PRELIMINARY PREPRINT VERSION – DO NOT CITE
This is not the version of record. The final version will appear in the Proceedings of IJCAI-ECAI 2026.

EnzyPGM: Pocket-conditioned Generative Model for Substrate-specific Enzyme Design

Zefeng Lin[1] , **Zhihang Zhang**[4] , **Weirong Zhu**[1] , **Tongchang Han**[1]
Xianyong Fang[3*] , **Tianfan Fu**[2*] , **Xiaohua Xu**[1*]
[1]School of Computer Science and Technology, University of Science and Technology of China, Hefei, Anhui, China
[2]State Key Laboratory for Novel Software Technology at Nanjing University, School of Computer Science, Nanjing University, Nanjing, Jiangsu, China
[3]School of Computer Science and Technology, Anhui University, Hefei, Anhui, China
[4]School of Airspace Science and Engineering, Shandong University, Jinan, Shandong, China
{zflin, zwr211355,tchan,xiaohuaxu}@mail.ustc.edu.cn, futianfan@nju.edu.cn, fangxianyong@ahu.edu.cn, 202200800170@mail.sdu.edu.cn

IJCAI-ECAI 2026 PREPRINT

Abstract

Designing enzymes with substrate-binding pockets is a critical challenge in protein engineering, as catalytic activity depends on the precise interaction between pockets and substrates. Currently, generative models dominate functional protein design but cannot model pocket-substrate interactions, which limits enzyme generation with precise catalytic environments. To address this issue, we propose **EnzyPGM**, a unified framework that jointly generates enzymes and substrate-binding pockets conditioned on functional priors and substrates, with a particular focus on learning accurate pocket–substrate interactions. At its core, EnzyPGM includes two main modules: a Residue-atom Bi-scale Attention (RBA) that jointly models intra-residue dependencies and fine-grained interactions between pocket residues and substrate atoms, and a Residue Function Fusion (RFF) that incorporates enzyme function priors into residue representations. Also, we curate EnzyPock, an enzyme–pocket dataset comprising 84,336 enzyme–substrate pairs across 1,036 four-level enzyme families. Extensive experiments demonstrate that EnzyPGM achieves state-of-the-art performance on EnzyPock. Notably, EnzyPGM reduces the average binding energy by 0.47 kcal/mol over EnzyGen, showing its superior performance on substrate-specific enzyme design. The code is available at https://github.com/John-Lin98/EnzyPGM.

1 Introduction

Enzymes are nature's most efficient biocatalysts, accelerating chemical reactions by 10^3–10^{17} fold [Bar-Even *et al.*, 2011] while maintaining unique substrate specificity [Huang *et al.*, 2021]. The catalytic activity of enzymes arises from precisely folded 3D structures that form confined spatial regions, called substrate-binding pockets [Stank *et al.*, 2016], where specific substrates are recognized, bound, and rapidly transformed into products. Recently, deep learning methods have demonstrated impressive potential for functional protein design [Chu *et al.*, 2024; Kortemme, 2024; Notin *et al.*, 2024], where these methods can be classified as: (1) general functional protein design guided by functional labels and partial protein information [Baek *et al.*, 2021; Nijkamp *et al.*, 2023]; (2) ligand-binding protein design relies on the geometric structure of protein-ligand complexes [Watson *et al.*, 2023; Dauparas *et al.*, 2025]; and (3) substrate-specific enzyme design based on the substrate properties and enzyme function [Du *et al.*, 2025; Wang *et al.*, 2025].

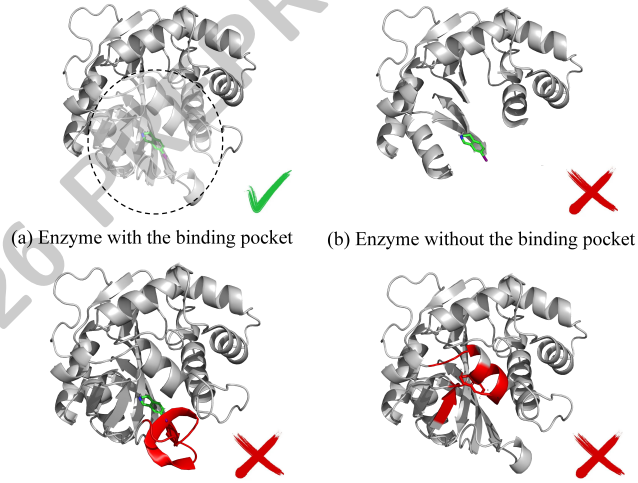


Figure 1: Four cases of enzyme design. None of the enzymes in (b), (c), and (d) bind to the substrate because there are no pockets in (b), and atom clashes in (c) and (d), which are represented by red.

Challenges. However, joint modeling of enzymes and

*Corresponding authors.

PRELIMINARY PREPRINT VERSION – DO NOT CITE
This is not the version of record. The final version will appear in the Proceedings of IJCAI-ECAI 2026.

substrate-binding pockets remains rarely studied as it faces several key challenges: (1) How to learn an effective representation of substrate-binding pockets to capture pocket-substrate interactions? These interactions affect the binding configuration and geometric structure of the enzyme and substrate [Stärk *et al.*, 2022]. Figure 1(b) shows the case without the pocket-substrate interaction modeling, where the predicted enzyme does not have a binding pocket and cannot bind to the substrate. (2) How to accurately model the residues and substrate in the pocket? The minor prediction deviations on residue and substrate significantly reduce catalytic efficiency [Lesk, 2001]. Figure 1(c) and (d) show the cases without accurately modeling the residue and substrate, where the predicted pocket and substrate have an atom clash. (3) A lack of public datasets that contain enzyme and substrate binding pockets.

Solution. To tackle these challenges, in this paper, we propose the **Pocket-conditioned Enzyme Generative Model (EnzyPGM)**, which explicitly models the substrate-binding pocket to design substrate-specific enzymes. Our key motivation is to model pocket-substrate interactions by bi-scale representation learning, thereby dynamically adjusting pocket sites in the enzyme to form a valid pocket. Furthermore, given that enzymes have obvious functional categories and functionally conserved sites, we integrate them into the language model for enzyme design. Specifically, we propose the Residue-atom Bi-scale Attention (RBA) module to accurately model the interaction between pocket residues and substrate atoms. It mainly consists of intra-residue and residue-atom attention mechanisms, which individually capture neighborhood correlations among residues, and the interaction between pocket residues and substrate atoms, thus facilitating bi-scale representation learning across pockets and substrates. To fuse the enzyme function and structure into the residue representation, we employ the Residue Function Fusion (RFF) module. It integrates Enzyme Commission (EC) number as the function embedding and updates residue representations and 3D coordinates based on their spatial neighborhoods. Also, to solve the lack of public datasets in enzyme-pocket modeling, we curate EnzyPock, the first enzyme–pocket dataset comprising 84,336 enzyme–substrate pairs across 17,404 PDB entries.

Main contributions are summarized as: (1) We propose EnzyPGM, the first model to generate enzyme and substrate-binding pockets conditioned on specific substrates jointly. (2) We design a bi-scale attention mechanism and fuse enzyme function priors, which accurately captures the interaction across residues and substrate atoms to generate a binding pocket. (3) We curate EnzyPock, a refined enzyme-pocket dataset to tackle the lack of datasets for enzyme-pocket jointly modeling. Experimental results show that EnzyPGM consistently outperforms existing methods across EnzyPock, generating enzymes with higher catalytic activity, structural confidence, and substrate selectivity.

