FBINDGEN: FUNCTION-CONDITIONED PROTEIN BINDER SEQUENCE GENERATION WITH TARGET-AWARE PAIR MODELING

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing protein binder design methods primarily focus on structural compatibility with a specified target, while direct control over binder function remains less explored. We introduce FBindGen, a diffusion protein language model that generates binder sequences from a target protein and functional annotations at the residue level. FBindGen combines functional conditioning with target-aware pair modeling by allowing sequence and pair representations to update each other throughout denoising. To evaluate how well generated binders follow the specified function, we introduce EnzBind, a benchmark of protein complexes involving enzymes with functional annotations and substrate information. On EnzBind and DB5.5, FBindGen shows improved agreement between generated sequences and the specified functions while maintaining competitive predicted interface confidence. On EnzBind, it also achieves the highest predicted catalytic efficiency among the evaluated methods. Evaluation on Heterodimer99 further shows that the learned pair representations capture interchain contact information. Together, these results show that FBindGen provides functional control over binder sequence generation while preserving compatibility with the target.

1 INTRODUCTION

Protein binder design aims to generate proteins that interact with a specified target. Recent diffusion models enable target-conditioned protein backbone generation and joint sequence–structure generation (Watson et al., 2023; Song et al., 2025; Stark et al., 2025). However, compatibility with the target alone does not specify the function of the designed binder. Beyond target binding, the generated protein may need to belong to a specified protein family, contain particular functional domains, or exhibit a desired enzymatic activity. When functionally important residues and their geometric arrangements are known, such requirements can be imposed through structural motifs or atom-level constraints (Watson et al., 2023; Butcher et al., 2025). Functional annotations provide an alternative way to describe the desired protein function without requiring a predefined structural motif (Yin et al., 2025).

Function-conditioned binder generation is challenging because a successful design must satisfy several requirements simultaneously. The binder must adopt a plausible structure, form a compatible interface with the target, and satisfy the specified functional requirements. Target compatibility alone does not determine these properties: residues important for function may lie outside the binding interface, while interface residues must remain compatible with the overall structure of the binder. A model must therefore capture residue relationships both within the binder and across the target–binder interface (Liu et al., 2026), while maintaining explicit control over the desired function.

To address this challenge, we introduce FBindGen, a diffusion protein language model for function-conditioned binder generation. Given a target sequence and structure together with functional annotations for the desired binder, FBindGen generates the binder sequence through iterative denoising. The model combines a pair representation module with a functional conditioning module. The pair representations model residue relationships within the binder and across the target–binder interface without requiring a predefined binder structure, while the functional module conditions sequence generation on the desired function. Bidirectional sequence–pair updates couple these representa-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

tions throughout denoising, allowing both functional annotations and target-aware pair information to guide sequence generation.

We introduce EnzBind, a benchmark of enzyme-associated protein complexes for evaluating functional annotation agreement and target compatibility. We evaluate functional alignment and predicted interface confidence on EnzBind and the public DB5.5 benchmark (Guest et al., 2021). For EnzBind, we also use UniKP to predict catalytic efficiency. We further evaluate interchain contact prediction on Heterodimer99 to examine the interaction information captured by the learned pair representations (Lin et al., 2023a). Across these evaluations, FBindGen more closely follows the supplied functional conditions while maintaining competitive predicted interface confidence, and its learned pair representations capture interchain contact information.

Our main contributions are summarized as follows:

- We introduce FBindGen, a diffusion protein language model that generates binder sequences from a target protein and functional annotations. The model combines functional conditioning with target-aware pair modeling during denoising.
- We introduce EnzBind, a benchmark of enzyme-associated protein complexes with functional annotations and available substrate information. It supports evaluation of functional annotation agreement, predicted interface confidence, and predicted catalytic efficiency.
- Experiments on EnzBind and DB5.5 show improved sequence–function alignment with competitive predicted interface confidence. On EnzBind, FBindGen also achieves the highest predicted catalytic efficiency among the evaluated methods. Evaluation on Heterodimer99 further shows that the learned pair representations capture interchain contact information.

2 Related Work

Target-conditioned binder design. Target-conditioned binder design uses several strategies, including backbone generation, sequence optimization through structure predictors, and joint sequence–structure modeling. RFdiffusion generates binder backbones conditioned on a target, and ProteinMPNN then assigns sequences to these backbones (Watson et al., 2023; Dauparas et al., 2022). Structure predictors also support binder design: BindCraft uses AlphaFold2-based optimization to design binder interfaces (Pacesa et al., 2025), while ESMFold2 has been used for binder design through structure-predictor inversion (Candido et al., 2026). PPDiff jointly denoises binder sequences and structures, while BoltzGen unifies all-atom design and structure prediction (Song et al., 2025; Stark et al., 2025). Proteina-Complexa combines an atomistic generative prior with inference-time optimization (Didi et al., 2026). Functional requirements can also be expressed as geometric constraints, as in RFdiffusion's motif scaffolding and RFdiffusion3's atom-level constraints (Watson et al., 2023; Butcher et al., 2025). Direct sequence generation offers another route: DSM(ppi) generates binder sequences conditioned on a target sequence (Hallee et al., 2025).

Function-conditioned protein generation. Protein generation can be guided by structured functional labels or natural-language descriptions. ProGen conditions autoregressive generation on protein-property tags (Madani et al., 2023), while ProteinDT and ProtDAT use natural-language descriptions to guide sequence generation (Liu et al., 2025; Guo et al., 2025). ESM3 jointly models sequence, structure, and function and supports multimodal prompts (Hayes et al., 2025). CFP-Gen combines composable functional annotations with optional sequence-motif constraints and supports structure-conditioned functional inverse folding (Yin et al., 2025). In its structure-conditioned setting, the supplied structure describes the protein being designed rather than a separate interaction partner. FBindGen combines functional conditioning with information about a separate target: residue-aligned InterPro identifiers specify the desired binder annotations, while the target sequence and structure describe the binder's interaction partner.

Diffusion language models and sequence–structure generation. DPLM learns protein sequence distributions through discrete denoising and supports conditional generation (Wang et al., 2024). DPLM-2 extends this approach to joint sequence and structure modeling with discrete structural tokens, enabling co-generation, folding, and inverse folding (Wang et al., 2025). FBindGen encodes structural relationships in internal residue-pair features and generates binder amino acid sequences.

Coupled sequence and pair representations. AlphaFold2's Evoformer exchanges information between multiple-sequence-alignment and pair representations (Jumper et al., 2021). ESMFold combines language-model features with a folding trunk that updates sequence and pair states (Lin et al., 2023b). AlphaFold3 uses a Pairformer to process single and pair representations before diffusion-based coordinate prediction (Abramson et al., 2024). ESMFold2 combines ESMC language-model representations with a recurrent folding architecture and diffusion-based all-atom structure prediction (Candido et al., 2026).

Learning protein–protein interaction representations. DeepInter combines protein language models with geometric triangle-aware modeling to predict interchain contacts (Lin et al., 2023a). PPLM jointly encodes paired protein sequences to learn representations for interaction, affinity, and contact prediction (Liu et al., 2026). Its PPLM-Contact predictor combines paired-sequence attention with evolutionary features and monomer distance maps (Liu et al., 2026). These methods predict properties of supplied protein sequences, whereas FBindGen generates the binder sequence conditioned on target information and functional annotations.

3 METHOD

3.1 PROBLEM FORMULATION

Given a target sequence x^T, its backbone coordinates C^T, and a specified binder length L_B, we aim to generate a binder sequence x^B conditioned on functional annotations F^B:

$$p_\theta(x^B \mid x^T, C^T, F^B, L_B).$$

We specify the functional condition using per-residue InterPro annotations, $F^B = (F_1^B, \ldots, F_{L_B}^B)$, where F_i^B is the set of annotations requested at binder position i. The target sequence and structure describe the interaction partner, while F^B specifies the requested functional annotations of the binder. In our benchmarks, F^B is derived from the reference binder annotations. Binder coordinates are not required as inputs.

FBindGen generates binder sequences through discrete diffusion and is trained to reconstruct masked binder residues. At timestep t, the model takes the corrupted binder sequence x_t^B, together with the fixed target inputs and functional annotations, and predicts an amino acid distribution at each binder position. At inference, generation starts from a fully masked binder and progressively assigns amino acids according to the model predictions, keeping the target inputs, binder length, and annotations fixed. Appendix A.3 details the decoding procedure used for EnzBind.

3.2 SEQUENCE AND PAIR REPRESENTATIONS

FBindGen combines a bidirectional Transformer sequence backbone with a residue-pair module (Figure 1). Each forward pass starts from the concatenated target and current binder tokens, with learned chain embeddings distinguishing the two chains. The backbone produces per-position hidden states H for amino acid prediction, while the pair module maintains features Z for residue pairs within and across the two chains. Let $h_i^{(\ell)}$ denote the hidden state at position i in layer ℓ.

Encoding the functional condition. We add annotation features to the binder token embeddings so that functional information remains available when amino acid identities are masked. At each binder position, we sum the learned embeddings E_k of the InterPro identifiers in F_i^B and project the sum into the token-embedding space:

$$e_i^F = \sum_{k \in F_i^B} E_k, \qquad \widetilde{e}_i^{\mathrm{seq}} = e_i^{\mathrm{seq}} + \mathrm{MLP}_F(e_i^F).$$

Here, e_i^{seq} is the token embedding. A separate projection of e_i^F is added to the aggregated sequence features used to compute pair updates.

