Generalise or Memorise? Benchmarking Ligand-Conditioned Protein Generation from Sequence-Only Data

Anonymous Authors[1]

Abstract

Proteins can bind small molecules with high specificity. However, designing proteins that bind user-defined ligands remains a challenge, typically relying on structural information and costly experimental iteration. While protein language models (pLMs) have shown promise for unconditional generation and conditioning on coarse functional labels, instance-level conditioning on a specific ligand has not been evaluated using purely textual inputs. Here we frame small-molecule protein binder design as a sequence-to-sequence translation problem and train ligand-conditioned pLMs that map molecular strings to candidate binder sequences. We curate large-scale ligand–protein datasets (>17M ligand-protein pairs) covering different data regimes and train a suite of models, spanning 16 to 700M parameters. Results reveal a consistent trade-off driven by supervision ambiguity: when each ligand is paired with few proteins, models generate near-neighbour, foldable sequences; when each ligand is paired with many proteins, generations are more diverse but less consistently foldable. Our study exposes how annotation diversity and sampling choices elicit this behaviour and how it changes with the data distribution. These insights highlight dataset redundancy and incompleteness as key bottlenecks for sequence-only binder design. We release the curated datasets, trained models, and evaluation tools to support future work on ligand-conditioned protein generation.

1. Introduction

Proteins can sense small molecules with exceptional sensitivity, initiating signaling cascades, transferring information, modulating the activity of other proteins, or catalyzing reactions that generate bioactive compounds and polymers. This functional diversity has long fascinated scientists and inspired efforts to engineer artificial proteins that recognize user-defined ligands. Yet, despite major progress, designing small-molecule binders remains exceptionally difficult: the protein must fold correctly and present side chains in a precise geometry that allows ligand access and stable binding, with even subtle deviations often abolishing affinity. Consequently, most successful strategies begin with weakly binding scaffolds and enhance performance through laboratory-directed evolution, aside from a few notable exceptions(Kortemme, 2024).

In recent years, protein design has been transformed by methods that leverage large-scale data and machine learning (ML) architectures. In this context, protein language models (pLMs) have shown strong performance in generating proteins that explore sequence space while retaining natural-like properties, all without the need for structural annotation. These advances have enabled the design of diverse proteins and enzymes, including lysozymes(Madani et al., 2023), triose-phosphate isomerases(Romero-Romero et al., 2024), malate dehydrogenases(Johnson et al., 2025), or nucleases(Ivančić et al., 2025). Together, these studies illustrate the ability of pLM-based approaches to expand natural protein family repertoires with artificial sequences. However, pLMs have not yet been explored for small-molecule binder design, and most state-of-the-art pLMs are not trained as generators conditioned on specific functions. Crucially, when conditioning is supported, it has been typically limited broad labels (e.g., taxonomy tags(Madani et al., 2020) or EC/GO classes(Munsamy et al., 2024)) and training set instances. This setup does not allow generalisation to user-defined molecules or target definitions at the molecular level, which will be crucial to fully reach user-defined controllable protein design. Therefore, it is crucial to evaluate generators with respect to a specific target instance.

A widely successful strategy in NLP for such conditioning is to frame diverse tasks as text-to-text (sequence-to-sequence) problems, where an encoder transforms the input into a contextual representation and a decoder generates an output

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

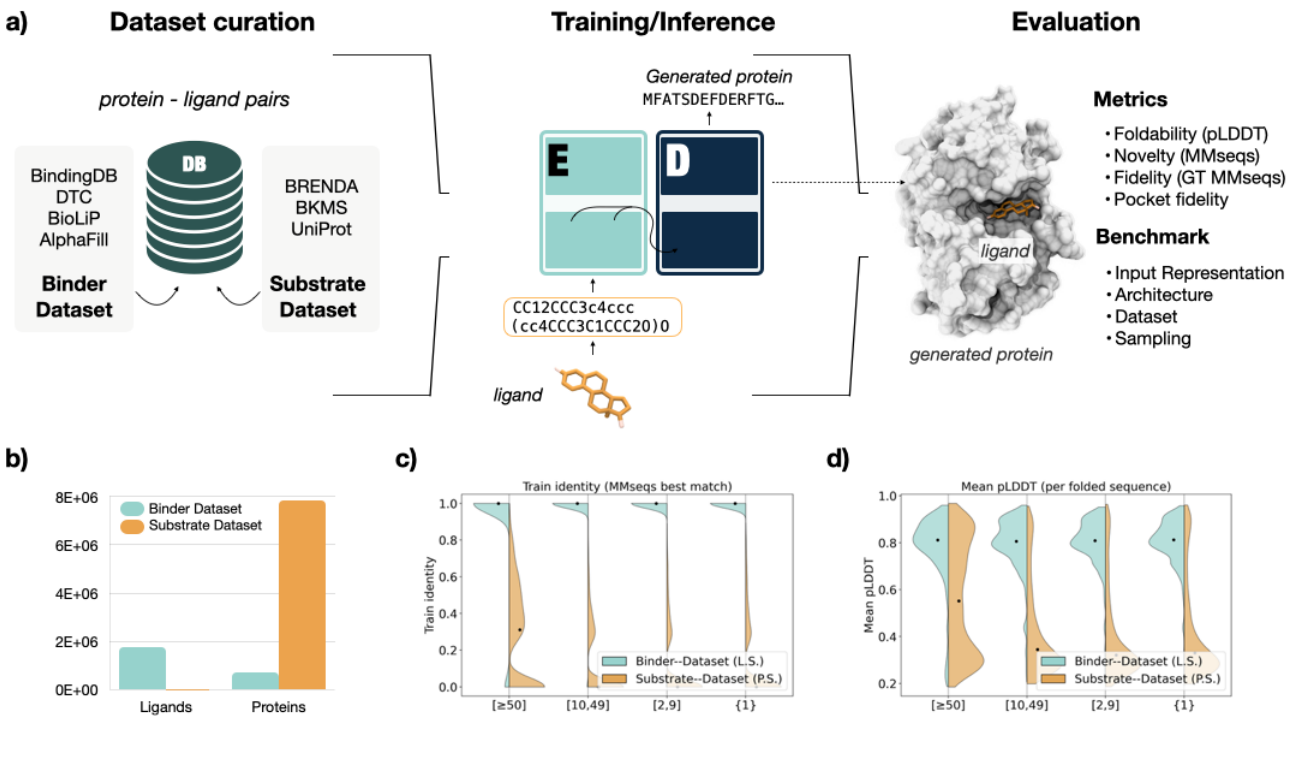

Figure 1. **(a)** Pipeline Summary **(b)** Dataset sizes **(c)** Protein novelty for the $[\geq 50]$ train evaluation split for the Ligand–Sampling model trained on the Binder–Dataset and Pair–Sampling model trained on the Substrate–Dataset, representing the two extremes of the distribution **(d)** pLDDT evaluation for the same models and split.

sequence conditioned on it. This paradigm has also inspired protein models that generate sequences conditioned on user-defined structures (Dauparas et al., 2022). However, protein design conditioned on a small molecule or substrate has not been explored to date. To address this gap, we provide a comprehensive perspective on the performance of brute-force protein language modeling for small-molecule binding design.

Here, we curate several ligand–protein datasets (with $>$ $17M$ ligand-protein pairs) and develop several models (from $16M$ to $700M$ parameters) which take a target ligand as input and generate a binder-protein as output. We evaluate generalisation across data regimes, and benchmark architectures, model sizes, sampling schemes, input representations, and a pre-training strategy. We analyse generalisation across data regimes and observe a trade-off: settings with a low molecule/sequence ratio favour enhanced diversity during generation but reduce foldability, whereas the opposite yields a "retrieval-like behaviour" returning highly–foldable low–novelty sequences. Even for low–novelty generations, our results provide evidence of "ligand novelty" by discovering unseen protein–ligand interactions. These results highlight the potential and limitations of current datasets and training paradigms, which have an impact on the broader training of pLMs. To facilitate future work, we will release our datasets and pretrained models.

2. Related Work

Recent generative-AI approaches to small-molecule binder design have been predominantly structure-centric, and many of the best-known demonstrations leverage diffusion models to generate ligand-compatible backbones. For example, RFdiffusion All-Atom (Krishna et al., 2024) generates backbones conditioned on a target structure and/or hotspot constraints (Fox et al., 2025), and has been applied to the design of digoxigenin, heme, and bilin binders. A common pipeline first generates a backbone containing a ligand-binding site, then derives an amino-acid sequence by inverse folding with methods such as ProteinMPNN, and finally ranks or refines candidates via re-folding or co-folding models such as AlphaFold2 (Baek et al., 2021), AlphaFold3 (Abramson et al., 2024), or Boltz (Passaro et al., 2025). In a complementary direction, frameworks such as BoltzDesign (Cho et al., 2025) directly optimize sequences (or designs) under a co-folding model by maximizing the likelihood of protein–ligand contacts under the model distribution. Beyond diffusion-based backbone generation, LigandMPNN (Dauparas et al., 2025) takes protein backbones together with docked molecules as input and designs sequences for the

specified conformation, with experimental validation reported for the redesign of multiple ligand-binding proteins.