2 Related Work

Deep Learning for Protein Design. Recent protein design methods mainly cover inverse folding, *de novo* sequence generation, and joint sequence-structure design. Inverse folding models such as ProteinMPNN [Dauparas *et al.*, 2022] and ESM-IF [Hsu *et al.*, 2022] recover sequences for given backbones. De novo sequence models such as ProtGPT2 [Ferruz *et al.*, 2022], DPLM [Wang *et al.*, 2024], and DiMA [Meshchaninov *et al.*, 2025] learn protein sequence distributions, while joint models such as EGNN [Satorras *et al.*, 2021] and ESM3 [Hayes *et al.*, 2025] models both sequence and structure. However, these methods do not explicitly model enzyme catalytic functions and substrate-specific binding pockets.

Ligand-binding Pocket Generation. Owing to the importance of the ligand-pocket binding in drug design, several studies propose ligand-aware pocket generation frameworks. PocketGen [Zhang *et al.*, 2024a] and PocketFlow [Zhang *et al.*, 2024b] explore ligand-binding pocket generation by integrating molecular priors. These methods usually require a non-pocket scaffold and focus on local pocket redesign. In contrast, EnzyPGM jointly generates enzymes and substrate-binding pockets from conserved-site and substrate conditions, thus their task settings are essentially different.

Substrate-specific Enzyme Generation. EnzyGen [Song *et al.*, 2024], GENzyme [Hua *et al.*, 2024], and EnzyControl [Song *et al.*, 2025] introduce functional, reaction, or substrate-specific control into enzyme generation. Compared with these methods, EnzyPGM explicitly models residue-atom pocket-substrate interactions in a unified enzyme generation framework for substrate-specific enzyme generation.

3 EnzyPGM: Pocket-conditioned Enzyme Generative Model

3.1 Substrate-specific Enzyme Design

This work aims to design substrate-specific enzymes with the binding pocket by modeling pocket-substrate interactions and fusing enzyme function priors.

An N-residue enzyme $\mathcal{E}$ is represented as a sequence of residues $\mathcal{E} = \{r_1, r_2, \ldots, r_N\}$, where each residue $r_i = (s_i, x_i)$, with $s_i \in \mathcal{A}$ denoting the amino acid type, and $x_i \in \mathbb{R}^3$ being its 3D coordinate of the C_α atom. Here, $\mathcal{A}$ represents the 20-amino-acid alphabet. A substrate molecule $\mathcal{V}$ consisting of M atoms is denoted as $\mathcal{V} = \{v_1, v_2, \ldots, v_M\}$, where each atom $v_j = (a_j, y_j)$, with $a_j \in \mathbb{R}^5$ representing the five atom-level chemical feature composed of element type, aromaticity, connectivity, hydrogen atom number, and hybridization type, and $y_j \in \mathbb{R}^3$ being its 3D coordinate.

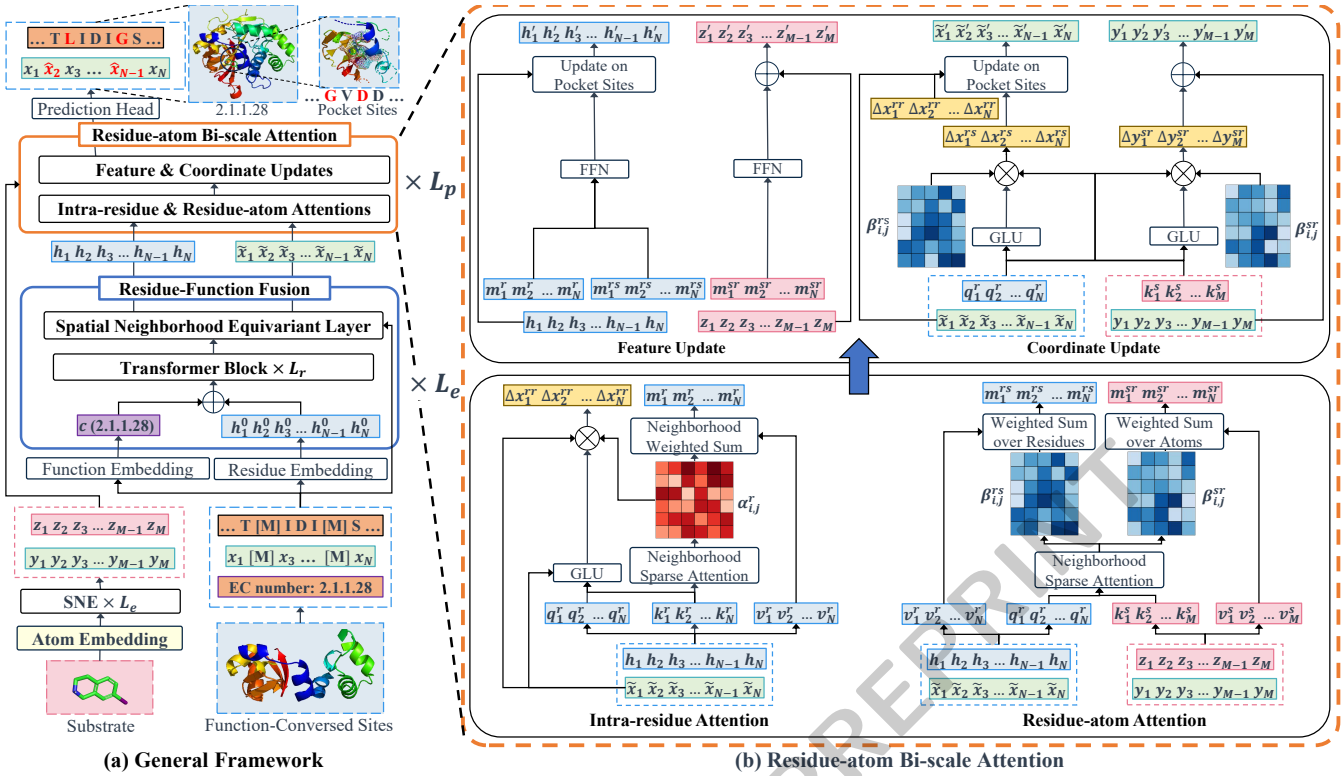
Given an enzyme $\mathcal{E}$ and substrate $\mathcal{V}$, the substrate-binding pocket $\mathcal{P}$ is the residue set spatially proximal to the substrate:

$$\mathcal{P} = \{r_i \in \mathcal{E} \mid \min_{v_j \in \mathcal{V}} \|x_i - y_j\|_2 < d\}, \tag{1}$$

where $\|\cdot\|_2$ denotes the L2 norm, and d is a distance threshold (in Å) and set to 10 in our practice.

