ROSETTASEARCH: MULTI OBJECTIVE LANGUAGE MODEL DRIVEN SEARCH FOR PROTEIN SEQUENCE DESIGN

Anonymous authors
Paper under double-blind review

ABSTRACT

Sequences produced by state-of-the-art backbone-conditioned design models such as LigandMPNN often fail to fold into the target structure when checked with an independent structure predictor. We introduce RosettaSearch, an inference-time, multi-objective search that repairs such sequences without any model retraining. RosettaSearch uses an LLM as a generative optimizer inside a priority-based parallel search. At each step, the LLM proposes targeted edits after reading structured feedback from RosettaFold3: global rewards (pLDDT, TM-score, Cα-RMSD), residue-level annotations of problem regions, and text constraints that discourage reward hacking. On 400 LigandMPNN designs for native PDB monomers, RosettaSearch improves fidelity metrics by 18% to 68% and raises the design success rate 2.5$\times$ (7.9% to 20.5%). The gains persist under two independent predictors not used in the search (Chai-1 and Boltz-2), and a random-mutation baseline with identical feedback and oracle budget shows no improvement. On 275 *de novo* Dayhoff backbones, RosettaSearch raises the success rate of ProteinMPNN designs from 72% to 90%, and it rescues sequences for difficult to design backbones on which ProteinMPNN fails. RosettaSearch can also optimize a physics-based Rosetta energy term, and a small policy fine-tuned with a MultiMax reinforcement-learning reward improves the search's success rate. In a preliminary wet-lab test, we find that a designed sequence for a *de novo* backbone shows a single cooperative thermal unfolding transition at 89 °C. To our knowledge, this is the first large-scale demonstration that LLMs can serve as effective generative optimizers for backbone-conditioned protein sequence design.

1 INTRODUCTION

Protein sequence design aims to find amino acid sequences that fold into a target three-dimensional structure, and it underpins drug discovery and enzyme engineering. We focus on *backbone-conditioned sequence design*: given a fixed backbone, find a sequence that confidently and accurately folds into it. We call this property *fidelity*, i.e., whether the structure predicted independently for the designed sequence recapitulates the target backbone with high confidence and geometric accuracy. Despite the progress of LigandMPNN (Dauparas et al., 2025) and ProteinMPNN (Dauparas et al., 2022), sequences from these models frequently exhibit low fidelity (Anishchenko et al., 2021). This is a key bottleneck between computational design and experimental success, since a sequence that cannot be predicted to adopt the target structure is unlikely to do so in the wet lab.

The problem is a high-dimensional combinatorial optimization: the sequence space is astronomically large, each evaluation needs a full structure-prediction call, and the sequence–fold relationship is highly non-linear. Single-pass design models cannot revise their output in response to downstream structural feedback. Optimization-based alternatives (reinforcement learning, directed evolution, preference fine-tuning) allow iterative improvement but need task-specific training, which is expensive and inflexible to new objectives. LLMs have been tried for protein design only in one-shot settings, where they give sequences of much lower fidelity than task-specific models (Xia et al., 2025). Generative optimization with LLMs has shown promise for prompts, code and hyperparameters (Khattab et al., 2023; Agrawal et al., 2025), but its use for backbone-conditioned design has not been systematically explored.

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

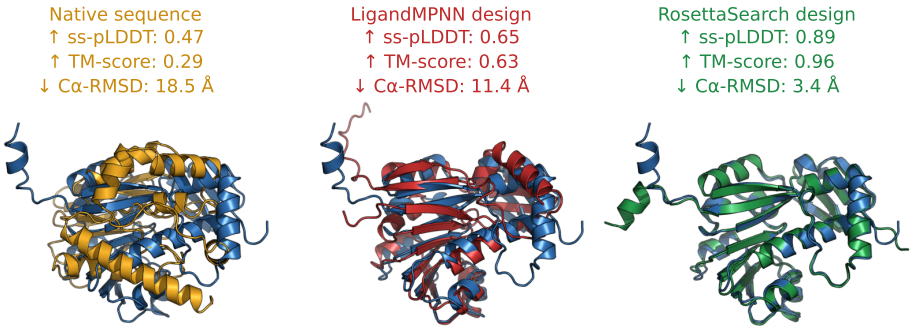


Figure 1: **Inference-time optimization with RosettaSearch dramatically improves structural fidelity.** For PDB: `7z05`, RosettaSearch transforms a low-fidelity LigandMPNN design into a near-native structure without model retraining. All pLDDT values are single-sequence (no MSA). *(Blue)* Native PDB structure. *(Golden)* Predicted structure of native sequence. *(Red)* Best LigandMPNN design. *(Green)* Best RosettaSearch design. Arrows indicate desired direction of improvement. See Table 10 and Figures 9, 10 for additional examples.

We introduce **RosettaSearch** (Figure 7), an inference-time multi-objective optimization approach that deploys LLMs as generative optimizers guided by structural rewards. RosettaSearch *refines* an existing candidate by reasoning over structured feedback from RosettaFold3 (Corley et al., 2025), combining global scalar rewards with local residue-level annotations, and proposes targeted edits within a *priority search* that explores multiple trajectories in parallel. Because rewards and feedback are defined at inference time, no retraining is needed for new objectives. We also extend the framework to vision-language models (VLMs) that receive rendered structure images (Appendix). To our knowledge, this is the first large-scale demonstration that LLMs can serve as effective generative optimizers for backbone-conditioned design, yielding systematic gains over popular baselines without fine-tuning.

2 Related Work

Backbone-conditioned sequence design. ProteinMPNN (Dauparas et al., 2022) and LigandMPNN (Dauparas et al., 2025) learn a direct mapping from backbone coordinates to sequences and are widely used. They decode in a single autoregressive pass, are optimized for sequence recovery rather than fidelity under independent evaluation, and cannot detect or correct failures at inference time. We treat their output as a starting point for feedback-driven refinement.

Protein sequence optimization. Energy-based methods (Stracquadanio & Nicosia, 2011; Zhou et al., 2024), reinforcement learning (Lutz et al., 2023), directed evolution (Yang et al., 2019), and probabilistic and diffusion-based models (Riesselman et al., 2018; Tang et al., 2025) improve sequences toward target properties but need task-specific training when objectives change. ResiDPO (Xue et al., 2025) fine-tunes design models with AlphaFold-derived preferences, coupling optimization to a fixed training procedure. RosettaSearch needs no fine-tuning.

LLMs for protein design. Protein language models such as ProtGPT2 (Ferruz et al., 2022) generate plausible sequences but are hard to control for backbone-conditioned design, and frontier LLMs in one-shot settings have much lower fidelity than task-specific models (Xia et al., 2025). The Virtual Lab (Swanson et al., 2024) and AI co-scientist (Gottweis et al., 2025) orchestrate trained modules with multi-agent workflows rather than performing inference-time optimization. We use the LLM as a generative optimizer that receives quantitative structural feedback within a principled search.

Generative optimization with LLMs. LLMs have been used as optimizers over discrete spaces with textual feedback as a surrogate for gradients (Khattab et al., 2023; Cheng et al., 2024; Yuksekgonul et al., 2024; Nie et al., 2024), combined with beam search (Sun et al., 2023; Pryzant et al., 2023; Chen et al., 2024), MCTS (Wang et al., 2023), Gibbs sampling (Xu et al., 2023), FunSearch (Romera-Paredes et al., 2024) and GEPA (Agrawal et al., 2025), for prompts, code and scientific discovery (Wei et al., 2024; Ghareeb et al., 2025; Narayanan et al., 2025). We extend the Trace framework (Cheng

et al., 2024) to protein design, adding priority-based search, multi-objective rewards and residue-level feedback.

3 DESIGN OF ROSETTASEARCH

RosettaSearch adopts generative optimization (Khattab et al., 2023; Cheng et al., 2024), in which LLMs propose parameter changes from rich feedback. We write $\hat{\mathbf{x}} = f(s)$ for the structure predicted for sequence s by model f, and $\mathbf{x}^*$ for the reference backbone coordinates. We use *reference (native) structure* and *backbone* interchangeably.

3.1 FIDELITY METRICS

We measure how well a designed sequence adheres to the target structure with three metrics. **pLDDT** (Tunyasuvunakool et al., 2021) is a per-residue confidence in $[0, 100]$ output by predictors such as RosettaFold3 (Corley et al., 2025); it is widely used as a proxy for how likely a sequence is to adopt the predicted fold (Yin et al., 2022). **Cα-RMSD** is the root mean square deviation between corresponding backbone α-carbons of $\hat{\mathbf{x}}$ and $\mathbf{x}^*$ (equal lengths required; Appendix A.1). **TM-score** measures global similarity between $\hat{\mathbf{x}}$ and $\mathbf{x}^*$; unlike RMSD it is length-normalized and less sensitive to local deviations.

3.2 REWARD AND FEEDBACK DESIGN

The feedback contains information about a design's global and local quality and constraint violations, and is configurable for each run (example in Figure 14).

Global reward and feedback. We predict the structure of a proposed sequence with RosettaFold3 (RF3), keeping the best of its multiple predictions (highest average pLDDT). Each metric is averaged over residues and scaled to $[0, 1]$; Cα-RMSD values beyond a penalty threshold incur a negative correction. A textual description of the value (e.g., "0.3 is a very low reward. Significant improvements needed.") accompanies each reward.

Local text feedback. Global information alone did not give substantial improvements in preliminary experiments, so we add *local* feedback that directs the LLM to problematic regions. Low-confidence regions are found with adaptive pLDDT thresholds and reported as contiguous residue ranges; per-residue distances between $\hat{\mathbf{x}}$ and $\mathbf{x}^*$ are binned and reported as residue intervals per bin; TM-score feedback is built analogously. This encourages localized refinement rather than global rewrites.

Image feedback. With a VLM as optimizer, we also provide a rendered image of the predicted structure, which adds global spatial context (secondary structure, topology, buried versus exposed regions). Results are in Appendix.

3.3 FEEDBACK TO PREVENT REWARD HACKING

Under optimization pressure, proxy rewards such as pLDDT decouple from design quality. The LLM exploits this by repeating high-pLDDT motifs, producing trivially short or overly long sequences, appending poly-alanine runs, or (when the native sequence is in context) returning the native sequence. These behaviors resemble classic reward hacking (Skalse et al., 2022). Inspired by constrained optimization, we add text feedback that acts as soft constraints on the LLM's proposals without changing the reward: (i) *repetition*, flagged when repeated 6-mers exceed 5% of all 6-mers; (ii) *length*, flagged when the sequence is longer or shorter than the desired length n; and (iii) *ligand binding*, where we give residue-level annotations of ligand-binding positions to preserve function.

3.4 GENERATIVE OPTIMIZATION WITH LLMS

We consider two ways to deploy generative optimizers.

Sequential Revision iteratively calls the LLM to improve the latest candidate. We adapt the Trace optimizer (Cheng et al., 2024), treating the sequence as a trainable text variable. At each iteration the

optimizer builds a structured problem instance (variables, constraints, outcomes, scalar and textual feedback), reasons about how to change the variable, and proposes targeted updates.

Priority Search. Sequential optimization is prone to early convergence and instability, so we design a priority-based parallel search (Algorithm 1). A buffer stores all previous candidates and their evaluations, initialized with the starting sequence. In each iteration we take the top-K candidates from the buffer, evaluate them in parallel to obtain reward and feedback, independently generate one new candidate from each, evaluate these, and add all $2K$ candidates back to the buffer.