While these structure-first methods benefit from strong geometric inductive biases, their applicability depends on access to high-resolution protein–ligand complex structures (or reliable poses and binding-site definitions). This motivates sequence-based alternatives: the breadth of accumulated sequence data is on the order of six orders of magnitude larger than the set of structurally resolved complexes, making approaches that operate directly on 1D sequences, such as protein language models (pLMs), particularly appealing(Richardson et al., 2023).

Prior work has shown that pLMs can incorporate useful control signals during training. Decoder-only models such as ProGen condition generation on taxonomic and functional keyword tags, while ZymCTRL (Munsamy et al., 2024) conditions on Enzyme Commission (EC) labels to generate sequences matching a desired catalytic reaction class. These models can generate active enzymes, but their controllability is largely limited to the EC numbers and GO terms represented in their training sets, which restricts generalisation to unseen functions. Other efforts have targeted covalent binding prediction: T5ProtChem (Kelly et al., 2025) trains an encoder–decoder model that takes protein–ligand pairs as input and outputs a SMILES string and binding position, together with a probability of covalent binding and can also be used for protein function classification and reaction prediction.

Encoder–decoder architectures have therefore been widely explored for structure-conditioned sequence design (Ingraham et al., 2019; Dauparas et al., 2022), however, the application of sequence-to-sequence conditioning directly on small molecules for protein generation remains completely underexplored.

3. Methods

3.1. Dataset

We created two datasets for training representing opposing data-availability regimes. The Binder dataset contains more unique molecules than proteins, whereas the Substrate dataset maps multiple unique proteins to each molecule (Figure 1b, and Figure A.2)

Binder–Dataset The training dataset was assembled by aggregating curated data from BindingDB (Liu et al., 2025), Drug Target Commons (DTC) (Tanoli et al., 2018), BioLiP (Yang et al., 2012), and AlphaFill (Hekkelman et al., 2023). Records missing identifiers, measurements, or displaying non-exact affinity relations were discarded. Ligand notation was standardised converting InChi codes to canonical SMILES. For DTC, compound/target identifiers were resolved via ChEMBL and UniProt to obtain ligand InChI and target sequences, respectively. Following common practice in literature, we retained only interactions with a reported affinity (IC50, Kd, or Ki) $\leq 10\,\mu$M, except for BioLip which we include fully. Finally, duplicated protein–ligand pairs were removed, and the set of binding proteins for each ligand was filtered to 90% identity with MMSeqs2 (Steinegger & Söding, 2017). The final dataset contains $\sim$1.8M unique ligand SMILES, $\sim$774k unique protein sequences and $\sim$10M total ligand–protein pairs, averaging 5.55 proteins per ligand (and 2.0 median). The dataset follows a long-tailed distribution, with 0.25% most annotated summing up to 51% of the total pairs, corresponding to promiscuous ligands that interact with thousands of proteins each. This is addressed using sampling strategies.

Substrate–Dataset As an additional experiment, we also retrieved data from enzyme datasets. Given a protein known to catalyse a reaction, we can assume the reactants also bind the protein. We extract reaction–enzyme pairs from Rhea (Bansal et al., 2022), BKMS (Lang et al., 2011), Uniprot (Bateman et al., 2025) and BRENDA (Chang et al., 2021). Then, from the substrate side of each one, we extract candidate ligands and filter by RDKit validity, heavy-atom count $\in [5, 50]$, and remove a small list of trivial species and ubiquitous metabolites (e.g., acetate/succinate/ATP). We discard reactions containing unspecified (wildcard) atoms, and allow at most a 2 hydrogen / 1 oxygen mismatch between substrate and product atoms (to account for common redox / protonation conventions / loss of a water molecule). SMILES are canonicalised, allowing stereo-chemistry when available. The final training set contains 4015 unique ligands, $\sim$7.8M protein sequences, and a total of $\sim$17M pairs, averaging $\sim$3600 proteins per ligand. Additionally, we also define Extended–dataset as the union of Substrate–dataset and the Binder–dataset.

SAIR We create a third dataset for testing, retrieved from the Structurally Augmented IC50 Repository (SAIR) (Lemos et al., 2025) and curated by discarding molecules failing basic chemistry and size constraints (valence, heavy-atom count, molecular weight, and rotatable-bond limits). We then extract per-residue heavy-atom coordinates for the polymer chain and all non-polymer components as candidate ligands/cofactors. Non-polymer components are clustered using contact-based merging: residue–ligand contacts are defined by an atom-pair distance threshold given by the sum of van der Waals radii plus a margin of 0.5Å (as computed in BioLip), then components are merged and considered in the same pocket when their contacting-residue sets have Jaccard overlap ≥ 0.30 and their minimum inter-component atom distance is ≤ 4Å. Like this, for each protein-ligand pair, we obtain a list of the residues in contact with the ligand and index them with their pocket identity, allowing to identify multiple pockets for the same ligand if available.

The resulting dataset collects ∼735K unique ligands, ∼5k unique protein sequences and ∼1M total ligand–protein pairs, averaging ∼1.4 proteins per ligand.

3.2. Model and training

Models were trained with a cross-entropy loss between the input molecule x' (text-based tokenized) and the target enzyme sequence $y = (y_1, \ldots, y_T)$ (amino acid tokens). Full details in Section A.1. We performed experiments with the architectures summarised in Table A.1. The main encoder–decoder followed the original T5 architecture (Raffel et al., 2020), using the T5-Base configuration, unless otherwise stated. Section 4.6.2 explores the integration of a LLama3 model (Grattafiori et al., 2024) as decoder, with pretrained weights from large scale protein language modelling. This pre-trained model was trained on 43M protein sequences obtained from Uniref90 and Gigaref (small molecule binders and non-binders). To condition the generation on an encoder representation, cross-attention layers were added to the decoder after each self-attention one. The input embedding matrix and the language modelling (LM) head were reshaped to the size of the new vocabulary. Then, training was performed in two stages. Stage–1 trained the encoder plus decoder's cross-attention, input embeddings and LM head, aiming to achieve a local minima that reuses decoder representations. Then, Stage–2 trained all parameters.

Sampling To compensate for the effect of distribution imbalance (Figure 1), we devise a *Unique–Ligand* sampling strategy, where each input (ligand) SMILES is sampled once per epoch and the target output protein is chosen at random (without replacement) from its binding list. This follows the same general principle used in previous large-scale pLM training to improve sample efficiency by preventing redundant sequence groups from dominating optimisation (Cheng et al., 2024; Meier et al., 2021). When unique-input sampling is performed, the number of epochs is increased to keep the number of training steps comparable to naive Pair sampling, given that each epoch only samples one solution per input. Comparative results between the two sampling strategies are provided in Table 1.

Tokenizer In our database, we represent ligands as SMILES (Weininger, 1988) strings (or SELFIES (Krenn et al., 2020) for the input comparison experiments), while output proteins are represented using the standard 20–amino-acid alphabet. We developed a tokenizer tailored for ligand-conditioned protein generation, which we release to the community. Full details available in Section A.2.

3.3. Metrics

Protein generation (25 generations per input ligand; top-k sampling with k=15 and temperature t=1) was evaluated both for the test and training sets. Test set ligands are not present in the training set, while associated proteins can overlap. The two sets were further divided grouping by number of annotated proteins per ligand, in ranges: $[50, \infty), [10, 49], [2, 9], \{1\}$. This stratification enables evaluation across regimes of annotation density and label ambiguity (from highly multimodal to near-singleton supervision). We selected 200 unique ligands per split.

To evaluate structural reliability of the model's designs, we predicted their folded structure using ESMFold (Lin et al., 2021). This approach has been common practice in the protein design community, given the correlation between pLDDT and the likelihood of the sequence being ordered (Tunyasuvunakool et al., 2021). To quantify the similarity of predicted proteins to the training set, for each prediction, we compute alignment–based similarity through MMSeqs2 against all training samples (Train Id.). This quantifies the novelty of a sequence, indicating to what extent the protein is retrieved from the training distribution.

As a proxy for binding success on zero-shot ligands, we compute ground-truth (GT) retrieval: since test ligands are not seen during training, a prediction that matches an annotated GT binder (or a close homolog) provides evidence that the model generated a plausible binder for this hold-out ligand. On the contrary, not matching GT is no guarantee of an invalid solution. For GT matches, generalisation is arguably proportional to how different the test ligand is to the training examples. Using MMSeqs2 search, non-exact matches will also be identified, as sequences with few amino acid mutations have a high likelihood of converging to a similar fold. We compute GT accuracy (GT Acc.) by selecting the maximum identity match among all generated proteins per input (25), and averaging across all test samples. We also report the mean and standard deviation of all matches.

