ENZYMECOT: DE NOVO ENZYME DESIGN AS A CHAIN OF REASONING COUPLED TO A CHAIN OF STRUCTURE

Anonymous authors
Paper under double-blind review

ABSTRACT

Enzyme design aims to produce a sequence and a structure that together catalyse a specified reaction. Existing conditional generators are given either a compact functional label, from which every decision is then made implicitly inside the sampler, or structural information such as catalytic sites or a motif, which de novo design is meant to produce rather than consume. Neither writes the design down in steps. We propose EnzymeCoT, which runs two coupled chains. From a description of the reaction and a target length, a language model writes a *chain of reasoning*, the four remaining states of a five-state design record, moving from a mechanistic prior through a catalytic core and an active site to a backbone and a sequence. Each state names a wider part of the molecule than the last, and a masked discrete diffusion model over joint structure and sequence tokens is reconditioned on it and asked to commit that part alone, so the written chain drives a *chain of structure* through the same five states. Where chain-of-thought in a language model feeds only the language model, each state here conditions the generator directly, and at two depths: residue–position pairs at the input embedding, and the full token sequence by cross attention late in the denoiser. The corpus pairs enzymes with staged design rationales: deterministic code produces every claim and a verifier checks every sentence against it. Conditioned on the reaction and a target length, EnzymeCoT improves by 2.2 points of EC3 agreement on the strongest baseline evaluated under a specification that does not name the target class, and refolds closer to the reference backbone.

1 INTRODUCTION

Enzyme design aims to engineer sequences that catalyse a chosen chemical transformation, enabling biocatalysts for reactions with no natural counterpart, metabolic pathways assembled from parts, and therapeutics whose activity is specified rather than discovered (Munsamy et al., 2022; 2024; Hua et al., 2024). Because an enzyme is defined by what it does, the natural specification for such a design is functional, and recent generative models take exactly such a specification as their condition.

What those models are given falls into two families (Figure 1b,c), beside generators that take no functional specification at all (Figure 1a). The first conditions on a compact functional label: Munsamy et al. (2022) generate sequences from an EC number, Hua et al. (2024) take a catalytic reaction as input, and Chen et al. (2026) take a set of functional labels and derive the structural prior they use from those labels alone. The second is primed with structural information alongside the functional specification: Song et al. (2024) generate sequence and coordinates from functionally important sites and the substrate, Lin et al. (2026) take function-conserved sites together with an EC number and substrate atoms, and Yin et al. (2025) condition on function, structure and sequence modalities jointly. Systems conditioned on free text (Liu et al., 2025; Wang et al., 2023; Lv et al., 2024; Dai et al., 2024; Song et al., 2025) avoid the second requirement, but consume the description in a single pass.

One limitation is shared by all of them: the reasoning that connects a specification to a molecule stays inside the sampler, where it cannot be inspected, corrected or reused, and when a design fails there is no intermediate object to blame. The second family carries a further one, that whatever structural information the model needs must be supplied from outside, which is what de novo design

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.

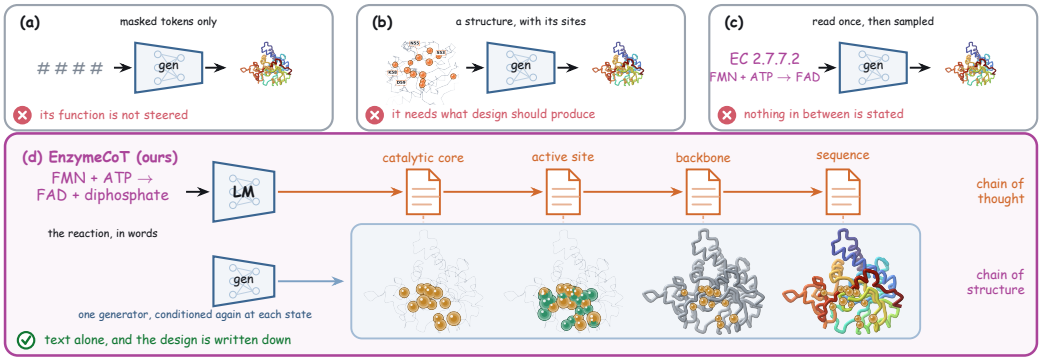

Figure 1: **The condition each family of generators is given.** **(a)** Unconditional generators (Yim et al., 2023b; Wu et al., 2024; Lin & AlQuraishi, 2023; Hesslow et al., 2022). **(b)** Generators primed with structural information (Lin et al., 2024; Song et al., 2024; Lin et al., 2026). **(c)** Generators conditioned on a compact label or on one block of text, read once (Munsamy et al., 2022; Hua et al., 2024; Madani et al., 2020; Yin et al., 2025; Dai et al., 2024; Chen et al., 2026). **(d)** EnzymeCoT is given the reaction in words, with a target length. A language model writes one state per stage, naming a wider part of the molecule each time, and the same generator is re-conditioned on each state in turn.

is supposed to produce. Nor does supplying more text by itself help if it arrives as one object: a description pooled into a single conditioning vector carries the enzyme class and little that identifies the record.

Our starting point is an observation from language modelling. A model asked for the answer to a hard multi-step problem answers it better when it is first allowed to write the intermediate steps, because each step conditions the next and because the written steps can be checked (Wei et al., 2022). Enzyme design admits a factorisation of exactly this kind, which we adopt as a modelling choice and not as a claim about biology: the chemistry constrains which residues must carry out the reaction, those residues constrain the pocket that holds the substrate, and the pocket constrains the fold that scaffolds it. Each step of that argument settles a wider part of the molecule than the one before, so the steps are also a nested sequence of structural commitments. A label-conditioned sampler makes all of these choices at once, silently, inside one denoising trajectory.

We therefore introduce EnzymeCoT, which carries that factorisation as two chains advancing together. The first runs in text. From a natural language description of the reaction and a target length, a language model writes a *design record* whose first state is the given specification and whose four written states move from a mechanistic prior, through a named catalytic core and a named active site, to a full backbone and finally a full sequence, each written from the states before it. We call this the *chain of reasoning*, a chain of thought in the sense of Wei et al. (2022); the term is used operationally, for a sequence of explicit textual design states, and carries no claim about latent reasoning. The second chain runs in the molecule. A masked discrete diffusion model over joint structure and sequence tokens is re-conditioned on each state in turn and asked to commit the part of the molecule that state names, so the region it has fixed grows from a few catalytic residues to a pocket, a backbone and a sequence. We call that nested sequence of commitments the *chain of structure*. Every link of the first chain produces a link of the second, which is where this differs from chain-of-thought in language models: there the intermediate states are fed back to the language model, here each one conditions a generator that emits structure and sequence. A state does not enter as one vector; it is decomposed into two conditions, each injected at the depth where the network can use it.

Training this requires a corpus of staged rationales. Deterministic code reads each predicted complex and emits a set of *claims*, a language model writes freely but only about those claims, and a verifier checks every sentence and returns it for rewriting with the reason attached, so every sentence is either model written and verified or a direct transcription.

The division of labour is what keeps that corpus small. Ge et al. (2026) fine-tune a 7B model to write a rationale and then the sequence itself, and extrapolate some 10^9 supervised pairs before text–sequence alignment reaches the natural level, while finding that the written plans improve readily.

We therefore never ask the language model for the molecule: it writes the record, a generator writes the protein, and $50\,420$ records train the pathway.

Conditioned on the reaction and a target length, EnzymeCoT improves EC3 agreement by 2.2 points on the strongest baseline given a specification that does not name the target class. Our contributions are summarised as follows.

- We propose EnzymeCoT, which couples a chain of reasoning in text to a chain of structure in the molecule: each written state commits a wider part of the molecule and re-conditions the generator, and the only inputs are a natural language description of the reaction and the length of the chain to be built.
- We propose a two-depth conditioning interface. Rather than hand a design record to the generator as one object, we write the residue–position pairs it names into the input embedding and read the sentences it is written in by cross attention late in the denoiser, so each part enters where the generator can act on it.
- We construct and release a corpus of 56 022 records paired with staged design rationales, in which every claim is produced by code and every sentence is verified.

