mRNAWM: A Latent World Model for Multi-Objective Full-Length mRNA Design

Anonymous authors
Paper under double-blind review

Abstract

Designing full-length messenger RNA (mRNA) sequences requires balancing translation-related properties, RNA structure, sequence composition, and manufacturability while preserving the encoded protein. These objectives interact across the coding sequence (CDS) and untranslated regions (UTRs), meaning that locally beneficial edits do not necessarily yield globally optimal designs. We formulate full-length mRNA design as a long-horizon, multi-objective decision problem and introduce mRNAWM, a protein-conditioned latent world model for controllable mRNA optimization. mRNAWM encodes a full-length mRNA into a latent state and learns action-conditioned dynamics over protein-preserving sequence edits. By rolling these dynamics forward, the model predicts the multi-step consequences of alternative edit trajectories on mRNA properties and future utility. An uncertainty-aware model predictive controller plans over these predicted trajectories, while preference-independent dynamics allow the same world model to support different objective trade-offs. mRNAWM (Full) outperforms all external methods on seven of the eight primary metrics, while the CDS-only configuration achieves the strongest Kozak score. The full-mRNA configuration reaches a codon adaptation index (CAI) of 0.945 and a full-mRNA minimum free energy per nucleotide (MFE/nt) of -0.401. It further reduces homopolymer burden by 14.5% and uridine–guanosine (UG)-rich sequence burden by 19.1% relative to the strongest baseline for each metric. These results position latent world-model planning as a practical strategy for controllable biological sequence engineering under multiple interacting objectives.

1 Introduction

Messenger RNA (mRNA) has emerged as a validated therapeutic modality, with applications extending from infectious-disease vaccines to individualized cancer immunotherapy. The rapid development and large-scale deployment of mRNA vaccines during the COVID-19 pandemic established the advantages of the platform for rapid antigen design and scalable manufacturing (Pardi & Krammer, 2024). The platform has since expanded beyond COVID-19. Moderna developed mRESVIA, an mRNA vaccine against respiratory syncytial virus that has reached clinical use (U.S. Food and Drug Administration, 2024), and, together with Merck, developed intismeran autogene (formerly mRNA-4157/V940), an individualized neoantigen-encoding mRNA therapy that, in combination with pembrolizumab, has shown durable improvements in recurrence-free and distant metastasis-free survival in resected high-risk melanoma (Khattak et al., 2026). These developments highlight a central opportunity for mRNA medicines: once a therapeutic protein or antigen is selected, its encoding sequence can be engineered to improve properties relevant to therapeutic performance.

Sequence design is therefore an engineering problem for mRNA therapeutics. For a fixed protein, the coding sequence (CDS) admits an enormous space of synonymous variants due to genetic-code degeneracy, while the 5' and 3' untranslated regions (UTRs) are functional components that regulate translation and transcript stability (Fig. 1). The nucleotide sequence further determines RNA structure, local composition, regulatory motifs, and other properties associated with expression, stability, and manufacturability (Metkar et al., 2024; Zhang et al., 2023). These properties are coupled: improving one can degrade another, and a locally favorable substitution may alter long-range structure or change the value of subsequent edits. Joint UTR–CDS optimization is therefore naturally a multi-objective sequential design problem rather than a single codon-optimization task.

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

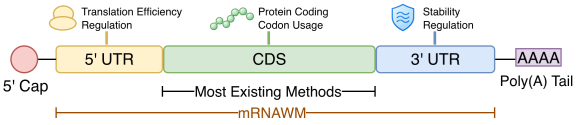


Figure 1: **Motivation for full-length mRNA design.** Most existing methods focus on the CDS, whereas mRNAWM jointly optimizes the CDS and UTRs.

Existing computational approaches address important parts of this problem. LinearDesign jointly optimizes codon usage and RNA secondary structure through dynamic programming (Zhang et al., 2023). Learning-based methods model contextual codon preferences or generate coding sequences, including CodonTransformer (Fallahpour et al., 2025), Trias (Faizi et al., 2025), and codonGPT (Rajbanshi & Guruacharya, 2025). GEMORNA extends generative modeling to coding and untranslated regions (Zhang et al., 2025), while CodonRL formulates synonymous sequence design as multi-objective reinforcement learning (Du et al., 2026). Despite this progress, these approaches generally generate a sequence directly or optimize under a prescribed search procedure, rather than learning reusable action-conditioned dynamics that can compare the downstream consequences of alternative edit trajectories.

We formulate mRNA optimization as a long-horizon decision problem over legal, protein-preserving edits in the CDS and UTRs. Because each edit can alter multiple coupled properties and change the value of later edits, greedy decisions based on immediate gains can be myopic. World models address this by learning action-conditioned latent dynamics that predict how candidate edits move the current design state. These dynamics can be rolled forward to imagine multi-step trajectories, compare their downstream consequences before committing an edit, and capture effects that emerge only after several steps. By replanning from the realized state after each edit, the model can further adapt its decisions as the sequence evolves (Hansen et al., 2024; Hafner et al., 2025).