Finally, we also evaluated generations through Boltz2 cofolding (Passaro et al., 2025). We select the generation with highest ESMFold pLDDT per target (among the 25 top-k samples). We use Boltz2 binding confidence as metric. Boltz2 uses two predictors and averages their output. Therefore, we report the percentage of sequences in which the average confidence is >0.5 and also those in which the average confidence >0.5 and at least one predictor is >0.70. This metric does not depend on GT annotation, on the contrary, its limits are defined by Boltz2's accuracy, therefore it complements GT Acc. as an orthogonal evidence.

4. Results

Results in Table 1, under "Ligand Sampling", present the performance of our best scoring model, which is trained on the Binder–dataset with Unique–Ligand sampling. Mean pLDDT indicates that the network yields foldable sequences across different splits, decaying only slightly for the most

promiscuous ligands. GT accuracy demonstrates non-trivial success in the retrieval of annotated binders across most ligand strata; particularly in the moderately annotated regime (90.89% GT acc.). In contrast, GT retrieval decreases for the most promiscuous test ligands (46.17% GT acc.), which likely reflects a harder multi-modal distribution following a less deterministic pattern. For ligands that only map to a single protein in the dataset, performance is lower but non-negligible (32.86% GT acc.).

Leveraging the pocket annotations provided by the SAIR split (Section 3), we extract the ground-truth (GT) pocket residues and compute sequence identity specifically over the binding site. Then we compare it to the identity towards the full (GT) protein. We sample 400 unique ligands for each SAIR evaluation set, one with ligands seen during training and another with hold-outs (protein/molecule ratio presented in Figure A.2). Results in Table 1 indicate a closely similar score for pocket ("SAIR Pocket") vs full protein matches ("SAIR"), with "Pocket" being slightly higher. Binding specificity is primarily determined by residues in the binding pocket; therefore, these results suggest that GT matches are not driven solely by non-functional sequence regions but are also supported by similarity at the binding site.

Finally, as previously explained, Unique–Ligand sampling is used to prevent the over-sampling of heavily annotated ligands. Table 1 presents results for the same model under regular Pair–Sampling, using the same number of training steps. Ligand sampling improves both structural confidence and GT retrieval relative to pair sampling, with the largest gains appearing in sparsely supervised regimes (e.g., test singletons and the $[2, 9]$ stratum).

Table 1. Main model evaluation trained with Ligand and Pair sampling at equal number of steps.

	Split range	Mean±std pLDDT	GT MMSeqs Acc.
Ligand Sampling			
Train	[50, 24570]	77.67±15.14	78.37%
	[10, 49]	79.05±12.23	92.46%
	[2, 9]	79.70±11.27	72.51%
	{1}	79.91±10.49	46.34%
	SAIR	79.07±10.68	38.31%
	SAIR Pocket	–	39.51%
Test	[50, 346253]	75.94±18.35	46.17%
	[10, 49]	79.12±12.78	90.89%
	[2, 9]	79.75±12.05	51.29%
	{1}	78.41±13.14	32.86%
	SAIR	79.36±10.42	38.82%
	SAIR Pocket	–	39.56%
Pair Sampling			
Train	[50, 24570]	65.66±24.33	72.57%
	[10, 49]	77.79±14.31	93.50%
	[2, 9]	77.24±15.25	45.77%
	{1}	77.14±15.17	17.50%
Test	[50, 346253]	55.11±26.19	37.39%
	[10, 49]	76.49±16.37	89.94%
	[2, 9]	76.59±16.34	34.53%
	{1}	73.76±19.25	9.49%

4.1. Ligand novelty

To assess generalisation, we measure the novelty of test ligands relative to the training set. Specifically, we quantify how dissimilar a target ligand is from ligands associated with the closest training proteins in sequence space. Correct generations that match the GT while involving ligands far from any training example indicate generalisation beyond memorisation of observed protein–ligand pairs.

For each generated protein, we compute its MMseqs identity versus all train samples and select its nearest neighbours (top 50 or all the maximally similar ones if they are more than 50). Instead of regular identity id, given alignment length a and target lenght t we define an effective identity ($\text{eff} = id \cdot \frac{a}{t}$), to favour matches of larger coverage. We retain the set of closest training proteins within a tolerance: $\text{eff} \geq \text{eff}_{\max} - 0.05$ (with the exception of GT sequences that are always included if they are listed as a match). We then pool all ligands paired with these proteins in the training set and measure their similarity towards the test ligand by means of Tanimoto similarity with Extended Connectivity Fingerprints (Rogers & Hahn, 2010) (radius 2, 2048 bits). Finally, the maximum Tanimoto similarity (MTS) is reported. This value indicates the following: If the generated protein closely matches a training example *E*, the reported value is the maximum chemical similarity found for a ligand associated to *E* and the test ligand.

Figure 2.A plots GT identity versus MTS for each generated sample with the main model on the $[\geq 50]$ test split (results for the rest of splits available in supplementary Section B). We observe that MTS spans a broad range even when the generated protein closely matches an annotated GT binder. Successful GT retrievals (high GT identity) occur not only for ligands that are near-duplicates of ligands seen with similar training proteins, but also for ligands with low chemical similarity to any ligand in the neighbourhood. We interpret these cases as ligand-level generalisation: the model recovers an annotated protein (or close homolog) for a ligand that is chemically dissimilar to ligands observed in the training neighbourhood of that protein family.

To contextualise how much of the GT retrieval can be explained by chemical-similarity retrieval, Section 4.5 computes evaluates GT matching with a Tanimoto retrieval baseline.

4.2. Protein novelty

We compute MMSeqs2 similarity of generated sequences against the training set (Train Id.). For the model trained

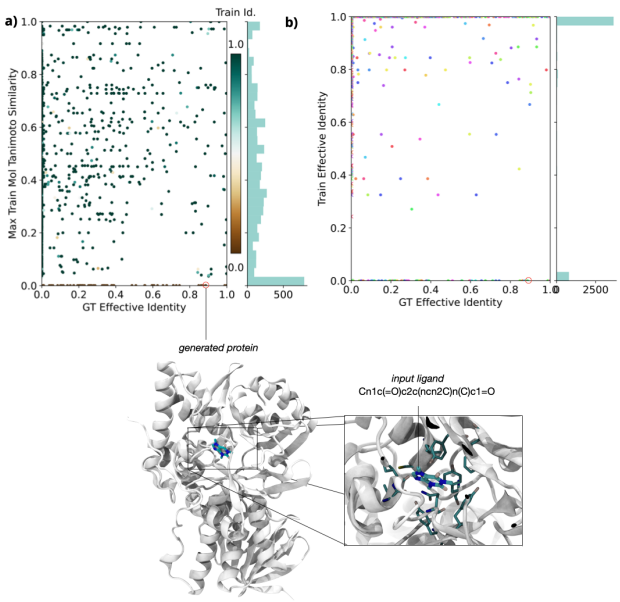

Figure 2. **(a)** MTS versus GT eff id. per generation in the $[\geq 50]$ test split. **(b)** Train eff id. versus GT eff id. on the same split. Bottom: Co-folding by Boltz2 of a high GT identity generation with no training match (0.97 complex pLDDT, 0.85 iPTM).

with Pair Sampling, training hits go from ∼52–95% depending on stratum, while Ligand sampling increases to ∼85–99% (full results in Supplementary Figure A.1). The distribution can be visualised in Figure 1.C. This same figure compares the distribution to the one obtained with the Substrate–Dataset.

Figure 2.B visualizes the relationship between GT retrieval and protein-level novelty by plotting the effective identity of each generation to the GT (x-axis) and to the closest training neighbour (y-axis). Cases where a generation matches the GT more closely than any training sequence provide a direct signature of protein-level generalisation: the model recovers a binder sequence that is closer to the held-out annotation than to the training distribution. Such events are expected to be rare, since they require the GT binder itself to be sufficiently distant from training proteins, which occurs infrequently in natural datasets. Nevertheless, the presence of such examples indicates that the model can move beyond nearest-neighbour retrieval and generate sequences aligned with held-out binders. Figure 2 displays the co-folding prediction of a generation with 0.88 GT effective identity (0.99 target coverage) and no training match, which was targeting the caffeine molecule. The co-folding was computed with Boltz2 using MSA templates, which predicts successful binding with high confidence (0.97 pLDDT and 0.85 iPTM). Therefore, despite the absence of caffeine binders in training, the model generated a sequence with high likelihood of binding according to Boltz2, while remaining 10% divergent from the closest GT annotation.

Table 2. Effect of sampling temperature (T) and top-k (K) on SAIR evaluation.

