Joint protein, mRNA and DNA sequence design and optimization with nucleotide-level Potts models

Błażej Banaszewski[1,*] **Lars J. Dornfeld**[2,3,*]
Dexiong Chen[1,‡] **Karsten Borgwardt**[1,†] **Lukas F. Milles**[2,3,†]

[1]Department of Machine Learning and Systems Biology,
Max Planck Institute of Biochemistry, Martinsried, Germany
[2]Research Group Biomolecular Design,
Max Planck Institute of Biochemistry, Martinsried, Germany
[3]Gene Center and Department of Biochemistry, LMU Munich, Munich, Germany

[*]These authors contributed equally.
[†] These authors jointly supervised this work. To whom correspondence should be addressed.
{banaszewski,dornfeld,borgwardt}@biochem.mpg.de, milles@lmu.de

Abstract

Protein design typically generates amino acid sequences first and then translates them into DNA post-hoc limiting control over the nucleotide sequence via codon choice at design time. We introduce NuCaliby, a structure-conditioned model that jointly designs amino acid, mRNA, and DNA sequences of a protein at nucleotide resolution. NuCaliby predicts a Potts model over a nucleotide graph, trained using amino acid supervision by marginalizing over synonymous codons. It preserves residue-level designability on par with the original Caliby model. Its energy-based formulation supports composable inference-time guidance from sequence-level objectives through proposal construction and candidate acceptance, allowing the incorporation of biological proxies into the coding nucleotide sequence. Under organism-specific tRNA adaptation guidance, NuCaliby improves adaptation to host tRNA pools beyond synonymous sequence space, trading-off self-consistency. NuCaliby also embeds specified RNA motif sequences directly into coding sequences, avoiding the combinatorial growth required for the same task at the amino-acid level. In wet lab experiments, NuCaliby produces expressible and soluble proteins on par with state of the art methods, while enabling more complex nucleotide-level control and optimization. Together, these results extend protein design beyond the amino acid level, enabling joint optimization across the central dogma of molecular biology. The code and model weights are available here: github.com/blazejba/NuCaliby

1 Introduction

Protein design is commonly formulated as a two-stage process: backbone generation followed by sequence design through inverse folding [7, 13, 1, 12]. Existing inverse folding approaches operate at the residue level, predicting amino acid sequences that are assumed to fold into the target structure. Here, we introduce NuCaliby, a nucleotide-level inverse folding framework that jointly designs protein, mRNA, and DNA sequences under composable nucleotide-level constraints beyond synonymous space.

In nature, protein expression follows the central dogma of molecular biology [6]: a DNA sequence encoding a design is introduced into a host organism, such as *E. coli*, transcribed into mRNA, and

[‡]Now at the Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

translated into the corresponding protein sequence. This introduces an additional layer of complexity, as the mapping from codons to amino acids is degenerate, with multiple synonymous codons encoding the same residue. While this degeneracy expands the nominal sequence space, synonymous codon choices are not necessarily neutral. They influence translation efficiency, mRNA stability, and other regulatory properties, and are often adapted differently across organisms and contexts [10, 26, 43, 44].

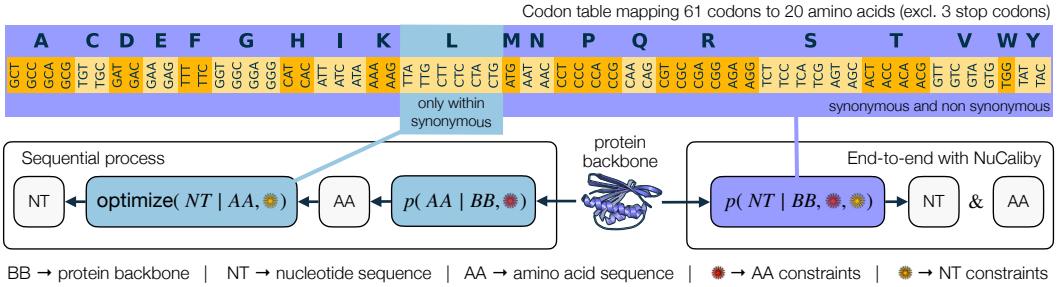

Figure 1: Conceptual differences between protein design performed sequentially on amino acid level and end-to-end on nucleotide-level. The former determines the amino acid sequence first, and optimization for DNA and RNA properties happens post hoc with optimization space limited to synonymous codons. NuCaliby enables amino acid and nucleotide level constraints to be enforced jointly during sampling, which allows design and optimization across the central dogma of life.

As a result, the set of nucleotide sequences compatible with a given protein in its biological context is heavily constrained in practice. Biological systems provide clear evidence of such constraints: for example viral genomes are under strong pressure to minimize their size and therefore frequently encode multiple functional elements within the same sequence, including regulatory RNA structures and even overlapping protein-coding sequences, by using the flexibility provided by the degenerate codons. [9].

Current protein design pipelines typically choose an amino-acid sequence first and optimize its DNA encoding afterward. Yet many different amino-acid sequences can fold into the same backbone, and each has many DNA encodings [32]. Choosing the protein sequence first therefore limits the options available for satisfying DNA- and RNA-level objectives. NuCaliby keeps both choices open, bringing sequence design across the central dogma into a single optimization process. Methods that bridge this gap, such as CodonMPNN [28], predict codons directly from backbone but inherit average codon-usage statistics from natural genes and offer no inference-time control over downstream properties such as expression or mRNA stability.

Beyond this decoupling, existing inverse folding models [7, 28, 27] share an architectural limitation: they operate on residue-level graphs whose nodes represent amino acids or codons, obscuring nucleotide-level interactions such as dependencies across codon boundaries and long-range base-pairing effects that govern mRNA secondary structure.

These limitations motivate a finer-grained modeling approach that operates directly at nucleotide resolution. We introduce NuCaliby, a structure-conditioned Potts framework that co-designs protein and DNA sequences in a single pass. NuCaliby predicts both residue-level and nucleotide-level Potts parameters from the input protein structure, enabling sampling at either level. Crucially, the energy-based formulation provides a natural mechanism for incorporating external guidance potentials at inference time, analogous to classifier guidance in diffusion models, without retraining [38].

These energy biases allow navigating the multi-objective landscape of protein and nucleotide properties at generation time. By operating at nucleotide resolution, properties that were previously fixed at reverse-translation or ignored entirely, such as codon adaptation, mRNA stability, and RNA sequence motif inclusion become explicit yet adaptable design objectives alongside protein sequence-structure compatibility.

Our main contributions are: (a) a nucleotide-level Potts model for joint protein, mRNA, and DNA sequence design, enabling simultaneous optimization across the central dogma, (b) an inference-time energy guidance framework that allows controllable multi-objective optimization, e.g. tRNA adaptation, RNA motif incorporation or improvement in calculated RNA-stability, without retraining, (c) computational and wet lab experimental evidence that nucleotide-level control can be introduced

with minimal loss in protein designability, while enabling objectives that are difficult or infeasible in synonymous sequence space.

2 Related work

Inverse folding Inverse folding models take a target backbone as input and generate sequences predicted to fold into that structure. ProteinMPNN [7], building on earlier work by John Ingraham *et al.* [12], remains the most widely used protein design method [39]. It is a graphical model based on message-passing neural networks, where residues are represented as nodes, while edges are featurized using pairwise geometric relationships derived from local protein coordinate frames, using autoregressive sampling for sequence generation. Chroma is among the first methods to use neural-network-derived Potts models for inverse folding [18, 13]. Neural Potts models represent functional energy landscapes of protein sequence alignment to a given structure. Caliby [27], an extension of neural Potts models, introduces protein-ensemble-based averaging of the Potts parameters using ensembles generated by partial diffusion with Protpardelle [5]. With this Caliby partially removes non-structural biases tied to a single backbone, and shows consistent improvements in in silico designability metrics (self-consistency and pLDDT).

Codon optimization Designed amino-acid sequences are typically reverse-translated post-hoc. DNA Chisel [44] applies rule-based constraints (codon adaptation, restriction-site avoidance, GC content); deep-learning approaches [41] map amino-acid to nucleotide sequences within the synonymous space using patterns from natural sequences. CodonMPNN [28] instead learns codon preferences end-to-end from genomic sequences paired with AlphaFold-predicted structures, conditioning on organism labels. Organism conditioning, however, requires taxonomically diverse training data that PDB alone cannot provide; the resulting reliance on AFDB structures degrades inverse-folding designability [31] (Appendix G), and natural genomic sequences are not necessarily optimized for expression or synthesizability.

Inference-time guidance for discrete sampling Classifier guidance has been extended to discrete sequence generation and applied to autoregressive inverse-folding models [38, 24]. During sampling, an auxiliary classifier biases transition probabilities toward sequences with desired properties, either by explicitly evaluating all single-mutant candidates at each step or by using a first-order Taylor approximation to adjust transition probabilities using classifier gradients. In both cases, the guiding classifier must accept partial or masked sequences, since autoregressive generation produces sequences one token at a time. Neural-Potts inverse-folding approaches such as Caliby [27] use discrete MCMC to sample complete sequences. This provides a natural setting for objectives that require fully specified sequences, motivating our use of DLMC [30] for nucleotide-level guidance.

3 Inverse folding with nucleotide-level Potts models

Problem statement. Given a protein backbone $\mathcal{X}$ over N residues, inverse folding seeks a sequence that encodes a protein folding into $\mathcal{X}$. *Amino-acid design* predicts a residue sequence $a = (a_1, \ldots, a_N) \in \mathcal{A}^N$ over the amino-acid alphabet $\mathcal{A}$ with $|\mathcal{A}| = 20$. *Nucleotide-level design* predicts a coding sequence $s = (s_1, \ldots, s_{3N}) \in \Sigma^{3N}$ over the nucleotide alphabet $\Sigma = \{A, C, G, T\}$, with amino acids recovered through the genetic code $\tau : \Sigma^3 \to \mathcal{A}$. We write $\mathrm{syn}(a) = \tau^{-1}(\{a\}) \subseteq \Sigma^3$ for the codons translating to a single amino acid a.

Classical codon optimisation first fixes an amino-acid target $a^\star \in \mathcal{A}^N$ and searches only within its synonymous codon space $\prod_{i=1}^{N} \mathrm{syn}(a_i^\star) \subseteq \Sigma^{3N}$, optimising an auxiliary objective E_{bias} subject to a hard synonymous constraint. NuCaliby instead models the full nucleotide space Σ^{3N} with a structure-conditioned Potts: amino acid sequence is encoded as a soft preference by the model rather than being enforced. Sampling can therefore move outside this synonymous space for any single $a^\star$ when this improves the trade-off between designability and E_{bias}.

A Potts model on a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ over an alphabet Ω assigns each configuration $x \in \Omega^{|\mathcal{V}|}$ the energy

$$E(x) = \sum_{v \in \mathcal{V}} h_v(x_v) + \sum_{(u,v) \in \mathcal{E}} J_{uv}(x_u, x_v), \tag{1}$$

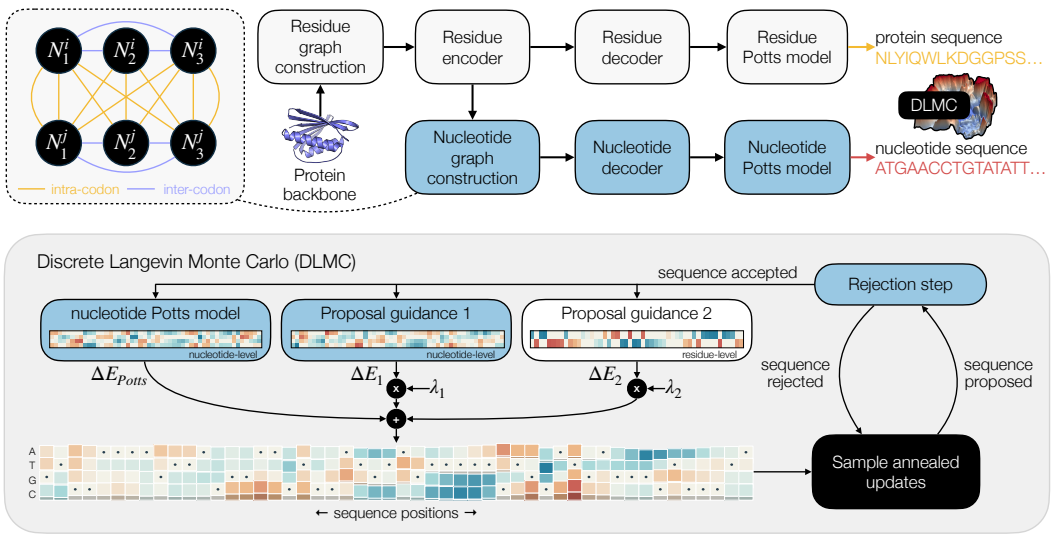

Figure 2: NuCaliby predicts the parameters of both residue- and nucleotide-level Potts models. The nucleotide graph is constructed with intra- and inter-codon edges explicitly distinguished, helping the model learn both codon-level organisation and structure-sequence relationships. DLMC explores the sequence space by inducing an energy field around the current sequence to satisfy structural compatibility, measured by the Potts model, and other objectives, which enter the optimization via proposal guidance and rejection step.

with single-site fields h_v and pairwise couplings J_{uv} predicted by the model. The induced distribution $p_{\text{Potts}}(x) \propto \exp(-E(x))$ has a partition function over $\Omega^{|\mathcal{V}|}$ that is intractable in general, so we never form it. Training and sampling rely only on site conditionals

$$p_{\text{Potts}}(x_v \mid x_{\setminus v}) \propto \exp(-E(x)), \tag{2}$$

in which the partition function cancels and the per-site normaliser sums over Ω alone. NuCaliby predicts two such models conditioned on the backbone: a residue-level Potts on $\mathcal{G}^{\text{res}}$ over $\mathcal{A}$ and a nucleotide-level Potts on the expanded $\mathcal{G}^{\text{nuc}}$ over Σ.

