AN INTEGRATED COMPUTATIONAL-EXPERIMENTAL PLATFORM FOR HOLISTIC MRNA SEQUENCE DESIGN, BUILD, TEST, AND LEARN

Anonymous authors
Paper under double-blind review

ABSTRACT

Messenger RNA therapeutics hold broad potential across infectious disease, oncology, and rare genetic disorders, yet designing sequences that simultaneously optimize stability, translation efficiency, manufacturability, and immunogenicity remains challenging due to the combinatorial size of sequence space and trade-offs between therapeutic objectives. Here we present an integrated design-build-test-learn platform that addresses these challenges through three contributions: (1) ChimeraFold, a codon-graph dynamic programming algorithm achieving 2.9× speedup and 522% expanded sequence space coverage over prior methods; (2) a high-throughput automated wet-lab pipeline generating 29,000+ multimodal measurements; and (3) a contrastive learning framework for active learning-guided sequence selection. Evaluation on GFP and SpCas9 systems demonstrated 2.9-fold median improvement in stability, 61.8% average enhancement in expression across four cell lines, and 1.5-fold improvement in gene editing efficiency over wild-type controls. The platform achieves mRNA half-lives up to 173 hours while preserving burst expression, and generalizes to commercial therapeutic targets in the hands of external partners with multiple-fold improvements in expression and stability.

1 INTRODUCTION

Messenger RNA therapeutics represent a transformative modality for treating diseases from infections to cancer to rare genetic disorders. COVID-19 vaccine deployment validated mRNA's capacity for rapid design and scalable manufacturing, while exposing key challenges: therapeutic efficacy depends jointly on *in vivo* stability, translation efficiency, expression kinetics, manufacturing quality, and immunogenicity. These properties exhibit complex interdependencies and opposing gradients in sequence space.

Sequence design has progressed through several generations. Early methods optimized individual metrics (GC content, codon adaptation index (CAI), minimum free energy (MFE)) via independent heuristics (Sharp & Li, 1987; Alexaki et al., 2019). LinearDesign (Zhang et al., 2023) demonstrated that co-optimizing secondary structure free energy with codon usage improves stability and yield (see Ward et al., 2025, for a comprehensive review), but returns single sequences per parameter setting, sampling only the Pareto frontier. This approach has two limitations: (1) it does not characterize the interior feasible region, limiting candidate diversity and information gained per design cycle; and (2) computational proxies (MFE, CAI) explain only a fraction of observed variance in cellular assays, particularly with clinically relevant modifications like N1-methylpseudouridine that are absent from model training data.

We present an integrated platform addressing both limitations through:

1. **ChimeraFold**, an exact codon-graph algorithm exposing a four-parameter penalty family over nucleotide composition. Varying these penalties recovers the *supported (exposed) optima* and additional stochastic suboptimal sampling fills interior regions achieving 522% hypervolume expansion over single-solution methods.

2. **High-throughput wet-lab pipeline** generating 29,000+ measurements across expression kinetics, stability, manufacturability, and function in four cell lines, with 96-construct daily throughput and standardized QC eliminating distribution shift.
3. **Contrastive active learning** that trains pairwise discriminators on multimodal data and applies calibration-aware acquisition to guide iterative sequence selection.

2 RESULTS

2.1 DESIGN: DIVERSE SEQUENCE GENERATION WITH GUARANTEED COVERAGE

Sequence design takes an amino acid sequence plus specifications (expression kinetics, organism, UTRs, constraints) and generates diverse candidates spanning the feasible region.

ChimeraFold Algorithm. Let $\pi = (c_1, \ldots, c_n) \in \Pi$ denote a codon path encoding the target protein (thus fixing the CDS). Let

$$\Delta G^\star(\pi) := \min_{s \in \mathcal{S}(\pi)} \Delta G(s \mid \pi)$$

be the MFE of the induced full mRNA (fixed UTRs included) under a nearest-neighbor model. With codon weights $w(c) > 0$, nucleotide-count vector $\mathbf{c}(\pi) \in \mathbb{Z}^4_{\geq 0}$ (A,C,G,U counts), tradeoff $\lambda \geq 0$ (energy units), and penalties $\mathbf{p} \in \mathbb{R}^4$ (energy per nucleotide), ChimeraFold solves

$$\pi^\star(\lambda, \mathbf{p}) \in \arg\min_{\pi \in \Pi} \; F_{\lambda,\mathbf{p}}(\pi) := \Delta G^\star(\pi) \; - \; \lambda \sum_{t=1}^{n} \ln w(c_t) \; + \; \mathbf{p}^\top \mathbf{c}(\pi). \tag{1}$$

(Equivalently, if $\mathrm{CAI}(\pi) := (\prod_{t=1}^n w(c_t))^{1/n}$ then $\sum_t \ln w(c_t) = n \ln \mathrm{CAI}(\pi)$.) Writing $B(\pi) := \Delta G^\star(\pi) - \lambda \sum_t \ln w(c_t)$, the objective is $B(\pi) + \mathbf{p}^\top \mathbf{c}(\pi)$; varying $\mathbf{p}$ recovers supported optima on the lower convex hull of $\{(\mathbf{c}(\pi), B(\pi)) : \pi \in \Pi\}$, while stochastic suboptimal rounds sample interior regions (Appendix B).

Benchmarks. On eGFP, SpCas9, and SARS-CoV-2 Spike sequences: ChimeraFold achieved **2.91**$\times$ geometric mean speedup over LinearDesign, **3.57**$\times$ memory reduction, and **522%** hypervolume expansion (Table 1).

Table 1: ChimeraFold vs. LinearDesign performance across three sequences (Hydra-parallel ChimeraFold).

Metric	eGFP	SpCas9	Spike
Speedup	4.29$\times$	2.27$\times$	2.52$\times$
Memory reduction	18.83$\times$	1.42$\times$	1.70$\times$
Hypervolume expansion	6.22$\times$ (522% increase)		

Sequence Optimization. For optional downstream optimization of designed sequences against arbitrary oracle fitness functions, we developed Phoenix, an adaptive beam search algorithm with improvement-gated rollouts and plateau-triggered adaptation (Appendix B.3). Benchmarked against AdaBeam on 12 BPNet transcription factor binding tasks from NucleoBench, Phoenix achieved **43% higher fitness on average** at equal evaluation budgets (2000 oracle queries), winning 12/12 tasks with improvements ranging from 2% to 89% (Table 2).

Constraint Scrubbing. Generated sequences undergo synonymous substitutions to eliminate restriction sites, T7 terminator sequences, homopolymers, and hydrolysis hotspots while maintaining GC bounds. A gradient-boosted model predicts per-nucleotide in-line hydrolysis susceptibility; hotspots are repaired via concentric codon shell expansion (Appendix B.4).

Diversity Selection. From scrubbed candidates $\mathcal{C}$, we select n sequences maximizing coverage in normalized feature space $\mathbf{x}_i = (\widetilde{\mathrm{GC}}, \widetilde{\mathrm{CAI}}, \widetilde{\mathrm{EFE}}, \widetilde{\mathrm{AUP}}, \widetilde{H})$ via farthest-first traversal (max-min dispersion): each new sequence maximizes its minimum distance to the already-selected set. Under a metric distance (we use ℓ_2), this greedy procedure achieves a $\frac{1}{2}$-approximation for the max-min separation objective (Appendix B.5).

Table 2: Phoenix vs AdaBeam at equal evaluation budget (2000 evals). Representative subset of tasks shown; full results across all 12 tasks are in Appendix Fig. 4.

Task	Phoenix	AdaBeam	Improvement
ATAC	4.99 ± 0.39	3.66 ± 0.25	$+36\%$
CTCF	4.13 ± 0.22	2.71 ± 0.42	$+52\%$
E2F3	3.72 ± 0.11	2.47 ± 0.10	$+50\%$
GATA2	4.03 ± 0.26	2.37 ± 0.32	$+70\%$
MYC	1.98 ± 0.20	1.05 ± 0.34	$+89\%$
Average across all 12 tasks			$+\mathbf{43}\%$

2.2 BUILD: HIGH-THROUGHPUT DNA-TO-RNA MANUFACTURING

All data were generated under identical manufacturing and QC workflows, minimizing distribution shift between discovery and validation.

Pipeline Overview. Designs are automatically translated to manufacturable assemblies via Fragmogrifier, which decomposes sequences into synthesis-compatible fragments optimized for Gibson assembly (Appendix C.1). Constructs are assembled via high-throughput protocols (up to 48-plex), validated through automated Opentrons OT-2 colony screening coupled to Oxford Nanopore sequencing, and transcribed via automated 96-well IVT with standardized QC (spectrophotometry, capillary electrophoresis).

Throughput. The platform achieves **96 constructs/day** from fragment receipt to transfection-ready RNA, with 2–6 week end-to-end turnaround. Quality standards: dsRNA $<0.01\%$ by mass, residual protein and DNA below detection limits.

Failure Mode Analysis. Automated screening revealed sequence-design relationships informing computational mitigations (Table 3): high-GC windows ($>75\%$ over 40bp) required $2.2\times$ more colonies screened; multi-fragment constructs required $2.2\times$ more screening than single-fragment. Mitigations (GC smoothing, secondary structure redesign) substantially improved build success.

Table 3: Build failure modes and screening burden (n=21 constructs)

Failure Mode	Colonies Screened	Fold Increase
High GC windows present	15.0	$2.2\times$
No high GC windows	6.9	
Multi-fragment	15.1	$2.2\times$
Single-fragment	6.8	

2.3 TEST: HIGH-THROUGHPUT EFFECT SPACE ASSAYS

Designed sequences were evaluated against wild-type controls, industry standards (Twist, GenScript, IDT), and leading foundation models (CodonTransformer (Outeiral & Deane, 2025), mDD-0 (Ginkgo Bioworks, 2025)) across four cell lines (HepG2, HEK293T, Jurkat, A549).

Dataset. Over **29,000 measurements** spanning: expression kinetics (8h, 24h, continuous), accelerated degradation stability, manufacturability (IVT yield, full-length product fraction), and functional readouts (GFP fluorescence, Cas9 editing).