Multi-objective optimization. Protein design must balance structural confidence, geometric fidelity and sequence validity. We adopt a weighted-sum reward $R(s) = w_{tm}\,\mathrm{TM}(\mathbf{x}, \mathbf{x}^*) + w_p\,\mathrm{pLDDT}(\hat{\mathbf{x}}) + w_r\,C\alpha\text{-RMSD}(\mathbf{x}, \mathbf{x}^*)$ (with RMSD entering as a normalized score to be maximized). It gave the most consistent behavior among the variants we tried (Appendix A.2).

Step-size and best-of-N. To control exploration, we state a textual step-size limiting residue edits per step (unconstrained, conservative: 1–2, moderate: 5, aggressive: 10, drastic: 20), analogous to step-size selection in gradient-based optimization. We run one Priority Search per step-size regime (unconstrained, moderate and aggressive, which gave the best exploration/exploitation balance) with 25 RF3 evaluations each, giving $N = 75$ candidates per protein, and return the highest-reward one. This query-limited best-of-N strategy is closer to budgeted black-box optimization than to RL with thousands of gradient updates.

Scalar vs. vector search In scalar (weighted-sum) search, the objectives (pLDDT, TM-score, Cα-RMSD) are combined into a single reward with fixed weights, and the search ranks candidates by that number. In vector search, the objectives stay separate: candidates are ranked by Pareto dominance, and the LLM sees all objectives at every step, so it can trade them off without a hand-chosen weighting. We found that the two behave similarly with vector search's advantage being not needing to set the weights.

Hyperparameters. All experiments use `o4-mini`. The weights $w_p = \frac{1}{11}$, $w_{tm} = w_r = \frac{5}{11}$ were selected on 20 held-out proteins (Appendix A.3): equal weights give conflicting signals, and pLDDT is easily inflated by edits that degrade TM-score and Cα-RMSD.

3.5 Reinforced fine-tuning (MultiMax)

We also fine-tune a sequence-editing policy π_θ, which proposes a sequence q from a parent p, with GRPO (Shao et al., 2024). With normalized scores r_1, r_2, r_3 (to be maximized) and $\Delta_i = r_i(q) - r_i(p)$, the reward is $u = \max_i \Delta_i$: the largest single-objective improvement, which lets the preferred objective vary with the parent. Since $\max_q \max_i \Delta_i = \max_i \max_q \Delta_i$, an optimal proposal under exact optimization is the best of three objective-specific optimizers. A positive MultiMax reward guarantees the proposal is not Pareto-dominated by its parent, making the policy well suited as a proposal operator in Pareto-style local search (Tricoire, 2012), and it is much simpler than other multi-objective RL methods (Roy et al., 2023; Chakraborty et al., 2024; Park et al., 2024; Xiong & Singh, 2025).

4 Datasets

PDB monomers (LigandMPNN dataset). From PDB-D (Xue et al., 2025) we sampled ≈400 monomeric structures. For each we use the native sequence and sequences redesigned with LigandMPNN (Dauparas et al., 2025), which lets us test whether the LLM improves fidelity while preserving the structure.

Dayhoff. The Dayhoff BackboneRef dataset (Yang et al., 2025) has *de novo* backbones generated with RFDiffusion (Watson et al., 2023) and sequences designed with ProteinMPNN. These start with high fidelity, giving a stringent test on well-folded designs (details in Appendix A.4).

Remark. We report single-sequence pLDDT (no MSA), because MSA-derived homology signals can inflate confidence for poorly foldable sequences (Korbeld et al., 2025).

Table 1: Sequential revision vs Priority search, absolute and % improvement (or reduction). For absolute improvement, the average value of the metric before and after are shown. See Appendix Table 11 for statistical significance.

	Single objective (pLDDT)			*Multi objective* (all metrics)		
	ss-pLDDT ↑	TM-score ↑	$C\alpha$-RMSD ↓	ss-pLDDT ↑	TM-score ↑	$C\alpha$-RMSD ↓
Sequential Rev.	0.59→0.67	0.44→0.54	19.2→14.6	0.59→0.71	0.44→0.61	19.2→11.7
	17.0%	35.5%	-20.0%	**24.2**%	65.1%	-32.6%
Priority Search	0.59→0.69	0.44→0.57	19.2→13.0	0.59→0.68	0.44→0.62	19.2→11.2
	19.0%	**50.9**%	**-30.1**%	18.0%	**67.7**%	**-37.7**%

5 Experimental Results

We ask: (i) does inference-time, feedback-driven optimization reliably improve fidelity over baselines; (ii) do gains persist under independent predictors and diverse initializations; and (iii) why does it work?

5.1 Setup

Success rate-1: the fraction of designs whose predicted structure has pLDDT ≥ 0.8 and TM-score ≥ 0.8 against the reference. **Success rate-2** additionally requires Cα-RMSD ≤ 1.5 Å.

Independent evaluation. RF3 guides the search, so success rates computed with RF3 alone may overstate generalization. We therefore also evaluate final designs with Chai-1 (Discovery et al., 2024) and Boltz-2 (Passaro et al., 2025), which differ in architecture, training data and inductive biases (AlphaFold3 access is restricted, and ESMFold is less accurate in single-sequence mode).

RF3-guided random mutation baseline. To separate the value of search from random exploration, this baseline is (1) information-matched (same residue-level feedback), (2) compute-matched (same number of RF3 calls), and (3) initialization-matched (LigandMPNN designs). At each step it makes the same number of residue changes as RosettaSearch but random substitutions within low-metric regions, and returns the best design over runs. A **BLOSUM62-weighted** variant replaces uniform substitutions with BLOSUM62 probabilities.

Computation. Experiments used NVIDIA A100 GPUs (80GB). Wall-clock time per protein ranges from ≈2 min to ≈1.4 hours (mean 30.6 min) and is dominated by RF3 inference.

5.2 Main results

On ≈400 monomer redesign problems with low single-sequence fidelity, single-objective RosettaSearch improves ss-pLDDT by 18.1%, TM-score by 56.9% and reduces Cα-RMSD by 31.2% (Table 1); multi-objective optimization improves the metrics further overall.

We then test whether these gains raise design success rates (Table 2) for two starting distributions: (a) native monomer sequences and (b) LigandMPNN designs. In setting (a), RosettaSearch matches LigandMPNN's success rate even though LigandMPNN conditions directly on the backbone and we use it only through feedback. In setting (b), RosettaSearch achieves a 2.5× improvement (20.5% vs. 7.9%), i.e., from 32 to 79 successful designs. Under the stricter success rate-2, it reaches 8.9% (35/394) vs. 2.5% (10/394) for LigandMPNN, a 3.6× gain.

Independent oracles. Under Chai-1, absolute metrics are lower for all methods (Chai-1 gives more conservative confidence than RF3; Appendix Figure 15), but relative gains persist: roughly 2× higher success (2.4% to 4.7% for native starts, 3.1% to 6.7% for LigandMPNN starts). Boltz-2 gives higher values than RF3, and RosettaSearch raises LigandMPNN success from 14.7% to 24.0% (Appendix Table 6).

Baselines. The RF3-guided mutation baseline fails to improve on its LigandMPNN initialization (7.9%): the best candidate is typically the original design, since random substitutions in flagged

Table 2: ***(Upper)*** Baseline success rates for starting sequences and mutation baselines, evaluated by RF3 and Chai-1 (— : not reported). ***(Lower)*** RosettaSearch with `o4-mini`: success rates under different initializations and objectives. Compare each initialization (a), (b) to its baseline above.

Baseline success rates of various starting sequences		**RF3**	**Chai-1**
(a) Native monomer sequences		7.0%	2.4%
(b) LigandMPNN designs		7.9%	3.1%
RF3 Guided mutations baseline initialized with LigandMPNN designed sequences		7.9%	3.1%
BLOSUM62-weighted mutations baseline, LigandMPNN init.		8.3%	—
RosettaSearch Initialization	**Objective function**		`o4-mini` results
(a) Native monomer sequences	Single obj. (ss-pLDDT)	7.5%	4.2%
	Multi-obj. (ss-pLDDT, TM-score)	**9.1%**	**4.7%**
	Multi-obj. (ss-pLDDT, TM-score, Cα-RMSD)	8.4%	4.3%
	Cumulative	**10.1%**	—
(b) LigandMPNN designs	Single obj. (ss-pLDDT)	18.4%	5.0%
	Multi-obj. (ss-pLDDT, TM-score)	17.1%	**6.7%**
	Multi-obj. (ss-pLDDT, TM-score, Cα-RMSD)	**20.5%**	5.3%
	Cumulative	**21.6%**	—

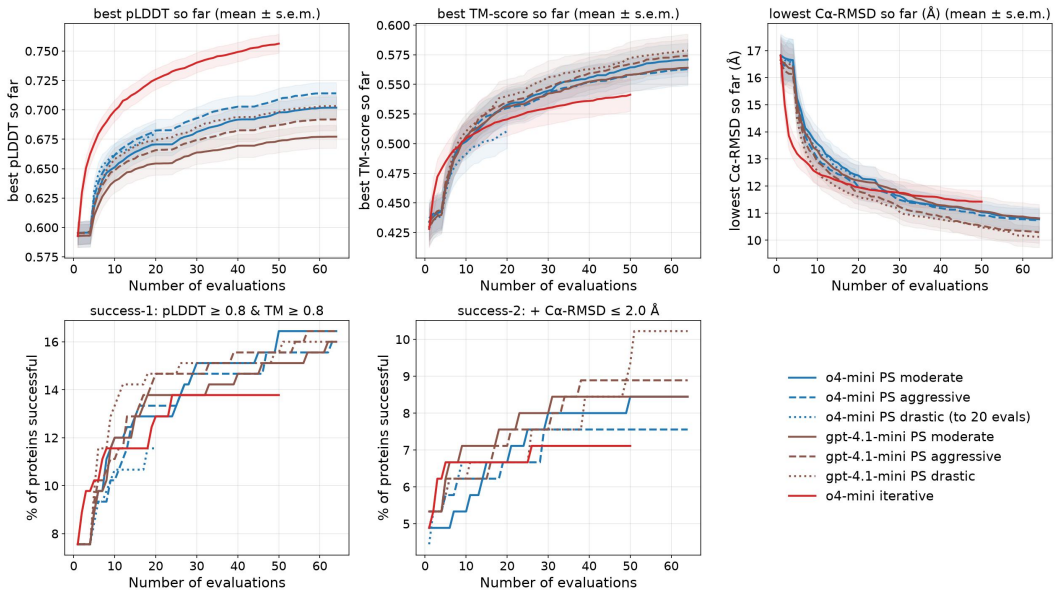


Figure 2: **Averaged metric evolution.** RosettaSearch results with Priority Search and iterative search showing the evolution of multiple objectives over the search. Results are shown for various LLMs and averaged over all proteins in the native dataset.

regions degrade fidelity. BLOSUM62-weighted substitutions reach only 8.3%, so evolutionary priors do not explain the gains. Residue-level feedback alone is insufficient: the LLM's ability to interpret it and make targeted substitutions is the critical ingredient.

5.3 Metrics show consistent evolution

Best-so-far curves improve consistently on all three fidelity metrics and the composite reward, averaged over all native proteins' trajectories on the PDB dataset.

