning tools have catalyzed the development of numerous predictive models to discern patterns within data. However, these models often rely on complicated weighted operations that are challenging to interpret. Many ML studies fail to consider designing human-friendly interpretations of *how* their models' predictions are calculated. While interpretability is not a requirement for a high-performing model, a lack of interpretability can be a hurdle to the acceptance of such models as users may doubt the trustworthiness of a "black-box" model.[48] One potential approach for bridging the "explainability" gap is the use of attention weights to corroborate existing protein−ligand contacts (cf. Figure 4).[86,121,142,150,151] Attention weights highlight regions in PLI models that converge with higher weight values but may result in "false positives" whereby higher binding weights are inadvertently assigned to non-binding regions. Unfortunately, a systematic assessment of "false positives" in attention weights has yet to be performed, leaving it unclear whether they are a reliable metric.[98] Such false positives are one facet of a larger debate on whether attention weights provide sufficient explanatory power for PLI models.[230−232]

While NLP presents attention mechanisms as one possible avenue, other methods of explainability are starting to be explored for interpretable PLI predictions. One example is a game-theory approach to compute "Shapley values", which quantify the importance of individual features by evaluating each feature's contribution to the final prediction across all possible combinations of those features.[233,234] Visualizations are another intuitive approach to aid our understanding of

https://doi.org/10.1021/acs.jcim.4c01907
J. Chem. Inf. Model. 2025, 65, 2191−2213

predictive models. For example, graph visualization can depict the predicted bonds between an interacting protein and ligand, and "saliency maps"[235] can highlight specific subregions of protein and ligand that are the most influential in a prediction, by discerning how subtle perturbations in individual input features affect the output. Several avenues for interpretability remain to be tested,[236] but none have been established as standard. Determining a reliable interpretability method for PLI prediction models will be critical for the field.

Another important aspect of modeling protein−ligand interactions is *generalizability*—or how well a model performs on unseen data. During the evaluation of machine learning models, test sets are typically selected with a presumed *a priori* understanding of the expected sample distribution to ensure accurate evaluation. However, the true sample distribution may differ, and it is important that a model can accommodate potentially unseen variations of input data. Although many PLIs have been identified to date, the full scope and distribution of all possible protein interactions remains unknown. However, there exist several means through which generalizability can be improved, including the production of additional data novel examples, reducing diversity bias, different strategies for splitting data into training and test sets, or alternative training schema.[214,237]

A similar task for which highly generalizable models have emerged is protein language modeling, where patterns are observed from analyzing massive data sets of protein sequences for purposes such as predicting protein stability or studying the evolutionary relationships between proteins.[189,238] While protein language models (PLMs) have achieved great predictive success, they require immense amounts of diverse data. Although the total number of unique tokens is much smaller than for human languages, protein sequences may contain far more tokens in total for data sets than for human languages. For example, UniProt's UniRef50,[93,239] totals over 9.5 billion amino acids in length, and assuming that each AA is a token, that is a substantially larger corpus than most NLP data sets.[194] Currently, there may not be enough data available forPLI studies to train on the same scale as in PLM studies. However, with high-throughput analysis and the natural progression of PLI prediction studies, this may eventually be feasible.

5.4. The Insufficiency of an NLP-Only Approach for PLI Studies? While NLP offers beneficial strategies for the study of PLIs, it is not a panacea, and there may be opportunities from other disciplines within computer science to contribute to the study of PLIs. For example, computer vision techniques may be favorable to use in handling structural information over NLP techniques designed to handle text.[240] Complementary approaches to NLP such as multimodal methods that integrate information from images and textual descriptions, can be applied to capture richer representations.[185,204] More advanced architectures, such as those exploring generative modeling, offer further avenues for integrating diverse data sources.[202] While the success of such hybrid strategies has yet to exceed the performance of other neural networks in PLI predictive tasks,[241] they demonstrate the potential for innovation by taking inspiration from other subdomains of ML and computer science beyond NLP.

Approaches informed by a deep domain-specific understanding have led to the practical success of ML methods. This has been demonstrated by the AlphaFold initiative in which nuanced awareness guided which biological features merited focus.[83,162] For example, the researchers behind AlphaFold-Multimer's protein−protein interaction prediction algorithm[242] created an interface-aware protocol that crops protein structures to reduce computational burden and decrease the representation of non-interfacial amino acids while maintaining an important balance of interacting and non-interacting regions. Although AlphaFold-Multimer performs very well on predicting protein complexes in several cases,[243,244] preliminary results suggest that the more recently released AlphaFold 3 may offer further improvements.[89]

Whereas AlphaFold2 used a highly specified geometry-based module to generate protein structures, AlphaFold3[89] incorporates a diffusion model similar to those that are popular in image generation tasks.[245,246] The diffusion model[247] of AlphaFold3 begins with a "noise" cloud of atoms placed at random and then iteratively converges to an accurate representation of the input sequences' 3D structure. This initial inclusion of "noise" induces the model to refine the local structure rather than quickly converging to a local minimum. Whereas the previous AlphaFold2 geometry-based module was specific to proteins alone, a simplified diffusion model allows for the prediction of protein interactions with biological objects such as nucleic acids and small molecules. AlphaFold3 has been a significant advance, outperforming both molecular docking tools and diffusion-based-only models on structure prediction tasks.[248] The recent release of AlphaFold3's open-source code makes the model highly accessible, allowing researchers to examine new predictions of protein interactions.

Due to the recency of AlphaFold3's release in May 2024, independent validation of AlphaFold3's predictions has thus far been limited. While studies have looked into AlphaFold3's limitations on protein−protein interactions[249] and protein-nucleic acid interactions,[250,251] the limitations associated with predicting PLIs are unclear. Studies to date suggest that AlphaFold3 has difficulties predicting accurate ligand-binding poses, pocket shape, and the assembly of domains for flexible proteins.[252] In a case study of flexible domains of receptor proteins, AlphaFold3 was shown to generate plausible but not the most stable conformations of proteins. AlphaFold3 predictions represent just onestable, averaged conformation based on inferred patterns within the training data, while ground-truth experimental methods like cryo-EM capture stable states that may be influenced by particular environmental contexts. Other possible limitations include how AlphaFold3 predicts some categories of interactions more accurately than others, the possibility of model hallucinations, and restrictive hardware requirements to run the model.[251,253] AlphaFold3 is a powerful tool that is effective for general use, but only time will tell how it, along with other competing tools like RoseTTAFold[254] and OpenFold,[255] will perform in future PLI studies.

The study of PLIs may eventually outgrow NLP methods, but for the foreseeable future, advances in NLP have established a strong foundation for processing texts representing biological objects. NLP still plays a key role in text-driven tasks such as the *de novo* generation of SMILES strings for automated molecular design.[256−258] Regardless, machine learning-based PLI studies will need to rely on close collaborations between experts in both biological and computational domains to catalyze further innovations in what is an interdisciplinary goal.

https://doi.org/10.1021/acs.jcim.4c01907
J. Chem. Inf. Model. 2025, 65, 2191−2213

6. CONCLUSION

Natural language processing (NLP), a subdiscipline of machine learning (ML), offers myriad tools for both experimental and computational researchers to accelerate exploratory studies in structural biology. The prediction of protein−ligand interactions (PLIs) can be reimagined through NLP by treating protein and ligand representations like text. Protein sequences resemble readable text with inherent meaning to be inferred, while chemical text formats such as theSMILES allows for limited NLP application to small molecules. Current efforts seek to leverage multiple or augmented SMILES representations to address these limitations.

Approaches to tackling PLI prediction tasks using sequence-only data, structural data, or a combination of both, have all yielded successful predictions, although the advantage of one input data type over others remains unclear. Sequence-only data approaches are simple and amenable to NLP but requires a significant abstraction of chemical information; structural data is informationally rich but computationally expensive to handle, while combining both sequence and structural data types offers balance at the expense of complexity.

The transformer architecture, in general, and attention mechanisms, in particular, have yielded the most promising NLP-based PLI prediction results to date. Incorporating complementary data (e.g., multiple sequence alignments, ligand polarities, etc.) can improve predictive success but at a significant increase in computational cost. After data selection and preparation, all methods have followed a general ML Extract-Fuse-Predict model creation framework of: (i) extracting feature embeddings for protein and ligand, (ii) fusing protein and ligand embeddings, and (iii) making predictions based on the created ML model.

The first step of data set selection is crucial for any ML-based study of PLIs, and no single data set can satisfy all needs, with many suffering from missing data or the lack of negative data. Data sets must align with specific research goals, requiring thoughtful consideration as to what inputs, formats, and target variable(s) are selected for the ML model. Appropriate tokenization and embedding methods, which convert proteins and ligands into numerical representations, are vital for a successful model. Atoms or amino acids typically serve as tokens, and neural networks (NNs) have helped identify hidden patterns more quickly. NLP-inspired NNs, such as Long Short-Term Memory NNs, along with attention mechanisms and transformer architectures, have shown particular promise for understanding PLIs. A modular approach combining multiple embeddings can capture diverse perspectives, improving prediction accuracy, especially for the prediction of binding affinities. After appropriate embeddings are obtained, graph-based methods and cross-attention mechanisms have been shown to be effective in combining data from diverse sources.