Expression. All constructs—including wild-type controls, industry-optimized references (Twist, GenScript, IDT), and foundation model outputs (CodonTransformer, mDD-0)—were synthesized with N1-methylpseudouridine, shared identical $5'$ and $3'$ UTRs and poly(A) tails, and were manufactured, purified, and transfected under the same standardized pipeline described in Section 2.2. Top designs achieved **61.8% average improvement** over wild-type across cell lines: HepG2 $45.2\pm7.9\%$, HEK293T $63.8\pm15.5\%$, Jurkat $85.0\pm16.6\%$, A549 $53.0\pm11.1\%$ (Figure 1A). Cas9 expression improved 2-fold (8h) and 4-fold (24h), translating to **1.5-fold gene editing efficiency** improvement.

Stability. 2.9-fold median stability improvement (41-fold best design). Critically, our designs navigated the stability-expression trade-off, achieving **173-hour mRNA half-life** while preserving burst expression (Figure 1B). Stability correlated strongly with hydrolysis score ($p < 0.05$).

Parameter-Effect Relationships. CAI showed generalizable expression influence across cell lines (Table 7), with significant Min-CAI vs Max-CAI differences ($p < 0.001$ for HepG2, HEK293T, A549), weakest in Jurkat. MFE correlated with durable expression in HEK293T and Jurkat ($p < 0.001$) but not HepG2 ($p > 0.10$). However, linear regression achieved modest correlations (max $r = 0.546$; Table 10), indicating complex nonlinear interactions requiring learned models.

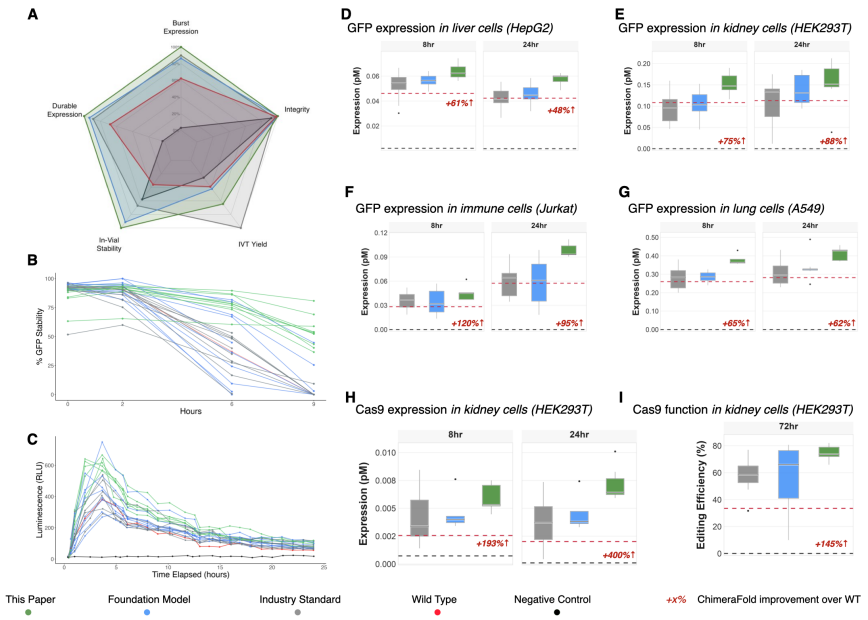

Figure 1: **Effect space outcomes.** (A) Overview radar plot showing highest performance across burst expression, durable expression, stability, IVT yield, and integrity. (B) GFP stability timecourse at 50°C. (C) 24-hour kinetics profiles. (D–G) Expression at 8h and 24h across cell lines. (H–I) Cas9 expression and editing efficiency. AI designs (green) vs. wild-type (red), industry standards (gray), and foundation models (blue). Replicates used in assays were as follows: stability assays (n=4), lytic expression assays (n=8), kinetic expression assays (n=4), functional assays (n=4)

2.4 LEARN: CONTRASTIVE ACTIVE LEARNING

The learn component enables closing the design-build-test loop by training discriminative models on experimental data and applying principled active learning to guide subsequent iterations.

Pairwise Discriminator. Fix a scalar assay target $y(\mathbf{s})$ (higher is better after preprocessing), and define the pairwise label

$$\mathbf{s}_1 \prec \mathbf{s}_2 \quad \Longleftrightarrow \quad y(\mathbf{s}_1) < y(\mathbf{s}_2).$$

Given sequences $\mathbf{s}_1, \mathbf{s}_2$, let $\{\mathbf{p}_{1,i}\}_{i=1}^{n_1}$ and $\{\mathbf{p}_{2,j}\}_{j=1}^{n_2}$ denote their K-codon sliding-window patches (indexed by codon position), and let $\mathbf{e}_i$ denote a learned position embedding. A patch-pair comparator predicts

$$P_\theta(\mathbf{p}_{1,i} \prec \mathbf{p}_{2,j}) = \sigma\big(a\, g_\theta\big(\phi(\mathbf{p}_{1,i}) \,\|\, \mathbf{e}_i,\ \phi(\mathbf{p}_{2,j}) \,\|\, \mathbf{e}_j\big) + b\big)\,, \tag{2}$$

where $\|$ denotes vector concatenation (not semiring addition), ϕ is a learned patch encoder, g_θ outputs a scalar logit, and (a, b) are Platt-scaling parameters fit on held-out comparisons. We aggregate patch-level probabilities into a sequence-level score via geometric-mean pooling:

$$\mu_\theta(\mathbf{s}_1, \mathbf{s}_2) := \exp\left(\frac{1}{n_1 n_2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \log P_\theta(\mathbf{p}_{1,i} \prec \mathbf{p}_{2,j})\right).$$

We treat $\mu_\theta(\mathbf{s}_1, \mathbf{s}_2)$ as a calibrated proxy for $\mathbb{P}[\mathbf{s}_1 \prec \mathbf{s}_2]$ and evaluate calibration on held-out data (e.g., ECE).

Acquisition Function. At round t, let $\mathcal{X}_t$ denote the finite candidate pool and let $\mathcal{S}_t$ be the set of already-acquired (wet-lab assayed) sequences. We form a reference set $\mathcal{R}_t \subseteq \mathcal{S}_t$ (e.g., the top-k by observed outcome for the current target) and train an ensemble of M calibrated discriminators. From the ensemble we compute a reference-averaged win-rate estimate $\hat{\mu}_t(\mathbf{x})$ and a high-probability confidence radius $c_t(\mathbf{x})$ (empirical Bernstein), both defined in Appendix E.2–E.3. We then select sequences using

$$\alpha_t(\mathbf{x}) = \hat{\mu}_t(\mathbf{x}) + c_t(\mathbf{x}) - \lambda_t \max_{\mathbf{s} \in \mathcal{S}_t} \mathrm{sim}(\psi(\mathbf{x}), \psi(\mathbf{s})), \tag{3}$$

where $\psi(\cdot)$ is a learned embedding and λ_t may be annealed across rounds.

Loop Closure. The framework directly parameterizes ChimeraFold: learned compositional preferences become nucleotide penalties $\mathbf{p}$; CAI-expression relationships inform λ ranges; structural motifs become pairing constraints; discriminator uncertainty modulates stochastic exploration. We note that the results reported herein reflect the platform's first design iteration; closed-loop active learning with iterative re-design informed by the contrastive discriminator is ongoing and will be reported in subsequent work.

3 Discussion

This platform addresses a fundamental limitation in mRNA therapeutics: the disconnect between computational proxies and therapeutic outcomes. The modest correlations we observed (max $r = 0.546$) confirm that simple linear relationships explain only a fraction of variance, motivating our active learning approach that learns complex interactions directly from multimodal data.

Why clonal builds over MPRAs? Massively parallel assays exclude N1-methylpseudouridine and other clinically essential modifications, and cellular production does not recapitulate IVT, purification, and formulation. Our platform trades raw throughput for therapeutically relevant molecules with defined QC.

Generalization. Beyond GFP and Cas9 reporters, the platform demonstrated robust generalization in external collaborations: >10-fold stability improvement for an infectious disease therapeutic candidate, and stability optimization enabling novel targeted delivery for a gene replacement program. However, because those studies are ongoing and some results cannot yet be disclosed in full, we treat them as qualitative evidence and focus the quantitative evaluation and ablations in this submission on the GFP and SpCas9 systems.

Limitations. Throughput remains constrained by synthesis costs; contrastive learning requires sufficient per-asset data (transfer learning in development); we focused on CDS optimization with UTR, poly(A), and chemical modification work not discussed here.

Availability. The software components (ChimeraFold, Fragmogrifier, contrastive learning framework) are proprietary. Experimental data underlying the main results will be deposited in a public repository upon publication. Additional implementation details will be provided in a public repository or supplementary material when release constraints permit. We welcome inquiries regarding collaboration.

Future Directions. Vec2Vec-style embedding alignment could unify sequence, structure, and functional embeddings into shared latent spaces. Integration with LNP optimization will extend toward end-to-end therapeutic design and is ongoing. Interpretability research may reveal novel biological mechanisms and inform novel design directions.

Conclusion. We demonstrated an integrated platform achieving substantial improvements in stability (2.9-fold median, 173h half-life), expression (61.8% average), and function (1.5-fold editing) through theoretically grounded sequence generation, high-throughput standardized manufacturing, and active learning-guided iteration providing a foundation for accelerating mRNA therapeutic development.

AUTHOR CONTRIBUTIONS

Blinded for review.

ACKNOWLEDGMENTS

We thank our co-founders, investors, and partners for their support of this work.

REFERENCES

Aikaterini Alexaki, Jacob Kames, David D. Holcomb, John Athey, Luis V. Santana-Quintero, Phuc Vihn Nguyen Lam, Nobuko Hamasaki-Katagiri, Ekaterina Osipova, Vahan Simonyan, Haim Bar, Anton A. Komar, and Chava Kimchi-Sarfaty. Codon and codon-pair usage tables (CoCoPUTs): Facilitating genetic variation analyses and recombinant gene design. *Journal of Molecular Biology*, 431(13):2434–2441, 2019.