NuCaliby is trained as a discrete masked denoiser. At each step we sample a conditioning level $t \in [0, 1]$, the probability of retaining a residue's sequence identity in the model input. Residues are independently masked with probability $1 - t$, with at least one residue masked per example; all three nucleotide positions of a masked codon are masked together. Thus, high t corresponds to mostly visible sequence and low t to heavily masked sequence. The model reconstructs native tokens from the backbone $\mathcal{X}$ and corrupted sequence $\tilde{x}(t)$; at inference we sample from the learned Potts model with optional auxiliary guidance.

3.1 Architecture and nucleotide graph construction

NuCaliby's encoder, an AtomMPNN [27], processes the input residue graph constructed from protein structure into node and edge embeddings, which then split into two parallel decoder pathways predicting residue-level and nucleotide-level Potts parameters respectively (Figure 2).

Given a backbone with N residues, we build a residue-level k-NN graph with N nodes and Nk directed edges (including self-loops). We lift it to a nucleotide graph with $3N$ nodes and $3N(3k-1)$ directed edges. Intra-codon edges allow direct interactions among the three nucleotides encoding a residue, while inter-codon edges connect nucleotide choices across structurally neighbouring residues. This preserves codon organization while allowing dependencies between codons; Table 2 assesses the contribution of each edge family. Let $\mathbf{h}_i^{\text{res}}, \mathbf{e}_{ij}^{\text{res}}$ be the residue-level node and edge embeddings (with $i, j \in \{1, \ldots, N\}$ indexing residues), and $\tilde{s}_\mu \in \Sigma \cup \{[\text{MASK}]\}$ the (possibly masked) nucleotide

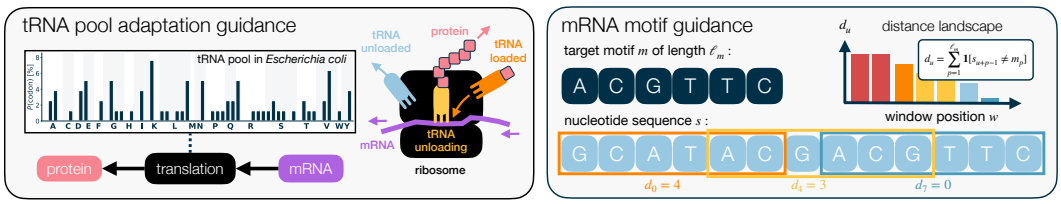

Figure 3: Auxiliary energy guidance terms that can improve tRNA pool adaptation for a given organism (left) or include an arbitrary mRNA motif into the coding sequence of a protein (right).

input at position $\mu \in \{1, \ldots, 3N\}$. Nucleotide nodes and edges are featurised as

$$\mathbf{h}^{\text{nuc}}_{i,p} = \mathbf{h}^{\text{res}}_i + \phi_{\text{pos}}(p) + \phi_{\text{nuc}}(\tilde{s}_{3i-2+p}), \tag{3}$$

$$\mathbf{e}^{\text{nuc}}_{(i,p)\to(j,p')} = \mathbf{e}^{\text{res}}_{ij} + \phi_{\text{type}}(p, p'), \tag{4}$$

where $p, p' \in \{0, 1, 2\}$ index codon positions and $\phi_{\text{pos}}, \phi_{\text{nuc}}, \phi_{\text{type}}$ are learnable embeddings; intra-codon edges ($i = j$) set $\mathbf{e}^{\text{res}}_{ii} \equiv 0$ since no residue-level edge exists.

The nucleotide decoder refines the node and edge features of the nucleotide graph, which are then projected to predict Potts fields $h^{\text{nuc}}_\mu \in \mathbb{R}^4$ at each nucleotide site $\mu \in \{1, \ldots, 3N\}$ and couplings $\mathbf{J}_{\mu\nu} \in \mathbb{R}^{4\times4}$ for all edges $(\mu, \nu) \in \mathcal{E}^{\text{nuc}}$. We use a factorized parameterization $\mathbf{J}_{\mu\nu} = \mathbf{L}_{\mu\nu}\mathbf{R}_{\mu\nu}$, where the matrices $\mathbf{L}_{\mu\nu}, \mathbf{R}_{\mu\nu} \in \mathbb{R}^{4\times4}$ are predicted from the refined edge features.

3.2 Training

We optimise a joint objective combining residue-level and nucleotide-level losses, training the entire model end-to-end from scratch. The residue-level objective provides an auxiliary supervision signal for the shared structural encoder. Joint training modestly improves nucleotide-level designability compared with NT-only training, while residue-level performance remains similar (Appendix E). Organism-specific codon preferences are not learned from genomic data; they are introduced at inference through tAI-based energy biases that proxy translation efficiency (§3.3).

Write $c_i = (s_{3i-2}, s_{3i-1}, s_{3i}) \in \Sigma^3$ for the codon at residue i. Alongside the Potts head, the decoder outputs a per-position categorical $q^{\text{dec}}_{i,p}(\cdot)$ over Σ for each codon position $p \in \{0, 1, 2\}$, giving a factorised codon distribution $\prod_{p=0}^{2} q^{\text{dec}}_{i,p}(c_p)$ for $c = (c_0, c_1, c_2)$.

At each training step we sample a synonymous nucleotide sequence $s \sim \prod_i \text{syn}(a_i)$ from the native amino-acid sequence. This s plays two roles: a corrupted copy $\tilde{s}(t)$ feeds the model as input, while the uncorrupted $s_{\setminus c_i}$ supplies the conditioning context for the Potts conditional. Because the training target specifies an amino-acid sequence rather than a preferred nucleotide encoding, supervision marginalises over the synonymous codon set instead of treating the sampled encoding as the unique correct target:

$$\mathcal{L}_{\text{nuc}} = -\frac{1}{N}\sum_{i=1}^{N}\Big[\log \sum_{c\in\text{syn}(a_i)} \prod_{p=0}^{2} q^{\text{dec}}_{i,p}(c_p) \;+\; \log \sum_{c\in\text{syn}(a_i)} p_{\text{Potts}}(c_i = c \mid s_{\setminus c_i})\Big], \tag{5}$$

where a_i is the native amino acid at residue i. This avoids explicitly supervising arbitrary codon choices, although it does not enforce equal energies across synonymous encodings. The codon-level marginal is the three-position analogue of the site conditional (2); its normaliser sums over $|\Sigma|^3 = 64$ codons and remains tractable. To prevent overfitting of the Potts loss at low masking rates, we apply a noise-dependent weight $\omega(t) = (1 - t)^\alpha$, with hyperparameter $\alpha > 0$, that down-weights the easy high-t regime, where most input residues remain visible (Appendix D.1).

3.3 Sampling and guidance

We use DLMC [30] to explore the discrete nucleotide space through energy-informed proposals. The chain maintains a complete sequence throughout sampling, allowing nucleotide-level objectives to be evaluated over the full sequence and combined with the structural energy. At inference time, DLMC computes per-site energy gaps $\Delta E_\mu(b) = E_\mu(b) - E_\mu(s_\mu)$ for all candidate states $b \in \Sigma$, where

$E_\mu(b) = h_\mu^{\text{nuc}}(b) + \sum_{\nu:(\mu,\nu)\in\mathcal{E}^{\text{nuc}}} J_{\mu\nu}(b, s_\nu)$ is the site-μ contribution to the Potts energy with the rest of s held fixed. These gaps construct local transition probabilities and let the chain efficiently explore the combinatorial sequence space while respecting the Potts energy landscape.

NuCaliby incorporates sequence-level objectives during sampling through proposal guidance and / or acceptance guidance. Proposal guidance uses local energy changes, or approximations to them, to favor candidate nucleotide substitutions. Acceptance guidance evaluates objectives on complete current and proposed sequences without requiring local substitution scores. The choice depends on the information available from an objective and the cost of evaluating it (Figure 2).

For proposal guidance, let $x = \text{onehot}(s)$ and let $F(x)$ be a continuous extension of an auxiliary sequence energy. We use its derivatives at the current sequence to augment the Potts energy gaps:

$$\Delta\widetilde{E}_\mu(b) = \Delta E_\mu(b) + \lambda \left[\frac{\partial F}{\partial x_{\mu b}} - \frac{\partial F}{\partial x_{\mu s_\mu}} \right]_{x=\text{onehot}(s)} . \tag{6}$$

Here, λ controls guidance strength. The derivative difference gives a local linear estimate of the auxiliary energy change when replacing s_μ with b. For tAI and stop-codon energies, this equals the exact single-nucleotide energy change with all other positions fixed. If gradients are used, they are computed directly with respect to the one-hot representation.

We demonstrate the mechanism with the following composable energy biases:

tAI guidance. The tRNA Adaptation Index (tAI) [8] estimates how well a codon matches the host organism's tRNA pool by considering the tRNA gene copies while correcting for non-standard tRNA pairing at the third codon position, called wobble correction. It is commonly used as a proxy for adaptation of coding sequences to the host tRNA pool and correlates with translational efficiency and protein expression [21, 2]. We use the length-normalized energy

$$E_{\text{tAI}}(s) = -\frac{1}{N} \sum_{i=1}^{N} \log w_{\text{tAI}}(c_i), \tag{7}$$

weighted by λ during guidance. The implementation assigns neutral log weights to start and stop codons, with stops handled by a separate penalty. Expressing each codon's contribution as a multilinear function of its nucleotide one-hot entries allows exact single-nucleotide energy differences to be obtained through the derivative differences in Eq. 6.

mRNA sequence motif guidance. A compelling application of nucleotide-level design is embedding functional RNA motifs such as frame-shifting elements directly into the coding sequence. When the motif has no preferred position in the coding sequence, we use a sliding-window energy bias that lets the sampler find the best location (Figure 3).

For a target motif $m = (m_1, \ldots, m_{\ell_m})$ and designed sequence s, the Hamming distance at window position $\mu \in \{1, \ldots, 3N - \ell_m + 1\}$ is $d_\mu = \sum_{p=1}^{\ell_m} \mathbf{1}[s_{\mu+p-1} \neq m_p]$ represented in one-hot coordinates as $d_\mu(x) = \sum_{p=1}^{\ell_m} (1 - x_{\mu+p-1,m_p})$. We aggregate over all windows using a smooth-min:

$$E_{\text{motif}} = -\frac{\lambda}{\beta} \log \sum_{\mu} \exp(-\beta\, d_\mu), \tag{8}$$

where β controls the sharpness ($\beta \to \infty$ recovers the hard min over windows; smaller β flattens the weighting across windows) and λ controls the weight relative to the Potts energy. We differentiate this smooth-min energy with respect to the current one-hot sequence and use derivative differences to guide nucleotide proposals. These provide local linear approximations to changes in the motif energy. The motif energy already includes its guidance weight λ, which is applied only once. Multiple motifs can be placed at once by summing their individual E_{motif} terms, requiring no additional sampling runs.

Acceptance guidance evaluates auxiliary objectives on complete current and proposed sequences, without requiring local substitution scores. When acceptance/rejection is enabled, the proposed sequence s' is accepted under the Metropolis-Hasting criterion, with probability

$$A(s \to s') = \min\left(1,\, \frac{q_T(s \mid s')}{q_T(s' \mid s)} \exp\left[-\frac{U(s') - U(s)}{T}\right]\right), \tag{9}$$

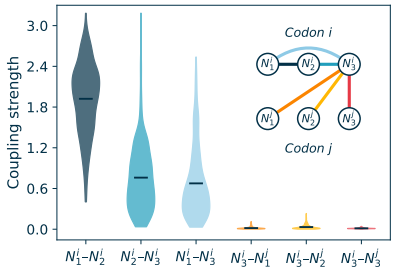

Figure 4: Codon-level organization in the Potts model arises from a triangle of within-codon couplings.