Initializing residue-pair features. We project input sequence representations into residue features s_i and combine them with relative-position and chain encodings:

$$z_{ij}^{(0,\mathrm{in})} = W_L s_i + W_R s_j + e_{ij}^{\mathrm{pos}} + e_{ij}^{\mathrm{chain}} + \mathbf{1}[i,j \in \mathcal{I}_T]\, \phi(C_i^T, C_j^T).$$

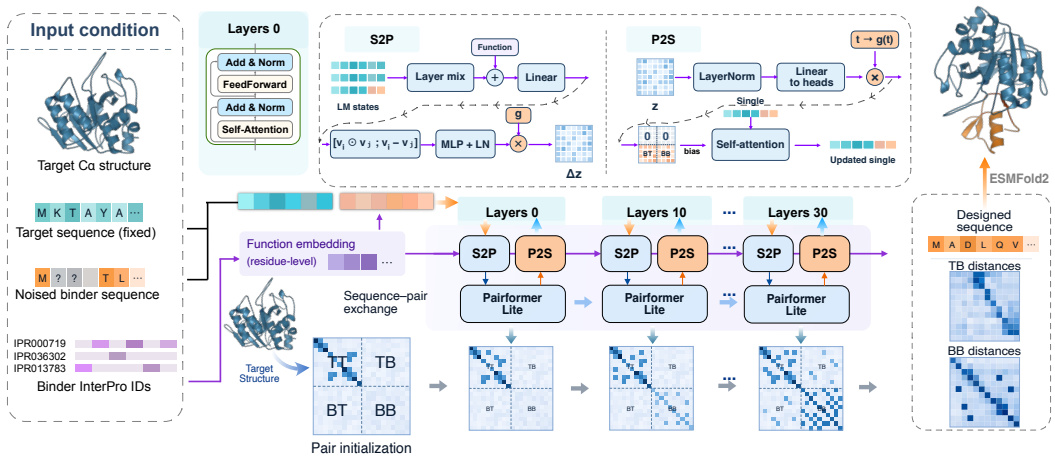

Figure 1: **FBindGen architecture.** At each forward pass, the model takes the target sequence and structure, current binder tokens, and per-residue InterPro annotations as inputs. Projected annotation embeddings are added to binder token embeddings and to the aggregated sequence features used to construct sequence-to-pair (S2P) updates. Target structure features are added to the target–target pair representation at initialization. Sequence features update the pair representations, which are refined by Pairformer blocks. The pair-to-sequence (P2S) projection converts the refined pair representations into attention biases for binder queries in the sequence backbone. Prediction heads use the final sequence representations to predict binder amino acids and the final pair representations to predict auxiliary target–binder (TB) and binder–binder (BB) distance distributions.

Here, $\mathcal{I}_T$ and $\mathcal{I}_B$ denote target and binder residue indices, excluding special tokens. The target-structure feature ϕ encodes Cα distances and directions in local backbone frames. Only target–target pairs receive geometry; target–binder, binder–target, and binder–binder pairs are initialized from residue, position, and chain features. Appendix A.1 gives the projections, coordinate-validity requirements, and embedding settings.

3.3 Bidirectional Sequence–Pair Coupling

Sequence states update the pair representations, which in turn provide attention biases to the sequence backbone. This bidirectional exchange occurs at four points in each forward pass and uses two Pairformer blocks to refine the updated pair representations. Pair representations are passed from one exchange to the next within a forward pass and reinitialized from the current sequence tokens, functional annotations, and target structure at the start of every forward pass.

Step 1: From sequence states to pair updates. At each exchange b, we aggregate sequence states from an associated layer group using learned weights, add a projection of the binder annotations, and project the result to residue features $v_i^{(b)}$. Their element-wise product and signed difference provide symmetric and directional pair features:

$$\Delta z_{ij}^{\mathrm{seq},b} = g_b \,\mathrm{LN}\Big[\mathrm{MLP}_b\Big[\Big(v_i^{(b)} \odot v_j^{(b)}\Big) \,\|\, \Big(v_i^{(b)} - v_j^{(b)}\Big)\Big]\Big],$$

where $\|$ denotes concatenation and g_b is a learned scale. The first exchange uses the input representations.

Step 2: Refining the updated pair representations. The first exchange refines $Z^{(0,\mathrm{in})} + \Delta Z_0^{\mathrm{seq}}$. Later exchanges also add updates from the latest residue states ($\Delta Z_b^{\mathrm{res}}$) and bidirectional attention maps ($\Delta Z_b^{\mathrm{attn}}$) to the preceding pair state:

$$Z^{(b)} = \mathrm{Pairformer}_b\Big(Z^{(b^-)} + \Delta Z_b^{\mathrm{seq}} + \Delta Z_b^{\mathrm{res}} + \Delta Z_b^{\mathrm{attn}}\Big),$$

where b^- is the preceding exchange and all updates act on valid residue pairs. Triangle operations exchange information through a third residue, while pair transitions transform each pair independently.

Step 3: Returning pair information to the sequence backbone. To feed the refined pair features back into sequence generation, we project $Z^{(b)}$ into additive attention biases. The biases are added only to attention logits for binder queries and are scaled by a learned, timestep-dependent scalar gate $\gamma_b(t)$:

$$B_{ij}^{(b)}(t) = \mathbf{1}[i \in \mathcal{I}_B]\, \gamma_b(t) \left[W_b^Z \, \mathrm{LN}(Z^{(b)}) \right]_{ij},$$

$$\mathrm{Attn}_b(Q, K, V) = \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d_k}} + B^{(b)}(t) \right) V.$$

Here, Q, K, and V are the attention queries, keys, and values, and d_k is the key dimension; the attention-head index is omitted for clarity. Adding $B^{(b)}(t)$ changes how binder positions attend to residues in either chain according to the learned pair features. The resulting hidden states are used to compute subsequent pair updates, completing the bidirectional exchange. Appendix A.1 specifies the update positions, layer groups, updates from attention maps, and timestep-dependent gates.

3.4 Training objectives

We jointly train all model components using sequence reconstruction and auxiliary distance prediction losses. An amino acid prediction head uses the final sequence representations to predict amino acid distributions at binder positions, while separate target–binder (TB) and binder–binder (BB) distance prediction heads use the final pair representations to predict residue-pair distances. During training, reference binder coordinates and interchain distances provide supervision for the distance prediction heads; they are not model inputs.

Learning to reconstruct the binder. At timestep t of a T-step absorbing-state masking process, binder residues are independently masked with probability t/T, leaving target and special tokens unchanged. The sequence loss $\mathcal{L}_{\mathrm{seq}}$ is a timestep-weighted cross-entropy over masked binder positions. Because this weighting downweights highly corrupted examples while generation starts from a fully masked binder, we also include an unweighted reconstruction term $\mathcal{L}_{\mathrm{full}}$ for fully masked binders.

Auxiliary distance prediction. Both distance heads predict coarse Cα-distance classes and finer distance distributions within the contact range. The coarse objective combines class-wise cross-entropies, while the fine objective minimizes the KL divergence from Gaussian reference-distance targets to predicted distributions. The full objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{seq}} + \lambda_F \mathcal{L}_{\mathrm{full}} + \sum_{r \in \{TB, BB\}} \lambda_r \left(\frac{2}{3} \mathcal{L}_r^{\mathrm{coarse}} + \frac{1}{3} \mathcal{L}_r^{\mathrm{fine}} \right).$$

Appendix A.2 specifies timestep sampling and weighting, loss normalization, distance bins, coefficients, and optimization settings.

4 Experiments

We evaluate whether FBindGen follows the supplied functional annotations, whether stronger annotation agreement is accompanied by competitive predicted interface confidence, and what interchain contact information the pair representations encode. EnzBind and DB5.5 assess generation, while Heterodimer99 probes contact prediction. Controlled annotation perturbations and paired generation tests then examine how these outputs depend on the functional condition.

4.1 Experimental setup

Training. We train FBindGen on 497,581 interacting chain pairs from experimental PDB structures and predicted AlphaFold heterodimers. The sequence backbone and added modules are jointly optimized with AdamW. Optimization settings and training data construction are described in Appendices A.2 and B.

Datasets. EnzBind is an enzyme-focused benchmark comprising 275 protein complexes. It includes InterPro functional annotations, Enzyme Commission (EC) annotations, and, where available, substrate representations derived from the corresponding reaction records. DB5.5 is a protein–protein docking benchmark that includes enzyme-associated complexes, antibody–antigen complexes, and other types of protein complexes (Guest et al., 2021). We use a subset of 127 two-chain complexes for binder-generation evaluation. Heterodimer99 comprises 99 heterodimeric protein complexes curated from the PDB, with experimentally determined structures that define reference interchain contacts (Lin et al., 2023a). Dataset and evaluation protocols are provided in Appendices C–E.

Baselines. For binder generation, we compare FBindGen with BoltzGen, Proteina-Complexa, PPDiff, and DSM(ppi) on EnzBind and DB5.5 (Stark et al., 2025; Didi et al., 2026; Song et al., 2025; Hallee et al., 2025). On EnzBind, we additionally compare with RFdiffusion and RFdiffusion3, each paired with ProteinMPNN for sequence design (Watson et al., 2023; Butcher et al., 2025; Dauparas et al., 2022). Throughout the experimental results, RFdiffusion and RFdiffusion3 denote these ProteinMPNN-based pipelines. For each baseline, we use the generation parameters recommended in its official code repository. We evaluate one binder sequence per method and case.

Generation. On EnzBind, FBindGen gives priority to high-confidence amino acid predictions during unmasking. The current masked fraction determines the timestep. After sequence completion, we perform up to two rounds of local refinement by remasking positions with high predictive entropy and resampling their amino acids. Appendices A.3–A.4 detail the sampling procedure, composition controls, and generation settings.

Evaluation. We assess sequence–function alignment using ProTrek similarity between generated sequences and descriptions of the requested functional annotations (Su et al., 2024). We use InterProScan to assess recovery of the supplied InterPro identifiers (Jones et al., 2014). We predict target–binder complex structures with ESMFold2-Fast on EnzBind and DB5.5, and report structural and interface confidence scores. For the enzyme-associated designs, UniKP predicts catalytic properties from the generated sequences and substrate representations (Yu et al., 2023).