John Athey, Aikaterini Alexaki, Ekaterina Osipova, Alexandre Rostovtsev, Luis V. Santana-Quintero, Upendra Katneni, Vahan Simonyan, and Chava Kimchi-Sarfaty. A new and updated resource for codon usage tables. *BMC Bioinformatics*, 18(1):391, 2017.

Ginkgo Bioworks. mDD-0: A foundation model for mRNA codon optimization. Technical report, Ginkgo Bioworks, 2025. URL `https://cms.ginkgo.bio/assets/resources/white-papers/2025-02-12-mdd-0/mdd-0-white-paper.pdf`. White paper.

Rune B. Lyngsø, Michael Zuker, and Christian N. Pedersen. Fast evaluation of internal loops in RNA secondary structure prediction. *Bioinformatics*, 15(6):440–445, 1999.

Shigeo Nagashima, Putu Prathiwi Primadharsini, Takashi Nishiyama, Masaharu Takahashi, Kazumoto Murata, and Hiroaki Okamoto. Development of a HiBiT-tagged reporter hepatitis E virus and its utility as an antiviral drug screening platform. *Journal of Virology*, 97(9):e00508–23, 2023. doi: 10.1128/jvi.00508-23.

Carlos Outeiral and Charlotte M. Deane. CodonTransformer: A multispecies codon optimizer using context-aware neural networks. *Nature Communications*, 15:5139, 2025.

Neville E. Sanjana, Ophir Shalem, and Feng Zhang. Improved vectors and genome-wide libraries for CRISPR screening. *Nature Methods*, 11(8):783–784, 2014.

Marie K. Schwinn, Thomas Machleidt, Kris Zimmerman, Christopher T. Eggers, Andrew S. Dixon, Robin Hurst, Mary P. Hall, Lance P. Encell, Brock F. Binkowski, and Keith V. Wood. CRISPR-mediated tagging of endogenous proteins with a luminescent peptide. *ACS Chemical Biology*, 13 (2):467–474, 2018. doi: 10.1021/acschembio.7b00549.

Paul M. Sharp and Wen-Hsiung Li. The codon adaptation index: A measure of directional synonymous codon usage bias, and its potential applications. *Nucleic Acids Research*, 15(3):1281–1295, 1987.

Joel Shor, Erik Strand, and Cory Y. McLean. NucleoBench: A large-scale benchmark of neural nucleic acid design algorithms. *bioRxiv*, 2025. doi: 10.1101/2025.06.20.660785. Preprint.

Goro Terai, Satoshi Kamegai, and Kiyoshi Asai. CDSfold: An algorithm for designing a protein-coding sequence with the most stable secondary structure. *Bioinformatics*, 32(6):828–834, 2016.

Max Ward, Mary Richardson, and Mihir Metkar. mRNA folding algorithms for structure and codon optimization. *Briefings in Bioinformatics*, 26(4):bbaf386, 2025.

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, 2023.

A FIGURES

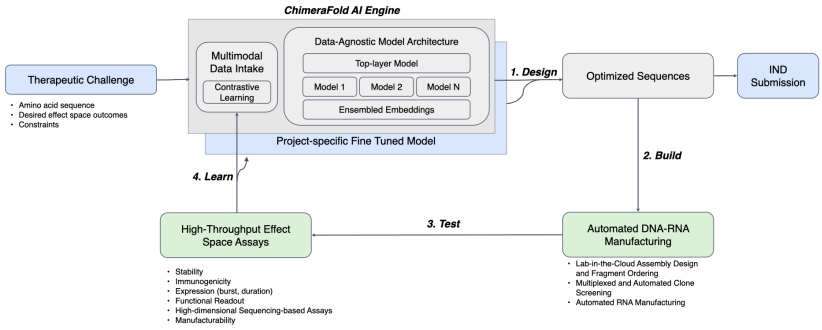

Figure 2: **Overview Schematic of Integrated Computational-Experimental Platform for Holistic mRNA Design.** A simplified overview of the design-build-test-learn workflow employed in the integrated computational-experimental platform described herein with the ultimate goal of solving therapeutic challenges through iterative rounds of active learning sequence design in preparation for future therapeutic investigation.

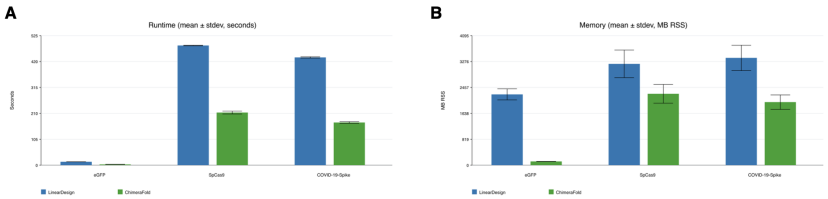

Figure 3: **ChimeraFold computational benchmarks versus LinearDesign.** Performance comparison across three representative coding sequences; benchmarks use *protein sequences only* (no UTRs), so the optimized region is the CDS. We report both amino-acid length and CDS length (nt; 3×AA, with a fixed 3-nt stop codon appended in constructs as needed): eGFP (284 aa; 852 nt + stop), SpCas9 (1,368 aa; 4,104 nt + stop), and SARS-CoV-2 Spike (1,273 aa; 3,819 nt + stop). **Top:** Peak memory usage (max RSS, MB). In Hydra-parallel mode, ChimeraFold reduces peak RSS by 1.42–18.83× (geometric mean 3.57×): eGFP 2,237 MB → 119 MB (18.83×), SpCas9 3,202 MB → 2,256 MB (1.42×), Spike 3,391 MB → 1,993 MB (1.70×). **Bottom:** Wall-clock runtime (seconds). ChimeraFold achieves 2.27–4.29× speedup (geometric mean 2.91×): eGFP 13.8 s → 3.2 s (4.29×), SpCas9 484.7 s → 213.2 s (2.27×), Spike 436.3 s → 172.8 s (2.52×). Error bars indicate standard deviation across three timed runs following one warmup run. Benchmarks performed on Apple silicon (arm64) with Rust 1.92.0 and Python 3.11.12.

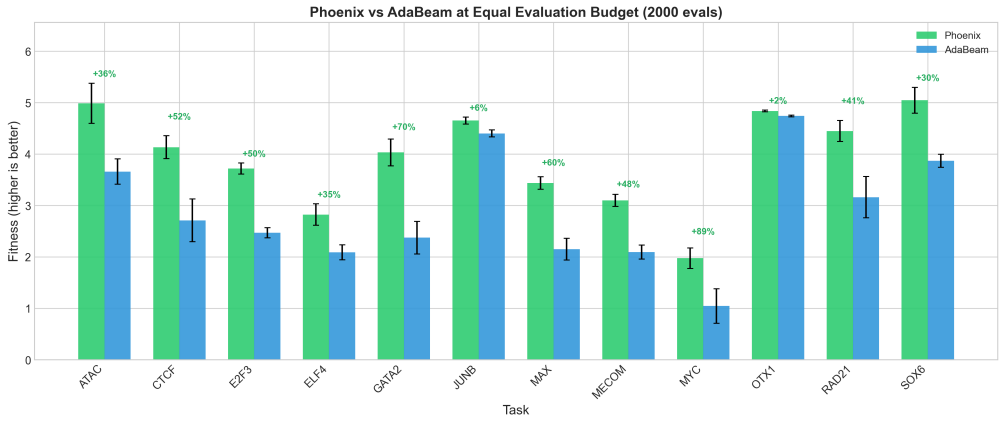

Figure 4: **Phoenix vs AdaBeam at equal evaluation budget.** Performance comparison across 12 BPNet transcription factor binding tasks from NucleoBench. Both optimizers were given identical evaluation budgets (2000 oracle queries) and hyperparameters. Phoenix (green) achieves 43% higher fitness on average, with improvements ranging from +2% (OTX1) to +89% (MYC). Error bars indicate standard deviation across 5 random seeds.

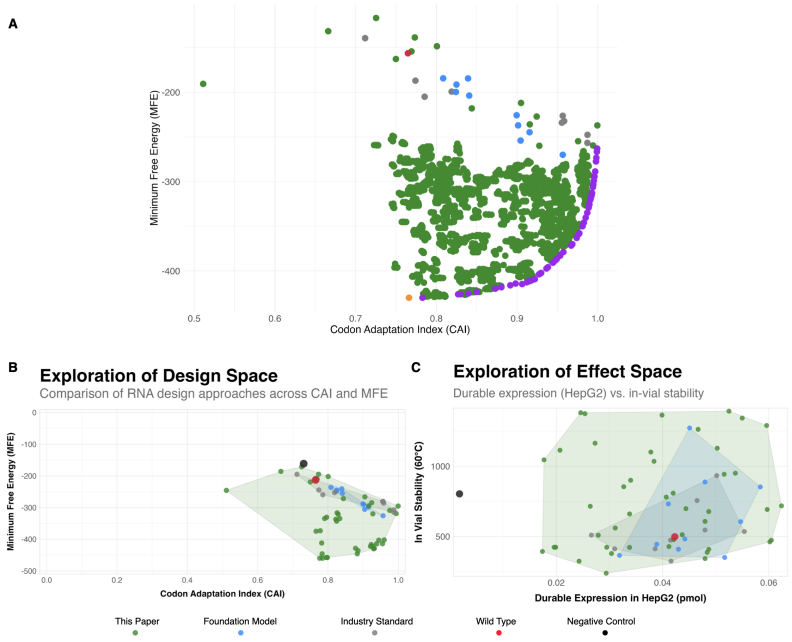

Figure 5: **Sequence Space Exploration.** (A) A representation of sequences generated and analyzed in this study across the CAI-MFE Pareto frontier, including designs produced from our platform (green), state-of-the-art foundation models (blue), industry standard algorithms and vendors (grey), wild-type sequences (red), and a HiBit-tag null control (black). Purple sequences model the Pareto frontier using a sweep of the λ parameter in LinearDesign and the orange sequence is derived from CDSFold (Terai et al., 2016) which only optimizes for MFE. Our sequences show a unique ability to explore both the Pareto frontier and internal landscape to fully explore sequence space. This full exploration is further exemplified in (B) which subsets to a representative set of sequences used in the described design-build-test-learn platform. As noted, this exploration of sequence and metric space allowed for a greater level of exploration in effect space (C) in this case comparing in vial stability against durable expression in HepG2.