Enzyme Commission (EC) numbers are a standard for enzymes based on catalytic reaction types, where the fourth-level classification precisely defines functional families and provides a basis for inferring enzyme functions via homology [Tipton and Boyce, 2000]. Functionally conserved sites of enzymes are highly conserved across evolution, as these residues directly determine catalytic activity and structural stability, serving as key signatures of enzymes' functions and

IJCAI-ECAI 2026 PREPRINT



(a) General Framework (b) Residue-atom Bi-scale Attention

Figure 2: Overview of EnzyPGM. (a) EnzyPGM takes enzyme function-conversed sites as a masked input sequence, conditioned on EC number and substrate atom, and predicts the complete enzyme and pocket sites via RFF and RBA modules. The mask sites at the input are represented by [M] both in sequence and coordinate space, their prediction results are indicated in **red** at the output. (b) present the specific process of the intra-residue and residue-atom attention, the feature and coordinate updates.

structures [Capra and Singh, 2007]. To fully utilize these functional priors, we obtain conserved sites $\mathcal{M}$ under EC label c as part of the input using multiple sequence alignment (MSA), and then predict other masked sites using a model.

In summary, our study problem can be formulated as a conditional masked language modeling (MLM) task: given a EC label c, a given substrate $\mathcal{V}$, a functionally conserved site index set $\mathcal{M}$, its residues set $\mathcal{E}_{\mathcal{M}}$, generate a complete enzyme $\hat{\mathcal{E}}$ and its substrate-binding pocket $\hat{\mathcal{P}}$ that can bind the substrate. Essentially, we need to train a generative model to learn the probability $P(\hat{\mathcal{E}}, \hat{\mathcal{P}} \mid \mathcal{E}_{\mathcal{M}}, \mathcal{V}, c)$.

3.2 Overall Framework

EnzyPGM is proposed for generating enzyme and substrate-binding pockets conditioned on both enzyme functional class and functionally conserved sites (Figure 2 (a)). It mainly consists of two modules: the **Residue Function Fusion (RFF)** and the **Residue-atom Bi-scale Attention (RBA)**.

At a high level, RFF aims to fuse residue-level contextual information from the enzyme sequence with functional priors in EC number and maintain their SE(3)-equivariance, while RBA focuses on enhancing pocket residues in both feature and coordinate spaces selectively.

Next, we will detail the components of EnzyPGM.

3.3 Residue–Function Fusion (RFF)

RFF is stacked Transformer [Vaswani *et al.*, 2017] blocks with a function embedding and Spatial Neighborhood Equivariant (SNE) layer. The former embeds the EC number into residue embeddings to obtain function-aware residue embeddings, while the latter performs message passing within the residue neighborhood graphs to ensure SE(3)-equivariance of residue representations and coordinates. We also use the SNE layer to model the substrate atom and coordinates.

Given $\mathcal{E}$ and $\mathcal{M}$, each s_i in $\mathcal{E}$ are first embedded into a residue $\mathbf{h}_i^{(1)} \in \mathbb{R}^{d_h}$:

$$\mathbf{h}_i^{(1)} = \begin{cases} \mathrm{Emb}(s_i), & i \in \mathcal{M}, \\ \mathrm{Emb}([\mathrm{mask}]), & \text{otherwise}, \end{cases} \tag{2}$$

where [mask] is the special token to mask the sites we need to predict, and d_h denotes the dimensions of the representation.

For introducing the enzyme family priors, we encode the EC number c into a function embedding $\mathrm{Emb}(c) \in \mathbb{R}^{d_h}$ using a learnable embedding table and add it to the residue embedding. Then, we apply L_r Transformer blocks with function embedding to model long-range residue dependencies. In the t-th transformer block, a multi-head self-attention (MHA) and fully connected feedforward network (FFN) are performed to

IJCAI-ECAI 2026 PREPRINT

obtain the function-aware residue representation $\mathbf{h}_i^{(t+1)}$:

$$\tilde{\mathbf{h}}_i^{(t)} = \mathbf{h}_i^{(t)} + \text{Emb}(c), \tag{3}$$

$$\mathbf{h}_i^{(t+0.5)} = \tilde{\mathbf{h}}_i^{(t)} + \text{MHA}\Big(\text{LN}(\tilde{\mathbf{h}}_i^{(t)}), \text{LN}(\tilde{\mathbf{H}}^{(t)})\Big), \tag{4}$$

$$\mathbf{h}_i^{(t+1)} = \mathbf{h}_i^{(t+0.5)} + \text{FFN}\Big(\text{LN}(\mathbf{h}_i^{(t+0.5)})\Big). \tag{5}$$

Here $\text{LN}(\cdot)$ denotes the layer normalization (LN) operator, Pre-LN instead of Post-LN [Xiong *et al.*, 2020] is adopted, and $\tilde{\mathbf{H}}^{(t)} = [\tilde{\mathbf{h}}_1^{(t)}, \ldots, \tilde{\mathbf{h}}_N^{(t)}]$.

The output $\{\mathbf{h}_i^{(L_r+1)}\}_{i=1}^N$ of the last Transformer block and the residue coordinates are then input to the SNE layer to ensure their SE(3)-equivariance:

$$\mathbf{m}_{ik} = \text{FFN}\Big(\mathbf{h}_i^{(L_r+1)}; \mathbf{h}_k^{(L_r+1)}; \|x_i - x_k\|_2^2\Big), \tag{6}$$

$$\tilde{\mathbf{h}}_i = \text{FFN}\left(\mathbf{h}_i^{(L_r+1)}; \sum_{k\in\mathcal{N}(i)} \mathbf{m}_{ik}\right), \tag{7}$$

$$\tilde{x}_i = x_i + \frac{1}{|\mathcal{N}(i)|} \sum_{k\in\mathcal{N}(i)} (x_i - x_k) \cdot \text{FFN}(\mathbf{m}_{ik}). \tag{8}$$

Here, $\mathbf{m}_{ik}$ denotes messages exchanged between residues, $\mathcal{N}(i)$ is the local neighboring nodes set of residue r_i within a predefined radius, $|\mathcal{N}(i)|$ represents the number of the set, and [;] denotes concatenation of vectors. Finally, the SNE layer outputs refined residue representations $\{\tilde{\mathbf{h}}_i\}_{i=1}^N$ with updated coordinates $\{\tilde{x}_i\}_{i=1}^N$, which are passed to the RBA module.

Similarly, we stack L_e of SNE layers to learn the substrate atom representation $\{\mathbf{z}_j\}_{j=1}^M$ from the embedding of the j-th substrate atom chemical feature $\mathbf{z}_j^{(1)} = \text{Emb}(a_j)$, but we do not update the atom coordinates.

3.4 Residue-atom Bi-scale Attention (RBA)

RBA aims to enhance pocket residues in both feature and coordinate spaces by jointly modeling (I) residue–residue and (II) residue–substrate interactions (Figure 2(b)).