	AA ↑	pLDDT ↑	TM ↑	RMSD ↓
NATIVE	100	40.9 ±14.8	.372 ±.166	5.21 ±1.54
PROTEINMPNN	44.4 ±7.7	54.3 ±18.7	.509 ±.233	4.37 ±1.81
CODONMPNN	44.5 ±9.0	47.0 ±17.1	.441 ±.211	4.85 ±1.75
CALIBY	39.2 ±7.2	61.7 ±19.4	.597 ±.248	3.85 ±1.88
$\text{CALIBY}_{\text{ENS}}$	28.5 ±5.9	63.8 ±16.8	.591 ±.227	3.82 ±1.71
$\text{NUCALIBY}^{\text{AA}}$	39.6 ±7.2	61.8 ±19.6	.593 ±.245	3.84 ±1.85
$\text{NUCALIBY}^{\text{AA}}_{\text{ENS}}$	29.2 ±5.3	67.7 ±17.2	.659 ±.229	3.48 ±1.75
$\text{NUCALIBY}^{\text{NT}}$	38.6 ±7.1	60.8 ±19.1	.579 ±.245	3.85 ±1.82
$\text{NUCALIBY}^{\text{NT}}_{\text{ENS}}$	27.2 ±6.0	63.6 ±16.3	.580 ±.228	3.92 ±1.64

Table 1: Fraction of recovered amino acids (*AA %*), prediction confidence (*pLDDT %*), and self-consistency TM-score and RMSD in Å.

where T is the current sampling temperature, q_T is the proposal probability, and U includes the Potts energy and all weighted auxiliary energies used for acceptance. Auxiliary energies used to construct proposals also contribute to U; additional objectives can enter U without altering the proposal.

Our experiments use annealed DLMC without acceptance/rejection except when applying ViennaRNA acceptance guidance (Appendix A.1).

mRNA secondary structure guidance. We use the ViennaRNA [19] ensemble free energy per nucleotide as an acceptance-only objective. This favors sequences with more favorable predicted mRNA folding energies, while proposals remain guided by the Potts energy and stop-codon penalty. Only complete-sequence energy evaluations are required.

4 Experiments

Our experiments establish one claim: NuCaliby enables joint optimization across the central dogma (protein, mRNA, and DNA) with explicit control over the trade-offs between protein designability and nucleotide-level objectives. Each experiment isolates a separate facet of this claim. **Codon-level organization** (§4.1) tests whether the Potts model recovers codon structure from purely pairwise nucleotide couplings, despite operating at the level of individual nucleotides. **Designability** (§4.1) shows that nucleotide-level operation does not reduce *in silico* self-consistency of the designed proteins. **tRNA adaptation** (§4.2) illustrates a controllable trade-off between a nucleotide objective and structural compatibility, including comparison with synonymous optimization. **RNA motif inclusion** (§4.3) tests direct enforcement of sequence constraints whose compatible amino-acid patterns grow combinatorially in a motif-aware residue-level formulation. **Predicted RNA folding energy** (§4.4) demonstrates how a complete-sequence evaluator can guide acceptance without supplying local substitution scores. Finally, **wet-lab validation** (§4.5) assesses whether guided designs, including motif-containing sequences, remain expressible and soluble.

Experiments use 183 soluble monomer backbones from the Caliby test set (Appendix A.6). NuCaliby is trained on ∼231k PDB structures from the Boltz-1 [37] training set using a temporal split, clustered at 40% sequence identity with MMseqs2 [29] and excluding transmembrane proteins. We compare against ProteinMPNN, CodonMPNN, and Caliby. Since NuCaliby learns to predict the weights of both a residue-level (AA) and a nucleotide-level (NT) Potts model, we evaluate sampling from each independently. For Caliby and NuCaliby, we additionally report ensemble-conditioned results (ENS) with Potts parameters aggregated across the original structure plus 15 variants generated by Protpardelle-1c [5] via partial diffusion. See Appendix A.1, Table 3 for guidance settings.

4.1 Codon-level organization and protein designability

The Potts model includes only pairwise (2-body) interactions, while codons involve constraints over nucleotide triplets. We ask whether codon structure is captured through pairwise couplings. We measure interaction strength between NT sites μ and ν as the Frobenius norm of the coupling matrix, $\|\mathbf{J}_{\mu\nu}\|_F$.

For pairs with the same sequence distance, coupling strength is higher for codon-aligned pairs than for frame-shifted pairs. This suggests that the model organizes pairwise interactions into codon-aligned triangles. Within codons, couplings between the first and second nucleotide positions are strongest, while those involving the third position are weaker, consistent with the wobble position (Figure 4).

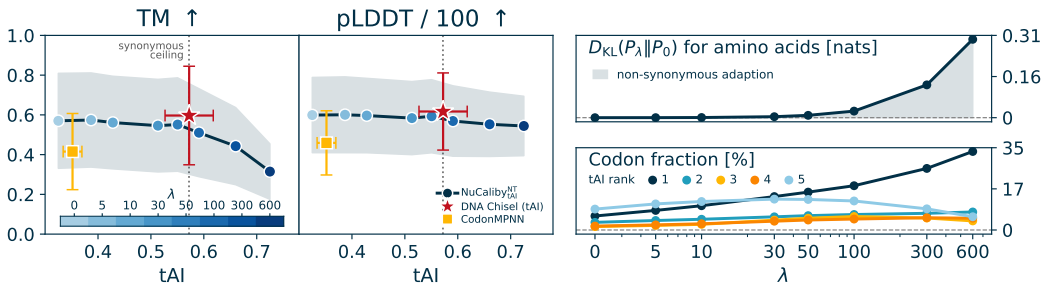

Figure 5: tRNA adaptation index (tAI) and in-silico designability for NuCaliby with increasing tAI guidance strength λ, compared with CodonMPNN and Caliby + DNA Chisel with synonymous tAI optimization. All methods are scored using the same BL21(DE3) tAI weights. Right: amino acid KL divergence relative to unbiased sampling and codon-rank frequencies as a function of λ.

Designability is assessed via AlphaFold2 self-consistency (Table 1; see Appendix A for protocol). The baselines include folding with single-sequence AF2 for the (NATIVE) sequence, ProteinMPNN, CodonMPNN and Caliby. Structure prediction for NATIVE is challenging compared to re-designed sequences, which is a well described limitation of the current in-silico evaluation strategies [16]. Sequence recovery measures similarity to the native sequence, whereas inverse-folding design seeks sequences compatible with the target backbone. Lower recovery therefore does not by itself indicate poorer design, and we interpret it alongside structural self-consistency and experimental validation.

NuCaliby matches Caliby on sequence recovery and designability while additionally enabling nucleotide-level design. Both sampling from the residue-level Potts (NuCaliby^{AA}), and the nucleotide-level Potts (NuCaliby^{NT}), preserve inverse-folding performance, and ensemble conditioning further improves designability. CodonMPNN, by contrast, retains ProteinMPNN-level recovery but pays a designability cost consistent with training on predicted structures (Appendix G). For evaluation on *de novo* proteins see Appendix F.

Ablations. We assess the contributions of nucleotide-graph connectivity and joint amino-acid/nucleotide training (Table 2). Removing either intra-codon or inter-codon edges reduces nucleotide-level designability, supporting the inclusion of both interaction types. Training the nucleotide branch alone produces a smaller decrease in performance, indicating that joint supervision benefits nucleotide-level design. Full results and experimental details are provided in Appendix E.

NT variant	pLDDT ↑	TM ↑	RMSD ↓
FULL MODEL	60.8 $\pm$19.1	.579 $\pm$.245	3.85 $\pm$1.82
NO INTRA-CODON	55.1 $\pm$18.2	.504 $\pm$.225	4.40 $\pm$1.72
NO INTER-CODON	54.1 $\pm$17.7	.500 $\pm$.223	4.49 $\pm$1.80
NT-ONLY TRAINING	58.9 $\pm$18.5	.548 $\pm$.227	4.15 $\pm$1.73

Table 2: Ablations of nucleotide-level designability. Mean $\pm$ standard deviation across 183 test proteins; RMSD in Å.

4.2 Adaptation to organism-specific tRNA pools

We evaluate sequences generated by CodonMPNN, Caliby + DNA Chisel and NuCaliby^{NT} using the same *E. coli* BL21(DE3) tRNA adaptation index (tAI) [8]. NuCaliby$^{NT}_{tAI}$ applies these tAI weights as an inference-time energy bias with strength λ, while DNA Chisel optimizes tAI within the synonymous encodings of Caliby-generated amino-acid sequences. CodonMPNN uses species-level *E. coli* conditioning because BL21(DE3) is sparsely represented in its training data (Appendix A.3).

Figure 5 (left) shows the trade-off between tAI and designability. Synonymous tAI optimization with Caliby + DNA Chisel achieves a mean tAI of 0.572. NuCaliby can extend this trade-off toward higher tAI under stronger guidance, reaching 0.660 at $\lambda = 300$ and 0.725 at $\lambda = 600$, at an increasing cost to self-consistency. pLDDT remains comparatively stable, while decreasing TM-scores indicate structural drift from the input backbone. Thus, synonymous optimization already provides substantial tAI gains for a fixed amino-acid sequence, while joint amino-acid and nucleotide design provides access to a broader trade-off.

Figure 5(right) measures the KL divergence between the amino-acid distribution under bias strength P_λ and the unbiased distribution P_0. Up to $\lambda = 30$, amino-acid composition changes little, although this aggregate statistic does not establish that individual substitutions are synonymous. Stronger guidance shifts amino-acid composition and increases the frequency of rank-1 codons as self-

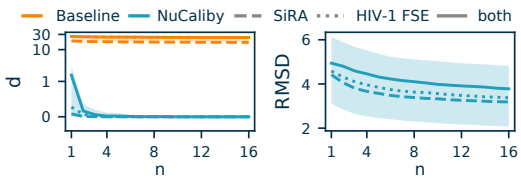

Figure 6: Best-of-n motif inclusion on 183 test monomers, for SiRA aptamer, HIV-1 FSE and both simultaneously. d: minimum Hamming distance to the motif. pLDDT and TM are zero when a sample has $d{>}0$.

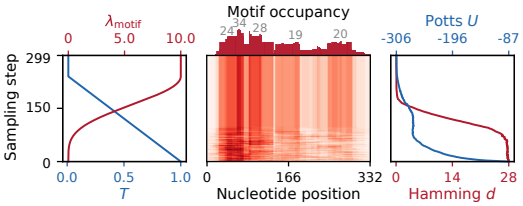

Figure 7: Heatmap of motif occupancy during sampling. Position of the best-matching SiRA aptamer window across 100 independent sampling trajectories where 93/100 runs converge to $d{=}0$.

consistency declines. These results are consistent with exploiting non-synonymous degrees of freedom to improve tAI beyond the optimized Caliby baseline. The baseline value of 0.572 is specific to those amino-acid sequences, rather than a universal limit on synonymous optimization.

4.3 Inclusion of mRNA sequence motif

We demonstrate motif guidance by embedding two RNA elements into designed coding sequences: SiRA aptamer [36] and HIV-1 frameshifting element (FSE) [14, 23], explained in Appendix A.4.

To construct a residue-level baseline, we translate the RNA motif into a set of candidate amino-acid patterns. The number of codon variants is bounded by $\ell_m \bmod 3$ across reading frames; stop-codon-containing variants are excluded, and the remainder translate via τ (see problem statement in §3) to a smaller set of amino-acid sequences (Figure 10). The SiRA aptamer and HIV-1 FSE yield 264 and 129 codon variants that collapse to 68 and 31 amino-acid patterns respectively. The baseline samples sequences from Caliby and checks whether any contains one of these patterns. NuCaliby$^{\text{NT}}_{\text{motif}}$ instead applies the motif energy bias (Eq. (8)) at $\lambda{=}50$, $\beta{=}2$ directly at the nucleotide level, with a stop-codon penalty (weight 100) to prevent internal stops.

Figure 6 shows that residue-level sampling does not solve the problem: none of the 183 test proteins contain the motif by chance at $n{=}16$. NuCaliby reaches $d{=}0$ on all 183 proteins by $n{=}2$ (single motif) and $n{=}16$ (two motifs); designability of motif-embedded designs improves with n.

Figure 7 traces how the embedded motif emerges during a single sampling run: the position of its best-matching window narrows progressively as the temperature anneals and the motif weight ramps up, with most independent trajectories converging to the same location and to an exact match. Sensitivity analyses across guidance strengths, sampling temperatures, sampling budgets, and motif choices are reported in Appendix A.5.

4.4 mRNA folding energy optimization

We use the ViennaRNA [19] partition-function ensemble free energy per nucleotide as an acceptance-only objective (Eq. 9). As shown in Figure 8, the designs migrate toward more-negative free energy. At $\lambda{=}10$, designs for 182 of 183 backbones have lower predicted ensemble free energy per nucleotide than the unguided baseline, with a mean shift of $-0.07\,\text{kcal}/(\text{mol}\times\text{nt})$. This demonstrates that complete-sequence energy evaluations can guide design without supplying local substitution scores.