2 Related work

Conditional protein and enzyme generation. Generators divide by what they model: amino acids (Madani et al., 2020; Hesslow et al., 2022; Lv et al., 2024; Alamdari et al., 2023; Wang et al., 2024b), backbone geometry with a sequence assigned afterwards (Trippe et al., 2022; Wu et al., 2024; Yim et al., 2023b; Lin & AlQuraishi, 2023; Yim et al., 2023a; Bose et al., 2023; Huguet et al., 2024; Lin et al., 2024; Gao et al., 2022), or both at once (Anand & Achim, 2022; Campbell et al., 2024; Wang et al., 2024c; Hsieh et al., 2025; Zhou et al., 2025; Eris, 2026), the setting we adopt. Function enters as a compact label, an EC number (Munsamy et al., 2022; 2024) or the catalysed reaction (Hua et al., 2024); Chen et al. (2026) build their structural prior from the label alone, so the label is still the only input. A second group is additionally primed with structure, which is what separates it from ours: functionally important sites and the substrate (Song et al., 2024; 2026), function-conserved sites with substrate atoms (Lin et al., 2026), atom-level catalytic geometry (Ahern et al., 2026), and function, structure and sequence jointly (Yin et al., 2025). Du et al. (2025) instead retrieve enzymes for a specified substrate and condition on what is retrieved. Adapters (Yang et al., 2024) and data augmentation (Ren et al., 2025) pursue the same control by other means.

Text and reasoning as the condition. Free text removes the structural requirement, and the work built on it consumes the description in one pass (Liu et al., 2025; Wang et al., 2023; Pei et al., 2023; Fang et al., 2023; Dai et al., 2024; Song et al., 2025; Truong & Bepler, 2025). Closest to us, Wu et al. (2026) lets a multimodal language model name functionally essential residues and hands them to a separate geometric generator. We share that decoupling and differ in three ways: our interface is two simultaneous channels rather than one hand-off, because a single hand-off loses most of the record; our record is staged, and the generator is re-conditioned at every stage; and we generate structure and sequence jointly. Rector-Brooks et al. (2026) reach de novo enzymes from the other direction, conditioning on the reactive intermediate as geometry to be bound and validating in the laboratory, and the design itself is never written down. Making the reasoning agentic (Ge et al., 2026; Teneggi et al., 2026; Rahman & Rahman, 2026; Fan et al., 2026) places it outside the generator, and aligning a generator to function (Wang et al., 2024a; 2025; Cao et al., 2025; Zhang et al., 2026; Xiong et al., 2025; Stocco et al., 2025) is complementary: it changes what a model prefers to generate, where conditioning changes what it is told about one target.

3 Design records

We construct a corpus of enzymes paired with staged design rationales. The pipeline never lets the model establish anything: deterministic code turns an all-atom Boltz-1x prediction of each holo complex into a set of checkable *claims*; a language model writes only about those claims; and a verifier checks every sentence back against them by nine rules, four of which make a residue,

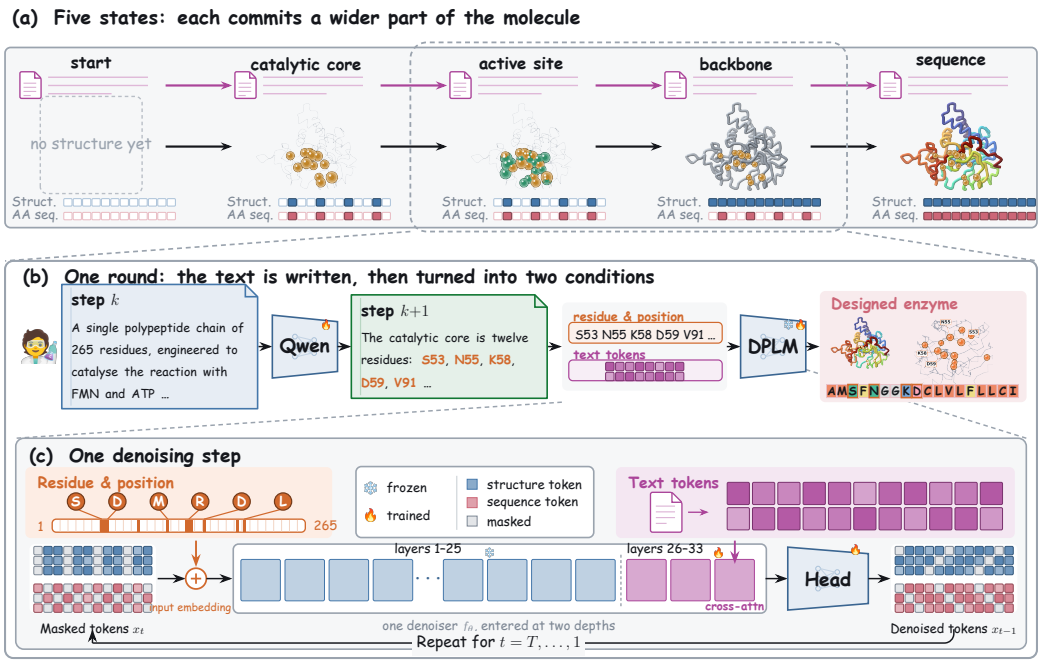

Figure 2: **The system.** **(a)** The five text states, one card each, with the structure each has committed to. **(b)** One round: the language model writes the next state, which becomes two conditions. **(c)** One denoising step, with the two conditions entering at two depths. Content is one record of the test split (EC 2.7.7.2, 265 residues).

a number, a subsequence or a metal the claims do not contain a rejection, while the rest forbid the restatements and the shortcuts that would let the model satisfy those four without reasoning. Every sentence in the corpus is therefore either model-written and verified, or a claim transcribed. Clustering by sequence leaves 50 420 training, 2 801 validation and 2 801 test records. Appendix A gives the claim schema, the four nested substructures, the causal contract between stages, the nine rules and the filters.

4 EnzymeCoT

Figure 2 shows the system.

4.1 Problem formulation

Joint discrete representation. A protein of length L is a pair $\mathcal{P} = (s, x)$ with sequence $s = [s_1, \ldots, s_L]$, $s_i \in \mathcal{V}_{\text{aa}}$, and backbone coordinates $x \in \mathbb{R}^{L\times 4\times 3}$. Following Wang et al. (2024c) we quantise the local structure around each residue into a discrete token, $z = [z_1, \ldots, z_L]$ with $z_i \in \mathcal{V}_{\text{st}}$, $|\mathcal{V}_{\text{st}}| = 2^{13}$, and model the two modalities in one sequence

$$u \;=\; \big[\, z_1, \ldots, z_L,\; s_1, \ldots, s_L \,\big] \;\in\; \mathcal{V}^{2L}, \qquad \mathcal{V} = \mathcal{V}_{\text{st}} \cup \mathcal{V}_{\text{aa}} \cup \{[\text{M}]\}. \tag{1}$$

The design record. A *design record* is $y = (y_1, \ldots, y_K)$ with $K = 5$. State y_1 states a mechanistic prior and commits to no coordinates; y_2 names the catalytic core, y_3 the active site around it, y_4 the backbone and y_5 the sequence. A language model g_ϕ writes each state from the states before it, $y_{k+1} = g_\phi(y_{\le k})$, so at inference only y_1 is given.