(I) Intra-Residue Attention. To capture global message exchange within the enzyme, we perform the intra-residue attention to integrate intra-residue message using a radial basis function ϕ_{rbf} based on the refined residue representations $\{\tilde{\mathbf{h}}_i\}_{i=1}^N$ and coordinates $\{\tilde{x}_i\}_{i=1}^N$. Specifically, the attention weight α_{ik}^r between the residue r_i and r_k is calculated as:

$$\alpha_{ik}^r = \text{softmax}\left(\frac{(\mathbf{q}_i^r)^\top \mathbf{k}_k^r}{\sqrt{d}} + \mathbf{W}_r^\top \phi_{\text{rbf}}(\|\tilde{x}_i - \tilde{x}_k\|_2)\right), \tag{9}$$

where $\mathbf{q}_i^r = \mathbf{W}_Q^r \cdot \text{LN}(\tilde{\mathbf{h}}_i)$, $\mathbf{k}_k^r = \mathbf{W}_K^r \cdot \text{LN}(\tilde{\mathbf{h}}_k)$, and $\mathbf{v}_k^r = \mathbf{W}_V^r \cdot \text{LN}(\tilde{\mathbf{h}}_k)$ are the query, key, and value projections of residue representation, and $\mathbf{W}_{[\cdot]}$ is a linear projection weight.

The intra-residue message $\mathbf{m}_i^r$ from neighboring residues and the coordinate update amount Δx_i^{rr} for residue r_i are then computed as:

$$\mathbf{m}_i^r = \sum_{k\in\mathcal{N}(i)} \alpha_{ik}^r \mathbf{v}_k^r, \tag{10}$$

$$g_{ik}^r = \text{GLU}\left(\mathbf{q}_i^r; \mathbf{k}_k^r; \phi_{\text{rbf}}\left(\|\tilde{x}_i - \tilde{x}_k\|_2\right)\right), \tag{11}$$

$$\Delta x_i^{rr} = \sum_{k\in\mathcal{N}(i)} \alpha_{ik}^r \cdot g_{ik}^r \cdot \frac{\tilde{x}_i - \tilde{x}_k}{\|\tilde{x}_i - \tilde{x}_k\|_2}. \tag{12}$$

Here g_{ik}^r is the learnable gate weight controlling the magnitude of coordinate update between residue r_i and r_k, computed by a Gated Linear Unit (GLU) [Dauphin *et al.*, 2017].

(II) Residue–Atom Attention. To model the interaction between residues and substrate atoms, we perform cross-modal attention between them. The attention weights β_{ij}^{rs} between the residue r_i and the substrate atom v_j are computed as:

$$\beta_{ij}^{rs} = \text{softmax}_i\left(\frac{(\mathbf{q}_i^r)^\top \mathbf{k}_j^s}{\sqrt{d}} + \mathbf{W}_s^\top \phi_{\text{rbf}}(\|\tilde{x}_i - y_j\|_2)\right), \tag{13}$$

where $\mathbf{k}_j^s = \mathbf{W}_K^s \cdot \text{LN}(\mathbf{z}_j)$, $\mathbf{v}_j^s = \mathbf{W}_V^s \cdot \text{LN}(\mathbf{z}_j)$, softmax_i means normalizing over residues.

We perform the softmax_j to the same attention score to obtain the attention weight β_{ij}^{sr} over atoms. The residue message $\mathbf{m}_i^{rs}$ and substrate atom message $\mathbf{m}_j^{sr}$ is then aggregated as:

$$\mathbf{m}_i^{rs} = \sum_{j\in\mathcal{N}(i)} \beta_{ij}^{rs} \mathbf{v}_j^s, \tag{14}$$

$$\mathbf{m}_j^{sr} = \sum_{i\in\mathcal{N}(j)} \beta_{ij}^{sr} \mathbf{v}_j^r. \tag{15}$$

Feature and Coordinate Updates. Next, we update the representations and coordinates of pocket residues and substrate atoms based on the obtained messages and attention weights. To avoid disturbing the global enzyme scaffold, coordinate updates are sparsely applied only to non-conserved residues within the predicted pocket region $\tilde{\mathcal{P}} = \{i \mid \min_{v_j\in\mathcal{V}} \|\tilde{x}_i - y_j\|_2 < d\}$ and all atoms of the substrate. This local refinement improves pocket geometry while preserving structural continuity. Specifically, we integrate intra-residue and residue–substrate messages to obtain the updated representations $\mathbf{h}_i'$ and $\mathbf{z}_i'$:

$$\mathbf{h}_i' = \begin{cases} \mathbf{h}_i + \text{FFN}\left(m_i^r; m_i^{rs}\right), & i \in \overline{\mathcal{M}} \cap \tilde{\mathcal{P}}, \\ \mathbf{h}_i, & \text{otherwise}, \end{cases} \tag{16}$$

$$\mathbf{z}_i' = \mathbf{z}_i + \text{FFN}\left(m_i^{sr}\right). \tag{17}$$

Here $\overline{\mathcal{M}}$ is the non-functionally conserved sites.

Then, we use two GLUs to respectively compute the gate weight g_{ij}^{rs} and g_{ij}^{sr} of residues and atoms based on Equation 11 and $\mathbf{q}_i^r, \mathbf{k}_j^s, \tilde{x}_i, y_j$. The coordinate update amount Δx_i^{sr} for residue r_i and Δy_j^{rs} for atom v_j are computed as:

$$\Delta x_i^{rs} = \sum_{j\in\mathcal{N}(i)} \beta_{ij}^{rs} \cdot g_{ij}^{rs} \cdot \frac{y_j - \tilde{x}_i}{\|\tilde{x}_i - y_j\|_2}, \tag{18}$$

$$\Delta y_j^{sr} = \sum_{i\in\mathcal{N}(j)} \beta_{ij}^{sr} \cdot g_{ij}^{sr} \cdot \frac{\tilde{x}_i - y_j}{\|\tilde{x}_i - y_j\|_2}. \tag{19}$$

IJCAI-ECAI 2026 PREPRINT

Finally, the coordinates of residues and atoms are updated as follows:

$$\tilde{x}_i' = \begin{cases} \tilde{x}_i + \Delta x_i^{rr} + \Delta x_i^{rs}, & i \in \overline{\mathcal{M}} \cap \tilde{\mathcal{P}}, \\ \tilde{x}_i, & \text{otherwise}, \end{cases} \tag{20}$$

$$y_j' = y_j + \Delta y_j^{sr}. \tag{21}$$

We stack L_p of RBA modules to obtain the final representation $\hat{\mathbf{h}}_i$ and coordinates $\hat{x}_i$ and $\hat{y}_j$, a prediction head is then employed to predict amino acid type:

$$\hat{s}_i = \text{softmax}\Big(\mathbf{W}_{\text{aa}}\hat{\mathbf{h}}_i + \mathbf{b}_{\text{aa}}\Big), \tag{22}$$

where $\mathbf{W}_{\text{aa}} \in \mathbb{R}^{|\mathcal{A}|\times h_d}$ and $\mathbf{b}_{\text{aa}} \in \mathbb{R}^{|\mathcal{A}|}$ are trainable parameters, and $|\mathcal{A}| = 20$ denotes the amino acid vocabulary size.