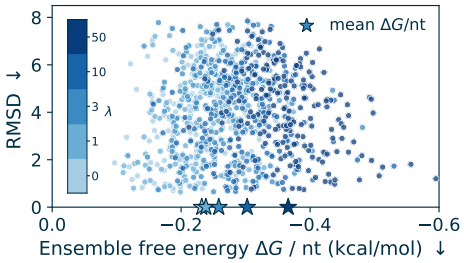

Figure 8: Self-consistency RMSD versus predicted mRNA ensemble free energy per nucleotide. Guidance shifts designs toward lower predicted RNA folding energies while preserving comparable self-consistency.

4.5 *In vitro* wet lab validation

We experimentally evaluated three capabilities of NuCaliby in vitro: (i) whether it can directly generate expressible DNA sequences without post-hoc codon optimization, (ii) whether inference-time tAI guidance improves protein yield, and (iii) whether RNA sequence motifs can be embedded into coding sequences while retaining expression.

Eight structurally diverse backbones were tested, with 24 sequences sampled per backbone before filtering. Unless excluded due to synthesis constraints, two designs per method and backbone were evaluated for expression and monodispersity using size-exclusion chromatography traces.

Overall, the experiments provide an initial indication that NuCaliby can generate soluble and expressible designs across unbiased, tAI-guided, and multi-objective tAI / RNA-motif-conditioned settings (Figure 9). For moderate tAI guidance $\lambda = 30$, soluble yield increased across all eight backbones. After averaging the two designs per backbone, the median yield increased by approximately 24%, from 0.042 to 0.052 mg/mL. This improvement was significant in an exact one-sided Wilcoxon signed-rank test (n = 8 paired backbones, W = 36, p = 0.0039). Motif-conditioned designs remained expressible in several cases, indicating that RNA-level constraints can be incorporated into coding sequences without fully abolishing protein production. CodonMPNN was overall low in yield and additionally suffered from synthesizability issues, which reduced the number of experimentally testable constructs (See Appendix B for more details).

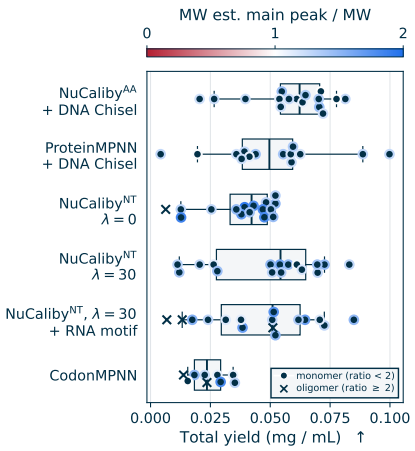

Figure 9: Experimental protein expression. Yields estimated from SEC traces. λ indicates tAI guidance strength. X indicates higher-order oligomeric state.

5 Conclusion

NuCaliby is a structure-conditioned nucleotide-level Potts framework that jointly optimizes amino-acid identity and nucleotide coding sequence while maintaining structural consistency. Its energy-based formulation incorporates external constraints at both levels during design, allowing us to realize nucleotide objectives that are infeasible to achieve through synonymous codon choice alone.

Synonymous marginalization encourages approximate invariance to codon choice, while inference-time guidance enables explicit control over nucleotide-level objectives. Guidance can shape nucleotide proposals using local objective information or assess complete candidate sequences through acceptance, allowing multiple sequence-level objectives to be combined during design.

We show that nucleotide-level design embeds RNA sequence motifs directly into coding sequences and adapts codon usage to host tRNA pools, while retaining good self-consistency. Post-hoc codon optimization restricted to synonymous substitutions cannot satisfy these constraints: identifying potentially compatible amino-acid patterns requires combinatorial enumeration that scales poorly. ViennaRNA guidance lowers predicted mRNA ensemble free energy while preserving comparable self-consistency. Wet lab experiments on a diverse set of de novo protein backbones demonstrate that NuCaliby achieves performance on par with state of the art protein design methods like ProteinMPNN[7] producing expressible and soluble proteins, while performing a more complex optimization task.

Future directions include finer-grained guidance and a broader range of RNA elements. Natural coding sequences exhibit position-dependent behavior, such as the low-tAI ramp at the 5' end [33]; targeting such position-specific tAI profiles, rather than globally maximizing tAI, and incorporating proxies for RNA secondary structure formation, would allow a more biologically nuanced control of translation and cotranslational protein folding [20]. We also plan to incorporate a broader range of regulatory and functional RNA elements directly into coding sequences: programmed ribosomal frameshift motifs or translational stalling signals could be embedded to modulate protein folding or generate multiple protein products from a single transcript. Although we primarily demonstrate our approach on de novo design, the same framework applies to existing proteins, with limited amino acid substitutions to satisfy nucleotide-level properties.

Nucleotide-level design combines constraints along the central dogma of molecular biology. As a proof-of-concept, NuCaliby points toward protein design as a multi-scale optimization problem spanning amino-acid structure, nucleotide sequence, and cellular context.

6 Acknowledgments

The authors thank Michael A. Jendrusch for helpful discussions, and Nele Dierlamm and Emma Weißert for help with experiments. L.J.D. and L.F.M. are grateful for funding by the Deutsche Forschungsgemeinschaft (DFG) – Project number 529882161. This work was funded under Germany's Excellence Strategy – EXC3092 – 533751719.

References

[1] Deniz Akpinaroglu, Kosuke Seki, Amy Guo, Eleanor Zhu, Mark J. S. Kelly, and Tanja Kortemme. Structure-conditioned masked language models for protein sequence design generalize beyond the native sequence space, December 2023.

[2] Shir Bahiri-Elitzur and Tamir Tuller. Codon-based indices for modeling gene expression and transcript evolution. *Computational and Structural Biotechnology Journal*, 19:2646–2663, January 2021. ISSN 2001-0370. doi: 10.1016/j.csbj.2021.04.042.

[3] Foster Birnbaum and Amy E. Keating. Beyond native sequence recovery: Improved modeling of the sequence-energy landscape of protein structures. *Proceedings of the National Academy of Sciences*, 123(27):e2535494123, July 2026. doi: 10.1073/pnas.2535494123.

[4] Marta Chronowska, Michael J. Stam, Derek N. Woolfson, Luigi F. Di Costanzo, and Christopher W. Wood. The Protein Design Archive (PDA): insights from 40 years of protein design. *Nature Biotechnology*, 43(5):669–671, 3 2025. doi: 10.1038/s41587-025-02607-x. URL `https://doi.org/10.1038/s41587-025-02607-x`.

[5] Alexander E. Chu, Jinho Kim, Lucy Cheng, Gina El Nesr, Minkai Xu, Richard W. Shuai, and Po-Ssu Huang. An all-atom protein generative model. *Proceedings of the National Academy of Sciences*, 121(27):e2311500121, July 2024. doi: 10.1073/pnas.2311500121.

[6] Francis Crick. Central dogma of molecular Biology. *Nature*, 227(5258):561–563, 8 1970. doi: 10.1038/227561a0. URL `https://doi.org/10.1038/227561a0`.

[7] 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.

[8] Mario dos Reis, Renos Savva, and Lorenz Wernisch. Solving the riddle of codon usage preferences: a test for translational selection. *Nucleic Acids Research*, 32(17):5036–5044, 2004. doi: 10.1093/nar/gkh834.

[9] Andrew E. Firth. Mapping overlapping functional elements embedded within the protein-coding regions of RNA viruses. *Nucleic Acids Research*, 42(20):12425–12439, 10 2014. doi: 10.1093/nar/gku981. URL `https://doi.org/10.1093/nar/gku981`.

[10] Alexander Goncearenco and Igor N Berezovsky. The fundamental tradeoff in genomes and proteomes of prokaryotes established by the genetic code, codon entropy, and physics of nucleic acids and proteins. *Biology Direct*, 9(1), 12 2014. doi: 10.1186/s13062-014-0029-2. URL `https://doi.org/10.1186/s13062-014-0029-2`.

[11] Casper A. Goverde, Martin Pacesa, Nicolas Goldbach, Lars J. Dornfeld, Petra E. M. Balbi, Sandrine Georgeon, Stéphane Rosset, Srajan Kapoor, Jagrity Choudhury, Justas Dauparas, Christian Schellhaas, Simon Kozlov, David Baker, Sergey Ovchinnikov, Alex J. Vecchio, and Bruno E. Correia. Computational design of soluble and functional membrane protein analogues. *Nature*, 631(8020):449–458, July 2024. ISSN 1476-4687. doi: 10.1038/s41586-024-07601-y.

[12] John Ingraham, Vikas Garg, Regina Barzilay, and Tommi Jaakkola. Generative models for graph-based protein design. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[13] John B. Ingraham, Max Baranov, Zak Costello, Karl W. Barber, Wujie Wang, Ahmed Ismail, Vincent Frappier, Dana M. Lord, Christopher Ng-Thow-Hing, Erik R. Van Vlack, Shan Tie, Vincent Xue, Sarah C. Cowles, Alan Leung, João V. Rodrigues, Claudio L. Morales-Perez, Alex M. Ayoub, Robin Green, Katherine Puentes, Frank Oplinger, Nishant V. Panwar, Fritz Obermeyer, Adam R. Root, Andrew L. Beam, Frank J. Poelwijk, and Gevorg Grigoryan. Illuminating protein space with a programmable generative model. *Nature*, 623(7989):1070–1078, November 2023. ISSN 1476-4687. doi: 10.1038/s41586-023-06728-8.

[14] Tyler Jacks, Hiten D. Madhani, Frank R. Masiarz, and Harold E. Varmus. Signals for ribosomal frameshifting in the rous sarcoma virus gag-pol region. *Cell*, 55(3):447–458, 11 1988. doi: 10.1016/0092-8674(88)90031-1. URL https://doi.org/10.1016/0092-8674(88)90031-1.

[15] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, August 2021. ISSN 1476-4687. doi: 10.1038/s41586-021-03819-2.

[16] Kerlen T. Korbeld, Vsevolod Viliuga, and Maximilian Fürst. Limitations of the refolding pipeline for *de novo* protein design, December 2025.

[17] Grzegorz Kudla, Andrew W. Murray, David Tollervey, and Joshua B. Plotkin. Coding-sequence determinants of gene expression in ¡i¿escherichia coli¡/i¿. *Science*, 324(5924):255–258, 2009. doi: 10.1126/science.1170160. URL https://www.science.org/doi/abs/10.1126/science.1170160.

[18] Alex J. Li, Mindren Lu, Israel Desta, Vikram Sundar, Gevorg Grigoryan, and Amy E. Keating. Neural network-derived Potts models for structure-based protein design using backbone atomic coordinates and tertiary motifs. *Protein Science*, 32(2):e4554, 12 2022. doi: 10.1002/pro.4554. URL https://doi.org/10.1002/pro.4554.

[19] Ronny Lorenz, Stephan H Bernhart, Christian Höner Zu Siederdissen, Hakim Tafer, Christoph Flamm, Peter F Stadler, and Ivo L Hofacker. ViennaRNA Package 2.0. *Algorithms for Molecular Biology*, 6(1):26, 11 2011. doi: 10.1186/1748-7188-6-26. URL https://doi.org/10.1186/1748-7188-6-26.

[20] Saurav Mallik, Johannes Venezian, Arseniy Lobov, Meta Heidenreich, Hector Garcia-Seisdedos, Todd O. Yeates, Ayala Shiber, and Emmanuel D. Levy. Structural determinants of co-translational protein complex assembly. *Cell*, 188(3):764–777.e22, February 2025. ISSN 00928674. doi: 10.1016/j.cell.2024.11.013.

[21] Orna Man, Joel L. Sussman, and Yitzhak Pilpel. Examination of the tRNA Adaptation Index as a Predictor of Protein Expression Levels. In Eleazar Eskin, Trey Ideker, Ben Raphael, and Christopher Workman, editors, *Systems Biology and Regulatory Genomics*, pages 107–118, Berlin, Heidelberg, 2006. Springer. ISBN 978-3-540-48540-7. doi: 10.1007/978-3-540-48540-7_10.

[22] Milot Mirdita, Konstantin Schütze, Yoshitaka Moriwaki, Lim Heo, Sergey Ovchinnikov, and Martin Steinegger. ColabFold: Making protein folding accessible to all. *Nature Methods*, 19(6):679–682, June 2022. ISSN 1548-7105. doi: 10.1038/s41592-022-01488-1.

[23] Kathryn D. Mouzakis, Andrew L. Lang, Kirk A. Vander Meulen, Preston D. Easterday, and Samuel E. Butcher. HIV-1 frameshift efficiency is primarily determined by the stability of base pairs positioned at the mRNA entrance channel of the ribosome. *Nucleic Acids Research*, 41(3):1901–1913, 12 2012. doi: 10.1093/nar/gks1254. URL https://doi.org/10.1093/nar/gks1254.

[24] Hunter Nisonoff, Junhao Xiong, Stephan Allenspach, and Jennifer Listgarten. Unlocking Guidance for Discrete State-Space Diffusion and Flow Models, March 2025.