T	K	TRAIN ID. | MATCHES	GT MMSEQS ACC.
0.8	15	1.00±0.01 | 99.1%	40.48%
1.0	15	1.00±0.01 | 98.2%	38.82%
1.4	50	1.00±0.04 | 91.1%	35.23%
1.6	50	0.99±0.06 | 81.0%	31.40%
1.6	100	0.98±0.06 | 78.4%	31.19%
1.8	50	0.97±0.08 | 68.4%	26.86%
2.0	50	0.93±0.11 | 51.3%	18.86%

As a final analysis, we study the influence of generation temperature in sequence novelty. The model learned to reproduce low entropy distributions; therefore, we test whether higher sampling temperatures can increase such entropy and increase novelty. Table 2 does not support the hypothesis: increasing temperature and *k* in *top-k* sampling reduces the fraction of generations that match any training sequence, but the identity of the remaining matches remains similar. Together with the reduction in GT matching, this suggests off-target generations rather than useful novelty. Following the same trend, lowering temperature to < 1 values increases GT matching. This is coherent with a model that learnt a narrow output distribution and rapidly falls out of distribution when deviating from the greedy decoding.

4.3. Substrate–Dataset results

Using the Substrate–Dataset (defined in Section 3) we train an identical model on this alternative data distribution. Figure 1.C and 1.D compare the Train Id. and pLDDT distributions (respectively) of this model to the one trained on the Binder–Dataset, revealing substantial differences between the two. As depicted in Figure 1.B, the Substrate–Dataset contains orders of magnitude fewer ligands, and orders of magnitude more proteins, providing a wide distribution of outputs for each input. This causes the model to learn a distribution of higher entropy per input, closer to what it is observed in non-conditional pLMs when learning the full distribution of natural proteins.

Notice that the higher variability in generation comes at the cost of lower pLDDT values, which is expected, as retrieving known proteins from training set is arguably an upper bound for foldability, while deviating from the natural distribution is known to reduce it. Table 3 reports the complete results, including Ligand and Pair sampling. As expected, when Ligand Sampling allows the lower entropy regions ($[2, 9]$, $\{1\}$) to be trained on as much as the largest ones, they move closer to a retrieval behaviour.

The combination of Substrate and Binder datasets yielded similar results to training exclusively on the Binder–Dataset, without showing improvement. Results are available in Supplementary C.

Table 3. Evaluation on the substrate–enzyme dataset.

	Split range	pLDDT	Train Id. | matches	GT MMSeqs Acc.
Ligand Sampling				
Train	[50, 439644]	35.63±15.59	0.44±0.20 | 48.1%	50.33%
	[10, 49]	49.65±25.07	0.65±0.30 | 66.5%	72.94%
	[2, 9]	85.10±13.22	0.98±0.09 | 97.6%	99.83%
	{1}	87.24±7.86	1.00±0.02 | 99.7%	100.00%
Test	[50, 132623]	39.87±22.01	0.62±0.31 | 35.3%	15.13%
	[10, 49]	42.06±23.12	0.62±0.32 | 46.4%	11.43%
	[2, 9]	57.83±28.36	0.83±0.26 | 60.6%	32.22%
	{1}	57.88±29.22	0.86±0.26 | 62.0%	30.43%
Pair Sampling				
Train	[50, 439644]	56.47±23.59	0.48±0.16 | 59.6%	61.43%
	[10, 49]	42.99±20.00	0.39±0.13 | 34.1%	19.50%
	[2, 9]	40.95±19.72	0.40±0.14 | 28.4%	10.09%
	{1}	42.77±21.31	0.41±0.15 | 30.1%	6.26%
Test	[50, 132623]	55.92±25.00	0.49±0.15 | 53.0%	21.15%
	[10, 49]	52.94±24.16	0.46±0.15 | 48.7%	6.86%
	[2, 9]	46.80±22.14	0.46±0.17 | 36.4%	6.53%
	{1}	41.84±20.61	0.43±0.16 | 28.1%	3.28%

Table 4. Boltz2 co-folding results. "Binder > 0.5" indicates percentage of generations with predicted binder probability > 0.5. "Confident Binder" indicates Binder > 0.5 plus at least one > 0.7 confident prediction (among the two classification heads).

Split range	Binder > 0.5	Confident Binder
Substrate Dataset		
$\{1\}$	45.96%	27.33%
$[2, 9]$	48.63%	36.99%
$[10, 49]$	36.36%	24.24%
$[50, \max]$	27.64%	17.09%
Binder Dataset		
$\{1\}$	37.34%	23.42%
$[2, 9]$	31.65%	20.25%
$[10, 49]$	40.12%	27.78%
$[50, \max]$	23.65%	20.27%

4.4. Co-folding evaluation

Results show that a non-trivial fraction of generated sequences receive favorable Boltz2 binding-confidence scores across both datasets. Only one generation per target was evaluated, in a design campaign, a larger number of hits could be achieved by co-folding multiple generations per target. Interestingly, under this metric, the Substrate dataset model achieves comparable scores to the Binder dataset one. This is consistent with co-folding scores being less tied to annotation overlap than GT Acc.

Although this metric still remains a computational proxy, it serves as "orthogonal evidence" complementing the GT Acc. results, as it is independent of the annotation of the data set, therefore subject to different strengths and limitations.

4.5. Fingerprint Nearest Neighbour retrieval baseline

As a diagnostic baseline to probe dataset overlap and the sensitivity of the GT identity metric to retrieval, we implement fingerprint k-nearest-neighbour (FP-NN) retrieval. For each test ligand, we compute a Morgan fingerprint (ECFP4; radius 2, 2048 bits) and retrieve the most similar ligand from the training set by Tanimoto similarity. We then output the protein sequence associated with each retrieved training ligand. The resulting FASTA outputs are evaluated with the same GT MMSeqs pipeline used for model generations.

Table 5. Model's greedy decoding vs FP-NN. FP-NN Acc. indicates FP-NN GT Acc. FP-NN$>$ indicates the fraction of inputs where FP-NN achieves higher maximum identity with respect to GT, and Model$>$ the fraction where the model does. $\Delta_{\text{model}>}$ is the mean (model $-$ FP) identity among inputs where the model has higher identity.

Split range	FP-NN Acc.	FP-NN $>$	Model $>$	Ties	$\Delta_{\text{model}>}$
[50, 346253]	40.26%	35.38%	7.18%	57.44%	0.693
[10, 49]	46.62%	28.50%	22.50%	49.00%	0.890
[2, 9]	65.12%	57.50%	6.00%	36.50%	0.912
{1}	77.02%	70.50%	0.00%	29.50%	–

Table 5 shows how FP-NN retrieves proteins matching the GT for a large fraction of test instances. Then, we use these results to quantify how often the model goes beyond simple retrieval. We compared to the model's greedy decoding, to make a fair 1-shot comparison, and computed the percentage of targets for which the model generation was closer to the GT than the FP-NN protein. We also verify that in these cases, the model proposal differs substantially in GT matching identity from the retrieval one (Table 5 $\Delta_{\text{model}>}$ column). This reinforces the conclusions of the generalisation studies, which showed examples beyond retrieval.

In terms of average matching, FP-NN matches the GT protein more frequently than the model. This suggests that the network does an imperfect job of memorising all training proteins and matching them to suitable targets. It was seen how, in cases where retrieval is not enough, the model can go beyond, but the occurrence of these cases in the Binder-dataset is not high enough to make the global average higher than pure retrieval. The smallest margin is found in the $[10, 49]$ stratum, the greedy decoding matches GT at 38.83% and FP-NN at 46.62%.

As a final note, these results should be interpreted with caution because GT MMSeqs rewards matching the specific annotated binder sequences and therefore favours retrieval mechanisms. Plausible binders from alternative families (or novel designs) are penalised even if they could bind, biasing towards retrieving the proteins present in the dataset.

4.6. Architecture experiments

4.6.1. Decoder-only architecture

Following the procedure described in Section 3, we trained a decoder-only model to explore the impact of the architectural choice in performance. Results in Table 6 show

Table 6. Encoder–decoder (t5-tiny) versus decoder-only (GPT2) in a common 100-sample test set. Both models were trained for 2 epochs with pair sampling.

MODEL	PARAMS (M)	PLDDT	GT MMSEQS ACC.
ENCODER–DECODER	7.6	65.20±25.18	22.24%
DECODER–ONLY	14.6	61.67±11.37	17.01%

a lower test GT accuracy along with decreased average pLDDT compared to the encoder-decoder version.

It is important to note that these results are obtained for a specific dataset and architecture size with limited hyper-parameter tuning. Nevertheless, given the proven data and computing efficiency of encoder-decoder architectures for conditioning tasks (Tay et al., 2022; Elfeki et al., 2025), the results confirm that they are easier to train for our task.

4.6.2. PROTEIN LANGUAGE MODELLING PRETRAINING

The performance of the pretrained Llama3 model was evaluated in the SAIR test set and compared to the baseline t5 one, both trained in the main Binder–dataset and also in the Extended–dataset. Baseline models were trained from scratch for 45k steps. Pretrained ones employed 10k in Stage–1 (frozen decoder, as defined in Section 3) and 45k steps in Stage–2, where all parameters are trained. All models used a common batch size (1152), consequently, they were exposed to the same number of examples during unconstrained training, avoiding an unfair advantage for the baseline ones.