Absorbing forward process. We corrupt u by replacing tokens with the modality's mask symbol [M] (Sahoo et al., 2024; Lou et al., 2023). With $u^{(0)}$ the clean sequence and $u^{(t)}$ the state at step $t \in \{0, \ldots, T\}$, transitions are independent across positions,

$$q\big(u_i^{(t)} \mid u_i^{(t-1)}\big) \;=\; \text{Cat}\big(u_i^{(t)};\, u_i^{(t-1)} Q_t\big), \qquad Q_t \;=\; (1-\beta_t)\, I \;+\; \beta_t\, \mathbf{1}\, e_{[\text{M}]}^{\top}, \tag{2}$$

so that $u^{(T)}$ is fully masked. The denoiser f_θ predicts the clean token at every masked position. Sequence positions are ordinary $|\mathcal{V}_{\mathrm{aa}}|$-way classification. Structure positions are not: a 13-bit codebook makes a flat softmax over 8192 classes both wasteful and badly conditioned, so f_θ emits 13 independent logits per position and the token is recovered by

$$\hat{z}_i \;=\; \sum_{b=1}^{13} \mathbf{1}\Big[\sigma\big(f_\theta(u^{(t)})_{i,b}\big) > \tfrac{1}{2}\Big]\, 2^{\,b-1} \;+\; o_{\mathrm{st}}, \tag{3}$$

with o_{st} the offset of $\mathcal{V}_{\mathrm{st}}$ in the joint vocabulary. Training therefore uses a bitwise cross-entropy on the structure half and a token cross-entropy on the sequence half, and a structure token that is wrong in one bit costs one bit rather than everything.

4.2 The chain of structure

The record is a chain of structure rather than one long prompt because each state is paired with the part of the molecule it has committed to. Let $\mathcal{C} \subset \{1,\dots,L\}$ be the catalytic core and $\mathcal{S} \supset \mathcal{C}$ the active site. Round k reveals a structure set R_k^{st} and a sequence set R_k^{aa},

$$\big(R_k^{\mathrm{st}},\, R_k^{\mathrm{aa}}\big) \;=\; \begin{cases} (\mathcal{C},\, \mathcal{C}) & k=2,\\ (\mathcal{S},\, \mathcal{S}) & k=3,\\ (\{1..L\},\, \mathcal{S}) & k=4,\\ (\{1..L\},\, \{1..L\}) & k=5, \end{cases} \qquad R_1^{\mathrm{st}} = R_1^{\mathrm{aa}} = \emptyset, \tag{4}$$

so round 2 starts from a fully masked sequence. The input to the denoiser at round k keeps the previous round's reveal and masks the rest,

$$u_i^{[k]} \;=\; \begin{cases} u_i^{(0)} & i \in R_{k-1}^{\mathrm{st}} \cup \big(L + R_{k-1}^{\mathrm{aa}}\big),\\ [\mathrm{M}] & \text{otherwise.} \end{cases} \tag{5}$$

The asymmetry at $k=4$ is the point of the schedule: the backbone is completed while the sequence is still known only at the site, so round 5 assigns a sequence to a backbone that already exists rather than inventing both at once.

4.3 Two conditions, two depths

Let $h^{(\ell)} \in \mathbb{R}^{2L\times d}$ be the denoiser's hidden states at layer ℓ of $N=33$, and let $G_k \in \mathbb{R}^{n_k\times d_g}$ be the per-token final hidden states of the writer g_ϕ over the span of y_k; T is reserved for the number of denoising steps. The record enters at two depths.

Centring the text features. The hidden states of a decoder are strongly anisotropic: they concentrate around a large mean direction shared by every input, so the component that distinguishes one record from another is small next to the component all records have in common. Feeding G_k to cross-attention unmodified spends the adapter's capacity on that shared direction. A per-stage mean is subtracted before the features are used,

$$\tilde{G}_k \;=\; G_k - \mu_k, \qquad \mu_k \;\leftarrow\; (1-\eta)\,\mu_k \;+\; \eta\,\tfrac{1}{|B|}\sum_{r\in B} \overline{G_k(r)}, \tag{6}$$

where $\overline{G_k(r)}$ is the mean over the tokens of stage k in record r and B is the batch. The mean is per stage because the five states differ in register, and it is an exponential moving average because a mean estimated once expires as soon as g_ϕ is unfrozen and its representation starts to move. The cross-attention below reads $\tilde{G}_k$.

Residue and position → the input embedding. The record names residues in the compact one-letter form, a letter followed by a position, as in `S53`. An extractor scans y_k for that pattern, keeps a match only when the letter is one of the twenty amino acids and the position lies within the chain, converts to zero-based indexing and deduplicates, giving

$$\mathcal{R}_k \;=\; \big\{\,(a,i)\;:\; a \,\|\, (i{+}1) \text{ occurs in } y_k,\; a\in\mathcal{V}_{\mathrm{aa}},\; 0\le i < L\,\big\}, \tag{7}$$

with $a \,\|\, (i{+}1)$ the letter a followed by the decimal numeral $i{+}1$. Nothing here is learned and nothing is repaired, so the channel is a faithful reading of the record rather than a corrected one. The identities are *added* to the input embedding of the sequence half, never written over it, with $\pi(\mathcal{R}_k) = \{\, i : (a, i) \in \mathcal{R}_k \,\}$ the positions the record names and $\bar{E}_{\mathrm{st}}$ the mean structure embedding,

$$h^{(0)}_{L+i} \;\leftarrow\; h^{(0)}_{L+i} \;+\; \gamma_{\mathrm{aa}}\, \mathbf{1}\big[(a, i) \in \mathcal{R}_k\big]\, E_{\mathrm{aa}}[a], \qquad h^{(0)}_{i} \;\leftarrow\; h^{(0)}_{i} \;+\; \gamma_{\mathrm{st}}\, \mathbf{1}\big[i \in \pi(\mathcal{R}_k)\big]\, \bar{E}_{\mathrm{st}}, \quad (8)$$

with gates γ initialised at zero. Adding rather than substituting leaves the denoiser free to overrule a pair it disagrees with, which a hard write would not.

Text tokens → cross-attention in the upper layers. The record is kept as a sequence, never pooled, and read by cross-attention at the last 8 of the 33 layers,

$$h^{(\ell)} \;\leftarrow\; h^{(\ell)} \;+\; s_\ell\, W_o^{(\ell)}\, \mathrm{MHA}\big(\mathrm{LN}\, h^{(\ell)},\ \mathrm{LN}\, W_c \tilde{G}_k,\ \mathrm{LN}\, W_c \tilde{G}_k\big), \qquad \ell = N-7, \ldots, N, \quad (9)$$

where $W_c \in \mathbb{R}^{d \times d_g}$ projects the writer's width onto the denoiser's and s_ℓ is a per-layer read-out of how much text that layer uses, initialised at one, and the adapter's output projection $W_o^{(\ell)}$ is initialised at zero, so step 0 reproduces the pretrained denoiser exactly. Only one of the two is zeroed: the gradient on each is proportional to the other, so zeroing both would leave the branch unable to open. The adapter can only be opened by gradient (Zhang et al., 2023). The two channels differ in what they can express: the first is exact and positional but carries only what a regular expression can name; the second carries the whole record but must compete with the denoiser's own representation.

A third channel, discarded. A global channel is available: pool the same text, classify it into EC classes, and add the posterior as a low-rank residual on the output logits, shared by every position. It can say *what* the enzyme is but, by construction, not *where* anything goes, and measured against the two channels above it costs accuracy during training: a signal available at every position that carries no positional content competes with the two that do.

4.4 Inference

Generation is the loop of Algorithm 1. Its only inputs are y_1 and the target length L, which fixes the width $2L$ of the masked sequence the loop starts from and is given to every method we compare against.

Algorithm 1 Staged generation from y_1 and the target length L. Round k updates only M_k, the positions that round asks for; everything else is held.