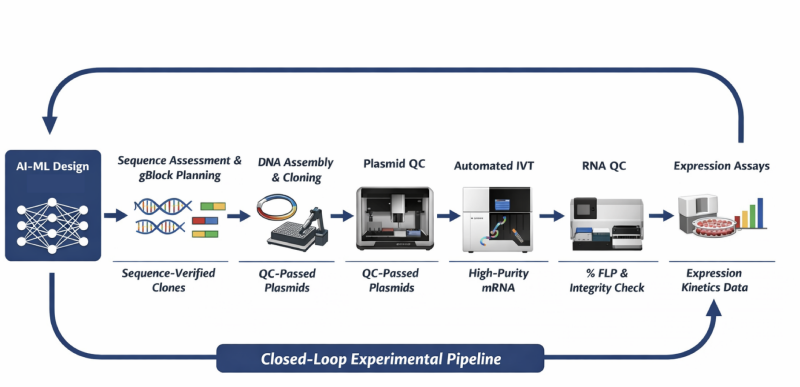

Figure 6: **Closed-loop experimental pipeline for scalable RNA design–build–test.** Computational sequence proposals are translated into manufacturable DNA assemblies via automated fragment planning and ordering, followed by high-throughput assembly and cloning. Clone identity is confirmed using an automated Opentrons OT-2 workflow that generates multiplexed ONT sequencing libraries and performs reference-guided sequence verification. Sequence-confirmed clones are advanced to plasmid preparation and QC (including sequence verification and poly(A) confirmation where required), followed by automated IVT and high-throughput mRNA quality control (e.g., purity and full-length product fraction). QC-passed mRNA is evaluated in downstream expression assays, and resulting functional data are used to inform subsequent design rounds.

B DESIGN METHODS

B.1 CHIMERAFOLD ARCHITECTURE

ChimeraFold's constraint system is DP-integrated, not post-hoc.

Semiring view of edge weights (MFE vs. Boltzmann). ChimeraFold expresses its dynamic programs over a commutative semiring $(\mathcal{S}, \oplus, \otimes, 0_{\mathcal{S}}, 1_{\mathcal{S}})$: extending a partial path by an edge of weight $w_e \in \mathcal{S}$ uses $\otimes$ (path extension), while combining alternative paths uses $\oplus$ (path aggregation). Concretely: (i) **MFE mode** uses the tropical (min-plus) semiring $(\mathbb{R} \cup \{+\infty\}, \min, +, +\infty, 0)$, so edge weights represent additive energy contributions and hard constraints are encoded as $+\infty$; (ii) **Boltzmann mode** uses the sum-product semiring $(\mathbb{R}_{\geq 0}, +, \times, 0, 1)$, so edge weights represent multiplicative Boltzmann factors (e.g., $w_e = \exp(-E_e/RT)$) and hard constraints are encoded as 0. User-facing soft penalties $q \in [0, 1]$ are interpreted as multiplicative factors in Boltzmann mode (so $q = 0$ is a hard ban); when the same penalty is used in MFE mode, it corresponds to an additive energy offset $-RT \log q$ for $q \in (0, 1]$ (with $q = 0$ interpreted as $+\infty$).

Additional structural controls are exposed through pair-bonus operations to encourage or discourage base pairs and/or regions to pair. These are resolved in CDS-relative coordinates with UTR offsets to enable full-sequence structural design while enforcing UTR preservation. The codon usage pipeline ingests tables from FDA HIVE-CUTs (Athey et al., 2017) as custom frequency priors.

Architecturally, ChimeraFold is written in pure Rust with explicit `SAFETY` invariants for pointer-backed views enabling parallel DP fills. The DP core uses Lyngsø internal-loop acceleration (Lyngsø et al., 1999) with a dedicated 4D Lyngsø table, and a thread-local internal-loop energy cache that is heap-allocated to avoid stack pressure. The DP computes exact MFE under the Turner nearest-neighbor energy model with Lyngsø's $O(n^3)$ internal-loop acceleration, which is exact for the standard loop decomposition (no sparsification or beam pruning is applied). Rayon parallelism is used for table fills and concurrent suboptimal rounds, and the CLI enforces strict error contracts (config parsing, codon rules, motif validation). This design yields an exact, multi-objective sequence generator that is deliberately engineered to explore the edge and non-edge interior regions of sequence space at scale. This is an essential prerequisite for engineering sequence diversity and enabling iterative design.

B.2 CHIMERAFOLD BENCHMARKING

Benchmark setup. We benchmarked ChimeraFold against LinearDesign using protein sequences only (no UTRs), so both methods optimize the CDS; we used eGFP (284 aa), SpCas9 (1368 aa), and SARS-CoV-2 Spike (1273 aa). Each tool/sequence pair was run with 1 warmup followed by 3 timed runs. Wall-clock runtime was measured with `time.perf_counter()` around the child process. Peak memory is max RSS from `os.wait4` (macOS reports bytes), converted to MB.

Hardware and software. Benchmarks were executed on a local Apple-silicon macOS workstation (arm64, Darwin 25.2.0). Software versions: Rust 1.92.0 (Homebrew), Cargo 1.92.0 (Homebrew), Python 3.11.12.

Results. On three representative CDS targets:

1. ChimeraFold achieved 2.27×–4.29× wall-clock speedup over LinearDesign, with a geometric-mean speedup of 2.91×.
2. ChimeraFold reduced peak memory footprint, using 1.42×–18.83× less RSS than LinearDesign across these sequences (geometric-mean reduction 3.57×).
3. The novel objective results in a 6.22× hypervolume expansion resulting in a 522% increase in novel candidate sequences generated.

B.3 PHOENIX: ADAPTIVE BEAM SEARCH FOR SAMPLE-EFFICIENT SEQUENCE OPTIMIZATION

Phoenix is a novel adaptive beam search algorithm designed for sample-efficient optimization of nucleic acids. Unlike AdaBeam, which uses fixed exploration-exploitation trade-offs, Phoenix em-

ploys three key innovations that together achieve approximately 4× better sample efficiency across diverse optimization objectives.

Algorithm Overview. Phoenix maintains a beam of high-scoring sequences and performs improvement-gated rollouts from a diverse subset of roots selected via maximum marginal relevance (MMR). The algorithm consists of three phases per iteration:

1. Root Selection. From the current beam $\mathcal{B}_t$, we select $k = \lfloor \rho \cdot |\mathcal{B}_t| \rfloor$ root sequences using MMR with diversity parameter $\lambda_{\text{root}} = 0.35$. We construct the root set greedily; at each step S denotes the set of roots already selected in the current iteration (initialized as $S = \emptyset$), and $\text{sim}(x, y) \in [0, 1]$ is a chosen similarity measure.

$$\text{MMR}(x) = \lambda_{\text{root}} \cdot \hat{F}(x) + (1 - \lambda_{\text{root}}) \cdot \left(1 - \max_{y \in S} \text{sim}(x, y)\right)$$

where $\hat{F}(x)$ is the normalized fitness and $\text{sim}(x, y)$ is sequence similarity.

2. Rollout Expansion. For each root, we perform r parallel rollouts. Each rollout continues while child fitness exceeds parent fitness (improvement-gated), up to a maximum depth $d_{\max}$. Mutations are sampled according to:

$$k_t = \text{clip}\left(\text{round}(\text{scale}_t \cdot k_{\text{base}}), 1, L\right)$$

where scale_t adapts based on optimization progress.

3. Beam Update. All candidates (roots, rollout sequences, previous beam) are merged and the top $|\mathcal{B}|$ sequences are selected via MMR with an annealed diversity parameter:

$$\lambda_t = (1 - t/T) \cdot \lambda_{\text{init}} + (t/T) \cdot \lambda_{\text{final}}$$

transitioning from exploration-dominant ($\lambda_{\text{init}} = 0.55$) to exploitation-dominant ($\lambda_{\text{final}} = 0.85$).

Plateau-Triggered Adaptation. When optimization progress stalls for p consecutive rounds (no improvement in best fitness), Phoenix adapts its search strategy:

$$\text{scale}_{t+1} = \min(\text{scale}_{\max}, \alpha \cdot \text{scale}_t), \quad d_{\max} \leftarrow \min(d_{\text{cap}}, d_{\max} + 1)$$

This increases mutation radius and rollout depth, enabling escape from local optima. Upon improvement, the scale factor partially resets: $\text{scale}_t \leftarrow \max(1.0, \text{scale}_t/\sqrt{\alpha})$.

Hyperparameters. Default values used in all experiments: beam size $|\mathcal{B}| = 20$, rollouts per root $r = 4$, base mutations $k_{\text{base}} = 2$, root fraction $\rho = 0.25$, initial rollout depth $d_{\max} = 10$, plateau patience $p = 5$, adaptation factor $\alpha = 1.5$, maximum scale $\text{scale}_{\max} = 5.0$, depth cap $d_{\text{cap}} = 20$.

B.3.1 Phoenix Benchmarking

Benchmark Setup. We evaluated Phoenix against AdaBeam, the prior state-of-the-art optimizer from NucleoBench (Shor et al., 2025), on 12 BPNet transcription factor binding prediction tasks. Both optimizers used identical hyperparameters: beam size 20, 2.0 mutations per sequence, 4 rollouts per root, and batch size 8. Critically, both optimizers were given **equal evaluation budgets of 2000 oracle queries** to ensure fair comparison of sample efficiency. Each configuration was run with 5 random seeds (0–4), starting from randomly initialized 3000bp DNA sequences.

Hardware. Experiments were conducted on Apple silicon (arm64, Darwin 25.2.0) with Python 3.11. BPNet evaluations averaged ∼10ms each, with total benchmark runtime of approximately 8 hours for all 120 runs.

Results. Phoenix achieved 43% higher fitness on average across all 12 tasks, winning 100% of benchmarks (Figure 4). Improvements ranged from +2% (OTX1, JUNB) to +89% (MYC), with 7 of 12 tasks showing >35% improvement. The largest gains occurred on tasks where AdaBeam struggled to escape local optima within the evaluation budget, validating Phoenix's plateau-triggered adaptation mechanism.