[25] Jason Qian, Lukas F. Milles, Basile I. M. Wicky, Amir Motmaen, Xinting Li, Ryan D. Kibler, Lance Stewart, and David Baker. Accelerating protein design by scaling experimental characterization. *bioRxiv*, 2025. doi: 10.1101/2025.08.05.668824. URL https://www.biorxiv.org/content/early/2025/08/06/2025.08.05.668824.

[26] Xukang Shen, Siliang Song, Chuan Li, and Jianzhi Zhang. Synonymous mutations in representative yeast genes are mostly strongly non-neutral. *Nature*, 606(7915):725–731, 2022. ISSN 1476-4687. doi: 10.1038/s41586-022-04823-w. URL https://www.nature.com/articles/s41586-022-04823-w.

[27] Richard W. Shuai, Tianyu Lu, Subhang Bhatti, Petr Kouba, and Po-Ssu Huang. Ensemble-conditioned protein sequence design with Caliby, October 2025. URL http://biorxiv.org/lookup/doi/10.1101/2025.09.30.679633.

[28] Hannes Stark, Umesh Padia, Julia Balla, Cameron Diao, and George Church. CodonMPNN for Organism Specific and Codon Optimal Inverse Folding, 2024.

[29] Martin Steinegger and Johannes Söding. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature Biotechnology*, 35(11):1026–1028, November 2017. ISSN 1546-1696. doi: 10.1038/nbt.3988.

[30] Haoran Sun, Hanjun Dai, Bo Dai, Haomin Zhou, and Dale Schuurmans. Discrete Langevin Sampler via Wasserstein Gradient Flow, 2022.

[31] Cheng Tan, Zhenxiao Cao, Zhangyang Gao, Siyuan Li, Yufei Huang, and Stan Z. Li. Alphafold database debiasing for robust inverse folding. *arXiv preprint arXiv:2506.08365*, 2025.

[32] Pengfei Tian and Robert B. Best. How Many Protein Sequences Fold to a Given Structure? A Coevolutionary Analysis. *Biophysical Journal*, 113(8):1719–1730, October 2017. ISSN 0006-3495. doi: 10.1016/j.bpj.2017.08.039.

[33] Tamir Tuller, Asaf Carmi, Kalin Vestsigian, Sivan Navon, Yuval Dorfan, John Zaborske, Tao Pan, Orna Dahan, Itay Furman, and Yitzhak Pilpel. An evolutionarily conserved mechanism for controlling the efficiency of protein translation. *Cell*, 141(2):344–354, 4 2010. doi: 10.1016/j.cell.2010.03.031. URL https://doi.org/10.1016/j.cell.2010.03.031.

[34] Michel van Kempen, Stephanie S. Kim, Charlotte Tumescheit, Milot Mirdita, Jeongjae Lee, Cameron L. M. Gilchrist, Johannes Söding, and Martin Steinegger. Fast and accurate protein structure search with Foldseek. *Nature Biotechnology*, 42(2):243–246, February 2024. ISSN 1546-1696. doi: 10.1038/s41587-023-01773-0.

[35] Joseph L. Watson, David Juergens, Nathaniel R. Bennett, Brian L. Trippe, Jason Yim, Helen E. Eisenach, Woody Ahern, Andrew J. Borst, Robert J. Ragotte, Lukas F. Milles, Basile I. M. Wicky, Nikita Hanikel, Samuel J. Pellock, Alexis Courbet, William Sheffler, Jue Wang, Preetham Venkatesh, Isaac Sappington, Susana Vázquez Torres, Anna Lauko, Valentin De Bortoli, Emile Mathieu, Sergey Ovchinnikov, Regina Barzilay, Tommi S. Jaakkola, Frank DiMaio, Minkyung Baek, and David Baker. De novo design of protein structure and function with RFdiffusion. *Nature*, 620(7976):1089–1100, 7 2023. doi: 10.1038/s41586-023-06415-8. URL https://doi.org/10.1038/s41586-023-06415-8.

[36] Regina Wirth, Peng Gao, G. Ulrich Nienhaus, Murat Sunbul, and Andres Jäschke. SIRA: A Silicon Rhodamine-Binding aptamer for Live-Cell Super-Resolution RNA imaging. *Journal of the American Chemical Society*, 141(18):7562–7571, 4 2019. doi: 10.1021/jacs.9b02697. URL https://doi.org/10.1021/jacs.9b02697.

[37] Jeremy Wohlwend, Gabriele Corso, Saro Passaro, Mateo Reveiz, Ken Leidal, Wojtek Swiderski, Tally Portnoi, Itamar Chinn, Jacob Silterra, Tommi Jaakkola, and Regina Barzilay. Boltz-1: Democratizing Biomolecular Interaction Modeling, November 2024.

[38] Junhao Xiong, Ishan Gaur, Maria Lukarska, Hunter Nisonoff, Luke M. Oltrogge, David F. Savage, and Jennifer Listgarten. ProteinGuide: On-the-fly property guidance for protein sequence generative models, 5 2025. URL http://arxiv.org/abs/2505.04823.

[39] Wei Yang, Shunzhi Wang, Gyu Rie Lee, Jason Z. Zhang, Alexis Courbet, David Juergens, Xinru Wang, Thomas Schlichthaerle, Mohamad Abedi, Robert Ragotte, Linna An, Indrek Kalvet, Sam Pellock, Ljubica Mihaljevic, Cameron Glasscock, Arvind Pillai, Adam Broerman, Nathan Ennist, Ella Haefner, Nora McNamara-Bordewick, Ian Haydon, Lance Stewart, Gaurav Bhardwaj, and David Baker. The past, present and future of de novo protein design. *Nature*, 652(8112):1139–1152, April 2026. ISSN 1476-4687. doi: 10.1038/s41586-026-10328-7.

[40] He Zhang, Liang Zhang, Ang Lin, Congcong Xu, Ziyu Li, Kaibo Liu, Boxiang Liu, Xiaopin Ma, Fanfan Zhao, Huiling Jiang, Chunxiu Chen, Haifa Shen, Hangwen Li, David H. Mathews, Yujian Zhang, and Liang Huang. Algorithm for optimized mRNA design improves stability and immunogenicity. *Nature*, 621(7978):396–403, 5 2023. doi: 10.1038/s41586-023-06127-z. URL https://doi.org/10.1038/s41586-023-06127-z.

[41] He Zhang, Hailong Liu, Yushan Xu, Haoran Huang, Yiming Liu, Jia Wang, Yan Qin, Haiyan Wang, Lili Ma, Zhiyuan Xun, Xuzhuang Hou, Timothy K. Lu, and Jicong Cao. Deep generative models design mRNA sequences with enhanced translational capacity and stability. *Science*, 390(6773):eadr8470, 8 2025. doi: 10.1126/science.adr8470. URL https://doi.org/10.1126/science.adr8470.

[42] Yang Zhang and Jeffrey Skolnick. TM-align: A protein structure alignment algorithm based on the TM-score. *Nucleic Acids Research*, 33(7):2302–2309, April 2005. ISSN 0305-1048. doi: 10.1093/nar/gki524.

[43] Fangzhou Zhao, Chien-hung Yu, and Yi Liu. Codon usage regulates protein structure and function by affecting translation elongation speed in drosophila cells. *Nucleic Acids Research*, 45(14):8484–8492, 06 2017. ISSN 0305-1048. doi: 10.1093/nar/gkx501. URL https://doi.org/10.1093/nar/gkx501.

[44] Valentin Zulkower and Susan Rosser. Dna chisel, a versatile sequence optimizer. *Bioinformatics*, 36(16):4508–4509, 07 2020. ISSN 1367-4803. doi: 10.1093/bioinformatics/btaa558. URL https://doi.org/10.1093/bioinformatics/btaa558.

A Experimental details

NuCaliby experiments use a single checkpoint trained on the Boltz-1 training split (see Appendix D.1 for training details). Sampling is performed with DLMC for 500 sweeps with the Potts temperature annealed to a final value T_{final}=0.01. Designability is evaluated by folding the designed amino-acid sequence with AlphaFold2 [15] via ColabFold [22] in single-sequence mode with 4 recycles using model 1, and comparing predicted structures to the input backbone with TM-align [42].

Interpreting sequence recovery. Native-sequence recovery is a useful reference metric, but the native sequence is not the only possible solution to an inverse-folding task. Furthermore, high sequence recovery does not relate to high self-consistency [3].We therefore report sequence recovery alongside self-consistency and prediction confidence, and use expression and solubility measurements to assess experimental feasibility. Neither high nor low recovery alone establishes design quality.

A.1 Guidance settings

Table 3 summarizes the guidance used in the main nucleotide-level design experiments. In every listed setting, the DLMC proposal includes the nucleotide Potts energy with weight 1 and a premature stop-codon penalty with weight 100. The table lists additional proposal terms and indicates whether Metropolis–Hastings (M–H) acceptance/rejection is enabled. "None" denotes no additional proposal guidance.

Table 3: Guidance settings for NuCaliby nucleotide-level sampling. All rows share the Potts and stop-codon terms described above. The wet-lab row describes the combined tAI-plus-motif condition; the separate tAI-only conditions are detailed in Appendix B.

Experiment	Additional DLMC proposal guidance	Acceptance/rejection
Designability	None	Disabled
tAI adaptation	tAI, $\lambda \in \{0, 5, 10, 30, 50, 100, 300, 600\}$	Disabled
RNA motif inclusion	LogSumExp motif, $\lambda = 50$, $\beta = 2$	Disabled
mRNA stability	None	Enabled; adds ViennaRNA energy ($\lambda \in \{1, 3, 10, 50\}$) to the Potts and stop-codon energies
Wet-lab expression (tAI + motif)	LogSumExp motif, $\lambda = 6$, $\beta = 10$; tAI, $\lambda = 30$	Disabled

When enabled, acceptance uses the total energy, including terms that also influence proposal construction. The $\lambda = 0$ RNA comparison uses the unguided baseline.

A.2 Compute resources

Training data preprocessing with AtomWorks [27] takes 10–20 hours on a multi-CPU node. Pretraining NuCaliby end-to-end takes approximately 15 hours on 8×NVIDIA B200 GPUs. At inference, the encoder predicts the Potts parameters for a single backbone in well under one second on a single H100 GPU; subsequent DLMC sampling reaches roughly 10^3 sweeps per second for proteins of ∼240 residues, so 500 sweeps complete in < 1 s and a single design takes ∼1–2 s end-to-end. Sampling the full 183-protein test set therefore takes only a few minutes per configuration. Structure prediction with ColabFold AF2 in single-sequence mode runs on commodity CPU servers; a GPU is not required, and parallelising independent designs across CPU workers makes evaluation throughput essentially proportional to the available core count.

A.3 Adaptation to organism-specific tRNA pools

The tAI weights for *E. coli* BL21(DE3) (NCBI tax ID 469008) are computed from the host's tRNA gene copy numbers following [8]. We sweep the bias strength $\lambda \in \{0, 5, 10, 30, 50, 100, 300, 600\}$ and sample one sequence per protein at each setting. All methods are evaluated on the same 183 test backbones using a best-of-1 sampling budget and the same BL21(DE3) tAI weights. NuCaliby

uses these weights for inference-time guidance, and DNA Chisel uses them to optimize synonymous encodings of Caliby-generated amino-acid sequences for tAI. CodonMPNN is conditioned on the species-level *E. coli* label (NCBI tax ID 562), represented by approximately 21,000 training examples, rather than BL21(DE3), represented by only one example. The tRNA gene copy numbers used to compute the tAI weights are strain-specific. Thus, the evaluation objective is shared, while organism conditioning follows the information available to each method.

A.4 mRNA sequence motif inclusion

Motifs. **SiRA aptamer [36]** is a synthetic RNA aptamer that binds silicon-rhodamine fluorophores and has previously been inserted into untranslated regions for live-cell RNA imaging using super-resolution microscopy. Here, it serves as a representative example of a complex RNA sequence motif. We do not evaluate the downstream functional consequences of motif embedding on translation efficiency **HIV-1 frameshifting element (FSE) [14, 23]** is a short RNA element embedded within the HIV coding sequence that induces programmed ribosomal -1 frameshifting. This causes the ribosome to shift into an alternative reading frame, enabling a single mRNA to encode multiple protein products.

NuCaliby$^{\text{NT}}_{\text{motif}}$. We sample with the motif energy bias (Eq. (8)) at λ=50 and β=2, and additionally apply a stop-codon penalty with weight 100 to keep the embedded motif from introducing internal stops. The two-motif setting sums the per-motif energies. We sample n=16 sequences per protein and report best-of-n success.

Motif placement is robust across a broad range of (λ, β) values: in our experiments the model achieves a perfect single-motif embedding (d=0) within at most 2 samples and a two-motif embedding within at most 16 samples on every tested backbone. We therefore use slightly different settings in different experiments (e.g. λ=6, β=10 in the wet-lab pipeline of Appendix B, where we additionally combine motif guidance with λ_{tAI}=30) without observing material differences in success rate.