4.2 Functional control on EnzBind

FBindGen leads all six ProTrek metrics among the evaluated generators on EnzBind (Table 1). Its mean cosine similarity is 0.31, compared with 0.23 for PPDiff. The requested description is the top match for 45.5% of generated sequences, versus 10.9% for BoltzGen; the paired case entry ranks in the top 10 for 70.9%, versus 24.7% for PPDiff.

Table 1: **Sequence–text alignment on EnzBind using ProTrek.** Bold indicates the best result among generators, excluding the native reference. Metric definitions and ranking conventions are given in Appendix C.3.

Method	Mean cosine ↑	MRR ↑	Text Top-1 (%) ↑	R@5 (%) ↑	R@10 (%) ↑	Text macro-F1 ↑
Native reference	0.37	0.70	71.3	86.9	94.2	0.67
PPDiff	0.23	0.15	10.5	18.5	24.7	0.13
DSM(ppi)	0.19	0.04	1.5	2.9	5.5	0.01
BoltzGen	0.17	0.14	10.9	14.9	20.0	0.10
RFdiffusion	0.15	0.03	1.5	3.6	5.5	0.00
Proteina-Complexa	0.13	0.10	6.9	10.5	14.2	0.06
RFdiffusion3	0.07	0.02	0.4	1.8	5.1	0.00
FBindGen	**0.31**	**0.47**	**45.5**	**61.8**	**70.9**	**0.35**

InterProScan tests whether the supplied InterPro identifiers can also be recovered from the generated sequences. FBindGen reaches a micro-F1 of 0.75, compared with 0.15 for PPDiff (Table 2). Its micro-precision of 0.82 exceeds its micro-recall of 0.69: most detected identifiers match the requested sets. Text retrieval and identifier recovery provide evidence of annotation agreement beyond embedding similarity alone.

Table 2: **Functional annotation recovery on EnzBind using InterProScan.** Predicted InterPro identifiers are compared with the supplied annotations. Bold indicates the best result among generators, excluding the native reference. Metric definitions are given in Appendix C.4.

Method	Micro precision ↑	Micro recall ↑	Micro F1 ↑	Macro F1 ↑	Macro AP ↑	Macro ROC-AUC ↑
Native reference	1.00	1.00	1.00	1.00	0.96	0.96
PPDiff	0.27	0.11	0.15	0.11	0.10	0.54
DSM(ppi)	0.10	0.00	0.00	0.00	0.01	0.50
BoltzGen	0.14	0.10	0.12	0.10	0.09	0.54
RFdiffusion	0.03	0.00	0.00	0.00	0.01	0.50
Proteina-Complexa	0.05	0.03	0.04	0.04	0.05	0.52
RFdiffusion3	0.00	0.00	0.00	0.00	0.01	0.50
FBindGen	**0.82**	**0.69**	**0.75**	**0.52**	**0.52**	**0.75**

The stronger annotation agreement is accompanied by a mean ipTM of 0.30, close to BoltzGen's 0.29 (Figure 2). Across 321 case–substrate pairs from 255 complexes, FBindGen also has the highest arithmetic mean UniKP-predicted catalytic efficiency (12.49 s^{-1}·mM^{-1}, versus 12.12 for DSM(ppi)).

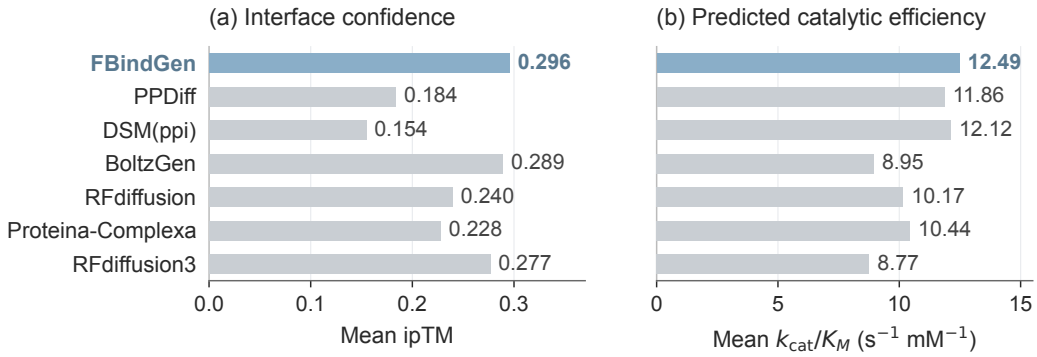

Figure 2: **Interface confidence and predicted catalytic efficiency on EnzBind.** (a) Mean ipTM over 275 predicted complexes. (b) Arithmetic mean UniKP-predicted k_{cat}/K_M over 321 case–substrate pairs from 255 complexes. Blue denotes FBindGen; gray denotes baselines.

4.3 Functional control and interface confidence on DB5.5

The public DB5.5 benchmark extends the evaluation beyond the enzyme-focused EnzBind collection to a broader range of protein–protein interactions. FBindGen reaches a mean ProTrek similarity of 0.28 versus 0.21 for PPDiff, with Top-1 accuracy of 41.3% versus 13.5% and MRR of 0.50 versus 0.19 (Table 3). The retrieval advantage therefore also appears outside EnzBind.

FBindGen's mean ipTM is 0.40, compared with 0.36 for BoltzGen. The number of designs with ipTM $\geq$ 0.8 rises to 27 (21.3%), compared with 14 (11.0%) for Proteina-Complexa, the strongest baseline at this threshold (Table 5). Thus, the interface-confidence advantage includes both the mean score and the fraction above the reported threshold.

4.4 Interchain contact information in pair representations

We test whether the pair representations learned for binder generation also support interchain contact prediction in reference complexes. On Heterodimer99, we evaluate the frozen model with complete reference sequences for both chains and either experimental or AF2-predicted target structures (Lin et al., 2023a). Further evaluation details are given in Appendix E.1.

Table 3: **Sequence–text alignment and predicted structure confidence on DB5.5.** Cosine, ipTM, pTM, and pLDDT are means; pLDDT uses a 0–1 scale. Evaluation protocols are given in Appendix D.

	ProTrek			Structure confidence		
Method	Cosine ↑	MRR ↑	Top-1 (%) ↑	ipTM ↑	pTM ↑	pLDDT ↑
DSM(ppi)	0.19	0.10	4.0	0.18	0.61	0.66
BoltzGen	0.17	0.14	8.7	0.36	**0.69**	**0.82**
Proteina-Complexa	0.15	0.15	8.7	0.35	0.68	**0.82**
PPDiff	0.21	0.19	13.5	0.20	0.62	0.67
FBindGen	**0.28**	**0.50**	**41.3**	**0.40**	**0.69**	0.78

Table 4: **Interchain contact prediction on Heterodimer99.** Precision (%) is averaged equally over complexes; L is the shorter complete sequence length. Bold indicates the highest value within each structure setting. † marks published results from Supplementary Data 2 of Liu et al. (2026), evaluated with method-specific inputs.

Structure	Method	Top-1	Top-10	Top-25	Top-L/10	Top-L/5	Top-L
Experimental	GLINTER†	27.27	25.86	23.72	26.45	23.38	17.31
Experimental	CDPred†	34.34	30.91	30.51	31.36	29.97	25.11
Experimental	DeepInter†	40.40	41.41	39.60	40.65	39.66	35.04
Experimental	PLMGraph-Inter†	49.49	46.16	41.70	43.33	41.14	30.99
Experimental	PPLM-Contact†	**62.63**	58.08	**56.85**	56.98	**56.52**	**48.94**
Experimental	FBindGen	**62.63**	**59.70**	54.10	**59.09**	53.66	35.66
AF2	GLINTER†	26.26	24.65	22.71	24.54	22.40	16.84
AF2	CDPred†	37.37	37.47	34.79	37.86	35.54	27.48
AF2	DeepInter†	42.42	38.18	37.49	38.55	36.98	32.94
AF2	PLMGraph-Inter†	38.38	36.26	32.28	35.24	31.89	25.04
AF2	PPLM-Contact†	56.57	54.04	52.36	53.61	52.39	**45.12**
AF2	FBindGen	**61.62**	**57.88**	**53.01**	**58.61**	**53.81**	34.25

FBindGen achieves the highest precision among the compared methods at Top-10 and Top-$L/10$ in both structure settings, with Top-10 precision of 59.70% for experimental targets and 57.88% for AF2-predicted targets (Table 4). This advantage does not extend to Top-L, where FBindGen trails PPLM-Contact with experimental structures (35.66% versus 48.94%) and AF2 structures (34.25% versus 45.12%). This pattern indicates that the pair representations are useful for prioritizing a short list of interchain contacts. We use probes on complete sequences to examine what interaction information the learned pair states capture. Appendix F provides an architectural comparison and an analysis of generation trajectories.

4.5 Sensitivity to functional conditioning

Contact prediction. We perturb binder annotations while keeping model weights, protein sequences, and target structures fixed. NONE removes the supplied annotations; position shuffling changes their residue-level placement while preserving total label counts; label permutation replaces annotation identities while preserving positions, coverage intervals, and per-residue label counts. Figure 3 reports Top-L and Top-1 precision; Appendix E.2 details the protocol.

Annotation removal and label permutation both reduce mean precision at Top-L and Top-1 in the two structure settings, with Top-L losses ranging from 3.56 to 4.21 percentage points (Figure 3). The loss after label permutation shows that annotation identities contribute information beyond their positions and coverage. Position shuffling leaves mean Top-L precision close to the reference but lowers mean Top-1 precision in both settings. This pattern is consistent with residue-level placement affecting the highest-ranked prediction, although the Top-1 seed ranges overlap the reference values.