NLP has been central to ML studies of PLIs and has yielded promising results, although many challenges remain. Explaining ML model predictions is essential for their trustworthiness and acceptance. Current explanatory metrics, such as attention weights and Shapley values, offer some degree of interpretability but remain to be fully validated. A major challenge is the lack of well-annotated non-binding protein−ligand pairs, or "negative data". Unsupervised methods or manually curated selections of non-binding pairs are potential solutions. Popular PLI data sets may contain biases that cause models to "memorize" idiosyncratic patterns rather than "learn" the true mechanics of PLIs. Ensuring balanced training data sets (positive vs. negative data, number of proteins vs. ligands, etc.) would be essential to avoid such bias.

As protein and ligand sequence representations differ from human language, it may be difficult to capture their complexity with NLP methods alone, especially as much of the variation in protein function can often be explained by simple amino acid interactions rather than complex higher-order interactions.[259] While NLP has contributed significantly to the advance of PLI studies, future improvements may come from both modifying machine learning architectures and incorporating nuanced biological domain knowledge. For instance, the researchers behind AlphaFold-Multimer's protein−protein interaction prediction algorithm[242] created an interface-aware protocol that crops protein structures to reduce computational burden and the representation of non-interfacial amino acids while maintaining an important balance of interacting and non-interacting regions. Some researchers have also integrated mass spectrometry data to improve model predictions of protein complexes.[260] More recently in AlphaFold3,[89] a diffusion layer has been added to Alphafold's previous workflow to enable the study of PLIs. Time will tell to what degree AlphaFold3 will advance predictions of PLIs but progress in PLI research will undoubtedly require interdisciplinary collaborations between computer scientists, chemists, and biologists.

Although it is best practice to evaluate model performance against ground-truth experimental results or results from physics-based computer simulations, few studies to date have benchmarked their model predictions in this way. Formal competition may prove to be a promising avenue for future advances in PLI prediction. Other grand challenges, such as protein folding and protein assembly, have had significant progress facilitated through competitions like Critical Assessment of Structural Prediction (CASP)[100] and Critical Assessment of Prediction of Interactions (CAPRI).[101,102] These well-adjudicated competitions use unpublished test sets for objective model comparisons. Milestone algorithms like AlphaFold[261] and RosettaFold[254] were formed, improved, and refined through the crucible of such contests. Creating a dedicated competition devoted to protein−ligand interactions could similarly inspire innovation and catalyze seminal algorithmic advances for PLI prediction.

■ ASSOCIATED CONTENT

Data Availability Statement

No data or software was generated for this review.

■ AUTHOR INFORMATION

Corresponding Authors

Hong Xiao − *Department of Computer and Information Science and Institute for Data Science, University of Mississippi, University, Mississippi 38677, United States*; Email: hxiao1@olemiss.edu

Jing Li − *Department of BioMolecular Sciences, School of Pharmacy, University of Mississippi, University, Mississippi 38677, United States*; orcid.org/0000-0003-3277-6818; Email: jli15@olemiss.edu

Erik F. Y. Hom − *Department of Biology and Center for Biodiversity and Conservation Research, University of Mississippi, University, Mississippi 38677, United States*; orcid.org/0000-0003-0964-0031; Email: erik@fyhom.com

https://doi.org/10.1021/acs.jcim.4c01907
J. Chem. Inf. Model. 2025, 65, 2191−2213

Authors

James Michels − *Department of Computer and Information Science, University of Mississippi, University, Mississippi 38677, United States*

Ramya Bandarupalli − *Department of BioMolecular Sciences, School of Pharmacy, University of Mississippi, University, Mississippi 38677, United States*

Amin Ahangar Akbari − *Department of BioMolecular Sciences, School of Pharmacy, University of Mississippi, University, Mississippi 38677, United States*

Thai Le − *Department of Computer Science, Indiana University, Bloomington, Indiana 47408, United States*

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jcim.4c01907

Author Contributions

JM: Conceptualization (lead); investigation (lead); project administration (lead); visualization (lead); writing—original draft preparation; writing—review and editing (co-lead). **RB**: Investigation (supporting); visualization (supporting); review and editing (equal). **AAA**: Investigation (supporting); review and editing (equal). **TL**: Conceptualization (supporting); funding acquisition (equal). **HX**: Conceptualization (supporting); supervision (supporting); review and editing (equal). **JL**: Conceptualization (supporting); funding acquisition (equal); review and editing (equal). **EH**: Conceptualization (supporting); funding acquisition (equal); project administration (supporting); supervision (lead); writing—review and editing (co-lead).

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

This work was supported in part by NIGMS/NIH Institutional Development Award (IDeA) #P20GM130460 to J.L, NSF award #1846376 to E.F.Y.H, and University of Mississippi Data Science/AI Research Seed Grant award #SB3002 IDS RSG-03 to J.M., J.L., T.L, and E.F.Y.H.

■ REFERENCES

(1) Songyang, Z.; Cantley, L. C. Recognition and specificity in protein tyrosine kinase-mediated signalling. *Trends Biochem. Sci.* **1995**, *20*, 470−475.

(2) Johnson, L. N.; Lowe, E. D.; Noble, M. E.; Owen, D. J. The Eleventh Datta Lecture. The structural basis for substrate recognition and control by protein kinases. *FEBS Lett.* **1998**, *430*, 1−11.

(3) Kristiansen, K. Molecular mechanisms of ligand binding, signaling, and regulation within the superfamily of G-protein-coupled receptors: molecular modeling and mutagenesis approaches to receptor structure and function. *Pharmacol. Ther.* **2004**, *103*, 21−80.

(4) West, I. C. What determines the substrate specificity of the multi-drug-resistance pump? *Trends Biochem. Sci.* **1990**, *15*, 42−46.

(5) Vivier, E.; Malissen, B. Innate and adaptive immunity: specificities and signaling hierarchies revisited. *Nat. Immunol.* **2005**, *6*, 17−21.

(6) Desvergne, B.; Michalik, L.; Wahli, W. Transcriptional regulation of metabolism. *Physiol. Rev.* **2006**, *86*, 465−514.

(7) Atkinson, D. E. Biological feedback control at the molecular level: Interaction between metabolite-modulated enzymes seems to be a major factor in metabolic regulation. *Science* **1965**, *150*, 851−857.

(8) Huang, S.-Y.; Zou, X. Advances and challenges in protein-ligand docking. *Int. J. Mol. Sci.* **2010**, *11*, 3016−3034.

(9) Chaires, J. B. Calorimetry and thermodynamics in drug design. *Annu. Rev. Biophys.* **2008**, *37*, 135−151.

(10) Serhan, C. N. Signalling the fat controller. *Nature* **1996**, *384*, 23−24.

(11) McAllister, C. H.; Beatty, P. H.; Good, A. G. Engineering nitrogen use efficient crop plants: the current status: Engineering nitrogen use efficient crop plants. *Plant Biotechnol. J.* **2012**, *10*, 1011−1025.

(12) Goldsmith, M.; Tawfik, D. S. Enzyme engineering: reaching the maximal catalytic efficiency peak. *Curr. Opin. Struct. Biol.* **2017**, *47*, 140−150.

(13) Vajda, S.; Guarnieri, F. Characterization of protein-ligand interaction sites using experimental and computational methods. *Curr. Opin. Drug Discovery Devel.* **2006**, *9*, 354−362.

(14) Du, X.; Li, Y.; Xia, Y.-L.; Ai, S.-M.; Liang, J.; Sang, P.; Ji, X.-L.; Liu, S.-Q. Insights into protein-ligand interactions: Mechanisms, models, and methods. *Int. J. Mol. Sci.* **2016**, *17*, 144.

(15) Fan, F. J.; Shi, Y. Effects of data quality and quantity on deep learning for protein-ligand binding affinity prediction. *Bioorg. Med. Chem.* **2022**, *72*, 117003.

(16) Sousa, S. F.; Ribeiro, A. J. M.; Coimbra, J. T. S.; Neves, R. P. P.; Martins, S. A.; Moorthy, N. S. H. N.; Fernandes, P. A.; Ramos, M. J. Protein-Ligand Docking in the New Millennium A Retrospective of 10 Years in the Field. *Curr. Med. Chem.* **2013**, *20*, 2296−2314.

(17) Morris, C. J.; Corte, D. D. Using molecular docking and molecular dynamics to investigate protein-ligand interactions. *Mod. Phys. Lett. B* **2021**, *35*, 2130002.

(18) Lecina, D.; Gilabert, J. F.; Guallar, V. Adaptive simulations, towards interactive protein-ligand modeling. *Sci. Rep.* **2017**, *7*, 8466.

(19) Ikebata, H.; Hongo, K.; Isomura, T.; Maezono, R.; Yoshida, R. Bayesian molecular design with a chemical language model. *J. Comput. Aided Mol. Des.* **2017**, *31*, 379−391.