Caliby AA-level baseline. We sample n=64 residue-level sequences per protein from Caliby and check whether any translation matches one of the amino-acid variants derived from the motif across the three reading frames (Figure 10).

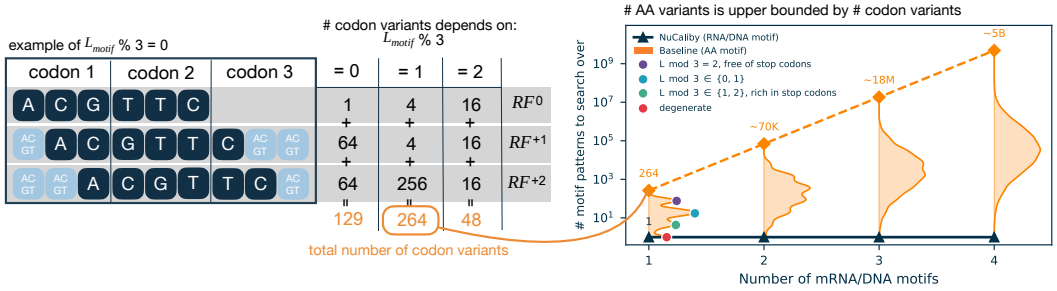

Figure 10: A residue-level baseline (orange) must enumerate every distinct AA pattern that the motifs produce across the three reading frames (RF^0, RF^{+1}, RF^{+2}). Theoretical worst case is 264^k, attained at $\ell_m \bmod 3 = 2$ free of stop codons. Densities at each number of motifs k summarize 1,000 random RNA motifs of length 30–100 nt (products of k i.i.d. samples for $k > 1$). The degenerate case means there is only one possible amino acid variant. NuCaliby adds each motif as a nucleotide-level energy term, keeping the search at a single design problem regardless of k.

A.5 Sensitivity to sampling hyperparameters

We assess motif inclusion on 183 test monomers using one sample per configuration (Table 4). Each configuration varies one setting from the reference configuration: SiRA motif, guidance strength $\lambda = 50$, smooth-min sharpness $\beta = 2$, initial temperature $T_{\text{init}} = 1.0$, final temperature $T_{\text{final}} = 0.01$, and 500 DLMC sweeps. Temperature is annealed over the first 80% of sweeps. We report the percentage of designs with an exact motif match ($d = 0$) or a match within one nucleotide ($d \leq 1$), together with TM and pLDDT.

Within the tested ranges, guidance strength, smooth-min sharpness, final temperature, and sampling budget have relatively small effects on designability, and all designs remain within one nucleotide

of the target motif. Initial temperature has a larger effect on exact inclusion: increasing T_{init} from 1.0 to 2.0 lowers the exact-match rate from 91.8% to 74.9%. Motif choice also affects inclusion and designability. Simultaneous SiRA and FSE inclusion achieves 61.2% exact matches in best-of-1 sampling, with lower TM and pLDDT than either motif alone, although all backbones achieve exact inclusion with up to 16 samples in the best-of-n experiment (Figure 6). Because these motifs differ in sequence as well as length, this comparison does not isolate the effect of motif length.

Table 4: Sensitivity of motif inclusion to sampling hyperparameters and motif identity. Each row uses one sample per test backbone. The two inclusion columns are percentages; scTM and pLDDT are mean designability metrics. Settings other than the one indicated follow the reference configuration described above.

Parameter	Setting	$d = 0$ (%) ↑	$d \leq 1$ (%) ↑	TM ↑	pLDDT ↑
Guidance strength and smooth-min sharpness					
λ	25	87.4	100.0	.487	52.6
λ	50	91.8	100.0	.503	53.4
λ	100	87.4	100.0	.490	53.0
β	1	90.2	100.0	.484	52.7
β	2	91.8	100.0	.503	53.4
β	5	92.3	100.0	.500	53.6
Sampling temperature					
T_{final}	0.005	92.3	100.0	.494	53.0
T_{final}	0.01	91.8	100.0	.503	53.4
T_{final}	0.05	91.3	100.0	.492	53.2
T_{init}	0.5	94.5	100.0	.491	53.0
T_{init}	1.0	91.8	100.0	.503	53.4
T_{init}	2.0	74.9	100.0	.496	53.3
Sampling budget					
DLMC sweeps	250	87.4	100.0	.493	53.7
DLMC sweeps	500	91.8	100.0	.503	53.4
DLMC sweeps	750	89.1	100.0	.495	53.4
Motif identity and length					
Motif	SiRA (50 nt)	91.8	100.0	.504	53.4
Motif	FSE (66 nt)	72.7	100.0	.465	50.3
Motifs	SiRA + FSE (50 + 66 nt)	61.2	88.0	.410	47.4

A.6 List of pdbs in test set

The 183 soluble monomers from the Caliby benchmark used in all evaluations.

```
7qnl, 7qpk, 7qpl, 7qqd, 7qsj, 7sc4, 7t4w, 7tl6, 7u08, 7u71, 7uba, 7ui4,
7uk8, 7uqu, 7ur2, 7urp, 7v34, 7wkp, 7wmy, 7wnh, 7woi, 7wrn, 7x2x, 7xcc,
7xh4, 7xhz, 7xrx, 7xs3, 7xvi, 7xwy, 7xz3, 7y01, 7y1s, 7y3z, 7y43, 7y4h,
7y7o, 7y8h, 7yia, 7ykm, 7yl4, 7ynx, 7ypq, 7ysi, 7yuk, 7yyg, 7z3i, 7z64,
7zao, 7zcb, 8a0a, 8a16, 8acc, 8acr, 8adc, 8add, 8agg, 8agr, 8ahd, 8an0,
8au6, 8axj, 8ay2, 8b2e, 8b2h, 8b2s, 8b4c, 8b5w, 8b6a, 8b6p, 8b6q, 8b6u,
8bat, 8bbr, 8bbz, 8bdq, 8bf8, 8bfy, 8bl6, 8bux, 8bvi, 8bw6, 8c27, 8c4d,
8cih, 8ck0, 8cmq, 8cpk, 8cpn, 8cpq, 8cr7, 8d40, 8daj, 8deo, 8dhj, 8dvq,
8dwj, 8dx0, 8e5d, 8e5e, 8ec3, 8eel, 8efz, 8ehn, 8eil, 8eov, 8erm, 8ex9,
8exa, 8ezw, 8f5g, 8f9y, 8fkl, 8g0n, 8g32, 8g53, 8g8e, 8ga6, 8gld, 8gmx,
8gmz, 8gsx, 8gsy, 8gty, 8gxr, 8gy4, 8gyr, 8h0l, 8h18, 8h1j, 8h55, 8h5s,
8hbb, 8hbr, 8he2, 8hm4, 8hp8, 8htx, 8huz, 8i3j, 8iib, 8in6, 8iom, 8iqc,
8irq, 8j69, 8jah, 8jdg, 8jw3, 8jws, 8k5u, 8oei, 8ok3, 8olj, 8ou1, 8ow8,
8oyy, 8p0c, 8p5q, 8paq, 8pmu, 8pof, 8pxc, 8qcw, 8r8a, 8sot, 8ssu, 8sxs,
```

8t1m, 8t3j, 8t4c, 8t8l, 8tfs, 8tjg, 8tnm, 8tvl, 8uk7, 8upi, 8urn, 8uz8,
8v1k, 8w7p, 8y4u

B Wet-lab validation

To obtain a diverse set of protein backbones for sequence generation, we curated a set of eight structures spanning different sources and design methods (Table 5). The set includes four published de novo backbones retrieved from the protein design archive [4], and four additional de novo backbones generated with RF diffusion [35]. This selection covers a range of backbone generation approaches, including AlphaFold hallucination, RFjoint2, Chroma, and RF diffusion.

Name	PDB	BB method	Seq method	Category	Publication	SSE	Length [#AA]
halc1_878	8CYK	AF hallucination	ProteinMPNN	Published de novo	Science (2022)	α/β	136
BBn6	9XZT	RFjoint2	ProteinMPNN	Published de novo	PNAS (2024)	β	95
Unc_239	8TNO	Chroma	Chroma	Published de novo	Nature (2023)	α	285
tbf_24	8OYS	AF hallucination	ProteinMPNN	Published de novo	Nature (2024)	α/β	192
RF1_14	–	RF diffusion 1	–	De novo (this work)	–	α/β	120
RF1_3	–	RF diffusion 1	–	De novo (this work)	–	α/β	120
RF1_2	–	RF diffusion 1	–	De novo (this work)	–	α	120
RF1_6	–	RF diffusion 1	–	De novo (this work)	–	α/β	120

Table 5: Protein backbones used for experimental evaluation. SSE = secondary structure estimation.

B.1 Creating sequences

We generated 24 sequences per backbone for each method, providing an equal sampling budget across sequence design approaches:

- **ProteinMPNN** with soluble weights and default inference settings [11].
- **CodonMPNN** with default inference settings, conditioned on *E. coli* (Taxon ID 562), which constitutes 22% of the taxon IDs in its training set.
- **NuCaliby** with tAI energy biases $\lambda \in \{0, 30, 50, 300\}$.
- **NuCaliby** tAI energy bias $\lambda = 30$ and RNA motif guidance, embedding a random 29-nucleotide sequence: `GTCCCCGCAGGTTGATCCTAAACGGCTCT`.
- **NuCaliby** (amino acid branch), sampling from the residue-level Potts model.

For methods that generate amino acid sequences, namely ProteinMPNN and the NuCaliby amino acid branch, we obtained DNA sequences by reverse translation followed by codon optimization with DNA Chisel. Optimization targeted *E. coli* codon usage using CodonOptimize and MaximizeCAI, while enforcing translation, removing BsaI sites, constraining GC content to 30–70%, and requiring 80-mer uniqueness.

We applied a multi-stage filtering pipeline to ensure constructability and structural quality. First, we removed sequences containing the BsaI restriction site required for Golden Gate cloning, as well as sequences with premature stop codons. For motif-guided designs, we used a tAI guidance strength of $\lambda = 30$ and a motif guidance strength of $\lambda = 6$ with a smooth min sharpness of $\beta = 10.0$ and retained only sequences containing the target motif.

We then evaluated all translated amino acid sequences with AlphaFold2, 4 recycles, model 1, single-sequence mode. Designs were screened by pLDDT > 80 and self-consistency RMSD (scRMSD) < 2 Å. Filtering was applied independently for each backbone and design method. When more than three candidates passed these structural filters, we preferred designs with isoelectric point pI < 6.5; if this yielded fewer than three candidates, we relaxed the preference to pI < 6.5 or pI > 8.5. When fewer than three candidates passed the structural filters, we retained the top three candidates ranked by increasing scRMSD, using pLDDT as a secondary criterion.

All surviving candidates were screened using the GenScript sequence complexity check. If more than two designs remained after filtering, we selected the two candidates with the lowest self-consistency RMSD (scRMSD) for synthesis. Designs failing the GenScript screen were instead submitted to Twist fragment synthesis. Sequences that also failed Twist synthesis constraints were classified as unsynthesizable.

This procedure evaluates all methods under an equal sampling budget and a shared selection pipeline, comparing them based on their best achievable designs.

B.2 Expression

Experimental validation procedures followed the SAPP pipeline [25] Final evaluations of protein expressibility were based on size exclusion chromatography after Ni-NTA based HIS-tag purification of a 4 mL autoinduction culture grown for 22h at 37 C. Protein yield and expected monodispersity were derived from the elution profile on a Cytiva Superdex 75 Increase 5/150 GL column in PBS at room temperature.

B.3 Details for in-silico filtering and tested designs

Prefiltering sequence motifs We started with 8 backbones $\times$ 24 sequences = 192 sequences per method. Overall, there were 9 methods, as described in B1. Thus, 1728 sequences were subject to filtering.

Sequences that were guided to contain the motif were filtered to check whether they contained the motif:

- Motif `GTCCCCGCAGGTTGATCCTAAACGGCTCT` occurs in 123 out of 192 sequences (64.1%).

Further filtering removed Type IIS restriction sites reserved for the cloning reaction (commonly removed):

- `GGTCTC` and `GAGACC`.
- This removed 116 designs: 95 for CodonMPNN, 14 for NuCaliby NT, 5 for NuCaliby NT 30 tAI, and 2 for NuCaliby NT 30 tAI + Motif.