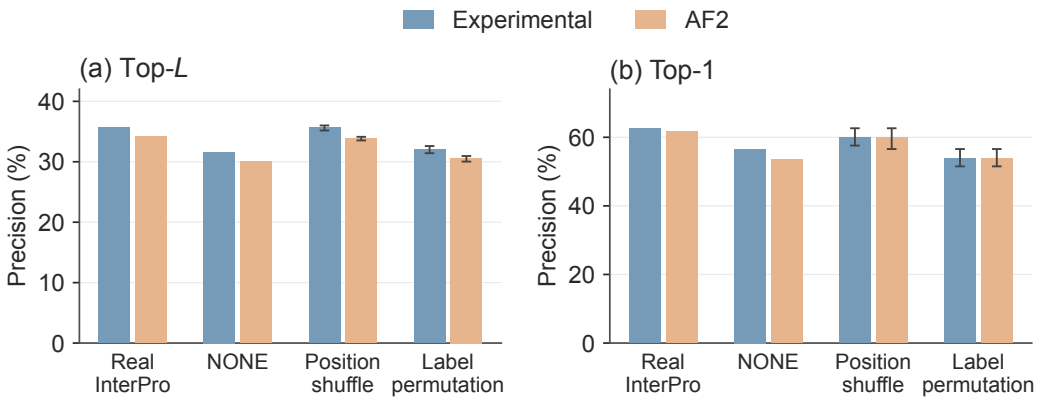

Figure 3: **Contact sensitivity to functional annotations on Heterodimer99.** Precision (%) is averaged equally over 99 complexes. Bars compare experimental and AF2 target structures. For position shuffling and label permutation, heights are three-seed means and error bars span the seed minimum and maximum.

Sequence generation. For each of 16 DB5.5 targets, we generate one binder with the original annotations and another with annotations from a different complex. Target inputs, binder length, generation seed, model weights, and decoding protocol are fixed, so the functional condition is the only changed input. ProTrek measures each sequence's similarity to both the original and replacement descriptions. Annotation replacement decreases mean similarity to the original description by 0.174 and increases mean similarity to the replacement description by 0.108 (Appendix D.3). These directions occur in 14 of 16 cases for the original description and 12 of 16 for the replacement description. The paired comparison therefore supports an effect of the supplied annotations on generated sequence–text alignment.

5 Limitations

Our evaluation focuses on benchmark settings with specified binder lengths and functional annotations. Extending the framework to more flexible annotation specifications is a direction for future work. Experimental validation of binding and enzymatic activity would further complement the computational evaluation.

6 Conclusion

We introduced FBindGen, a diffusion protein language model that generates binder sequences conditioned on a target protein and functional annotations. During denoising, sequence features update target-aware pair representations, which in turn guide sequence predictions. We also introduced EnzBind, an enzyme-focused benchmark for evaluating functional alignment and predicted interface confidence. On EnzBind and DB5.5, FBindGen achieves closer agreement with the requested functional descriptions and higher mean predicted interface confidence than the evaluated baselines. Paired generation experiments further show that changing the annotations can steer generated sequences toward new functional descriptions while keeping the target fixed.

AI Use Statement

ChatGPT and Codex were used to assist with language editing, paragraph and manuscript organization, and LaTeX formatting.

REFERENCES

Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J. Ballard, Joshua Bambrick, Sebastian W. Bodenstein, David A. Evans, Chia-Chun Hung, Michael O'Neill, David Reiman, Kathryn Tunyasuvunakool, Zachary Wu, Akvilė Žemgulytė, Eirini Arvaniti, Charles Beattie, Ottavia Bertolli, Alex Bridgland, Alexey Cherepanov, Miles Congreve, Alexander I. Cowen-Rivers, Andrew Cowie, Michael Figurnov, Fabian B. Fuchs, Hannah Gladman, Rishub Jain, Yousuf A. Khan, Caroline M. R. Low, Kuba Perlin, Anna Potapenko, Pascal Savy, Sukhdeep Singh, Adrian Stecula, Ashok Thillaisundaram, Catherine Tong, Sergei Yakneen, Ellen D. Zhong, Michal Zielinski, Augustin Žídek, Victor Bapst, Pushmeet Kohli, Max Jaderberg, Demis Hassabis, and John M. Jumper. Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature*, 630(8016): 493–500, 2024. doi: 10.1038/s41586-024-07487-w. URL https://doi.org/10.1038/s41586-024-07487-w.

Jasper Butcher, Rohith Krishna, Raktim Mitra, Rafael I. Brent, Yanjing Li, Nathaniel Corley, Paul T. Kim, Jonathan Funk, Simon Mathis, Saman Salike, Aiko Muraishi, Helen Eisenach, Tuscan Rock Thompson, Jie Chen, Yuliya Politanska, Enisha Sehgal, Brian Coventry, Odin Zhang, Bo Qiang, Kieran Didi, Max Kazman, Frank DiMaio, and David Baker. De novo Design of All-atom Biomolecular Interactions with RFdiffusion3. *bioRxiv*, 2025. doi: 10.1101/2025.09.18.676967. URL https://doi.org/10.1101/2025.09.18.676967.

Salvatore Candido, Thomas Hayes, Alexander Derry, Roshan Rao, Zeming Lin, Robert Verkuil, Bryan Wu, Jin Sub Lee, Elise S. Bruguera, Jehan A. Keval, Mykhailo Kopylov, John E. Pak, Wesley Wu, Neil Thomas, Samson Mataraso, Alvin Hsu, Ashton C. Trotman-Grant, Kilian Fatras, Allan dos Santos Costa, Rohil Badkundri, Halil Akın, Deniz Oktay, Jonathan Deaton, Elizabeth Montabana, Hrishita Sitwala, Yue Yu, Marius Wiggert, Dylan Alexander Carlin, Anthony W. Goering, Tomasz Blazejewski, McCullen Sandora, Michael Hla, Tina Z. Jia, Leon H. Kloker, Nicholas J. Sofroniew, Masatoshi Uehara, Jassi Pannu, Sharrol Bachas, Daniel S. Liu, Tom Sercu, and Alexander Rives. Language Modeling Materializes a World Model of Protein Biology. bioRxiv preprint, 2026. URL https://www.biorxiv.org/content/10.64898/2026.06.03.729735. Preprint.

J. Dauparas, I. Anishchenko, N. Bennett, H. Bai, R. J. Ragotte, L. F. Milles, B. I. M. Wicky, A. Courbet, R. J. de Haas, N. Bethel, P. J. Y. Leung, T. F. Huddy, S. Pellock, D. Tischer, F. Chan, B. Koepnick, H. Nguyen, A. Kang, B. Sankaran, A. K. Bera, N. P. King, and D. Baker. Robust deep learning-based protein sequence design using ProteinMPNN. *Science*, 378(6615):49–56, 2022. doi: 10.1126/science.add2187. URL https://doi.org/10.1126/science.add2187.

Kieran Didi, Zuobai Zhang, Guoqing Zhou, Danny Reidenbach, Zhonglin Cao, Sooyoung Cha, Tomas Geffner, Christian Dallago, Jian Tang, Michael M. Bronstein, Martin Steinegger, Emine Kucukbenli, Arash Vahdat, and Karsten Kreis. Scaling Atomistic Protein Binder Design with Generative Pretraining and Test-Time Compute. arXiv preprint arXiv:2603.27950v1, 2026. URL https://arxiv.org/abs/2603.27950v1. Version 1; architecture description refers to this version.

Roland L. Dunbrack. Rēs ipSAE loquuntur: What's wrong with AlphaFold's ipTM score and how to fix it. *bioRxiv*, 2025. doi: 10.1101/2025.02.10.637595. URL https://doi.org/10.1101/2025.02.10.637595.

Johnathan D. Guest, Thom Vreven, Jing Zhou, Iain Moal, Jeliazko R. Jeliazkov, Jeffrey J. Gray, Zhiping Weng, and Brian G. Pierce. An expanded benchmark for antibody-antigen docking and affinity prediction reveals insights into antibody recognition determinants. *Structure*, 29(6):606–621.e5, 2021. doi: 10.1016/j.str.2021.01.005. URL https://doi.org/10.1016/j.str.2021.01.005.

Xiao-Yu Guo, Yi-Fan Li, Yuan Liu, Xiaoyong Pan, and Hong-Bin Shen. Ab-initio amino acid sequence design from protein text description with ProtDAT. *Nature Communications*, 16: 10544, 2025. doi: 10.1038/s41467-025-65562-w. URL https://doi.org/10.1038/s41467-025-65562-w.

Logan Hallee, Nikolaos Rafailidis, David B. Bichara, and Jason P. Gleghorn. Diffusion Sequence Models for Enhanced Protein Representation and Generation. arXiv preprint arXiv:2506.08293, 2025. URL https://arxiv.org/abs/2506.08293.

Thomas Hayes, Roshan Rao, Halil Akin, Nicholas J. Sofroniew, Deniz Oktay, Zeming Lin, Robert Verkuil, Vincent Q. Tran, Jonathan Deaton, Marius Wiggert, Rohil Badkundri, Irhum Shafkat, Jun Gong, Alexander Derry, Raul S. Molina, Neil Thomas, Yousuf A. Khan, Chetan Mishra, Carolyn Kim, Liam J. Bartie, Matthew Nemeth, Patrick D. Hsu, Tom Sercu, Salvatore Candido, and Alexander Rives. Simulating 500 million years of evolution with a language model. *Science*, 387(6736):850–858, 2025. doi: 10.1126/science.ads0018. URL https://doi.org/10.1126/science.ads0018.

Philip Jones, David Binns, Hsin-Yu Chang, Matthew Fraser, Weizhong Li, Craig McAnulla, Hamish McWilliam, John Maslen, Alex Mitchell, Gift Nuka, Sebastien Pesseat, Antony F. Quinn, Amaia Sangrador-Vegas, Maxim Scheremetjew, Siew-Yit Yong, Rodrigo Lopez, and Sarah Hunter. InterProScan 5: genome-scale protein function classification. *Bioinformatics*, 30(9):1236–1240, 2014. doi: 10.1093/bioinformatics/btu031. URL https://doi.org/10.1093/bioinformatics/btu031.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, 2021. doi: 10.1038/s41586-021-03819-2. URL https://doi.org/10.1038/s41586-021-03819-2.

