

Programmable Design of Functional Proteins from Natural Language

Corresponding Author: Professor Fajie Yuan

Any redactions in this file are there to maintain patient confidentiality, the confidentiality of unpublished data, or to remove third-party material.

This file contains all reviewer reports in order by version, followed by all author rebuttals in order by version.

Attachments originally included by the reviewers as part of their assessment can be found at the end of this file.

Version 0:

Reviewer comments:

Referee #1

(Remarks to the Author)

This manuscript describes a protein design algorithm called Pinal that takes a textual description of the desired function as input and outputs novel sequences. The authors argue that training a machine learning architecture that goes directly from natural language to protein sequences is intractable. Pinal, in contrast, is split into two phases: 1) generating a structural model from natural language and 2) generating sequences given the input text and the structural model. The success of the algorithm also stems from the 1.7 billion synthetic data points that they generated for training. They present four different applications of Pinal to demonstrate its power.

I found this paper highly interesting/stimulating but also feel that it failed to live up to the expectations set in the abstract.

Major concerns:

My biggest concern is that the four demonstrations were underwhelming. My understanding is that the motivation for developing Pinal was to allow protein design without detailed structural/biophysical insight being required from the user. However, two of the designs required just such insight.

For the GFP design, the authors write that, "To confer function, we computationally grafted the essential "TYG" chromophore-forming 251 motif into these novel scaffolds and synthesized all 15 designs." It seems that a function-oriented architecture like Pinal should be able to figure out those requirements for function without the user having to know them in advance. Instead, it appears that Pinal just made a β -barrel scaffold and the authors then grafted all the function onto this.

Similarly, for the PETase, I was disappointed to read that "we selected 30 candidates that contained both the 264 canonical Ser-His-Asp catalytic triad and a key substrate-binding motif ("SMGGG-D-H") required to form 265 266a dynamic active site." Again, this requires that the user have significant insight, whereas I was led to believe that Pinal should make that unnecessary.

Finally, the claim based on the 1.7-fold improvement over nature for the ADH protein that "We are no longer limited to rediscovering or 335 incrementally improving what nature has provided;" is overblown. Less than a factor of 2 improvement still seems incremental.

Minor concerns:

The authors may want to consider changing the name of the algorithm. Pinal is very similar to "penal." The name also seemed arbitrary as I can't guess how it relates to what the architecture does and no explanation of the origin of the name was given.

Is T2struct a new development? I assume so, but the neutral language made me wonder if it was taken from elsewhere or published previously.

I would have appreciated a little more explanation of what a 3Di structural token from Foldseek is.

Why were different approaches needed to generate textual descriptions from different databases (e.g. Swissprot and EMBL)?

Fig. 2d isn't a representation I've seen before and could use more explanation; e.g. labeling the shaded regions would be helpful. It's also hard to tell the difference between the colors for the Ground Truth and Random. Using more different colors for the other lines instead of just shades of blue would also be helpful. Also, what is the x-axis for the plots called "Designing protein on..."?

In Fig. 3a the blues are hard to tell apart again. Please also explain the color scheme in panel c.

(Remarks on code availability)

Referee #2

(Remarks to the Author)

A. Summary of the key results

This manuscript proposes a novel 16B-parameter generative model, Pinal, trained on 1.7B protein-text pairs derived from SwissProt annotations and augmented synthetic descriptions generated using LLMs and ProTrek for proteins from TrEMBL. Evaluations on three distinct settings (i.e., long and descriptive prompts, short and function-specific phrases, and specialized keywords) demonstrate that Pinal surpasses previous protein design methods including ProteinDT, Chroma, and ESM3. Further wet-lab experiments on designing four proteins from different functional classes (i.e., GFP, PETase, ADH, and H-protein) validate the design capability of Pinal. The pinal-designed proteins were functionally active and some of them (e.g., H-protein) surpass its natural type.

B. Originality and significance

The proposed new paradigm of natural language-guided de novo protein design may have a broader impact on scientific discovery.

C. Data & methodology

The manuscript is well written and easy to follow. Overall, the proposed architecture, a two-stage process in which the first stage uses a text-to-structure module and the second designs protein sequences based on text and structural tokens, is clear. However, the core hypothesis that the language-to-structure model exhibits better generalizability is questionable and thus not well supported. Moreover, there are additional concerns regarding the use of synthetic data. Please see F for more details.

D. Appropriate use of statistics and treatment of uncertainties

The manuscript reports experimental results and analyses in an appropriate manner.

E. Conclusions

The conclusions and data interpretation are valid and robust. Some major concerns remain regarding the hypothesis and synthetic data construction. Please see F for more details.

F. Suggested improvements

- On the proposed "text sequence"->"Foldseek 3Di token sequence"->"amino acid sequence" pipeline.

* The motivation for introducing structured token sequences as an intermediate step is not sufficiently justified. First, the generalization error in Fig. 1b could be influenced by many factors (e.g., different vocabularies). In fact, the red and blue curves are two different learning tasks (predicting structured tokens and amino acid tokens), thus, what a stronger generalization error can imply is uncertain. The next token prediction accuracy (and PPL) also does not reliably reflect a model's performance on protein design tasks (i.e., NTP accuracy may not be related to pLDDT, ProTrek scores). Therefore, the weaker generalization error does not convincingly support the necessity of introducing intermediate structure token sequences.

* Second, it could be more informative to directly compare models with/without intermediate structure token sequences. Some concurrent work shows that direct text-to-amino acid sequence can be competitive (e.g., <https://arxiv.org/abs/2505.18966>).

* It would be informative to include experiments that assess whether the structural tokens generated by the T2Struct decoder can be directly used to design amino acid sequences (e.g., using ProST5 or ProteinMPNN as inverse folding models).

- On the data augmentation: Although augmenting SwissProt annotations with LLMs is a good starting point, the synthetic dataset requires matching the unannotated proteins from TrEMBL with descriptions from SwissProt. This raises the following concerns:

* How accurate are the protein-text matching scores computed by ProTrek? Will those inaccurate (or with low ProTrek model confidence) protein-text pairs affect the model performance?

* Since TrEMBL protein descriptions are paired with those from SwissProt, what happens if the corresponding

(undiscovered or less-discovered) functions are not documented in the SwissProt database?

* Both the synthetic dataset construction and the evaluation on Pinal rely on the ProTrek model. Does this imply that Pinal's strong performance on the ProTrek score is affected by its training data predicted by ProTrek?

* It seems that the main purpose of applying LLM is to enhance the linguistic diversity of text-protein pairs (e.g., prompts in Supplemental Table 5). As one can expect, diversity could improve language alignment scores. It is less intuitive how diversity improves foldability and novelty (though not included in the evaluation metric).

* It would be beneficial to inspect how augmented text data (with some controlled perturbation) contributes to other evaluation metrics, besides alignment scores in Fig. 2b.

- On T2Struct and SaProt-T's text encoding

* The main text (lines 97-100) states that T2Struct's encoder is from the pretrained PubMedBERT model, while Section 7.1 (lines 541-542) indicates that T2Struct's encoder is initialized from a pretrained T5. It is unclear which text encoder is applied in T2Struct.

* The main text (lines 105-107) states that SaProt-T's text representation is obtained by the projection of T2Struct's encoder (which is unclear, as mentioned in the above comments), while Section 7.1 (lines 552-554) states that SaProt-T introduces PubMedBERT as another text encoder.

* Fig. 1d also shows two separate text encoders (text encoder 1 and text encoder 2). This inconsistency is confusing, and the authors should carefully check and verify the model and the results.

* If the text embeddings used for decoding structural tokens and those used for SaProt-T are indeed different, could you explain why two separate text encoders are needed to encode the same textual description? The authors are also encouraged to include additional ablation studies (e.g., sharing the same text encoder).

G. References

The manuscript appropriately references previous work.

H. Clarity and context

The abstract, introduction, and conclusion of the manuscript are appropriate.

(Remarks on code availability)

Referee #3

(Remarks to the Author)

I co-reviewed this manuscript with one of the reviewers who provided the listed reports.

(Remarks on code availability)

I have reviewed the code available at <https://github.com/westlake-repl/Denovo-Pinal>. Although the training scripts and datasets are not publicly accessible, the inference code runs correctly. The instructions are well documented, and I believe the codebase is useful to the community.

Version 2:

Reviewer comments:

Referee #1

(Remarks to the Author)

The authors have done a thorough and compelling job of responding to my concerns. I think this manuscript is well-deserving of publication.

I have a few minor comments:

1. Keeping the name Pinal is fine for the reasons the authors give. I recommend explaining the origins of the name when Pinal is first mentioned in the introduction.

2. For each of the four design tasks, it would be valuable to say what the sequence identity with the closest natural sequence is. I see there's a range given, but I'd like to know the specific number for each task. I'm curious what the closest natural sequence is for the PETase? If such a comparison doesn't make sense, the authors should consider explaining why as it may highlight a strength of the approach.

(Remarks on code availability)

Referee #2

(Remarks to the Author)

1. Some concern about the necessity of an intermediate structure sequence is still there.

- We have observed some successful text-to-sequence models (e.g., ProLLaMA), a weak text-to-sequence model here

needs some further discussion.

- Another hint is from the unconditional amino acid sequence generation (e.g., protgpt2, progen2). Though they are not directly comparable here, they suggest that a model can have non-trivial next token prediction performances even if it only looks at previous amino acid tokens.
- Would you please provide code/data for reproducing curves in Fig. 2?

2. The authors' response to the question "how scaling influences pLDDT/PAE" is clear. I would suggest adding the discussion to the revised version. It helps us to know where/why scaling works.

3. Regarding the designed sequences for ADH/GFP/PETases/H-protein:

- To inspect the designed sequences (Tables 8-11 of the previous version), we perform some primary experiments on Pinal's outputs.
- The inference setting: we use the 15B T2Struct model, sample $K = 1000$ structure sequences, and use greedy decoding (top-1) from SaProt-T. For each text instruction in Table 7, we obtain 1000 designed sequences.
- We define a sequence-level similarity measure (it evaluates how the two sequences align, we computed sequence identity using MMseqs2[1] with default settings), and use it to identify similar predictions among 1000 generations for each sequence in Tables 8-11.
- On PETases, we find that it is hard to find similar sequences in 1000 generations (e.g., the most similar generation has a similarity <50%). It may suggest that it is hard to generate those sequences.
- On ADH/H-protein, we can find similar sequences (>80% scores), while their perplexity is significantly higher compared with other sequences in the 1000 generations (i.e., when ranking according to ppl, those sequences are ranked at the bottom 10%). It may suggest that the following filter process could have structured bias.
- Since our inference setting slightly differs from the paper, would you please conduct (and open source) some similar experiments and discuss the results?

[1] Kallenborn F, Chacon A, Hundt C, Sirelkhatim H, Didi K, Cha S, Dallago C, Mirdita M, Schmidt B, Steinegger M: GPU-accelerated homology search with MMseqs2. Nature Methods, doi: 10.1038/s41592-025-02819-8 (2025)

4. During our experiment, we found that some details could be made clearer.

- Regarding the initial candidates, for GFP, Pinal generates 50,000 initial samples (line 919, revised manuscript), but it is not clear whether the other three types get the same number.
- It would also be helpful to explicitly describe the process of generating the initial candidates. We first guess that first sampling $K = 50$ structures, then sample 1000 sequences from SaProt-T, but after checking the code, SaProt-T only supports greedy decoding (missing code for "multinomial sampling" in line 757).
- Compared the two versions of Tables 8-11, PETases36 is completely new, and Table 9 (ADH) removes some rows. The update may need some explanation.
- Would you please provide the structure tokens for the sequences in Tables 8-11?

(Remarks on code availability)

Referee #3

(Remarks to the Author)

I co-reviewed this manuscript with one of the reviewers who provided the listed reports.

(Remarks on code availability)

Version 3:

Reviewer comments:

Referee #2

(Remarks to the Author)

Many thanks to the authors for the explanation and discussion. I'll wrap up my review with the following feedback:

- I appreciate the authors' response regarding experiment details (point 4). I think the details are sufficient to reproduce Pinal's generations.

- To characterize the designed sequences, Fig. R1 is informative: ADH/H-protein samples with low PPL rankings can still have reasonable foldability and alignment scores (they are on the learned manifold). It would be more informative to indicate the position of designed sequences (for the four proteins) in the figure.

- Regarding PETase generations, the point is that if a model can generate a sequence, it may be easy to generate some similar sequences (i.e., those sequences in the same local mode). Although enlarging the sample pool can help, it seems that generating similar PETase sequences is harder than for the other three proteins: it requires a larger sample pool to get moderately similar sequences. I would suggest 1) conducting more careful analyses for PETase generations, and 2)

elaborating the difficulty explicitly in the paper.

- Regarding the intermediate structure token prediction (Fig. 2a), it is potentially helpful to include structural features in the conditional generation pipeline, but it is also important to implement the baseline (direct aa token prediction) correctly. Depending on the context, "the same model setting" may or may not lead a fair comparison. I would suggest 1) trying some alternative configurations (e.g., different hyperparameters) for a stronger next aa prediction baseline, and 2) discussing some follow-up works on direct aa sequence generation for an up-to-date literature review.

(Remarks on code availability)

Referee #3

(Remarks to the Author)

I co-reviewed this manuscript with one of the reviewers who provided the listed reports.

(Remarks on code availability)

Open Access This Peer Review File is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

In cases where reviewers are anonymous, credit should be given to 'Anonymous Referee' and the source.

The images or other third party material in this Peer Review File are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

To view a copy of this license, visit <https://creativecommons.org/licenses/by/4.0/>

Point-by-point Response Letter

We would like to express our sincere gratitude to the three reviewers and the editor for their careful evaluation, constructive suggestions, and recognition of our work. We have carefully considered each comment and provide detailed point-by-point responses below. In addition, we have revised the manuscript accordingly, with the key changes made in response to the reviewers' comments highlighted in red.

Reviewer #1 (Remarks to the Author):

Major concerns:

I found this paper highly interesting/stimulating but also feel that it failed to live up to the expectations set in the abstract. My biggest concern is that the four demonstrations were underwhelming. My understanding is that the motivation for developing Pinal was to allow protein design without detailed structural/biophysical insight being required from the user. However, two of the designs required just such insight.

For the GFP design, the authors write that, “To confer function, we computationally grafted the essential “TYG” chromophore-forming²⁵¹ motif into these novel scaffolds and synthesized all 15 designs.” It seems that a function-oriented architecture like Pinal should be able to figure out those requirements for function without the user having to know them in advance. Instead, it appears that Pinal just made a β -barrel scaffold and the authors then grafted all the function onto this.

Similarly, for the PETase, I was disappointed to read that “we selected 30 candidates that contained both the²⁶⁴ canonical Ser-His-Asp catalytic triad and a key substrate-binding motif (“SMGGG-D-H”) required to form^{265 266a} dynamic active site.” Again, this requires that the user have significant insight, whereas I was led to believe that Pinal should make that unnecessary.

Our Response:

We sincerely thank the reviewer for this thoughtful and important comment. We particularly appreciate the reviewer's concerns regarding the degree of human intervention in our initial demonstrations. These comments have been highly valuable in helping us refine the manuscript, better define the scope of our claims, and carry out additional experiments that have substantially strengthened the work. We agree that the original framing of the manuscript—especially in the Abstract & Introduction—was overly broad and could have been read as implying that domain-specific structural insight is unnecessary across all protein design settings. This was not our intention, and we are grateful to the reviewer for prompting us to clarify this issue more carefully.

Below, we provide a detailed response along with the corresponding conceptual clarifications, new experimental data, and revisions to the manuscript.

1. Clarifying "Democratization": The Distinction Between Generation and Filtering

Reviewer's Concern: The use of motif filtering (e.g., for PETase) appears to be inconsistent with the claim that Pinal enables protein design without detailed structural or biophysical insight, weakening the “democratization” claim.

Our Response:

We greatly appreciate this critique, as it highlights the need to clarify both the scope of our claim and the meaning of "democratization" in the context of modern computational protein design. Furthermore, we present new wet-lab results to more directly address the reviewers' concerns.

In practice, protein design typically involves two stages with very different technical requirements: **generation** and **filtering**.

- a) **The Primary Design Bottleneck (Generation):** Historically, generating functional protein candidates from scratch has required **substantial expertise** in energy landscapes, geometric constraints, and the implementation of complex computational pipelines to navigate an astronomically large sequence space (e.g., 20^{100} – 20^{1000} **possible sequences**). Because of these steep technical barriers, protein design has traditionally been accessible only to a limited number of laboratories with deep expertise in computational biophysics. In this context, Pinal lowers this barrier by allowing users to specify a desired function through natural-language prompts and generate candidate pools without manually constructing complex structure-based design or generation pipelines.
- b) **The Pragmatic Step (Filtering):** In contrast, once the generative step has reduced the astronomically large sequence space to a much smaller and more tractable candidate pool, typically on the order of 10^2 – 10^4 sequences, screening for specific functional motifs or other biologically informed criteria can be carried out using relatively accessible tools, such as BLAST and PyMOL, together with domain-specific knowledge. While this step still requires biological expertise, as the reviewer rightly noted, it is substantially more accessible in practice than the computational generation step and often falls within the existing capabilities of many experimental biology laboratories. These laboratories represent an important intended user group for Pinal: they often possess the biological insight needed to define functional goals, design assays, and prioritize candidates, but may not have the specialized design expertise required to construct complex protein-design pipelines.

Indeed, in practical wet-lab protein design workflows, generative-model outputs are rarely taken directly to synthesis and testing without some form of prior inspection or sanity check; applying available domain knowledge in this way is therefore a routine and pragmatic strategy for reducing experimental burden. In this respect, downstream filtering is not unique to Pinal, but is repeatedly observed as a practical step in **virtually all** experimentally validated protein-design studies. For example:

- **RFdiffusion2 [3]** (Nature Methods, 2025) **reports filtering initial pools of ~960,000 sequences down to ~9,800** based on structural and motif-related criteria (see attached **Fig. E4**).
- **Chroma [4]** (Nature, 2023) operates on the premise that **viable candidates occur at ~1% frequency**, necessitating heavy filtering from pools of 10,000–50,000 (see attached **Fig. E3**).
- **ESM3 [1]** (Science, 2025), **BindCraft [2]** (Nature, 2025), and **Riff-Diff [9]** (Nature 2025) all employ extensive post-generation filters **combining similar motif-informed constraints and RMSD checks** (see more attached figs at the end).

More broadly, in protein design, the practical objective is often not for every generated sequence to fully satisfy the target function, but for the generative model to produce a candidate pool enriched with promising functional solutions that can then be prioritized through downstream screening. This distinction is important because computational protein design remains highly challenging, and even advanced AI-based methods cannot achieve consistently high success rates across all targets; for difficult design tasks, **reported hit rates can be well below 1%** [3,4]. On the other hand, filtering alone cannot create function if the generative model has failed to produce relevant candidates in the first place; it can only help identify the most promising designs among those already generated.

At the same time, we fully agree with the reviewer that this distinction was not communicated clearly enough in the original manuscript, and that our previous wording could have given readers the impression that domain knowledge is unnecessary. In the revised manuscript, we have **substantially rewritten the Abstract and Introduction**, and have added an explicit discussion in the Discussion section (Lines 344-357, see screenshots below) to acknowledge the role of post-generation filtering as a currently necessary practical step. These revisions to the Abstract, Introduction, and Discussion are intended to reflect our narrower claim more accurately: Pinal aims to lower the barrier to specialized design expertise at the generation stage by enabling users to specify a desired function in natural language and obtain a large pool of promising candidates, without requiring them to manually construct complex design pipelines or explicitly impose detailed structural constraints during generation. Our intention is not to suggest that biological or domain-specific knowledge is no longer useful, particularly for downstream candidate prioritization and experimental design.