We introduce **mRNAWM**, a protein-conditioned latent world model for joint UTR–CDS optimization. Rather than directly generating a final sequence or greedily ranking isolated edits, mRNAWM learns how protein-preserving edits change the design state and uses these dynamics to plan over multi-step futures. A single trained model supports different objective trade-offs at inference time, with uncertainty-aware planning over coupled mRNA properties. Our contributions are threefold:

- We formulate joint UTR–CDS mRNA optimization as a protein-preserving, multi-objective sequential decision problem in which individual edits can have coupled and delayed consequences.
- We introduce a protein-conditioned latent world model that separates reusable, preference-independent sequence dynamics from preference-conditioned control, together with an uncertainty-aware receding-horizon planner for multi-step mRNA optimization.
- Across CDS-only and full-mRNA modes, mRNAWM attains the best mean result on all eight primary metrics, with full-length optimization leading on seven.

These results suggest that latent world-model planning offers a practical framework for controllable biological sequence design, where multiple coupled objectives must be balanced.

2 Related Work

Naturalness-Oriented mRNA Design. Learning-based mRNA design methods often model codon usage patterns observed in natural sequences. CodonTransformer learns host-specific, context-dependent codon preferences (Fallahpour et al., 2025), while Trias learns protein-conditioned codon patterns from large eukaryotic sequence collections (Faizi et al., 2025). These models provide sequence priors, but natural-sequence likelihood does not directly optimize therapeutic objectives such as translation, RNA structure, stability, sequence composition, and manufacturability. Accordingly, matching natural sequence statistics alone provides limited control over competing objectives.

Multi-Objective mRNA Optimization. Other methods optimize functional sequence properties more directly. LinearDesign jointly optimizes codon usage and RNA structure through dynamic programming (Zhang et al., 2023). codonGPT and CodonRL use reinforcement-learning objectives for synonymous sequence optimization (Rajbanshi & Guruacharya, 2025; Du et al., 2026), while GEMORNA extends learned design to both coding and untranslated regions (Zhang et al., 2025). These approaches broaden the objective space, but optimization remains tied to a search procedure, reward, generator, or policy. They do not learn reusable action-conditioned dynamics that can be rolled forward to compare multi-step edit trajectories before committing an action.

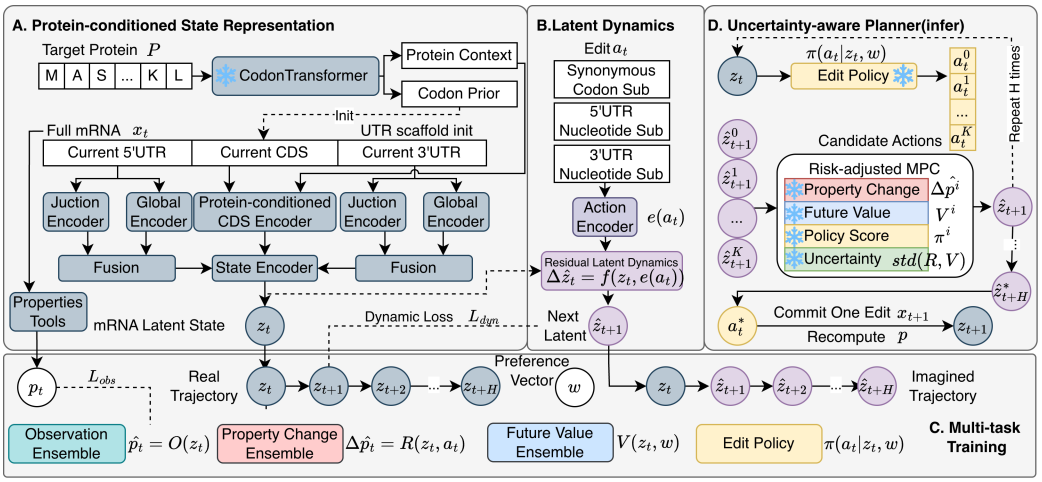


Figure 2: Overview of mRNAWM. (A) Protein-conditioned state representation of the 5'UTR–CDS–3'UTR transcript. (B) Decoder-free action-conditioned latent dynamics. (C) Multi-task training from real and recursively imagined trajectories. (D) Uncertainty-aware receding-horizon planning that proposes candidate edits, imagines their consequences, commits one edit, and replans.