Peicong Lin, Huanyu Tao, Hao Li, and Sheng-You Huang. Protein–protein contact prediction by geometric triangle-aware protein language models. *Nature Machine Intelligence*, 5(11):1275–1284, 2023a. doi: 10.1038/s42256-023-00741-2. URL https://doi.org/10.1038/s42256-023-00741-2.

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

Jun Liu, Hungyu Chen, and Yang Zhang. A paired sequence language model for protein-protein interaction modeling. *Nature Communications*, 17(1):3733, 2026. doi: 10.1038/s41467-026-70457-5. URL https://doi.org/10.1038/s41467-026-70457-5.

Shengchao Liu, Yanjing Li, Zhuoxinran Li, Anthony Gitter, Yutao Zhu, Jiarui Lu, Zhao Xu, Weili Nie, Arvind Ramanathan, Chaowei Xiao, Jian Tang, Hongyu Guo, and Anima Anandkumar. A text-guided protein design framework. *Nature Machine Intelligence*, 7:580–591, 2025. doi: 10.1038/s42256-025-01011-z. URL https://doi.org/10.1038/s42256-025-01011-z.

Ali Madani, Ben Krause, Eric R. Greene, Subu Subramanian, Benjamin P. Mohr, James M. Holton, Jose Luis Olmos, Caiming Xiong, Zachary Z. Sun, Richard Socher, James S. Fraser, and Nikhil Naik. Large language models generate functional protein sequences across diverse families. *Nature Biotechnology*, 41(8):1099–1106, 2023. doi: 10.1038/s41587-022-01618-2. URL https://doi.org/10.1038/s41587-022-01618-2.

Martin Pacesa, Lennart Nickel, Christian Schellhaas, Joseph Schmidt, Ekaterina Pyatova, Lucas Kissling, Patrick Barendse, Jagrity Choudhury, Srajan Kapoor, Ana Alcaraz-Serna, Yehlin Cho, Kourosh H. Ghamary, Laura Vinué, Brahm J. Yachnin, Andrew M. Wollacott, Stephen Buckley, Adrie H. Westphal, Simon Lindhoud, Sandrine Georgeon, Casper A. Goverde, Georgios N. Hatzopoulos, Pierre Gönczy, Yannick D. Muller, Gerald Schwank, Daan C. Swarts, Alex J. Vecchio, Bernard L. Schneider, Sergey Ovchinnikov, and Bruno E. Correia. One-shot design of

functional protein binders with BindCraft. *Nature*, 646(8084):483–492, 2025. doi: 10.1038/s41586-025-09429-6. URL https://doi.org/10.1038/s41586-025-09429-6.

Zhenqiao Song, Tianxiao Li, Lei Li, and Martin Renqiang Min. PPDiff: Diffusing in Hybrid Sequence-Structure Space for Protein-Protein Complex Design. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 56319–56336. PMLR, 2025. URL https://proceedings.mlr.press/v267/song25d.html.

Hannes Stark, Felix Faltings, MinGyu Choi, Yuxin Xie, Eunsu Hur, Timothy O'Donnell, Anton Bushuiev, Talip Uçar, Saro Passaro, Weian Mao, Mateo Reveiz, Roman Bushuiev, Tally Portnoi, Tomáš Pluskal, Josef Sivic, Karsten Kreis, Arash Vahdat, Shamayeeta Ray, Jonathan T. Goldstein, Andrew Savinov, Jacob A. Hambalek, Anshika Gupta, Diego A. Taquiri-Diaz, Yaotian Zhang, Samuel J. Snyder, A. Katherine Hatstat, Angelika Arada, Nam Hyeong Kim, Haoyu Fan, Ethel Tackie-Yarboi, Dylan Boselli, Lee Schnaider, Chang C. Liu, Gene-Wei Li, Denes Hnisz, David M. Sabatini, William F. DeGrado, Jeremy Wohlwend, Gabriele Corso, Regina Barzilay, and Tommi Jaakkola. BoltzGen: Toward Universal Binder Design. *bioRxiv*, 2025. doi: 10.1101/2025.11.20.689494. URL https://doi.org/10.1101/2025.11.20.689494.

Jin Su, Xibin Zhou, Xuting Zhang, and Fajie Yuan. ProTrek: Navigating the Protein Universe through Tri-Modal Contrastive Learning. *bioRxiv*, 2024. doi: 10.1101/2024.05.30.596740. URL https://doi.org/10.1101/2024.05.30.596740.

Xinyou Wang, Zaixiang Zheng, Fei Ye, Dongyu Xue, Shujian Huang, and Quanquan Gu. Diffusion Language Models Are Versatile Protein Learners. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 52309–52333. PMLR, 2024. URL https://proceedings.mlr.press/v235/wang24ct.html.

Xinyou Wang, Zaixiang Zheng, Fei Ye, Dongyu Xue, Shujian Huang, and Quanquan Gu. DPLM-2: A Multimodal Diffusion Protein Language Model. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/57c30b677add9aa78e1745f0643104d0-Abstract-Conference.html.

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, Basile I. M. Wicky, Nikita Hanikel, Samuel J. Pellock, Alexis Courbet, William Sheffler, Jue Wang, Preetham Venkatesh, Isaac Sappington, Susana Vázquez Torres, Anna Lauko, Valentin De Bortoli, Emile Mathieu, Sergey Ovchinnikov, Regina Barzilay, Tommi S. Jaakkola, Frank DiMaio, Minkyung Baek, and David Baker. De novo design of protein structure and function with RFdiffusion. *Nature*, 620(7976):1089–1100, 2023. doi: 10.1038/s41586-023-06415-8. URL https://doi.org/10.1038/s41586-023-06415-8.

Junbo Yin, Chao Zha, Wenjia He, Chencheng Xu, and Xin Gao. CFP-Gen: Combinatorial Functional Protein Generation via Diffusion Language Models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 72485–72504. PMLR, 2025. URL https://proceedings.mlr.press/v267/yin25g.html.

Han Yu, Huaxiang Deng, Jiahui He, Jay D. Keasling, and Xiaozhou Luo. UniKP: a unified framework for the prediction of enzyme kinetic parameters. *Nature Communications*, 14(1):8211, 2023. doi: 10.1038/s41467-023-44113-1. URL https://doi.org/10.1038/s41467-023-44113-1.

Wensi Zhu, Aditi Shenoy, Petras Kundrotas, and Arne Elofsson. Evaluation of AlphaFold-Multimer prediction on multi-chain protein complexes. *Bioinformatics*, 39(7):btad424, 2023. doi: 10.1093/bioinformatics/btad424. URL https://doi.org/10.1093/bioinformatics/btad424.

A ARCHITECTURE, OPTIMIZATION, AND DECODING

A.1 REPRESENTATIONS AND CONDITIONING

Sequence and annotation representations. The sequence backbone follows the ESM-2 architecture and is initialized from the pretrained DPLM-150M checkpoint (Wang et al., 2024). We concatenate the target and binder sequences as `[BOS, target, EOS, BOS, binder, EOS]` and distinguish the two chains with learned chain embeddings. Target and special tokens remain unchanged during corruption. Pairwise features are computed over residue positions, excluding special tokens.

The annotation vocabulary contains 54,193 entries, including PAD (ID 0), NONE (ID 1), and UNK (ID 2); InterPro identifiers have IDs greater than 2. The PAD embedding is fixed at zero, while all other embeddings are learned. At positions with overlapping annotations, we sum the corresponding embeddings. A two-layer MLP with SiLU and a zero-initialized output projection maps these embeddings to the sequence input space, where its output is added after token dropout.

Pair representations and sequence–pair updates. Pair states are recomputed at every model forward pass. Let $h_i^{(0)}$ denote the input representation after adding chain and annotation embeddings and before the first Transformer layer. We apply LayerNorm and a learned linear projection to $h_i^{(0)}$ to obtain the residue feature s_i. The projections W_L and W_R map these residue features to pair features, which are combined with the relative-position encoding e_{ij}^{pos} and chain encoding e_{ij}^{chain} defined in Section 3.2.

Sequence and pair representations interact at positions 0, 10, 20, and 30, with two Pairformer blocks at each position. Pair features have 96 channels, projected residue features have 256 channels, and sequence-to-pair features have 64 channels. The first exchange combines initialized pair features with the sequence-to-pair update. At subsequent exchanges, the three updates defined in Section 3.3 are applied to valid residue pairs, and gradients propagate into the sequence backbone.

At update position $b = 0$, we use the input representations. At positions $b = 10, 20, 30$, we compute learned weighted sums of the hidden states from layers 1–9, 10–19, and 20–29, respectively. We denote these sets of layer indices by $\mathcal{S}_b$, with $\mathcal{S}_0 = \{0\}$ for the input representations. We add a projection of the binder annotation representation to these aggregated sequence features:

$$u_i^{(b)} = \sum_{\ell \in \mathcal{S}_b} \alpha_\ell^{(b)} W_b \,\mathrm{LN}_b(h_i^{(\ell)}) + R_b e_i^F, \qquad \alpha^{(b)} = \mathrm{softmax}(a^{(b)}),$$

where the annotation term $R_b e_i^F$ applies only to binder positions. We then project each aggregated representation to obtain $v_i^{(b)} = D_b u_i^{(b)} + c_b$.

Each exchange has a separate bias-free, zero-initialized projection for the annotation residual $R_b e_i^F$. It maps the summed InterPro embeddings to the 64-dimensional aggregated feature space and applies only to valid binder residues. We add this residual after hidden-layer aggregation and before sequence-to-pair conversion. Both conditioning paths use the same per-residue InterPro annotations and share the annotation embedding table.