341 Several limitations remain. Pinal’s capabilities are constrained by the granularity and coverage of avail-
342 able biological annotations, which currently limit fine-grained quantitative control over specified properties.
343 For example, prompting for a precise emission wavelength in a fluorescent protein remains challenging.
344 In addition, although our experiments validate the efficacy of the approach, the candidates advanced to
345 experimental testing were prioritized conservatively using available *in silico* proxies, including AlphaFold
346 confidence, ProTrek scores and, where appropriate, motif-based criteria. These heuristics can improve prac-
347 tical hit rates, but they do not imply that candidates outside such filters are non-functional. Rather, they
348 reflect the current lack of universally reliable computational metrics for evaluating protein function *a priori*.
349 As a result, such prioritization may restrict exploration to a limited region of the design space. Coupling
350 language-guided generation to high-throughput functional screening will therefore be important for more
351 comprehensively probing the accessible design space and the limits of Pinal.

352 More broadly, our work suggests a path towards protein design systems specified increasingly in terms
353 of function rather than manual constraint engineering. As biological datasets become richer and AI systems
354 become more capable of integrating mechanistic reasoning with generative design, language-guided models
355 may support increasingly open-ended exploration of protein function. In this sense, Pinal represents a step
356 toward a programmable interface for biology, in which high-level intent can be translated into experimentally
357 testable molecular candidates.

1.1 New PETase validation without motif- or geometry-based prioritization

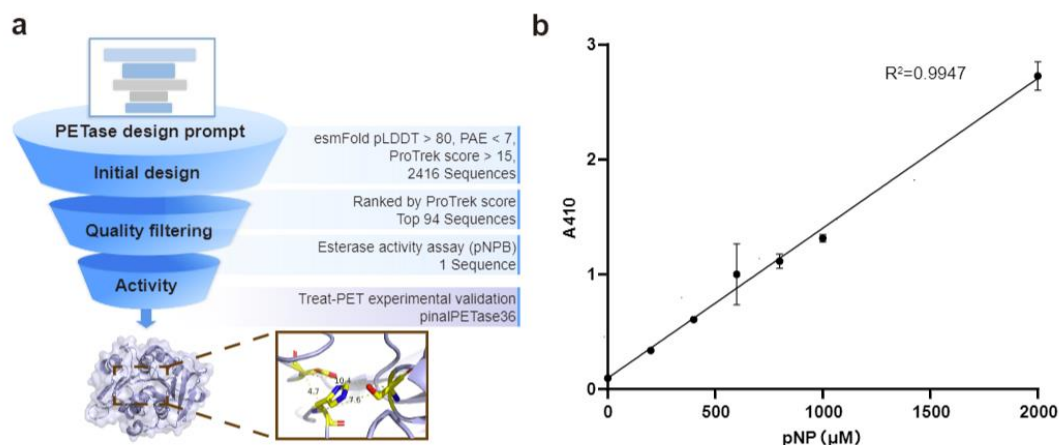
We thank the reviewer for raising this important concern. We agree that our original PETase validation relied on domain-informed motif- and geometry-based prioritization, including the “SMGGG-D-H” motif and empirical Ser–His–Asp geometric criteria. As discussed above, while such downstream motif-informed prioritization is common in practical protein-design workflows, we have now performed a new PETase validation using a selection procedure that did not use PETase-specific sequence motifs or active-site geometry filters, relying instead on only general foldability and text–protein alignment metrics. This experiment was designed to more directly address the reviewer’s concern and to further challenge Pinal’s ability to produce active PET-degrading enzymes.

We now report this new PETase validation in the revised main text, Lines 274–288, and Extended Data Fig. 8. In this experiment, Pinal was guided by a high-level natural-language prompt specifying PET-degrading activity, with no structural or sequence template provided as input. Importantly, candidates were not selected using the “SMGGG-D-H” motif or the Ser–His–Asp catalytic-geometry filters used in the original validation. Rather, we prioritized candidates using only general computational criteria: predicted foldability (i.e., pLDDT) and text–protein semantic alignment (i.e., ProTrek score) with the PETase prompt. This yielded 2,416 candidates, from which the top 94 proteins ranked by ProTrek score were subsequently synthesized and experimentally tested.

Screening crude *E. coli* lysates using a p-nitrophenyl butyrate esterase assay identified pinalPETase36 as an active candidate. After purification, pinalPETase36 demonstrated good structural stability with a melting temperature of $56.84 \pm 0.29^\circ\text{C}$, allowing us to evaluate its performance on solid PET substrates at 40°C and pH 8.0. HPLC analysis over seven days showed progressive accumulation of the PET degradation products terephthalic acid (TPA) and mono-(2-hydroxyethyl) terephthalate (MHET). In addition, scanning electron microscopy (SEM) analysis revealed clear micro-pores and pitting on PET substrates

treated with pinalPETase36, in contrast to the smooth surface observed in buffer-treated controls. Together, these chemical and physical analyses demonstrate that pinalPETase36 functions as an active PET-degrading enzyme.

Overall, these results provide a more direct test of Pinal's generative ability. They show that a Pinal-generated candidate pool, prioritized using only general foldability and text–protein alignment metrics, can contain experimentally active PET-degrading enzymes that are identifiable by routine biochemical screening, without applying PETase-specific sequence-motif or catalytic-geometry filters during candidate selection.



Extended Data Fig. 8: Computational screening pipeline and pNP standard curve for pinalPETase identification. **a**, Flowchart illustrating the multi-tiered screening strategy used to identify the active PETase candidate from the initial pool of Pinal-generated sequences. Candidates were first filtered on the basis of predicted structural quality (ESMFold pLDDT > 80, PAE < 7) and text–protein alignment (ProTrek score > 15), yielding 2,416 sequences. The top 94 candidates, ranked by ProTrek score, were then subjected to esterase activity screening using the *p*-nitrophenyl butyrate (pNPB) assay. This workflow led to the identification of a single standout variant, pinalPETase36, which was subsequently advanced to Treat-PET experimental validation. The inset shows the predicted active-site architecture of pinalPETase36. **b**, Standard curve for *p*-nitrophenol (pNP) used for quantification of esterase activity in the pNPB assay. Absorbance at 410 nm was plotted against pNP concentration, showing a strong linear relationship ($R^2 = 0.9947$). Data are shown as mean $\pm$ s.d.

2. Addressing the GFP Design

Reviewer's Concern: For the GFP design, it appears Pinal just made a $\alpha\beta$ -barrel scaffold and the authors then manually grafted the chromophore, requiring advance user insight.

Our Response:

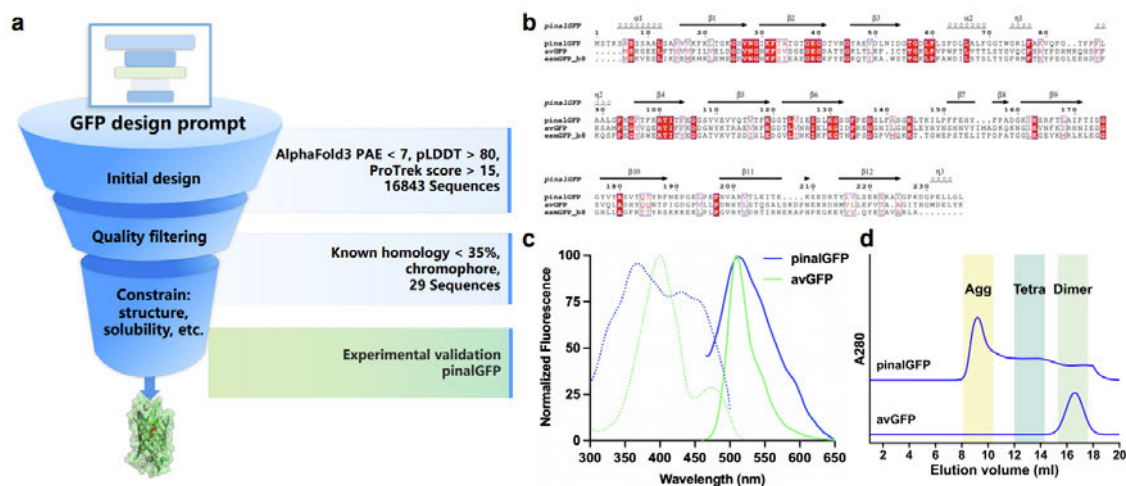
We sincerely thank the reviewer for this important and valid observation. We agree that the original GFP experiment did not fully address whether Pinal could generate a fluorescent protein sequence without post-generation chromophore insertion. To directly address this concern, we performed an additional wet-lab validation in which Pinal-generated fluorescent protein candidates were synthesized and tested exactly as generated by the model, without manual chromophore grafting.

The steps and results are as follows (Extended Data Fig. 7a):

- End-to-end sequence generation without downstream motif grafting:** Based on the text prompt, Pinal generated an initial pool of 50,000 sequences, a scale comparable to recent generative protein-design studies [3,4]. We first applied general quality filters based on AF3 structural confidence and ProTrek alignment confidence, retaining 16,843 sequences with AF3 pLDDT > 80, PAE < 7, and ProTrek score > 15. We then prioritized candidates using additional criteria, including sequence novelty (<35% identity to known proteins) and the presence of one of six candidate chromophore-forming tripeptides, used as a **conservative and non-exhaustive screen** for GFP-like chromophore chemistry: SYG, TYG, SHG, TWG, QYG, or HYG (see details in Methods 7.11.1). These motifs were

not introduced after generation, but were already present in the selected Pinal-generated sequences. This procedure yielded 29 candidates for experimental testing.

- b) **Experimental Validation:** The 29 candidates were synthesized and tested exactly as generated by Pinal, with no post-generation sequence modification or chromophore motif grafting.
- c) **Results:** We identified one fluorescent variant, pinalGFP, that produced clear green fluorescence under our assay conditions. To test whether the fluorescence depended on the predicted chromophore-forming residues, we mutated the TWG motif to TGG, which abolished detectable fluorescence under the same assay conditions.



Extended Data Fig. 7: Computational design, screening, and characterization of pinalGFP. **a**, Flowchart illustrating the multi-tiered computational filtering strategy applied to the initial Pinal-generated sequences. This pipeline culminated in the selection of 29 candidates for experimental validation. **b**, Multiple sequence alignment of pinalGFP, avGFP, and esmGFP-b8. The alignment highlights the substantial sequence divergence of pinalGFP from both the natural protein and the other synthetic variant. **c**, Normalized fluorescence excitation (dashed lines, $\lambda_{em} = 510$ nm) and emission (solid lines, $\lambda_{ex} = 460$ nm) spectra of pinalGFP (blue) and avGFP (green). **d**, Size-exclusion chromatography profiles of purified GFP variants on a Superdex 200 column. Elution volumes reflected their oligomeric states in solution: avGFP was predominantly dimeric, whereas pinalGFP exhibited a more complex profile dominated by higher-order multimers.

These new results are now described in the main text, Lines 256–271. They directly address the reviewer’s specific concern by showing that Pinal can generate a complete fluorescent protein sequence containing a chromophore-forming motif without manual grafting. Specifically, in this new experiment, we did not modify the generated sequences by adding a chromophore-forming motif. Instead, the motif criterion was applied only after generation, as a prioritization filter to select candidates in which such motifs were already present in the Pinal-generated sequences. This experiment therefore addresses the reviewer’s concern about post-generation grafting, while still acknowledging that biologically informed filtering was used to reduce the number of candidates taken forward for synthesis. We further note that the **motif-based filter used here was intentionally conservative and non-exhaustive**: GFP-like proteins can accommodate a broader repertoire of chromophore-forming tripeptides beyond the six motifs considered in our screen. Therefore, failing this filter should not be interpreted as evidence that a generated sequence is incapable of fluorescence. Rather, the filter was used as a practical prioritization strategy to enrich for candidates with recognizable chromophore-forming motifs before experimental testing.

While we are encouraged by the new no-grafting GFP result, we agree with the reviewer that the original GFP experiment should be interpreted more narrowly. We have therefore moved it to the Supplementary Information and now present it as a scaffold-compatibility test rather than as evidence of fully end-to-end generation of a functional fluorescent protein. Even under this narrower interpretation, the result remains meaningful. Fluorescence requires more than a β -barrel fold plus a chromophore-forming tripeptide; the surrounding scaffold must provide a local geometry and chemical environment compatible with chromophore maturation. Indeed, the recent ESM3 fluorescent-protein study (Science, 2025) obtained functional designs by relying on explicit user-specified constraints around the chromophore, including prespecified residue identities, residue coordinates, and iterative refinement. By contrast, Pinal generated the β -barrel-like scaffold in the initial

generation step, without residue-level or coordinate-level constraints, and the resulting scaffold nevertheless supported detectable fluorescence after chromophore grafting. Thus, although this experiment does not demonstrate fully end-to-end GFP generation, it still shows progress toward functional fluorescent-protein design with reduced user-specified structural constraints.

More broadly, these results demonstrate how Pinal can be used across different protein-design settings:

- **Designs requiring no or only minimal intervention:** In the H-protein and ADH cases, Pinal generated experimentally functional proteins without explicitly using domain-specific constraints or domain-informed filtering.
- **Domain-informed prioritization for more challenging targets:** For more challenging targets, empirical motif-based filters can be used, consistent with common practice in protein design literature. These filters should be viewed as conservative prioritization strategies for selecting candidates for experimental testing; they do not imply that sequences lacking the specified motifs or geometries are necessarily non-functional.

Together, these examples illustrate that Pinal can be used either as a standalone generator for relatively tractable targets or as a practical design tool for more challenging targets, where limited domain-informed intervention may help prioritize candidates for experimental testing. In both cases, Pinal provides functionally enriched candidate pools from high-level functional descriptions, while downstream prioritization, when applied, helps select candidates for experimental evaluation rather than serving as the source of function.

3. Regarding The 1.7-Fold Improvement in H-protein

Reviewer's Concern: Finally, the claim based on the 1.7-fold improvement over nature for the ADH protein that “We are no longer limited to rediscovering or incrementally improving what nature has provided;” is overblown. Less than a factor of 2 improvement still seems incremental.

Our Response:

We thank the reviewer for this important and helpful comment. We appreciate the reviewer's careful reading, which helped us refine the interpretation of this result and present the conclusion in a more precise and appropriately scoped manner. We agree that the original statement went beyond what this result alone can support, and have therefore removed this statement and revised the corresponding description accordingly. We would like to gently clarify that the 1.7-fold improvement discussed in the manuscript refers to the H-protein case, rather than the ADH experiment mentioned in the comment. Specifically, this improvement reflects an increase in total product titer in a CO₂-fixation pathway after replacing the native H-protein with a Pinal-designed variant, rather than an increase in the catalytic rate (K_{cat}) of an isolated enzyme. This distinction is important because the result reflects a systems-level improvement in CO₂ fixation achieved through a single natural-language-guided design cycle, without directed evolution or conventional enzyme engineering. In this context, we consider the result to be informative for three reasons:

- Systems-level complexity vs. single-enzyme optimization.** Unlike ADH, which functions as a relatively self-contained catalyst, the H-protein serves as a central molecular shuttle that must interact productively with multiple partner proteins in the pathway. Improving such a component is therefore fundamentally different from optimizing a single active site, because gains in one interaction can compromise others. In this setting, a 1.7-fold increase in total pathway product resulting *solely* from replacement of the native H-protein with a computationally designed variant is therefore non-trivial. To our knowledge, no previous literature has reported the computational design of a dynamic shuttle protein capable of balancing competing interface requirements within a multi-enzyme network.
- A first-round pathway-level improvement over the native counterpart is informative in this setting.** In the context of computational protein design, it remains uncommon for an initial design to match or exceed the performance of a natural protein without additional rounds of experimental optimization. In most published examples, computationally generated proteins show only detectable or promising activity in the first round, but

substantial improvement typically requires subsequent mutagenesis and optimization [3,7,10]. In contrast, in this case the Pinal-designed H-protein produced a 1.7-fold improvement at the pathway level over the native counterpart **directly from the initial generation step**, using only natural-language prompting. While this fold-change should be interpreted cautiously, it provides useful evidence that first-round Pinal designs can yield measurable pathway-level gains in this setting.

- c) **The result was obtained under a language-driven design setting with a high functional hit rate.** Furthermore, this task was a language-driven exploration, which introduces an additional layer of complexity compared to conventional structure-conditioned design (To our best knowledge, no peer-reviewed study has yet reported experimentally validated protein designs generated solely from a functional text description). Despite this difficulty, 5 of the 20 H-protein designs exhibited confirmed activity in the CO₂-fixation pathway. In the context of generative design—where simply yielding a functional protein remains a significant challenge—generating multiple candidates that not only function but also match or outperform the native H-protein at the system level supports the feasibility of our approach.

In response to the reviewer’s concern, we have revised the manuscript to avoid overgeneralization. We no longer present this result as a broad assertion that our approach transcends incremental optimization. Instead, we describe it more narrowly as a proof-of-concept that natural-language-driven generation can, in some cases, yield system-compatible protein candidates with measurable improvement over a natural counterpart in a single design step.

4. Conceptual Motivation—Function-Guided Protein Generation

Finally, we would like to clarify the conceptual motivation of Pinal. We do not claim that Pinal or natural-language prompting can replace protein design paradigms based on structural or biophysical reasoning. Rather, we view Pinal as an early exploration of a complementary direction in which generation can start from a functional objective expressed in natural language, followed by downstream filtering, experimental testing, and, when needed, further optimization.

Importantly, the value of natural-language-based protein design is not limited to user-friendliness. Many protein functions, as commonly described in curated annotations such as Swiss-Prot, can be difficult to translate into a complete set of predefined structural constraints. Examples include substrate- or cofactor-specific enzymatic activities, participation in multi-enzyme metabolic pathways, transport or regulatory roles, and chromophore-forming or maturation-dependent functions. These functions often depend on distributed sequence and structural features rather than a single motif, backbone geometry, or active-site arrangement. Such function-level objectives are not always directly tractable with structure-conditioned workflows, such as those based on RFdiffusion [8] or ProteinMPNN [11], which typically rely on explicit structural inputs, scaffolds, or constraints.

Thus, Pinal explores whether richer functional descriptions—such as hydrolysis of ester bonds in polyethylene terephthalate or related polyester substrates, lipoate-bearing carrier function in the glycine cleavage system, or fluorescence arising from autocatalytic chromophore maturation—can guide the generation of candidate pools enriched for relevant functional features. We fully acknowledge that this capability remains preliminary and does not imply quantitative control over arbitrary functional properties from text alone. Nevertheless, our results suggest that natural language can serve as a useful representation of functional intent in this setting, motivating further investigation of function-driven protein generation as a complement to structure-based design.

Minor concerns:

The authors may want to consider changing the name of the algorithm. Pinal is very similar to “penal.” The name also seemed arbitrary as I can’t guess how it relates to what the architecture does and no explanation of the origin of the name was given.

Our Response:

We sincerely thank the reviewer for pointing this out. We completely understand the concern regarding the phonetic similarity to "penal," which was an unintended oversight on our part.

The manuscript has undergone multiple revisions, with seven versions posted on bioRxiv since Aug 2024. During this iterative process, Pinal's name may not be directly decipherable in the current version. "Pinal" was originally coined as an acronym for **P**rotein **I**nterface for **N**ature **A**nd **L**anguage. This name was chosen to reflect the model's core contribution: redefining the "interface" for the field by bridging biological nature and human language, enabling users to design proteins through natural language prompts.