B.4 Constraint Scrubbing Algorithm

Following initial sequence generation, candidates undergo a constraint-satisfaction scrubbing pipeline that iteratively applies synonymous codon substitutions to eliminate negative sequence or

structural elements while preserving the encoded protein. Let $\mathcal{S}$ denote the input sequence and $\mathcal{M} = \{m_1, \ldots, m_k\}$ the set of prohibited motifs. We define the scrubbing objective:

$$\min_{\mathcal{S}' \in \text{Syn}(\mathcal{S})} |\{(i, m) : m \in \mathcal{M}, \mathcal{S}'[i : i + |m|] = m\}|$$

subject to:

$$\text{GC}_{\text{global}}(\mathcal{S}') \leq 0.70, \quad \text{GC}_{\text{local}}(\mathcal{S}', w) \in [0.30, 0.75] \; \forall w$$

where $\text{Syn}(\mathcal{S})$ is the set of synonymous variants and w indexes a specified sliding window.

The motif set $\mathcal{M}$ spans multiple categories: (i) restriction sites for cloning compatibility; (ii) T7 promoter/pause sequences that interfere with transcription; (iii) homopolymeric runs creating synthesis complexity; and (iv) translation regulatory elements. Multi-pattern detection uses an Aho-Corasick automaton achieving $O(|\mathcal{S}| + \sum_m |m| + \#\text{matches})$ complexity versus naive $O(|\mathcal{S}| \cdot |\mathcal{M}|)$.

For hydrolysis-prone positions, we integrate a DS-XGB gradient boosting model that predicts per-nucleotide inline cleavage susceptibility. Hotspots exceeding threshold τ are repaired via concentric shell expansion: for each hotspot at codon c_i, we search synonymous substitutions at radius $r \in \{0, 1, \ldots, r_{\max}\}$ (i.e., codons $c_{i-r}, \ldots, c_{i+r}$), accepting the first swap that eliminates the hotspot without violating other constraints. This locality-prioritized search typically resolves 90% of hotspots within $r \leq 2$.

B.5 Diversity-Based Sequence Selection

From the scrubbed candidate pool $\mathcal{C}$, we select n sequences to maximize *max-min* diversity in a normalized metric space. Let $\mathbf{x}_i \in \mathbb{R}^d$ denote the feature vector for sequence i:

$$\mathbf{x}_i = \left(\widetilde{\text{GC}}_i, \widetilde{\text{CAI}}_i, \widetilde{\text{EFE}}_i, \widetilde{\text{AUP}}_i, \widetilde{H}_i\right),$$

where each metric is min–max normalized across $\mathcal{C}$ and H denotes the full hydrolysis score. Define the pairwise distance $d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|_2$.

Objective (max-min dispersion). For a set $S \subseteq \mathcal{C}$, define its minimum separation

$$\Delta(S) := \min_{\substack{i,j \in S \\ i \neq j}} d_{ij}.$$

We aim to solve

$$\max_{S \subseteq \mathcal{C}',\, |S| = n} \Delta(S),$$

where $\mathcal{C}' \subseteq \mathcal{C}$ is an optional filtered pool (e.g., CAI bounds, no prohibited motifs, no hotspots).

Selection Algorithm (farthest-first / max-min).

1. $\mathcal{C}' \leftarrow \{c \in \mathcal{C} : \Phi(c) = \text{True}\}$ (Filter, optional)
2. $(s_1, s_2) \leftarrow \arg\max_{i,j \in \mathcal{C}'} d_{ij}$ (Initialize with farthest pair)
3. $S \leftarrow \{s_1, s_2\}$
4. **while** $|S| < n$:
$$s^* \leftarrow \arg\max_{c \in \mathcal{C}' \setminus S} \min_{s \in S} d_{c,s}, \qquad S \leftarrow S \cup \{s^*\}$$
5. **return** S

Approximation guarantee. Assume $d(\cdot, \cdot)$ is a metric on $\mathcal{C}'$ (true for ℓ_2). Let $S^\star$ denote an optimal size-n solution and let $\text{OPT} := \Delta(S^\star)$.

Proposition. The farthest-first procedure returns S such that $\Delta(S) \geq \text{OPT}/2$.

Proof. Consider any intermediate selected set S with $|S| < n$. We claim there exists a point $x \in S^\star$ such that $d(x, S) \geq \text{OPT}/2$, where $d(x, S) := \min_{s \in S} d(x, s)$. If not, then for every $x \in S^\star$ there

exists $s(x) \in S$ with $d(x, s(x)) < \mathrm{OPT}/2$. Since $|S| < |S^\star| = n$, by pigeonhole there exist distinct $x_1, x_2 \in S^\star$ with $s(x_1) = s(x_2) =: s$. By the triangle inequality,

$$d(x_1, x_2) \le d(x_1, s) + d(s, x_2) < \mathrm{OPT}/2 + \mathrm{OPT}/2 = \mathrm{OPT},$$

contradicting the definition of OPT as the minimum pairwise distance in $S^\star$. Therefore some $x \in S^\star$ satisfies $d(x, S) \ge \mathrm{OPT}/2$. The greedy rule chooses s^* maximizing $d(c, S)$, hence $d(s^*, S) \ge \mathrm{OPT}/2$. This implies every newly added point is at distance at least $\mathrm{OPT}/2$ from all previously selected points, and thus $\Delta(S) \ge \mathrm{OPT}/2$. □

Empirically, we additionally validated that this selection yields near-maximal coverage of the feature-space convex hull on small pools where brute-force evaluation is feasible.

C BUILD METHODS

C.1 FRAGMOGRIFIER: AUTOMATED FRAGMENT PLANNING

Given a target sequence $\mathcal{S}$ of length L and a prescribed overlap length $\delta \in \mathbb{Z}_{>0}$, Fragmogrifier chooses a fragment count k and integer boundary indices

$$0 = s_1 < e_1 < e_2 < \cdots < e_k = L, \qquad s_{i+1} = e_i - \delta \ \ (i = 1, \ldots, k-1).$$

Fragment i is the substring $f_i := \mathcal{S}[s_i : e_i)$ (half-open indexing). These constraints imply coverage $\bigcup_{i=1}^{k} [s_i, e_i) = [0, L)$ and an overlap of exactly δ bases between consecutive fragments, since

$$[s_i, e_i) \cap [s_{i+1}, e_{i+1}) = [e_i - \delta, e_i) \quad \Rightarrow \quad \big|[s_{i+1}, e_i)\big| = \delta.$$

We additionally require $e_1 \geq \delta$ (so that $s_2 \geq 0$) and $e_i - s_i \geq 1$ for all i (non-empty fragments).

Algorithm:

1. Generate candidate split points via boundary + interior sampling
2. Enumerate k-fragment combinations ($k \leq 4$) with fixed overlap δ
3. Score each combination by manufacturability and assembly complexity
4. Select minimum-cost set with 100% coverage
5. Iteratively refine complex fragments until convergence

Fragment complexity is assessed via vendor API integration (GC content, homopolymer runs, repeat density), with constructs classified as STANDARD, COMPLEX, or HIGH difficulty. The scoring function penalizes non-viable fragments ($+5 \times 10^5$), suboptimal sizes ($+2.5 \times 10^5$), and excessive fragment count, driving convergence toward 2–3 fragment assemblies in the [500, 3200] bp sweet spot.

For vendor routing, we model the DNA supply network as a graph and employ competitive bidding, given fragment specifications and vendor constraints, to identify the cost- and turnaround-optimal synthesis strategy across providers.

C.2 COLONY SCREENING WORKFLOW

To validate clone identity at scale, we developed an Opentrons OT-2 robotic workflow for ultra-high-throughput colony screening coupled to Oxford Nanopore Technologies (ONT) sequencing. The automated protocol separates cellular debris from genetic material, enriches plasmid DNA relative to genomic background using rolling circle amplification, and generates 96-plex ONT sequencing libraries via automated fragmentation, barcoding, pooling, and SPRI-based cleanup. The workflow supports variable throughput (8–96 samples per library; 1–2 libraries per run) by dynamically adjusting dead volumes and pipetting strategies via scripted automation, enabling a practical screening capacity of up to $\sim$192 clones per day under routine operation, with higher theoretical capacity limited by sequencing throughput. Across the dataset, a median of 8 clones were screened per construct, though challenging designs occasionally required substantially deeper screening to identify a sequence-correct clone.

C.3 PLASMID QUALITY CONTROL WORKFLOW

Following clone selection, plasmids were prepared in high-throughput formats (up to 96 constructs/day), using either automated magnetic-bead low-endotoxin miniprep workflows for smaller-scale needs or vacuum-manifold-based workflows for higher-yield preparations. Plasmid identity and integrity were assessed using reference-guided assembly against the intended design sequence, with additional poly(A) tail length verification performed via outsourced Sanger sequencing where required.

C.4 AUTOMATED IVT WORKFLOW

We developed an automated OT-2-based in vitro transcription (IVT) workflow to produce mRNA in 96-well format, addressing key constraints of low-cost robotics and conventional IVT chemistry

(e.g., limited agitation & temperature control, limited automated plate sealing, and deck capacity). These limitations were mitigated through development of automation strategy to prevent evaporative volume loss through periodic water addition during bioreaction incubation, an additive buffering system that accommodated plasmid linearization, IVT, DNase treatment, proteinase K treatment, and initiation of precipitation-based purification, and the use of isothermal processes. For larger-scale mRNA requirements (e.g., >100 μg per construct), we used a parallel manual IVT process in larger reaction vessels to support agitation-dependent reactions; this maintained common reagent mixes and modest hands-on time ($\sim$45 minutes/day) while reducing batch size capacity (e.g., $\sim$24 constructs/day versus 96-well automated operation).

C.5 High-Throughput mRNA Quality Control

Final mRNA quality was evaluated in high-throughput plate format using spectrophotometry and capillary electrophoresis (Agilent Fragment Analyzer), with release decisions based on predefined acceptance criteria including purity ratios and full-length product fraction (%FLP). All QC data and batch records for other processes were stored in cloud-based electronic lab notebooks. Additional in-depth QC was performed on a subset of the mRNA generated to measure dsRNA (consistently $<0.01\%$ by mass), residual protein (below LoD) and residual DNA (below LoD).