At each noninitial exchange, we apply the same LayerNorm and linear projection used at initialization to the latest hidden state of each residue. We then sum the W_L projection of residue i and the W_R projection of residue j to obtain the corresponding entry of ΔZ_b^{res}. Within the Pairformer blocks, triangle multiplication and triangle attention update each pair (i, j) using pair features involving a third residue k, while pair transitions apply a feed-forward network independently to each pair.

At each noninitial exchange, the attention-based update concatenates the preceding layer's attention probabilities for both directions of each residue pair, yielding $2N_h$ features for N_h attention heads. These features pass through LayerNorm and a two-layer MLP with a 16-dimensional hidden layer, GELU activation, and a 96-dimensional output. A learned scalar sigmoid gate scales the update. Each exchange has separate parameters; the gate logit is initialized to -1, and the output projection's weights and bias are initialized to zero.

Timestep-dependent attention biases. The pair-to-sequence attention bias is scaled by $\gamma_b(t) = \mathrm{sigmoid}(\beta_b + w_b^\top e(t))$, a scalar gate shared across attention heads and residue pairs within each

example. To obtain $e(t)$, we encode $\frac{1}{4}\log(t/T)$ with 16 fixed random Fourier features, followed by LayerNorm, a bias-free linear projection to 16 dimensions, and SiLU. This timestep embedding is shared across exchanges. Each exchange has its own β_b and w_b, initialized to -2.5 and zero, respectively.

Structural conditioning and prediction heads. The target-distance encoder uses 39 bins spanning 3.25–50.75 Å. Local frames require available N, Cα, and C atoms, whereas distance labels require a valid Cα coordinate. We provide target structure features only for target–target pairs; binder coordinates and reference interchain distances are used only as supervision. The amino acid head predicts residue distributions from the final sequence states for sequence sampling. Independent TB and BB heads provide auxiliary distance distributions from the final pair states.

A.2 Optimization and auxiliary supervision

Optimization. We optimize the pretrained backbone and added modules jointly with AdamW, using learning rates 5×10^{-5} and 10^{-4}, respectively. We use $\beta = (0.9, 0.98)$, weight decay 10^{-4}, and a cosine learning-rate schedule with 2,000 warmup steps and a 500,000-step horizon. Each device processes batches of at most 1,280 tokens, with at most 576 combined residues per sample. There is no additional limit on the total number of residue pairs in a batch.

Sequence reconstruction. We use an absorbing-state masking process and sample t_n uniformly from $\{1, \ldots, T\}$, with $T = 500$, for each training example n. At timestep t_n, each binder residue is independently replaced by the mask token with probability t_n/T, leaving target and special tokens unchanged. Let M_n denote the set of masked binder positions in training example n. We compute the sequence reconstruction loss over these positions:

$$\mathcal{L}_{\text{seq}} = -\frac{\sum_n w(t_n) \sum_{i \in M_n} \log p_\theta(x^B_{n,i} \mid x^B_{n,t_n}, x^T_n, C^T_n, F^B_n, L_{B,n}, t_n)}{\sum_n |M_n|}, \qquad w(t) = \frac{T - t + 1}{T}.$$

The timestep weighting downweights highly corrupted examples, whereas generation starts from a fully masked binder. We therefore include a reconstruction term $\mathcal{L}_{\text{full}}$ for fully masked binders. This term uses the same masked-residue cross-entropy at $t = T$, with all binder residues masked and the timestep weight fixed to 1.

Distance labels and losses. The loss coefficients in Section 3.4 are $\lambda_F = 0.05$, $\lambda_{TB} = 1.0$, and $\lambda_{BB} = 0.25$. For coarse distance prediction, we use three Cα-distance classes: contact ($d \leq 8$ Å), nearby ($8 < d \leq 12$ Å), and distant ($d > 12$ Å), weighted by 0.50, 0.35, and 0.15, respectively. Fine distance prediction uses 12 bins of width 0.5 Å over 2–8 Å, with Gaussian targets of standard deviation 0.5 Å centered on reference distances clipped to 2–8 Å. The TB head receives the mean of the two directional pair representations.

For each coarse class, we sum the timestep-weighted cross-entropies over valid pairs across all devices in the current batch, then normalize by the class's unweighted pair count. We combine the class losses using the weights above, renormalized over nonempty classes. We compute the fine loss by summing timestep-weighted KL divergences over valid pairs whose reference Cα distance is at most 8 Å, then dividing by their unweighted count. An empty pair set contributes zero. The sequence loss also sums masked-residue contributions across devices and normalizes by the unweighted number of masked residues. The fully masked reconstruction term uses unit timestep weights at $t = T$ and enters the objective with coefficient λ_F.

A.3 Confidence-adaptive decoding

Sequence completion. We initialize all designable binder positions with mask tokens and keep target and special tokens fixed. The reference binder sequence provides the binder length and reference metadata. We complete the sequence in stages $f \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$, assigning residues to $\lceil f L_B \rceil$ positions by the end of each stage. Each stage allows at most 32 iterations; a minimum number of assignments per iteration ensures that the stage finishes within this limit. At each iteration, the number of masked positions $|M|$ determines the denoising timestep:

$$t = \text{clip}(\lceil T|M|/L_B \rceil, 1, T).$$

We convert the annotation-conditioned logits to 32-bit floating-point precision and restrict the vocabulary to the 20 standard amino acids. Composition penalties adjust both the residue proposals and their acceptance priorities.

Composition control. For a threshold of 0.25, we assign each amino acid the integer quota

$$Q = \max(1, \lfloor 0.25 L_B \rfloor).$$

Let n_a be the number of assigned residues of type a. We calculate the proposed count relative to the quota and its additive logit penalty as

$$u_a = \frac{n_a + 1}{Q}, \qquad b_a = -3.0 \, \frac{\max(u_a - 0.5, 0)}{1 - 0.5}.$$

The penalty begins when a proposed count exceeds half the quota and reaches -3.0 nats at the quota. We exclude an amino acid from new proposals once its assigned count reaches the quota. If multiple positions propose the same amino acid, we accept the highest-priority proposals that fit within its remaining quota. The last permitted iteration of each stage bypasses this simultaneous-assignment check to complete the stage. The resulting sequence may therefore exceed the quotas; its composition is checked after refinement.

Proposal sampling and acceptance. Let q_i denote the annotation-conditioned amino acid distribution at position i, after vocabulary restriction and composition penalties but before sampling noise. We use Gumbel-max sampling with noise scale 1.0 and clip uniform random values to $[10^{-7}, 1 - 10^{-7}]$ before the Gumbel transform. We classify a proposal as high-confidence if it matches $\arg\max_a q_i(a)$, has probability at least 0.75, and exceeds the second-highest probability by at least 0.20.

With $|M|$ masked positions, the nominal acceptance count ranges from $\lceil 0.1|M| \rceil$ to $\lceil 0.5|M| \rceil$, with a minimum of one. The first fully masked iteration instead uses an upper bound of $\lceil 0.1|M| \rceil$. If N further assignments are required within R remaining iterations, the progress lower bound is $\lceil N/R \rceil$, capped at N. We first clip the number of high-confidence proposals to the regular acceptance range, then increase it to the progress lower bound if needed and cap it at N. The progress bound can therefore override the regular acceptance bounds.

After determining the acceptance count, we rank positions by the proposed amino acid's log-probability, adding 1000 for high-confidence proposals. We apply the quota-limited acceptance rule to the selected positions, leaving rejected positions masked. Accepted amino acids remain fixed until the completion phase ends. Each completion stage ends with the required number of assigned positions.

Local refinement. At the start of each refinement round, we evaluate the complete sequence with no masked positions, composition penalties, or sampling noise. We use the entropy of each position's predicted amino acid distribution as a heuristic selection score:

$$H_i = -\sum_a p_i(a) \log p_i(a).$$

We remask the $\lceil 0.1 L_B \rceil$ highest-entropy positions and resample their residues with noise scale 1.0. We set the denoising timestep from the actual fraction of remasked positions, which typically gives a value near 50.

We accept proposed replacements according to amino acid quotas, without requiring an improvement in sequence score. Quota checks use the amino acid counts at the start of the round and may reject substitutions that preserve the counts. Refinement ends after at most two rounds, or earlier when no residue changes and no amino acid exceeds 35% of the sequence. The final composition check uses a 25% threshold.

A.4 EnzBind generation configuration

For EnzBind, we use the FBindGen checkpoint saved after 100,000 training steps and generate sequences with the annotation-conditioned branch alone, using the default inference settings. We generate one binder sequence for each of the 275 target complexes, conditioned on the target inputs, binder length, and per-residue InterPro annotations. We evaluate these sequences with fine-tuned ProTrek, InterProScan, UniKP, and ESMFold2, following the procedures in Appendix C.

B Data construction and partitioning

B.1 Sources and structural filtering

The training set comprises 497,581 interacting chain pairs with at most 576 combined residues: 464,637 from experimental PDB complexes and 32,944 from predicted AlphaFold heterodimers. The validation set contains 6,972 PDB-derived chain pairs after length filtering. Each chain pair represents an interface, and one complex may contribute several interfaces. We sample training examples with replacement and with probability inversely proportional to interface-cluster size.

We preprocess PDB structures with OpenFold3 and extract chain pairs from the resulting interface metadata. We assign structures released by June 30, 2025, to training and those released from July 1, 2025, onward to validation. Eligible entries contain at least two protein chains of at least 10 residues, at least two distinct protein entities, and at least one protein–protein interface. We retain each distinct unordered chain pair annotated as an interface. Because the requirement for distinct entities applies to the whole entry, an eligible assembly may also contribute interfaces between copies of the same entity. For training, we accept structures with reported resolution of at most 9 Å or no reported resolution; for validation, we require reported resolution of at most 4.5 Å.