3.5 Training Objective

EnzyPGM is trained jointly using sequence reconstruction and structural refinement. The training loss $\mathcal{L}$ considers the joint optimization objective, including the enzyme sequence loss $\mathcal{L}_{\text{s}}$, pocket loss $\mathcal{L}_{\text{ps}}$, enzyme coordinates loss $\mathcal{L}_{\text{c}}$, pocket coordinates loss $\mathcal{L}_{\text{pc}}$, and substrate coordinates loss $\mathcal{L}_{\text{sub}}$:

$$\mathcal{L}_{\text{s}} = \frac{1}{|\overline{\mathcal{M}}|}\sum_{i\in\overline{\mathcal{M}}} l_{\text{CE}}(s_i^{\text{gt}}, \hat{s}_i), \tag{23}$$

$$\mathcal{L}_{\text{ps}} = \frac{1}{|\mathcal{P}|}\sum_{i\in\mathcal{P}} l_{\text{CE}}(s_i^{\text{gt}}, \hat{s}_i), \tag{24}$$

$$\mathcal{L}_{\text{c}} = \frac{1}{|\overline{\mathcal{M}}|}\sum_{i\in\overline{\mathcal{M}}} l_{\text{Huber}}(x_i^{\text{gt}}, \hat{x}_i), \tag{25}$$

$$\mathcal{L}_{\text{pc}} = \frac{1}{|\mathcal{P}|}\sum_{i\in\mathcal{P}} l_{\text{Huber}}(x_i^{\text{gt}}, \hat{x}_i), \tag{26}$$

$$\mathcal{L}_{\text{sub}} = \frac{1}{|M|}\sum_{j} l_{\text{Huber}}(y_j^{\text{gt}}, \hat{y}_j). \tag{27}$$

Here $\mathcal{P}$ is the ground-truth of pocket residue sites. s_i^{gt} denotes the ground-truth amino acid type, x_i^{gt} and y_j^{gt} are correspond to real 3D coordinates of residue r_i and substrate atom v_j. l_{CE} and l_{Huber} denote cross-entropy and Huber loss, respectively. Then we obtain the total loss:

$$\mathcal{L} = \lambda_{\text{s}}\mathcal{L}_{\text{s}} + \lambda_{\text{ps}}\mathcal{L}_{\text{ps}} + \lambda_{\text{c}}\mathcal{L}_{\text{c}} + \lambda_{\text{pc}}\mathcal{L}_{\text{pc}} + \lambda_{\text{sub}}\mathcal{L}_{\text{sub}}, \tag{28}$$

where $\lambda_{[\cdot]}$ are hyperparameters for balancing these losses.

4 Experiments

4.1 EnzyPock: Dataset Curation

To build EnzyPock, we collect protein–ligand pairs from PDBbind [Wang *et al.*, 2004] and CrossDocked [Francoeur *et al.*, 2020], map PDB IDs to UniProt EC annotations, group enzymes by fourth-level EC labels, and infer conserved sites using Muscle5 [Edgar, 2022]. Pockets are extracted using Equation 1. For multi-chain proteins, we retain the first EC-annotated chain to avoid low-quality MSA analysis and excessive sequence lengths. When records conflict, we prioritize PDBbind over CrossDocked because PDBbind contains experimentally validated complexes. To prevent leakage, we cluster 17,404 PDB entries at 70% sequence identity and split non-overlapping clusters into training, validation, and test sets, as shown in Table 1. We choose this threshold to balance leakage control and dataset scale, since stricter thresholds substantially reduce the number of usable families and lead to overfitting.

Dataset	Training	Validation	Test
Sample Size	83,062	483	791
Third-Level Family	161	18	24
Fourth-Level Family	973	30	33
Average Seq. Length	324	314	322

Table 1: Statistics of EnzyPock. Sample size indicates the number of enzyme-substrate pairs. The middle two rows are the number of third and fourth-level EC families.

Method	AAR ↑	Vina ↓	pLDDT ↑	scRMSD ↓	scTM ↑
ESM3 (1.4B)	0.76	-3.46	**77.87**	5.26	0.89
RFD2+IF (84M)	0.56	-6.65	68.70	6.36	0.86
EnzyControl (21M)	0.49	-6.90	58.74	8.54	0.77
EnzyGen (714M)	0.64	-6.65	60.46	9.64	0.75
EnzyPGM (798M)	**0.77**	**-7.12**	74.87	**3.10**	**0.91**
w/o res-res attn	0.73	-7.08	73.63	4.54	0.88
w/o res-atom attn	0.73	-7.13	72.48	4.55	0.87
w/o pock & sub loss	0.73	-7.13	74.43	3.54	0.89
Freeze RFF	0.70	-6.82	69.75	6.16	0.84

Table 2: Quantitative results on EnzyPock.

Method	AAR ↑	Vina ↓	pLDDT ↑	scRMSD ↓	scTM ↑
ESM3 (1.4B)	0.64	-6.58	**71.64**	13.89	0.72
RFD2+IF (84M)	0.32	-7.15	61.91	11.81	0.72
EnzyControl (21M)	0.35	-7.18	60.64	10.88	0.71
EnzyGen (714M)	**0.88**	**-8.27**	70.10	**6.37**	**0.82**
EnzyPGM (798M)	0.71	-8.17	67.77	10.48	0.72

Table 3: Quantitative results on EnzyBench.

4.2 Experimental Settings

Baselines: We compare EnzyPGM with four representative baselines: ESM3 [Hayes *et al.*, 2025], RFD2+IF, EnzyControl [Song *et al.*, 2025], and EnzyGen [Song *et al.*, 2024]. RFD2+IF first uses RFDiffusion2 [Ahern *et al.*, 2025] to generate enzyme structures conditioned on functionally important sites, and then applies ProteinMPNN [Dauparas *et al.*, 2022] for inverse folding. Since ESM3 and RFDiffusion2 do not release their training scripts, we use their released model weights for generation. For EnzyControl, we use the public checkpoint and follow its evaluation pipeline.

Evaluation Metrics: We evaluate generated enzymes using amino acid recovery (AAR), Vina score (Vina) [Trott and Olson, 2010], pLDDT [Jumper *et al.*, 2021], scRMSD and scTM [Trippe *et al.*, 2023]. AAR measures the percentage of correctly recovered residue recovery in generated enzymes, Vina score estimates enzyme-substrate binding affinity, and pLDDT reflects the foldability of generated sequences. scRMSD and scTM measure structural consistency. For fair comparison across modeling paradigms, we compute structure-related metrics mainly on ESMFold-predicted structures and confirm that using native structures for structure-only baselines leads to consistent conclusions.

IJCAI-ECAI 2026 PREPRINT

PRELIMINARY PREPRINT VERSION – DO NOT CITE
This is not the version of record. The final version will appear in the Proceedings of IJCAI-ECAI 2026.