Regarding the sensible suggestion to change the name, we find ourselves in a practical dilemma and would like to humbly seek the reviewer's understanding. We first shared the preprint of this work on bioRxiv in early 2024. Since then, the name "Pinal" has been adopted by the community, incorporated into downstream open-source discussions (e.g., https://x.com/ssiddhant_/status/1994889342918299946), and cited multiple times in the literature (e.g., Google Scholar: https://scholar.google.com/citations?view_op=view_citation&hl=en&user=kaFmPDEAAAAJ&citation_for_view=kaFmPDEAAAAJ:Y0pCki6q_DkC).

Recently, the model was featured and explicitly referred to as the "Pinal" AI model in a *Nature* News article (*Nature* 641, 1079-1080, 2025; <https://www.nature.com/articles/d41586-025-01586-y>). Because the name has already established a footprint in the public domain and scientific literature, we are highly concerned that renaming the algorithm at this late stage might cause fragmentation and confusion among researchers trying to track or build upon our work.

However, we deeply respect the reviewer's perspective. If the reviewer strongly believes that the phonetic connotation outweighs the risk of literature confusion, we are completely open and willing to change the name.

Is T2struct a new development? I assume so, but the neutral language made me wonder if it was taken from elsewhere or published previously.

Our Response:

We thank the reviewer for the comment. T2struct, the text-to-3Di token generation module, is introduced for the first time in this paper. Moreover, the model was trained from scratch on over 100 billion word tokens using substantial computational resources. For completeness, we would like to mention that an earlier preprint version of this work appeared on bioRxiv on August, 2024: <https://www.biorxiv.org/content/10.1101/2024.08.01.606258v7>

I would have appreciated a little more explanation of what a 3Di structural token from Foldseek is.

Our Response:

We thank the reviewer for this helpful comment. We have added a brief clarification at the first mention of 3Di tokens in the revised manuscript. A 3Di token is a per-residue discrete descriptor of local tertiary interaction geometry. In Foldseek, each residue is assigned one structural token based on the spatial relationship between its local backbone environment and that of its nearest neighboring residue in 3D space. These geometric relationships are summarized by continuous structural features and then discretized into one of 20 structural states, forming the 3Di structural alphabet. Thus, a protein backbone can be represented as a one-dimensional sequence of 3Di tokens, with one token aligned to each residue.

In our framework, a protein structure can therefore be represented as a one-dimensional sequence of 3Di tokens aligned with the amino-acid sequence. This allows us to formulate structure-conditioned protein generation as a sequence-to-sequence problem, where structural information is encoded in a compact token sequence. Compared with models that directly operate

on 3D coordinates or residue graphs, this representation is computationally convenient and compatible with scalable Transformer/LLM architectures, while retaining informative tertiary-interaction patterns derived from protein structures.

Why were different approaches needed to generate textual descriptions from different databases (e.g. Swissprot and EMBL)?

Our Response:

We sincerely thank the reviewer for raising this insightful question. The necessity for employing different approaches stems from the fundamental differences in annotation quality between the two databases.

Specifically, the UniProtKB/Swiss-Prot database consists of manually curated and human-reviewed annotations. We treat these high-quality descriptions as the gold standard and therefore utilize them directly.

In contrast, the UniProtKB/TrEMBL database contains unreviewed annotations that are automatically generated by empirical and computational pipelines, which can often be noisy or less accurate. Crucially, these automated annotations suffer from severe sparsity (i.e., widespread missing entries) and highly skewed data distributions. Directly utilizing such imbalanced and incomplete textual data would severely destabilize the learning process, ultimately leading to model training failure.

Therefore, while retaining the large-scale protein sequence coverage provided by TrEMBL, we discarded its default textual annotations and instead used ProTrek, a protein annotation retrieval model published in *Nature Biotechnology* in 2025, to retrieve functional descriptions for these proteins. As demonstrated in the original ProTrek study, ProTrek achieves state-of-the-art performance in protein annotation retrieval and even outperforms conventional sequence- and structure-based retrieval tools, including BLAST, MMseqs2, and Foldseek, for annotation retrieval tasks. We further filtered the retrieved candidates using high-confidence ProTrek’s sequence-to-function alignment scores. This strategy allowed us to obtain functional descriptions that are more comprehensive, consistent, and suitable for large-scale text–protein generative learning than the default automated annotations in TrEMBL.

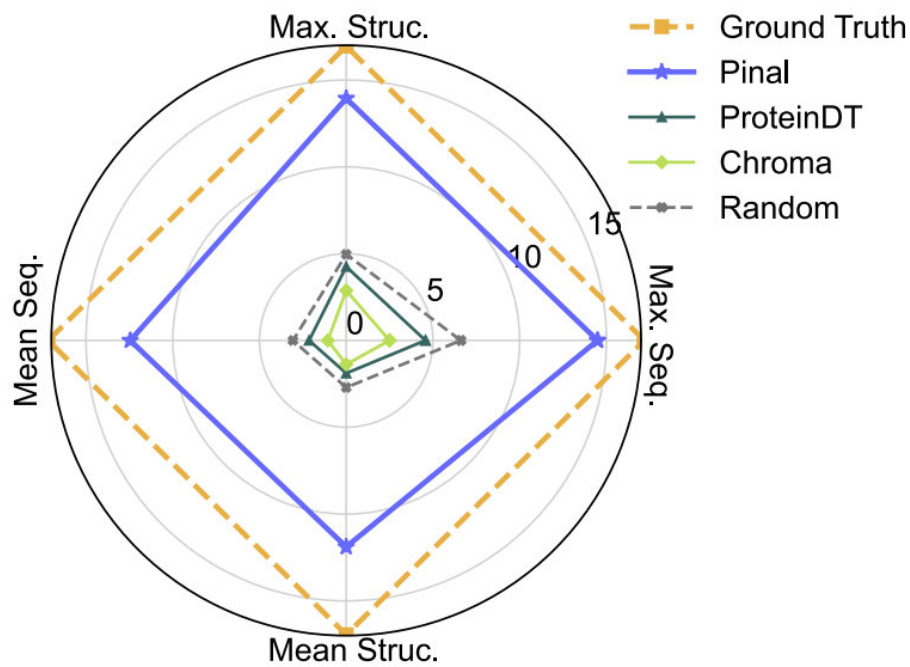
Fig. 2d isn’t a representation I’ve seen before and could use more explanation; e.g. labeling the shaded regions would be helpful. It’s also hard to tell the difference between the colors for the Ground Truth and Random. Using more different colors for the other lines instead of just shades of blue would also be helpful. Also, what is the x-axis for the plots called “Designing protein on...”?

Our Response:

We thank the reviewer for these helpful suggestions. We agree that the original Fig. 2d was not sufficiently clear. The left panel of Fig. 2d is a radar chart. The four labeled directions, Max. Seq., Mean Seq., Max. Struc., and Mean Struc., correspond to four ProTrek-based alignment metrics, as now explained in the revised Fig. 2d legend and in Methods Section 7.7.3. Each method forms a polygon by connecting its values on these four axes; values farther from the center and a larger enclosed polygon indicate better overall performance.

In the revised figure, we have clarified this interpretation in the caption and added radial tick labels, such as 5, 10, and 15, to make the scale easier to read. We also removed the visually confusing shaded overlap and instead used distinct colors, markers, and line styles for different methods. In particular, Ground Truth and Random are now shown with clearly different colors and dashed-line styles, making them easier to distinguish (see the revised figure below).

d



We also apologize for the missing x-axis explanation in the plots titled “Designing protein on ...”. The x-axis in Fig. 2b represents the number of word tokens in the training corpus, which is a commonly used scale metric in large language model studies. We have added this explanation to the revised figure legend and marked the corresponding revision in red.

In Fig. 3a the blues are hard to tell apart again. Please also explain the color scheme in panel c.

Our Response:

We thank the reviewer for pointing out these visual ambiguities. We have made the following corrections to Fig. 3 to improve its clarity:

- Colors in Fig. 3a: We have updated the color palette in the maximum-likelihood phylogenetic tree (Fig. 3a). Specifically, we changed the colors for Natural and Clade 1 to more distinct and contrasting colors, making it much easier to differentiate between the groups.
- Color scheme in Fig. 3c: We have updated the figure caption for Fig. 3c to explicitly describe the color scheme. It now clarifies that the colored structures (in rainbow gradient from N- to C-terminus) represent the novel proteins designed by Pinal, while the superimposed grey structures represent their closest structural homologues found in the PDB.

Finally, we sincerely thank Reviewer #1 for the thoughtful and constructive comments. We greatly appreciate the reviewer’s careful reading of our manuscript and the many helpful suggestions, which have substantially improved the clarity, rigor, and presentation of the work. In the revised manuscript, we have carefully addressed each concern through textual clarifications, additional analyses, figure revisions, and new experimental validations where appropriate. In particular, the reviewer’s comments helped us better define the scope of our claims and strengthen the manuscript with additional wet-lab evidence. We hope that the revised version satisfactorily resolves the reviewer’s concerns and more clearly communicates the motivation, methodology, limitations, and findings of our study. We would be happy to provide any further clarification if needed.

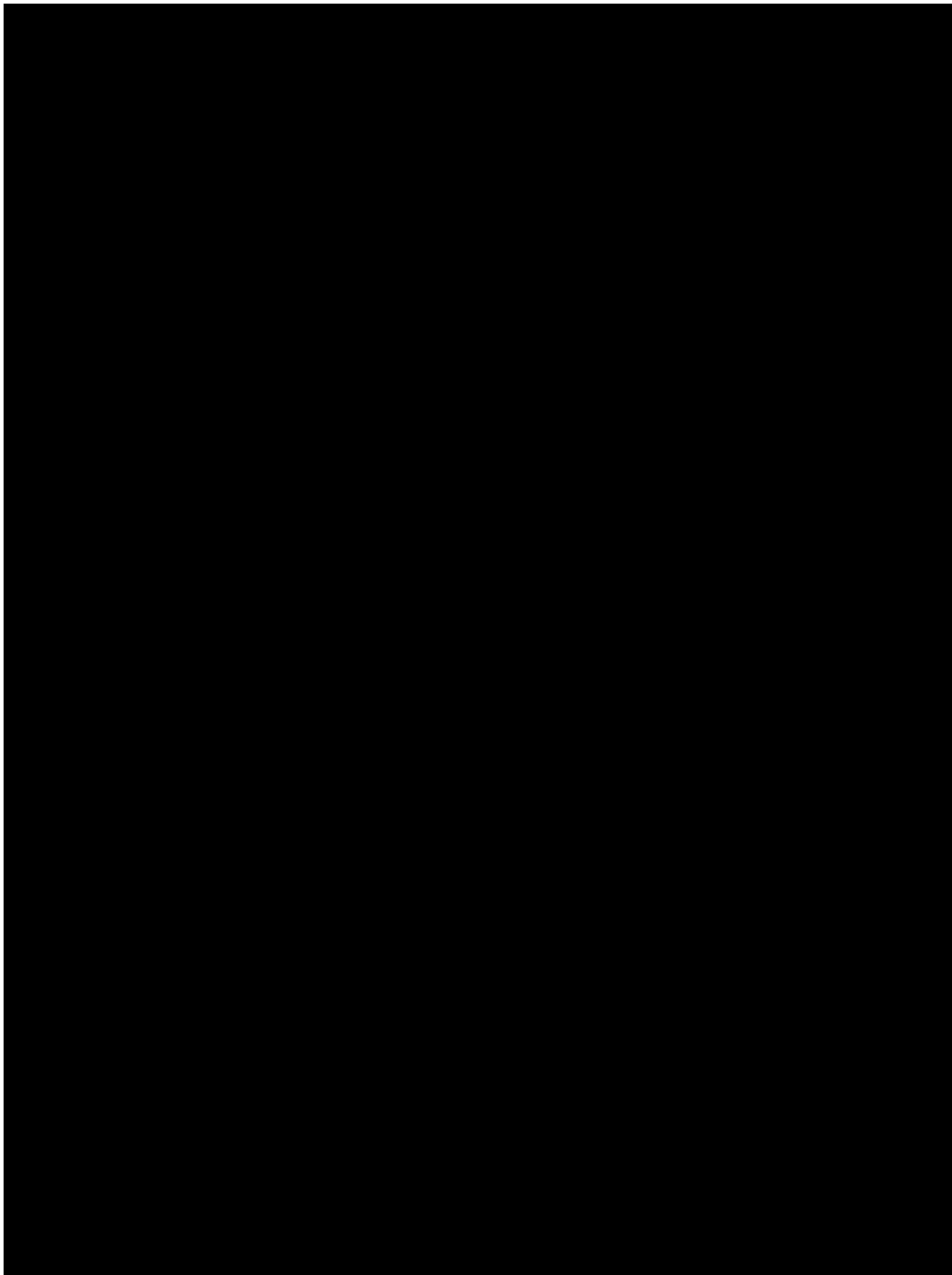
References

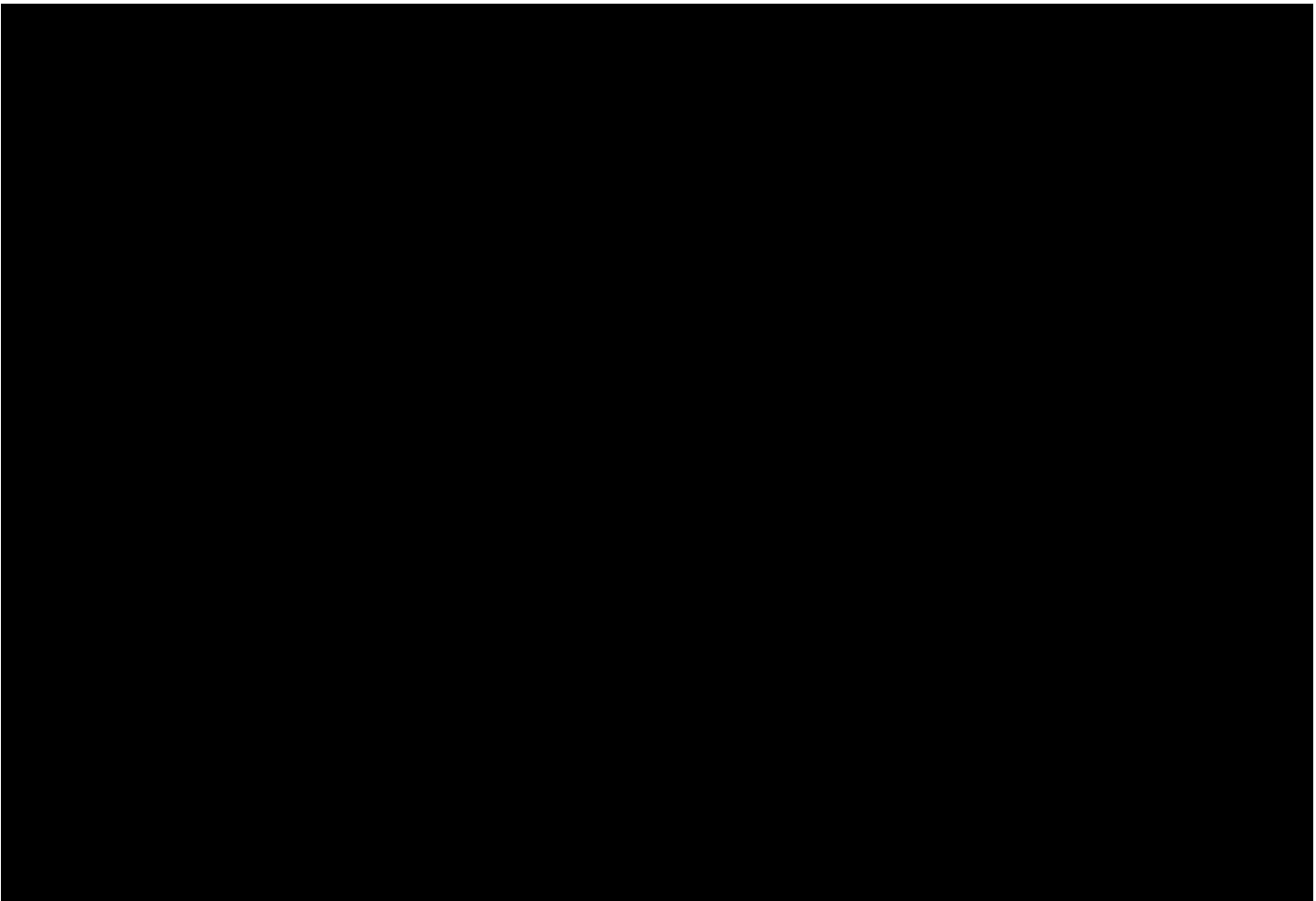
- [1] Hayes, Thomas, et al. "Simulating 500 million years of evolution with a language model." *Science* (2025)
- [2] Pacesa, Martin, et al. "One-shot design of functional protein binders with BindCraft." *Nature* (2025).

- [3] Woody Ahern. et al. “Atom-level enzyme active site scaffolding using RFdiffusion2” Nature Methods (2025).
- [4] John B. Ingraham. et al. “Illuminating protein space with a programmable generative model”. Nature (2023)
- [5] Hannes Stark. et al. “BoltzGen: Toward Universal Binder Design” (MIT) 2025.
- [6] Anna Lauko, et al. "Computational design of serine hydrolases". Science (2025)
- [7] Andy Hsien-Wei Yeh, et al. "De novo design of luciferases using deep learning". Nature (2023)
- [8] Joseph L. Watson et al. “De novo design of protein structure and function with RFdiffusion. " Nature (2023).
- [9] Markus Braun et al. “Computational enzyme design by catalytic motif scaffolding”. Nature (2025)
- [10] JD Schnettler et al. “Selection of a promiscuous minimalist cAMP phosphodiesterase from a library of de novo designed proteins” Nature Chemistry (2025)
- [11] J. DAUPARAS,et al. " Robust deep learning–based protein sequence design using ProteinMPNN." Science (2023).

Attached Evidence

As illustrated below, post-generation filtering is broadly adopted as a practical quality-control step in recent high-impact generative protein-design studies.





the 1990s, the number of people in the UK who are aged 65 and over has increased from 10.5 million to 13.5 million, and the number of people aged 75 and over has increased from 4.5 million to 6.5 million (Office for National Statistics 2000).

There is a growing awareness of the need to develop strategies to meet the needs of older people, and the importance of the role of the family in supporting older people. The Department of Health (1999) has published a strategy for older people, which sets out the government's commitment to older people and the need to develop a 'culture of ageing' in which older people are seen as a valuable resource. The strategy also sets out the need to develop a 'culture of ageing' in which older people are seen as a valuable resource.

The strategy also sets out the need to develop a 'culture of ageing' in which older people are seen as a valuable resource. The strategy also sets out the need to develop a 'culture of ageing' in which older people are seen as a valuable resource. The strategy also sets out the need to develop a 'culture of ageing' in which older people are seen as a valuable resource.

The strategy also sets out the need to develop a 'culture of ageing' in which older people are seen as a valuable resource. The strategy also sets out the need to develop a 'culture of ageing' in which older people are seen as a valuable resource. The strategy also sets out the need to develop a 'culture of ageing' in which older people are seen as a valuable resource.

The strategy also sets out the need to develop a 'culture of ageing' in which older people are seen as a valuable resource. The strategy also sets out the need to develop a 'culture of ageing' in which older people are seen as a valuable resource. The strategy also sets out the need to develop a 'culture of ageing' in which older people are seen as a valuable resource.

The strategy also sets out the need to develop a 'culture of ageing' in which older people are seen as a valuable resource. The strategy also sets out the need to develop a 'culture of ageing' in which older people are seen as a valuable resource. The strategy also sets out the need to develop a 'culture of ageing' in which older people are seen as a valuable resource.

The strategy also sets out the need to develop a 'culture of ageing' in which older people are seen as a valuable resource. The strategy also sets out the need to develop a 'culture of ageing' in which older people are seen as a valuable resource. The strategy also sets out the need to develop a 'culture of ageing' in which older people are seen as a valuable resource.