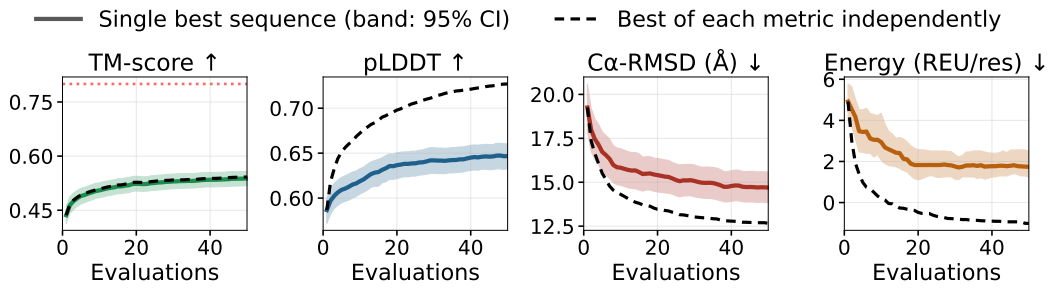


Figure 3: Physics-based energy improves alongside RF3 structure metrics during search. Rosetta ref2015 energy (REU/residue) and RF3 metrics of each protein's best-scoring sequence over evaluations. Sequences are selected on pLDDT, TM-score and RMSD only, so energy is an independent check. Lines show the median over proteins (energy) and mean (other metrics); bands are 95% bootstrap CIs.

5.4 De novo Dayhoff backbones

On 275 Dayhoff backbones (Table 3), where ProteinMPNN designs already have high fidelity and no reference sequence is given to the LLM, RosettaSearch raises success rate-1 from 72.4% to 89.5% and success rate-2 from 45.5% to 67.3%. It also rescues designs for backbones on which ProteinMPNN often fails: starting from a failing sequence, it changes only 13–21% of residues, and reduces Cα-RMSD from 32.6 to 3.8 Å for a backbone that no ProteinMPNN sample solves (Appendix Table 4 and Figure **??**).

Table 3: Evaluation on Dayhoff backbones with ProteinMPNN starting sequences. RosettaSearch uses multi-objective optimization.

Method	ss-pLDDT ↑	TM-score ↑	Cα-RMSD ↓	success rate-1	success rate-2
ProteinMPNN	0.81	0.90	3.32$\mathring{A}$	72.4% (199/275)	45.5% (125/275)
RosettaSearch	0.85	0.96	1.78$\mathring{A}$	89.5% (246/275)	67.3% (185/275)

5.5 Why RosettaSearch works

Search, not luck. Best-so-far curves improve consistently on all metrics over the search, and Priority Search improves faster than Sequential Revision; more capable LLMs perform better (Appendix Figure 2).

Priority search outperforms sequential revision (Table 1). Sequential revision greedily conditions on the current best sequence, which can cause premature convergence and error accumulation. Priority search explores several trajectories in parallel and defers commitment until downstream structural signals are observed, which makes it more robust to noisy intermediate evaluations.

Works across LLMs. With other optimizers (Appendix Table **??**), o3-mini reaches 12.3% success (above the 7.9% LigandMPNN baseline but below o4-mini), and Gemini-3 improves over o4-mini's 20.5%. The gains are therefore not specific to one model family.

A physics-based objective can be added. To test whether gains extend beyond deep-learning predictors, we added the Rosetta ref2015 energy per residue (REU/residue), computed with PyRosetta on the RF3 structure, as a fourth feedback objective (o4-mini, 415 proteins, 50 evaluations each). We select, after each evaluation, the single highest-scoring sequence per protein (per-metric best-so-far curves can overstate progress, since best values may come from different sequences). Energy is not used for selection here, so it acts as an independent check. Figure 3 shows that 84% of proteins improve, median energy falls from 4.9 to 1.7 REU/residue (60% of proteins end lower than they started), and success rate-1 rises from 7.7% to 10.6% (and with RMSD ≤ 2 Å from 3.6% to 6.5%). Energies remain above the -2 to -4 typical of well-packed native proteins.

Table 4: Tough backbones: difficult to design for backbones from Dayhoff, designed with RosettaSearch. "ProteinMPNN pass" is the share of 150 ProteinMPNN sequences that pass, and "min scRMSD" is the lowest self-consistency RMSD among these. The starting design is chosen randomly from the set of 150 ProteinMPNN designs. The metrics on the start sequence and the RosettaSearch design are shown w.r.t two oracles: RF3 and Chai-1; "changed" is the fraction of residues in the design that differ from the start; "evals" is the number of RF3 guided search steps that were needed.

	ProteinMPNN (150 samples)		RF3 eval. (start seq → design)		Chai-1 eval. (start seq → design)		num residues	num RF3
Backbone (length)	pass (%)	min scRMSD (Å)	TM-score	RMSD (Å)	TM-score	RMSD (Å)	changed (%)	evals
nR0eWr (147)	21.4	2.30	0.21 → 0.83	20.0 → 8.3	0.70 → 0.83	13.6 → 7.6	21	72
D8IVRe (149)	0.0	2.01	0.20 → 0.88	32.6 → 3.8	0.58 → 0.82	8.6 → 4.8	13	233

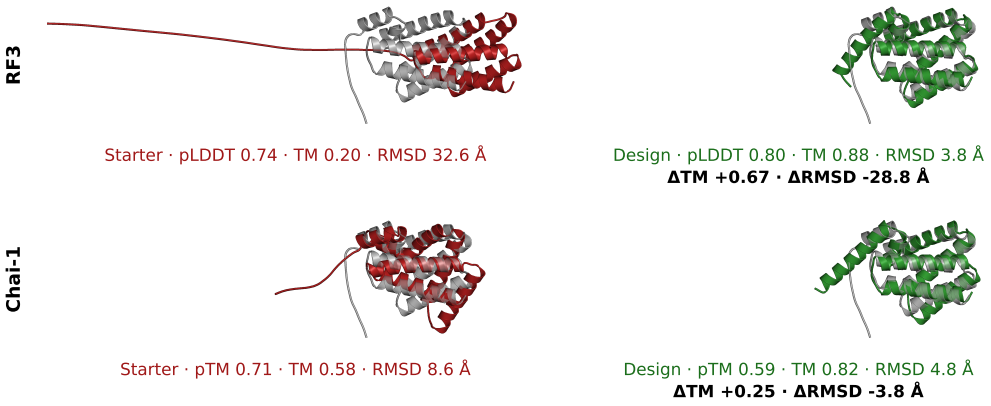


Figure 4: **D8IVRe** (`20240811526_D8IVRe_00149AA_4`): RosettaSearch improves the adherence of a failing ProteinMPNN starter to a difficult-to-design Dayhoff backbone (no ProteinMPNN sample passes). The panel overlays a single-sequence prediction (red: starter; green: design) on the target backbone (grey). RF3 is the search oracle; Chai-1 is an independent predictor. TM-score and Cα-RMSD are computed against the target, and Δ is design minus starter. Examples are hand-picked from a set in which most starters are not rescued.

The oracle matters. Search guided by ESMFold produces sequences that ESMFold scores highly but RF3 and Chai-1 do not (Appendix Table 7): ESMFold's confidence tracks a language-model prior rather than target fold, and the search exploits this. RF3 and Chai-1 agree with each other, so the gains we report are not specific to one predictor.

5.6 Difficult-to-design Dayhoff backbones

We select Dayhoff backbones where few or none of the ProteinMPNN sequences pass self-consistency Cα-RMSD $\leq$ 8.0. For each, we generate 150 ProteinMPNN sequences, count how many fold to the backbone, and start RosettaSearch from a randomly chosen failing sequence. `D8IVRe` is a backbone that no ProteinMPNN sequence passes; the starter folds far from the target with a long displaced segment, and after a few hundred RF3 evaluations the design recovers the target's helical arrangement, with a large drop in Cα-RMSD and gain in TM-score. `nR0eWr` is passed by a minority of ProteinMPNN sequences; the design reaches the correct overall topology in fewer evaluations, though a residual deviation remains. For both, Chai-1, an independent predictor, moves in the same direction, although absolute agreement with the target is lower than under RF3, with a 21% change in residues. All structures are single-sequence predictions. See Table 4 and the structures in Figures 4 and Appendix Figure 6.

5.7 Reinforced fine-tuning improves success rate

We fine-tune Qwen3.8-27B with single-turn RL on the Tinker service (LoRA rank 32, learning rate 4e$-$5, 2 proteins per batch, 8 rollouts, 300 steps). Each sampled redesign is

LLM	Success rate-1
Qwen3.8-27B	10%
+ MultiMax	16%
o4-mini	18%

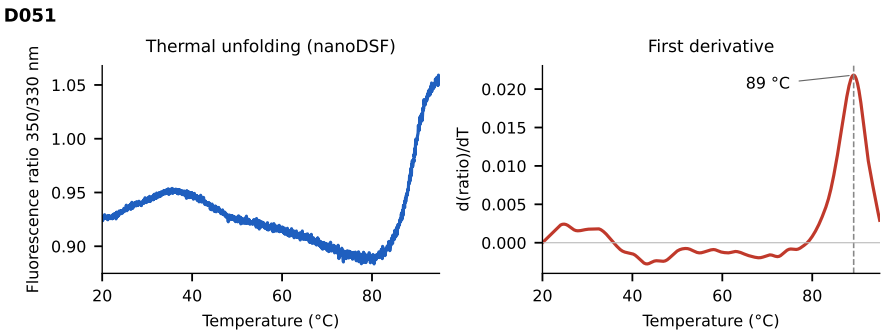


Figure 5: Thermal unfolding of design D051 (53 residues) by nanoDSF (Adaptyv Bio, 20–95 °C). Left: 350/330 nm fluorescence ratio. Right: its first derivative, with a single peak at 89 °C. The ratio has not plateaued at 95 °C, so the derivative maximum is reported rather than a fitted T_{m}.

folded 5 times with RF3, and the reward is MultiMax over equally weighted normalized gains (TM-score, pLDDT, Cα-RMSD). Training uses 200 proteins (100 PDB, 100 Dayhoff); we evaluate the same multi-objective priority search on a fixed 50-protein subset (25 PDB, 25 Dayhoff). MultiMax fine-tuning raises the success rate-1 of Qwen3.8-27B from 10% to 16%, showing that RL on the multi-objective reward improves the base model's design quality.

5.8 Experimental validation: thermal stability

We synthesized 10 RosettaSearch designs for one *de novo* backbone and measured thermal stability (Adaptyv Bio, nanoDSF, 20–95 °C). Design D051 (53 residues) shows a single sharp unfolding transition (Figure 5): the 350/330 nm ratio rises from 0.89 to 1.06 above 80 °C, and its first derivative peaks once at 89 °C (half-maximum width 85–92 °C) on a flat baseline. This is the clearest cooperative transition among the 10 variants, but the ratio has not plateaued at 95 °C and total 330 nm intensity falls to 18% of its start, so we report the derivative maximum rather than a fitted T_{m}.

6 Conclusion

RosettaSearch yields consistent structural improvements across datasets and initializations, outperforming sequential revision and random and BLOSUM62-weighted mutation. Gains persist under independent evaluation with Chai-1 (and Boltz-2 for LigandMPNN-initialized designs) and across LLM families, and extend to a physics-based objective and to a fine-tuned open-weight optimizer. Together with a preliminary wet-lab stability measurement, these results suggest LLM-driven search is a strong complement to existing design pipelines where single-shot inverse folding struggles. A limitation is that we optimize a single sequence per backbone rather than a diverse set of high-fidelity candidates; diversity-aware rewards are an important direction for future work.

Code and Data Availability. All code, data and generated sequences will be made available on GitHub.

AI Use Statement

LLMs are the optimizer in RosettaSearch (gpt4.1, o4-mini, o3-mini, Gemini-3) and Gemini-3 served as a judge of reasoning traces (Appendix A.13). We also used Claude (Anthropic) via Claude Code for analysis scripts, figures, run monitoring and LaTeX editing. The authors specified all analyses, verified assistant-produced results against stored logs and outputs, and take full responsibility for the content.

REFERENCES

Lakshya A Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J Ryan, Meng Jiang, et al. Gepa: Reflective prompt evolution can outperform reinforcement learning. *arXiv preprint arXiv:2507.19457*, 2025.