(20) Rives, A.; Meier, J.; Sercu, T.; Goyal, S.; Lin, Z.; Liu, J.; Guo, D.; Ott, M.; Zitnick, C. L.; Ma, J.; Fergus, R. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proc. Natl. Acad. Sci. U. S. A.* **2021**, DOI: 10.1073/pnas.2016239118.

(21) Cao, Y.; Shen, Y. TALE: Transformer-based protein function Annotation with joint sequence−Label Embedding. *Bioinformatics* **2021**, *37*, 2825−2833.

(22) Wang, S.; Guo, Y.; Wang, Y.; Sun, H.; Huang, J. SMILES-BERT: Large Scale Unsupervised Pre-Training for Molecular Property Prediction. *Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics. New York, NY, USA* **2019**, 429−436.

(23) Chithrananda, S.; Grand, G.; Ramsundar, B. ChemBERTa: Large-Scale Self-Supervised Pretraining for Molecular Property Prediction. *arXiv*, 2020.

(24) Kumar, N.; Acharya, V. Machine intelligence-driven framework for optimized hit selection in virtual screening. *J. Cheminform.* **2022**, *14*, 48.

(25) Erikawa, D.; Yasuo, N.; Sekijima, M. MERMAID: an open source automated hit-to-lead method based on deep reinforcement learning. *J. Cheminform.* **2021**, *13*, 94.

(26) Zhou, M.; Duan, N.; Liu, S.; Shum, H.-Y. Progress in neural NLP: Modeling, learning, and reasoning. *Engineering (Beijing)* **2020**, *6*, 275−290.

(27) Patwardhan, N.; Marrone, S.; Sansone, C. Transformers in the real world: A survey on NLP applications. *Inf.* **2023**, *14*, 242.

(28) Bijral, R. K.; Singh, I.; Manhas, J.; Sharma, V. Exploring Artificial Intelligence in Drug Discovery: A Comprehensive Review. *Arch. Comput. Methods Eng.* **2022**, *29*, 2513−2529.

(29) Ray, P. P. ChatGPT: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope. *Internet of Things and Cyber-Physical Systems* **2023**, *3*, 121−154.

(30) Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. Improving language understanding by generative pre-training. https://www.mikecaptain.com/resources/pdf/GPT-1.pdf, Accessed: 2023−10−27.

(31) Goodside, R, Papay, Meet Claude: Anthropic's Rival to ChatGPT. https://scale.com/blog/chatgpt-vs-claude, 2023.

(32) Bing Copilot. Bing Copilot; https://copilot.microsoft.com/.
(33) Rahul; Adhikari, S.; Monika. NLP based Machine Learning Approaches for Text Summarization. *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)* **2020**, 535−538.
(34) Nasukawa, T.; Yi, J. Sentiment analysis: capturing favorability using natural language processing. *Proceedings of the 2nd international conference on Knowledge capture. New York, NY, USA* **2003**, 70−77.
(35) Lample, G.; Charton, F. Deep Learning for Symbolic Mathematics. *arXiv*, 2019.
(36) Feng, Z.; Guo, D.; Tang, D.; Duan, N.; Feng, X.; Gong, M.; Shou, L.; Qin, B.; Liu, T.; Jiang, D.; Zhou, M. CodeBERT: APre-Trained Model for Programming and Natural Languages. *arXiv*, 2020.
(37) Mielke, S. J.; Alyafeai, Z.; Salesky, E.; Raffel, C.; Dey, M.; Gallé, M.; Raja, A.; Si, C.; Lee, W. Y.; Sagot, B.; Tan, S. Between words and characters: A Brief History of Open-Vocabulary Modeling and Tokenization in NLP. *arXiv*, 2021.
(38) Camacho-Collados, J.; Pilehvar, M. T. From word to sense embeddings: A survey on vector representations of meaning. *J. Artif. Intell. Res.* **2018**, *63*, 743−788.
(39) Ashok, V. G.; Feng, S.; Choi, Y. Success with style: Using writing style to predict the success of novelsd.
(40) Barberá, P.; Boydstun, A. E.; Linn, S.; McMahon, R.; Nagler, J. Automated text classification of news articles: A practical guide. *Polit. Anal.* **2021**, *29*, 19−42.
(41) Wang, H.; Wu, H.; He, Z.; Huang, L.; Church, K. W. Progress in machine translation. *Engineering (Beijing)* **2022**, *18*, 143−153.
(42) Sønderby, S. K.; Winther, O. Protein Secondary Structure Prediction with Long Short Term Memory Networks. *arXiv*, 2014.
(43) Guo, Y.; Li, W.; Wang, B.; Liu, H.; Zhou, D. DeepACLSTM: deep asymmetric convolutional long short-term memory neural models for protein secondary structure prediction. *BMC Bioinformatics* **2019**, *20*, 341.
(44) Bhasuran, B.; Natarajan, J. Automatic extraction of gene-disease associations from literature using joint ensemble learning. *PLoS One* **2018**, *13*, e0200699.
(45) Pang, M.; Su, K.; Li, M. Leveraging information in spatial transcriptomics to predict super-resolution gene expression from histology images in tumors. *bioRxiv*, 2021, 2021.11.28.470212.
(46) Bouatta, N.; Sorger, P.; AlQuraishi, M. Protein structure prediction by AlphaFold2: are attention and symmetries all you need? *Acta Crystallogr. D Struct Biol.* **2021**, *77*, 982−991.
(47) Skolnick, J.; Gao, M.; Zhou, H.; Singh, S. AlphaFold 2: Why It Works and Its Implications for Understanding the Relationships of Protein Sequence, Structure, and Function. *J. Chem. Inf. Model.* **2021**, *61*, 4827−4831.
(48) Adadi, A.; Berrada, M. Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). *IEEE Access* **2018**, *6*, 52138.
(49) Box, G. E. P. Science and Statistics. *J. Am. Stat. Assoc.* **1976**, *71*, 791−799.
(50) Geirhos, R.; Jacobsen, J.-H.; Michaelis, C.; Zemel, R.; Brendel, W.; Bethge, M.; Wichmann, F. A. Shortcut learning in deep neural networks. *Nature Machine Intelligence* **2020**, *2*, 665−673.
(51) Outeiral, C.; Nissley, D. A.; Deane, C. M. Current structure predictors are not learning the physics of protein folding. *Bioinformatics* **2022**, *38*, 1881−1887.
(52) Steels, L. Modeling the cultural evolution of language. *Phys. Life Rev.* **2011**, *8*, 339−356.
(53) Maurya, H. C.; Gupta, P.; Choudhary, N. Natural language ambiguity and its effect on machine learning. *Int. J. Modern Eng. Res.* **2015**, *5*, 25−30.
(54) Tenney, I.; Xia, P.; Chen, B.; Wang, A.; Poliak, A.; McCoy, R. T.; Kim, N.; Van Durme, B.; Bowman, S. R.; Das, D.; Pavlick, E. What do you learn from context? Probing for sentence structure in contextualized word representations. *arXiv*, 2019.
(55) Miyagawa, S.; Berwick, R. C.; Okanoya, K. The emergence of hierarchical structure in human language. *Front. Psychol.* **2013**, *4*, 71.
(56) Liu, H.; Xu, C.; Liang, J. Dependency distance: A new perspective on syntactic patterns in natural languages. *Phys. Life Rev.* **2017**, *21*, 171−193.
(57) Frank, S. L.; Bod, R.; Christiansen, M. H. How hierarchical is language use? *Proc. Biol. Sci.* **2012**, *279*, 4522−4531.
(58) Oesch, N.; Dunbar, R. I. M. The emergence of recursion in human language: Mentalising predicts recursive syntax task performance. *J. Neurolinguistics* **2017**, *43*, 95−106.
(59) Ferruz, N.; Höcker, B. Controllable protein design with language models. *Nature Machine Intelligence* **2022**, *4*, 521−532.
(60) Ofer, D.; Brandes, N.; Linial, M. The language of proteins: NLP, machine learning & protein sequences. *Comput. Struct. Biotechnol. J.* **2021**, *19*, 1750−1758.
(61) Ptitsyn, O. B. How does protein synthesis give rise to the 3D-structure? *FEBS Lett.* **1991**, *285*, 176−181.
(62) Yu, L.; Tanwar, D. K.; Penha, E. D. S.; Wolf, Y. I.; Koonin, E. V.; Basu, M. K. *Grammar of protein domain architectures* **2019**, *116*, 3636−3645.
(63) Petsko, G. A.; Ringe, D. *Protein Structure and Function*; Primers in Biology; Blackwell Publishing: London, England, 2003.
(64) Shenoy, S. R.; Jayaram, B. Proteins: sequence to structure and function-current status. *Curr. Protein Pept. Sci.* **2010**, *11*, 498−514.
(65) Takahashi, M.; Maraboeuf, F.; Nordén, B. Locations of functional domains in the RecA protein. Overlap of domains and regulation of activities. *Eur. J. Biochem.* **1996**, *242*, 20−28.
(66) Liang, W.; KaiYong, Z. Detecting "protein words" through unsupervised word segmentation. *arXiv*, 2014.
(67) Kuntz, I. D.; Crippen, G. M.; Kollman, P. A.; Kimelman, D. Calculation of protein tertiary structure. *J. Mol. Biol.* **1976**, *106*, 983−994.
(68) Rodrigue, N.; Lartillot, N.; Bryant, D.; Philippe, H. Site interdependence attributed to tertiary structure in amino acid sequence evolution. *Gene* **2005**, *347*, 207−217.
(69) Eisenhaber, F.; Persson, B.; Argos, P. Protein structure prediction: recognition of primary, secondary, and tertiary structural features from amino acid sequence. *Crit. Rev. Biochem. Mol. Biol.* **1995**, *30*, 1−94.
(70) Garfield, E. Chemico-linguistics: computer translation of chemical nomenclature. *Nature* **1961**, *192*, 192.
(71) Wigh, D. S.; Goodman, J. M.; Lapkin, A. A. A review of molecular representation in the age of machine learning. *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **2022**, DOI: 10.1002/wcms.1603.
(72) Weininger, D. SMILES, a chemical language and information system. 1 Introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci.* **1988**, *28*, 31−36.
(73) Wang, Y.; Xiao, J.; Suzek, T. O.; Zhang, J.; Wang, J.; Bryant, S. H. PubChem: a public information system for analyzing bioactivities of small molecules. *Nucleic Acids Res.* **2009**, *37*, W623−33.
(74) Degtyarenko, K.; de Matos, P.; Ennis, M.; Hastings, J.; Zbinden, M.; McNaught, A.; Alcántara, R.; Darsow, M.; Guedj, M.; Ashburner, M. ChEBI: a database and ontology for chemical entities of biological interest. *Nucleic Acids Res.* **2007**, *36*, D344−50.
(75) Wishart, D. S.; Knox, C.; Guo, A. C.; Cheng, D.; Shrivastava, S.; Tzur, D.; Gautam, B.; Hassanali, M. DrugBank: a knowledgebase for drugs, drug actions and drug targets. *Nucleic Acids Res.* **2008**, *36*, D901−6.
(76) Wang, X.; Hao, J.; Yang, Y.; He, K. Natural language adversarial defense through synonym encoding. *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence* **2021**, 823−833.
(77) Bjerrum, E. J. SMILES Enumeration as Data Augmentation for Neural Network Modeling of Molecules. *arXiv*, 2017.
(78) Lee, I.; Nam, H. Infusing Linguistic Knowledge of SMILES into Chemical Language Models. *arXiv*, 2022.
(79) Skinnider, M. A. Invalid SMILES are beneficial rather than detrimental to chemical language models. *Nature Mach. Intell.* **2024**, *6*, 437.
(80) O'Boyle, N.; Dalke, A. DeepSMILES: An Adaptation of SMILES for Use in Machine-Learning of Chemical Structures. *ChemRxiv*, 2018.