1: $u \leftarrow [\textsc{m}]^{2L}$ ▷ nothing is committed yet
2: **for** $k = 2, \ldots, K$ **do**
3: $\quad y_k \leftarrow g_\phi(y_{<k})$ ▷ write the next state
4: $\quad \mathcal{R}_k \leftarrow \textsc{ExtractPairs}(y_k)$; $\quad G_k \leftarrow \textsc{Hidden}(g_\phi, y_k)$
5: $\quad u \leftarrow \textsc{Reveal}(u,\, R^{\mathrm{st}}_{k-1},\, R^{\mathrm{aa}}_{k-1})$ ▷ keep round $k{-}1$, mask the rest
6: $\quad M_k \leftarrow \big(R^{\mathrm{st}}_k \cup (L{+}R^{\mathrm{aa}}_k)\big) \setminus \mathrm{obs}(u)$ ▷ this round's positions
7: $\quad$ **for** $t = T, \ldots, 1$ **do**
8: $\quad\quad u_{M_k} \leftarrow \textsc{Denoise}\big(f_\theta,\, u,\, \mathcal{R}_k,\, G_k\big)$ ▷ two conditions, two depths
9: $\quad$ **end for**
10: **end for**
11: **return** $(\hat{z}, \hat{s})$ ▷ a structure and a sequence, jointly

Two properties of this loop follow from its definition. First, the record is never given to the denoiser as a whole: round k is conditioned on y_k alone, so the conditioning stays the size of one state instead of growing with the record. That localises what each round *reads* but not what an error costs, since a wrong state still propagates through the states written after it and through the reveal it produced. Second, the two chains are carried by different objects: line 3 advances the chain of reasoning, line 5 advances the chain of structure, and the two are consistent by construction *in training*, where y_k was written from the same commitments the reveal encodes. At inference they can drift: the writer does not read the round it conditioned, so a state and the structure produced from it agree only as far

as the generator made them agree. The loop is therefore open on the text side, and we report it as staged planning rather than as reasoning that observes its own output.

4.5 TRAINING

The denoiser optimises the masked cross-entropy at the positions round k asks for, weighted per round, and the language model optimises the next-state text loss,

$$\mathcal{L}_{\text{den}} = \sum_{k=2}^{K} w_k \, \mathbb{E}_{t,\, u^{(t)}} \Big[\omega(t) \sum_{i \in M_k} - \log p_\theta \big(u_i^{(0)} \,\big|\, u^{(t)}, u^{[k]}, \mathcal{R}_k, \tilde{G}_k \big) \Big],$$
$$t \sim \mathcal{U}\{t_{\min}, \ldots, T\}, \quad u^{(t)} \sim q(\cdot \mid u^{(0)}, u^{[k]}), \quad M_k = \big(R_k^{\text{st}} \cup (L + R_k^{\text{aa}}) \big) \setminus \text{obs}(u^{[k]}), \tag{10}$$

The corruption is applied only within M_k: positions revealed by $u^{[k]}$ are held, and positions outside the round are left masked, so neither is a target and neither is resampled. One t is shared by the two halves and is drawn above $t_{\min}$ rather than from the whole range, which concentrates training on the high-noise states where the text is the only thing left to read; $\omega(t) = 1 - (t-1)/T$ weights the sample. Within M_k the corruption is uniform, while inference reveals the structure in the staged order, and that mismatch is a known limitation of the present training scheme.

$$\mathcal{L}_{\text{text}} \;=\; \sum_{k=2}^{K} \sum_{t \in \text{span}(y_k)} c_t \big[- \log p_\phi \big(y_{k,t} \mid y_{k,<t},\, y_{<k} \big) \big], \qquad \mathcal{L} \;=\; \mathcal{L}_{\text{den}} \;+\; \lambda \, \mathcal{L}_{\text{text}}, \tag{11}$$

with $c_t > 1$ on the tokens that spell a residue identifier and 1 elsewhere. Those tokens are a small fraction of a stage, but they are the only part of the text the discrete channel can read, so an unweighted text loss lets the model write fluent prose that names the wrong residues at no cost.

Balancing the two losses. Three choices decide whether the two can be optimised together. First, the round weights w_k are renormalised to mean one, $w_k \leftarrow (K-1)\, w_k / \sum_{j=2}^{K} w_j$, so that changing their ratio changes what the model attends to without changing the effective learning rate. Second, the two losses are computed over different vocabularies and differ by an order of magnitude, so λ is calibrated once, at the first step on which both are live, from the ratio of their gradient norms,

$$\lambda \;=\; \big\| \nabla_\theta \mathcal{L}_{\text{den}} \big\| \big/ \big\| \nabla_\phi \mathcal{L}_{\text{text}} \big\|, \tag{12}$$

and then pinned, so that a later change in either loss cannot silently rescale the other. Third, training corrupts the revealed context: with probability p_{ctx} a revealed position is re-masked before the round is built, because at inference round k is conditioned on round $k-1$'s own output and a model trained only on clean context has never seen the distribution it will meet.

Trained parameters. The lower layers of the denoiser are frozen and the upper eight are trained, together with the two conditioning modules and a low-rank adapter on g_ϕ. The language model is held fixed for an initial warm-up, since unfreezing it earlier makes the cross-attention chase a representation that is itself changing and the adapter never settles. Sizes, layer counts, adapter rank, the warm-up length and the values of w_k and p_{ctx} are given in Section 5.1.

5 EXPERIMENTS

5.1 SETUP

Model and training. The denoiser is DPLM-2 Bit 650M (Hsieh et al., 2025), a masked discrete diffusion model over joint structure and sequence tokens, with $N = 33$ layers, hidden width 1280 and 20 heads; the writer g_ϕ is Qwen3.5-9B, a decoder of the Qwen3 family (Yang et al., 2025). Layers 1 to $N - 8$ of the denoiser are frozen, and the last eight layers, the two conditioning modules and a LoRA of rank 32 on g_ϕ are trained: 160.8M in the denoiser, 70.8M in the conditioning modules and 58.2M in the adapter, 289.8M in total. The language model is frozen for the first 1000 steps. Round weights before renormalisation are $(1.5, 1.2, 1.0, 1.0)$ for $k = 2, \ldots, 5$, and the context corruption rate is $p_{\text{ctx}} = 0.15$. Training runs two epochs over the training split.

Table 1: Designs from a functional specification. Arrows give the direction of improvement.

Method	EC3 ↑	mic-F1 ↑	mac-F1 ↑	pLDDT ↑	TM ↑	MMD ↓
CFP-Gen (Yin et al., 2025)	44.3	0.323	**0.178**	85.1	0.526	0.169
CodeFP (Chen et al., 2026)	45.0	0.433	0.145	80.0	0.510	0.211
Pinal (Dai et al., 2024)	39.6	0.277	0.136	**88.5**	0.422	0.153
DPLM (Wang et al., 2024b)	30.7	0.245	0.139	83.3	0.423	**0.127**
EnzymeCoT	**47.2**	**0.451**	0.138	87.3	**0.555**	0.210
natural sequences	98.7	0.876	0.771	90.6	0.917	0.070

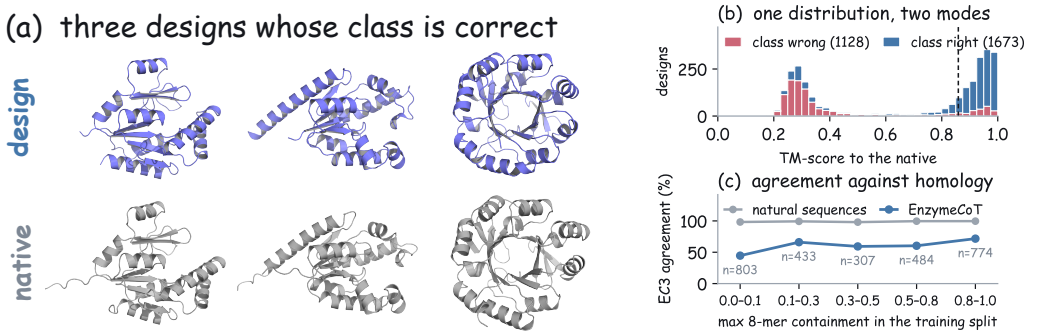

Figure 3: **(a)** Three designs whose predicted class matches the target, each above the natural sequence of the same record; left to right, EC 2.7.4, 2.7.1 and 5.3.1, of 191, 186 and 242 residues, all at TM 0.913 to the native. **(b)** TM to the native over all 2,801 test records, split by whether the predicted class was the requested one; the dashed line is the median of the whole set. **(c)** EC3 against the maximum 8-mer containment of a test record in the training split, with the natural sequences on the same axis.