Ivan Anishchenko, Samuel J Pellock, Tamuka M Chidyausiku, Theresa A Ramelot, Sergey Ovchinnikov, Jingzhou Hao, Khushboo Bafna, Christoffer Norn, Alex Kang, Asim K Bera, et al. De novo protein design by deep network hallucination. *Nature*, 600(7889):547–552, 2021.

Souradip Chakraborty, Jiahao Qiu, Hui Yuan, Alec Koppel, Furong Huang, Dinesh Manocha, Amrit Singh Bedi, and Mengdi Wang. Maxmin-rlhf: Alignment with diverse human preferences. *arXiv preprint arXiv:2402.08925*, 2024.

Yongchao Chen, Jacob Arkin, Yilun Hao, Yang Zhang, Nicholas Roy, and Chuchu Fan. Prompt optimization in multi-step tasks (promst): Integrating human feedback and preference alignment. *CoRR*, 2024.

Ching-An Cheng, Allen Nie, and Adith Swaminathan. Trace is the next autodiff: Generative optimization with rich feedback, execution traces, and llms. *Advances in Neural Information Processing Systems*, 37:71596–71642, 2024.

Nathaniel Corley, Simon Mathis, Rohith Krishna, Magnus S Bauer, Tuscan R Thompson, Woody Ahern, Maxwell W Kazman, Rafael I Brent, Kieran Didi, Andrew Kubaney, et al. Accelerating biomolecular modeling with atomworks and rf3. *bioRxiv*, 2025.

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

Justas Dauparas, Gyu Rie Lee, Robert Pecoraro, Linna An, Ivan Anishchenko, Cameron Glasscock, and David Baker. Atomic context-conditioned protein sequence design using LigandMPNN. *Nature Methods*, 22(4):717–723, April 2025. ISSN 1548-7105. doi: 10.1038/s41592-025-02626-1.

Chai Discovery, Jacques Boitreaud, Jack Dent, Matthew McPartlon, Joshua Meier, Vinicius Reis, Alex Rogozhnikov, and Kevin Wu. Chai-1: Decoding the molecular interactions of life, October 2024.

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

Ali Essam Ghareeb, Benjamin Chang, Ludovico Mitchener, Angela Yiu, Caralyn J Szostkiewicz, Jon M Laurent, Muhammed T Razzak, Andrew D White, Michaela M Hinks, and Samuel G Rodriques. Robin: A multi-agent system for automating scientific discovery. *arXiv preprint arXiv:2505.13400*, 2025.

Juraj Gottweis, Wei-Hung Weng, Alexander Daryin, Tao Tu, Anil Palepu, Petar Sirkovic, Artiom Myaskovsky, Felix Weissenberger, Keran Rong, Ryutaro Tanno, et al. Towards an ai co-scientist. *arXiv preprint arXiv:2502.18864*, 2025.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T Joshi, Hanna Moazam, et al. Dspy: Compiling declarative language model calls into self-improving pipelines. *arXiv preprint arXiv:2310.03714*, 2023.

Kerlen T Korbeld, Vsevolod Viliuga, and Maximilian Fürst. Limitations of the refolding pipeline for de novo protein design. *bioRxiv*, pp. 2025–12, 2025.

Isaac D Lutz, Shunzhi Wang, Christoffer Norn, Alexis Courbet, Andrew J Borst, Yan Ting Zhao, Annie Dosey, Longxing Cao, Jinwei Xu, Elizabeth M Leaf, et al. Top-down design of protein architectures with reinforcement learning. *Science*, 380(6642):266–273, 2023.

Siddharth M Narayanan, James D Braza, Ryan-Rhys Griffiths, Albert Bou, Geemi Wellawatte, Mayk Caldas Ramos, Ludovico Mitchener, Samuel G Rodriques, and Andrew D White. Training a scientific reasoning model for chemistry. *arXiv preprint arXiv:2506.17238*, 2025.

Allen Nie, Ching-An Cheng, Andrey Kolobov, and Adith Swaminathan. The importance of directional feedback for llm-based optimizers. *arXiv preprint arXiv:2405.16434*, 2024.

Giseung Park, Woohyeon Byeon, Seongmin Kim, Elad Havakuk, Amir Leshem, and Youngchul Sung. The max-min formulation of multi-objective reinforcement learning: From theory to a model-free algorithm. *arXiv preprint arXiv:2406.07826*, 2024.

Saro Passaro, Gabriele Corso, Jeremy Wohlwend, Mateo Reveiz, Stephan Thaler, Vignesh Ram Somnath, Noah Getz, Tally Portnoi, Julien Roy, Hannes Stark, et al. Boltz-2: Towards accurate and efficient binding affinity prediction. *BioRxiv*, 2025.

Reid Pryzant, Dan Iter, Jerry Li, Yin Tat Lee, Chenguang Zhu, and Michael Zeng. Automatic prompt optimization with" gradient descent" and beam search. *arXiv preprint arXiv:2305.03495*, 2023.

Adam J Riesselman, John B Ingraham, and Debora S Marks. Deep generative models of genetic variation capture the effects of mutations. *Nature methods*, 15(10):816–822, 2018.

Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M Pawan Kumar, Emilien Dupont, Francisco JR Ruiz, Jordan S Ellenberg, Pengming Wang, Omar Fawzi, et al. Mathematical discoveries from program search with large language models. *Nature*, 625(7995):468–475, 2024.

Abhishek Roy, Geelon So, and Yi-An Ma. Optimization on pareto sets: On a theory of multi-objective optimization. *arXiv preprint arXiv:2308.02145*, 2023.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Joar Skalse, Nikolaus Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward gaming. *Advances in Neural Information Processing Systems*, 35:9460–9471, 2022.

Giovanni Stracquadanio and Giuseppe Nicosia. Computational energy-based redesign of robust proteins. *Computers & chemical engineering*, 35(3):464–473, 2011.

Hao Sun, Xiao Liu, Yeyun Gong, Yan Zhang, Daxin Jiang, Linjun Yang, and Nan Duan. Allies: Prompting large language model with beam search. *arXiv preprint arXiv:2305.14766*, 2023.

Kyle Swanson, Wesley Wu, L Bulaong Nash, John E Pak, and James Zou. The virtual lab: Ai agents design new sars-cov-2 nanobodies with experimental validation. biorxiv. 2024.

Sophia Tang, Yinuo Zhang, and Pranam Chatterjee. Peptune: De novo generation of therapeutic peptides with multi-objective-guided discrete diffusion. *ArXiv*, pp. arXiv–2412, 2025.

Fabien Tricoire. Multi-directional local search. *Computers & operations research*, 39(12):3089–3101, 2012.

Kathryn Tunyasuvunakool, Jonas Adler, Zachary Wu, Tim Green, Michal Zielinski, Augustin Žídek, Alex Bridgland, Andrew Cowie, Clemens Meyer, Agata Laydon, et al. Highly accurate protein structure prediction for the human proteome. *Nature*, 596(7873):590–596, 2021.

Xinyuan Wang, Chenxi Li, Zhen Wang, Fan Bai, Haotian Luo, Jiayou Zhang, Nebojsa Jojic, Eric P Xing, and Zhiting Hu. Promptagent: Strategic planning with language models enables expert-level prompt optimization. *arXiv preprint arXiv:2310.16427*, 2023.

Joseph L. Watson, David Juergens, Nathaniel R. Bennett, Brian L. Trippe, Jason Yim, Helen E. Eisenach, Woody Ahern, Andrew J. Borst, Robert J. Ragotte, Lukas F. Milles, Basile I. M. Wicky, Nikita Hanikel, Samuel J. Pellock, Alexis Courbet, William Sheffler, Jue Wang, Preetham Venkatesh, Isaac Sappington, Susana Vázquez Torres, Anna Lauko, Valentin De Bortoli, Emile Mathieu, Sergey Ovchinnikov, Regina Barzilay, Tommi S. Jaakkola, Frank DiMaio, Minkyung Baek, and David Baker. De novo design of protein structure and function with RFdiffusion. *Nature*, pp. 1–3, July 2023. ISSN 1476-4687. doi: 10.1038/s41586-023-06415-8.

Anjiang Wei, Allen Nie, Thiago SFX Teixeira, Rohan Yadav, Wonchan Lee, Ke Wang, and Alex Aiken. Improving parallel program performance with llm optimizers via agent-system interfaces. *arXiv preprint arXiv:2410.15625*, 2024.

Yingce Xia, Peiran Jin, Shufang Xie, Liang He, Chuan Cao, Renqian Luo, Guoqing Liu, Yue Wang, Zequn Liu, Yuan-Jyue Chen, et al. Nature language model: Deciphering the language of nature for scientific discovery. *arXiv preprint arXiv:2502.07527*, 2025.

Nuoya Xiong and Aarti Singh. Projection optimization: A general framework for multi-objective and multi-group rlhf. *arXiv preprint arXiv:2502.15145*, 2025.

Weijia Xu, Andrzej Banburski-Fahey, and Nebojsa Jojic. Reprompting: Automated chain-of-thought prompt inference through gibbs sampling. *arXiv preprint arXiv:2305.09993*, 2023.

Fanglei Xue, Andrew Kubaney, Zhichun Guo, Joseph K. Min, Ge Liu, Yi Yang, and David Baker. Improving Protein Sequence Design through Designability Preference Optimization, May 2025.

Kevin K Yang, Zachary Wu, and Frances H Arnold. Machine-learning-guided directed evolution for protein engineering. *Nature methods*, 16(8):687–694, 2019.

Kevin K Yang, Sarah Alamdari, Alex J Lee, Kaeli Kaymak-Loveless, Samir Char, Garyk Brixi, Carles Domingo-Enrich, Chentong Wang, Suyue Lyu, Nicolo Fusi, et al. The dayhoff atlas: scaling sequence diversity for improved protein generation. *bioRxiv*, pp. 2025–07, 2025.

Rui Yin, Brandon Y Feng, Amitabh Varshney, and Brian G Pierce. Benchmarking alphafold for protein complex modeling reveals accuracy determinants. *Protein Science*, 31(8):e4379, 2022.

Mert Yuksekgonul, Federico Bianchi, Joseph Boen, Sheng Liu, Zhi Huang, Carlos Guestrin, and James Zou. Textgrad: Automatic" differentiation" via text. *arXiv preprint arXiv:2406.07496*, 2024.

Yang Zhang and Jeffrey Skolnick. Tm-align: a protein structure alignment algorithm based on the tm-score. *Nucleic acids research*, 33(7):2302–2309, 2005.

Xiangxin Zhou, Dongyu Xue, Ruizhe Chen, Zaixiang Zheng, Liang Wang, and Quanquan Gu. Antigen-specific antibody design via direct energy-based preference optimization. *Advances in Neural Information Processing Systems*, 37:120861–120891, 2024.

A Appendix

A.1 Calculation of $C\alpha$-RMSD

To measure structural fidelity $C\alpha$-RMSD $(\mathbf{x}, \mathbf{x}^*)$ of $\hat{\mathbf{x}}$ to the provided reference native structure or reference backbone $\mathbf{x}^*$, where $L = L^*$, we superimpose the predicted structure of the generated proposal over the reference structure using an algorithm (Zhang & Skolnick, 2005) that aligns remotely homologous protein structures (pairs of proteins whose sequences have sequence identity less than 30%). We use this approach since the generated sequence could have an arbitrarily low sequence similarity to the native sequence.

A.2 Multi-objective optimization details

First, we evaluated a round-robin scheme in which optimization alternates between objectives across iterations, allowing the model to focus on a single criterion at a time while implicitly encouraging coverage of the full objective set. Candidate evaluations are ranked using a single objective at a time, with the active objective cycling across searches to encourage balanced coverage of the objective space.