B.2 Chain assignment and coordinate representation

During training, we assign either chain in each sampled pair as the binder with equal probability, with the other serving as the target. This assignment is resampled each time a pair is drawn, and functional conditioning uses the annotations of the selected binder. During validation and evaluation, target–binder assignments remain fixed.

For each chain pair, we retain the complete sequences, available N, Cα, C, and O coordinates, and masks indicating atom availability. Residues without coordinates remain in the sequences. We exclude chain pairs with more than 576 combined residues.

B.3 Annotation mapping

For the primary PDB source, we map SIFTS InterPro cross-references and regions to binder sequences through PDBe residue indices. Exact sequence matches take precedence over alignment-based mappings. An alignment-based mapping must have identical non-X residues at 95% or more of the query positions and unambiguous annotation assignments. Mismatched or unmapped positions remain unannotated, while mappings with conflicting residue identities or out-of-range regions are rejected.

AlphaFold samples use sequence-validated, UniProt-linked InterPro regions. Across all sources, we use the same 54,193-entry vocabulary and retain overlapping annotations. Annotations condition only the binder, and samples without annotations use NONE. Binder annotations are available for 432,682 of the 464,637 PDB samples that satisfy the length limit.

B.4 Clustering and partitions

We exclude from the training set all samples whose PDB identifiers occur in DB5.5 or Heterodimer99. We cluster chains with MMseqs2 at 40% identity and 80% coverage in both directions, with sensitivity 8. We define an interface cluster as the unordered pair of its chain-cluster identifiers. We retain all eligible training interfaces. For validation, we select one sample per interface cluster

using seed 42, preferentially choosing pairs with at most 512 combined residues. Applying the final 576-residue limit gives 6,972 validation samples.

B.5 EnzBind benchmark construction

EnzBind contains 275 enzyme-associated complexes from PDB structures released after July 1, 2025, with combined target–binder sequence lengths below 600 residues. We obtain functional annotations using InterProScan. The enzyme-annotated chain serves as the binder and its interaction partner as the target. All enzymes included in EnzBind are excluded from the pretraining validation set.

C EnzBind evaluation and kinetic predictions

C.1 Designs and structure prediction

We evaluate the 275 binders generated with the configuration in Section A.4. The reference binder amino acid sequence is excluded from generation inputs.

For BoltzGen and Proteina-Complexa, we match the available designs to the 275 EnzBind cases by case identifier without applying an additional length filter. All design summaries include generated binders only; native sequences are evaluated separately as a reference.

We predict complex structures with ESMFold2 using three loops, 50 sampling steps, and one diffusion sample. The reported baseline configurations use these same three settings. Sequence–text alignment and InterPro annotation recovery are described in Sections C.3 and C.4.

C.2 Kinetic predictor and inputs

We use the pretrained UniKP models to predict kinetic properties from generated sequences and substrate SMILES (Yu et al., 2023). UniKP combines mean-pooled ProtT5-XL sequence embeddings with SMILES Transformer substrate embeddings. Separate ExtraTrees regressors predict turnover number k_{cat}, the Michaelis constant K_M, and catalytic efficiency k_{cat}/K_M on a base-10 logarithmic scale. We convert each prediction ℓ to its linear-scale value as 10^{ℓ} in the reported units.

We pair each generated binder with every substrate SMILES available for its case. Each method is evaluated on 321 case–substrate pairs from 255 complexes, covering 79 unique substrate SMILES; the remaining 20 complexes lack substrate SMILES and are excluded.

We run UniKP with PyTorch 2.1.0+cu121, scikit-learn 1.5.1, and transformers 4.46.1. Evaluated sequences contain 15–298 residues, below the 1,000-residue truncation threshold. Preprocessing replaces U, Z, O, and B with X and truncates SMILES exceeding 218 tokens. Some substrate strings in the evaluation reach this truncation limit.

Figure 2 reports the arithmetic mean predicted catalytic efficiency over the 321 case–substrate pairs for each method. Complexes with multiple substrates therefore contribute more than one observation.

C.3 Sequence–text evaluation

The sequence–text evaluation in Table 1 compares seven generators and a native reference on the same 275 cases.

Sequence and text representations. We use ProTrek-650M UniRef50 (Su et al., 2024), fine-tuned on the InterPro dataset, for all ProTrek evaluations on EnzBind and DB5.5. We L2-normalize both sequence and text embeddings. Each functional description consists of the reference binder's InterPro entry names, separated by semicolons. The retrieval set contains one entry per case, giving 275 entries with 145 distinct descriptions. We compute cosine similarity as the dot product of the normalized embeddings, without temperature scaling or clipping. Mean cosine similarity averages the scores between generated sequences and their paired descriptions.

Retrieval metrics. MRR is the mean reciprocal rank of the paired case entry. We define this rank as one plus the number of entries with strictly higher computed scores, so only exact floating-point ties receive the best rank. Text Top-1 accuracy measures how often the highest-scoring entry has the reference description, including entries from other cases with the same text. We report this accuracy as a percentage in Table 1. Text macro-F1 is the unweighted mean classification F1 across the 145 distinct description classes. R@5 and R@10 use the rank of the paired case entry.

C.4 InterPro annotation recovery

InterProScan annotations. For Table 2, we evaluate the same 275 binder sequences per method as in the ProTrek evaluation, with native binders as a reference. We run InterProScan 5.78-109.0 on all 2,200 sequences from the seven generators and the native reference, retaining only hits with integrated InterPro identifiers. The supplied annotations comprise 911 case–identifier assignments across 384 distinct InterPro identifiers. For each sequence, we compare the detected identifiers with the supplied annotation set.

Annotation matching. For each case, an identifier is a true positive if it is both supplied and detected, a false positive if it is detected but not supplied, and a false negative if it is supplied but not detected. We pool these counts across the 275 cases and compute micro-precision as $TP/(TP + FP)$, micro-recall as $TP/(TP + FN)$, and micro-F1 as $2TP/(2TP + FP + FN)$. False positives include detected identifiers outside the set of 384 supplied identifiers. Macro-F1 is the unweighted mean of the F1 scores computed separately for these 384 identifiers. The native reference yields 911 true positives and no false positives or false negatives, giving micro-F1 and macro-F1 of 1.00.

Annotation ranking. We score each sequence–identifier pair by the largest $-\log_{10}(E)$ among its reported hits, flooring E at 10^{-300} and assigning zero to undetected identifiers. For each supplied identifier, we compute average precision and ROC-AUC across the 275 sequences using scikit-learn. We report the unweighted mean over identifiers with both positive and negative cases.

D DB5.5 binder-design evaluation

D.1 Design inputs and structure prediction

We evaluate binder design on the 127 two-chain cases in the 165-case DB5.5 source collection, excluding 38 multichain cases. The reference binder amino acid sequence is excluded from generation inputs. All methods receive the same target sequences and matched binder lengths. We generate binder sequences with FBindGen and predict the complexes with ESMFold2-Fast using three loops, 50 sampling steps, and one sample.

Table 5: **Interface-confidence threshold results on DB5.5.** Counts and percentages are over 127 generated complexes per method.

Method	ipTM ≥ 0.8, n (%)
DSM(ppi)	1 (0.8)
BoltzGen	12 (9.4)
Proteina-Complexa	14 (11.0)
PPDiff	1 (0.8)
FBindGen	**27 (21.3)**

D.2 Sequence–text evaluation

For sequence–text evaluation, we use the generated sequences in Table 3. We exclude complex 2ABZ for all methods because its integrated InterPro entry names are unavailable, leaving 126 cases and 756 sequences across five generators and the native reference.

We use the same ProTrek-650M UniRef50 model fine-tuned on the InterPro dataset as in the EnzBind evaluation (Section C.3). We calculate cosine similarity from L2-normalized sequence

and text embeddings without temperature scaling or clipping. We form each description by concatenating complete InterPro entry names in accession order. The rank of the paired description is one plus the number of descriptions with a strictly higher score; ties receive the best rank. We use this rank to compute Top-1 accuracy and mean reciprocal rank (MRR). Eight native sequences contain X or Z residues, which we retain; generated sequences contain standard amino acids.

Table 6: **Sequence–text alignment on DB5.5.** All methods use the same annotated cases, with native sequences as a reference.

Method	Mean cosine	Median cosine	Top-1 (%)	MRR
Native reference	0.37	0.38	71.4	0.82
BoltzGen	0.17	0.15	8.7	0.14
Proteina-Complexa	0.15	0.15	8.7	0.15
PPDiff	0.21	0.20	13.5	0.19
DSM(ppi)	0.19	0.20	4.0	0.10
FBindGen	**0.28**	**0.34**	**41.3**	**0.50**

D.3 Functional annotation replacement

Paired generation. We evaluate the effect of replacing functional annotations on generated sequence–text alignment using 16 annotated, two-chain DB5.5 complexes with standard amino acids. Cases are sampled uniformly across total complex length, with distinct InterPro accession sets. Target lengths range from 61 to 296 residues, and binder lengths from 51 to 203 residues. For each case, we generate one binder using its original annotations and one using annotations from a paired case, keeping the target inputs, binder length, generation seed, model weights, and decoding protocol fixed. This comparison comprises 32 generated sequences.

Annotation replacement. We sort the case identifiers and pair each case with the next, wrapping the last back to the first: 1D6R, 1E96, 1EFN, 1FFW, 1FLE, 1GCQ, 1JTG, 1KAC, 1PVH, 1UDI, 1YVB, 2O8V, 2PCC, 3AAD, 3SGQ, and 5E5M. For an anchor binder of length L and a paired binder of length L', position r receives the paired annotations at position $\lfloor rL'/L \rfloor$, using zero-based indices $r = 0, \ldots, L-1$.