Metrics. Enzyme class is predicted from sequence by CLEAN (Yu et al., 2023) under its maximum-separation rule, and EC3 agreement is the fraction of designs whose predicted class matches the target to three levels of the EC hierarchy. It is a predictor's agreement, not a measurement of catalysis: no substrate is docked, no active-site geometry is scored and no turnover is established anywhere in this paper. The F1 scores are multi-label over full EC numbers, on the union of true and predicted classes rather than the truth alone. pLDDT is ESMFold's confidence for the refolded design and TM is that structure against the reference backbone, normalised by the reference. MMD is the 3-mer spectrum distance to the natural sequences of the same records; a split of those sequences against itself gives 0.070, a reference rather than a bound. Each baseline is given the strongest specification it supports that does not name the target class: GO terms with InterPro domains for CFP-Gen, GO terms for CodeFP, a thirty-residue native prefix for DPLM, and the reaction text for Pinal. Generation is free running throughout: round k is conditioned on the model's own round $k-1$, never on the true one.

5.2 Main results

Table 1 reports every arm on the same records, at each record's own length and under the same predictors. EnzymeCoT leads on the two columns that ask whether the design is the one that was requested: EC3 47.2 against 45.0 for the next arm, and TM 0.555 against 0.526. From a description of the reaction and a target length, that is ahead of the three arms given a richer specification, and of Pinal, which is given the same reaction text.

Where it does not lead is informative. Pinal is the most confidently folded of the five generated arms (pLDDT 88.5) and the furthest from the reference backbone (TM 0.422): a description can be read fluently enough to fold and still not place the chain. DPLM, handed a thirty-residue native prefix, folds at 83.3 and names the requested class least often (30.7), while its 3-mer spectrum is the closest to natural of any generated arm (0.127), so a prefix buys residue statistics rather than function.

Table 2: Removing one part of the method at a time, on the records of Table 1.

	EC3 ↑	mic-F1 ↑	mac-F1 ↑	pLDDT ↑	TM ↑	MMD ↓
EnzymeCoT	**47.2**	**0.451**	**0.138**	**87.3**	**0.555**	**0.210**
no chain of reasoning	44.1	0.412	0.127	82.5	0.478	0.215
no continuous channel	42.7	0.408	0.118	81.6	0.463	0.228

Macro-F1 is where EnzymeCoT trails, 0.138 against CFP-Gen's 0.178: it averages over classes rather than records, so the gap sits in the long tail. Natural sequences give the evaluator's reference performance, 98.7 EC3, 90.6 pLDDT and 0.917 TM.

Figure 3a shows three designs whose predicted class matches the target; they are median-representative rather than best: the fold is recovered closely enough for the class predictor to read the requested chemistry off it, at median identity 0.078. Figure 3b shows that 57.4% of designs reach TM 0.7 and that the successful mode is sharp, median 0.931 against 0.308 where the class is wrong. Figure 3c places that against homology: agreement is lowest in the least homologous bin and highest in the most homologous one, running from 44.6 at the low-containment end to 71.7 at the high one without being monotone in between, while the natural sequences stay flat at 98.3 to 99.6, so the difference across bins is unlikely to come from the predictor.

5.3 Ablations

Table 2 removes one part at a time. Both removals cost on every column: pLDDT falls 4.8 and 5.7 points, TM 0.077 and 0.092, and EC3 3.1 and 4.5 points. The continuous channel is the more expensive to remove, since dropping the staging delivers the record once but intact, while dropping the cross-attention path keeps the schedule and reduces the record to the pairs a regular expression can extract. The stripped model at 42.7 sits between DPLM's 30.7 and CodeFP's 45.0.

6 Limitations

Although EnzymeCoT gives the generator a written design to condition on at every state, it has two limitations. First, the loop is open on the text side: the writer produces state k from the states before it and never reads the structure that round $k-1$ generated, so the two chains can drift apart, and a commitment made early is not revised when a later round contradicts it. Closing the loop needs a writer that reads structure, and the writer here is a general language model with a low-rank adapter, never pretrained or aligned against protein sequence and structure; that missing alignment also bounds what the continuous channel can carry, and the centring of Section 4.3 offsets only part of it. Second, a design costs four rounds of denoising and four language-model generations against one sampler pass for a label-conditioned generator, and the rounds cannot be run in parallel, since each is conditioned on the state written before it. Pretraining the writer against protein sequence and structure would let it read the round it conditioned and would loosen that bound; we leave it to future work.

7 Conclusion

We presented EnzymeCoT, which couples a *chain of reasoning* in text to a *chain of structure* in the molecule: a language model writes a design record whose states each name a wider part of the molecule, and one generator is re-conditioned on each in turn and commits that part. A record delivered as one object is read only in part, so it is decomposed and injected where each part can be used, and the corpus such a model needs can be built without a single unverified sentence. Conditioned on the reaction and a target length, it leads the conditional baselines on EC3 agreement and on the reference backbone, and recovers a fold or misses it outright rather than degrading smoothly.

AI use statement

We declare two uses of generative AI tools in this work, both described in the paper.

The first is for generating synthetic datasets. The design records that make up the corpus of Appendix A are written by a language model, which is a component of the method rather than an aid to the authors: the model writes each record under a verifier that rejects any residue, number, subsequence or metal absent from the deterministic claims, and the claim schema, the nine verifier rules, the rewrite policy and the counts at each stage are given there in full.

The second is to aid or polish writing. We used generative AI tools to edit prose in this paper and to write routine code, and that code was read and tested before use.

We have not used generative AI tools to generate research ideas, to design experiments, to analyse results, or to produce any number, table or figure reported here; every value was computed from the measurements described in Section 5.1 and checked against them by the authors. We have reviewed all AI-assisted work, and we take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

Ethics statement

This work generates enzyme sequences and backbones from a description of a reaction. The same capability could be directed at enzymes acting on hazardous substrates, and a reaction description is an unusually low barrier to specifying such a target, since no structure, motif or catalytic geometry is needed to pose the request. Three properties of the present system bound that risk without removing it. The designs are not validated: nothing here establishes substrate binding, active-site geometry or turnover, and the agreement we report is a class predictor's, so a design that scores well is a candidate for testing and not a working enzyme. The corpus is built from enzymes already annotated in UniProt and carries no information about reactions absent from the public literature. And the writer is auditable, so a request and the plan it produces can be read before any structure is generated, which is the point at which a screen would apply.

We will release the corpus and the code under terms requiring the same screening of reaction requests that we apply ourselves, and we will not release generated designs in bulk. The screening question is not settled by those measures, and a text-conditioned interface makes it more pressing than a structure-conditioned one.

Reproducibility statement

The generative model, the two conditioning operators and the training objective are stated in Section 4, with the inference loop given as Algorithm 1. Section 5.1 gives the model sizes, the trained parameters, the round weights, the context corruption rate, the class predictor and the folding model used for every structural column. The corpus is described end to end in Appendix A: the claim schema, the four nested substructures, the causal contract between stages, the nine verifier rules and the filters, with the counts at each stage. Appendix B gives the composition of the dataset and the clustered split, including the homology band reported separately. Code, the corpus and the evaluation scripts will be released.