The strategy also sets out the need to develop a 'culture of ageing' in which older people are seen as a valuable resource. The strategy also sets out the need to develop a 'culture of ageing' in which older people are seen as a valuable resource. The strategy also sets out the need to develop a 'culture of ageing' in which older people are seen as a valuable resource.

The strategy also sets out the need to develop a 'culture of ageing' in which older people are seen as a valuable resource. The strategy also sets out the need to develop a 'culture of ageing' in which older people are seen as a valuable resource. The strategy also sets out the need to develop a 'culture of ageing' in which older people are seen as a valuable resource.

Reviewer #2 (Remarks to the Author):

A. Summary of the key results

This manuscript proposes a novel 16B-parameter generative model, Pinal, trained on 1.7B protein-text pairs derived from SwissProt annotations and augmented synthetic descriptions generated using LLMs and ProTrek for proteins from TrEMBL. Evaluations on three distinct settings (i.e., long and descriptive prompts, short and function-specific phrases, and specialized keywords) demonstrate that Pinal surpasses previous protein design methods including ProteinDT, Chroma, and ESM3. Further wet-lab experiments on designing four proteins from different functional classes (i.e., GFP, PETase, ADH, and H-protein) validate the design capability of Pinal. The pinal-designed proteins were functionally active and some of them (e.g., H-protein) surpass its natural type.

The proposed new paradigm of natural language-guided de novo protein design may have a broader impact on scientific discovery.

The conclusions and data interpretation are valid and robust. Some major concerns remain regarding the hypothesis and synthetic data construction. Please see F for more details.

Our Response:

We sincerely thank the reviewer for the careful and thorough evaluation of our manuscript, as well as for the insightful and constructive comments. We particularly appreciate the reviewer's suggestions regarding the motivation for the intermediate structural-token representation, the construction and use of synthetic protein-text data, and the consistency of the model description. These comments have helped us substantially clarify the rationale, evaluation, and presentation of our work. In response, we have conducted additional experiments, added new analyses, and revised the manuscript accordingly. Below, we provide a point-by-point response to each concern.

Question F1-1: The motivation for introducing structured token sequences as an intermediate step is not sufficiently justified. First, the generalization error in Fig. 1b could be influenced by many factors (e.g., different vocabularies). In fact, the red and blue curves are two different learning tasks (predicting structured tokens and amino acid tokens), thus, what a stronger generalization error can imply is uncertain. The next token prediction accuracy (and PPL) also does not reliably reflect a model's performance on protein design tasks (i.e., NTP accuracy may not be related to pLDDT, ProTrek scores). Therefore, the weaker generalization error does not convincingly support the necessity of introducing intermediate structure token sequences.

Our Response:

We sincerely thank the reviewer for this insightful comment. We fully agree that comparing losses across tasks (e.g., with **different vocabularies**) would be invalid, and that Next Token Prediction (NTP) accuracy does not directly translate to downstream protein design quality. We appreciate the opportunity to clarify these points below:

1. Clarification on the Structural Token Vocabulary (20 3Di Tokens vs. 20 Amino Acids)

In our setup, the protein structure is represented using Foldseek [1] 3Di tokens, which also form a **vocabulary of 20 states**. Thus, both the direct text-to-sequence task and the text-to-structure task are 20-way token prediction problems and share the same **uniform-random baseline accuracy of $1/20 = 0.05$** . We hope this clarification regarding the vocabulary size effectively addresses our reviewer's key concern.

At the same time, we also agree that even equal vocabulary cardinality does not imply equal intrinsic task difficulty. Therefore, rather than presenting Fig. 1b as a strict comparison of generalization, our intention is to highlight a practical contrast in learnability. Specifically, the significant accuracy gap reflects a meaningful divergence in the model's ability to learn the

respective text-to-modality mappings. It empirically illustrates that mapping text directly to amino acids (AA) is prohibitively difficult for this model (also see below explanation), which ultimately motivates our use of an intermediate structural step.

2. NTP accuracy only as a "Gatekeeper" for whether the conditional mapping is learnable

We fully agree with the reviewer that NTP accuracy or perplexity should not be used as a direct measure of protein design quality. In particular, a model with high token-prediction accuracy may still generate proteins that are not functional, not well folded, or poorly aligned with the input description. Therefore, we do not interpret NTP accuracy as a substitute for downstream design metrics such as pLDDT, PAE, TM-score, or ProTrek score.

Our intended use of NTP accuracy in Fig. 1b is more limited: it serves as a diagnostic test for whether the model has learned a non-trivial conditional mapping from text to the target modality under the given training setting. In this sense, NTP accuracy is not a sufficient condition for successful protein design, **but very low accuracy provides useful negative evidence**. If a conditional generative model cannot predict the next target token substantially better than a random 20-way baseline, it is unlikely to have captured enough information about the text-conditioned target distribution to support reliable generation.

This distinction is important for interpreting Fig. 1b. For the direct text-to-amino-acid setting, the test NTP accuracy remains close to chance, **around 0.1** compared with the **0.05 uniform-random baseline** (see figure below). This suggests that, under our training regime, the model learns only a very weak text-to-amino-acid mapping. By contrast, the text-to-3Di task reaches an accuracy of approximately 0.6 under the same 20-token vocabulary size, indicating that the text-to-structure-token mapping is much more learnable for the model.

Thus, our conclusion is not that higher NTP accuracy alone proves better protein design. Rather, the near-random performance of the direct text-to-amino-acid model identifies a clear bottleneck in learning the conditional mapping, whereas the substantially stronger text-to-3Di performance provides an empirical rationale for introducing structural tokens as an intermediate representation. We therefore present Fig. 1b as a learnability analysis that motivated our model design, and we further validate its relevance to protein design quality through the additional pLDDT experiment described below.

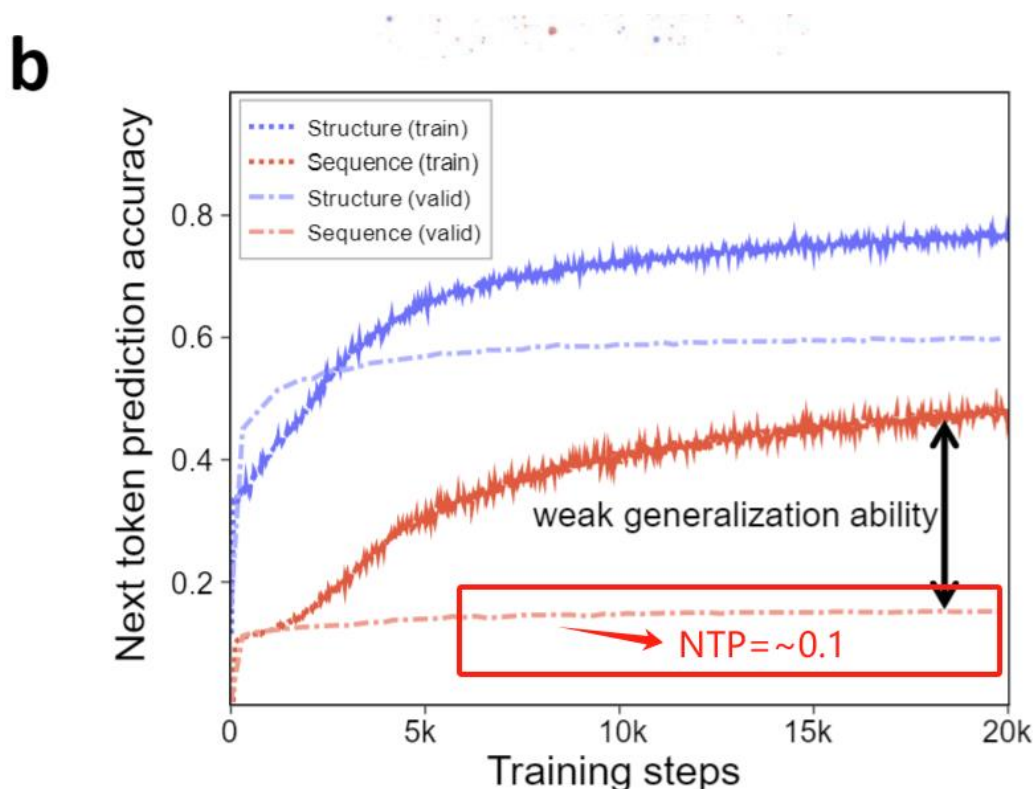


Fig. 1b: The NTP accuracy of the text-to-amino-acid model on the testing set is only around 0.1, close to the random baseline ($1/20=0.05$).

3. Direct Validation of Protein Quality via pLDDT (New Experiment)

To directly address the reviewer's concern about whether the two models differ in a foldability-related metric, we conducted an additional experiment evaluating (Methods 7.2) the generated proteins using ESMFold pLDDT, a commonly used proxy for structural plausibility in protein design.

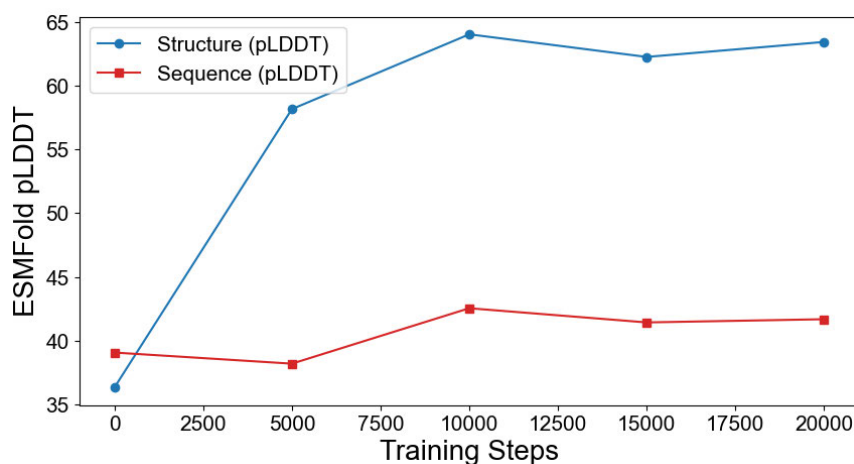


Fig. R1: Assessment of designed protein quality using ESMFold pLDDT, as suggested by Reviewer #2. Note that the text-to-structure-token model directly generates Foldseek 3Di tokens, which cannot be converted into 3D coordinates and therefore do not allow direct pLDDT comparison. We therefore used SaProt to convert the generated 3Di token sequences into amino-acid sequences, which were then folded by ESMFold to obtain the corresponding pLDDT scores. Other evaluation settings are the same as in Fig. 1b, see Methods 7.2 for details.

As shown in Fig. R1, the pLDDT results show a trend consistent with the NTP accuracy in Fig. 1b. The model trained with intermediate structural token sequences exhibits a substantial increase in pLDDT during training, whereas the direct text-to-sequence baseline remains at a much lower level and shows little improvement even after extended training.

In conclusion, we hope these details and the new experimental data help address the reviewer's concerns. First, the identical 20-token vocabulary sizes establish an equivalent random baseline for both tasks. Furthermore, our reference to NTP accuracy stems from the observation that the direct text-to-amino-acid model struggles to meaningfully exceed random guessing, which is now further corroborated by the poor pLDDT scores of its generated outputs. Together, we hope these findings better support the rationale for introducing the intermediate structured token sequences.

We have also added more analytical and comparative details in the revised version (Lines 74-79) and Methods 7.2.

[1] Kempen, et al. Fast and accurate protein structure search with Foldseek. Nature Biotechnology, 2024.

Question F1-2: Second, it could be more informative to directly compare models with/without intermediate structure token sequences. Some concurrent work shows that direct text-to-amino acid sequence can be competitive (e.g., ProDVa, <https://arxiv.org/abs/2505.18966>).

Our Response:

We thank the reviewer for this constructive suggestion and for pointing out ProDVa as a relevant recent baseline. In response, we have conducted the requested comparison and added the corresponding results in Supplementary Fig. 1.

1. Methodological Distinction

Although ProDVa ultimately generates amino-acid sequences, its design pipeline incorporates a different mechanism from a

purely direct text-to-sequence formulation. Specifically, ProDVA dynamically retrieves biologically meaningful protein fragments of varying lengths, including domains, families, and functional sites identified via InterPro, based on the input textual description. Therefore, in our view, ProDVA is not a strictly token-by-token generation-from-scratch baseline, as it also leverages structural and evolutionary priors from naturally occurring protein fragments through retrieval. This distinction is relevant when interpreting the comparison, as it suggests that direct generation of amino-acid sequences from natural language alone remains challenging.

2. Direct comparison under a unified evaluation framework

Following the reviewer’s suggestion, we benchmarked Pinal against ProDVA under the same evaluation protocol as used in Fig. 2d–f. As shown in Supplementary Fig. 1, Pinal consistently outperforms ProDVA across all major metrics, including functional alignment, structural alignment, and foldability-related metrics, for both **short** and **long** text prompts.

We observed that ProDVA performs relatively well on some prompts that resemble its training distribution, such as certain Swiss-Prot-style category descriptions or predefined prompt templates, but its performance decreases on more open-ended or compositionally complex inputs. This trend is consistent with its relatively restricted training regime: ProDVA is trained on ~712K text–protein pairs (primarily Swiss-Prot and Mol-Instructions), which is orders of magnitude smaller than Pinal's training corpus. Its training descriptions are largely templated, limiting its generalization to diverse, free-form natural language. Furthermore, we note that the ProDVA paper primarily benchmarked against our smaller 2B-parameter model. Under the comprehensive evaluation setting in Supplementary Fig. 1 (see below) against our main 16B system, the performance gap remains highly significant.

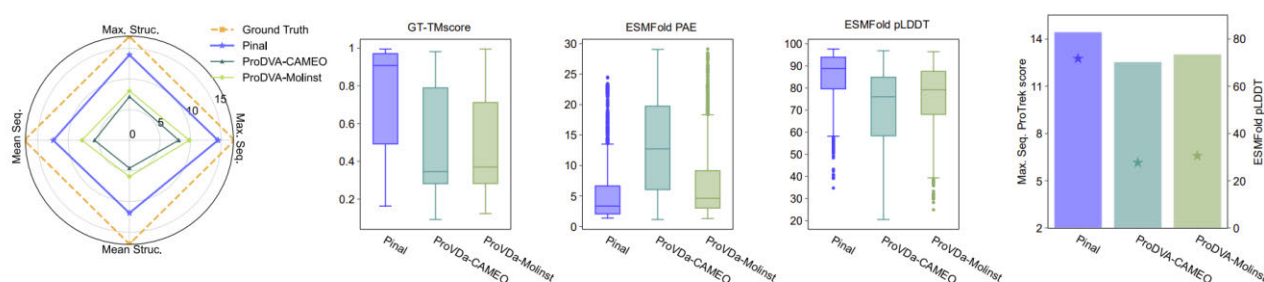


Fig. 1: Benchmarking Pinal against the recent text-to-sequence model ProDVA. Following the evaluation framework established in the main text, we compared the 16B-parameter Pinal model with ProDVA-CAMEO and ProDVA-Molinst on the standard test set. From left to right: (1) Radar plot and GT-TMscore distribution assessing functional alignment and structural accuracy, respectively; (2) ESMFold PAE and pLDDT distributions evaluating structural foldability; (3) Bar and star plots illustrating the foldability and textual alignment (ProTrek score), respectively, conditioned on short text prompts. Across all generative metrics, Pinal consistently outperforms ProDVA, with distributions indicating more reliable generation quality across samples.

3. Existing Baselines and Wet-Lab Validation

For completeness, we respectfully point out that our original manuscript already included a comparison with a purely direct text-to-sequence baseline (ProteinDT, Fig. 2d,e), against which Pinal also demonstrated clear advantages. In addition, although computational metrics provide useful proxies for protein quality, empirical wet-lab testing remains the gold standard for validating successful protein design. A key contribution of our work is the extensive wet-lab validation of the generated sequences, including 12 experimentally validated successful designs (Supplementary Table 6), which goes beyond the *in silico* evaluations currently reported for models such as ProDVA and ProteinDT.

4. Timeline and Publication Context

Finally, we briefly clarify the timeline regarding ProDVA. Since ProDVA was released on arXiv in May 2025—approximately nine months after our initial bioRxiv preprint [2] in August 2024—it is technically a subsequent work that builds upon and

extensively cites Pinal as a strong baseline. Nevertheless, we fully agree with the reviewer that comparing with the latest literature strengthens our paper. Therefore, we have included these new comparative results and discussions in the Supplementary Information (Method 1 and Fig. 1) for completeness.

[2] Dai, et. Toward De Novo Protein Design from Natural Language. <https://www.biorxiv.org/content/10.1101/2024.08.01.606258v1>, 2024/08.

Question F1-3: It would be informative to include experiments that assess whether the structural tokens generated by the T2Struct decoder can be directly used to design amino acid sequences (e.g., using **ProstT5** or **ProteinMPNN** as inverse folding models).

Our Response:

We sincerely thank the reviewer for this insightful suggestion. As detailed in our original manuscript (Extended Data Fig. 5a), we evaluated amino acid sequence design directly from the generated structural tokens using the standard SaProt model. Similar to ProstT5, SaProt is a widely recognized protein language model capable of generating amino acid sequences directly from 1D Foldseek (3Di) tokens. Previous studies [3,4] have also demonstrated that both models yield comparable performance (e.g., Supplementary Table 5 in [3]).

To directly address our reviewer's suggestion, we have now explicitly included ProstT5 as an additional baseline in our revised manuscript. As shown in Fig. R2 (below), our proposed SaProt-T/O achieves better semantic alignment and higher pLDDT scores than both standard SaProt and ProstT5, further highlighting the crucial role of text conditioning in our sequence design module.

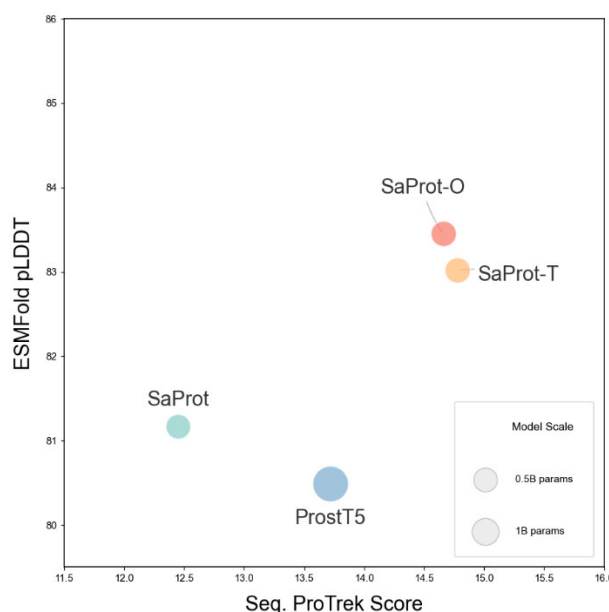


Fig. R2: Text conditioning is crucial for the sequence design module.

Regarding the suggestion to compare with ProteinMPNN, we respectfully note a fundamental difference in input modalities. ProteinMPNN requires explicit 3D backbone coordinates. In contrast, our T2Struct decoder generates 1D Foldseek structural tokens. Applying ProteinMPNN would require an intermediate step to convert these 1D tokens back into 3D structures, which could introduce cascading errors. Therefore, models like SaProt and ProstT5, which operate directly in the 1D token space, provide a more direct and methodologically sound validation of our decoder's immediate output. The results are present in Extended Data Fig. 5 in the manuscript.