(81) Krenn, M.; Häse, F.; Nigam, A.; Friederich, P.; Aspuru-Guzik, A. Self-referencing embedded strings (SELFIES): A 100% robust molecular string representation. *Mach. Learn.: Sci. Technol.* **2020**, *1*, 045024.
(82) Gohlke, H.; Mannhold, R.; Kubinyi, H.; Folkers, G. In *Protein-Ligand Interactions*; Gohlke, H., Ed.; Methods and Principles in Medicinal Chemistry; Wiley-VCH Verlag: Weinheim, Germany, 2012.
(83) Jumper, J.; et al. Highly accurate protein structure prediction with AlphaFold. *Nature* **2021**, *596*, 583−589.
(84) Landrum, G. RDKit: A software suite for cheminformatics, computational chemistry, and predictive modeling. http://www.rdkit.org/RDKit_Overview.pdf, 2013; Accessed: 2023−12−13.
(85) Mukherjee, S.; Ghosh, M.; Basuchowdhuri, P. *Proceedings of the 2022 SIAM International Conference on Data Mining (SDM)*; Proceedings; Society for Industrial and Applied Mathematics, 2022; pp 729−737.
(86) Chen, L.; Tan, X.; Wang, D.; Zhong, F.; Liu, X.; Yang, T.; Luo, X.; Chen, K.; Jiang, H.; Zheng, M. TransformerCPI: improving compound−protein interaction prediction by sequence-based deep learning with self-attention mechanism and label reversal experiments. *Bioinformatics* **2020**, *36*, 4406−4414.
(87) Aly Abdelkader, G.; Ngnamsie Njimbouom, S.; Oh, T.-J.; Kim, J.-D. ResBiGAAT: Residual Bi-GRU with attention for protein-ligand binding affinity prediction. *Comput. Biol. Chem.* **2023**, *107*, 107969.
(88) Li, Q.; Zhang, X.; Wu, L.; Bo, X.; He, S.; Wang, S. PLA-MoRe: AProtein−Ligand Binding Affinity Prediction Model via Comprehensive Molecular Representations. *J. Chem. Inf. Model.* **2022**, *62*, 4380−4390.
(89) Abramson, J. Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature* **2024**, *636*, E4.
(90) Gaulton, A.; Bellis, L. J.; Bento, A. P.; Chambers, J.; Davies, M.; Hersey, A.; Light, Y.; McGlinchey, S.; Michalovich, D.; Al-Lazikani, B.; Overington, J. P. ChEMBL: a large-scale bioactivity database for drug discovery. *Nucleic Acids Res.* **2012**, *40*, D1100−7.
(91) Wishart, D. S.; Knox, C.; Guo, A. C.; Shrivastava, S.; Hassanali, M.; Stothard, P.; Chang, Z.; Woolsey, J. DrugBank: a comprehensive resource for in silico drug discovery and exploration. *Nucleic Acids Res.* **2006**, *34*, D668−72.
(92) Berman, H. M.; Westbrook, J.; Feng, Z.; Gilliland, G.; Bhat, T. N.; Weissig, H.; Shindyalov, I. N.; Bourne, P. E. The Protein Data Bank. *Nucleic Acids Res.* **2000**, *28*, 235−242.
(93) Acids research, N. 2017 UniProt: the universal protein knowledgebase. *Nucleic Acids Res.* **2017**, *45*, D158−D169.
(94) Davis, M. I.; Hunt, J. P.; Herrgard, S.; Ciceri, P.; Wodicka, L. M.; Pallares, G.; Hocker, M.; Treiber, D. K.; Zarrinkar, P. P. Comprehensive analysis of kinase inhibitor selectivity. *Nat. Biotechnol.* **2011**, *29*, 1046−1051.
(95) Tang, J.; Szwajda, A.; Shakyawar, S.; Xu, T.; Hintsanen, P.; Wennerberg, K.; Aittokallio, T. Making sense of large-scale kinase inhibitor bioactivity data sets: a comparative and integrative analysis. *J. Chem. Inf. Model.* **2014**, *54*, 735−743.
(96) Wang, R.; Fang, X.; Lu, Y.; Wang, S. The PDBbind database: collection of binding affinities for protein-ligand complexes with known three-dimensional structures. *J. Med. Chem.* **2004**, *47*, 2977−2980.
(97) Chen, S.; Zhang, S.; Fang, X.; Lin, L.; Zhao, H.; Yang, Y. Protein complex structure modeling by cross-modal alignment between cryo-EM maps and protein sequences. *Nat. Commun.* **2024**, *15*, 8808.
(98) Bishop, M.C. *Pattern Recognition and Machine Learning*, 1st ed.; Information Science and Statistics; Springer: New York, NY, 2006.
(99) Yang, J.; Shen, C.; Huang, N. Predicting or Pretending: Artificial Intelligence for Protein-Ligand Interactions Lack of Sufficiently Large and Unbiased Datasets. *Front. Pharmacol.* **2020**, *11*, 69.
(100) Kryshtafovych, A.; Schwede, T.; Topf, M.; Fidelis, K.; Moult, J. Critical assessment of methods of protein structure prediction (CASP)−Round XIII. *Proteins* **2019**, *87*, 1011−1020.
(101) Janin, J.; Henrick, K.; Moult, J.; Eyck, L. T.; Sternberg, M. J. E.; Vajda, S.; Vakser, I.; Wodak, S. J. CAPRI: A Critical Assessment of PRedicted Interactions. *Proteins* **2003**, *52*, 2−9.
(102) Lensink, M. F.; Nadzirin, N.; Velankar, S.; Wodak, S. J. Modeling protein-protein, protein-peptide, and protein-oligosaccharide complexes: CAPRI 7th edition. *Proteins* **2020**, *88*, 916−938.
(103) Schomburg, I.; Chang, A.; Schomburg, D. BRENDA, enzyme data and metabolic information. *Nucleic Acids Res.* **2002**, *30*, 47−49.
(104) Liu, T.; Lin, Y.; Wen, X.; Jorissen, R. N.; Gilson, M. K. BindingDB: a web-accessible database of experimentally determined protein-ligand binding affinities. *Nucleic Acids Res.* **2007**, *35*, D198−201.
(105) Amemiya, T.; Koike, R.; Kidera, A.; Ota, M. PSCDB: a database for protein structural change upon ligand binding. *Nucleic Acids Res.* **2012**, *40*, D554−8.
(106) Mysinger, M. M.; Carchia, M.; Irwin, J. J.; Shoichet, B. K. Directory of useful decoys, enhanced (DUD-E): better ligands and decoys for better benchmarking. *J. Med. Chem.* **2012**, *55*, 6582−6594.
(107) Warren, G. L.; Do, T. D.; Kelley, B. P.; Nicholls, A.; Warren, S. D. Essential considerations for using protein-ligand structures in drug discovery. *Drug Discovery Today* **2012**, *17*, 1270−1281.
(108) Puvanendrampillai, D.; Mitchell, J. B. O. L/D Protein Ligand Database (PLD): additional understanding of the nature and specificity of protein-ligand complexes. *Bioinformatics* **2003**, *19*, 1856−1857.
(109) Wang, C.; Hu, G.; Wang, K.; Brylinski, M.; Xie, L.; Kurgan, L. PDID: database of molecular-level putative protein-drug interactions in the structural human proteome. *Bioinformatics* **2016**, *32*, 579−586.
(110) Zhu, M.; Song, X.; Chen, P.; Wang, W.; Wang, B. dbHDPLS: A database of human disease-related protein-ligand structures. *Comput. Biol. Chem.* **2019**, *78*, 353−358.
(111) Gao, M.; Moumbock, A. F. A.; Qaseem, A.; Xu, Q.; Günther, S. CovPDB: a high-resolution coverage of the covalent protein-ligand interactome. *Nucleic Acids Res.* **2022**, *50*, D445−D450.
(112) Ammar, A.; Cavill, R.; Evelo, C.; Willighagen, E. PSnpBind: a database of mutated binding site protein-ligand complexes constructed using a multithreaded virtual screening workflow. *J. Cheminform.* **2022**, *14*, 8.
(113) Lingė, D. PLBD: protein-ligand binding database of thermodynamic and kinetic intrinsic parameters. *Database* **2023**, DOI: 10.1093/database/baad040.
(114) Wei, H.; Wang, W.; Peng, Z.; Yang, J. Q-BioLiP: A Comprehensive Resource for Quaternary Structure-based Protein−ligand Interactions. *bioRxiv*, 2023, 2023.06.23.546351.
(115) Korlepara, D. B. PLAS-20k: Extended dataset of protein-ligand affinities from MD simulations for machine learning applications. *Sci. Data* **2024**, DOI: 10.1038/s41597-023-02872-y.
(116) Xenarios, I.; Rice, D. W.; Salwinski, L.; Baron, M. K.; Marcotte, E. M.; Eisenberg, D. DIP: the database of interacting proteins. *Nucleic Acids Res.* **2000**, *28*, 289−291.
(117) Wallach, I.; Lilien, R. The protein-small-molecule database, a non-redundant structural resource for the analysis of protein-ligand binding. *Bioinformatics* **2009**, *25*, 615−620.
(118) Wang, S.; Lin, H.; Huang, Z.; He, Y.; Deng, X.; Xu, Y.; Pei, J.; Lai, L. CavitySpace: A Database of Potential Ligand Binding Sites in the Human Proteome. *Biomolecules* **2022**, *12*, 967.
(119) Otter, D. W.; Medina, J. R.; Kalita, J. K. A survey of the usages of deep learning for natural language processing. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 604−624.
(120) Wang, Y.; You, Z.-H.; Yang, S.; Li, X.; Jiang, T.-H.; Zhou, X. A high efficient biological language model for predicting Protein-Protein interactions. *Cells* **2019**, *8*, 122.
(121) Abbasi, K.; Razzaghi, P.; Poso, A.; Amanlou, M.; Ghasemi, J. B.; Masoudi-Nejad, A. DeepCDA: deep cross-domain compound−protein affinity prediction through LSTM and convolutional neural networks. *Bioinformatics* **2020**, *36*, 4633−4642.
(122) Zhou, G.; Gao, Z.; Ding, Q.; Zheng, H.; Xu, H.; Wei, Z.; Zhang, L.; Ke, G. Uni-Mol: AUniversal 3D Molecular Representation Learning Framework. *ChemRxiv*, 2023.