Binding affinity (Vina score ↓)										
EC Family	1.1	2.1	2.3	2.7	3.2	3.5	4.1	5.2	5.3	Avg.
ESM3 (1.4B)	-4.55	-0.60	-0.00	-2.04	-1.53	-5.67	-0.91	-3.71	-0.14	-3.52
RFD2+IF (84M)	**-5.62**	**-6.38**	-9.14	**-8.57**	**-7.27**	-6.47	-6.88	-6.79	-5.30	-7.01
EnzyControl (21M)	-4.74	-6.28	-8.57	-8.48	-6.59	-6.45	-6.96	**-7.10**	-5.88	-6.93
EnzyGen (714M)	-4.75	-6.12	-9.03	-8.44	-6.36	-5.96	-6.97	-6.80	**-6.06**	-6.66
EnzyPGM (798M)	-4.89	-6.21	**-9.37**	-8.39	-6.97	**-6.85**	**-7.15**	-6.98	-5.84	**-7.14**
Foldability (pLDDT ↑)										
EC Family	1.1	2.1	2.3	2.7	3.2	3.5	4.1	5.2	5.3	Avg.
ESM3 (1.4B)	**83.41**	**80.82**	**82.21**	**75.80**	74.84	**80.37**	**67.18**	78.89	84.58	**77.91**
RFD2+IF (84M)	77.15	77.90	70.07	73.15	43.66	71.84	56.38	73.07	81.63	68.56
EnzyControl (21M)	68.59	76.06	71.05	70.58	37.74	55.16	50.05	62.83	82.42	58.62
EnzyGen (714M)	63.31	75.78	53.34	70.47	45.00	58.73	50.65	63.08	76.60	60.33
EnzyPGM (798M)	79.96	76.27	75.68	71.34	**75.59**	79.25	49.68	**79.38**	**85.34**	74.89

Table 4: Binding affinity (Vina score) and Foldability (pLDDT) comparison across 9 main EC-2 families on EnzyPock.

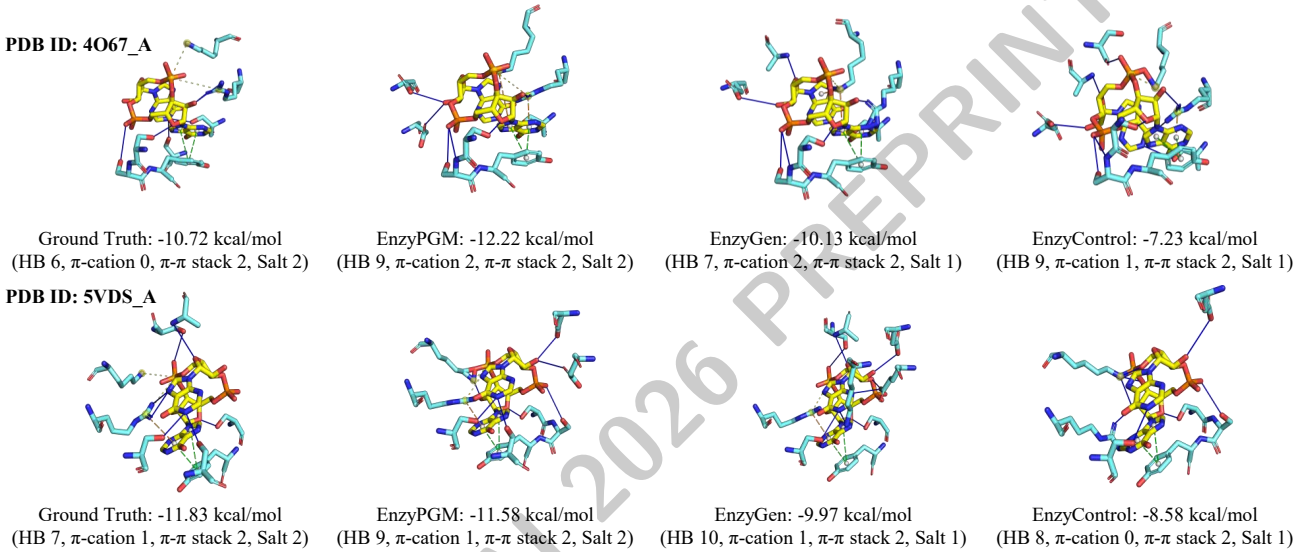


Figure 3: Visualization of ground-truth, EnzyPGM-generated, and two baseline-generated enzyme-substrate complexes. Each panel displays the enzyme pocket (blue) and substrate (yellow), alongside the Vina score and the number of enzyme-substrate interactions, including hydrogen bonds (HB), π-cation interactions, π-π stacks, and salt bridges (Salt).

IJCAI-ECAI 2026 PREPRINT

Implementation Details: EnzyPGM is composed of three stacked RFF and RBA modules, where each RFF contains 11 Transformer blocks initialized with ESM2 weights [Lin *et al.*, 2022]. A total of 3 individual SNE layers are used to process the substrate, and RBA only updates the pocket area within a 10Å radius of the substrate. The total parameter of EnzyPGM is 798M. The spatial K-nearest neighbor number in RFF, substrate SNE, and RBA are set to 30, 16, and 10, respectively. The hyperparameters λ_s, λ_{ps}, λ_c, λ_{pc}, and λ_{sub} are set to 1, 0.5, 1, 0.5, and 1, respectively. For training EnzyPGM on EnzyPock, we adopt AdamW with a learning rate of 1e-5 and a cosine learning schedule, dynamic batch size on GPU memory capacity, and use four NVIDIA A100 GPUs to train 30 epochs with 8000 warmup steps.

4.3 Quantitative Results

EnzyPGM realizes more effective enzyme design. As shown in Table 2, EnzyPGM achieves the highest AAR of 0.77 and the lowest Vina score of -7.12 on EnzyPock, surpassing EnzyGen by 0.13, reducing the average binding energy by 0.47 kcal/mol. Meanwhile, EnzyPGM exhibits state-of-the-art performance in scRMSD and scTM on EnzyPock, suggesting the generated enzymes are well-folded and of high substrate affinity. Although ESM3 obtains the highest pLDDT, EnzyPGM substantially outperforms it on Vina score while using fewer parameters, showing that explicit pocket-substrate interaction modeling is more effective for enzyme design than general protein pre-training. Also, we test all approaches on EnzyBench (Table 3), confirming that EnzyPGM has desirable performance and generalization, particularly in binding affinity. Specifically, we demonstrate the performance of all approaches on 9 main EC-2 families of EnzyPock in Table 4. These results demonstrate that pocket modeling benefits to generate enzymes with higher substrate-binding capability and structural validity, leading to more effective enzyme design.

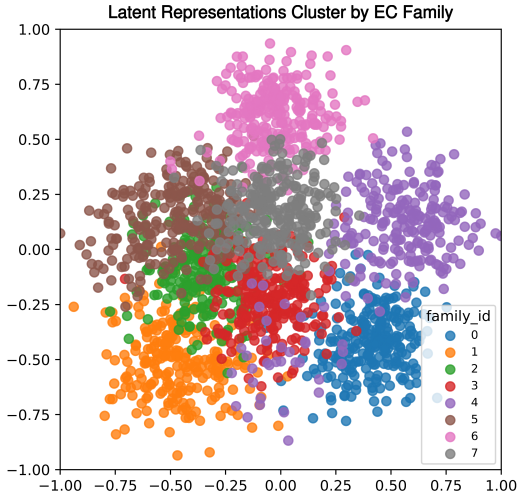


Figure 4: Analysis of representation learning.

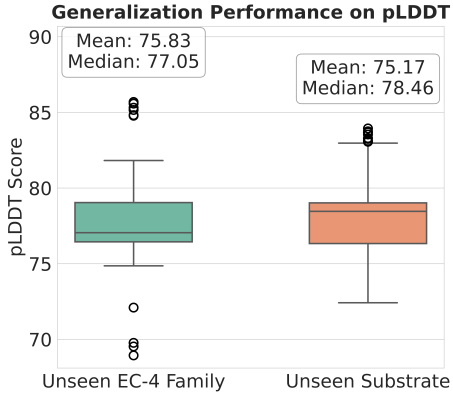


Figure 5: Analysis on pLDDT generalization.