World Models and Model-Based Planning. World models learn predictive representations that support decision making through imagined futures. DreamerV3 and UniZero demonstrate latent world-model learning and planning across control tasks (Hafner et al., 2025; Pu et al., 2025). More recent work emphasizes action-conditioned prediction and planning, including V-JEPA 2, IRASim, WorldGym, World Action Planner, and OSCAR (Assran et al., 2025; Zhu et al., 2025; Quevedo et al., 2026; Zhang & Du, 2026). Together, these studies motivate treating learned dynamics as internal simulators for evaluating future consequences before actions are executed. mRNAWM brings this model-based planning perspective to multi-objective full-length mRNA design.

3 Method

mRNAWM formulates full-length mRNA optimization as planning in a learned latent sequence-dynamics model. As illustrated in Fig. 2, the method consists of a protein-conditioned state representation, a preference-independent predictive world model, and a preference-conditioned controller that performs receding-horizon model predictive control (MPC).

3.1 Problem Formulation

Given a target protein sequence P, we optimize an mRNA $x_t = (U_t^{5'}, C_t, U_t^{3'})$ while requiring the coding sequence C_t to translate exactly to P. At step t, an action $a_t \in \mathcal{A}(x_t)$ is one of three legal modifications: a synonymous CDS substitution, a nucleotide substitution in the 5'UTR, or a nucleotide substitution in the 3'UTR. All edits preserve sequence length, and CDS edits preserve the encoded protein by construction. Each sequence is associated with $D = 18$ computable properties $\mathbf{p}_t = \mathbf{p}(x_t)$ spanning translation-related, RNA-structural, sequence-composition, and manufacturability objectives. A non-negative preference vector $\mathbf{w} \in \mathbb{R}_+^D$, $\mathbf{1}^\top \mathbf{w} = 1$, specifies their desired trade-off. After normalization and direction alignment, the preference-specific utility is $U_{\mathbf{w}}(x) = \mathbf{w}^\top q(\mathbf{p}(x))$. We denote the model input state by $s_t = (P, x_t, \mathbf{p}_t, \mathbf{m}_t)$, where $\mathbf{m}_t$ is a validity mask for the designated state-input properties.

3.2 mRNAWM: Predictive World Model and Controller

mRNAWM separates prediction of sequence dynamics from preference-specific control. Its predictive world model is $\mathcal{M}_\Theta = (h_\phi, f_\theta, O, R)$, where h_ϕ encodes the current protein-conditioned design state, f_θ predicts action-conditioned latent transitions, O predicts sequence properties from latent states, and R predicts vector-valued property changes induced by edits. None of these predictive components receives the preference vector $\mathbf{w}$.

The current latent state is $z_t = h_\phi(s_t)$. An edit a_t is mapped to an action embedding $e(a_t)$ using its edit type, position, and nucleotide or codon change. The residual transition model predicts

$$\Delta\hat{z}_t = f_\theta(z_t, e(a_t)), \qquad \hat{z}_{t+1} = \mathrm{LN}(z_t + \Delta\hat{z}_t).$$

Recursive application of the same transition model yields the imagined trajectory $z_t \to \hat{z}_{t+1} \to \hat{z}_{t+2} \to \cdots \to \hat{z}_{t+H}$, allowing alternative edit trajectories to be evaluated before any edit is committed to the real sequence. The latent states expose task-relevant consequences through $\hat{\mathbf{p}}_t = O(z_t)$ and $\Delta\hat{\mathbf{p}}_t = R(z_t, a_t)$. Rather than predicting only a task-specific scalar reward, R predicts a vector of biological property changes. The same learned sequence dynamics can therefore be evaluated under different preference vectors without retraining the predictive world model.

Preference-specific control is provided by a future-value model $V(z_t, \mathbf{w})$ and an edit policy $\pi(a_t \mid z_t, \mathbf{w})$. The policy prioritizes legal candidate edits, while the value model estimates continuation improvement beyond the imagined horizon. Final action selection is not performed directly by the policy; instead, MPC uses π, V, and the predictive world model to compare multi-step futures.

Unlike direct sequence generators that produce a completed mRNA, mRNAWM models how editable sequence states evolve under candidate actions. This supports comparison of multi-step futures before committing an edit and reuses the same predictive dynamics across design preferences.

3.3 Protein-Conditioned State Representation

As shown in Fig. 2A, the encoder h_ϕ represents the fixed protein target and the current editable transcript. The state combines protein-conditioned CDS information, complementary local and global UTR representations, designated state-input properties, and their validity mask. The property inputs are omitted from Fig. 2A for visual clarity; their definitions are provided in Appendix A.1.

Protein-conditioned CDS representation. CodonTransformer (Fallahpour et al., 2025) is frozen and evaluated once per target protein. Its protein representation and synonymous-codon prior are cached. The protein context is combined with embeddings of the codons to form a CDS representation capturing both what protein must be encoded and how it is currently encoded. The codon prior is not part of z_t; it is used for CDS initialization and as a weak proposal prior during inference.