[3] Su et., Democratizing protein language model training, sharing and collaboration. Nature Biotechnology. https://static-content.springer.com/esm/art%3A10.1038%2Fs41587-025-02859-7/MediaObjects/41587_2025_2859_MOESM1_ESM.pdf

Question F2: On the data augmentation: Although augmenting SwissProt annotations with LLMs is a good starting point, the synthetic dataset requires matching the unannotated proteins from TrEMBL with descriptions from SwissProt. This raises the following concerns:

Question F2-1: How accurate are the protein-text matching scores computed by ProTrek? Will those inaccurate (or with low ProTrek model confidence) protein-text pairs affect the model performance?

Our Response:

We sincerely thank the reviewer for raising this important point regarding the quality of the synthetic dataset. We fully agree that inaccurate protein–text pairs inevitably introduce noise, and that such noise can affect model performance if not properly controlled. In our augmentation pipeline, we therefore used ProTrek primarily as a retrieval and filtering tool, rather than assuming that its scores are perfectly calibrated estimates of semantic correctness. We address this concern from three perspectives.

1. High-confidence retrieval and quality control. In this work, we used ProTrek to retrieve candidate textual annotations for proteins in TrEMBL. From a machine learning perspective, training a discriminative retrieval model is typically much more tractable than training a generative modeling under limited supervision (because it only requires ranking candidate matches rather than generating full conditional distribution de novo). This makes searching model-based data augmentation a practical choice for constructing large-scale synthetic protein–text pairs. In fact, such retrieval-based augmentation strategy is also widely used in the computer vision community; for example, CLIP-like retrieval models are often used to curate and scale up text-image datasets for training large multimodal generative models. ProTrek (*Nature Biotechnology*, 2025) is a recent state-of-the-art retrieval model for protein-text annotation, and both the original work and our Extended Data Fig. 6 provide some initial analyses of its utility and robustness as a protein annotation tool.

At the same time, we agree that rigorously quantifying the correctness of every ProTrek-matched pair is challenging in practice, particularly at the billion-sample scale considered here. We therefore do not treat the absolute ProTrek score as a perfectly calibrated measure of semantic correctness, since its value may vary across protein families and functional categories. Instead, we use it as a relative confidence score for retrieval and filtering. This naturally introduces a trade-off between coverage and precision: lowering the threshold increases the number of synthetic pairs but may introduce more noise, whereas increasing the threshold improves confidence at the cost of scale. To control this trade-off, we adopted a conservative filtering strategy by retaining only the top 10 highest-scoring matches for each query (e.g. in Lines 641–642). As a result, the final synthetic dataset is enriched for relatively high-confidence protein-text pairs, rather than relying on all retrieved matches indiscriminately.

2. Empirical impact on downstream model quality. Ultimately, the most important test of the synthetic data is whether it improves the models trained on it. As shown in our paper, increasing the training corpus from 100 million to 10 billion training samples—most of which are synthetic—leads to notable improvements in ProTrek score (Fig. 2b, left y-axis), TM-score (Fig. 2b, right y-axis), novelty and diversity (Supplementary Fig. 3, Supplementary Method 2). These results suggest that, with the proposed data augmentation strategy by ProTrek, the synthetic pairs provide a net positive training signal despite residual noise.

3. Tolerance to residual noise. Finally, modern deep learning models can often tolerate a certain degree of label noise, provided that the overall signal-to-noise ratio remains sufficiently high. Our results are consistent with this view: the ProTrek-based augmentation pipeline appears to preserve enough useful signal for the model to benefit from the increased scale, rather than being dominated by a minority of inaccurate pairs.

More broadly, the proposed augmentation is essential in this setting. Without data augmentation, our model will be limited to

fewer than **600,000** annotated proteins in **Swiss-Prot**, which is a very small dataset for training large-scale autoregressive generative models. By using ProTrek-based retrieval, we expanded the corpus to nearly **200 million** proteins increasing the amount of training protein data by several hundred-fold and substantially improving the diversity of proteins observed during training. As mentioned above, our in silico results support the benefit of this scale-up, and our extensive wet-lab experiments further demonstrate that the proteins designed by the final model are experimentally valid.

Question F2-2: Since TrEMBL protein descriptions are paired with those from SwissProt, what happens if the corresponding (undiscovered or less-discovered) functions are not documented in the SwissProt database?

Our Response:

We thank the reviewer for raising this important question. We fully agree that anchoring TrEMBL proteins to Swiss-Prot descriptions has an inherent limitation: if a protein in TrEMBL has an additional undocumented function that is not captured in Swiss-Prot, then our current text assignment cannot represent that function explicitly. In such cases, the paired text may be incomplete.

Our goal in this augmentation step is therefore not to discover entirely new functions, but to expand the sequence and structural diversity associated with functions that are already represented in curated databases. We use Swiss-Prot as the textual anchor because it provides high-quality, expert-curated functional descriptions, even though it is necessarily incomplete. By contrast, annotations in TrEMBL are highly uneven in both coverage and reliability, and many entries are computationally inferred, sparse, or redundant. Relative to directly using raw TrEMBL annotations, this design helps reduce noise and redundancy in the text supervision.

More specifically, ProTrek allows us to associate curated functional descriptions with a much broader set of proteins in TrEMBL, thereby exposing Pinal to substantially greater sequence and structural diversity for functions that are already represented in Swiss-Prot (The original ProTrek paper demonstrates that its text-protein search can identify proteins with similar functions even when sequence or structural similarity is very low). For rare or underrepresented functions that have at least limited curated coverage, this strategy can enrich the training data by retrieving additional related proteins from TrEMBL. However, for functions that are entirely absent from curated resources, our current text-anchored pipeline cannot model them explicitly, and we agree that this remains an important limitation.

In summary, our current data expansion strategy is designed to broaden the sequence-structure coverage of known functions using reliable curated text, rather than to discover entirely new functions. We have described this limitation in the revised manuscript (Methods 7.4).

Question F2-3: Both the synthetic dataset construction and the evaluation on Pinal rely on the ProTrek model. Does this imply that Pinal's strong performance on the ProTrek score is affected by its training data predicted by ProTrek?

Our Response:

We thank the reviewer for this insightful comment. We agree that this is an important point to consider. Text-guided protein generation is still a very new task: before the release of our Pinal preprint, there was, to our knowledge, only one closely related prior work directly addressing text-guided protein generation, namely ProteinDT. Consequently, the community has not yet established a universally accepted, independent metric for measuring how well a generated protein matches a natural-language functional description. In this context, we use the ProTrek score as a semantic-alignment metric, but we interpret it as one component of the evaluation rather than as standalone evidence of model quality.

For this reason, we do not rely on the ProTrek score alone for alignment evaluation. In addition to ProTrek-based semantic alignment, we evaluated Pinal using complementary non-ProTrek-based metrics, including TM-score (Fig. 2b, right y-axis; Fig. 2d, right panel) and keyword recovery (Fig. 2g). TM-score provides an orthogonal structure-level assessment independent of ProTrek, while keyword recovery provides an additional function-related evaluation based on InterProScan-derived

annotations rather than ProTrek embeddings. Importantly, Pinal substantially outperforms the baselines not only in ProTrek score, but also in these non-ProTrek-based evaluations (see Fig. 2d,e,f,g) (a subsequent independent study [6] also reported the superior performance of Pinal in text-protein alignment using the EvoLlama score). These consistent improvements across independent structural and functional metrics suggest that Pinal's gains cannot be explained solely by better alignment with the ProTrek representation space. We have described this limitation and our use of complementary evaluations in the revised manuscript (Methods 7.7.3, Lines 838–849).

More broadly, the lack of fully independent and universally accepted computational evaluation protocols is a common challenge for emerging protein-design tasks, especially when experimental benchmarks are limited. Some degree of coupling between data construction, design guidance, and computational evaluation is therefore difficult to avoid. This issue is not unique to text-guided protein design. For example, in computational binder design, methods such as BindCraft use AlphaFold2-based hallucination as part of the design procedure and also use AlphaFold2-derived confidence or interface metrics, such as pLDDT, i_pAE, and i_pTM, for computational filtering and assessment, even though AlphaFold2-based complex prediction still has recognized limitations. This illustrates a broader challenge in the protein-design community: computational metrics are useful and often necessary, but they cannot fully replace experimental validation.

Ultimately, for protein design, the most reliable evaluation remains wet-lab validation. In this regard, our study provides experimental evidence for 12 successful designs (Supplementary Tables 6, 9, 10, 11, and 12) covering four distinct types of protein families (Section 5). By contrast, to our knowledge, the baselines considered here, including ProteinDT, Chroma, ESM3, and ProDVA, have not demonstrated biologically active proteins or enzymes generated directly from natural-language descriptions. For example, although ESM3 designed one functional GFP, this example relied on predefined sequence and structure constraints rather than the text prompt. We also thank the reviewers for any additional comments or suggestions on independent metrics or evaluation protocols, which would help further strengthen the assessment of text-guided protein generation.

[6] Kuang et al. PDFBench: A Benchmark for De novo Protein Design from Function (v1). <https://arxiv.org/pdf/2505.20346v1>

Question F2-4: It seems that the main purpose of applying LLM is to enhance the linguistic diversity of text-protein pairs (e.g., prompts in Supplemental Table 5). As one can expect, diversity could improve language alignment scores. It is less intuitive how diversity improves foldability and novelty (though not included in the evaluation metric).

Our Response:

We sincerely thank the reviewer for this insightful comment. We agree that LLM-based rewriting, by itself, should not be expected to directly improve the foldability or novelty of generated proteins. Its primary role in our pipeline is to increase the linguistic diversity and coverage of protein descriptions, thereby helping the model learn more robust mappings between natural-language prompts and protein-related concepts.

Importantly, however, the LLM rewriting is only one component of the overall data synthesis pipeline. The main source of sequence, structure, and functional diversity comes from ProTrek-based retrieval, which expands the paired training corpus from fewer than 600,000 experimentally annotated proteins in Swiss-Prot to nearly 200 million proteins from the broader AlphaFold/TrEMBL-scale protein space. Thus, the ProTrek+LLM data augmentation pipeline does not merely paraphrase existing annotations; rather, it exposes the model to a much broader sequence–structure–function landscape. We believe this broader coverage is the main reason why the synthetic data can improve properties beyond language alignment alone, especially novelty and diversity.

For foldability-related metrics highlighted by the reviewer, such as pLDDT and PAE, we do not necessarily expect a simple monotonic scaling trend with the size of the synthetic dataset. Foldability is influenced not only by data scale, but also by the structural quality of the training data and by the model components used for sequence generation. Proteins from the highly

curated Swiss-Prot subset have substantially higher average structural confidence, with an average AF2 pLDDT of approximately 87, whereas proteins from the much larger TrEMBL-derived synthetic subset have lower average structural confidence overall, with an average AF2 pLDDT of approximately 72. Therefore, adding large amounts of ProTrek-augmented data broadens sequence and structural coverage, but may also introduce training examples with lower average structural confidence, which can attenuate or mask improvements in pLDDT- or PAE-based foldability metrics.

This interpretation is also supported by our additional training analysis in Extended Data Fig. 4a. After pretraining T2struct on large-scale AFDB structures, further fine-tuning on a higher-confidence structural subset led to a clear improvement in the foldability of generated proteins. This result suggests that foldability-related metrics are strongly affected by the quality of structural supervision, rather than by dataset size alone. In addition, Pinal contains two core modules: T2struct and SaProt-T. SaProt-T is built upon the well-pretrained SaProt model, which itself can provide a strong prior for generating foldable amino acid sequences conditioned on structures. Therefore, even if T2Struct generates a structure that is not well matched to the input text prompt, SaProt-T (or SaProt) can still produce relatively high-quality protein sequences that fold with high pLDDT or low PAE scores. Consequently, a clean scaling ablation analysis of pLDDT and PAE would require jointly scaling multiple data sources, including the SaProt pretraining data, the SaProt-T fine-tuning data, and the T2Struct synthetic training data. Such a full factorial scaling study is computationally prohibitive and beyond the scope of the present work (for reference, pretraining the base SaProt model alone required approximately three months on 64 NVIDIA A100 GPUs, which would correspond to a substantial monetary cost if using commercial GPU resources).

Finally, regarding novelty and diversity, the effect of the proposed augmentation is more direct. By expanding the training corpus from a small curated set to a much broader protein space, the model is exposed to many more sequence and structure patterns than are available in Swiss-Prot alone. Therefore, the gains in novelty and diversity arise not because the LLM directly optimizes these properties, but because the overall ProTrek+LLM strategy broadens the biological design space available during training. In response to the reviewer's concern, we have added explicit evaluations of diversity and novelty in Question F2-5 (see below) and Supplementary Fig. 3, which provide direct empirical support for this interpretation.

We also note that although the alignment metrics such as TM-score (Fig. 2b, right y-axis; Fig. 2d right panel) and ProTrek score are not direct foldability metrics, they are difficult to optimize in practice if the generated structures are highly implausible. In particular, achieving a high TM-score against a reference structure generally requires a reasonable degree of structural plausibility. Consistent with this, our analysis in Extended Data Fig. 4 shows that generated proteins with high ProTrek scores tend to also exhibit high ESMFold pLDDT. We therefore view these metrics as complementary indicators rather than standalone measures of text alignment. In fact, improving ProTrek score and TM-score can be more challenging in practice than improving pLDDT/PAE-based foldability metrics alone, because high values of ProTrek score and TM-score require the generated proteins not only to be structurally plausible, but also to remain well aligned with the conditioning text or reference structure. Finally, we note that the pLDDT and PAE performance reported in Fig. 2e should be viewed as the outcome of the overall Pinal framework and its training-data strategy (see analysis in Extended Data Fig. 4), including the use of pretrained SaProt-based model, rather than as a direct consequence of LLM rewriting or dataset scaling alone.

Question F2-5: It would be beneficial to inspect how augmented text data (with some controlled perturbation) contributes to other evaluation metrics, besides alignment scores in Fig. 2b.

Our Response:

We thank the reviewer for this constructive suggestion. To further examine whether augmented text-protein data improve aspects beyond alignment-related metrics, we performed an additional analysis of the diversity and novelty of proteins generated by Pinal across different training-data scales.

Because novelty and diversity are only meaningful for valid designs, we first applied a consistent quality filter to all generated proteins before downstream analysis. Specifically, we retained designs with reliable predicted foldability and functional alignment, defined by ESMFold pLDDT > 70 and ProTrek score > 10. This filtering step was applied uniformly across all

data scales and was intended to ensure that the diversity and novelty analyses were performed on proteins that are both structurally plausible and reasonably consistent with the input design prompt. Without such a quality constraint, completely random sequences could appear artificially diverse or novel despite being biologically irrelevant.

We first evaluated generative diversity. Following a strategy adopted in DISCO [5], we quantified both sequence diversity and structural diversity by clustering the generated proteins (after above quality filtering) in sequence and structure space, respectively. As shown in Supplementary Fig. 3 (see screenshots below) left, both the number of sequence clusters and the number of structural clusters increase overall as the amount of training data increases. These results suggest that scaling augmented text-protein data expands the diversity of high-quality proteins generated by the model.

We next assessed novelty by comparing the generated proteins against UniRef100 and experimentally determined structures in the PDB. As shown in Supplementary Fig. 3 (see screenshots below) right, the number of sequence-novel and structure-novel designs increases progressively with dataset scale. These results also indicate that large-scale synthetic data improve not only alignment-related metrics, such as ProTrek score, TM score, and keyword recovery, but also the ability of the model to generate high-quality proteins that explore less populated regions of sequence and structural space.

We have added these analyses to the revised manuscript (Supplementary Method 2, Supplementary Fig. 3) to clarify the broader benefits of data scaling beyond text-protein alignment.

[5] Rector-Brooks J, Lambert T, Skreta M, et al. General Multimodal Protein Design Enables DNA-Encoding of Chemistry[J]. arXiv preprint arXiv:2604.05181, 2026.

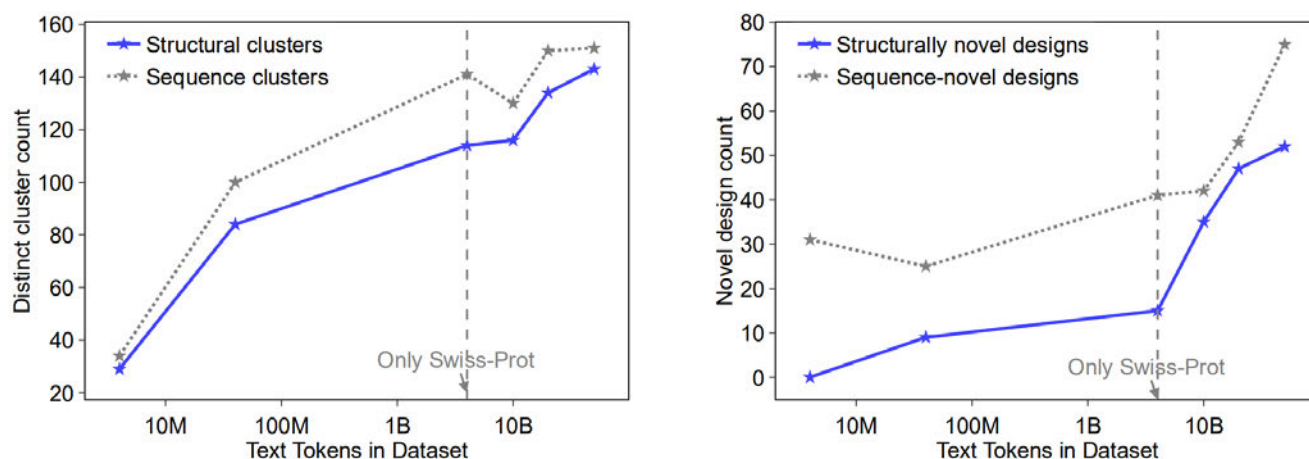


Fig. 3: Scaling of training data increases the yield of diverse and novel quality-filtered designs. To ensure rigorous assessment, all generated candidates across different training scales were pre-filtered for high structural confidence (ESMFold pLDDT > 70) and functional alignment (ProTrek score > 10) prior to downstream evaluation. **Left**, Diversity analysis of the generated pools from each training scale, quantified by the total number of distinct clusters in structural (blue) and sequence (grey) spaces. **Right**, Novelty analysis of these quality-filtered proteins. Novelty is reported as the number of designed candidates classified as structurally novel relative to the PDB (blue solid line) or sequence-novel relative to UniRef100 (grey dotted line). The number of novel designs exhibits an overall increasing trend with training-data scale, suggesting that large-scale synthetic data augmentation is associated with an increased yield of novel, quality-filtered candidates and broader exploration of the accessible design space.

Question F3: On T2Struct and SaProt-T's text encoding

Question F3-1, F3-2: The main text (lines 97-100) states that T2Struct's encoder is from the pretrained PubMedBERT model, while Section 7.1 (lines 541-542) indicates that T2Struct's encoder is initialized from a pretrained T5. It is unclear which text encoder is applied in T2Struct. The main text (lines 105-107) states that SaProt-T's text representation is obtained by the

projection of T2Struct's encoder (which is unclear, as mentioned in the above comments), while Section 7.1 (lines 552-554) states that SaProt-T introduces PubMedBERT as another text encoder.

Our Response:

We thank the reviewer for pointing out this inconsistency, and we apologize for the confusion. T2Struct uses a T5 encoder. The statement in the main text was left over from an earlier version of our experiments, in which both T2Struct and SaProt-T used PubMedBERT as the text encoder. During model scaling, we replaced the T2Struct encoder with T5 because T5 provides a broader range of model sizes (100M, 300M, and 4.8B), which allowed us to scale T2Struct from 200M to 1.2B and 15.5B parameters (Lines 161-168). In contrast, PubMedBERT is only available at approximately the 100M scale. We have revised Lines 96–99 to correctly state that T2Struct uses a T5 encoder.