Filtering amino acid properties Stronger guidance usually decreases the pass rate at the amino acid level as more constraints have to be satisfied, which is in line with the previously reported trade-off in tAI and motif inclusion. Without constraints, the pass rate of the amino acid and nucleotide level NuCaliby model is similar. We provide the filtering results below, aggregated over all backbones:

Method	Start	pLDDT $>$ 80	RMSD $<$ 2.0	pI $<$ 6.5	Loose pI	No pI	After pI	Final	Pass Rate
NuCalibyAA	192	178	173	173	0	0	173	173	0.90
CodonMPNN	97	68	64	61	0	9	70	70	0.72
NuCaliby	178	162	157	153	0	0	153	153	0.86
NuCaliby tAI 30	187	167	164	130	0	0	130	130	0.70
NuCaliby tAI 50	192	166	163	100	36	0	136	136	0.71
NuCaliby tAI 100	192	140	134	64	37	6	107	107	0.56
NuCaliby tAI 300	192	105	76	8	52	12	72	72	0.38
NuCaliby tAI 30 + NT motif	121	86	82	68	0	3	71	71	0.59
ProteinMPNN	192	163	162	162	0	0	162	162	0.84

Filtering synthesis constraints For each of these DNA sequences, we took all sequences that passed the filters and checked whether they could be synthesized by a commonly used gene synthesis company (GenScript). If more than two sequences per method passed, we chose those with the smallest scRMSD.

Overall, strongly guided sequences as well as CodonMPNN sequences had repetitive patterns, which made more than 50% of them too complex to be synthesized (we note that this issue did not occur for NuCaliby base). The pass rate decreased overall with increasing guidance.

Method	Passed	Failed	Total	Pass Rate (%)
NuCalibyAA	173	0	173	100.0
CodonMPNN	33	37	70	47.1
NuCaliby	153	0	153	100.0
NuCaliby tAI 30	87	43	130	66.9
NuCaliby tAI 50	72	64	136	52.9
NuCaliby tAI 100	47	60	107	43.9
NuCaliby tAI 300	10	62	72	13.9
NuCaliby tAI 30 + NT motif	54	17	71	76.1
ProteinMPNN	122	40	162	75.3

If fewer than two sequences passed the GenScript complexity filters per backbone and method, we ordered the two sequences with the best RMSD from another gene synthesis company specializing in more complex sequences (Twist). This worked for 12 out of 29 missing designs after two synthesis attempts.

Thus, overall, we obtained:

Method	**n**
NuCaliby AA	16
CodonMPNN	9
NuCaliby NT	16
NuCaliby NT tAI 30	16
NuCaliby NT tAI 50	16
NuCaliby NT tAI 100	15
NuCaliby NT tAI 300	8
NuCaliby NT tAI 30 + motif	16
ProteinMPNN	14

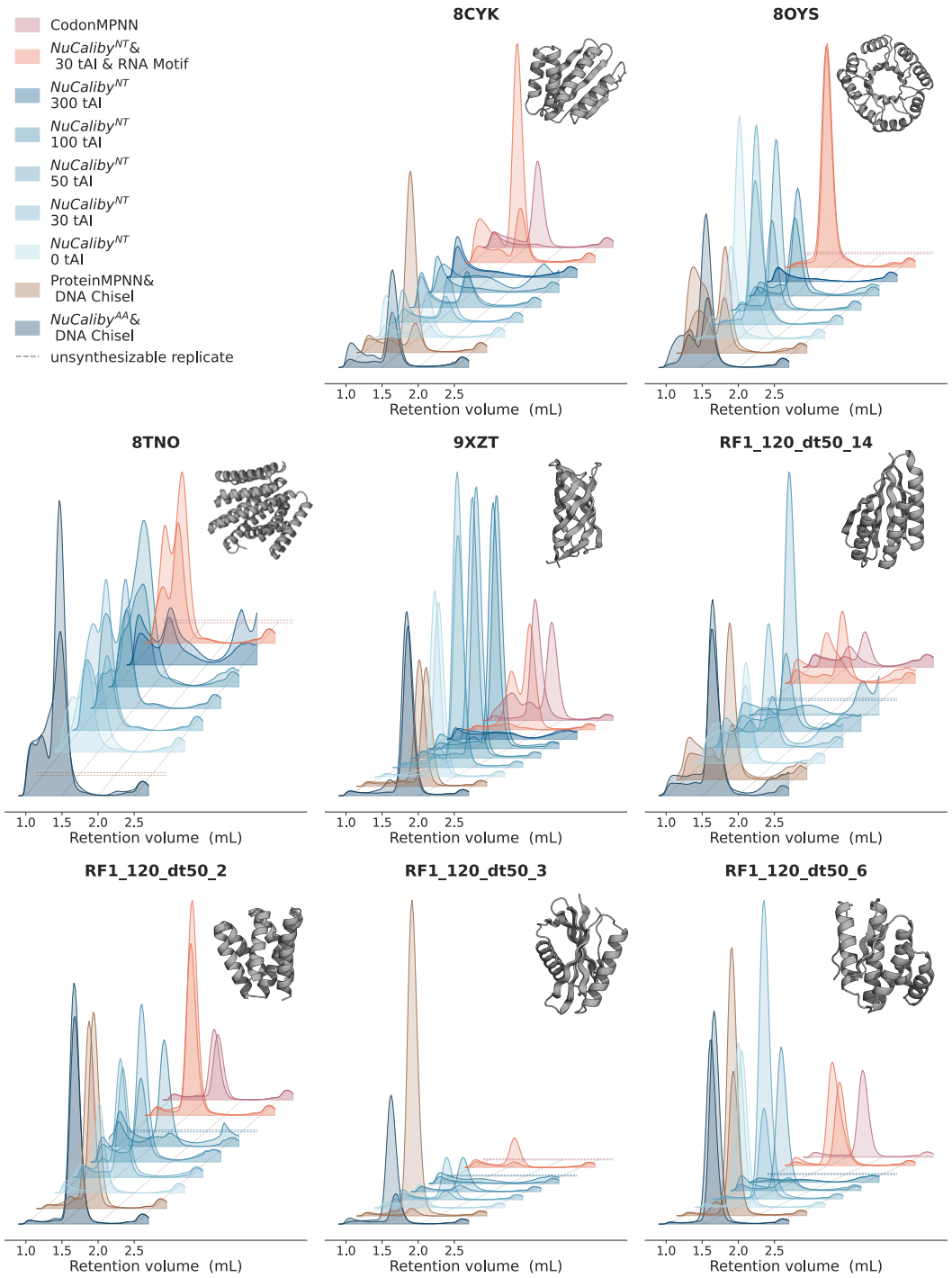

Figure 11: Size-exclusion chromatograms for different backbones and sequence design methods. Dashed traces indicate unsynthesizable designs for which the DNA sequence could not be synthesized as a fragment by GenScript or Twist.

C Limitations

Approximate synonymous invariance. Ideally, the structural energy would assign equal energies to synonymous encodings of the same amino-acid sequence. We assess this property across all 183 test monomers by sampling one unguided design per backbone and evaluating 100 random single-codon substitutions of each type. The median per-protein mean absolute energy change is 0.085 [IQR: 0.073–0.095] for synonymous substitutions and 3.54 [3.33–3.67] for non-synonymous substitutions (Figure 12A). The median within-protein ratio is 2.4%, indicating substantially weaker sensitivity to synonymous changes, but not exact invariance.

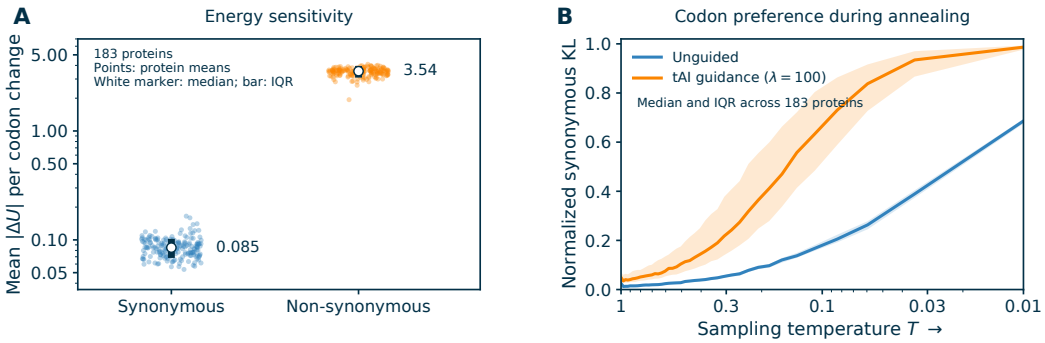

Figure 12: Approximate synonymous invariance across 183 test proteins. **A:** Each point is one protein's mean absolute Potts-energy change over 100 random single-codon substitutions, shown separately for synonymous and non-synonymous substitutions on a logarithmic energy axis. White markers and dark bars indicate the median and interquartile range across proteins. **B:** Normalized KL divergence of synonymous-codon conditional distributions from uniform during unguided and tAI-guided ($\lambda = 100$) DLMC sampling. Lines and shaded bands show the median and interquartile range across proteins; zero denotes uniform conditionals and one denotes concentration on a single codon. Measurements are averaged over eligible residues within each protein. Repeated measurements at the final temperature are averaged within each protein before aggregation.

To assess the effect of residual preferences during sampling, we evaluate conditional distributions over synonymous codons along one unguided and one tAI-guided DLMC trajectory per backbone, each comprising 500 sweeps with temperature annealed from 1.0 to 0.01 over the first 80% of sweeps. Every 10 sweeps, we hold the remaining sequence fixed and evaluate $p(c) \propto \exp(-U_c/T)$ over the synonymous codons of each eligible residue, including the tAI energy in the guided condition. We normalize the KL divergence from the uniform distribution by the logarithm of the synonymous-set size and average over residues with more than one synonymous codon. These preferences are weak at high temperature but become pronounced during annealing (Figure 12B): at $T = 0.01$, the median normalized divergence is approximately 0.69 without guidance and 0.99 with tAI guidance at $\lambda = 100$. Thus, small residual energy differences can substantially bias low-temperature codon choice. Synonymous marginalization encourages approximate invariance but does not guarantee uniform sampling over synonymous encodings.

Simplified biological proxies. Our nucleotide-level objectives measure tRNA adaptation, motif-sequence inclusion, and predicted RNA folding energy. The wet-lab experiments establish expression and solubility of guided designs, but do not directly measure translation efficiency, RNA motif activity, or cellular mRNA stability. Translation efficiency is governed by multiple interacting factors, including codon usage, mRNA secondary structure, and ribosome dynamics [17, 40]. Similarly, functional RNA motifs are not defined solely by their primary sequence but often depend on higher-order structural features, which are not captured by sequence-level motif incorporation alone [36]. The framework, however, is agnostic to the form of the bias term: more biologically faithful objectives (e.g. position-dependent tAI profiles, RNA secondary-structure constraints) can be plugged in via the same DLMC-with-guidance and Metropolis–Hastings interfaces without retraining.

D Implementation details

D.1 Noise-dependent Potts loss weighting

The Potts conditional is evaluated using the ground-truth sequence at non-target positions, while the predicted Potts parameters depend on the masked sequence supplied to the encoder. We use t for the probability of retaining a residue in that input, so the masking probability is $1 - t$. At high t, the encoder sees nearly the full native sequence. Training and validation losses stratified by t show that overfitting is concentrated in this mostly unmasked regime (Figure 13). We therefore apply the weight $\omega(t) = (1 - t)^{\alpha}$ to the Potts loss, reducing the contribution of these easy examples. This suppresses the observed overfitting and improves validation Potts loss across conditioning levels. We use $\alpha = 1.25$.

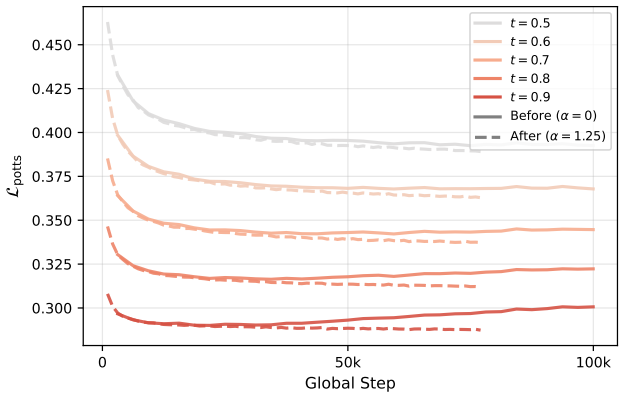

Figure 13: Noise-dependent Potts loss weighting helps prevent overfitting on highly unmasked sequences. Validation Potts loss stratified by sequence-retention probability t (masking probability $1 - t$). Without weighting (solid), high-t bins overfit after $\sim$40k steps. With $(1-t)^{\alpha}$ weighting (dashed, α=1.25), overfitting is eliminated and the loss improves across all t.

E Ablation studies

We evaluate nucleotide-graph connectivity and joint amino-acid/nucleotide supervision on the 183 test monomers using the designability protocol in Appendix A. Tables 6 and 7 report means and standard deviations across proteins.

Nucleotide-graph connectivity. We train two additional models from scratch, removing either intra-codon or inter-codon edges. Without intra-codon edges, interactions among nucleotides within a codon must be mediated indirectly through neighbouring codons. Without inter-codon edges, the nucleotide Potts coupling graph separates into codon-level components; the predicted fields can still carry contextual information, but direct pairwise couplings between codons are absent. Removing either edge type reduces designability (Table 6).