Results show how the model trained from scratch outperforms the pretrained one, both in pLDDT and GT matching. Regarding the train identity metric, it seems to indicate that the pretrained model did not generate a richer output distribution. Same as with the baseline model, generations finding alignments have maximum identity. This indicates that our training pipeline did not manage to preserve the richer distribution in the decoder, which converged to the same solution as the baseline. Additionally, lower performance indicates either that pretrained initialisation hampered convergence or that the hyper-parameters, which we kept the same as for the baseline, were suboptimal for this approach. It remains an open question whether additional techniques can be used to preserve the original distribution of the decoder while achieving competitive results.

4.6.3. PARAMETER SCALING

Performance across model sizes was computed for the tiny, base, and large configurations of the T5 family. Table 7 presents average pLDDT for the SAIR test set. Consistent with literature, we observe diminishing performance returns, hinting to logarithmic improvement over the number of parameters. In particular, the ∼700M configuration is only slightly superior to the ∼200M one (80.14 vs 79.36 average pLDDT). Hence, we select the ∼200M for the experiments in this work.

Table 7. Effect of model size on SAIR test pLDDT. t5-Tiny and T5-base were trained for 45K steps while T5-Large for 40K.

MODEL	PARAMS (M)	PLDDT
T5–TINY	7.62	47.00±25.56
T5–BASE	199.05	79.36±10.42
T5–LARGE	705.86	80.14±9.65

Table 8. Pretrained-llama models vs baseline in SAIR test set. "P." stands for pretrained, Binder for model trained on the Binder dataset and Extended for the joint Binder + Substrate dataset.

MODEL	PLDDT	TRAIN ID. | MATCHES	GT MMSEQS ACC.
BINDER	79.07±10.68	1.00±0.01 | 99.8%	38.82%
EXTENDED	70.62±22.78	0.99±0.02 | 66.6%	31.24%
P. BINDER	74.13±15.42	1.00±0.01 | 76.3%	19.85%
P. EXTENDED	57.22±16.70	1.00±0.02 | 71.0%	27.30%

4.7. Input format experiments

To assess the relevance of input formatting, here we report results with SELFIES (Self-Referencing Embedded Strings) (Krenn et al., 2020), an alternative molecular string representation. Unlike SMILES, in SELFIES every syntactically valid sequence corresponds to a valid chemical structure, aiming to simplify the space of solutions for generative models. This comes at the cost of creating longer strings. Table 9 presents results of identical models trained with SMILES vs SELFIES. We do not find benefit in this alternative. We hypothesise that the advantage of an "always-valid" space is relevant when used in the generation output, but potentially unnecessary at the input. We refer readers to studies comparing the two representations (Skinnider, 2024; Flam-Shepherd et al., 2022; Leon et al., 2024).

4.8. Maximal distant test set

As an additional test of the generalisation capabilities of the model, we construct a test set that is maximally distant from training in protein sequence space, while associated ligands are also held out from training. Full algorithmic details available in Section D.

Table 10 presents results of the evaluation in this set, showing a sharp drop in GT identity. Strata with the lower annotation volume retain some identity to GT and high pLDDT, while splits $\in [\geq 10]$ collapse, yielding sequences of lower foldability. Train identity remains high; therefore, the model is still retrieving seen protein families rather than adapting to the held-out distant ones. This behaviour is expected under our construction: when ground-truth binders are forced to be far in sequence space, any training-like output will

Table 9. pLDDT results for identical t5-tiny models using SMILES or SELFIES as ligand representation. All models were trained for 5 epochs and evaluated in the same sets of 100 examples each.

	INPUT	MEAN±STD PLDDT
TRAIN	SMILES	65.79±24.20
	SELFIES	61.44±26.34
TEST	SMILES	73.54±18.53
	SELFIES	71.71±21.58

Table 10. Maximally distant test set on Extended–Dataset. SAIR split remains the same as used for previous experiments.

	SPLIT RANGE	PLDDT	TRAIN ID. | MATCHES	GT MMSEQS ACC.
TEST	[50, 34153]	55.00±29.49	1.00±0.03 | 27.4%	0.80%
	[10, 49]	58.28±28.75	1.00±0.03 | 33.4%	0.00%
	[2, 9]	76.64±17.29	1.00±0.02 | 85.2%	7.73%
	{1}	77.69±14.42	1.00±0.01 | 92.5%	10.17%
	SAIR	70.62±22.78	0.99±0.02 | 66.6%	31.24%

score poorly against GT by sequence similarity, even if it were a plausible binder for the ligand. This test confirms that retrieval behaviour persists when prompted out of distribution. The validity of output sequences has a much lower likelihood of correctness in this regime; still, these results are not by themselves conclusive evidence of non-binding or overfitting, since alternative binder families may exist.

5. Discussion and conclusions

We benchmarked ligand-conditioned sequence-to-sequence pLMs for small-molecule binder design using purely textual inputs, with the goal of understanding when brute-force conditional language modeling can propose plausible binders. We find that available annotation for most ligands is sparse, pushing the model towards a retrieval-like behaviour. This behaviour yields low novelty at the protein level, but as supported by the results on held-out molecules, it has potential for the discovery of new protein-ligand interactions.

When a ligand is paired with a narrow set of sequences, the cross-entropy optimum concentrates probability mass on a small number of modes; this makes it possible to minimise the loss by memorising and retrieving previously seen proteins or close homologs. When supervision for a ligand spans a broader set of binders (high conditional entropy), the model is forced to represent a richer conditional distribution, moving closer towards the behaviour of unconditional pLMs, which represent the extreme of this trade-off by sharing the same input for all sequences. Therefore, higher generalisation will be expected for ligands with rich annotation. Among them, as suggested by the lower performance of the $[\geq 50]$ strata, GT will be most informative when the ligand is not excessively promiscuous, else, the output distribution will be highly multi–modal, making convergence harder, and probably requiring an even larger volume of annotation. This lesson helps to anticipate results based on data distribution, and hints at the current ceiling for language modelling on binder datasets, which is limited by the sparsity of annotation.

Ligand novelty analysis demonstrated how the model can generate a protein matching the GT (or close homolog) without seeing examples of such protein family being associated to chemically similar ligands. This requires a general understanding of chemical compatibility and binding activity, which allows to "fill the gaps" and understand which proteins could bind a newly found ligand. As discussed, these results are most often obtained by reproducing sequences seen during training, still we argue there is value in finding novel uses for known proteins. Additionally, we also find few cases where generation diverged from training but still proposed a plausible binder, such as the caffeine binder, validated by Boltz2 co-folding. Therefore, the model can be used to yield multiple binder proposals (in a matter of seconds) and then select the most promising ones for experimental validation. Filtering tests can include, among others, co-folding, docking or search against protein databases. We consider this post-processing to be an important component of future pipelines, especially since GT-based metrics are an imperfect proxy for binding success. It is of interest for future work to characterise success with higher fidelity measures and experimental validation.

To the best of our knowledge, pLM conditioning had not been yet studied for targets outside the training set, hence we believe an initial baseline was necessary. Here, we benchmark architectures, model sizes, generation parameters, input formats, and pLM pretraining. Together with the curated datasets, and evaluation pipelines, we hope to provide the necessary tools to lower the barrier for the exploration of ligand-conditioned protein generation.

Looking forward, the target for general–purpose AI binder design should be to push novelty, designing de-novo proteins which diverge from known sequences, while demonstrating high success rate in lab trials. As discussed, the role of pLMs in this development is tied to databases. Scaling text-based datasets is a way forward, but remains expensive. Alternatives include using richer annotation, such as binding affinity, or resorting to multi-modality with structural data. In addition, inductive biases, such as physics–based computations, might also prove necessary.

Acknowledgements

Anonymised

Impact Statement

The primary impact of this work is to provide rigorous foundation for instance-level conditioning in protein language models. Outcomes and insights achieved in this work assist researchers in building next-generation conditional pLMs and enable practitioners to triage candidate binders via downstream filtering (e.g., co-folding/docking) before any experimental work. There are no direct societal impacts from this work alone, as we do not report experimentally validated binders or deployable design protocols. In the long run, contributions to general purpose AI binder design aim to enable faster discovery for therapeutics, sustainability, or industrial purposes. The consequences of such developments are shared across all disciplines contributing to these ends.

References

Abramson, J., Adler, J., Dunger, J., Evans, R., Green, T., Pritzel, A., Ronneberger, O., Willmore, L., Ballard, A. J., Bambrick, J., et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630 (8016):493–500, 2024.