Question F3-3: Fig. 1d also shows two separate text encoders (text encoder 1 and text encoder 2). This inconsistency is confusing, and the authors should carefully check and verify the model and the results.

Our Response:

We thank the reviewer for pointing this out. As clarified above, Text Encoder 1 in Fig. 1d is the T5 encoder used by T2Struct, whereas Text Encoder 2 is PubMedBERT used by SaProt-T. We have revised the relevant text (Line 105) and added an explanation in the figure caption to make this distinction explicit.

Question F3-4: If the text embeddings used for decoding structural tokens and those used for SaProt-T are indeed different, could you explain why two separate text encoders are needed to encode the same textual description? The authors are also encouraged to include additional ablation studies (e.g., sharing the same text encoder).

Our Response:

We thank the reviewer for this important question. We used two separate text encoders mainly because T2Struct and SaProt-T play different roles and impose different requirements on text conditioning. T2Struct is responsible for inferring structural tokens directly from the natural-language prompt, and therefore benefits from a scalable and expressive text encoder. We used T5 for T2Struct because it offers more choices for model sizes, including 100M-, 300M-, and 4.8B-scale variants, which is important for its scaling experiment in Fig. 2c. In contrast, SaProt-T is conditioned not only on the text prompt but also on the 3Di structural token sequence generated by T2Struct. Therefore, its demand on text-encoding capacity is empirically less stringent, because the structural token sequence already provides strong guidance for sequence generation, as supported by the comparison between SaProt and SaProt-T in Fig. R3. Therefore, we retained the ~100M PubMedBERT encoder for SaProt-T.

Following the reviewer's suggestion, we conducted an ablation study in which we replaced the PubMedBERT text encoder in SaProt-T with the 4.8B T5 text encoder used in 15.5B T2Struct model, so that both modules use the same text encoder. As shown in Fig. R3, labeled "SaProt-T-T5xxl_encoder," this modification led to only a negligible performance change. This result indicates that Pinal performs similarly when SaProt-T uses either PubMedBERT or the much larger T5 encoder, suggesting that the choice of text encoder in SaProt-T has little impact on the overall performance.

For T2Struct, our model-scaling analysis in Fig. 2c shows that increasing the overall model capacity substantially improves performance. As the T2Struct model size increases from 200M to 1.2B and then to 15.5B parameters, we observe consistent improvements in next-token prediction (NTP) loss, ProTrek score, and ESMFold pLDDT (Fig. 2c, right y-axis). The best-performing T2Struct is the largest 15.5B version, which uses a 4.8B-scale T5 text encoder together with a 10.7B-parameter structure decoder. We note, however, that this scaling experiment treats T2Struct as an integrated encoder-decoder model, and therefore does not isolate whether the performance gain comes primarily from the larger text encoder, the larger structure decoder, or their combination.

A fully controlled ablation to isolate the effect of text-encoder size would require fixing the structure decoder, for example at

the ~10B scale, and retraining multiple T2Struct variants with different-sized text encoders. Because training such large-scale generative models generally requires large-scale training data to yield meaningful conclusions, these ablations would need to be performed on the same billion-scale training corpus used in this work, comprising hundreds of billions of tokens. Based on our training experience, **each such run** would require on the order of 100 NVIDIA A100 GPUs for about one month, and multiple such runs would be needed for a controlled comparison. While such an experiment would be informative, the required computational cost would be prohibitively high for the present study.

Finally, we would like to mention that the current two-encoder design emerged during the iterative development of Pinal, rather than from an a priori decision to use different text encoders. In the early version of Pinal, when both the model and training dataset were much smaller, T2Struct and SaProt-T both used PubMedBERT as the text encoder. As the study progressed, we substantially expanded the training corpus and scaled T2Struct from the 200M-parameter level to the 15.5B-parameter level. In this setting, continuing to use a ~100M-parameter PubMedBERT encoder for T2Struct appeared empirically mismatched to the much larger structure decoder and the substantially enlarged training data. We therefore replaced the T2Struct text encoder with a 4.8B T5 encoder, while keeping the SaProt-T encoder unchanged, leading to the current design with two different text encoders. We sincerely appreciate the reviewer’s insightful suggestion, which helped us clarify this design choice. If the paper is accepted, we would also be happy to make the response letter publicly available, so that readers can benefit from the reviewers’ insightful questions and our clarifications.

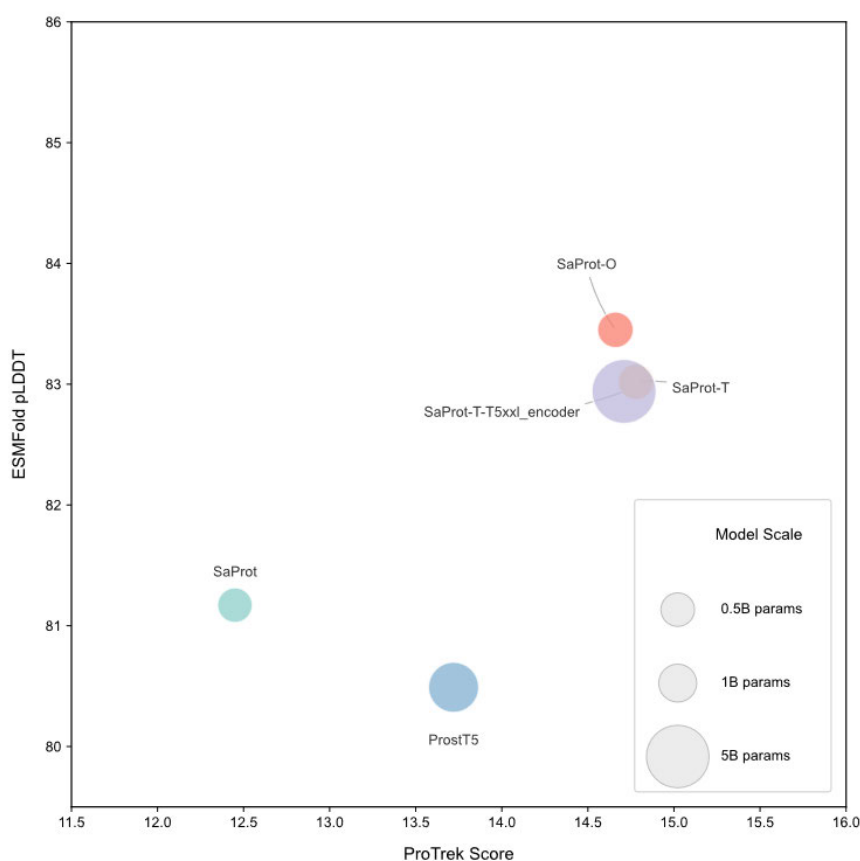


Fig. R3: Comparison of model performance across different text encoders and scales. The bubble chart illustrates the relationship between foldability (ESMFold pLDDT, y-axis) and functional alignment (ProTrek Score, x-axis). The size of each bubble represents the model's parameter scale.

We are grateful to Reviewer 2 for the thoughtful and constructive suggestions, which have helped us improve both the technical rigor and clarity of the manuscript. We hope that the additional experiments, new analyses, and revisions described above satisfactorily address the reviewer’s concerns. We would be happy to provide any further clarification or supporting materials if needed.

Referee #3:

I co-reviewed this manuscript with one of the reviewers who provided the listed reports.

Referee #3 (Remarks on code availability):

I have reviewed the code available at <https://github.com/westlake-repl/Denovo-Pinal>. Although the training scripts and datasets are not publicly accessible, the inference code runs correctly. The instructions are well documented, and I believe the codebase is useful to the community.

Our Response:

We sincerely thank Referee #3 for taking the time to examine our codebase and for confirming that the inference code runs correctly and is well documented. We are also grateful for the referee's positive assessment that the codebase will be useful to the community. For transparency during the review process, we have provided the reviewers with full access to our training and inference scripts, model weights, training datasets, and test datasets. We have also clarified this in Section 10, "Training Data and Code Availability", of the revised manuscript.

Point-by-point Response Letter

We sincerely thank the editor and reviewers for their careful re-evaluation of our revised manuscript and for their constructive comments. We are encouraged that Referee #1 considers the manuscript suitable for publication, and we appreciate Referee #2's careful attention to methodological clarity and reproducibility. In this revision, we have focused on addressing the remaining comments by clarifying the candidate-generation workflow, providing additional analyses, and releasing the materials for the related ablation study to facilitate reproducibility. We have revised the manuscript and Supplementary Information accordingly, with the key changes highlighted in purple.

Reviewer #1 (Remarks to the Author):

The authors have done a thorough and compelling job of responding to my concerns. I think this manuscript is well-deserving of publication.

We sincerely thank the reviewer for the positive assessment of our revised manuscript and for recognizing our efforts in addressing the previous concerns.

I have a few minor comments:

1. Keeping the name Pinal is fine for the reasons the authors give. I recommend explaining the origins of the name when Pinal is first mentioned in the introduction.

We thank the reviewer for this helpful suggestion. We have now explained the origin of the name Pinal when it is first mentioned in the Introduction (line 53 of the revised manuscript).

2. For each of the four design tasks, it would be valuable to say what the sequence identity with the closest natural sequence is. I see there's a range given, but I'd like to know the specific number for each task. I'm curious what the closest natural sequence is for the PETase? If such a comparison doesn't make sense, the authors should consider explaining why as it may highlight a strength of the approach.

We thank the reviewer for this constructive suggestion. In the previous revised version, we had reported in Supplementary Table 6 the sequence identity of each active, experimentally validated Pinal-designed protein to its closest natural sequence. Following the reviewer's recommendation, we have now expanded this table to include the Swiss-Prot and UniRef100 accession numbers of the corresponding closest natural sequences.

Table 6: Sequence identity between experimentally validated active Pinal-designed proteins and their closest matches in Swiss-Prot and UniRef100.

Sequence ID	Closest Swiss-Prot ID	Seq. Identity to Swiss-Prot	Closest UniRef100 ID	Seq. Identity to UniRef100
pinalGFP	Q8ISF8	30.8%	UPI0021506B9F	33.6%
pinalGFP-g	P42212	27.0%	UPI000FF8750E	31.1%
pinalADH01	P48814	54.5%	UPI0008115CC9	55.2%
pinalADH02	F8DVL8	57.4%	A0A3T0D359	62.9%
pinalADH03	P0DMP5	47.4%	A0A437RFF1	49.8%
pinalADH04	Q96XE0	49.0%	A0A8T5A9C9	51.0%
pinalADH05	F8DVL8	56.4%	A0A3T0D359	63.4%
pinalPETase36	E9LVH9	60.2%	A0A2R7SV75	62.6%
pinalH02	P39726	50.6%	G3B5R3	54.9%
pinalH04	P39726	50.6%	P39726	55.8%
pinalH07	O67573	55.8%	A0A3M2GYW8	58.2%
pinalH08	Q89I87	54.4%	A0A931HCK5	55.5%
pinalH12	O67573	55.4%	A0A7C5QLU9	57.9%

Reviewer #2 (Remarks to the Author):

1. Some concern about the necessity of an intermediate structure sequence is still there.

1.1 We have observed some successful text-to-sequence models (e.g., ProLLaMA), a weak text-to-sequence model here needs some further discussion.

We thank the reviewer for raising this important point. We appreciate the opportunity to clarify the purpose, basis, and intended scope of this experiment.

The preliminary experiment described in Section 7.2 (Methods) was designed as a strictly controlled, head-to-head comparison between two generation paradigms: **text-to-amino-acid sequence generation** and **text-to-structure-token sequence generation**. To avoid confounding factors, the two models used exactly the same encoder-decoder architecture, the same vocabulary size, the same curated SwissProt-Annot protein-text pairs, the same optimization protocol, and the same 30% sequence-identity split. The only difference was the prediction target: amino-acid tokens versus Foldseek 3Di structural tokens. Therefore, this experiment specifically tests which target representation is more learnable under the same data, architecture, and training budget.

Under this controlled setting, the direct text-to-amino-acid model achieved a very low next-token prediction accuracy on the validation set, approximately 0.1, close to the random baseline of 0.05 for a 20-token amino-acid vocabulary. We agree that next-token prediction accuracy is not itself a direct measure of the foldability or biological function of designed proteins. However, as explained in our previous round of response, in this particular controlled comparison, such a low next-token prediction accuracy provides meaningful negative evidence: if a conditional generative model cannot predict amino-acid tokens substantially better than a random-guessing baseline under a strict held-out split, it is unlikely to have captured a reliable text-conditioned amino-acid distribution. Consistently, Supplementary Fig. 2 also shows substantially lower ESMFold pLDDT scores for the direct text-to-amino-acid baseline. The comparison code (including NTP and pLDDT) is available at: <https://github.com/SuperCarryDFY/235091513709eab1b44080768e2caceb>.

We understand the reviewer's concern that the ProLLaMA paper appears to report promising results, such as seemingly favorable pLDDT scores. However, it should be noted ProLLaMA and our preliminary ablation study use different settings. In our experiment, both decoders were trained **from scratch** within the same controlled framework for a fair comparison. By contrast, ProLLaMA uses a decoder that has already been pre-trained on a large protein sequence corpus of approximately 50 million sequences before being fine-tuned on keyword-protein pair data. This pretraining provides extensive prior knowledge of protein sequence statistics. Therefore, it is plausible that ProLLaMA may achieve higher predicted structural metrics (e.g., pLDDT) than a **from-scratch text-to-sequence model trained only on curated SwissProt-scale protein-text pairs** (in fact, our direct empirical evaluation suggests that ProLLaMA's performance **remains limited**; see below **Table R1**) (Code comparison link is provided in the response of Q 1.3).

The weak performance of the direct text-to-sequence model in our preliminary experiment is mainly due to the limited scale of high-quality protein-text training data. As described in Methods (Section 7.2), in this preliminary study, we used SwissProt-Annot as the training dataset because SwissProt is currently the most comprehensive manually reviewed database for protein-function annotation and provides a reliable source of high-quality protein-text associations for this task. In other words, the community is **largely constrained by the scale of SwissProt-level protein-text data**: the total number of proteins with high-quality curated annotations remains on the order of only several hundred thousand (less than 600,000). Therefore, this limitation is not a consequence of our experimental design, but rather reflects a practical bottleneck in the availability of gold-standard protein-text annotation data.

For this reason, our preliminary study asks an important and deliberately constrained question: **given the amount of reliable protein–text data currently available, can a direct text-to-amino-acid model be learned from scratch under a strict sequence-identity split?** This question is central to our motivation because it reflects the realistic data regime available for high-confidence text-conditioned protein generation. Our results suggest that, under this regime, direct amino-acid generation is substantially harder to learn than structure-token generation. At the same time, we fully agree with our reviewer that this result should not be broadly interpreted as evidence that direct text-to-sequence generation is fundamentally impossible. Rather, the result indicates that direct text-to-sequence learning from **limited curated protein–text pairs is highly challenging** without prior large-scale protein-sequence pretraining.

Regarding the reported performance of ProLLaMA, we note that ProLLaMA and Pinal are evaluated under substantially different settings. ProLLaMA is mainly a **keyword-conditioned** generation model: it uses a finite set of only **11,268** superfamily annotations, and all annotation keywords in the test set have already **appeared during training**. Moreover, its instruction-tuning data are restricted to proteins shorter than 256 amino acids. Therefore, the evaluation mainly tests generation conditioned on a fixed and previously seen keyword vocabulary within a relatively short-sequence regime. In contrast, Pinal targets **free-form natural-language-conditioned** protein design. Its inputs are open-ended textual descriptions rather than fixed annotation keywords. All test descriptions of Pinal are rewritten to reduce direct memorization, and the training and validation proteins are separated by a 30% sequence-identity threshold. Pinal also supports much longer protein generation, up to 1022 amino acids.

As a result, ProLLaMA is evaluated in a more restricted setting. Learning to generate proteins from **only 11,268 seen** annotation keywords is different from, and **generally easier than**, open-ended generation from diverse natural-language descriptions. This difference may partly explain why ProLLaMA reports favorable metrics. Thus, we believe ProLLaMA should be viewed primarily as a keyword-conditioned protein generation model under a constrained evaluation protocol, rather than as direct evidence that text-to-amino-acid generation can solve open-ended natural-language-conditioned protein design. Finally, we note that ProLLaMA does not report wet-lab validation of its generated proteins. Therefore, its computational metrics, while useful, should not be considered equivalent to experimental evidence of successful protein design, which remains the most critical evaluation criterion in this field. (In contrast, our revised manuscript reports the experimental validation of more than 60 active Pinal-designed proteins across multiple protein families.)

To further address our reviewer’s concern, we additionally conducted a direct empirical comparison with ProLLaMA **under its own keyword-conditioned setting**. The code for this comparison is provided in our response to Question 1.3. Specifically, we **randomly sampled 500 keyword** prompts from the ProLLaMA training dataset and generated five sequences for each prompt using each method, resulting in 2,500 generated sequences per method. We then evaluated the generated proteins using predicted pLDDT and ProTrek score. The results are summarized in Table R1.

Table R1. Direct comparison between ProLLaMA and Pinal under the ProLLaMA keyword-prompt setting.

Method	Conditioning setting	# Prompts	# Sequences per prompt	# Generated sequences	Predicted pLDDT ↑	ProTrek score ↑
ProLLaMA	Keyword prompt	500	5	2,500	55.9	4.02
Pinal	Keyword prompt	500	5	2,500	81.9	9.18

This direct comparison further supports our clarification above. In this specific keyword-prompt evaluation, ProLLaMA-generated sequences show substantially lower predicted structural confidence and weaker protein–text alignment than those generated by Pinal, **even when the prompts used for evaluation are directly sampled from ProLLaMA’s training data**. These results suggest that text-to-amino-acid generation can remain challenging even with large-scale sequence pretraining and prompt overlap between training and evaluation. Importantly, however, we do not interpret this comparison as proving

that all direct text-to-sequence approaches are fundamentally infeasible. Rather, it shows that the existing ProLLaMA results do not invalidate our narrower conclusion: under a controlled setting that reflects the currently limited availability of high-quality protein–text data, text-to-structure-token sequence generation is more learnable than direct text-to-amino-acid sequence generation.

After the above clarification, one question may still remain: whether a direct text-to-amino-acid sequence model could achieve strong performance if trained on our entire dataset, i.e., the 1.7 billion synthetic protein–text pairs, corresponding to several hundred billion training tokens, constructed by our own pipeline in this work. This dataset is 100–1000 times larger than SwissProt-Annot. We agree that this is an interesting and important question. However, rigorously answering it would require a full-scale ablation experiment with nearly the same computational cost as training Pinal from scratch. In our setting, this would involve training a large model on the full dataset for approximately one month using around 100 NVIDIA A100 GPUs. Such an experiment is substantially beyond the scale of the preliminary motivational ablation presented in this work.

We thank the reviewer for this insightful comment. In the revised manuscript (Lines 72–85 and Methods 7.2), we have deliberately narrowed the claim supported by this experiment. Specifically, we do not claim that direct text-to-sequence generation is infeasible in general. Rather, our conclusion is that, under the controlled evaluation setting, text-to-3Di-token generation is more learnable than direct text-to-amino-acid generation when both models are trained from scratch using the same curated set of reliable protein–text pairs. This is why we describe this experiment as a preliminary motivational study in the manuscript.

At the early stage of this work, dating back to 2022–2023, Pinal was initially developed using SwissProt-Annot rather than large-scale synthetic data. As the project progressed, we further explored synthetic data construction and scaling strategies. We believe this reflects a natural research trajectory rather than an attempt to overstate the conclusion from a small ablation study.

We sincerely thank the reviewer for this insightful and constructive suggestion, which helped us clarify the scope and limitations of this preliminary study in the revised manuscript.