(123) Zhou, D.; Xu, Z.; Li, W.; Xie, X.; Peng, S. MultiDTI: drug−target interaction prediction based on multi-modal representation learning to bridge the gap between new chemical entities and known heterogeneous network. *Bioinformatics* **2021**, *37*, 4485−4492.
(124) Özçelik, R.; Öztürk, H.; Özgür, A.; Ozkirimli, E. ChemBoost: A chemical language based approach for protein - ligand binding affinity prediction. *Mol. Inform.* **2021**, *40*, e2000212.
(125) Gaspar, H. A.; Ahmed, M.; Edlich, T.; Fabian, B.; Varszegi, Z.; Segler, M.; Meyers, J.; Fiscato, M. Proteochemometric Models Using Multiple Sequence Alignments and a Subword Segmented Masked Language Model. *ChemRxiv*, 2021.
(126) Arseniev-Koehler, A. Theoretical foundations and limits of word embeddings: What types of meaning can they capture. *Sociol. Methods Res.* **2022**, No. 004912412211401.
(127) Lake, B. M.; Murphy, G. L. Word meaning in minds and machines. *Psychol. Rev.* **2023**, *130*, 401−431.
(128) Winchester, S. A Verb for Our Frantic Times. https://www.nytimes.com/2011/05/29/opinion/29winchester.html, 2011; Accessed: 2024−9-15.
(129) Panapitiya, G.; Girard, M.; Hollas, A.; Sepulveda, J.; Murugesan, V.; Wang, W.; Saldanha, E. Evaluation of deep learning architectures for aqueous solubility prediction. *ACS Omega* **2022**, *7*, 15695−15710.
(130) Wu, X.; Yu, L. EPSOL: sequence-based protein solubility prediction using multidimensional embedding. *Bioinformatics* **2021**, *37*, 4314−4320.
(131) Krogh, A. What are artificial neural networks? *Nat. Biotechnol.* **2008**, *26*, 195−197.
(132) Rumelhart, D.; Hinton, G. E.; Williams, R. J. Learning internal representations by error propagation. *cmapspublic2.ihmc.us* **1986**, 673−695.
(133) Salehinejad, H.; Sankar, S.; Barfett, J.; Colak, E.; Valaee, S. Recent Advances in Recurrent Neural Networks. *arXiv*, 2017.
(134) Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, 30.
(135) Sherstinsky, A. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *arXiv*, 2018.
(136) Chen, G. A gentle tutorial of recurrent neural network with error backpropagation. *arXiv*, 2016.
(137) Cho, K.; van Merrienboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *arXiv*, 2014.
(138) Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735−1780.
(139) Graves, A.; Schmidhuber, J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Netw.* **2005**, *18*, 602−610.
(140) Thafar, M. A.; Alshahrani, M.; Albaradei, S.; Gojobori, T.; Essack, M.; Gao, X. Affinity2Vec: drug-target binding affinity prediction through representation learning, graph mining, and machine learning. *Sci. Rep.* **2022**, *12*, 4751.
(141) Wei, B.; Zhang, Y.; Gong, X. 519. DeepLPI: A Novel Drug Repurposing Model based on Ligand-Protein Interaction Using Deep Learning. *Open Forum Infect. Dis.* **2022**, *9*, ofac492.574.
(142) Yuan, W.; Chen, G.; Chen, C. Y.-C. FusionDTA: attention-based feature polymerizer and knowledge distillation for drug-target binding affinity prediction. *Brief. Bioinform.* **2022**, DOI: 10.1093/bib/bbab506.
(143) West-Roberts, J.; Valentin-Alvarado, L.; Mullen, S.; Sachdeva, R.; Smith, J.; Hug, L. A.; Gregoire, D. S.; Liu, W.; Lin, T.-Y.; Husain, G.; Amano, Y.; Ly, L.; Banfield, J. F. Giant genes are rare but implicated in cell wall degradation by predatory bacteria. *bioRxiv*, 2023.
(144) Hernández, A.; Amigó, J. Attention mechanisms and their applications to complex systems. *Entropy (Basel)* **2021**, *23*, 283.
(145) Yang, X. An overview of the attention mechanisms in computer vision. 2020.
(146) Hu, D. An introductory survey on attention mechanisms in NLP problems. *arXiv*, 2018.
(147) Vig, J., Madani, A., Varshney, L. R., Xiong, C., Socher, R., Rajani, N. F. BERTology Meets Biology: Interpreting Attention in Protein Language Models. *arXiv*, 2020.
(148) Wu, H.; Liu, J.; Jiang, T.; Zou, Q.; Qi, S.; Cui, Z.; Tiwari, P.; Ding, Y. AttentionMGT-DTA: A multi-modal drug-target affinity prediction using graph transformer and attention mechanism. *Neural Netw.* **2024**, *169*, 623−636.
(149) Koyama, K.; Kamiya, K.; Shimada, K. Cross attention dti: Drug-target interaction prediction with cross attention module in the blind evaluation setup. *BIOKDD2020* **2020**.
(150) Kurata, H.; Tsukiyama, S. ICAN: Interpretable cross-attention network for identifying drug and target protein interactions. *PLoS One* **2022**, *17*, e0276609.
(151) Zhao, Q.; Zhao, H.; Zheng, K.; Wang, J. HyperAttentionDTI: improving drug−protein interaction prediction by sequence-based deep learning with attention mechanism. *Bioinformatics* **2022**, *38*, 655−662.
(152) Jiang, M.; Li, Z.; Zhang, S.; Wang, S.; Wang, X.; Yuan, Q. Drug-target affinity prediction using graph neural network and contact maps. *RSC Adv.* **2020**, *10*, 20701.
(153) Nguyen, T. M.; Nguyen, T.; Le, T. M.; Tran, T. GEFA: Early Fusion Approach in Drug-Target Affinity Prediction. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2022**, *19*, 718−728.
(154) Yu, J.; Li, Z.; Chen, G.; Kong, X.; Hu, J.; Wang, D.; Cao, D.; Li, Y.; Huo, R.; Wang, G.; Liu, X.; Jiang, H.; Li, X.; Luo, X.; Zheng, M. Computing the relative binding affinity of ligands based on a pairwise binding comparison network. *Nature Computational Science* **2023**, *3*, 860−872.
(155) Knutson, C.; Bontha, M.; Bilbrey, J. A.; Kumar, N. Decoding the protein−ligand interactions using parallel graph neural networks. *Sci. Rep.* **2022**, *12*, 1−14.
(156) Kyro, G. W.; Brent, R. I.; Batista, V. S. HAC-Net: A Hybrid Attention-Based Convolutional Neural Network for Highly Accurate Protein−Ligand Binding Affinity Prediction. *J. Chem. Inf. Model.* **2023**, *63*, 1947−1960.
(157) Yousefi, N.; Yazdani-Jahromi, M.; Tayebi, A.; Kolanthai, E.; Neal, C. J.; Banerjee, T.; Gosai, A.; Balasubramanian, G.; Seal, S.; Ozmen Garibay, O. BindingSite-AugmentedDTA: enabling a next-generation pipeline for interpretable prediction models in drug repurposing. *Brief. Bioinform.* **2023**, DOI: 10.1093/bib/bbad136.
(158) Yazdani-Jahromi, M.; Yousefi, N.; Tayebi, A.; Kolanthai, E.; Neal, C. J.; Seal, S.; Garibay, O. O. AttentionSiteDTI: an interpretable graph-based model for drug-target interaction prediction using NLP sentence-level relation classification. *Brief. Bioinform.* **2022**, DOI: 10.1093/bib/bbac272.
(159) Bronstein, M. M.; Bruna, J.; Cohen, T.; Veličković, P. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv*, 2021.
(160) Lim, J.; Ryu, S.; Park, K.; Choe, Y. J.; Ham, J.; Kim, W. Y. Predicting Drug−Target Interaction Using a Novel Graph Neural Network with 3D Structure-Embedded Graph Representation. *J. Chem. Inf. Model.* **2019**, *59*, 3981−3988.
(161) Jin, Z.; Wu, T.; Chen, T.; Pan, D.; Wang, X.; Xie, J.; Quan, L.; Lyu, Q. CAPLA: improved prediction of protein−ligand binding affinity by a deep learning approach based on a cross-attention mechanism. *Bioinformatics* **2023**, *39*, btad049.
(162) Lin, Z.; Akin, H.; Rao, R.; Hie, B.; Zhu, Z.; Lu, W.; Smetanin, N.; Verkuil, R.; Kabeli, O.; Shmueli, Y.; Dos Santos Costa, A.; Fazel-Zarandi, M.; Sercu, T.; Candido, S.; Rives, A. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science* **2023**, *379*, 1123−1130.
(163) Zhang, S.; Fan, R.; Liu, Y.; Chen, S.; Liu, Q.; Zeng, W. Applications of transformer-based language models in bioinformatics: a survey. *Bioinform. Adv.* **2023**, *3*, vbad001.
(164) Bahdanau, D.; Cho, K.; Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv*, 2014.