Baek, M., DiMaio, F., Anishchenko, I., Dauparas, J., Ovchinnikov, S., Lee, G. R., Wang, J., Cong, Q., Kinch, L. N., Schaeffer, R. D., Millán, C., Park, H., Adams, C., Glassman, C. R., DeGiovanni, A., Pereira, J. H., Rodrigues, A. V., van Dijk, A. A., Ebrecht, A. C., Opperman, D. J., Sagmeister, T., Buhlheller, C., Pavkov-Keller, T., Rathinaswamy, M. K., Dalwadi, U., Yip, C. K., Burke, J. E., Garcia, K. C., Grishin, N. V., Adams, P. D., Read, R. J., and Baker, D. Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, 373(6557):871–876, 2021. doi: 10.1126/science.abj8754. URL https://www.science.org/doi/abs/10.1126/science.abj8754.

Bansal, P., Morgat, A., Axelsen, K. B., Muthukrishnan, V., Coudert, E., Aimo, L., Hyka-Nouspikel, N., Gasteiger, E., Kerhornou, A., Neto, T. B., Pozzato, M., Blatter, M. C., Ignatchenko, A., Redaschi, N., and Bridge, A. Rhea, the reaction knowledgebase in 2022. *Nucleic Acids Research*, 50:D693–D700, 1 2022. ISSN 13624962. doi: 10.1093/nar/gkab1016.

Bateman, A. et al. Uniprot: the universal protein knowledgebase in 2025. *Nucleic Acids Research*, 53:D609–D617, 1 2025. ISSN 0305-1048. doi: 10.1093/nar/gkae1010. URL https://academic.oup.com/nar/article/53/D1/D609/7902999.

Chang, A., Jeske, L., Ulbrich, S., Hofmann, J., Koblitz, J., Schomburg, I., Neumann-Schaal, M., Jahn, D., and Schomburg, D. Brenda, the elixir core data resource in 2021: New developments and updates. *Nucleic Acids Research*, 49:D498–D508, 1 2021. ISSN 13624962. doi: 10.1093/nar/gkaa1025.

Cheng, X., Chen, B., Li, P., Gong, J., Tang, J., and Song, L. Training compute-optimal protein language models. *Advances in Neural Information Processing Systems*, 37: 69386–69418, 2024.

Cho, Y., Pacesa, M., Zhang, Z., Correia, B. E., and Ovchinnikov, S. Boltzdesign1: Inverting all-atom structure prediction model for generalized biomolecular binder design. *bioRxiv*, pp. 2025–04, 2025.

Dauparas, J., Anishchenko, I., Bennett, N., Bai, H., Ragotte, R. J., Milles, L. F., Wicky, B. I., Courbet, A., de Haas, R. J., Bethel, N., et al. Robust deep learning–based protein sequence design using proteinmpnn. *Science*, 378 (6615):49–56, 2022.

Dauparas, J., Lee, G. R., Pecoraro, R., An, L., Anishchenko, I., Glasscock, C., and Baker, D. Atomic context-conditioned protein sequence design using ligandmpnn. *Nature Methods*, pp. 1–7, 2025.

Elfeki, M., Liu, R., and Voegele, C. Return of the encoder: Maximizing parameter efficiency for slms. *arXiv preprint arXiv:2501.16273*, 2025.

Flam-Shepherd, D., Zhu, K., and Aspuru-Guzik, A. Language models can learn complex molecular distributions. *Nature Communications*, 13(1):3293, 2022.

Fox, D. R., Taveneau, C., Clement, J., Grinter, R., and Knott, G. J. Code to complex: Ai-driven de novo binder design. *Structure*, 33(10):1631–1642, 2025. ISSN 0969-2126. doi: https://doi.org/10.1016/j.str.2025.08.007. URL https://www.sciencedirect.com/science/article/pii/S0969212625003119.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Hekkelman, M. L., de Vries, I., Joosten, R. P., and Perrakis, A. Alphafill: enriching alphafold models with ligands and cofactors. *Nature Methods*, 20(2):205–213, 2023.

Ingraham, J., Garg, V., Barzilay, R., and Jaakkola, T. Generative models for graph-based protein design. *Advances in neural information processing systems*, 32, 2019.

Ivančić, D., Agudelo, A., Lindstrom-Vautrin, J., Jaraba-Wallace, J., Gallo, M., Das, R., Ragel, A., Herrero-Vicente, J., Higueras, I., Billeci, F., et al. Discovery and protein language model-guided design of hyperactive transposases. *Nature Biotechnology*, pp. 1–6, 2025.

Johnson, S. R., Fu, X., Viknander, S., Goldin, C., Monaco, S., Zelezniak, A., and Yang, K. K. Computational scoring and experimental evaluation of enzymes generated by neural networks. *Nature Biotechnology*, 43(3):396–405, 2025.

Kelly, T., Xia, S., Lu, J., and Zhang, Y. Unified deep learning of molecular and protein language representations with t5protchem. *Journal of Chemical Information and Modeling*, 65(8):3990–3998, 2025.

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

Krenn, M., Häse, F., Nigam, A., Friederich, P., and Aspuru-Guzik, A. Self-referencing embedded strings (selfies): A 100 *Machine Learning: Science and Technology*, 1 (4):045024, oct 2020. doi: 10.1088/2632-2153/aba947. URL https://doi.org/10.1088/2632-2153/aba947.

Krishna, R., Wang, J., Ahern, W., Sturmfels, P., Venkatesh, P., Kalvet, I., Lee, G. R., Morey-Burrows, F. S., Anishchenko, I., Humphreys, I. R., McHugh, R., Vafeados, D., Li, X., Sutherland, G. A., Hitchcock, A., Hunter, C. N., Kang, A., Brackenbrough, E., Bera, A. K., Baek, M., DiMaio, F., and Baker, D. Generalized biomolecular modeling and design with rosettafold all-atom. *Science*, 384(6693):eadl2528, 2024. doi: 10.1126/science.adl2528. URL https://www.science.org/doi/abs/10.1126/science.adl2528.

Lang, M., Stelzer, M., and Schomburg, D. Bkm-react, an integrated biochemical reaction database. *BMC Biochemistry*, 12, 2011. ISSN 14712091. doi: 10.1186/1471-2091-12-42.

Lemos, P., Beckwith, Z., Bandi, S., van Damme, M., Crivelli-Decker, J., Shields, B. J., Merth, T., Jha, P. K., De Mitri, N., Callahan, T. J., Nish, A., Abruzzo, P., Salomon-Ferrer, R., and Ganahl, M. Sair: Enabling deep learning for protein-ligand interactions with a synthetic structural dataset. 2025. doi: 10.1101/2025.06.17.660168.

Leon, M., Perezhohin, Y., Peres, F., Popovič, A., and Castelli, M. Comparing smiles and selfies tokenization for enhanced chemical language modeling. *Scientific Reports*, 14(1):25016, 2024.

Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., Verkuil, R., Kabeli, O., Shmueli, Y., Costa, A. D. S., Fazel-Zarandi, M., Sercu, T., Candido, S., and Rives, A. Evolutionary-scale prediction of atomic level protein structure with a language model. *Science*, 379, 2021. doi: 10.1101/2022.07.20.500902. URL https://doi.org/10.1101/2022.07.20.500902.

Liu, T., Hwang, L., Burley, S. K., Nitsche, C. I., Southan, C., Walters, W. P., and Gilson, M. K. Bindingdb in 2024: a fair knowledgebase of protein-small molecule binding data. *Nucleic acids research*, 53(D1):D1633–D1644, 2025.

Madani, A., McCann, B., Naik, N., Keskar, N. S., Anand, N., Eguchi, R. R., Huang, P.-S., and Socher, R. Progen: Language modeling for protein generation. *arXiv preprint arXiv:2004.03497*, 2020.

Madani, A., Krause, B., Greene, E. R., Subramanian, S., Mohr, B. P., Holton, J. M., Olmos Jr, J. L., Xiong, C., Sun, Z. Z., Socher, R., et al. Large language models generate functional protein sequences across diverse families. *Nature biotechnology*, 41(8):1099–1106, 2023.

Meier, J., Rao, R., Verkuil, R., Liu, J., Sercu, T., and Rives, A. Language models enable zero-shot prediction of the effects of mutations on protein function. *Advances in neural information processing systems*, 34:29287–29303, 2021.

Munsamy, G., Illanes-Vicioso, R., Funcillo, S., Nakou, I. T., Lindner, S., Ayres, G., Sheehan, L. S., Moss, S., Eckhard, U., Lorenz, P., and Ferruz, N. Conditional language models enable the efficient design of proficient enzymes. *bioRxiv*, 2024. doi: 10.1101/2024.05.03.592223. URL https://www.biorxiv.org/content/early/2024/05/05/2024.05.03.592223.

Passaro, S., Corso, G., Wohlwend, J., Reveiz, M., Thaler, S., Somnath, V. R., Getz, N., Portnoi, T., Roy, J., Stark, H., et al. Boltz-2: Towards accurate and efficient binding affinity prediction. *BioRxiv*, 2025.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21 (140):1–67, 2020.

Richardson, L., Allen, B., Baldi, G., Beracochea, M., Bileschi, M. L., Burdett, T., Burgin, J., Caballero-Pérez, J., Cochrane, G., Colwell, L. J., et al. Mgnify: the microbiome sequence data analysis resource in 2023. *Nucleic acids research*, 51(D1):D753–D759, 2023.