Sequence–text evaluation. Each description concatenates the InterPro entry names supplied in that condition, ordered by accession and separated by semicolons. We use ProTrek to compare both generated sequences against both descriptions using sequence–text cosine similarity.

Let B_F and $B_{F'}$ denote sequences generated with the original and replacement annotations, and let d_F and $d_{F'}$ denote their respective descriptions. With $s(B, d)$ denoting ProTrek cosine similarity, we compute paired changes for each description:

$$\delta_{\text{original}} = s(B_{F'}, d_F) - s(B_F, d_F), \tag{1}$$

$$\delta_{\text{replacement}} = s(B_{F'}, d_{F'}) - s(B_F, d_{F'}). \tag{2}$$

A negative δ_{original} indicates movement away from the original description; a positive $\delta_{\text{replacement}}$ indicates movement toward the replacement description.

Table 7: **ProTrek changes after functional annotation replacement.** Means are over 16 paired cases. Each change compares the replacement-conditioned sequence with the original-conditioned sequence against the same description.

Description	Mean cosine change	Cases in the indicated direction
Original	-0.174	14/16 decrease
Replacement	$+0.108$	12/16 increase

Table 7 shows that replacing annotations generally shifts sequence–text alignment away from the original description and toward the replacement description. These measurements assess changes in embedding alignment.

E COMPLETE-SEQUENCE CONTACT PREDICTION

E.1 INPUTS, REFERENCE CONTACTS, AND COMPARISONS

For the contact comparison in Table 4, we evaluate FBindGen with complete target and binder reference sequences at $t = 1$, using bfloat16 mixed precision and no sequence decoding. The model is conditioned on the target structure and binder InterPro annotations; binder coordinates and native interchain distances are not provided as inputs. The annotation-perturbation and contact-readout analyses also use complete reference sequences. InterProScan annotates 95 of the 99 binder sequences, yielding 542 intervals with integrated InterPro annotations (Jones et al., 2014). The four unannotated sequences use NONE, and all observed InterPro identifiers are in the model vocabulary.

For experimental and AF2 target structures, we use the same chain assignments and the sequences supplied with each benchmark setting. The AF2-associated sequences for 6m33-A and 6fqa-B have the same lengths as their experimental counterparts but differ in amino acid sequence; we use the supplied sequences in each case. Native binder coordinates are used only to define reference contacts, and AF2 binder coordinates are not provided to the model.

The published baseline values in Table 4 come from Supplementary Data 2 of Liu et al. (2026).

E.2 ANNOTATION PERTURBATIONS

We perturb the binder annotations while keeping FBindGen's weights, protein sequences, and target structures fixed. Position shuffling permutes the complete annotation sets among residues within each binder, preserving total label counts. Label permutation applies a derangement to the 230 observed InterPro identifiers while preserving positions, coverage intervals, and label counts per residue. NONE and PAD remain unchanged.

We use seeds 11, 22, and 33 for both randomized controls. Position shuffling alters the annotations of 89 of the 99 sequences, changing an average of 29.3% of the per-residue annotation sets; the four unannotated sequences are unaffected. For each seed, we average precision equally over all 99 complexes. Figure 3 shows the mean across seeds and the minimum–maximum range for each randomized control.

F ARCHITECTURAL COMPARISON AND GENERATION DIAGNOSTICS

F.1 CONTACT PROBING WITHOUT EXPLICIT PAIR REPRESENTATIONS

We test whether interchain contacts can be recovered from the cross-attention maps of a protein-complex generation model without explicit pair representations. This separately trained conditional DPLM generates binder sequences conditioned on the target sequence, using a diffusion decoder that cross-attends to embeddings from a frozen ESM encoder. The decoder has no Pairformer blocks, pair-derived attention biases, or distance heads.

We compare three contact readouts on Heterodimer99: mean attention, the mean of attention channels after average product correction (APC), and a trained linear probe (Table 8). All three use the same frozen backbone, allowing us to assess whether correcting or learning to combine its attention channels improves contact recovery over simple averaging.

Probe fitting. After training the conditional DPLM, we freeze the decoder and fit a linear probe that combines its 600 APC-corrected cross-attention channels (30 layers and 20 heads). The probe has 601 trainable parameters. To construct homology-aware partitions, we link complexes if a chain pair has at least 30% identity and 80% coverage in both directions, or if the complexes share a PDB identifier or an identical sequence. Connected components define the homology-linked groups. Excluding 82 DB5.5 complexes linked to Heterodimer99 leaves 66 training and 17 validation complexes for fitting the probe.

Test labels are excluded from probe fitting, normalization, and model selection. We select the epoch-3 probe by validation Top-L precision and use its logits to rank test residue pairs. All three readouts are evaluated on the 99 Heterodimer99 complexes using the heavy-atom contact criterion.

Table 8: **Contact prediction without explicit pair representations on Heterodimer99.** Precision (%) is averaged equally over complexes, with L defined as the length of the shorter complete chain. All three readouts use the same frozen backbone. APC denotes average product correction.

Contact predictor	Top-1	Top-10	Top-L/10	Top-L/5	Top-L
Mean attention	2.020	2.320	2.530	2.420	1.740
Mean APC-corrected attention	1.010	1.310	1.460	1.850	1.430
Trained linear predictor	2.020	1.410	1.510	1.630	1.550

Mean attention matches or exceeds both the APC-corrected mean and the trained linear probe at every reported cutoff (Table 8). At Top-L, the probe reaches 1.550% precision, compared with 1.740% for mean attention. The paired difference is -0.190 percentage points, with a 95% confidence interval of $[-0.850, 0.460]$, computed from 10,000 bootstrap resamples of complexes with seed 42. All three readouts have lower precision than FBindGen's contact head (Table 4).

F.2 Generation with the alternative architecture

We next evaluate binder generation with this alternative architecture. We generate four designs for each of 165 DB5.5 complexes. Decoding uses 500 steps and a temperature of 1.0, following a separate protocol from the confidence-adaptive procedure in Section A.3.

We predict structures with ESMFold2-Fast using three loops, 50 sampling steps, and one sample, without MSAs or templates. For multichain complexes, structure prediction includes all chains, while ipSAE and pDockQ2 are computed only for target–binder pairs (Dunbrack, 2025; Zhu et al., 2023). The resulting structure predictions have a mean pLDDT of 78.1 on a 0–100 scale, a mean ipTM of 0.277, and a mean ipSAE of 0.016.

F.3 Full-model trajectory example

For FBindGen, we examine how pair features and contact predictions change as binder residues are assigned during a single generation trajectory (Figure 4). The example uses the 9xu3 V–W complex, with a 91-residue target and an 80-residue binder.

We record states at $t = 500, 300, 175, 1$, containing 0, 32, 52, and 80 assigned binder residues, respectively. At each state, pair features are recomputed from the current tokens and timestep. We measure differences from the completed design at $t = 1$ using two quantities: absolute contact-probability differences (ΔP) and normalized pair-feature differences (ΔZ). To compute ΔZ, we take the channel-wise RMS difference in the pair features used by the distance heads and divide it by the global RMS of the completed-design representation.

At the fully masked input, mean ΔP is 0.0045 over the full pair map, but individual residue pairs differ from the completed-design predictions by as much as 0.47. This average includes the relatively stable target–target block and therefore does not measure changes in binder-associated pairs alone.

Mean ΔZ decreases from 0.111 at the fully masked input to 0.046 at $t = 175$. The differences form row and column bands in blocks involving binder residues. Because token context and timestep vary together, these changes reflect their joint variation during decoding.

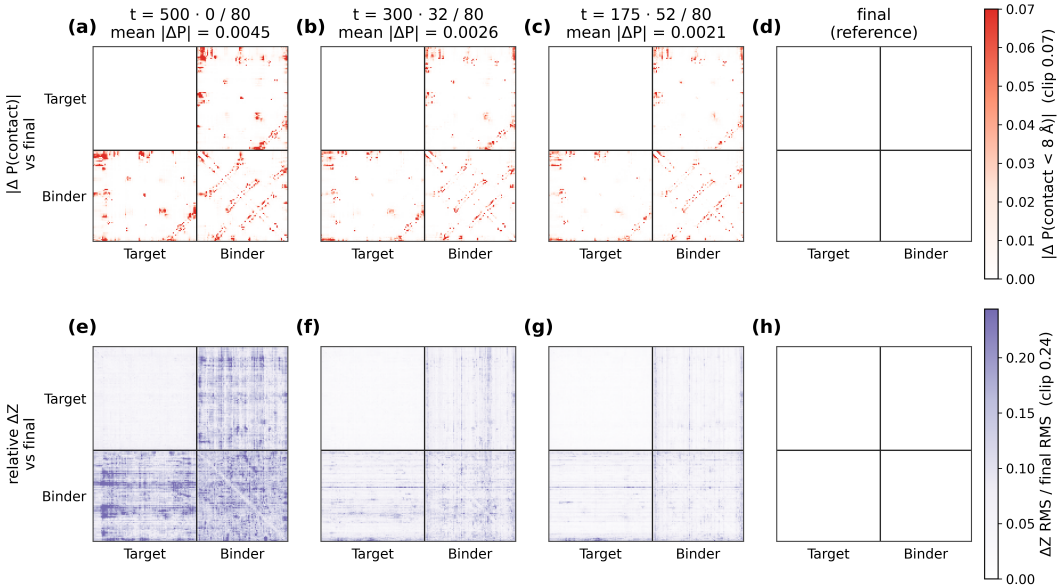

Figure 4: **Evolution of contact predictions and pair representations during generation.** Columns show the fully masked binder, two intermediate states, and the completed design used as the reference. (a–d) Absolute contact-probability differences ΔP. (e–h) Normalized pair-representation differences ΔZ. Black lines separate target and binder blocks; white denotes zero difference. Color scales are capped at 0.07 and approximately 0.24, corresponding to the 99th and 99.5th percentiles of the intermediate-stage values, respectively.