https://doi.org/10.1021/acs.jcim.4c01907
J. Chem. Inf. Model. 2025, 65, 2191−2213

(165) Vinyals, O.; Toshev, A.; Bengio, S.; Erhan, D. Show and tell: A neural image caption generator. *Pattern Recognition (CVPR)* **2015**, 3156−3164.
(166) Sutskever, I.; Vinyals, O.; Le, Q. V. Sequence to sequence learning with Neural Networks. *arXiv*, 2014;.
(167) Cho, K.; van Merrienboer, B.; Bahdanau, D.; Bengio, Y. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv*, 2014.
(168) Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv*, 2018.
(169) Zeyer, A.; Bahar, P.; Irie, K.; Schlüter, R.; Ney, H. A Comparison of Transformer and LSTM Encoder Decoder Models for ASR. *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)* **2019**, 8−15.
(170) Irie, K.; Zeyer, A.; Schlüter, R.; Ney, H. Language Modeling with Deep Transformers. *arXiv*, 2019.
(171) Zouitni, C.; Sabri, M. A.; Aarab, A. A Comparison Between LSTM and Transformers for Image Captioning. *Digital Technologies and Applications* **2023**, *669*, 492−500.
(172) Parisotto, E.; Song, F.; Rae, J.; Pascanu, R.; Gulcehre, C.; Jayakumar, S.; Jaderberg, M.; Kaufman, R. L.; Clark, A.; Noury, S.; Botvinick, M.; Heess, N.; Hadsell, R. Stabilizing Transformers for Reinforcement Learning. *Proceedings of the 37th International Conference on Machine Learning* **2020**, 7487−7498.
(173) Bilokon, P.; Qiu, Y. Transformers versus LSTMs for electronic trading. *arXiv*, 2023.
(174) Merity, S. Single Headed Attention RNN: Stop Thinking With Your Head. *arXiv*, 2019.
(175) Ezen-Can, A. A Comparison of LSTM and BERT for Small Corpus. *arXiv*, 2020.
(176) Unsal, S.; Atas, H.; Albayrak, M.; Turhan, K.; Acar, A. C.; Doğan, T. Learning functional properties of proteins with language models. *Nature Machine Intelligence* **2022**, *4*, 227−245.
(177) Brandes, N.; Ofer, D.; Peleg, Y.; Rappoport, N.; Linial, M. ProteinBERT: a universal deep-learning model of protein sequence and function. *Bioinformatics* **2022**, *38*, 2102−2110.
(178) Luo, S.; Chen, T.; Xu, Y.; Zheng, S.; Liu, T.-Y.; Wang, L.; He, D. One Transformer Can Understand Both 2D & 3D Molecular Data. *arXiv*, 2022.
(179) Clark, K.; Luong, M.-T.; Le, Q. V.; Manning, C. D. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. *arXiv*, 2020.
(180) Wang, J.; Wen, N.; Wang, C.; Zhao, L.; Cheng, L. ELECTRA-DTA: a new compound-protein binding affinity prediction model based on the contextualized sequence encoding. *J. Cheminform.* **2022**, *14*, 14.
(181) Shin, B.; Park, S.; Kang, K.; Ho, J. C. Self-Attention Based Molecule Representation for Predicting Drug-Target Interaction. *Proceedings of the 4th Machine Learning for Healthcare Conference* **2019**, 230−248.
(182) Huang, K.; Xiao, C.; Glass, L. M.; Sun, J. MolTrans: Molecular Interaction Transformer for drug−target interaction prediction. *Bioinformatics* **2021**, *37*, 830−836.
(183) Shen, L.; Feng, H.; Qiu, Y.; Wei, G.-W. SVSBI: sequence-based virtual screening of biomolecular interactions. *Commun. Biol.* **2023**, *6*, 536.
(184) Wang, J.; Hu, J.; Sun, H.; Xu, M.; Yu, Y.; Liu, Y.; Cheng, L. MGPLI: exploring multigranular representations for protein−ligand interaction prediction. *Bioinformatics* **2022**, *38*, 4859−4867.
(185) Qian, Y.; Wu, J.; Zhang, Q. CAT-CPI: Combining CNN and transformer to learn compound image features for predicting compound-protein interactions. *Front Mol. Biosci* **2022**, *9*, 963912.
(186) Cang, Z.; Mu, L.; Wei, G.-W. Representability of algebraic topology for biomolecules in machine learning based scoring and virtual screening. *PLoS Comput. Biol.* **2018**, *14*, e1005929.
(187) Chen, D.; Liu, J.; Wei, G.-W. Multiscale topology-enabled structure-to-sequence transformer for protein-ligand interaction predictions. *Nat. Mac. Intell.* **2024**, *6*, 799−810.
(188) Bubeck, S.; Chandrasekaran, V.; Eldan, R.; Gehrke, J.; Horvitz, E.; Kamar, E.; Lee, P.; Lee, Y. T.; Li, Y.; Lundberg, S.; Nori, H.; Palangi, H.; Ribeiro, M. T.; Zhang, Y. Sparks of Artificial General Intelligence: Early experiments with GPT-4. *arXiv*, 2023.
(189) Elnaggar, A.; Essam, H.; Salah-Eldin, W.; Moustafa, W.; Elkerdawy, M.; Rochereau, C.; Rost, B. Ankh: Optimized protein language model unlocks general-purpose modelling. *bioRxiv*, 2023.
(190) Hwang, Y.; Cornman, A. L.; Kellogg, E. H.; Ovchinnikov, S.; Girguis, P. R. Genomic language model predicts protein co-regulation and function. *Nat. Commun.* **2024**, *15*, 2880.
(191) Vu, M. H.; Akbar, R.; Robert, P. A.; Swiatczak, B.; Greiff, V.; Sandve, G. K.; Haug, D. T. T. Linguistically inspired roadmap for building biologically reliable protein language models. *arXiv*, 2022.
(192) Xu, M.; Zhang, Z.; Lu, J.; Zhu, Z.; Zhang, Y.; Ma, C.; Liu, R.; Tang, J. PEER: A comprehensive and multi-task benchmark for Protein sEquence undERstanding. *arXiv* 2022, 35156−35173.
(193) Schmirler, R.; Heinzinger, M.; Rost, B. Fine-tuning protein language models boosts predictions across diverse tasks. *Nat. Commun.* **2024**, *15*, 7407.
(194) Heinzinger, M.; Elnaggar, A.; Wang, Y.; Dallago, C.; Nechaev, D.; Matthes, F.; Rost, B. Modeling aspects of the language of life through transfer-learning protein sequences. *BMC Bioinformatics* **2019**, *20*, 723.
(195) Manfredi, M.; Savojardo, C.; Martelli, P. L.; Casadio, R. E-SNPs&GO: embedding of protein sequence and function improves the annotation of human pathogenic variants. *Bioinformatics* **2022**, *38*, 5168−5174.
(196) Anteghini, M.; Martins Dos Santos, V.; Saccenti, E. In-Pero: Exploiting deep learning embeddings of protein sequences to predict the localisation of peroxisomal proteins. *Int. J. Mol. Sci.* **2021**, *22*, 6409.
(197) Nguyen, T.; Le, H.; Quinn, T. P.; Nguyen, T.; Le, T. D.; Venkatesh, S. GraphDTA: predicting drug−target binding affinity with graph neural networks. *Bioinformatics* **2021**, *37*, 1140−1147.
(198) Nam, H.; Ha, J.-W.; Kim, J. Dual attention networks for multimodal reasoning and matching. *Pattern Recognition (CVPR)* **2017**, 299−307.
(199) Wang, X.; Liu, D.; Zhu, J.; Rodriguez-Paton, A.; Song, T. CSConv2d: A 2-D Structural Convolution Neural Network with a Channel and Spatial Attention Mechanism for Protein-Ligand Binding Affinity Prediction. *Biomolecules* **2021**, DOI: 10.3390/biom11050643.
(200) Anteghini, M.; Santos, V. A. M. D.; Saccenti, E. PortPred: Exploiting deep learning embeddings of amino acid sequences for the identification of transporter proteins and their substrates. *J. Cell. Biochem.* **2023**, *124*, 1803.
(201) Huang, K.; Fu, T.; Glass, L. M.; Zitnik, M.; Xiao, C.; Sun, J. DeepPurpose: a deep learning library for drug−target interaction prediction. *Bioinformatics* **2021**, *36*, 5545−5547.
(202) Zhao, L.; Wang, J.; Pang, L.; Liu, Y.; Zhang, J. GANsDTA: Predicting Drug-Target Binding Affinity Using GANs. *Front. Genet.* **2020**, *10*, 1243.
(203) Hu, F.; Jiang, J.; Wang, D.; Zhu, M.; Yin, P. Multi-PLI: interpretable multi-task deep learning model for unifying protein−ligand interaction datasets. *J. Cheminform.* **2021**, *13*, 30.
(204) Zheng, S.; Li, Y.; Chen, S.; Xu, J.; Yang, Y. Predicting Drug Protein Interaction using Quasi-Visual Question Answering System. *bioRxiv* **2019**, 588178.
(205) Tsubaki, M.; Tomii, K.; Sese, J. Compound−protein interaction prediction with end-to-end learning of neural networks for graphs and sequences. *Bioinformatics* **2019**, *35*, 309−318.
(206) Karimi, M.; Wu, D.; Wang, Z.; Shen, Y. DeepAffinity: interpretable deep learning of compound−protein affinity through unified recurrent and convolutional neural networks. *Bioinformatics* **2019**, *35*, 3329−3338.
(207) Li, S.; Wan, F.; Shu, H.; Jiang, T.; Zhao, D.; Zeng, J. MONN: a multi-objective neural network for predicting compound-protein interactions and affinities. *Cell Systems* **2020**, *10*, 308−322.
(208) Zhao, M.; Yuan, M.; Yang, Y.; Xu, S. X. CPGL: Prediction of Compound-Protein Interaction by Integrating Graph Attention