Second, we experimented with scheduled reward formulations in which the relative emphasis on different objectives is varied over the course of optimization, including curricula that prioritize coarse global structure in early iterations and increasingly emphasize local refinement signals in later iterations. With priority search which works in parallel, the priority assigned to different objectives varies across evaluations based on a predefined curriculum.

A.3 Sensitivity to reward weights

We selected the weights (w_p, w_{tm}, w_r) of the weighted-sum reward on a held-out set of 20 proteins. Because this set is small, we compared configurations by the % change in each metric (relative to the starting sequences). Table 5 lists the unnormalized weights to make their relative importance explicit.

Table 5: Reward-weight configurations on the held-out set of 20 proteins: % improvement (pLDDT, TM-score) or reduction (Cα-RMSD). The chosen configuration (normalized: $w_p = \frac{1}{11}, w_{tm} = w_r = \frac{5}{11}$) gives the largest TM-score and Cα-RMSD improvements.

Configuration	w_p	w_{tm}	w_r	pLDDT ↑	TM-score ↑	Cα-RMSD ↓
Equal weights	1	1	1	13.18%	34.84%	-30.21%
High weight on pLDDT	5	1	1	16.41%	42.12%	-28.92%
High weight on TM-score	1	5	1	16.36%	35.06%	-30.10%
High weight on RMSD	1	1	5	15.44%	46.84%	-30.62%
High TM-score and RMSD (chosen)	1	5	5	16.33%	**49.34%**	**-32.67%**

The three metrics are not independent: pLDDT measures local confidence, TM-score global topology, and Cα-RMSD geometric deviation. When all three are optimized jointly with equal weights, the LLM receives conflicting feedback signals and optimization is unstable. pLDDT is also the easiest metric to increase with nonsensical edits (i.e., it is prone to reward hacking, Section 3.3), which sharply worsens TM-score and Cα-RMSD; in other cases, an edit that improves pLDDT in a disordered region worsens TM-score by altering a loop conformation. We therefore choose the configuration with the largest TM-score and Cα-RMSD improvements, as this aligns with our goal of matching a given backbone (backbone-conditioned design).

A.4 Evaluation on the Dayhoff dataset

We show that our approach is not overfitting to the vagaries of the dataset or the sequence design approach involved, by evaluating it on another dataset which involves a subset of the Dayhoff atlas (Yang et al., 2025). The subset of Dayhoff that we use consists of 275 randomly sampled unconditionally generated RFDiffusion *de novo* backbones, for which ProteinMPNN was used to generate sequences. Note that this dataset involves sequences that already have very good fidelity

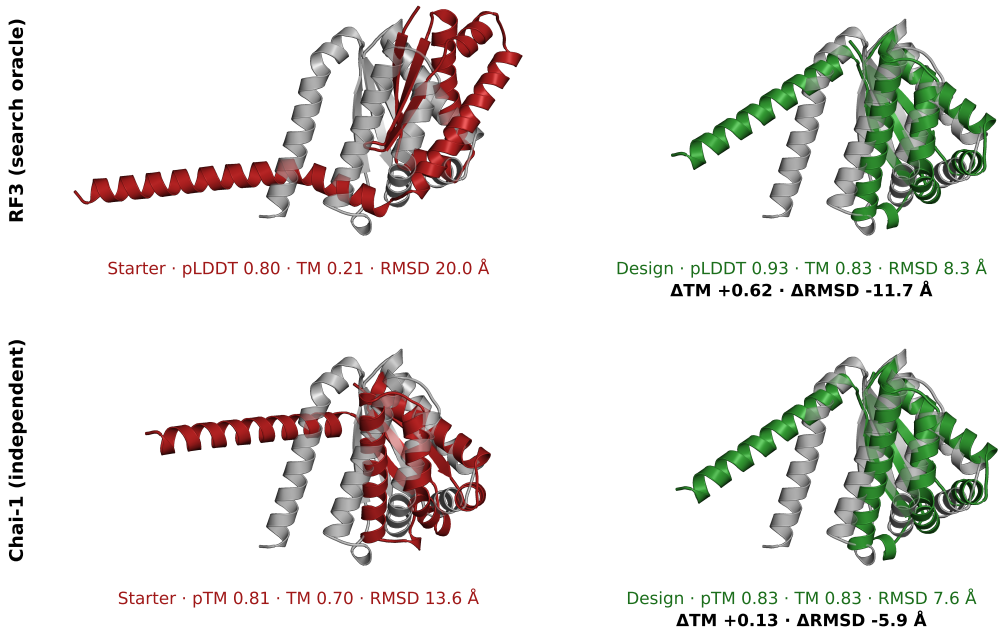


Figure 6: **nR0eWr** (`202408102344_nR0eWr_00147AA_2`): a second difficult-to-design Dayhoff backbone (a minority of ProteinMPNN samples pass). Conventions as in Figure 4: red is the starter, green is the design, and grey is the target; Δ is design minus starter.

metrics (and hence success rates) as compared to the PDB dataset, which was curated based on low values on structural fidelity metrics. As before, success-rate-1 is calculated as pLDDT ≥ 0.8, TM-score ≥ 0.8 and success-rate-2 is calculated as pLDDT ≥ 0.8, TM-score ≥ 0.8, Cα-RMSD $\leq 1.5\text{Å}$. We find that RosettaSearch improves success-rate-1 from 72.4% to 89.5% and success-rate-2 from 45.5% to 67.3%.

Unlike the PDB monomer evaluations, no reference sequence is provided to the LLM optimizer here: it must propose edits based solely on structural feedback from RosettaFold3. This makes the Dayhoff evaluation a stricter test of the LLM's ability to act on feedback alone, showing that RosettaSearch does not rely on reference-sequence context.

A.5 ESMFold as a search oracle

Search guided by ESMFold produces sequences that ESMFold scores highly but that RF3 and Chai-1 do not (Table 7). RF3 and Chai-1 agree with each other, while ESMFold's scores track RF3's only weakly, so the gains are specific to ESMFold's predictor rather than real improvements in foldability. A likely reason is that ESMFold predicts structure from a protein language model's learned prior without an MSA, so it is confident for sequences the language model finds plausible even when the structure does not match the target; the search learns to produce such sequences.

A.6 Additional structure-prediction oracle

Table 6 re-evaluates the best RosettaSearch designs with Boltz-2, a third structure predictor.

Table 6: Boltz-2 re-evaluation of the best results from Table 2 (LigandMPNN initialization). Boltz-2 yields higher metric values than RF3, which translates to higher success rates.

	RF3	**Chai-1**	**Boltz-2**
LigandMPNN designs	7.9%	3.1%	14.7%
RosettaSearch, multi-obj.	**20.5%**	**6.7%**	**24.0%**

Table 7: ESMFold-guided search on 415 native proteins. Left: ESMFold (the search oracle) on the starting sequence vs. the best sequence found; improvement is the mean per-protein change (mean ± s.e.m.; RMSD improvement is the reduction). Right: the same best sequences scored by RF3 and Chai-1.

	ESMFold (search oracle)			RF3	Chai-1
	Start	Best	Improvement	Best	Best
ss-pLDDT ↑	0.782	0.804	$+3.4\% \pm 0.4$	0.602	0.516
TM-score ↑	0.830	0.859	$+7.5\% \pm 1.3$	0.427	0.387
Cα-RMSD (Å) ↓	4.97	3.72	$+17.2\% \pm 1.0$	18.71	16.75
Success-1 (%) ↑	56.1	65.1	–	6.5	2.9
Success-2 (%) ↑	33.5	42.4	–	2.2	1.2

A.7 Results with a Vision Language Model (VLM) as the optimizer

We additionally evaluate a vision-language model (VLM) variant of RosettaSearch, where the optimizer receives both textual objective feedback and visual structural feedback from predicted protein structures. This extension is designed to test whether visual reasoning over 3D structural renderings can provide useful optimization signal beyond the text-only feedback used by the LLM optimizer. Because our current VLM implementation supports only iterative revision, we compare it against the corresponding text-only multi-objective optimizer using the same Sequential Revision algorithm. In both cases, the optimizer jointly targets single-sequence pLDDT and TM-score, and we evaluate final design success using two independent structure-prediction models, RF3 and Chai-1.

Table 8: Design success rates obtained by RosettaSearch with a vision-language model (VLM) with visual structural feedback, compared to the corresponding LLM performance while optimizing the two following objectives simultaneously: single sequence pLDDT and TM-score. Both approaches used the Sequential Revision algorithm as our VLM extension is currently limited to an iterative search.

			Success Rate (RF3)	**Success Rate (Chai-1)**
Initialization	**Algorithm**	**Approach**		
(a) Native monomer sequences	Sequential-Revision	unoptimized native protein sequences	7.0%	2.4%
		LLM multi-obj. (ss-pLDDT, TM-score)	**8.9**%	**4.7**%
		VLM multi-obj. (ss-pLDDT, TM-score)	8.3%	4.6%
(b) LigandMPNN designs	Sequential-Revision	unoptimized LigandMPNN designs	7.9%	3.1%
		LLM multi-obj. (ss-pLDDT, TM-score)	**17.4**%	6.7%
		VLM multi-obj. (ss-pLDDT, TM-score)	16.6%	**7.1**%

Table 8 shows that the VLM optimizer achieves broadly comparable performance to the text-only LLM optimizer. When initialized from native monomer sequences, the VLM variant reaches 8.3% success under RF3 and 4.6% under Chai-1, slightly below the corresponding LLM results of 8.9% and 4.7%. When initialized from LigandMPNN designs, the VLM variant achieves 16.6% success under RF3 and 7.1% under Chai-1, closely matching the LLM variant, which obtains 17.4% and 6.7%, respectively. Thus, visual structural feedback does not lead to a clear aggregate improvement in this preliminary experiment, but it preserves most of the gains from multi-objective text-only optimization and slightly improves Chai-1 success in the LigandMPNN initialization setting. Qualitatively, we find that the VLM produces richer optimization rationales that refer to spatial and structural features visible in the rendered predictions, suggesting that visual feedback may be useful for future extensions with stronger VLMs or more targeted structural objectives.