C.6 Failure Mode Analysis

Analysis of the plasmid builds identified several sequence- and assembly-driven failure modes that contributed to increased build complexity and multiple rounds of iteration. Constructs containing localized high-GC windows ($>75\%$ GC over 40 bp), particularly in larger designs such as SpCas9, were more susceptible to DNA synthesis errors, amplification challenges, and reduced assembly efficiency, requiring nearly twice as many colonies to be screened compared to constructs without high-GC regions (15.0 vs. 6.9 colonies on average). These risks were mitigated through targeted GC smoothing, codon optimization, and localized sequence refactoring to disrupt high-GC regions while preserving amino acid sequence.

Larger constructs exhibited strong predicted secondary structure, as reflected by highly negative ORF folding free energy values, which increased the likelihood of cloning inefficiencies and recombination events; mitigation strategies included redesign to reduce stable secondary structures and optimization of construct architecture to improve overall plasmid stability. The presence of internal ribosome-like features further contributed to plasmid instability and reduced clone viability and was addressed through synonymous sequence modifications to disrupt unintended translation initiation motifs.

Finally, constructs requiring multiple ordered fragments exhibited higher build failure rates and substantially increased screening burden, requiring over $2.2\times$ more colonies to be sequenced compared to single-fragment designs (15.1 vs. 6.8 colonies); these risks were reduced by redesign to enable single-fragment synthesis where feasible, and early in silico manufacturability screening. Collectively, these mitigation strategies substantially improved plasmid build success, stability, and overall manufacturability while reducing downstream iteration.

C.7 Materials

Plasmid Generation. Once sequence designs were approved, fragments were ordered from third-party vendors (Twist, Ansa). Upon receipt, plasmid assembly reactions were performed using the NEB HiFi Assembly Kit and transformed into NEB Stable *E. coli* cells. After a 48-hour incubation period, colonies were randomly picked and sequenced. Reference-guided analysis of colony sequencing reads was performed to verify identity. From these, top candidate plasmids were selected for preparation using the ZymoPURE midi prep kit. These plasmids were then validated by plasmid and Sanger sequencing to confirm sequence integrity and poly(A) tail length prior to in vitro transcription (IVT).

RNA Generation. mRNA was generated from sequence-verified plasmid DNA templates via restriction digest linearization followed by T7 in vitro transcription (IVT). Plasmids were linearized using XbaI (New England Biolabs), and linearized DNA templates were used as input for IVT reactions. Transcription reactions were performed using T7 RNA polymerase and buffer (NEB) with

standard NTPs and co-transcriptional capping (Areterna) following proprietary protocols to generate purified RNA resuspended in nuclease-free water for downstream characterization and functional testing.

C.8 BUILD DATA TABLES

Table 4: mRNA Construct Sequence Metrics and Build Outcomes

Target	Length (bp)	High GC Win.	ORF EFE (kcal/mol)	Inv. Repeats	IRF Score	Frag. Ordered	Colonies Seq.	Outcome
GFP	953	0	-419.1	7	223	1	2	Success
GFP	953	11	-272.9	0	0	1	2	Success
GFP	953	0	-154.5	1	20	1	2	Success
GFP	953	0	-145.4	0	0	1	2	Success
GFP	953	2	-268.9	0	0	1	2	Success
GFP	953	0	-229.9	0	0	1	1	Success
GFP	953	0	-177.7	1	22	1	2	Success
GFP	953	0	-172.1	0	0	1	2	Success
GFP	953	0	-205.1	1	20	1	4	Success
GFP	953	0	-202.3	1	20	1	4	Success
GFP	953	0	-234.2	0	0	1	40	Difficult
GFP	953	2	-270.4	0	0	1	19	Difficult
SpCas9	4625	5	-1137.8	7	152	4	3	Difficult
SpCas9	4625	52	-565.6	3	65	4	8	Difficult
SpCas9	4625	46	-1239.0	2	44	4	14	Difficult
SpCas9	4625	172	-1826.7	3	65	4	24	Failed
SpCas9	4625	44	-2679.9	25	871	2	22	Failed
SpCas9	4625	168	-1795.4	7	152	4	15	Failed
SpCas9	4625	10	-888.7	1	20	4	23	Failed
SpCas9	4625	35	-1376.6	0	0	4	13	Failed
SpCas9	4625	53	-1724.6	3	63	4	14	Failed

Length (bp) denotes the full mRNA insert length used for build/QC (5′ UTR + CDS + 3′ UTR), excluding the poly(A) tail. In contrast, the codon-optimization benchmarks (Appendix Fig. 3) operate on the CDS only (protein sequence input; no UTRs), so CDS lengths are reported separately there. Sequence-based failure mode metrics for 21 mRNA construct variants encoding GFP or SpCas9. High GC Windows calculated using 40 bp sliding windows (>75% GC threshold). ORF Folding Free Energy calculated using LinearPartition at 37°C (more negative = stronger secondary structure). Internal Ribosome Features (IRF) detected using Shine-Dalgarno motif recognition. Build outcomes: Success (n=10), Difficult (n=5), Failed (n=6).

Table 5: Failure Modes and Rates During Production

Category	Clone Screening Hit Rate	Plasmid QC Pass	IVT %FLP	IVT Yield
Robust designs	∼80%+	∼90%+	>95%	Near standard
Challenging designs	Very low	Low	≤60%	∼25% of standard

Comparative analysis of plasmid design categories throughout production pipeline. Challenging designs required up to 170 colonies screened to identify correct clones.

Table 6: Wet-Lab Throughput by Process Stage

Stage	Throughput	Primary Bottleneck
Assembly & cloning	∼96 constructs/day	Fragment synthesis cost
Clone screening	∼192 clones/day	ONT sequencing capacity
Automated IVT	96 constructs/day	Plate format
Manual IVT (high mass)	24 constructs/day	Agitation/incubation space
End-to-end TAT	2–6 weeks	Resource prioritization

Daily wet-lab process throughput capacity and identified bottlenecks. End-to-end turnaround time (TAT) variability driven primarily by resource allocation across concurrent projects.

D Test Methods

D.1 In Vial Stability and Manufacturability

In vial stability and manufacturability were performed in high-throughput using a 48-channel fragment analyzer and automated stability challenge to assess accelerated degradation over time. Data scraped from the build stage including cloning burden, colonies sequenced, IVT yield per reaction, and absorbance ratios were also factored into the manufacturability of each sequence.

D.2 Cells and Cell Culture Materials

Cell culture was performed using ATCC-recommended media for each cell line purchased from ATCC or Thermo with 10% FBS. Cell lines used in this study were HepG2 (ATCC, #HB-8065), HEK293T (ATCC, #CRL-3216), Jurkat E6-1 (ATCC, #TIB-152), and A549 (ATCC, #CRM-CCL-185).

D.3 Cell Transfection

Cells were seeded in 96-well plates 12–24 hours prior to transfection at optimized densities per cell line. mRNA transfections (100 ng per well) were performed using Lipofectamine MessengerMAX (Invitrogen) according to manufacturer protocols with cell line-specific reagent ratios. All experiments included four technical replicates per construct.

D.4 Stable Cell Line Generation

HepG2, HEK293T, and A549 cells were stably transfected with pCMV-LgBIT vector (Promega) using Lipofectamine 3000 in 6-well format. Cells were selected with Hygromycin B and maintained under selection pressure. Stable integration was confirmed by functional mRNA transfection assays prior to experimental use.

D.5 HiBit Lytic Assay

Protein expression levels were quantified at 8 and 24 hours post-transfection using the Nano-Glo HiBit Lytic Detection System (Promega). Cells were lysed and luminescence measured using a SpectraMAX i3x plate reader (1500 ms integration). Standard curves using purified LgBIT protein enabled conversion to molar concentrations. Data were normalized using reference constructs to control for plate-to-plate variation.

D.6 HiBit Kinetic Assay

Real-time expression kinetics were monitored in stable LgBIT-expressing cell lines using Vivazine substrate (Promega) with hourly luminescence measurements over 24 hours. Results are reported as relative luminescence units (RLU) (Nagashima et al., 2023) (Schwinn et al., 2018).

D.7 Gene Editing Assay

Functional assessment employed co-transfection of SpCas9 mRNA (100 ng) and CD47-targeting sgRNA (5 ng). After 72 hours, genomic DNA was extracted using QuickExtract solution, and target regions were PCR-amplified using locus-specific primers. Editing efficiency was determined by Sanger sequencing analysis using the ICE CRISPR Analysis Tool (Synthego) (Sanjana et al., 2014).

D.8 Additional RNA Analysis

dsRNA was quantified using the Revvity TR-FRET assay per the manufacturer's recommendations. Residual protein was quantified using a Qubit instrument and the ThermoFisher Qubit Protein Assay. Residual DNA was quantified using a qPCR assay with an amplicon for our kanamycin resistance marker.

D.9 Statistical Analysis

All experiments included biological replicates across multiple cell lines with technical replicates ($n \geq 4$). Outliers exceeding 40% deviation from replicate means were excluded from analysis. Data scaling and normalization procedures controlled for inter-plate variability prior to downstream machine learning applications. Parameter min-max statistical tests were performed with Welch's t-tests assuming unequal variances and Cohen's d for effect size assessment.

D.10 Statistical Tables

Table 7: Statistical Significance of CAI Across Expression Assays

Cell Line	Time	Pair	Min vs Max	Min vs WT	Max vs WT
A549	24hr	CAI	9.79e-05	0.0356	1.55e-07
A549	8hr	CAI	9.80e-07	0.00299	1.06e-05
HEK293T	24hr	CAI	6.94e-06	0.238	6.15e-06
HEK293T	8hr	CAI	2.48e-05	0.313	4.78e-06
HepG2	24hr	CAI	7.00e-06	3.50e-04	2.71e-04
HepG2	8hr	CAI	2.10e-07	0.0102	8.97e-07
Jurkat	24hr	CAI	7.66e-04	0.00767	8.08e-04
Jurkat	8hr	CAI	0.00305	0.0372	0.0133

Welch's t-test p-values comparing expression between wild-type, minimum-CAI, and maximum-CAI designed sequences across cell lines and timepoints.