Network With Long Short-Term Memory Neural Network. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2023**, *20*, 1935−1942.
(209) Yu, L.; Qiu, W.; Lin, W.; Cheng, X.; Xiao, X.; Dai, J. HGDTI: predicting drug−target interaction by using information aggregation based on heterogeneous graph neural network. *BMC Bioinformatics* **2022**, *23*, 126.
(210) Lee, I.; Nam, H. Sequence-based prediction of protein binding regions and drug-target interactions. *J. Cheminform.* **2022**, *14*, 5.
(211) Gönen, M.; Heller, G. Concordance probability and discriminatory power in proportional hazards regression. *Biometrika* **2005**, *92*, 965−970.
(212) Deller, M. C.; Rupp, B. Models of protein-ligand crystal structures: trust, but verify. *J. Comput. Aided Mol. Des.* **2015**, *29*, 817−836.
(213) Kalakoti, Y.; Yadav, S.; Sundar, D. TransDTI: Transformer-based language models for estimating DTIs and building a drug recommendation workflow. *ACS Omega* **2022**, *7*, 2706−2717.
(214) Chatterjee, A.; Walters, R.; Shafi, Z.; Ahmed, O. S.; Sebek, M.; Gysi, D.; Yu, R.; Eliassi-Rad, T.; Barabási, A.-L.; Menichetti, G. Improving the generalizability of protein-ligand binding predictions with AI-Bind. *Nat. Commun.* **2023**, *14*, 1989.
(215) Gudivada, V.; Apon, A.; Ding, J. Data quality considerations for big data and machine learning: Going beyond data cleaning and transformations. *Int. J. Adv. Softw.* **2017**, *10*, 1−20.
(216) Nasteski, V. An overview of the supervised machine learning methods. *Horizons* **2017**, *4*, 51−62.
(217) Kozlov, M. So you got a null result. Will anyone publish it? *Nature* **2024**, *631*, 728−730.
(218) Edfeldt, K.; et al. A data science roadmap for open science organizations engaged in early-stage drug discovery. *Nat. Commun.* **2024**, *15*, 5640.
(219) Mlinarić, A.; Horvat, M.; Šupak Smolčić, V. Dealing with the positive publication bias: Why you should really publish your negative results. *Biochem. Med.* **2017**, *27*, 030201.
(220) Fanelli, D. Negative results are disappearing from most disciplines and countries. *Scientometrics* **2012**, *90*, 891−904.
(221) Albalate, A.; Minker, W. *Semi-supervised and unervised machine learning: Novel strategies*; Wiley-ISTE, 2013.
(222) Sajadi, S. Z.; Zare Chahooki, M. A.; Gharaghani, S.; Abbasi, K. AutoDTI++: deep unsupervised learning for DTI prediction by autoencoders. *BMC Bioinformatics* **2021**, *22*, 204.
(223) Najm, M.; Azencott, C.-A.; Playe, B.; Stoven, V. Drug Target Identification with Machine Learning: How to Choose Negative Examples. *Int. J. Mol. Sci.* **2021**, *22*, 5118.
(224) Sieg, J.; Flachsenberg, F.; Rarey, M. In Need of Bias Control: Evaluating Chemical Data for Machine Learning in Structure-Based Virtual Screening. *J. Chem. Inf. Model.* **2019**, *59*, 947−961.
(225) Volkov, M.; Turk, J.-A.; Drizard, N.; Martin, N.; Hoffmann, B.; Gaston-Mathé, Y.; Rognan, D. On the Frustration to Predict Binding Affinities from Protein−Ligand Structures with Deep Neural Networks. *J. Med. Chem.* **2022**, *65*, 7946−7958.
(226) Shivakumar, D.; Williams, J.; Wu, Y.; Damm, W.; Shelley, J.; Sherman, W. Prediction of absolute solvation free energies using molecular dynamics free energy perturbation and the OPLS force field. *J. Chem. Theory Comput.* **2010**, *6*, 1509−1519.
(227) El Hage, K.; Mondal, P.; Meuwly, M. Free energy simulations for protein ligand binding and stability. *Mol. Simul.* **2018**, *44*, 1044−1061.
(228) Ngo, S. T.; Pham, M. Q. Umbrella sampling-based method to compute ligand-binding affinity. *Methods Mol. Biol.* **2022**, *2385*, 313−323.
(229) Pandey, M.; Fernandez, M.; Gentile, F.; Isayev, O.; Tropsha, A.; Stern, A. C.; Cherkasov, A. The transformational role of GPU computing and deep learning in drug discovery. *Nat. Mach. Intell.* **2022**, *4*, 211−221.
(230) Bibal, A.; Cardon, R.; Alfter, D.; Wilkens, R.; Wang, X.; François, T.; Watrin, P. Is Attention Explanation? An Introduction to the Debate. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Dublin, Ireland,* **2022**, 3889−3900.
(231) Wiegreffe, S.; Pinter, Y. Attention is not not Explanation. *arXiv*, 2019.
(232) Jain, S.; Wallace, B. C. Attention is not Explanation. *arXiv*, 2019.
(233) Lundberg, S. M.; Lee, S.-I. A unified approach to interpreting model predictions. *Neural Inf. Process. Syst.* **2017**, *30*, 4765−4774.
(234) Gu, Y.; Zhang, X.; Xu, A.; Chen, W.; Liu, K.; Wu, L.; Mo, S.; Hu, Y.; Liu, M.; Luo, Q. Protein-ligand binding affinity prediction with edge awareness and supervised attention. *iScience* **2023**, *26*, 105892.
(235) Rodis, N.; Sardianos, C.; Papadopoulos, G. T.; Radoglou-Grammatikis, P.; Sarigiannidis, P.; Varlamis, I. Multimodal Explainable Artificial Intelligence: A Comprehensive Review of Methodological Advances and Future Research Directions. *arXiv [cs.AI]* 2023.
(236) Gilpin, L. H.; Bau, D.; Yuan, B. Z.; Bajwa, A.; Specter, M.; Kagal, L. Explaining explanations: An overview of interpretability of machine learning. *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)* **2018**, 80−89.
(237) Luo, D.; Liu, D.; Qu, X.; Dong, L.; Wang, B. Enhancing generalizability in protein-ligand binding affinity prediction with multimodal contrastive learning. *J. Chem. Inf. Model.* **2024**, *64*, 1892−1906.
(238) Rao, R.; Bhattacharya, N.; Thomas, N.; Duan, Y.; Chen, X.; Canny, J.; Abbeel, P.; Song, Y. S. Evaluating protein transfer learning with TAPE. *bioRxiv*, 2019.