NT graph	AA recovery (%)	pLDDT	scTM	scRMSD (Å)
Intra + inter	38.6 $\pm$7.1	60.8 $\pm$19.1	.579 $\pm$.245	3.85 $\pm$1.82
No intra-codon edges	37.9 $\pm$7.3	55.1 $\pm$18.2	.504 $\pm$.225	4.40 $\pm$1.72
No inter-codon edges	36.8 $\pm$6.6	54.1 $\pm$17.7	.500 $\pm$.223	4.49 $\pm$1.80

Table 6: Effect of nucleotide-graph connectivity on nucleotide-level sampling.

Joint AA/NT training. We compare joint training of both branches with training either branch alone (Table 7). The amino-acid branch performs similarly under joint and AA-only training. The nucleotide branch benefits from joint supervision, with higher pLDDT and scTM and lower scRMSD than NT-only training.

Training	Sampling	AA recovery (%)	pLDDT	scTM	scRMSD (Å)
AA + NT	AA	39.6 $\pm$7.2	61.8 $\pm$19.6	.593 $\pm$.246	3.84 $\pm$1.85
AA only	AA	40.0 $\pm$7.2	62.1 $\pm$19.5	.603 $\pm$.245	3.71 $\pm$1.86
AA + NT	NT	38.6 $\pm$7.1	60.8 $\pm$19.1	.579 $\pm$.245	3.85 $\pm$1.82
NT only	NT	38.7 $\pm$7.2	58.9 $\pm$18.5	.548 $\pm$.227	4.15 $\pm$1.73

Table 7: Effect of joint and separate branch training, evaluated by sampling from the indicated Potts model.

F Benchmark on *de novo* proteins

To assess model performance on backbones that are highly dissimilar to the training set, we constructed a set of de novo backbones. We generated 10'000 backbones of 100 to 500 amino acids length using RFdiffusion1 [35]. A Foldseek search [34] (TM-align mode) was performed against the PDB mirror used for training and testing (date: January 4 2026), and we kept all backbones with a query-normalized `qtmscore` of less than 0.5 to every PDB entry, or with no hit at all. The remaining backbones were clustered with Foldseek, such that two backbones are in the same cluster if their TM-score is at least 0.5, normalized by the query length. Cluster representatives were then compared to achieve a random selection (fixed seed) of 200 representatives, such that the maximum TM-score between any two selected backbones, over both length normalizations `qtmscore` and `ttmscore` in both search directions, is at most 0.5. This yields a set of 200 backbones that are dissimilar to each other and for which no PDB hit with a TM-score of 0.5 or higher was found. The Foldseek commands were (all other options at their defaults):

```
# search against the PDB mirror
foldseek easy-search backbones pdb_mirror hits.tsv tmp \
    -s 7.5 --alignment-type 1 \

# cluster backbones that passed the PDB filter
foldseek easy-cluster eligible clusters tmp \
    --alignment-type 1 \
    --tmscore-threshold 0.5 --tmscore-threshold-mode 1 \
    --single-step-clustering 1 --tmalign-fast 0 \
    -s 7.5 -e inf --max-seqs <N_passed>

# compare cluster representatives with each other
foldseek easy-search reps reps pairwise.tsv tmp \
    --exhaustive-search 1 --alignment-type 1 --tmalign-fast 0 \
    -e inf --max-seqs <N_cluster_representatives> \
```

Each sequence design method creates one sequence per backbones and self-consistency is assessed as described in Section A.

On these structurally dissimilar de novo backbones, NuCaliby NT achieved protein sequence design capabilities comparable to the widely used and extensively experimentally validated, state of the art model, ProteinMPNN while operating directly in nucleotide space. NuCaliby NT also substantially outperformed CodonMPNN, a model specifically trained for the task of design DNA sequences encoding protein backbones, across all three designability metrics. Ensemble conditioning further improved NuCaliby NT, resulting in higher mean pLDDT and scTM compared to ProteinMPNN. NuCaliby AA achieved the strongest overall protein sequence-level performance, with additional improvements from ensemble conditioning. The pdbs used are provided in the GitHub.

Table 8: Sequence design on 200 *de novo* backbones (one sequence per backbone, mean ± standard deviation over backbones). Structures predicted with ColabFold (single sequence). Best value per group in bold.

	Method	pLDDT ↑	scTM ↑	scRMSD (Å) ↓
Nucleotide	CodonMPNN	53.3 ± 11.5	0.48 ± 0.17	4.96 ± 1.36
	NuCaliby NT	67.9 ± 15.1	$\mathbf{0.69} \pm 0.22$	$\mathbf{3.57} \pm 1.64$
	NuCaliby NT + ensemble	$\mathbf{69.6} \pm 13.8$	$\mathbf{0.69} \pm 0.21$	$\mathbf{3.57} \pm 1.61$
Amino acid	ProteinMPNN	62.8 ± 15.8	0.64 ± 0.23	3.94 ± 1.74
	NuCaliby AA	73.1 ± 15.6	0.76 ± 0.21	3.05 ± 1.71
	NuCaliby AA + ensemble	$\mathbf{76.7} \pm 14.5$	$\mathbf{0.79} \pm 0.21$	$\mathbf{2.73} \pm 1.63$

G Effect of training data on designability

CodonMPNN [28] achieves comparable sequence recovery to ProteinMPNN (44.5% vs 44.4%) yet substantially lower designability (TM .441 vs .509, pLDDT 47.0 vs 54.3). CodonMPNN is trained on genomic sequences paired with AlphaFold-predicted structures rather than experimentally determined PDB structures. Tan et al. [31] show that training inverse folding models on AFDB structures degrades sequence recovery by up to 27 percentage points relative to PDB-trained models, and that even debiasing predicted structures does not fully close this gap. Our results suggest that the penalty manifests not in recovery rates but in designability, where the structural fidelity of the training data directly impacts whether designed sequences fold correctly and are self-consistent with the original backbone (Figure 14). PDB alone does not provide enough examples for different taxons to train an organism-conditioned model like CodonMPNN, motivating NuCaliby's approach of training on PDB and deferring organism-specific preferences to inference-time energy biases.

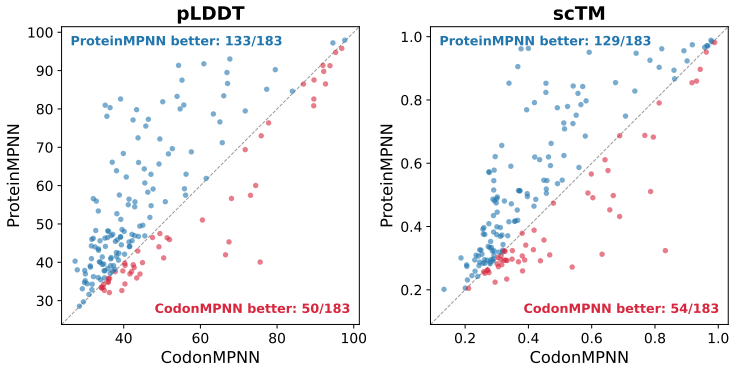

Figure 14: Per-protein designability comparison between ProteinMPNN trained on experimentally determined structures deposited at PDB vs CodonMPNN trained on predicted strcutures from AlphaFold Database. Each point is one of the 183 test proteins. Points above the diagonal indicate higher designability for ProteinMPNN. ProteinMPNN produces more designable sequences on 133/183 proteins by pLDDT and 129/183 by scTM, suggesting that training on predicted structures degrades designability.

H Broader Impact

NuCaliby's broader impact lies in extending protein design from a residue-level problem to a joint optimization problem across the central dogma of molecular biology, enabling simultaneous control over protein structure, coding-sequence properties, and cellular expression context. This could improve the reliability and cost-efficiency of producing therapeutic proteins, vaccines, enzymes, and other biologics by directly incorporating objectives such as host-specific translation adaptation, mRNA stability, and regulatory RNA elements during sequence generation rather than through post-hoc optimization. The framework may also accelerate research in synthetic biology and biotechnology by enabling programmable integration of RNA-level functionality into designed proteins.

Like all protein-design technologies, NuCaliby carries a general dual-use risk because methods that improve biomolecular engineering could in principle be misapplied. However, the method does not provide a targeted capability for harmful applications beyond existing public protein-design tools. Its released models are comparable in capability to already-public residue-level inverse-folding approaches such as ProteinMPNN, CodonMPNN, and Caliby, and practical use still requires substantial domain expertise, experimental infrastructure, and wet-lab validation.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction enumerate the three contributions (nucleotide-level Potts model, inference-time energy guidance, and computational + wet-lab evidence), and each is supported by a dedicated experimental section: codon organisation and designability (§4.1), tRNA adaptation (§4.2), motif inclusion (§4.3), mRNA stability (§4.4), and *in vitro* validation (§4.5, Appendix B).

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: A dedicated Limitations appendix discusses (i) the imperfect synonymous invariance of the learned nucleotide Potts model (Figure 12) and (ii) the use of tAI and motif objectives as simplified proxies for the underlying biological processes (translation efficiency, RNA secondary structure). The main text additionally acknowledges that strong guidance ($\lambda > 50$) trades self-consistency for nucleotide adaptation (§4.2), and that the wet-lab validation is limited to eight backbones and an initial expression assay rather than a full functional characterization (§4.5, Appendix B).

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.

- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not include formal theorems or proofs. All formulae (Potts energy, codon-marginalised loss, DLMC proposal, Metropolis–Hastings acceptance) are stated with their assumptions in §3.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The methods section specifies the architecture, nucleotide-graph construction, training objective, and DLMC sampling procedure. Training data (Boltz-1 PDB split clustered at 40% identity with MMseqs2), evaluation set (183 soluble monomers, listed by PDB ID in Appendix A.6), evaluation protocol (single-sequence ColabFold AF2 with 4 recycles, model 1, TM-align), and guidance hyperparameters (λ, β, stop-codon penalty) are all reported. Code and trained checkpoints will be released alongside the paper.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example

(a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The training/evaluation code, sampling scripts, and trained model weights will be released as an open-source repository upon publication. The PDB structures used for training and evaluation are publicly available; the test set is fully enumerated in Appendix A.6, and external tools (ColabFold, ViennaRNA, DNA Chisel, ProteinMPNN, CodonMPNN) are publicly distributed and cited. For the anonymous review period the code is provided as supplemental material.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The training set, masking schedule, loss formulation, and noise-dependent weighting are described in §3. Sampling settings (DLMC, simulated annealing, ensemble aggregation across 15 Protpardelle conformers), guidance strengths λ, smooth-min sharpness β, and stop-codon penalty weights are reported in the relevant experiment subsections and in the appendix. The full hyperparameter configurations are also released with the code.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Designability metrics in Table 1 are reported as mean $\pm$ standard deviation across the 183 test proteins. Figures 5, 6, and 8 include error bars or per-protein scatter to convey variability. The motif-trajectory analysis aggregates 100 independent sampling runs (Figure 7).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Compute requirements are reported in Appendix A.2: data preprocessing with AtomWorks takes 10–20 h on a multi-CPU node; pre-training takes $\sim$15 h on 8$\times$NVIDIA B200 GPUs; inference is dominated by Potts-parameter prediction ($<$ 1 s on a single H100) followed by DLMC sampling at $\sim 10^3$ sweeps/s, so a single design takes 1–2 s and the full 183-protein test set runs in a few minutes per configuration. Structure prediction with ColabFold AF2 in single-sequence mode runs on commodity CPU servers and parallelises trivially across designs.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.

- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work uses publicly available protein structure databases and standard *E. coli* laboratory strains. No human subjects, personally identifiable data, or scraped web data are involved. The authors have reviewed the NeurIPS Code of Ethics and the research conforms to it.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: NuCaliby's positive impacts include cheaper and more reliable production of therapeutic proteins, vaccines, and industrial enzymes by jointly optimizing across the central dogma. Like all protein-design methods, it carries a generic dual-use risk; however, the framework offers no targeted advantage for harmful applications, requires expert wet-lab infrastructure, and the released models are no more powerful than already-public residue-level inverse-folding tools (ProteinMPNN, CodonMPNN, Caliby).

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The released model is a structure-conditioned inverse-folding model in the same risk class as widely available tools such as ProteinMPNN and CodonMPNN. It does not generate biothreat-specific content and offers no targeted capability over public alternatives, so no additional safeguards beyond standard scientific code release are required.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All datasets, models, and tools used (PDB, Boltz-1 training split, ProteinMPNN, CodonMPNN, Caliby, Protpardelle, ColabFold/AlphaFold2, ViennaRNA, DNA Chisel, TM-align, MMseqs2) are cited at first use in the main text and appendix, and are used in compliance with their respective open-source or academic licenses.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The new assets, NuCaliby model code, trained checkpoints, evaluation scripts will be released with documentation in the public repository accompanying the paper.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.

- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects. *In vitro* validation uses standard *E. coli* BL21(DE3) expression and biochemical assays.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The research does not involve human subjects, so IRB approval is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: General-purpose LLMs are not part of the core methodology.

    Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.