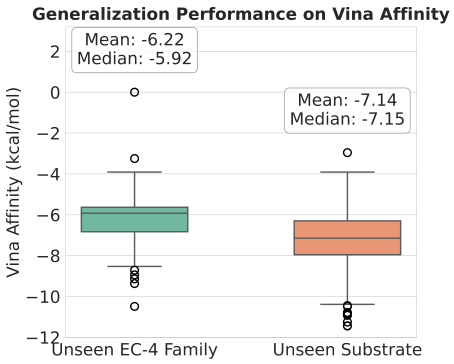


Figure 6: Analysis on Vina score generalization.

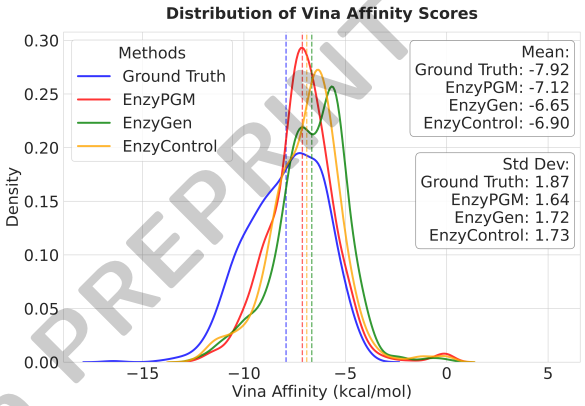


Figure 7: Analysis on Vina score distribution.

4.4 Ablation Studies

All designs in EnzyPGM contribute to enzyme generation. We conduct ablation experiments to assess the contribution of core designs in EnzyPGM, which are the intra-residue and residue-atom attention, pocket and substrate loss, and RFF module. As shown in Table 2, removing two attentions results in a 1.24 and 2.39 drop in pLDDT, respectively. This confirms that the bi-scale attention improves performance in enzyme sequence design. Removing the pocket and substrate loss leads to a modest drop in overall performance, confirming that they promote model training. Freezing the RFF during training results in a performance drop across all metrics, suggesting that the backbone model is essential for EnzyPGM. These results confirm that EnzyPGM's core designs jointly promote its enzyme design performance.

4.5 Case Studies

EnzyPGM designs high-affinity binding pockets. To demonstrate EnzyPGM's performance in generating pockets, we perform a case study to compare it with 2 baselines and the ground truth. Visualization in Figure 3 reveals that EnzyPGM generates valid substrate-binding pockets with a lower Vina score than baselines and more abundant hydrogen bonds, which means the residues in the pocket form strong chemical bonds with the substrate. In contrast, EnzyGen and EnzyControl generate irregular binding regions where residues struggle to form stable chemical bonds with the substrate. These results highlight that pocket-conditioned training not only improves enzyme design quality but also generates valid and stable substrate-binding pockets.

5 Conclusion

We propose EnzyPGM, a unified pocket-conditioned generative model for substrate-specific enzyme design. By integrating residue-atom bi-scale attention and enzyme function priors, EnzyPGM jointly models enzyme generation and pocket-substrate interactions. We also curate EnzyPock, a dataset containing 84,336 enzyme-substrate pairs across 1,036 fourth-level EC families. Experiments on EnzyPock show EnzyPGM outperforms state-of-the-art methods in both binding affinity and structure validity metrics, highlighting its effectiveness for substrate-specific enzyme design.

A Further Analysis

We provide additional t-SNE, generalization, and Vina-distribution analyses (Figure 4, 5, 6, and 7), showing that EnzyPGM learns function-aware representations and maintains stable substrate-specific binding performance.

IJCAI-ECAI 2026 PREPRINT

Acknowledgements

Tianfan Fu is supported by Young Scientists Fund (C Class) of the National Natural Science Foundation of China (Grant No. 62506154), the Fundamental Research Funds for the Central Universities and Nanjing University International Collaboration Initiative (Grant No. 020214380129) and the "111 Center"(No. B26023).

Contribution Statement

Zefeng Lin, Zhihang Zhang, Weirong Zhu, and Tongchang Han contributed equally to this work. Zefeng Lin led the method design, implementation, experiments, and manuscript writing. Zhihang Zhang contributed to model development and analysis. Weirong Zhu and Tongchang Han contributed to dataset construction, evaluation, and visualization.

References

[Ahern *et al.*, 2025] Woody Ahern, Jason Yim, Doug Tischer, Saman Salike, Seth M Woodbury, Donghyo Kim, Indrek Kalvet, Yakov Kipnis, Brian Coventry, Han Raut Altae-Tran, et al. Atom level enzyme active site scaffolding using rfdiffusion2. *bioRxiv*, page 2025.04.09.648075, 2025.

[Baek *et al.*, 2021] Minkyung Baek, Frank DiMaio, Ivan Anishchenko, Justas Dauparas, Sergey Ovchinnikov, Gyu Rie Lee, Jue Wang, Qian Cong, Lisa N Kinch, R Dustin Schaeffer, et al. Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, 373(6557):871–876, 2021.

[Bar-Even *et al.*, 2011] Arren Bar-Even, Elad Noor, Yonatan Savir, Wolfram Liebermeister, Dan Davidi, Dan S Tawfik, and Ron Milo. The moderately efficient enzyme: evolutionary and physicochemical trends shaping enzyme parameters. *Biochemistry*, 50(21):4402–4410, 2011.

[Capra and Singh, 2007] John A Capra and Mona Singh. Predicting functionally important residues from sequence conservation. *Bioinformatics*, 23(15):1875–1882, 2007.

[Chu *et al.*, 2024] Alexander E Chu, Tianyu Lu, and Po-Ssu Huang. Sparks of function by de novo protein design. *Nature Biotechnology*, 42(2):203–215, 2024.

[Dauparas *et al.*, 2022] Justas Dauparas, Ivan Anishchenko, Nathaniel Bennett, Hua Bai, Robert J Ragotte, Lukas F Milles, Basile IM Wicky, Alexis Courbet, Rob J de Haas, Neville Bethel, et al. Robust deep learning–based protein sequence design using proteinmpnn. *Science*, 378(6615):49–56, 2022.

[Dauparas *et al.*, 2025] Justas Dauparas, Gyu Rie Lee, Robert Pecoraro, Linna An, Ivan Anishchenko, Cameron Glasscock, and David Baker. Atomic context-conditioned protein sequence design using ligandmpnn. *Nature Methods*, pages 1–7, 2025.

[Dauphin *et al.*, 2017] Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. In *ICML*, pages 933–941. PMLR, 2017.

[Du *et al.*, 2025] Jiahe Du, Kaixiong Zhou, Xinyu Hong, Zhaozhuo Xu, Jinbo Xu, and Xiao Huang. Retrieval augmented zero-shot enzyme generation for specified substrate. In *ICML*, 2025.

[Edgar, 2022] Robert C Edgar. Muscle5: High-accuracy alignment ensembles enable unbiased assessments of sequence homology and phylogeny. *Nature communications*, 13(1):6968, 2022.