(239) Suzek, B. E.; Wang, Y.; Huang, H.; McGarvey, P. B.; Wu, C. H. UniProt Consortium UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics* **2015**, *31*, 926−932.
(240) Eguida, M.; Rognan, D. A Computer Vision Approach to Align and Compare Protein Cavities: Application to Fragment-Based Drug Design. *J. Med. Chem.* **2020**, *63*, 7127−7142.
(241) Öztürk, H.; Özgür, A.; Ozkirimli, E. DeepDTA: deep drug-target binding affinity prediction. *Bioinformatics* **2018**, *34*, i821−i829.
(242) Evans, R. et al. Protein complex prediction with AlphaFold-Multimer. *bioRxiv*, 2021.
(243) Omidi, A.; Møller, M. H.; Malhis, N.; Bui, J. M.; Gsponer, J. AlphaFold-Multimer accurately captures interactions and dynamics of intrinsically disordered protein regions. *Proc. Natl. Acad. Sci. U. S. A.* **2024**, *121*, e2406407121.
(244) Zhu, W.; Shenoy, A.; Kundrotas, P.; Elofsson, A. Evaluation of AlphaFold-Multimer prediction on multi-chain protein complexes. *Bioinformatics* **2023**, *39*, btad424.
(245) Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; Ommer, B. High-resolution image synthesis with latent diffusion models. *Pattern Recognition (CVPR)* **2022**, 10684−10695.
(246) Dhariwal, P.; Nichol, A. Diffusion models beat GANs on image synthesis. *Neural Inf. Process. Syst.* **2021**, 8780−8794.
(247) Karras, T.; Aittala, M.; Aila, T.; Laine, S. Elucidating the Design Space of Diffusion-Based Generative Models. *Adv. Neural Inf. Process. Syst.* **2022**, 26565−26577.
(248) Buttenschoen, M.; Morris, G.; Deane, C. PoseBusters: AI-based docking methods fail to generate physically valid poses or generalise to novel sequences. *Chem. Sci.* **2024**, *15*, 3130−3139.
(249) Wee, J.; Wei, G.-W. Benchmarking AlphaFold3's protein-protein complex accuracy and machine learning prediction reliability for binding free energy changes upon mutation. *arXiv*, 2024.
(250) Bernard, C.; Postic, G.; Ghannay, S.; Tahi, F. Has AlphaFold 3 reached its success for RNAs? *bioRxiv*, 2024.
(251) Zonta, F.; Pantano, S. From sequence to mechanobiology? Promises and challenges for AlphaFold 3. *Mechanobiology in Medicine* **2024**, *2*, 100083.
(252) He, X.-H.; Li, J.-R.; Shen, S.-Y.; Xu, H. E. AlphaFold3 versus experimental structures: assessment of the accuracy in ligand-bound G protein-coupled receptors. *Acta Pharmacol. Sin.* **2024**, 1−12.

https://doi.org/10.1021/acs.jcim.4c01907
J. Chem. Inf. Model. 2025, 65, 2191−2213

(253) Desai, D.; Kantliwala, S. V.; Vybhavi, J.; Ravi, R.; Patel, H.; Patel, J. Review of AlphaFold 3: Transformative advances in drug design and therapeutics. *Cureus* **2024**, *16*, e63646.
(254) Baek, M.; et al. Accurate prediction of protein structures and interactions using a three-track neural network. *Science* **2021**, *373*, 871−876.
(255) Ahdritz, G.; et al. OpenFold: retraining AlphaFold2 yields new insights into its learning mechanisms and capacity for generalization. *Nat. Methods* **2024**, *21*, 1514−1524.
(256) Liao, C.; Yu, Y.; Mei, Y.; Wei, Y. From words to molecules: A survey of Large Language Models in chemistry. *arXiv*, 2024.
(257) Bagal, V.; Aggarwal, R.; Vinod, P. K.; Priyakumar, U. D. MolGPT: Molecular generation using a transformer-decoder model. *J. Chem. Inf. Model.* **2022**, *62*, 2064−2076.
(258) Janakarajan, N.; Erdmann, T.; Swaminathan, S.; Laino, T.; Born, J. Language models in molecular discovery. *arXiv*, 2023.
(259) Park, Y.; Metzger, B. P. H.; Thornton, J. W. The simplicity of protein sequence-function relationships. *Nat. Commun.* **2024**, *15*, 7953.
(260) Stahl, K.; Warneke, R.; Demann, L.; Bremenkamp, R.; Hormes, B.; Brock, O.; Stülke, J.; Rappsilber, J. Modelling protein complexes with crosslinking mass spectrometry and deep learning. *Nat. Commun.* **2024**, *15*, 7866.
(261) Senior, A. W.; et al. Improved protein structure prediction using potentials from deep learning. *Nature* **2020**, *577*, 706−710.
(262) Kramer, M. A. Nonlinear principal component analysis using autoassociative neural networks. *AIChE J.* **1991**, *37*, 233−243.
(263) Rogers, D.; Hahn, M. Extended-connectivity fingerprints. *J. Chem. Inf. Model.* **2010**, *50*, 742−754.
(264) Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph Attention Networks. *arXiv*, 2017.
(265) Kipf, T. N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv*, 2016.
(266) Xu, Z.; Wang, S.; Zhu, F.; Huang, J. Seq2seq Fingerprint: An Unsupervised Deep Molecular Embedding for Drug Discovery. *Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics. New York, NY, USA* **2017**, 285−294.
(267) Gilmer, J.; Schoenholz, S.; Riley, P. F.; Vinyals, O.; Dahl, G. E. Neural Message Passing for Quantum Chemistry. *ICML* **2017**, 1263−1272.
(268) Asgari, E.; Mofrad, M. R. K. Continuous Distributed Representation of Biological Sequences for Deep Proteomics and Genomics. *PLoS One* **2015**, *10*, e0141287.
(269) He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.* **2015**, 770−778.
(270) Öztürk, H.; Ozkirimli, E.; Özgür, A. A novel methodology on distributed representations of proteins using their interacting ligands. *Bioinformatics* **2018**, *34*, i295−i303.
(271) Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. *Proc. IEEE Conf. Comput. Vis. Pattern Recognition* **2018**, 7132−7141.

https://doi.org/10.1021/acs.jcim.4c01907
J. Chem. Inf. Model. 2025, 65, 2191−2213