1.2 Another hint is from the unconditional amino acid sequence generation (e.g., protgpt2, progen2). Though they are not directly comparable here, they suggest that a model can have non-trivial next token prediction performances even if it only looks at previous amino acid tokens.

We thank the reviewer for the insightful comment. We fully agree that unconditional amino-acid sequence generation models, such as ProtGPT2 and ProGen2, can achieve non-trivial next-token prediction (NTP) performance by modeling protein sequences autoregressively from preceding amino-acid tokens. Our paper does not deny or contradict this observation.

Our point is instead that the **text-to-amino-acid generative mapping** considered in Pinal is substantially more challenging. In this setting, the model must generate a functional protein sequence conditioned on the given textual information, while high-confidence text–protein paired data for training are highly limited. The combination of **the vast amino-acid sequence space** and the limited availability of paired textual training data is a key factor behind the low next-token prediction performance observed in our preliminary study, where the text-to-sequence model was trained from scratch on curated text–protein pairs. Even the most comprehensive curated database, Swiss-Prot, contains fewer than 600K protein sequences with high-quality annotations. In contrast, unconditional protein language models do not require paired text–protein data and can be trained directly on large-scale unpaired protein sequence databases, such as UniProt and large metagenomic datasets, whose scale can be 1000–10000 times larger than Swiss-Prot. Therefore, it is entirely reasonable that unconditional sequence-generation models trained on much larger unpaired protein corpora can achieve better next-token prediction accuracy.

Finally, for text-guided protein design, next-token prediction (NTP) accuracy and predicted structural confidence, such as pLDDT, are only a basic requirement but not sufficient criteria. The central challenge is not only to generate protein-like sequences with plausible structures, but also to ensure that the generated proteins are semantically aligned with the conditioning text and exhibit the intended functions. Thus, while unconditional models such as ProtGPT2 and ProGen2 demonstrate non-trivial NTP performance, this does not necessarily imply that these generated sequences would exhibit specific functional properties in wet-lab settings, nor that such pre-trained sequence models would automatically acquire strong text-conditioned generation capability once fine-tuned on a Swiss-Prot-scale text–sequence paired dataset (the above mentioned ProLLaMA results in Table R1 serve as such a concrete counterexample). This is precisely the gap that Pinal aims to address.

We sincerely thank the reviewer for raising this important point, which helped us clarify the distinction between unconditional sequence modeling and text-guided protein design.

1.3 Would you please provide code/data for reproducing curves in Fig. 2?

We thank the reviewer for this request. We have provided the code and source data needed to reproduce the curves for the text-to-sequence and text-to-structure experiments (including both pLDDT and NTP accuracy) at:

<https://github.com/SuperCarryDFY/235091513709eab1b44080768e2caeeb>.

In addition, the evaluation code for the ProLLaMA comparison described above is available at:

https://github.com/SuperCarryDFY/ProLLaMA_eval. The corresponding Pinal evaluation code under the ProLLaMA superfamily-prompt setting is available at:

<https://github.com/SuperCarryDFY/b097bade29ef6c984ea1c6f28ec5ece165c3765a459f16085d618a97f9a60286#evaluation-on-prollama-superfamily-prompts>.

2. The authors' response to the question "how scaling influences pLDDT/PAE" is clear. I would suggest adding the discussion to the revised version. It helps us to know where/why scaling works.

We thank the reviewer for this helpful suggestion. We have added a brief pointer in the main text (Line 163) and incorporated the detailed discussion into **Supplementary Note 1**, where we discuss why data scaling primarily improves alignment, novelty, and diversity, whereas its effects on foldability-related metrics such as pLDDT and PAE are more nuanced.

4.3 Compared the two versions of Tables 8-11, PETases36 is completely new, and Table 9 (ADH) removes some rows. The update may need some explanation.

We thank the reviewer for pointing out these changes. We would like to address this point first, as it provides useful context for addressing several subsequent comments.

The newly included pinalPETase36 stems from a new PETase validation experiment conducted during the first revision cycle. In the first round of review, Reviewer 1 expressed a concern regarding our initial demonstrations, noting that Pinal was intended to enable protein design without requiring detailed structural or biophysical insight from the user. Specifically, the reviewer pointed out that our initial PETase screening relied on expert-guided filtering—such as selecting for the canonical catalytic triad and specific substrate-binding motifs—which contradicted this intended ease of use.

To address this concern thoroughly, we performed a new screening campaign without any PETase-specific motif or catalytic-geometry filtering. Unlike the initial screening, this round relied exclusively on general predicted foldability and text–protein

alignment criteria—namely, pLDDT and ProTrek scores (Section 5 and Extended Data Fig. 8). Through this simplified and generalized pipeline, pinalPETase36 emerged as an active candidate.

Accordingly, in our previous revised manuscript, we updated both the main text (Section 5) and the Supplementary Information (Tables 6, 9, and 13) to focus the PETase validation on this newly validated sequence. We are pleased to note that Reviewer 1 was satisfied with this targeted revision, as it successfully demonstrates Pinal's ability to function without requiring user-provided structural expertise. Further details are provided in our first-round response letter.

For the ADH table, we removed several rows corresponding to **three inactive designed proteins** to make the presentation consistent across the four experimental design cases. In the revised Supplementary Tables, we focus on experimentally validated active proteins, and therefore these inactive candidates are not included in the current table. This revision was made to improve clarity and maintain consistency across the different protein-design cases. If the reviewers would prefer these inactive candidates to be retained in the Supplementary Information, we would be happy to add them back.

3. Regarding the designed sequences for ADH/GFP/PETases/H-protein: To inspect the designed sequences (Tables 8-11 of the previous version), we perform some primary experiments on Pinal's outputs. The inference setting: we use the 15B T2Struct model, sample $K = 1000$ structure sequences, and use greedy decoding (top-1) from SaProt-T. For each text instruction in Table 7, we obtain 1000 designed sequences. We define a sequence-level similarity measure (it evaluates how the two sequences align, we computed sequence identity using MMseqs2 with default settings), and use it to identify similar predictions among 1000 generations for each sequence in Tables 8-11.

3.1 On PETases, we find that it is hard to find similar sequences in 1000 generations (e.g., the most similar generation has a similarity <50%). It may suggest that it is hard to generate those sequences.

We sincerely appreciate the reviewer's insightful and constructive analysis. First, we would like to respectfully clarify that the sequence space of natural and designed proteins is extraordinarily vast, typically exceeding 20^{100} possibilities. Consequently, when drawing a relatively small sample of only 1,000 generated sequences, it is statistically expected that finding candidates with high sequence identity to a specific, pre-selected target is highly challenging. We are grateful for the opportunity to address this point and provide a more detailed explanation below.

To provide a more direct analysis, we randomly sampled 1,000 sequences from our unfiltered PETase candidate pool and performed pairwise sequence-similarity search using MMseqs2. Using a sequence-identity threshold of 0.5, we found that **921 out of the 1,000 candidates** did not have any other sequence in the pool with identity at or higher than 0.5, excluding self-matches. This suggests that, in PETase design, the difficulty of finding highly similar sequences within a 1,000-sequence sample is a general phenomenon among generated candidates, not a specific issue for the experimentally validated PETase sequences. We have provided the corresponding analysis code for the reviewer's reference at https://github.com/westlake-repl/Denovo-Pinal/blob/rbt02/rbt/scripts/analysis/petase_selfsearch.ipynb. We used multinomial sampling in this analysis to remain consistent with the sampling strategy in the manuscript. Nevertheless, we expect that using greedy decoding would not change the main conclusion.

We further examined the effect of sampling depth by repeating the analysis with 50,000 generated candidates. In this larger pool, we were able to recover sequences with over 50% sequence identity to both **pinalPETase36** and pinalPETase01/02/03/07 (reported in the initial manuscript). This result indicates that our reviewer's observation is sensitive to sampling depth, and that a 1,000-sample pool may be insufficient to assess whether Pinal can generate PETase-like candidates similar to the experimentally validated sequences. The scripts for reproducing this analysis have been made available at <https://github.com/westlake-repl/Denovo-Pinal/blob/rbt02/rbt/README.md#analysis-for-question-31-petase>. Generated examples are at <https://github.com/westlake-repl/Denovo-Pinal/tree/rbt02/rbt/outputs/petase>.

3.2 On ADH/H-protein, we can find similar sequences (>80% scores), while their perplexity is significantly higher compared with other sequences in the 1000 generations (i.e., when ranking according to ppl, those sequences are ranked at the bottom 10%). It may suggest that the following filter process could have structured bias.

We thank the reviewer for this insightful analysis. We agree that, for ADH and H-protein, sequences similar to the experimentally tested designs can be ranked in the bottom portion, e.g., the bottom 10%, when candidates are ordered by perplexity. We believe that this observation is **reasonable** and reflects the distinction between perplexity-based ranking and the downstream prioritization criteria used in our design workflow.

To directly examine this issue, we analyzed the relationships among perplexity, ProTrek score, and ESMFold pLDDT for the ADH and H-protein candidate pools. For each task, we generated 50,000 candidates and computed their perplexity, ProTrek score, and ESMFold pLDDT. The corresponding analysis code has been provided at https://github.com/westlake-repl/Denovo-Pinal/blob/rbt02/rbt/scripts/analysis/ppl_analysis.ipynb.

As shown in **Fig. R1**, candidates with lower perplexity **generally tended to** have higher ProTrek scores and higher pLDDT values, indicating **broad consistency** among the three metrics. However, the lower panels of Fig. R1 show that the PPL distribution is **highly concentrated** for both ADH and H-protein: most generated candidates fall into **the first, lowest-PPL interval**, and more than 97% of candidates have very low PPL values, i.e., PPL < 3. Therefore, the “bottom 10%” ranking by PPL should be interpreted as a relative ranking within an already very low-PPL candidate pool, rather than as evidence that these sequences have high absolute PPL. In other words, a candidate falling into the second low-PPL interval can already be ranked in the bottom 10% by PPL, **simply because the majority of candidates are concentrated in the first interval**. Thus, even if such candidates are ranked in the bottom portion by relative PPL, they can still be high-quality generations in terms of their absolute PPL, ProTrek score, and pLDDT values (We also examined the **five active ADH designs**, which were ranked approximately in the lower **40%** by PPL, and have provided their corresponding protein sequences and 3Di token sequences for the reviewer’s reference in Supplementary Table 10.).

The upper panels of Fig. R1 further show that, despite this overall trend, the candidates with the highest ProTrek scores or ESMFold pLDDT values **do not necessarily** lie in the first, lowest-PPL interval (see the red vertical dashed line in Fig.R1). Instead, they can appear in the second low-PPL interval, where the absolute PPL values remain low but the relative PPL rank may already fall into the bottom 10%. This explains the reviewer’s observation: in our design workflow, ADH and H-protein candidates were not selected by PPL, but were instead prioritized using pLDDT and ProTrek score. Therefore, candidates selected by these downstream criteria are not necessarily expected to rank highly by PPL, even though their absolute PPL values remain low.

We further clarify that perplexity, ProTrek score, and pLDDT evaluate different aspects of a candidate sequence. Perplexity reflects model likelihood, ProTrek score evaluates text–protein alignment, and pLDDT estimates predicted structural confidence. In our workflow, metrics like ProTrek score, pLDDT, and PAE were used as coarse-grained feasibility and prioritization criteria, rather than as oracles for experimental activity. We expect stronger text–protein alignment and higher predicted structural confidence to enrich functional candidates at the population level, but we do not assume that small differences in these scores define a strict ordering of experimental success. Indeed, generated proteins with relatively lower ProTrek scores or pLDDT values may still be functional, although the limited throughput of wet-lab validation prevents us from systematically testing this possibility at large scale.

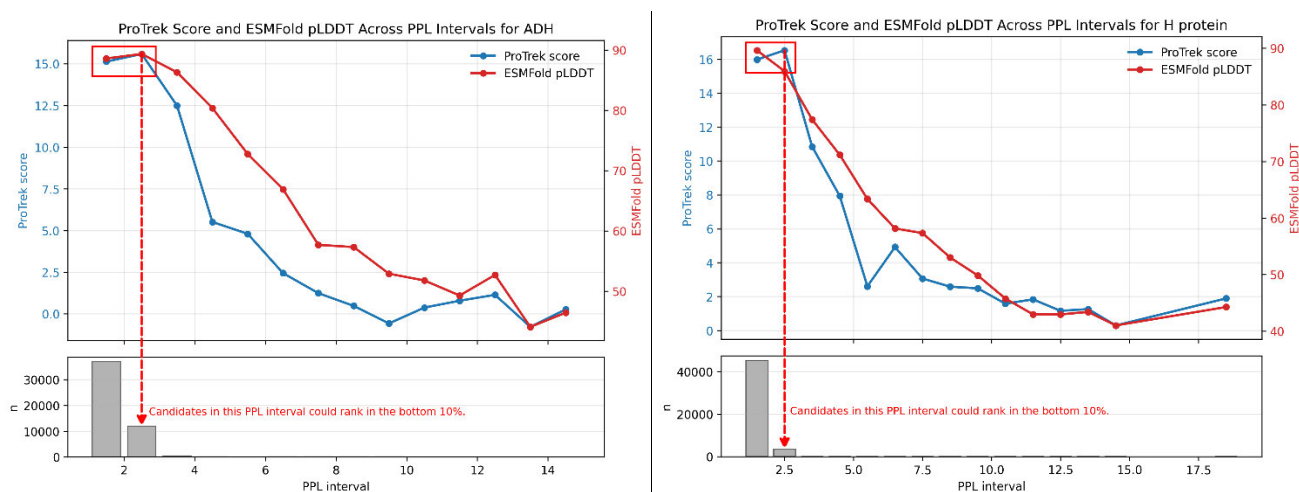


Fig. R1: Relationship between model perplexity and downstream ranking measures for ADH and H-protein candidate pools. For each task, 50,000 candidates were generated. The upper panels show the mean ProTrek score and mean ESMFold pLDDT within each PPL interval. The lower panels show **the number** of generated candidates in each PPL interval. Most candidates are concentrated in the first, lowest-PPL interval. Therefore, candidates in the second low-PPL interval **can already fall into the bottom 10% when ranked by PPL**, even though their absolute PPL values remain low. **Red boxes** highlight low-PPL regions where high ProTrek scores or high pLDDT values can occur outside the first PPL interval.

Finally, the apparent difference between PETase and ADH/H-protein may also be explained by differences in the training-data distribution across protein functions. PETase and related PET-hydrolyzing enzymes are likely underrepresented in curated databases such as Swiss-Prot, which likely leads to a sparser and more diverse PETase generation space. Consistently, PETase candidates in Fig. 4 show a broad range of ProTrek scores, indicating variable text–protein alignment with the PETase description. Therefore, for PETase, 1,000 samples may not be sufficient to recover sequences highly similar to the experimentally validated candidates. In contrast, ADH and H-protein may correspond to better-represented or more compact regions of the model’s learned distribution, making similar sequences easier to identify within 1,000 samples. As noted above, increasing the PETase sampling depth to 50,000 allowed us to recover sequences more similar to the experimentally validated PETase designs, supporting the idea that the 1,000-sample analysis is sensitive to sampling depth.

3.3 Since our inference setting slightly differs from the paper, would you please conduct (and open source) some similar experiments and discuss the results?

We thank the reviewer for this helpful suggestion. We have now made the generation scripts for all four experimentally tested design cases publicly available on GitHub, together with the exact inference hyperparameters used in the manuscript (Supplementary Table 13): <https://github.com/westlake-repl/Denovo-Pinal/tree/main/generation>.

In addition, we have provided the scripts, instructions and analysis for the PETase, ADH and H-protein discussed in **Question 3.1** and **3.2**.

4. During our experiment, we found that some details could be made clearer.

We thank the reviewer for pointing out this important issue, which has helped us improve the clarity and reproducibility of the manuscript. We agree that the implementation details would benefit from further elaboration and refinement. In response, we have revised and expanded the descriptions of the wet-lab validation experiments and task-specific workflows in both the manuscript and Supplementary Information to more fully and accurately reflect the implemented procedures. We have also released the corresponding inference code (see Q 3.3) and revised Section 5, Methods Section 7.6, and Supplementary Table

13 to include task-specific details on initial candidate generation, sampling settings, and screening procedures, thereby making the workflow more transparent and reproducible (as detailed below).

4.1 Regarding the initial candidates, for GFP, Pinal generates 50,000 initial samples (line 919, revised manuscript), but it is not clear whether the other three types get the same number.

We have added relevant descriptions to provide clearer task-specific details on the initial candidate-pool sizes. The initial candidate-pool sizes for all four experimental design tasks are now listed in Supplementary Table 13 and in the revised manuscript.

Table 13: Inference hyperparameters used to generate candidates for wet-lab validation. Initial N denote the number of generated candidates before filtering.

Task	T2Struct		SaProt-T		Initial N
	Sampling method	Temperature	Sampling method	Temperature	
GFP	multinomial	1.0	multinomial	0.1	50,000
GFP-g	multinomial	1.0	multinomial	0.3	50,000
PETase	multinomial	1.2	multinomial	0.3	200,000
ADH	multinomial	0.8	greedy	—	2,500
H protein	multinomial	1.0	multinomial	0.3	50,000

To be specific, the initial candidate-pool sizes were not identical across the four validation tasks. For the GFP and H-protein design tasks, Pinal generated 50,000 initial candidates for each task (Lines 264, 316). For the PETase validation experiment that led to the identification of pinalPETase36, we generated a larger initial pool of 200,000 candidates (Line 281). These sampling scales are within the range used in recent large-scale protein-design workflows (see [1,2]) and were chosen to provide sufficiently large candidate pools for downstream computational filtering and experimental prioritization. For ADH, which was the earliest wet-lab validation task in this study, we used a smaller initial pool. Pinal generated 2,500 ADH candidates, of which 1,639 non-redundant sequences passed the basic feasibility filters based on ESMFold pLDDT and PAE. These candidates were then provided to our wet-lab collaborators, who retained 1,000 sequences for further consideration (Methods 7.13.1).

The differences in initial pool size and screening procedure mainly reflected task-specific experimental designs, validation workflows, and available wet-lab throughput. The task-specific filtering thresholds and ranking criteria based on pLDDT, PAE, and ProTrek score were used for empirical candidate prioritization under limited experimental throughput, rather than as definitive boundaries for protein activity or functionality. Thus, the active proteins reported in the manuscript should be viewed as experimentally validated examples from the prioritized candidate pools, rather than an exhaustive survey of all functional candidates generated by Pinal.

This interpretation is supported by a follow-up H-protein design experiment performed by our collaborators using Pinal. In this follow-up experiment, similar empirical filtering strategies (AlphaFold3 PAE < 7, pLDDT > 80, and ProTrek score > 15) were applied to an initial pool of 6,250 Pinal-designed H-protein candidates. Of the 98 selected designs, 86 showed soluble expression and 49 exhibited measurable activity. Moreover, more than 20 candidates showed higher glycine-production activity than the native E. coli H-protein (ecH). These additional data, now included in the revised Supplementary Information (Tables 14 and 15, Fig. 7, and Method 4), further support the robustness and potential practical utility of the Pinal-based design workflow.