References

Woody Ahern, Jason Yim, Doug Tischer, Saman Salike, Seth M. Woodbury, Donghyo Kim, Indrek Kalvet, Yakov Kipnis, Brian Coventry, Han Raut Altae-Tran, Magnus S. Bauer, Regina Barzilay, Tommi S. Jaakkola, Rohith Krishna, and David Baker. Atom-level enzyme active site scaffolding using RFdiffusion2. *Nature Methods*, 23(1):96–105, 2026. doi: 10.1038/s41592-025-02975-x.

Sarah Alamdari, Nitya Thakkar, Rianne van den Berg, Neil Tenenholtz, Robert Strome, Alan M. Moses, Alex X. Lu, Nicolò Fusi, Ava P. Amini, and Kevin K. Yang. Protein generation with evolutionary diffusion: sequence is all you need. *bioRxiv*, 2023. doi: 10.1101/2023.09.11.556673.

Namrata Anand and Tudor Achim. Protein Structure and Sequence Generation with Equivariant Denoising Diffusion Probabilistic Models. *arXiv preprint arXiv:2205.15019*, 2022.

Avishek Joey Bose, Tara Akhound-Sadegh, Guillaume Huguet, Kilian Fatras, Jarrid Rector-Brooks, Cheng-Hao Liu, Andrei Cristian Nica, Maksym Korablyov, Michael Bronstein, and Alexander

Tong. SE(3)-Stochastic Flow Matching for Protein Backbone Generation. In *International Conference on Learning Representations*, 2023. doi: 10.48550/arXiv.2310.02391.

Andrew Campbell, Jason Yim, Regina Barzilay, Tom Rainforth, and Tommi Jaakkola. Generative Flows on Discrete State-Spaces: Enabling Multimodal Flows with Applications to Protein Co-Design. In *International Conference on Machine Learning*, 2024. doi: 10.48550/arXiv.2402.04997.

Hanqun Cao, Hongrui Zhang, Junde Xu, Zhou Zhang, Lingdong Shen, Minghao Sun, Ge Liu, Jinbo Xu, Wu-Jun Li, Jinren Ni, Cesar de la Fuente-Nunez, Tianfan Fu, Yejin Choi, Pheng-Ann Heng, and Fang Wu. From Supervision to Exploration: What Does Protein Language Model Learn During Reinforcement Learning? *arXiv preprint arXiv:2510.01571*, 2025. doi: 10.48550/arXiv.2510.01571.

Xinrui Chen, Yizhen Luo, Siqi Fan, and Zaiqing Nie. Co-Generative De Novo Functional Protein Design. *arXiv preprint arXiv:2605.00948*, 2026. doi: 10.48550/arXiv.2605.00948.

Fengyuan Dai, Shiyang You, Yudian Zhu, Yuan Gao, Lihao Fu, Xibin Zhou, Jin Su, Chentong Wang, Yuliang Fan, Qingsong Tan, Xiaoxiao Ma, Xianjun Deng, Letong Yu, Hui Qian, Yan He, Yitao Ke, Zhe Zhang, Chenchen Han, Xing Chang, Liangzhen Zheng, Sheng Wang, Yuan Ji, Kexin Hao, Longxing Cao, Yajie Wang, Anping Zeng, Shunzhi Wang, Chunjun Zhan, Jing Zhao, Tong Si, Jianming Liu, Hongyuan Lu, and Fajie Yuan. Programmable design of functional proteins from natural language. *bioRxiv*, 2024. doi: 10.1101/2024.08.01.606258.

Jiahe Du, Kaixiong Zhou, Xinyu Hong, Zhaozhuo Xu, Jinbo Xu, and Xiao Huang. Retrieval augmented zero-shot enzyme generation for specified substrate. In *International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 14719–14734, 2025.

Furkan Eris. No Generation without Representation: Efficient Causal Protein Language Models Enable Zero-Shot Fitness Estimation. *arXiv preprint arXiv:2602.01845*, 2026.

Chuanliu Fan, Zicheng Ma, Huanran Meng, Aijia Zhang, Wenjie Du, Jun Zhang, Yi Qin Gao, Ziqiang Cao, and Guohong Fu. Interleaved Tool-Call Reasoning for Protein Function Understanding. *Annual Meeting of the Association for Computational Linguistics*, 2026. doi: 10.48550/arXiv.2601.03604.

Yin Fang, Xiaozhuan Liang, Ningyu Zhang, Kangwei Liu, Rui Huang, Zhuo Chen, Xiaohui Fan, and Huajun Chen. Mol-Instructions: A Large-Scale Biomolecular Instruction Dataset for Large Language Models. *arXiv preprint arXiv:2306.08018*, 2023.

Zhangyang Gao, Cheng Tan, Pablo Chacón, and Stan Z. Li. PiFold: Toward effective and efficient protein inverse folding. *arXiv preprint arXiv:2209.12643*, 2022.

Yutang Ge, Guojiang Zhao, Sihang Li, Zheng Cheng, Zifeng Zhao, Hanchen Xia, Guolin Ke, Linfeng Zhang, Zhifeng Gao, and Yuguang Wang. ProtoCycle: Reflective Tool-Augmented Planning for Text-Guided Protein Design. *arXiv preprint arXiv:2604.16896*, 2026.

Daniel Hesslow, Niccoló Zanichelli, Pascal Notin, Iacopo Poli, and Debora Marks. RITA: a Study on Scaling Up Generative Protein Sequence Models. *arXiv preprint arXiv:2205.05789*, 2022.

Cheng-Yen Hsieh, Xinyou Wang, Daiheng Zhang, Dongyu Xue, Fei Ye, Shujian Huang, Zaixiang Zheng, and Quanquan Gu. Elucidating the Design Space of Multimodal Protein Language Models. In *International Conference on Machine Learning*, 2025. doi: 10.48550/arXiv.2504.11454.

Chenqing Hua, Jiarui Lu, Yong Liu, Odin Zhang, Jian Tang, Rex Ying, Wengong Jin, Guy Wolf, Doina Precup, and Shuangjia Zheng. Reaction-conditioned De Novo Enzyme Design with GENzyme. *arXiv preprint arXiv:2411.16694*, 2024.

Guillaume Huguet, James Vuckovic, Kilian Fatras, Eric Thibodeau-Laufer, Pablo Lemos, Riashat Islam, Cheng-Hao Liu, Jarrid Rector-Brooks, Tara Akhound-Sadegh, Michael Bronstein, Alexander Tong, and Avishek Joey Bose. Sequence-Augmented SE(3)-Flow Matching For Conditional Protein Backbone Generation. *arXiv preprint arXiv:2405.20313*, 2024.

Yeqing Lin and Mohammed AlQuraishi. Generating Novel, Designable, and Diverse Protein Structures by Equivariantly Diffusing Oriented Residue Clouds. In *International Conference on Machine Learning*, 2023. doi: 10.48550/arXiv.2301.12485.

Yeqing Lin, Minji Lee, Zhao Zhang, and Mohammed AlQuraishi. Out of Many, One: Designing and Scaffolding Proteins at the Scale of the Structural Universe with Genie 2. *arXiv preprint arXiv:2405.15489*, 2024.

Zefeng Lin, Zhihang Zhang, Weirong Zhu, Tongchang Han, Xianyong Fang, Tianfan Fu, and Xiaohua Xu. EnzyPGM: Pocket-conditioned Generative Model for Substrate-specific Enzyme Design. *arXiv preprint arXiv:2601.19205*, 2026. doi: 10.48550/arXiv.2601.19205.

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(4):580–591, 2025. doi: 10.1038/s42256-025-01011-z.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete Diffusion Modeling by Estimating the Ratios of the Data Distribution. *arXiv preprint arXiv:2310.16834*, 2023.

Liuzhenghao Lv, Zongying Lin, Hao Li, Yuyang Liu, Jiaxi Cui, Calvin Yu-Chian Chen, Li Yuan, and Yonghong Tian. ProLLaMA: A Protein Large Language Model for Multi-Task Protein Language Processing. *arXiv preprint arXiv:2402.16445*, 2024.