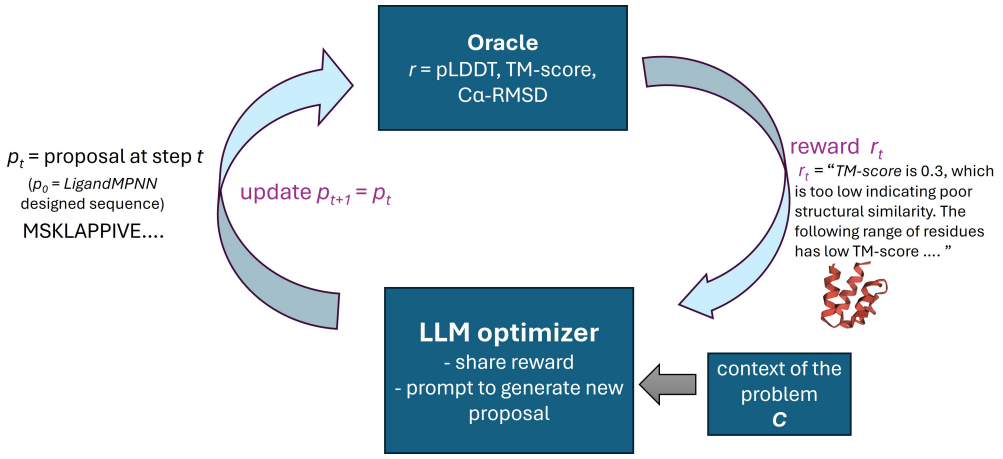


Figure 7: Schematic showing the approach followed by RosettaSearch

Algorithm 1 Search

Require: ϕ_0: starting protein sequence
$\mathcal{O}$: LLM Optimizer
G: Guide that provides feedback (which provides feedback about pLDDT, TM-score, $C\alpha$-RMSD)
K: number of candidates considered for exploration
N: number of proposals generated per candidate
$S(\cdot)$: score function to prioritize candidates

Initialize priority queue $\mathcal{M}$ which store candidates of parameters and their scores (i.e. $c = (\phi, \text{score})$.
Insert K copies of ϕ_0 into $\mathcal{M}$ with maximal priority
for iteration $t = 1, 2, \ldots$ **do**
  (Exploit) Select best candidate ϕ_t^* of $c^* \leftarrow \arg\max_{c \in \mathcal{M}} S(c)$
  (Explore) Select top-K candidates from $\mathcal{M}$ as $\mathcal{C}$
  for each candidate $c \in \mathcal{C}$ **do**
    Run c using guide G to extract score and feedback
    Collect rollouts $\mathcal{R}_c = \{(\phi, \text{score}, \text{feedback})\}$, where $\phi \in c$.
  end for
  (Propose) Initialize proposal set $\mathcal{P} \leftarrow \emptyset$
  for each candidate $c \in \mathcal{C}$ **do**
    $f \leftarrow \bigcup\{\text{feedback} | \text{feedback} \in r, r \in \mathcal{R}_c\}$
    for $p = 1$ to N **do**
      $\phi' \leftarrow$ optimizer.step(ϕ, f)
      Evalute ϕ' with guide G to get score$'$
      Add candidate $c' = (\phi', \text{score}')$ to $\mathcal{P}$
    end for
  end for
  (Memory Update)
  for each candidate $c \in \mathcal{C} \cup \mathcal{P}$ **do**
    Compute priority $S(c)$
    Insert c into priority queue $\mathcal{M}$
  end for
end for
return best designs $\{\phi_t^*\}$

A.8 Where are the most updates made?

To understand whether there is a propensity in the substitutions made by the LLM, we analyze the positional information where the updates are made by dividing each protein sequence into four bins:

Dayhoff backbone id	ProteinMPNN design (from Dayhoff atlas)	RosettaSearch optimized	Sequence identity (blastp)
20240891510_CvNknz_00140AA_2	AAEKEKEELKKAREKMEKEIKE LVEKLKGYIIISILTAVLREVSGK LVVEIKVVAGSESIKLEELEAKM KEYIAKKKAEFAALGLKVEEEIK RSEKEVVAKLTAPLDKLPPESD KLKRTLEIINLETGEVEVRKEEIT LE	MKAKQKLALKKAREKMALQIK ELVAKLKGYIIISILTAVLREVTA KLVVEIKVVAGSASIRLEELEAK MKEYIAKKKAEFAALGLKVEEE IKPSPEEVEVKLTAPLDKLPESD TLKRTLEIINLETGEVEVRKEEIT LAA	88.2%
20240891627_ZY18lJ_00304AA_2	ATLELTGPLTPEFLAKARELLKP LAAAEEALTALDEVVALGNATV TIYKQQGWTTTSTLTLTIERRPN EEKVTVTLTVTATAADGTTRTT TITATVTYVELPGGPPVLKRTLE EKASDGSSFTLSISTTGAMELSL QVPGKEPVSLTLDIIKGGELERK FYVDPSSGNRIIVIKATKDHIVVI IRIRKLTPESKELLKEILEEVFKE AKKEGAKVHVVILADTLENAKI ALDIVLEQALKEGVTLSISLGVA VKAEEADEVYEELVKYVEEKLE KIREEGKVKVERVEVRLPGKTA VFEVEEEI	MTLENTAALTAEFLAKARELLK PLAAAEEALTALDEVVALGLAT VTILALQGWTTTSTLTLTIERKA NEEKVTVTLTLTALAANGTTRT TTITATVTYVELPGGPPVLKRTL EEKASDGSSFTLSISTTGAMELS LQVPGKEPVSLTLDIITGGELER KFYVDPSSGNRIIVIKATKDHIV VIIRIRKLTPESKELLKEILEEVFK EAKKEGAKVHVVILADTLENAK IALDIVLEQALKEGVTLSISLGV AVKAEEADEVYEELVKYVEEKL EKIREEGKVKVERVEVRLPGKT AVFEVEEEI	95.4%
20240891538_B8FFVA_00355AA_0	TAQLQADLDRVVAALAANPRIQ AIRAILAQRPEVEPEAVSNERAL ELVMEVLGLRDRAEARALMVP AARLFGVPVPPLSPEEQELLAAA LARPVLTVALVFRVRYSRFEVL EEEEREYEDPERPGLRVRERRRV VRVDGYLYRFVHIEVRDEATGK VVRAGLLSVQVGPDHYVISVDF EAEPGASEEAVDLVVDRVNEEL REIAKEIKKATTLSLSIRDTAHAP KVIQAFVAAANEIAPYVKSLKV SIHVGKHPGPITLTLTLTPGVDA LTVIAENCAELNVNVTAGVKEL RFTAYSTKITVTLAPDAKIEKVE FVETTPETVINTIAEVVLECLATE EDALGVVVLREEVGAVI	TAQLQADLDAVVAALAANPEIQ AIRAILAQRPEVLPEAVSNERAL ELVMIVLGLRDRAEARALMVPA ARLFGVPVPPLSPEEQELLAAAL ARPVLTVALVFRVRYSRFEVLE EEEREYEDPERPGLRVRERRRV VRVDGYLYRFVHIEVRDEATGK VVRAGLLSVQVGPDHYVISVDF EAEPGASEEAVDLVVDRVNEEL REIAKEIKKATTLSLSIRDTAHAP KVIQAFVAAANEIAPYVKSLKV SIHVGKHPGPITLTLTLTPGVDA LTVIAENCAELNVNVTAGVKEL RFTAYSTKITVTLAPDAKIEKVE FVETTPETVINTIAEVVLECLATE EDALGVVVLREEVGAVI	98.9%

Table 9: Examples of ProteinMPNN sequences from the Dayhoff atlas that were redesigned using RosettaSearch, corresponding to the structures from Figure 10. The last column shows the percentage sequence identity (as calculate by blastp sequence similarity) between the starting ProteinMPNN design and the RosettaSearch optimized final design.

N-terminal representing the beginning (0-25%) of the sequence, two bins to represent the middle regions and C-terminal to represent the end of the protein sequence. We compare the final "successful" generated sequence (i.e. ones that satisfied the success criterion) to the starting sequence and calculate the number of updates that fall into each bin. As seen in Figure 12, at 30% the C-terminal sees the highest percentage of updates whereas the other regions have between 22 to 24% of the updates.

A.9 Sequence diversity of generated sequences

We next look at the diversity of the generated sequences on the PDB dataset where we optimize native monomers (corresponding to the results in Table 2). Figure 11 shows the distribution of per-protein average sequence identity between the starting sequences and the two best generated sequences on the LigandMPNN dataset. Although the mean sequence identity is high (82.4%), the distribution is wide, ranging from near-identical sequences to substantial divergence. This indicates that the method

PDB id	Native sequence	LigandMPNN design	RosettaSearch optimized
7z05	KESLTHASVLSAYTASYKTT AIKSIADNAVVLDIGYGKGN DGPRYAVRPLTVTGIDTAA RMLAIADQNKPENVTLVKQ GFFTHITKTSNTYTHVIAFNS LHYPLASSHPDTLVQRLPTC PANILIPCHHLLEGIQTPTYS VVKDEDMWCVKVTKNEFIE SSYNYDVFVKALESKYHVT IGSLLDCVEKPSTRSITPTLW TAMRNFVNNDQEMQRILSG YITFNLTPLPPKVEIINDWLD	EDGLTHADVLFAFRGSAQT RAVDAIPEGSTVLDIGVGTG RLFPLYARRPLTVTGIDVDA AALAKAEAHRPANVTLVHQ DLFTYLKETDVEYTYVLAF DALHYPLSRSDPRELVDLLP RAPAFILIPDFRLLEGVETPT FSVRPDGDRVLVRIGGRELT YDYYDLDAFEAALAERYDV RRGTLLDETTKPADPSITDE LWERMRALVANDEDLQRIL GGFRTYDLTPKPPPAAAAA AAAA	MESLTHADVLFAYTASYKT TAIDAIPEGSTVLDIGVGTGR LFPLYARRPLTVTGIDVDAA ALAKAEAHRPANVTLVHQD LFTYLKETDVEYTYVLAFD ALHYPLSRSDPRELVDLLPR APAFILIPDFRLLEGVETPTF SVRPDGDRVLVRIGGRELTY DYYDLDAFEAALAERYDVR RGTLLDETTKPADPSITDEL WERMRALVANDEDLQRILG GFRTYDLTPKPPPAEIINDW LD
4quc	EYVVEKILGKRFVNGRPQV LVKWSGFPNENNTWEPLEN VGNCMKLVSDFESEVFRL	SADPDKILGRLLRNGFPYYH IHWLHMTQENITWRSAAHL GNIGFQWILFRYALQRK	EYVVEKILGKRFVNGRPYY HIKWLGMTQENITWRSAAN VVGCMKLVSDFESELQRK
8os3	GSSSPPSPPLDLHVTDAGRK HIAIAWKPPEKNGGSPIIGY HVEMCPVGTEKWMRVNSR PIKDLKFKVEEGVVPDKEY VLRVRAVNAIGVSEPSEISE NVVAKD	GEVAAPTEPTSLKVTAASK DYIVIEWQSPESDGGTPKRG YSVQMVRVFTENWVEVNS ALIKSNSYQVTAGVTPNEVF KLRVTAFNDAGSSEPSQESE NVKKAT	GSAAAPTEPTSLKVTAASK DYIVIEWQSPEKNGGSPILGF SVQMVRVFTENWVEVNSRP IKSNSYQVTAGVTPNEVFKL RVTALDQAGSAEPSEESENV VKAT
8h02	TARLLRAPVAGTIKLGKKA RTRPYRTRHGEEALLAEAN FDLVLEGKGRKETFAILQGS TIFVQDGDKVAAEAILAEVP V	DYEVIFAHEPGFKILGDEAV LERYVTKDGNDVLRAKKNF TLIIVGKKKKHESNGPEGSG VLVRDGDTVDKGEPLAIVP L	TARLLRAPVAGTIILGDEAV LERYVTKDGNDVLRAKKNF TLIIVGKGRKETFNGPEGSG VLVRDGDKVAAGEPLAIVP L

Table 10: LigandMPNN sequences that were redesigned using RosettaSearch. The original native PDB monomer sequence that was being redesigned is also shown. The sequences shown here correspond to the structures shown in Figure 1 and Figure 9.

adapts the extent of sequence modification on a per-protein basis, making conservative edits when sufficient and exploring more aggressive redesigns when required.

Table 11: Sequential revision vs Priority search, average % improvement (or reduction) along with standard error and p-value from a paired t-test

Single objective (pLDDT)			
	ss-pLDDT	TM-score	$C\alpha$-RMSD
Sequential Revision	19.4%	27.4%	-15.0%
Priority Search	18.1 $\pm$0.85% (p=$1.61e^{-77}$)	56.9$\pm$4.92% (p=$3.42e^{-57}$)	-31.2 $\pm$1.28% (p=$8.71e^{-40}$)
Multi objective (ss-pLDDT, TM-score, $C\alpha$-RMSD)			
	ss-pLDDT	TM-score	$C\alpha$-RMSD
Sequential Revision	24.2$\pm$1.1% (p=$1.9e^{-81}$)	65.1$\pm$5.2% (p=$2.1e{-}65$)	-32.6$\pm$1.3% (p=$1.42e^{-40}$)
Priority Search	18.0 $\pm$0.82% (p=$4.9e^{-81}$)	67.7 $\pm$5.82% (p=$6.1e^{-68}$)	-37.7 $\pm$1.24% (p=$1.1e^{-45}$)

A.10 DOES THE REASONING INDICATE HALLUCINATION?