Table 8: Statistical Significance of GC Content Across Expression Assays

Cell Line	Time	Pair	Min vs Max	Min vs WT	Max vs WT
A549	24hr	GC	6.51e-07	1.28e-05	2.05e-05
A549	8hr	GC	2.37e-05	0.00259	1.41e-04
HEK293T	24hr	GC	2.45e-06	9.37e-05	8.00e-04
HEK293T	8hr	GC	2.23e-05	2.44e-05	9.18e-04
HepG2	24hr	GC	1.70e-08	2.30e-05	6.29e-05
HepG2	8hr	GC	1.20e-08	7.60e-06	3.92e-05
Jurkat	24hr	GC	6.70e-08	1.80e-05	1.10e-06
Jurkat	8hr	GC	0.00118	5.85e-05	0.00649

Welch's t-test p-values comparing expression between wild-type, minimum-GC, and maximum-GC designed sequences across cell lines and timepoints.

Table 9: Statistical Significance of MFE Across Expression Assays

Cell Line	Time	Pair	Min vs Max	Min vs WT	Max vs WT
A549	24hr	MFE	0.0552	0.359	0.00919
A549	8hr	MFE	0.0597	0.357	0.0159
HEK293T	24hr	MFE	1.93e-04	0.0834	0.00154
HEK293T	8hr	MFE	3.22e-04	0.00179	0.0481
HepG2	24hr	MFE	0.607	5.24e-04	0.0558
HepG2	8hr	MFE	2.23e-04	0.00225	0.00972
Jurkat	24hr	MFE	6.77e-08	6.38e-06	0.938
Jurkat	8hr	MFE	8.58e-04	3.49e-05	0.943

Welch's t-test p-values comparing expression between wild-type, minimum-MFE, and maximum-MFE designed sequences across cell lines and timepoints.

Table 10: Linear Regression Analysis: Parameter-Expression Correlations

Predictor	Outcome	Correlation (r)	p-value
GC Content	HEK293T 24hr	0.295	< 0.05
GC Content	A549 24hr	0.321	< 0.05
CAI	HepG2 8hr	0.546	< 0.001
CAI	HepG2 24hr	0.499	< 0.001
CAI	HEK293T 8hr	0.425	< 0.001
CAI	HEK293T 24hr	0.259	< 0.05
CAI	Jurkat 8hr	0.386	< 0.01
CAI	Jurkat 24hr	0.408	< 0.001
CAI	A549 8hr	0.531	< 0.001
CAI	A549 24hr	0.424	< 0.001

Correlation coefficients from linear regression analyses. Only statistically significant correlations shown. Note that MFE showed no significant linear correlations across assays, consistent with complex nonlinear parameter interactions discussed in main text.

E Learn Methods

E.1 Pairwise Patch Discriminator Architecture

We train a discriminator D_θ to predict pairwise ordering between sequences for a fixed scalar target $y(\mathbf{s})$ (higher is better after preprocessing). We write $\mathbf{s}_1 \prec \mathbf{s}_2$ iff $y(\mathbf{s}_1) < y(\mathbf{s}_2)$.

For a sequence $\mathbf{s}$, let $\{\mathbf{p}_i\}_{i=1}^{n(\mathbf{s})}$ be its K-codon sliding-window patches (indexed by codon position), and let $\mathbf{e}_i$ be a learned position embedding. A learned encoder ϕ maps patches to vectors, e.g. $\phi : \Sigma_{\text{cod}}^K \to \mathbb{R}^d$ where Σ_{cod} is the codon alphabet.

For a patch pair $(\mathbf{p}_{1,i}, \mathbf{p}_{2,j})$, we predict

$$P_\theta(\mathbf{p}_{1,i} \prec \mathbf{p}_{2,j}) = \sigma\big(a\, g_\theta\big(\phi(\mathbf{p}_{1,i}) \,\|\, \mathbf{e}_i,\ \phi(\mathbf{p}_{2,j}) \,\|\, \mathbf{e}_j\big) + b\big)\,,$$

where $\|$ denotes vector concatenation, g_θ outputs a scalar logit, and (a, b) are Platt-scaling parameters fit on held-out comparisons.

Sequence-level comparisons aggregate over all patch pairs via geometric-mean pooling:

$$\mu_\theta(\mathbf{s}_1, \mathbf{s}_2) = \exp\left(\frac{1}{n(\mathbf{s}_1)\, n(\mathbf{s}_2)} \sum_{i=1}^{n(\mathbf{s}_1)} \sum_{j=1}^{n(\mathbf{s}_2)} \log P_\theta(\mathbf{p}_{1,i} \prec \mathbf{p}_{2,j})\right).$$

E.2 Calibration-Aware Acquisition Function

We define a confidence-bound acquisition rule derived from an ensemble of calibrated pairwise discriminators.

Setup. Let $\mathcal{X}$ denote the candidate pool at round t and let $\mathcal{S}_t$ be the set of acquired (wet-lab assayed) sequences. Let $\mathcal{R}_t \subseteq \mathcal{S}_t$ be a reference set (e.g., top-k by observed outcome for the current target). We train an ensemble of M discriminators $\{D_{\theta_t^{(m)}}\}_{m=1}^M$ using independent random seeds and/or bootstrap resampling of the training comparisons, and apply Platt scaling on held-out comparisons for probability calibration.

Each ensemble member induces a sequence-level comparison score $\mu_t^{(m)}(\mathbf{r}, \mathbf{x}) \in (0, 1)$ for the event "$\mathbf{x}$ outperforms $\mathbf{r}$" via patch extraction and geometric-mean pooling (Appendix E.1). Define the reference-averaged win-rate score

$$Z_t^{(m)}(\mathbf{x}) := \frac{1}{|\mathcal{R}_t|} \sum_{\mathbf{r} \in \mathcal{R}_t} \mu_t^{(m)}(\mathbf{r}, \mathbf{x}) \in [0, 1].$$

We then define the ensemble mean and sample variance

$$\hat{\mu}_t(\mathbf{x}) := \frac{1}{M} \sum_{m=1}^{M} Z_t^{(m)}(\mathbf{x}), \qquad \hat{v}_t(\mathbf{x}) := \frac{1}{M-1} \sum_{m=1}^{M} \Big(Z_t^{(m)}(\mathbf{x}) - \hat{\mu}_t(\mathbf{x}) \Big)^2 .$$

Exploration bonus as a confidence radius. We use an empirical-Bernstein radius (stated and proved in Appendix E.3):

$$c_t(\mathbf{x}) := \sqrt{\frac{2\, \hat{v}_t(\mathbf{x})\, \log\Big(\frac{4|\mathcal{X}_t|\pi^2 t^2}{\delta}\Big)}{M}} + \frac{7\, \log\Big(\frac{4|\mathcal{X}_t|\pi^2 t^2}{\delta}\Big)}{3(M-1)}, \qquad (M \geq 2).$$

Acquisition. We combine exploitation, exploration, and diversity via

$$\alpha_t(\mathbf{x}) = \hat{\mu}_t(\mathbf{x}) + c_t(\mathbf{x}) - \lambda_t \max_{\mathbf{s} \in \mathcal{S}_t} \mathrm{sim}(\psi(\mathbf{x}), \psi(\mathbf{s})),$$

where $\psi(\cdot)$ is a learned embedding and λ_t may be annealed across rounds. In batch mode, we greedily select B sequences using maximum marginal relevance (Appendix E.5).

E.3 Confidence Bounds and a Lightweight UCB Guarantee

This section states explicit assumptions under which the exploration term $c_t(\mathbf{x})$ is a valid high-probability confidence radius, and gives a lightweight near-optimality guarantee for the resulting UCB-style acquisition rule.

Definitions. Fix a round t and candidate pool $\mathcal{X}$. Let $\mathcal{R}_t \subseteq \mathcal{S}_t$ be the reference set. Define the (reference-averaged) *true win probability*

$$p_t(\mathbf{x}) := \frac{1}{|\mathcal{R}_t|} \sum_{\mathbf{r} \in \mathcal{R}_t} \mathbb{P}\big[\mathbf{x} \text{ outperforms } \mathbf{r}\big],$$

where the probability is with respect to the stochasticity in the assay/noise model used to define the pairwise label. Our discriminator ensemble produces $Z_t^{(m)}(\mathbf{x}) \in [0, 1]$ and $\hat{\mu}_t(\mathbf{x})$ as defined in Appendix E.2.

Assumptions.

1. **(Ensemble conditional independence and boundedness)** For each fixed $(t, \mathbf{x})$, the random variables $\{Z_t^{(m)}(\mathbf{x})\}_{m=1}^M$ are conditionally i.i.d. given the training data at round t, and satisfy $Z_t^{(m)}(\mathbf{x}) \in [0, 1]$ almost surely.
2. **(Calibration bias bound)** There exists $\varepsilon_{\mathrm{cal}} \geq 0$ such that for all $(t, \mathbf{x})$,

$$\Big| \mathbb{E}\Big[Z_t^{(m)}(\mathbf{x}) \,\Big|\, \text{training data at round } t \Big] - p_t(\mathbf{x}) \Big| \leq \varepsilon_{\mathrm{cal}}.$$

   (In practice, $\varepsilon_{\mathrm{cal}}$ is controlled empirically via held-out Platt scaling and calibration diagnostics.)

Lemma (uniform empirical-Bernstein bound over $(t, \mathbf{x})$). Assume $M \geq 2$ and Assumption 1. Define

$$c_t(\mathbf{x}) := \sqrt{\frac{2\,\hat{v}_t(\mathbf{x})\,\log\!\left(\frac{4|\mathcal{X}_t|\pi^2 t^2}{\delta}\right)}{M}} + \frac{7\,\log\!\left(\frac{4|\mathcal{X}_t|\pi^2 t^2}{\delta}\right)}{3(M-1)}.$$

Then with probability at least $1-\delta$ over the ensemble randomness (conditioning on the round-t training data), the following holds simultaneously for all $t \geq 1$ and all $\mathbf{x} \in \mathcal{X}_t$:

$$\left|\hat{\mu}_t(\mathbf{x}) - \mathbb{E}\left[Z_t^{(m)}(\mathbf{x}) \,\middle|\, \text{training data at round } t\right]\right| \leq c_t(\mathbf{x}).$$

Proof. Fix $(t, \mathbf{x})$. By Assumption 1, the variables $\{Z_t^{(m)}(\mathbf{x})\}_{m=1}^M$ are conditionally i.i.d. in $[0,1]$ given the training data at round t. Applying the empirical Bernstein inequality of Maurer and Pontil (Theorem 4) to $Z_t^{(m)}(\mathbf{x})$ gives a one-sided bound on the mean in terms of the sample mean and sample variance; applying the same inequality to $1 - Z_t^{(m)}(\mathbf{x})$ yields the corresponding lower tail, and a union bound gives the two-sided deviation bound with the same functional form.