Rogers, D. and Hahn, M. Extended-connectivity fingerprints. *Journal of chemical information and modeling*, 50 (5):742–754, 2010.

Romero-Romero, S., Braun, A. E., Kossendey, T., Ferruz, N., Schmidt, S., and Höcker, B. De novo design of triosephosphate isomerases using generative language models. *bioRxiv*, pp. 2024–11, 2024.

Skinnider, M. A. Invalid smiles are beneficial rather than detrimental to chemical language models. *Nature Machine Intelligence*, 6(4):437–448, 2024.

Steinegger, M. and Söding, J. Mmseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature biotechnology*, 35(11):1026–1028, 2017.

Tanoli, Z., Alam, Z., Vähä-Koskela, M., Ravikumar, B., Malyutina, A., Jaiswal, A., Tang, J., Wennerberg, K., and Aittokallio, T. Drug target commons 2.0: a community platform for systematic analysis of drug–target interaction profiles. *Database*, 2018:bay083, 2018.

Tay, Y., Dehghani, M., Tran, V. Q., Garcia, X., Wei, J., Wang, X., Chung, H. W., Shakeri, S., Bahri, D., Schuster, T., et al. Ul2: Unifying language learning paradigms. *arXiv preprint arXiv:2205.05131*, 2022.

Tunyasuvunakool, K., Adler, J., Wu, Z., Green, T., Zielinski, M., Žídek, A., Bridgland, A., Cowie, A., Meyer, C., Laydon, A., Velankar, S., Kleywegt, G. J., Bateman, A., Evans, R., Pritzel, A., Figurnov, M., Ronneberger, O., Bates, R., Kohl, S. A., Potapenko, A., Ballard, A. J., Romera-Paredes, B., Nikolov, S., Jain, R., Clancy, E., Reiman, D., Petersen, S., Senior, A. W., Kavukcuoglu, K., Birney, E., Kohli, P., Jumper, J., and Hassabis, D. Highly accurate protein structure prediction for the human proteome. *Nature*, 596:590–596, 8 2021. ISSN 14764687. doi: 10.1038/s41586-021-03828-1.

Weininger, D. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences*, 28:31–36, 1988.

Yang, J., Roy, A., and Zhang, Y. Biolip: a semi-manually curated database for biologically relevant ligand–protein interactions. *Nucleic acids research*, 41(D1):D1096–D1103, 2012.

Appendix

A. Model details

A.1. Training

Given a network parametrised by weights θ, the loss is defined as the cross-entropy between the input molecule x' (text-based tokenized) and the target enzyme sequence $y = (y_1, \ldots, y_T)$ (amino acid tokens):

$$\mathcal{L}(\theta) = -\sum_{t=1}^{T} \log p_\theta\big(y_t \mid y_{<t}, x'\big),$$

where $p_\theta(y_t \mid y_{<t}, x')$ is the probability assigned by the model to token y_t given previous target tokens $y_{<t}$ and the encoder output.

The main model was trained for 30 epochs with unique-input sampling. Hyperparameters consisted of: $2 \times 10e^{-4}$ learning rate, cosine decay to $2 \times 10e^{-5}$ (through 5 epochs), 3000 warm-up steps (3.3% of the total), 1152 batch size, and bf16 mixed precision. Alternative models reused these hyper-parameters unless otherwise stated. The training hardware consisted of 8 H100 Nvidia GPUs with 64Gb VRAM.

A.2. Tokenizer details

Our datasets represent ligands as SMILES strings while proteins are represented using the standard 20–amino-acid alphabet. To avoid ambiguity between chemical symbols and amino-acid letters (e.g., “C” denoting both carbon and cysteine), we define two separate vocabularies and use a dedicated tokenizer for each modality. Concretely, if the ligand tokenizer defines n tokens, we offset the indices of the protein tokenizer by n, ensuring that the two token spaces do not overlap.

The amino acid tokenizer has a token for each possible amino acid. In contrast, the SMILES tokenizer uses a ”word-piece” tokenization algorithm, as the space of input characters and ”chemical words” is much larger.

When tokenizing chemical notation, one can simply define a token per unique character in the dataset. Still, LLM literature has demonstrated how grouping tokens that frequently appear together improves computational efficiency and generalisation. That is because a single token prediction will yield a larger amount of information and because the model is allowed to have a dedicated embedding for combinations of characters, starting from a larger library of unique concepts. As described in previous work (Leon et al., 2024), applying tokenization algorithms such as BPE directly to SMILES, can generate invalid tokens, as atom notations such as ”He” cannot be split as ”H” + ”e”. Therefore, as done in (Leon et al., 2024) with their APE tokenizer, we define a 2 stage algorithm, which first defines indivisible atom tokens, and then performs the BPE algorithm to join those that frequently co-occur. We name our algorithm ABPE, as it fully reproduces the original BPE algorithm for the second stage, unlike the APE algorithm.

A.3. Architectures

Table A.1 lists the architectural details of all models used in this work, differentiating encoder and decoder parameters, and specifying the size of the language modelling (LM) head, which depends on the vocabulary size.

Table A.1. Model architecture summary. For encoder–decoder models, we report encoder and decoder hyperparameters; when a value is identical for encoder and decoder we list it once, otherwise we use encoder|decoder (e|d). P_{enc} and P_{dec} denote the number of parameters (in millions) in the encoder/decoder, P_{LM} in the output head for a vocabulary size of 1069, and P_{tot} for the total.

MODEL	P_{enc} (M)	P_{dec} (M)	P_{LM} (M)	P_{tot} (M)	L	H	d_{model}	d_{ff}	d_{kv}
T5-TINY	3.1	4.2	0.27	7.62	4	4	256	1024	64
T5-BASE	85.0	113.3	0.82	199.05	12	12	768	3072	64
T5-LARGE	302.0	402.7	1.09	705.86	24	16	1024	4096	64
GPT2 (DECODER-ONLY)	–	14.2	0.41	14.61	8	6	384	1536	64
T5-BASE ENC + LLAMA DEC	85.0	146.8	1.09	232.87	12|10	12|16	768|1024	3072|2048	64

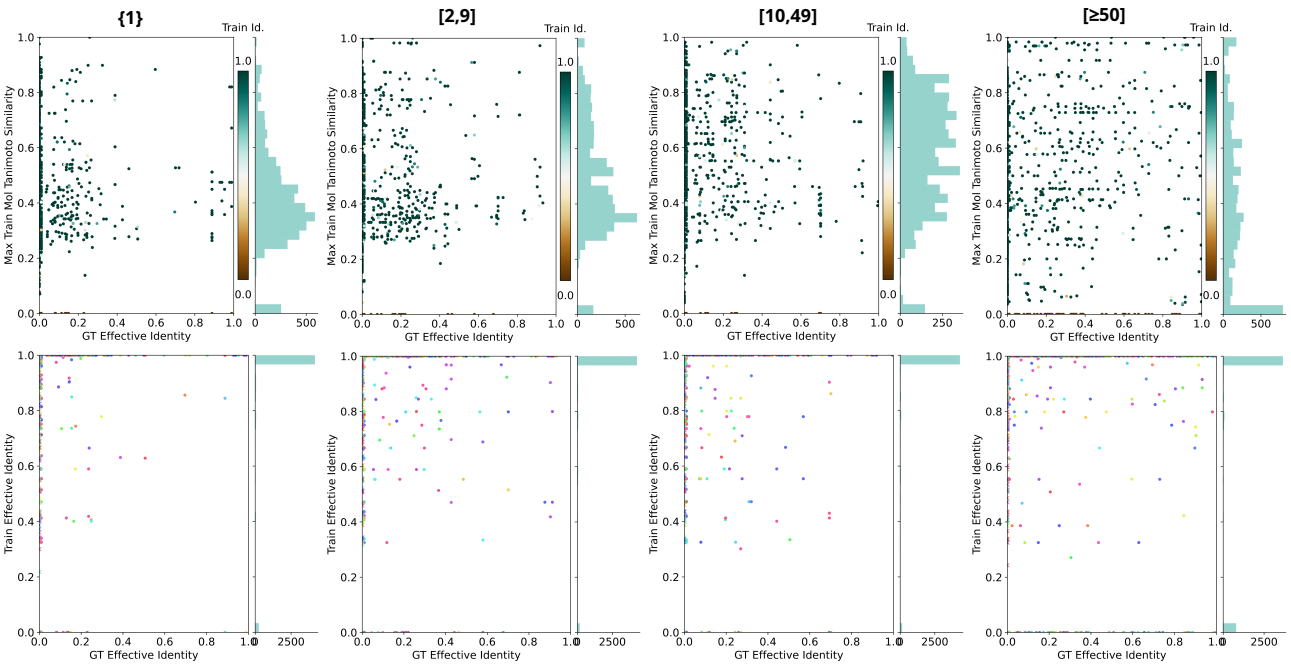

Figure A.1. Ligand novelty graphs per test split.