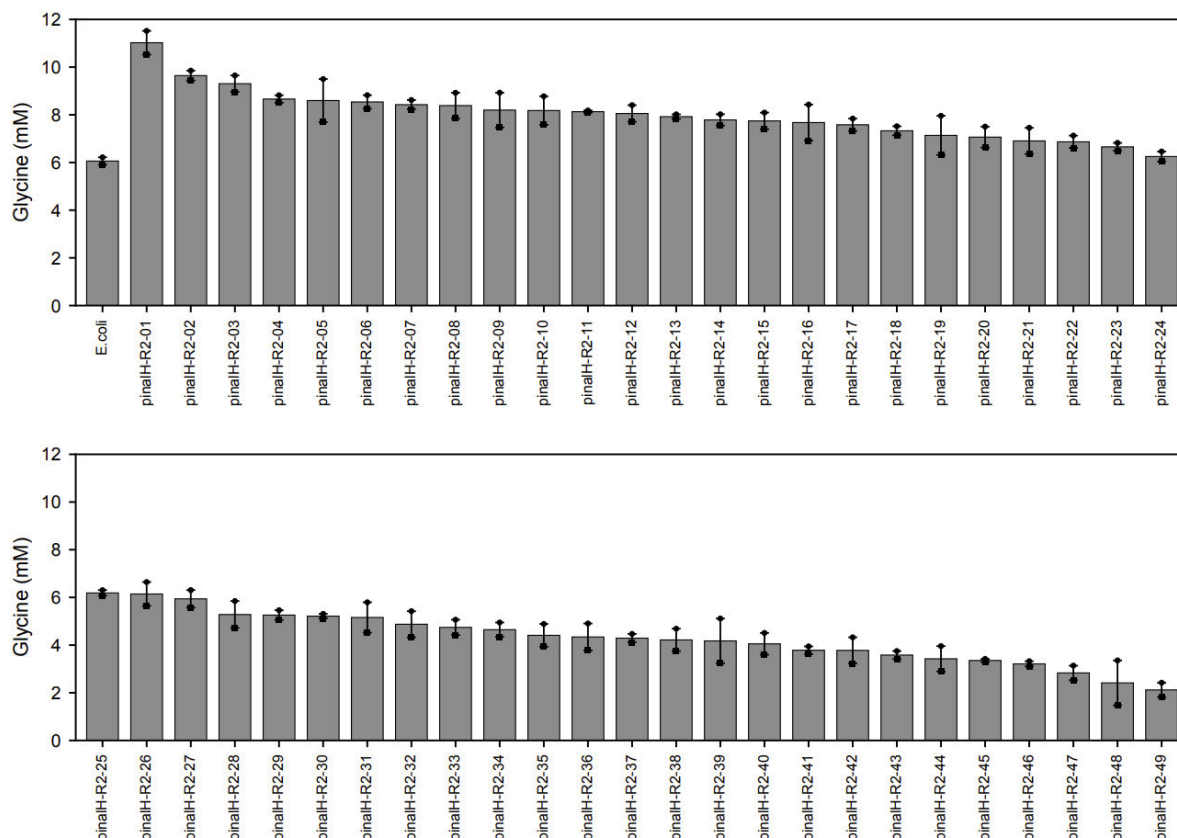


Fig. 7: Glycine production using Pinal-generated H proteins and the native *E. coli* H protein (ecH) in the second validation round.

- [1] Woody Ahern. et al. “Atom-level enzyme active site scaffolding using RFdiffusion2” Nature Methods (2025).
 [2] John B. Ingraham. et al. “Illuminating protein space with a programmable generative model”. Nature (2023)

4.2 It would also be helpful to explicitly describe the process of generating the initial candidates. We first guess that first sampling $K = 50$ structures, then sample 1000 sequences from SaProt-T, but after checking the code, SaProt-T only supports greedy decoding (missing code for "multinomial sampling" in line 757).

We thank the reviewer for pointing out this important issue. We have described the initial candidate-generation setup in the revised manuscript (**Methods Section 7.6 and Supplementary Table 13**).

The $K=50$ setting was used for our computational evaluation (Section 7.6, 7.7). Since this evaluation covered a large number of protein-description prompts, we sampled $K=50$ structure-token sequences for each text prompt to balance computational cost and design accuracy, as analyzed in Extended Data Fig. 3. For each generated structure-token sequence, SaProt-T was then used to decode **one** amino-acid sequence. The resulting candidates were ranked according to their joint structure and sequence log-probabilities, normalized by sequence length, and the top five candidates were retained for computational evaluation, as described in the manuscript (section 7.7.3). This ranking mechanism was based on the log-likelihood scores produced by Pinal itself.

For the wet-lab validation experiments, we followed the same overall two-stage design principle, but used a **substantially larger sampling scale** because each design task focused on a single target protein. Specifically, following common practice in protein design, we generated tens of thousands of structure-token sequences for each design prompt, such as 50,000 for GFP and H-protein, 2,500 for ADH, and 200,000 for PETase. Each structure-token sequence (along with the text prompt) was then decoded by SaProt-T into one amino-acid sequence. Unlike the computational evaluation, where candidates were ranked only using Pinal’s own log-likelihood scores, candidates for wet-lab validation were directly filtered and prioritized using

downstream in silico evaluation metrics, including the ProTrek score, AlphaFold/ESMFold pLDDT, and PAE, etc. These structural and functional assessment metrics were used as pre-screening criteria to enrich for candidates with a higher likelihood of successful experimental validation before final experimental testing.

We thank the reviewer for pointing out that the released SaProt-T code did not include the multinomial sampling option. This option was inadvertently omitted during code release. We have now added it to the code, as shown below, and updated the documentation at <https://github.com/westlake-repl/Denovo-Pinal#predicting-amino-acid-sequence-with-saprot-t>.

Predicting amino acid sequence with SaProt-T

Here, we provide a script for predicting amino acid sequences using natural language, enabling you to specify the desired structure. We support two decoding strategies: greedy decoding and temperature-controlled multinomial sampling.

```
from utils.design_utils import SaProtPrepareGenerationInputs, SaProtGeneration, load_SaProtT_and_tokenizers
desc = "Actin."
saprot, saprot_text_tokenizer, saprot_tokenizer = load_SaProtT_and_tokenizers()
structure = "dqdpafakewedfqfwifidtfpdqggqdfgqkkwafpdpppcvpdpddridgtvrrvvvvgtdmdgqdalqagdpvps
SaProtInputDict = SaProtPrepareGenerationInputs([" ".join(list(structure))], desc, saprot_text
seq = SaProtGeneration(saprot, SaProtInputDict, saprot_tokenizer, saprot_sample_type="multinomi
# For greedy decoding, use:
# seq = SaProtGeneration(saprot, SaProtInputDict, saprot_tokenizer, saprot_sample_type="argmax"
```

4.3 Compared the two versions of Tables 8-11, PETases36 is completely new, and Table 9 (ADH) removes some rows. The update may need some explanation.

This question was answered at the bottom of page 5 of this letter.

4.3 Would you please provide the structure tokens for the sequences in Tables 8-11?

We thank the reviewer for this suggestion. We have provided the corresponding structure token sequences for all designed sequences listed in Supplementary Table 9-12 and Table 15.

Summary:

We sincerely thank the reviewer for the constructive and in-depth analysis of our experimental validation procedures, implementation details, and data presentation. These comments have been highly valuable and have guided key revisions to the manuscript and Supplementary Information. Following the reviewer's suggestions, we have released the corresponding inference code and provided additional scripts, computational analyses, and task-specific descriptions on the protein-generation and screening workflows. We believe that these revisions substantially improve the readability, reproducibility, and interpretability of the study. We would also be very happy to address any further questions or concerns the reviewer may have.

Point-by-point Response Letter

Reviewer #2 (Remarks to the Author):

1. To characterize the designed sequences, Fig. R1 is informative: ADH/H-protein samples with low PPL rankings can still have reasonable foldability and alignment scores (they are on the learned manifold). It would be more informative to indicate the position of designed sequences (for the four proteins) in the figure.

We thank the reviewer for this helpful suggestion. Following the reviewer's recommendation, we have indicated the positions of the experimentally validated designs for the four protein targets within their corresponding PPL distributions.

As shown in Fig. R1, the validated ADH and H-protein designs are located in relatively low-PPL regions, while still exhibiting favorable ProTrek scores and predicted structural confidence, suggesting that these sequences remain on the learned protein manifold. More broadly, Fig. R1 illustrates that ProTrek score, predicted pLDDT, and PPL provide complementary but partially correlated measures for evaluating designed sequences. Among the four designed targets, ADH, H-protein, and GFP show higher overall generation quality than PETase in terms of ProTrek score and PPL, potentially reflecting the greater design difficulty of PETase. In the revised manuscript, we included this analysis in Supplementary Fig. 8 and Supplementary Note 2.

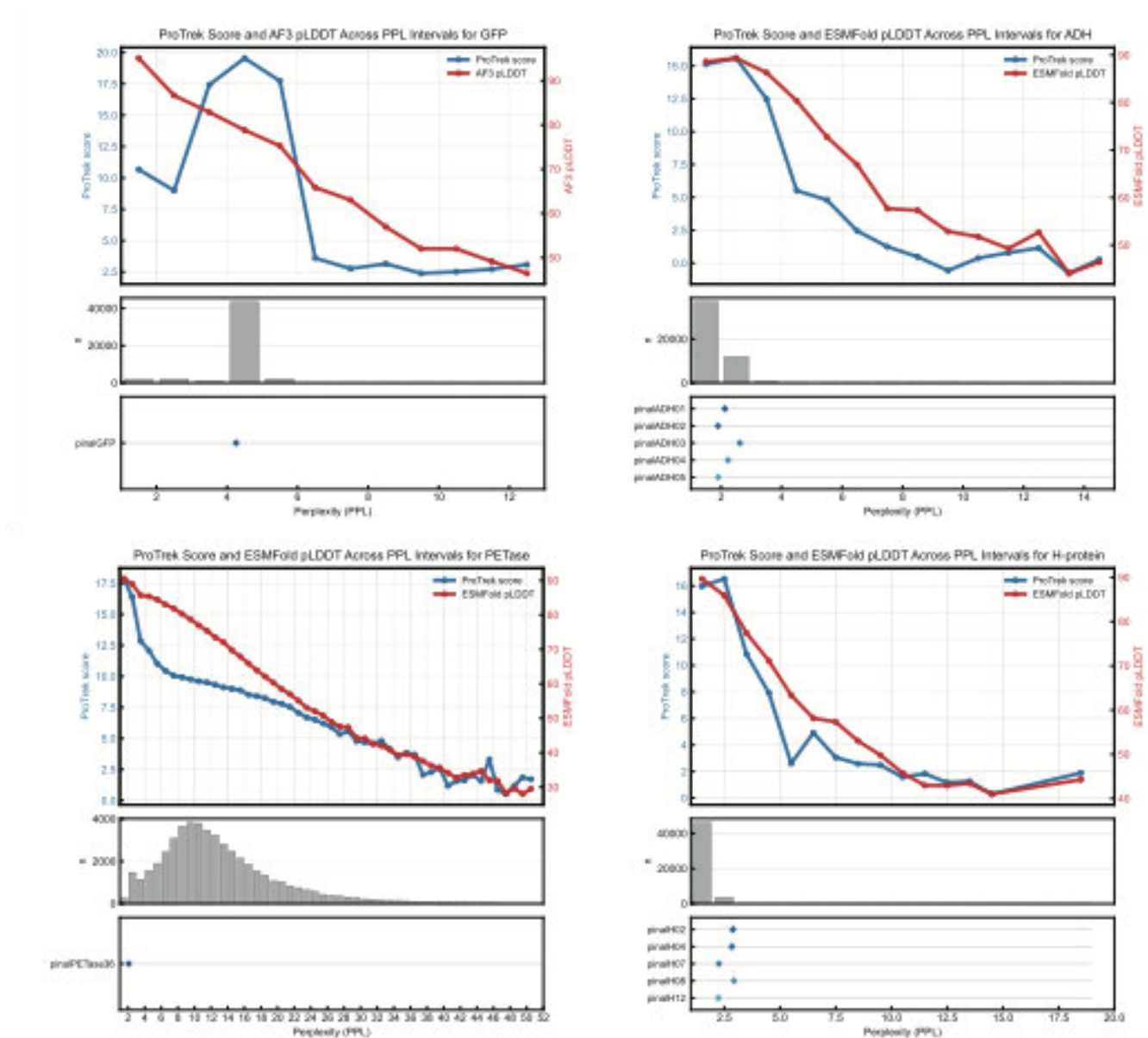


Fig. R1: Relationship between model perplexity and downstream ranking measures for designed protein candidate pools. For each target, the upper panel shows the ProTrek score and predicted structural confidence across PPL intervals. The middle panel shows the distribution of designed sequences across PPL intervals. Diamonds in the lower panel indicate the PPL positions of experimentally validated designs for the corresponding target. For comparability across tasks, each reference distribution was constructed from 50,000 candidate sequences.

- Regarding PETase generations, the point is that if a model can generate a sequence, it may be easy to generate some similar sequences (i.e., those sequences in the same local mode). Although enlarging the sample pool can help, it seems that generating similar PETase sequences is harder than for the other three proteins: it requires a larger sample pool to get moderately similar sequences. I would suggest 1) conducting more careful analyses for PETase generations, and 2) elaborating the difficulty explicitly in the paper.

We thank the reviewer for this observation. We agree that, for the PETase task, recovering sequences similar to a specific design, such as pinalPETase36, requires a larger candidate pool than for the other three targets. To better understand this difference, we further examined the distribution of generated candidates in sequence space under the same inference settings. When equally sized pools of 50,000 candidates were clustered at 30% sequence identity, the ADH pool formed 743 clusters, whereas the PETase pool formed 20,196 clusters. This indicates that PETase candidates are distributed across many more sequence neighbourhoods, with fewer sequences per neighbourhood on average. Therefore, independently sampled PETase candidates are less likely to fall into the same local region as a particular validated design, such as pinalPETase36, explaining why a larger candidate pool is needed to recover moderately similar variants.

A likely contributing factor is the limited representation of PETase/PET-hydrolase-related functions in curated protein databases. For example, only a small number of entries can be retrieved from Swiss-Prot using PETase- or PET hydrolase-related keywords. Such limited coverage may lead to a more diffuse conditional generation space for the PETase task, consistent with the broader PPL distribution in Fig. R1 and the wider range of ProTrek scores observed for PETase candidates in Fig. 4. We have added this discussion to Supplementary Note 2 of the manuscript and elaborate the difficulty of recovering sequences close to a specific validated PETase design.

- Regarding the intermediate structure token prediction (Fig. 2a), it is potentially helpful to include structural features in the conditional generation pipeline, but it is also important to implement the baseline (direct aa token prediction) correctly. Depending on the context, "the same model setting" may or may not lead a fair comparison. I would suggest 1) trying some alternative configurations (e.g., different hyperparameters) for a stronger next aa prediction baseline, and 2) discussing some follow-up works on direct aa sequence generation for an up-to-date literature review.

We thank the reviewer for this suggestion. To evaluate the robustness of the comparison, we repeated it under two alternative training configurations. Specifically, both the structure-token and amino-acid models were trained using a higher learning rate of 5×10^{-4} and a lower learning rate of 5×10^{-5} , relative to the original setting of 1×10^{-4} . All other settings remained unchanged, including the model architecture, SwissProt-Annot dataset, 30% sequence-identity split and training steps.

As shown in Fig. R2, the two additional learning-rate settings produced learning curves and ESMFold pLDDT trends that were highly consistent with the original results. Across all tested settings, the direct text-to-amino-acid model achieved substantially lower validation accuracy and generated sequences with lower predicted structural confidence than the text-to-structure-token model. Although hyperparameter choices can affect model performance, they usually did not qualitatively change the conclusion. In particular, the direct amino-acid prediction baseline remained much closer to the random amino-acid baseline than to the text-to-structure-token model, suggesting that the observed performance gap is unlikely to be resolved by hyperparameter adjustment alone. These results support the robustness of our conclusion that intermediate structure-token prediction is more readily learned than direct amino-acid prediction under the controlled settings tested here.

We have added this discussion to the Methods section 7.2 describing the text-to-sequence versus text-to-structure generation experiments, and expanded the discussion to include ProLLaMA, ProtDAT and ProDVa, which demonstrate recent progress in annotation-guided protein design using large-scale sequence pretraining, specialized cross-attention architectures or dynamic protein vocabularies. We further note that these studies were mainly evaluated in computational benchmark settings, leaving experimental validation for future work.

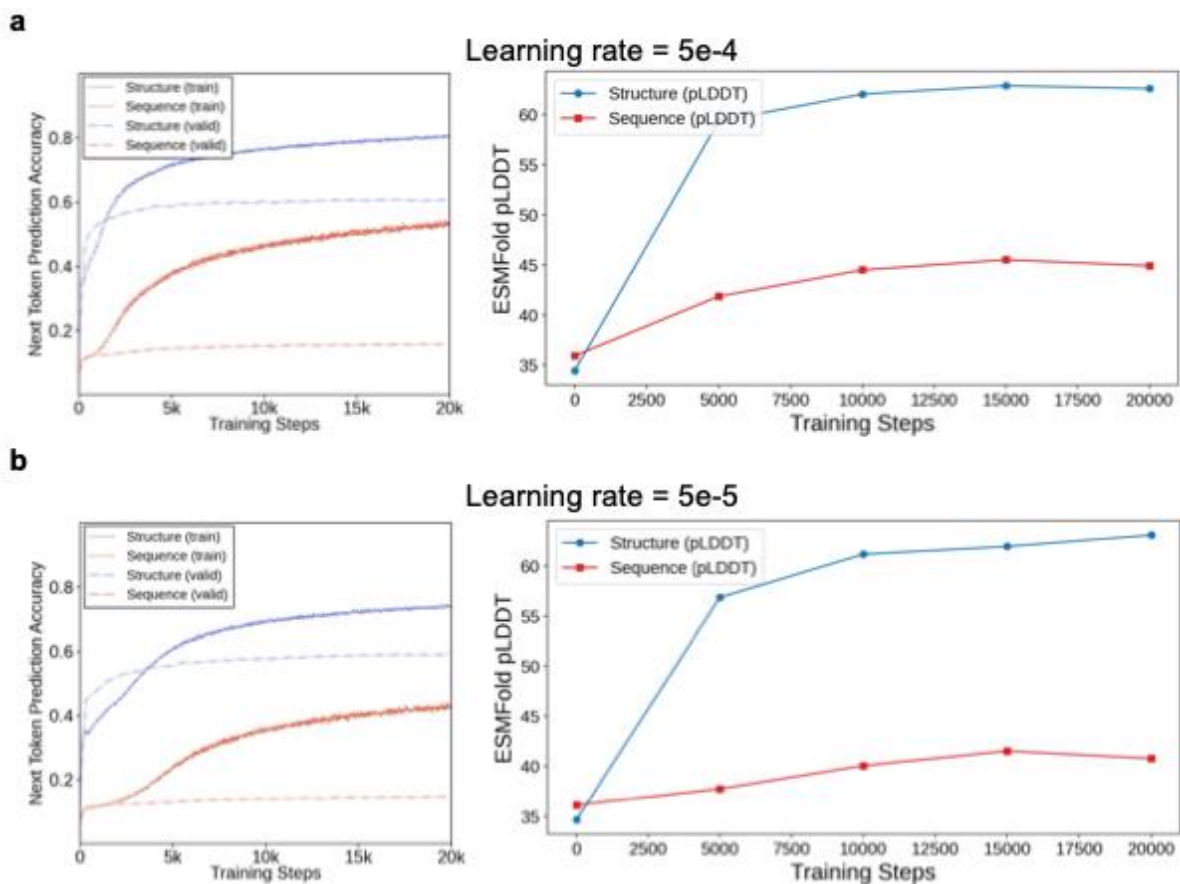


Fig. R2: Preliminary representation comparison under two alternative training configurations. Training and validation next-token prediction accuracies are shown on the left, with ESMFold pLDDT values of generated sequences on the right. Models were trained using learning rates of 5×10^{-4} and 5×10^{-5} in a and b, respectively. All other model, dataset and optimization settings were unchanged.