[Ferruz *et al.*, 2022] Noelia Ferruz, Steffen Schmidt, and Birte Höcker. Protgpt2 is a deep unsupervised language model for protein design. *Nature Communications*, 13(1):4348, 2022.

[Francoeur *et al.*, 2020] Paul G Francoeur, Tomohide Masuda, Jocelyn Sunseri, Andrew Jia, Richard B Iovanisci, Ian Snyder, and David R Koes. Three-dimensional convolutional neural networks and a cross-docked data set for structure-based drug design. *Journal of chemical information and modeling*, 60(9):4200–4215, 2020.

[Hayes *et al.*, 2025] Thomas Hayes, Roshan Rao, Halil Akin, Nicholas J Sofroniew, Deniz Oktay, Zeming Lin, Robert Verkuil, Vincent Q Tran, Jonathan Deaton, Marius Wiggert, et al. Simulating 500 million years of evolution with a language model. *Science*, 387(6736):850–858, 2025.

[Hsu *et al.*, 2022] Chloe Hsu, Robert Verkuil, Jason Liu, Zeming Lin, Brian Hie, Tom Sercu, Adam Lerer, and Alexander Rives. Learning inverse folding from millions of predicted structures. In *ICML*, pages 8946–8970. PMLR, 2022.

[Hua *et al.*, 2024] Chenqing Hua, Jiarui Lu, Yong Liu, Odin Zhang, Jian Tang, Rex Ying, Wengong Jin, Guy Wolf, Doina Precup, and Shuangjia Zheng. Reaction-conditioned de novo enzyme design with genzyme. *arXiv preprint arXiv:2411.16694*, 2024.

[Huang *et al.*, 2021] Juan Huang, Qin Xu, Zhuo Liu, Nitin Jain, Madhusudan Tyagi, Dong-Qing Wei, and Liang Hong. Controlling the substrate specificity of an enzyme through structural flexibility by varying the salt-bridge density. *Molecules*, 26(18):5693, 2021.

[Jumper *et al.*, 2021] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

[Kortemme, 2024] Tanja Kortemme. De novo protein design—from new structures to programmable functions. *Cell*, 187(3):526–544, 2024.

[Lesk, 2001] Arthur M Lesk. *Introduction to protein architecture: the structural biology of proteins*, volume 741. Oxford university press Oxford, 2001.

[Lin *et al.*, 2022] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Sal Candido, et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *bioRxiv*, 2022.

IJCAI-ECAI 2026 PREPRINT

[Meshchaninov *et al.*, 2025] Viacheslav Meshchaninov, Pavel Strashnov, Andrey Shevtsov, Fedor Nikolaev, Nikita Ivanisenko, Olga Kardymon, and Dmitry Vetrov. Diffusion on language model encodings for protein sequence generation. In *ICML*, 2025.

[Nijkamp *et al.*, 2023] Erik Nijkamp, Jeffrey A Ruffolo, Eli N Weinstein, Nikhil Naik, and Ali Madani. Progen2: exploring the boundaries of protein language models. *Cell Systems*, 14(11):968–978, 2023.

[Notin *et al.*, 2024] Pascal Notin, Nathan Rollins, Yarin Gal, Chris Sander, and Debora Marks. Machine learning for functional protein design. *Nature Biotechnology*, 42(2):216–228, 2024.

[Satorras *et al.*, 2021] Vıctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. In *ICML*, pages 9323–9332. PMLR, 2021.

[Song *et al.*, 2024] Zhenqiao Song, Yunlong Zhao, Wenxian Shi, Wengong Jin, Yang Yang, and Lei Li. Generative enzyme design guided by functionally important sites and small-molecule substrates. In *ICML*, pages 46259–46279. PMLR, 2024.

[Song *et al.*, 2025] Chao Song, Zhiyuan Liu, Han Huang, Liang Wang, Qiong Wang, Jian-Yu Shi, Hui Yu, Yihang Zhou, and Yang Zhang. Enzycontrol: Adding functional and substrate-specific control for enzyme backbone generation. In *NeurIPS*, 2025.

[Stank *et al.*, 2016] Antonia Stank, Daria B Kokh, Jonathan C Fuller, and Rebecca C Wade. Protein binding pocket dynamics. *Accounts of Chemical Research*, 49(5):809–815, 2016.

[Stärk *et al.*, 2022] Hannes Stärk, Octavian Ganea, Lagnajit Pattanaik, Regina Barzilay, and Tommi Jaakkola. Equibind: Geometric deep learning for drug binding structure prediction. In *ICML*, pages 20503–20521. PMLR, 2022.

[Tipton and Boyce, 2000] Keith Tipton and Sinéad Boyce. History of the enzyme nomenclature system. *Bioinformatics*, 16(1):34–40, 2000.

[Trippe *et al.*, 2023] Brian L Trippe, Jason Yim, Doug Tischer, David Baker, Tamara Broderick, Regina Barzilay, and Tommi S Jaakkola. Diffusion probabilistic modeling of protein backbones in 3d for the motif-scaffolding problem. In *ICLR*, 2023.

[Trott and Olson, 2010] Oleg Trott and Arthur J Olson. Autodock vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of Computational Chemistry*, 31(2):455–461, 2010.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 30, 2017.

[Wang *et al.*, 2004] Renxiao Wang, Xueliang Fang, Yipin Lu, and Shaomeng Wang. The pdbbind database: Collection of binding affinities for protein- ligand complexes with known three-dimensional structures. *Journal of medicinal chemistry*, 47(12):2977–2980, 2004.

[Wang *et al.*, 2024] Xinyou Wang, Zaixiang Zheng, Fei Ye, Dongyu Xue, Shujian Huang, and Quanquan Gu. Diffusion language models are versatile protein learners. In *ICML*, pages 52309–52333. PMLR, 2024.

[Wang *et al.*, 2025] Yuhang Wang, Shuangxin Han, Yi Wang, Quanfeng Liang, and Wei Luo. Artificial intelligence technology assists enzyme prediction and rational design. *Journal of Agricultural and Food Chemistry*, 73(12):7065–7073, 2025.

[Watson *et al.*, 2023] Joseph L Watson, David Juergens, Nathaniel R Bennett, Brian L Trippe, Jason Yim, Helen E Eisenach, Woody Ahern, Andrew J Borst, Robert J Ragotte, Lukas F Milles, et al. De novo design of protein structure and function with rfdiffusion. *Nature*, 620(7976):1089–1100, 2023.

[Xiong *et al.*, 2020] Ruibin Xiong, Yiming Yang, Di He, Kai Zheng, Shuxin Zheng, Chenxiang Xing, Huishuai Zhang, Yanyan Lan, Liqiang Wang, and Tie-Yan Liu. On layer normalization in the transformer architecture. *arXiv preprint arXiv:2002.04745*, 2020.

[Zhang *et al.*, 2024a] Zaixi Zhang, Wan Xiang Shen, Qi Liu, and Marinka Zitnik. Efficient generation of protein pockets with pocketgen. *Nature Machine Intelligence*, 6(11):1382–1395, 2024.

[Zhang *et al.*, 2024b] Zaixi Zhang, Marinka Zitnik, and Qi Liu. Generalized protein pocket generation with prior-informed flow matching. *NeurIPS*, 37:38559–38589, 2024.

IJCAI-ECAI 2026 PREPRINT