We analyze the LLM expert opinion (from the Gemini-3 experts) to assess for evidence of hallucination in the reasoning produced by RosettaSearch, by verifying that the proposed sequence edits are actually applied and present in the generated sequences.

Across the 22 cases deemed invalid by expert evaluation, failures overwhelmingly stem from reasoning–action inconsistencies rather than poor design intent. The dominant failure mode is mutation budget violation, where the LLM explicitly reasons about making at most 10 substitutions but produces sequences containing 15–30+ mutations. A second major class of errors involves hallucinated inputs or indexing mistakes, including incorrect assumptions about the native sequence, misidentified residue identities or positions, and reasoning about regions that do not correspond to the provided feedback.

Notably, a subset of failures also includes no-op generations, where the proposed sequence is unchanged despite detailed mutation plans in the reasoning.

A.11 FURTHER RESULTS

Results on the PDB dataset with multi-objective optimization and priority search algorithm (corresponding to Table 2) are analyzed to show the improvements in fidelity as a function of sequence length in Figure 8b. Here *short* refers to sequence lengths of less than 100 amino acids, a length of 100 to 350 is considered *medium* length and greater than 350 amino acids is considered a *long* protein. Our PDB dataset with $\approx$400 structures has 249 medium, 129 long and 36 short monomers. We find that the improvements obtained by RosettaSearch in pLDDT and TM-score are the highest for medium length proteins, whereas the improvements in Cα-RMSD are the highest for short proteins.

A.12 HUMAN EXPERTS' THOUGHTS ON THE LLM REASONING

We examined the reasoning trace produced by the large language model for every proposal that it generates. We had 3 domain experts with varying degrees of experience and background in biochemistry, structural biology, biophysics, machine learning, directed evolution and protein engineering who were shown details of each optimization step and the reasoning as shown in Figure 13. We observed that depending on the feedback, the LLM either proposes targeted, minimal edits focused on low-confidence regions or generates an entirely new sequence that satisfies global design constraints such as length, residue diversity, and lack of internal repeats.

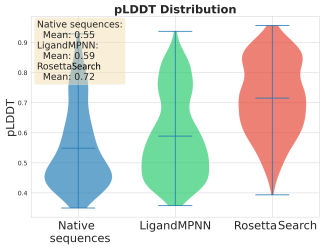
Qualitatively, we evaluate results in a setting where the LLM is provided the native sequence as context. The experts found the reasoning to have a more fixed pattern. The LLM focused on N-terminal regions that had low TM-scores. However, N- and C-terminal regions of predicted structures commonly exhibit poor alignment with the reference not because the sequence is infeasible, but because these regions are typically flexible and loop-rich, where close structural agreement is not expected.

Further, human experts agree that the sequence modifications suggested makes sense in isolation (e.g., replace buried charged residues with nonpolar ones) but that are not logical in the context (e.g., "the residue that was changed from charged to nonpolar was not actually buried").

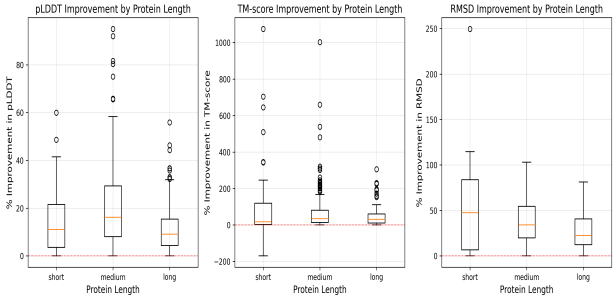
A.13 LLM EXPERTS' OPINION OF REASONING

We used Gemini-3 as an independent evaluator of LLM reasoning quality. Specifically, three Gemini-3 experts were shown the reasoning output generated by o4-mini for each proposed protein sequence modification and asked to assess whether the reasoning was logically sound and consistent with the described sequence edits. Across 50 evaluated reasoning cases, the group of Gemini-3 experts were unanimous in their opinion where they marked 28 instances (56%) as valid and 22 instances (44%) as invalid.

Figure 8: Results showing improvements in the various metrics



(a) Distribution of pLDDT values of the baseline LigandMPNN designs that were used as starter sequences for RosettaSearch to do single objective optimization of pLDDT and the obtained improved pLDDT values.



(b) Stratifying the % improvement in fidelity metrics by the length of the protein sequence. (*Left*) pLDDT improvement, (*Center*) TM-score improvement, (*Right*) $C\alpha$-RMSD improvement. Results are shown for Search with multi-objective optimization.

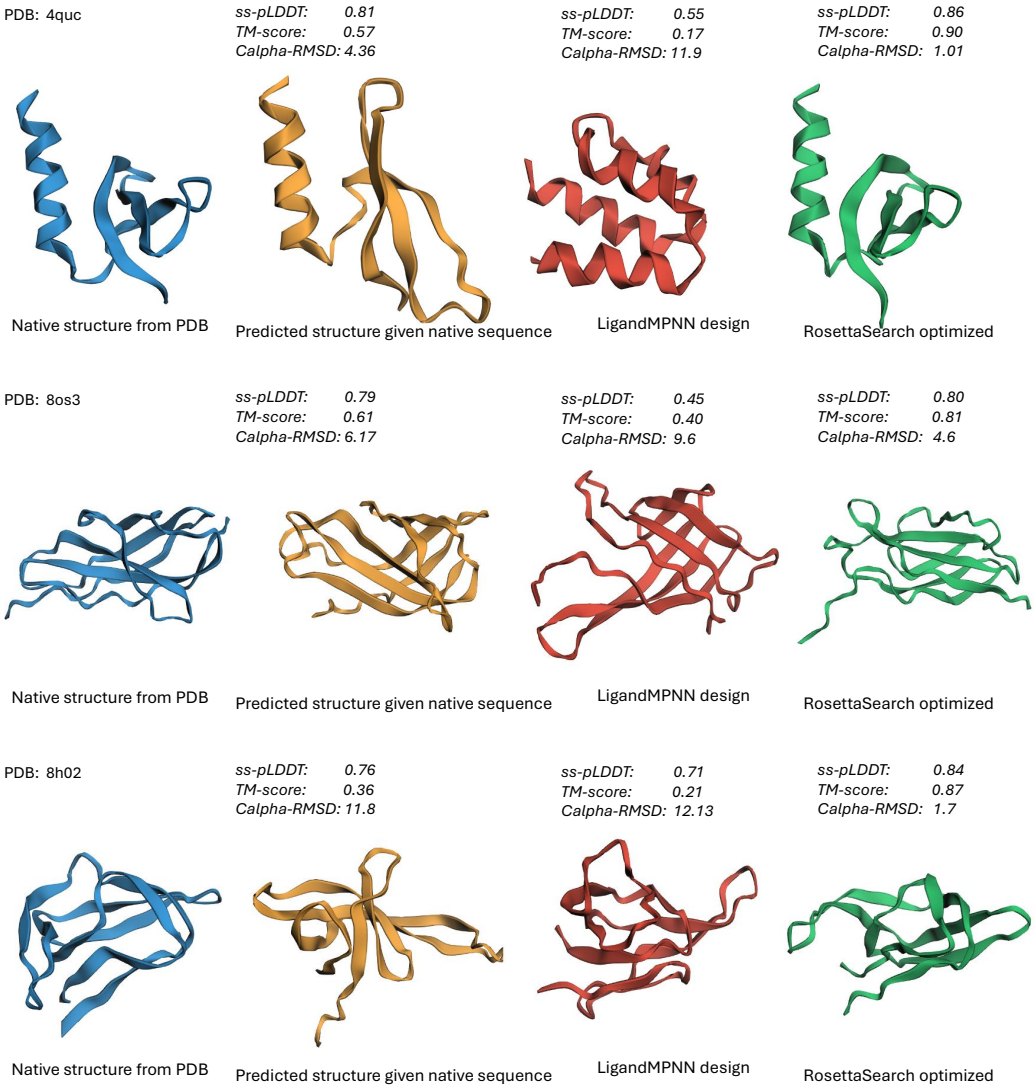


Figure 9: More examples of structures from the PDB dataset that were optimized using RosettaSearch successfully. See Table 10 for the amino-acid sequences corresponding to these structures.

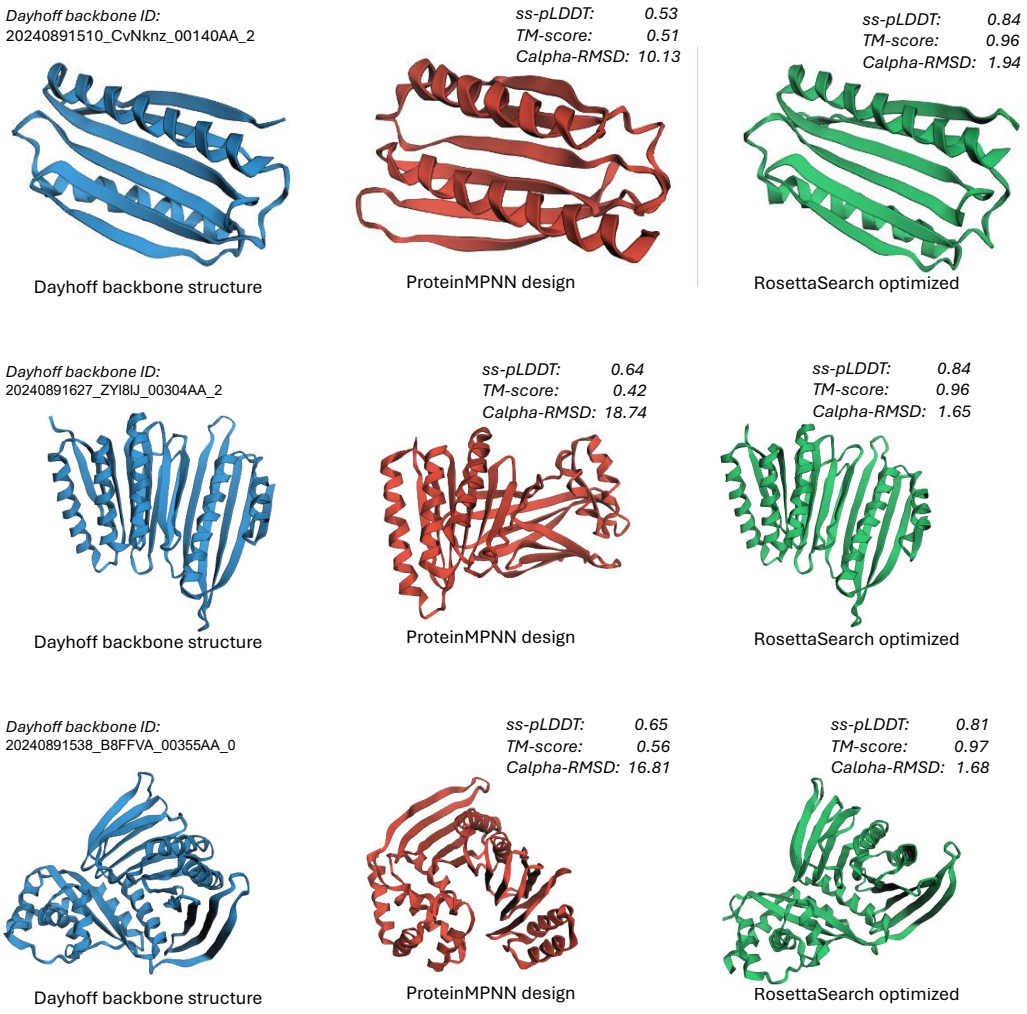


Figure 10: Examples of structures from the Dayhoff Atlas that were optimized using RosettaSearch successfully. See Table 9 for the amino-acid sequences corresponding to these structures.