B. Complete novelty results

Figure A.1 provides the full results for GT identity vs MTS and Train identity. Notice how the MTS distribution becomes most widespread (and furthest from training) for the $[\geq 50]$ split. This is consistent with intuition, as the most promiscuous ligands are most likely to be associated to widely different proteins, giving the model the opportunity to bind them with protein families which are not specialised in this specific chemical profile.

B.1. Protein Novelty

Regarding protein novelty, Figure A.1 Table A.2 presents the full results with Train Id. for each split.

C. Extended–dataset results

Additionally, we also test results on the combination of the main binder dataset with the substrate–enzyme one. This combined dataset contains, for training, ∼1.8M unique ligands, ∼4.2M unique proteins and ∼13M total pairs, averaging 6.9 proteins per molecule. To avoid the most promiscuous ligands from overshadowing dataset statistics, the top 0.25% most common ones were removed, reducing the size to ∼1.8M unique ligands, ∼61K unique proteins and ∼5M pairs, averaging 3.3 proteins per molecule. Table A.3 shows how results converge back to retrieval behaviour, achieving largely similar scores to the Binder–Dataset.

D. Maximally distant test set construction

We first run an all-to-all MMseqs2 search over dataset proteins and define an *effective identity* between proteins as $\mathrm{eff}(p_i, p_j) = \mathrm{id}(p_i, p_j) \cdot \frac{\ell_{\mathrm{aln}}}{\max(\ell_i, \ell_j)}$, where $\mathrm{id} \in [0, 1]$ is the MMseqs2 identity, ℓ_{aln} the alignment length, and ℓ_i, ℓ_j the protein lengths. Then, two proteins are considered connected if $\mathrm{eff}_{i,j} \geq \tau$, with threshold $\tau = 0.85$. When a protein is included in the test set, all connected proteins are also included, along with all their associated molecules. If the molecules are associated to other proteins, these are also included, causing a cascading effect. The algorithm ranks molecules by an estimated novelty score (average identity to the dataset scaled by number of connected neighbours) and iteratively adds connected sets. Cascades are killed when surpassing target test size, with examples at the boundary of the cascade not included, in order to prevent training leak. MMseqs2 search was performed with minimum coverage of 0.7, minimum

Table A.2. Per-stratum protein novelty statistics (MMSeqs2 similarity to the training set) together with pLDDT and GT retrieval. "Train Id. | matches" reports mean±std identity over generations with a valid MMSeqs hit, and the percentage of generations with any hit.

	Split range	Mean±std pLDDT	Train Id. | matches	GT MMSeqs Acc.
Ligand Sampling				
Train	[50, 24570]	77.67±15.14	1.00±0.01 | 90.8%	78.37%
	[10, 49]	79.05±12.23	1.00±0.01 | 98.8%	92.46%
	[2, 9]	79.70±11.27	1.00±0.00 | 98.7%	72.51%
	{1}	79.91±10.49	1.00±0.01 | 98.8%	46.34%
	SAIR	79.07±10.68	1.00±0.01 | 98.4%	38.31%
Test	[50, 346253]	75.94±18.35	1.00±0.02 | 84.8%	46.17%
	[10, 49]	79.12±12.78	1.00±0.01 | 97.7%	90.89%
	[2, 9]	79.75±12.05	1.00±0.01 | 97.7%	51.29%
	{1}	78.41±13.14	1.00±0.01 | 95.9%	32.86%
	SAIR	79.36±10.42	1.00±0.01 | 98.2%	38.82%
Pair Sampling				
Train	[50, 24570]	65.66±24.33	0.97±0.13 | 55.5%	72.57%
	[10, 49]	77.79±14.31	1.00±0.03 | 94.1%	93.50%
	[2, 9]	77.24±15.25	1.00±0.03 | 95.8%	45.77%
	{1}	77.14±15.17	1.00±0.03 | 95.0%	17.50%
Test	[50, 346253]	55.11±26.19	0.97±0.13 | 52.4%	37.39%
	[10, 49]	76.49±16.37	1.00±0.03 | 93.4%	89.94%
	[2, 9]	76.59±16.34	1.00±0.03 | 92.6%	34.53%
	{1}	73.76±19.25	1.00±0.05 | 87.3%	9.49%

Table A.3. Results on Extended–Dataset: Main and substrate–enzyme. We report pLDDT, MMSeqs identity with percentage of matched proteins, and GT MMSeqs performance. SAIR test set is used instead of range-based ones (which maximised distance to train and were not as representative).

	Split range	pLDDT	Train Id. | matches	GT MMSeqs Acc.
Train	[50, 134]	77.98±14.72	1.00±0.02 | 89.9%	79.52%
	[10, 49]	79.25±11.71	1.00±0.01 | 97.8%	90.84%
	[2, 9]	79.65±10.92	1.00±0.01 | 98.5%	65.98%
	{1}	78.77±11.30	1.00±0.00 | 97.9%	48.34%
	SAIR-400	79.35±10.33	0.99±0.00 | 98.7%	47.00%
Test	SAIR-400	70.62±22.78	0.99±0.02 | 66.6%	31.24%

alignment length of 50 and minimum sequence identity of 0.85. Using the aforementioned algorithm, a test set of 18,665 molecules was built. Additionally, we also created a test set of 400 molecules extracted from the SAIR dataset, which provides residue contact annotation, allowing to differentiate the binding pocket. For this set we do not maximise distance to training, instead, we simply ensure that its molecules are not present during training.

E. Pocket-only model

As a proof of concept, we also train a model to generate only the pocket region.

For this we leverage BioLip's residue contact annotation. Using the same algorithm employed for SAIR, we estimate pocket sub-sequences in the amino acid chains, constructing a database of 34K unique ligands, 142K unique pocket sequences, and a total of 252K pairs. We deem appropriate to perform our experiment with this set rather than SAIR one, given that the average GT per ligand is $\sim$5, as opposed to SAIR's $\sim$1. Notice that this database is orders of magnitude smaller than the Binder–dataset, given the scarcity of contact / pocket annotation. Consequently, the aim is not to achieve competitive performance, but to evaluate the learnability of the problem.

Results in Table A.4 show an average pLDDT for $[2, 9]$ and $\{1\}$ almost achieving the 70 cut-off, which is often used as reference for confident folding, and lower values for the more promiscuous strata. Regarding GT accuracy, it is substantially high for the least annotated ones in training, but falls in the test set, a sign of overfitting, most likely due to dataset size. Still, from this we extract that it is possible to associate the pocket sequences to the ligands they bind, but becomes harder when multiple pockets can accommodate that same ligand, as memorisation is less straight forward. With the current evidence we cannot claim whether this approach will still overfit to training with a larger database or, on the contrary, will become comparable to full protein approaches.

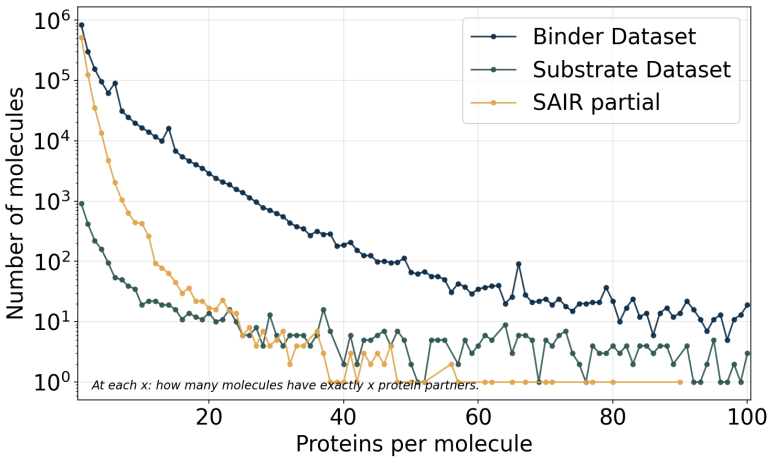

Figure A.2. Distribution of protein partners per molecule. For each value of proteins per molecule (x-axis), the y-axis shows how many molecules have exactly that many unique protein partners. Curves are shown for the Binder Dataset, Substrate Dataset, and SAIR partial (log scale; x capped at 100).

Table A.4. Results for pocket–only training.

	Split range	pLDDT	Train Id. | matches	GT MMSeqs Acc.
Train	[50, 20296]	49.29±16.86	0.99±0.07 | 13.6%	28.32%
	[10, 49]	57.54±17.72	0.99±0.06 | 34.3%	36.25%
	[2, 9]	67.59±15.12	1.00±0.04 | 75.2%	70.42%
	{1}	70.02±13.66	1.00±0.03 | 85.5%	76.82%
Test	[2, 9]	66.89±16.08	0.99±0.05 | 70.3%	15.10%
	{1}	68.03±15.00	0.99±0.04 | 74.3%	16.74%