Ali Madani, Bryan McCann, Nikhil Naik, Nitish Shirish Keskar, Namrata Anand, Raphael R. Eguchi, Po-Ssu Huang, and Richard Socher. ProGen: Language Modeling for Protein Generation. *bioRxiv*, 2020. doi: 10.1101/2020.03.07.982272.

Geraldene Munsamy, Sebastian Lindner, Philipp Lorenz, and Noelia Ferruz. ZymCTRL: a conditional language model for the controllable generation of artificial enzymes. In *Machine Learning for Structural Biology Workshop, NeurIPS*, 2022. URL `https://www.mlsb.io/papers_2022/ZymCTRL_a_conditional_language_model_for_the_controllable_generation_of_artificial_enzymes.pdf`.

Geraldene Munsamy, Ramiro Illanes-Vicioso, Silvia Funcillo, Ioanna T. Nakou, Sebastian Lindner, Gavin Ayres, Lesley S. Sheehan, Steven Moss, Ulrich Eckhard, Philipp Lorenz, and Noelia Ferruz. Conditional language models enable the efficient design of proficient enzymes. *bioRxiv*, 2024. doi: 10.1101/2024.05.03.592223.

Qizhi Pei, Wei Zhang, Jinhua Zhu, Kehan Wu, Kaiyuan Gao, Lijun Wu, Yingce Xia, and Rui Yan. BioT5: Enriching Cross-modal Integration in Biology with Chemical Knowledge and Natural Language Associations. *arXiv preprint arXiv:2310.07276*, 2023.

Sahil Rahman and Maxx Richard Rahman. AgentPLM: Agentic Protein Language Models with Reasoning-Augmented Decoding for Protein Sequence Design. *arXiv preprint arXiv:2606.02386*, 2026. doi: 10.48550/arXiv.2606.02386.

Jarrid Rector-Brooks, Théophile Lambert, Marta Skreta, Daniel Roth, Yueming Long, Zi-Qi Li, Xi Zhang, Miruna Cretu, Francesca-Zhoufan Li, Tanvi Ganapathy, Emily Jin, Avishek Joey Bose, Jason Yang, Kirill Neklyudov, Yoshua Bengio, Alexander Tong, Frances H. Arnold, and Cheng-Hao Liu. General multimodal protein design enables DNA-encoding of chemistry. *arXiv preprint arXiv:2604.05181*, 2026.

Chenyu Ren, Daihai He, and Jian Huang. De Novo Functional Protein Sequence Generation: Overcoming Data Scarcity through Regeneration and Large Models. *arXiv preprint arXiv:2503.21123*, 2025.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and Effective Masked Diffusion Language Models. *arXiv preprint arXiv:2406.07524*, 2024.

Zhenqiao Song, Yunlong Zhao, Wenxian Shi, Wengong Jin, Yang Yang, and Lei Li. Generative Enzyme Design Guided by Functionally Important Sites and Small-Molecule Substrates. *arXiv preprint arXiv:2405.08205*, 2024.

Zhenqiao Song, Ramith Hettiarachchi, Chuan Li, Jianwen Xie, and Lei Li. InstructPro: Natural Language Guided Ligand-Binding Protein Design. *arXiv preprint arXiv:2506.09332*, 2025.

Zhenqiao Song, Huichong Liu, Yunlong Zhao, Yang Yang, and Lei Li. Co-designing sequence and structure of functional de novo enzymes with EnzyGen2. *bioRxiv*, 2026. doi: 10.64898/2026.03.02.709205.

Filippo Stocco, Michele Garibbo, and Noelia Ferruz. Steering Generative Models for Protein Design: Aligning and Conditioning Strategies. *arXiv preprint arXiv:2511.21476*, 2025.

Jacopo Teneggi, S. M. Bargeen A. Turzo, Tanya Marwah, Alberto Bietti, P. Douglas Renfrew, Vikram Khipple Mulligan, and Siavash Golkar. Protein Design with Agent Rosetta: A Case Study for Specialized Scientific Agents. *arXiv preprint arXiv:2603.15952*, 2026.

Brian L. Trippe, Jason Yim, Doug Tischer, David Baker, Tamara Broderick, Regina Barzilay, and Tommi Jaakkola. Diffusion probabilistic modeling of protein backbones in 3D for the motif-scaffolding problem. In *International Conference on Learning Representations*, 2022. doi: 10.48550/arXiv.2206.04119.

Timothy Fei Truong and Tristan Bepler. Understanding protein function with a multimodal retrieval-augmented foundation model. *arXiv preprint arXiv:2508.04724*, 2025.

Chenyu Wang, Masatoshi Uehara, Yichun He, Amy Wang, Tommaso Biancalani, Avantika Lal, Tommi Jaakkola, Sergey Levine, Hanchen Wang, and Aviv Regev. Fine-Tuning Discrete Diffusion Models via Reward Optimization with Applications to DNA and Protein Design. In *International Conference on Learning Representations*, 2024a. doi: 10.48550/arXiv.2410.13643.

Xinyou Wang, Zaixiang Zheng, Fei Ye, Dongyu Xue, Shujian Huang, and Quanquan Gu. Diffusion Language Models Are Versatile Protein Learners. *arXiv preprint arXiv:2402.18567*, 2024b.

Xinyou Wang, Zaixiang Zheng, Fei Ye, Dongyu Xue, Shujian Huang, and Quanquan Gu. DPLM-2: A Multimodal Diffusion Protein Language Model. *arXiv preprint arXiv:2410.13782*, 2024c. doi: 10.48550/arXiv.2410.13782.

Zeyuan Wang, Qiang Zhang, Keyan Ding, Ming Qin, Xiang Zhuang, Xiaotong Li, and Huajun Chen. InstructProtein: Aligning Human and Protein Language via Knowledge Instruction. *Annual Meeting of the Association for Computational Linguistics*, 2023. doi: 10.48550/arXiv.2310.03269.

Ziwen Wang, Jiajun Fan, Ruihan Guo, Thao Nguyen, Heng Ji, and Ge Liu. ProteinZero: Self-Improving Protein Generation via Online Reinforcement Learning. *arXiv preprint arXiv:2506.07459*, 2025. doi: 10.48550/arXiv.2506.07459.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, 2022.

Fang Wu, Weihao Xuan, Heli Qi, Hanqun Cao, Heng-Jui Chang, Zeqi Zhou, Haokai Zhao, Ma Jian, Carl Ma, Yu-Chi Cheng, Kuan Pang, Xiangru Tang, Zehong Wang, Guanlue Li, Hanchen Wang, Kejun Ying, Pan Lu, Chiho Im, Seungju Han, Peng Xia, Tinson Xu, Yinxi Li, Deyao Zhu, Pheng-Ann Heng, Naoto Yokoya, Masashi Sugiyama, Li Erran Li, Jure Leskovec, and Yejin Choi. Proteo-R1: Reasoning Foundation Models for De Novo Protein Design. *arXiv preprint arXiv:2605.02937*, 2026.

Kevin E. Wu, Kevin K. Yang, Rianne van den Berg, Sarah Alamdari, James Y. Zou, Alex X. Lu, and Ava P. Amini. Protein structure generation via folding diffusion. *Nature Communications*, 2024. doi: 10.1038/s41467-024-45051-2.

Junhao Xiong, Ishan Gaur, Maria Lukarska, Hunter Nisonoff, Luke M. Oltrogge, David F. Savage, and Jennifer Listgarten. ProteinGuide: On-the-fly property guidance for protein sequence generative models. *arXiv preprint arXiv:2505.04823*, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, et al. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Jason Yang, Aadyot Bhatnagar, Jeffrey A. Ruffolo, and Ali Madani. Function-Guided Conditional Generation Using Protein Language Models with Adapters. *arXiv preprint arXiv:2410.03634*, 2024.