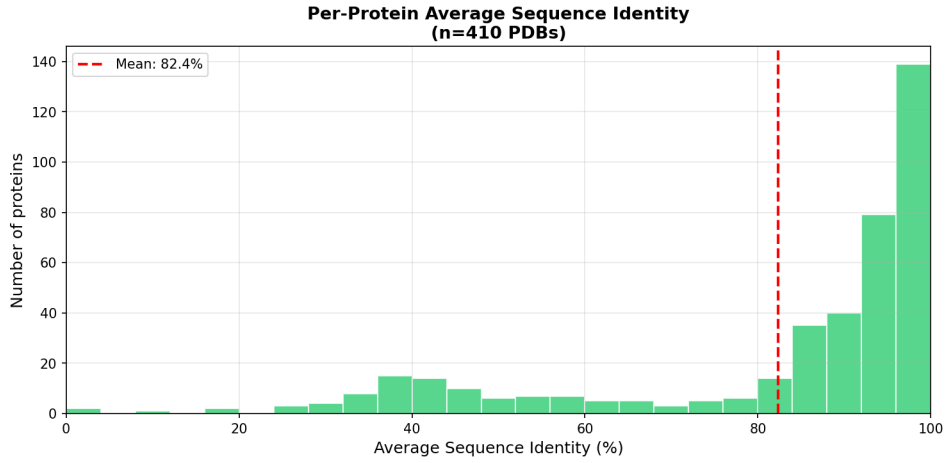


Figure 11: Distribution of the percentage sequence identity between the initial starting sequence and the generated protein sequences on the LigandMPNN dataset with 400 monomers.

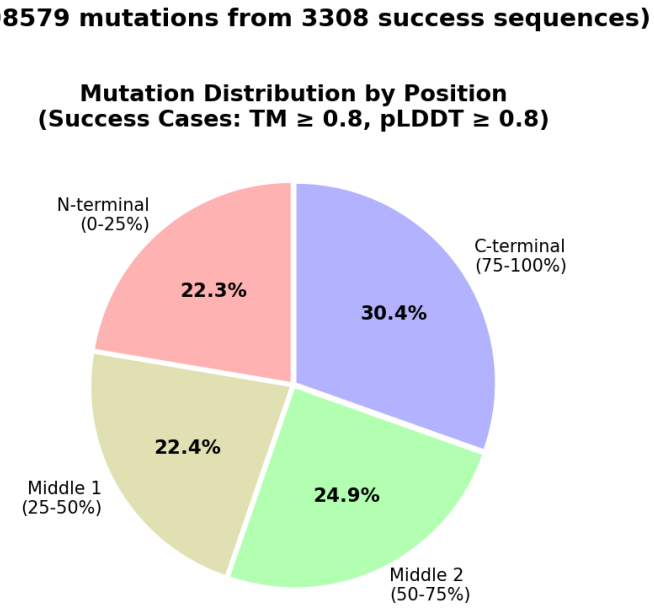


Figure 12: Distribution of positions that were updated in generated sequences as compared to the starting sequence. There were a total of 3308 generated sequences that satisfied the success criterion ((TM-score≥ 0.8) and (pLDDT≥ 0.8)) over the 400 proteins' roll-out trajectories explored with the LigandMPNN designs dataset.

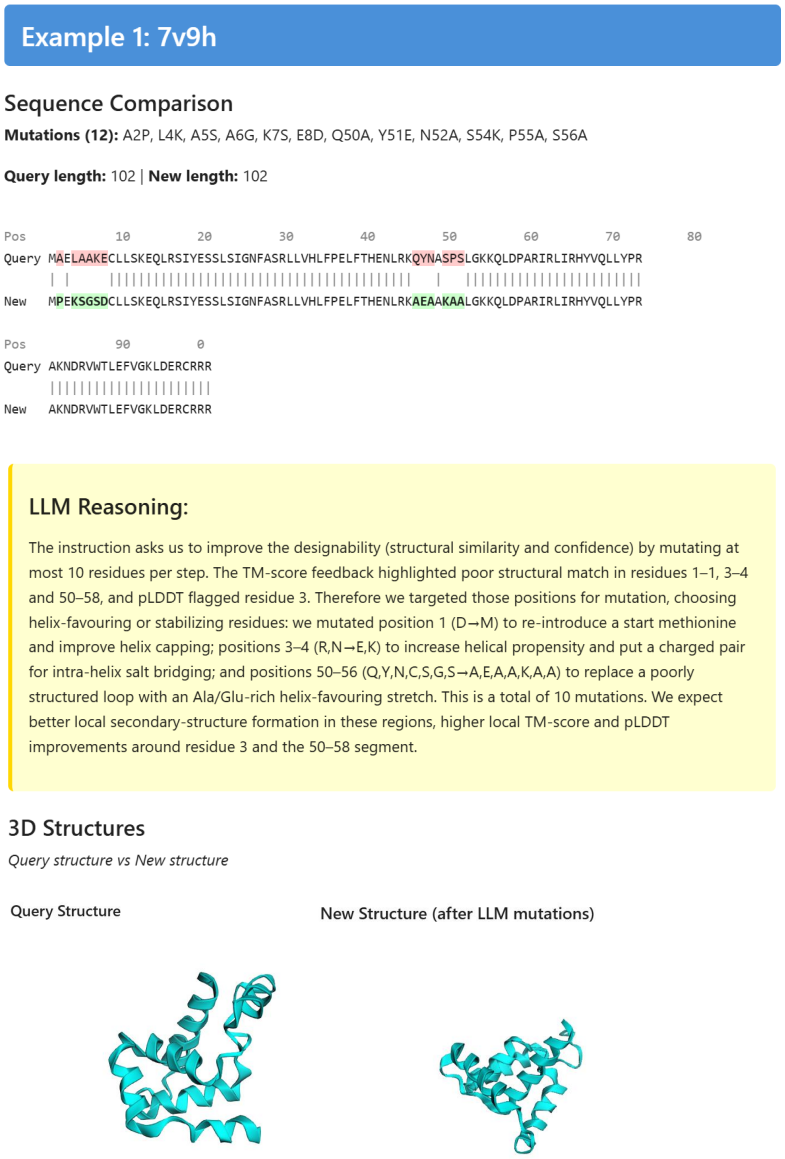


Figure 13: A screenshot of the expert annotation setup that also shows the LLM reasoning for one step of the protein sequence optimization. This is a randomly chosen example.

Figure 14: Prompt input to the LLM to describe the task of protein sequence optimization for a given input sequence

Example Optimization Prompt

```
#Instruction
We want to increase the structural fidelity of the proposed protein
sequence, by changing a maximum of 20 residues in each step.  The
constraint is to match the fold of this native protein sequence:
KSKLAKPIQDLIKMIFDVESMKKAMVEFEIDLQKMPLGKLSKRQIQSAYSILNEVQQAVSDGGSESQIL
DLSNRFYTLIPHDFGMKKPPLLSNLEYIQAKVQMLDNLLDIEVAYSLLRGGNEDGDKDPIDINYEKLRT
DIKVVDKDSEEAKIIKQYVKNTHAATHNAYDLKVVEIFRIEREGESQRYKPFKQLHNRQLLWHGSRTTN
FAGILSQGLRIAPPEAPVTGYMFGKGIYFADMVSKSANYCHTSQADPIGLILLGEVALGNMYELKNASH
ITKLPKGKHSVKGLGKTAPDPTATTTLDGVEVPLGNGISTGINDTCLLYNEYIVYDVAQVNLKYLLKLK
FNYKTS.
Use only standard amino acids (ACDEFGHIKLMNPQRSTVWY), no gaps.
Output only the raw sequence without any code.

#Variables
(str) protein_sequence= MSKLAPPIVELIKDIFDRERLEEELRRMQIDLERLPLDSLSEEMID
RALRILTEVSRVVKNGGSEEEIERLSEEFYALLPRDFGDEEPPLLDDPELIRQRVLDLDHLRRIRTTWR
LLNGGLDSKEKHPIDILYEKLRTRIEVVDPDSEEAKLIRKYVENTWGPDDNDFDIEVEEIFRIEREGAA
ERFAPYLKKENRWLLFHGTRITELAGILRDGLTLPPPEAPIEGYMFGKGIEFTSVVSKAAKNMETSREN
PIGYLLLAEVALGKMYKIKKPENIKELPEGYDSVHGIGKYAPDPKETTTLDGVKVPLGKLVETGETDTF
FKYDRFVVYDEAQVLLRYLLKLKFNFYGD

#Constraints
(str) A protein sequence of 351 amino acid residues.  Use only
standard amino acids:  ACDEFGHIKLMNPQRSTVWY. Do not include repeated
patterns or gaps or motifs.  Output ONLY the raw sequence without any
code, or formatting.  Example format:  MKTAYIAKQRQISFVK...

#Feedback TM-score feedback:  The TM-score is low, indicating poor
structural similarity.  Please make substantial improvements.  The
following residue regions have low structural similarity to the
native structure (local average deviation >5.0 Å): 1-29, 32-41,
44-44, 46-53, 63-64, 72-74, 78-125, 140-157, 161-212, 217-267,
271-286, 290-308.  Focus on improving these regions.  pLDDT feedback:
The following ranges of residues have low pLDDT (less than 35),
indicating low confidence in the predicted structure:  111:111,
121:121, 137:137, 139:139, 180:180, 196:201, 204:206, 212:214,
216:222, 224:226, 228:237, 239:239, 247:247, 251:252, 255:255,
257:257, 262:263, 265:265, 267:271, 287:288, 290:290, 292:293,
318:333, 335:335, 339:340, 345:345, 351:351.  Redundancy feedback:
This generated sequence is sufficiently non-redundant.  Low reward.
Significant improvements necessary.
```

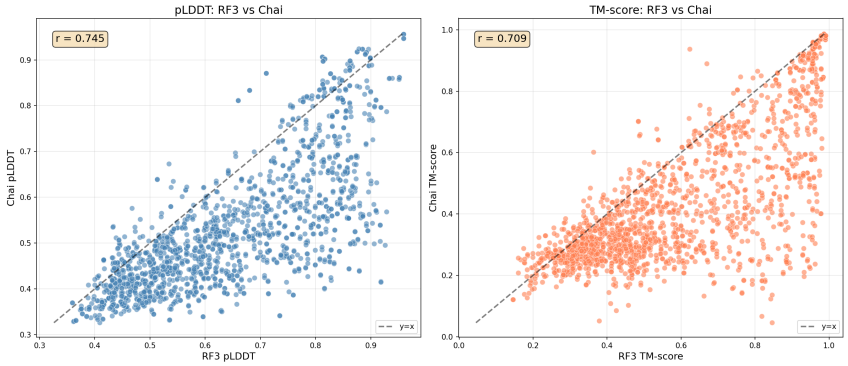


Figure 15: Comparison of the pLDDT and TM-score values of the predicted structures from RF3 and Chai-1, for sequences generated by the Search algorithm while doing multi-objective optimization. Each dot represents a protein sequence generated by RosettaSearch.