Set the per-pair failure probability to $\delta_{t,\mathbf{x}} := \delta/(|\mathcal{X}_t|\pi^2 t^2)$ and allocate half to each tail, so that each application uses $\delta_{t,\mathbf{x}}/2$. This replaces $\log(2/\cdot)$ by $\log(4/\delta_{t,\mathbf{x}}) = \log\!\left(\frac{4|\mathcal{X}_t|\pi^2 t^2}{\delta}\right)$. Finally, a union bound over $\mathbf{x} \in \mathcal{X}_t$ and $t \geq 1$ and the identity $\sum_{t\geq 1} t^{-2} = \pi^2/6$ yield an overall failure probability at most δ. □

Corollary (confidence interval for $p_t(\mathbf{x})$). Under Assumptions 1–2 and the high-probability event in the Lemma, we have for all $(t, \mathbf{x})$:

$$p_t(\mathbf{x}) \in \left[\hat{\mu}_t(\mathbf{x}) - c_t(\mathbf{x}) - \varepsilon_{\text{cal}},\ \hat{\mu}_t(\mathbf{x}) + c_t(\mathbf{x}) + \varepsilon_{\text{cal}}\right].$$

Theorem (lightweight UCB near-optimality). Let

$$\mathbf{x}_t^{\text{UCB}} \in \arg\max_{\mathbf{x}\in\mathcal{X}} \left(\hat{\mu}_t(\mathbf{x}) + c_t(\mathbf{x})\right), \qquad \mathbf{x}_t^\star \in \arg\max_{\mathbf{x}\in\mathcal{X}} p_t(\mathbf{x}).$$

On the same high-probability event as above, the UCB-selected point satisfies

$$p_t(\mathbf{x}_t^{\text{UCB}}) \geq p_t(\mathbf{x}_t^\star) - 2c_t(\mathbf{x}_t^{\text{UCB}}) - 2\varepsilon_{\text{cal}}.$$

Proof. By the Corollary applied to $\mathbf{x}_t^\star$,

$$p_t(\mathbf{x}_t^\star) \leq \hat{\mu}_t(\mathbf{x}_t^\star) + c_t(\mathbf{x}_t^\star) + \varepsilon_{\text{cal}} \leq \hat{\mu}_t(\mathbf{x}_t^{\text{UCB}}) + c_t(\mathbf{x}_t^{\text{UCB}}) + \varepsilon_{\text{cal}}.$$

By the Corollary applied to $\mathbf{x}_t^{\text{UCB}}$ and rearranging,

$$\hat{\mu}_t(\mathbf{x}_t^{\text{UCB}}) \leq p_t(\mathbf{x}_t^{\text{UCB}}) + c_t(\mathbf{x}_t^{\text{UCB}}) + \varepsilon_{\text{cal}}.$$

Combining,

$$p_t(\mathbf{x}_t^\star) \leq p_t(\mathbf{x}_t^{\text{UCB}}) + 2c_t(\mathbf{x}_t^{\text{UCB}}) + 2\varepsilon_{\text{cal}},$$

which is equivalent to the stated bound. □

E.4 Stopping Criterion

We define a stopping rule directly in terms of the confidence radii $c_t(\mathbf{x})$ and the UCB near-optimality guarantee in Appendix E.3.

Sets. At round t, let $\mathcal{X}$ denote the finite candidate pool under consideration, let $\mathcal{S}_t$ denote the set of already-acquired (wet-lab assayed) sequences, and let $\mathcal{R}_t \subseteq \mathcal{S}_t$ denote the reference set used to define the win probability $p_t(\mathbf{x})$ and the estimated score $\hat{\mu}_t(\mathbf{x})$ (Appendix E.2). Let $\delta \in (0,1)$ be the global failure probability used in the confidence bound.

Rule. Define the uniform uncertainty level

$$u_t := \max_{\mathbf{x}\in\mathcal{X}} c_t(\mathbf{x}),$$

where $c_t(\mathbf{x})$ is the empirical-Bernstein radius in Appendix E.3. We stop when

$$u_t \leq \varepsilon_{\text{stop}},$$

for a user-chosen tolerance $\varepsilon_{\text{stop}} > 0$.

Guarantee. On the high-probability event from Appendix E.3 (probability at least $1 - \delta$ over ensemble randomness), the UCB-selected candidate $\mathbf{x}_t^{\mathrm{UCB}}$ satisfies

$$p_t(\mathbf{x}_t^{\mathrm{UCB}}) \geq \max_{\mathbf{x} \in \mathcal{X}} p_t(\mathbf{x}) - 2u_t - 2\varepsilon_{\mathrm{cal}} \;\geq\; \max_{\mathbf{x} \in \mathcal{X}} p_t(\mathbf{x}) - 2\varepsilon_{\mathrm{stop}} - 2\varepsilon_{\mathrm{cal}}.$$

Thus, once $u_t \leq \varepsilon_{\mathrm{stop}}$, the selected candidate is within $2\varepsilon_{\mathrm{stop}} + 2\varepsilon_{\mathrm{cal}}$ of the best achievable reference-averaged win probability over the current candidate pool.

E.5 Batch Selection via Maximum Marginal Relevance

For batch selection of size B, we greedily construct $\mathcal{S}_{t+1}$ to maximize coverage:

$$\mathbf{x}^* = \mathrm{argmax}_{\mathbf{x} \in \mathcal{X} \setminus \mathcal{S}_{t+1}^{\text{partial}}} \left[\eta \cdot \alpha_t(\mathbf{x}) + (1 - \eta) \cdot \min_{\mathbf{s} \in \mathcal{S}_{t+1}^{\text{partial}}} d(\psi(\mathbf{x}), \psi(\mathbf{s})) \right]$$

with η annealed from exploration-dominant ($\eta = 0.3$) to exploitation-dominant ($\eta = 0.9$) across iterations, ensuring early rounds explore sequence space broadly while later rounds concentrate on high-performing regions.

E.6 Loop Closure: Learn → Design

The active learning framework closes the design-build-test loop by directly parameterizing ChimeraFold's configuration for subsequent iterations. Specifically:

1. Compositional preferences learned from high-performing sequences are encoded as nucleotide penalties $\mathbf{p} \in \mathbb{R}^4$, biasing the DP toward favorable A/U/C/G distributions.
2. Asset- and cell-line-specific relationships between CAI and expression inform the λ sweep range.
3. Structural motifs correlated with stability are translated into `encourage_unpaired` or `discourage_pair` constraints.
4. The discriminator ensemble's uncertainty, measured by the disagreement $\hat{v}_t(\mathbf{x})$ or the associated confidence radius $c_t(\mathbf{x})$ (Appendix E.3), modulates `subopt_randomness`. Regions where the ensemble is uncertain receive higher stochastic exploration, while confident regions receive more deterministic exploitation.

This tight coupling between the Learn and Design components ensures that each experimental iteration maximally reduces uncertainty and systematically improves therapeutic performance.

F Additional Discussion

F.1 Advantages of Clonal Builds Over Massively Parallel Approaches

A natural question is why we opted for targeted clonal construction rather than massively parallel reporter assays (MPRAs), which enable interrogation of vastly larger sequence libraries. The answer lies in therapeutic relevance. MPRAs of endogenous RNA exclude N1-methylpseudouridine and other clinically essential chemical modifications, and cellular production does not recapitulate the IVT, purification, and formulation steps that dominate real-world manufacturing. Moreover, we observed that non-deterministic library generation within synonymous codon space produces sequences that cluster tightly under key design parameters. 99.9995% of randomly sampled synonymous variants occupy a narrow region of the CAI-MFE landscape. Our platform strategically trades raw sequence throughput for the targeted design and construction of therapeutically relevant molecules with defined quality control at each step.

F.2 Limitations

Several limitations warrant discussion:

1. **Throughput constraints.** Our current experimental throughput, while substantially higher than traditional approaches, remains constrained by fragment synthesis costs and downstream screening capacity.
2. **Data requirements.** The contrastive learning framework requires sufficient experimental data per asset to learn meaningful representations; very early-stage programs may benefit from transfer learning approaches that we are actively developing.
3. **CDS focus.** While we have focused on CDS optimization, parallel workstreams on UTR design, poly(A) optimization, and chemical modification patterns are critically enabling for holistic sequence design but are not reported here.
4. **Model systems.** Our systematic evaluation focuses on GFP and Cas9 reporter systems, though external collaborations demonstrate generalization to therapeutic targets.

F.3 Future Directions

Looking forward, we see several promising research directions:

Multimodal foundation models. Vec2Vec-style embedding alignment could unify sequence embeddings from RNA language models, structural embeddings from folding predictors, and functional embeddings from expression and stability models into a shared latent space. This enables cross-modal retrieval, transfer learning, and more principled multi-objective optimization without requiring end-to-end co-training of disparate architectures.

End-to-end therapeutic design. Integration with high-dimensional RNA analytics including sequencing-based assessments of RNA structure and translation, lipid nanoparticle optimization and delivery modeling would extend the platform toward end-to-end therapeutic design, connecting sequence to formulation to biodistribution.

Interpretability. Interpretability research on our biological foundation models using sparse autoencoders and linear probing may reveal novel biological mechanisms and design hypotheses, enabling a self-improving system that informs its own advancement.

Expanded sequence elements. Parallel workstreams on UTR design, poly(A) optimization, and chemical modification patterns are critically enabling for holistic sequence design, not presented in this body of work, and represent natural extensions of the current platform.