Jason Yim, Andrew Campbell, Andrew Y. K. Foong, Michael Gastegger, José Jiménez-Luna, Sarah Lewis, Victor Garcia Satorras, Bastiaan S. Veeling, Regina Barzilay, Tommi Jaakkola, and Frank Noé. Fast protein backbone generation with SE(3) flow matching. *arXiv preprint arXiv:2310.05297*, 2023a.

Jason Yim, Brian L. Trippe, Valentin De Bortoli, Emile Mathieu, Arnaud Doucet, Regina Barzilay, and Tommi Jaakkola. SE(3) diffusion model with application to protein backbone generation. In *International Conference on Machine Learning*, 2023b. doi: 10.48550/arXiv.2302.02277.

Junbo Yin, Chao Zha, Wenjia He, Chencheng Xu, and Xin Gao. CFP-Gen: Combinatorial Functional Protein Generation via Diffusion Language Models. *arXiv preprint arXiv:2505.22869*, 2025.

Tianhao Yu, Haiyang Cui, Jianan Canal Li, Yunan Luo, Guangde Jiang, and Huimin Zhao. Enzyme function prediction using contrastive learning. *Science*, 379, 2023. doi: 10.1126/science.adf2465.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding Conditional Control to Text-to-Image Diffusion Models. *arXiv preprint arXiv:2302.05543*, 2023.

Yulin Zhang, He Cao, Zihao Jiang, Chenyi Zi, Zhipeng Zhou, Zijing Liu, Yu Li, Jia Li, and Ziqi Gao. ProteinOPD: Towards Effective and Efficient Preference Alignment for Protein Design. *arXiv preprint arXiv:2605.10189*, 2026. doi: 10.48550/arXiv.2605.10189.

Yi Zhou, Haohao Qu, Yunqing Liu, Shanru Lin, Le Song, and Wenqi Fan. HD-Prot: A Protein Language Model for Joint Sequence-Structure Modeling with Continuous Structure Tokens. *arXiv preprint arXiv:2512.15133*, 2025.

A THE DESIGN-RECORD CORPUS

The difficulty in building it is that a language model asked to narrate a structure will assert things the structure does not show, and a corpus with that property would teach a conditioned generator from text that is confidently wrong. The pipeline therefore never lets the model establish anything. Its first two stages (Figure 4) are where that is enforced: deterministic code turns a predicted complex into a set of checkable *claims*, and a language model turns claims into argued prose while a verifier checks every sentence back against them, returning what fails. A third stage clusters and splits what survives. This is the discipline that execution-based verification brings to synthetic code corpora, with nine rules in place of a unit test.

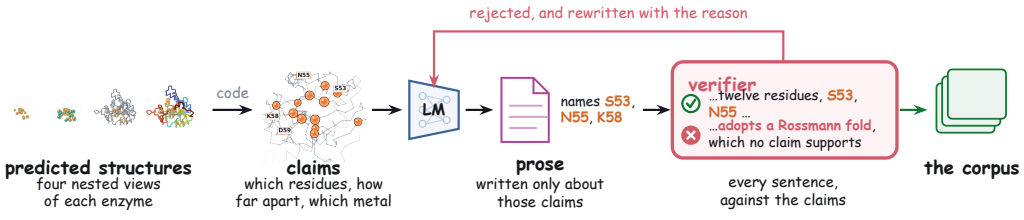

Figure 4: **Corpus construction.** Deterministic code turns each predicted complex into *claims*; a language model writes only about those claims; a verifier checks every sentence back against them and returns what fails. The rules, the filters and the counts are below.

Stage 1: complexes to claims. Each record starts from an all-atom Boltz-1x prediction of the holo complex, with the ligand present, and catalytic annotations from UniProt's `active`, `binding` and `site` fields. Two filters run before anything is written: the predictor's own confidence, which partitions the pool into high, acceptable and rejected bands and discards the rejected band; and a

degeneracy check, which drops a record with no residue within 4 Å of the ligand, since there is then no catalytic core to write about. Of 56 908 complexes, 885 fall to the first and one to the second, leaving 56 022. From each we cut four nested substructures, one per design stage: the annotated catalytic residues alone; those plus the first shell within 5 Å of the substrate; the complete backbone; and the full complex. Nesting is by construction, so the stages form a sequence of commitments each containing the last. Code then reads these structures with `gemmi` and `biotite`, separates ligands into substrate, cofactor and metal, and emits the claims. A claim is a structured object rather than a sentence: it records the stage it belongs to, whether it is a plan or an observation, the entities it names, the numbers it uses, and the provenance of the knowledge, one of predicted structure, database annotation, a quantity derived from others, a design specification or a motif match. Claims obey a causal contract, checked mechanically: stage k may plan only from quantities established before it, may observe only what stage k itself built, and may never reference a quantity that first appears later. The contract is what keeps a stage from referring to quantities it has not yet established.

Stage 2: writing under a verifier. A language model writes each of the nine sections from its claims, under a prompt that asks it to argue rather than to list. Every sentence is then checked by nine rules. Four ground every entity and quantity: a residue, a number, a subsequence or a metal that the claims do not contain is a rejection. They bound what may be *named*, not what may be asserted about it, so a sentence relating two grounded entities is checked for its terms and not for its mechanism. Four more exist because a model can satisfy the first four by restating the claims, which would yield a corpus of paraphrase rather than reasoning; they reject unsupported kinetic terms, fold names the claims do not license, plan sections with fewer than two causal connectives, and any mention of the enzyme's name, which would restore a lookup shortcut from name to answer. The ninth rejects a section of fewer than twenty-five words. A failing section is returned with the reason attached and rewritten, up to three attempts; one that still fails becomes a deterministic template built from its own claims. Every sentence in the corpus is therefore either model-written and verified, or a claim transcribed, and no entity or quantity in it is ungrounded.

Stage 3: clustering and splitting. Sequences are read from the coordinates rather than parsed out of the text, and the split is clustered by sequence. The length filter is applied *inside* each side afterwards, removing records without moving any across the boundary, so no sequence appears on both sides. Appendix B gives the resulting split and the composition it covers. A *high-density* subset of 24 476 records is marked as well, those in the high confidence band that also carry an above-median density of causal connectives; connective density is a property of the prose and not a measure of whether its reasoning is sound.

B THE DATASET

Where the records come from. Each record is one enzyme with a catalysed reaction, a sequence, and a predicted holo complex. Catalytic positions are the `active`, `binding` and `site` annotations of its UniProt entry, and the complex is an all-atom Boltz-1x prediction of the enzyme with its ligand, from which the four nested substructures of Appendix A are cut. A record is kept only when the predictor is confident and when its annotated catalytic positions are present in the structure, which is what removes 885 of the 56 908 candidates; one further record is dropped for having no residue within 4 Å of the ligand, leaving 56 022.

Size and composition. The corpus holds 56 022 records over 2,021 distinct four-level EC numbers, 127 at three levels, 53 at two and all 6 top-level classes. The classes are not balanced, and the imbalance is the natural one: transferases are the largest at 21,164 records and ligases the smallest at 707, with hydrolases, oxidoreductases, lyases and isomerases in between. Chains run from 100 to 997 residues, median 332 with an interquartile range of 244 to 424. A written record runs about 6,000 characters across its nine sections.

How it is split. The 56 022 records contain only 36 907 distinct sequences, because an enzyme appearing in several complexes appears as several records. A random split would therefore put exact copies of one protein on both sides and report recall as generalisation. We cluster by sequence and split the clusters, giving 50 420 training, 2 801 validation and 2 801 test records. Clustering removes duplicates but not homologues, and Figure 3c measures what remains: 803 of the test

records share under a tenth of their 8-mers with any training sequence, and that band is reported separately throughout rather than pooled with the rest.

What a record does not contain. The reaction is given as text and carries no EC number, no enzyme name and no sequence fragment of the target. The target length is supplied, and every method we compare against receives it. No structure, motif, site index or template reaches any method at inference.