Local and global UTR representation. The 5' and 3' UTRs are represented by complementary junction and global encoders. The junction branches model sequence context around the UTR–CDS boundaries, whereas the global branches summarize information across the complete UTRs. The resulting UTR representations are fused with the protein-conditioned CDS representation and state-input properties to obtain z_t. Detailed encoder architectures are given in Appendix A.2.

3.4 Offline Learning of Sequence Dynamics

Training uses offline trajectories of valid protein-preserving edits (Fig. 2C). Applying the recorded edits to nucleotide sequences and re-encoding each state produces the real latent trajectory $z_t, z_{t+1}, \ldots, z_{t+H}$. Starting from the same z_t, recursively applying the learned transition model to the same action sequence produces the imagined trajectory $z_t, \hat{z}_{t+1}, \ldots, \hat{z}_{t+H}$.

Latent dynamics. The transition model is trained to align each recursively imagined latent state with a stop-gradient encoding of the corresponding real future state:

$$\mathcal{L}_{\mathrm{dyn}} = \frac{1}{H}\sum_{k=1}^{H} d(\hat{z}_{t+k}, \mathrm{sg}(z_{t+k}))\,.$$

This directly supervises multi-step latent prediction rather than only one-step transitions.

Property prediction. The observation ensemble is trained on real and imagined latent states, encouraging imagined states to preserve interpretable property semantics. The property-change ensemble predicts $\mathbf{y}_t^{\Delta} = \kappa_\Delta(\tilde{\mathbf{p}}_{t+1} - \tilde{\mathbf{p}}_t)$ with $\kappa_\Delta = 50$; this scaling is inverted during planning.

Preference-conditioned control targets. For a sampled preference $\mathbf{w}$, the realized one-step utility improvement is $\delta_t^{\mathbf{w}} = U_{\mathbf{w}}(x_{t+1}) - U_{\mathbf{w}}(x_t)$. The future-value ensemble predicts discounted continuation improvement along the offline trajectory, while the edit policy predicts the selected legal edit. Thus, preference-independent targets train the representation, latent dynamics, observation, and property-change models, whereas preference-specific targets train the value and policy modules.

Algorithm 1 Uncertainty-Aware Latent MPC

Require: Protein P, current mRNA x_t, preference $\mathbf{w}$, horizon H, proposals K, beam width B
Evaluate x_t and encode the real state z_t, initialize beam $\mathcal{B} \leftarrow \{(x_t, z_t)\}$
for $h = 1, \ldots, H$ **do**
    Propose legal edits using $\pi(a \mid z, \mathbf{w})$ and the cached codon prior
    Apply candidate edits to obtain explicit candidate sequences
    Imagine next latents with $\Delta\hat{z} = f_\theta(z, e(a))$ and $\hat{z}' = \mathrm{LN}(z + \Delta\hat{z})$
    Evaluate candidate properties and model predictions
    Obtain $R(z, a)$, $V(\hat{z}', \mathbf{w})$, policy scores, and ensemble disagreement
    Score candidate branches with the risk-adjusted MPC objective and retain the top B
end for
Select the best imagined trajectory and let a_t^* be its first edit
Apply a_t^* to obtain x_{t+1} and recompute its properties
Re-encode the resulting real state z_{t+1}
if the real-state acceptance criterion fails **then**
    STOP
end if
Set $x_t \leftarrow x_{t+1}$ and repeat planning

Together, these objectives train a reusable predictive model of edit consequences and lightweight preference-conditioned control heads. The complete mRNAWM workflow is summarized in Appendix Algorithm 2, with training details in Appendix A.3.

3.5 Uncertainty-Aware Model Predictive Control

At inference time, the learned world model is used inside a receding-horizon controller (Fig. 2D; Algorithm 1). When an initial CDS is not supplied, it is constructed from the cached protein-conditioned codon prior, while the UTRs are initialized from valid scaffolds. Each planning iteration consists of proposal, recursive imagination, risk-adjusted scoring, and closed-loop replanning.

Proposal. Given the current real state z_t and preference $\mathbf{w}$, the edit policy proposes legal candidate actions $\{a_t^i\}_{i=1}^K$, with the cached codon prior providing an additional weak signal for CDS proposals. All proposed CDS edits remain synonymous, and UTR proposals preserve sequence length.

Recursive imagination. For each candidate action, the transition model predicts

$$\hat{z}_{t+1}^i = \mathrm{LN}\left(z_t + f_\theta(z_t, e(a_t^i))\right).$$

The dynamics are recursively applied for up to H imagined steps, while beam search retains the most promising partial trajectories. Because the transitions are action-conditioned, different edit sequences produce different latent futures that can be compared before the real transcript is modified.

Each branch also carries its explicit candidate sequence for legal-action enumeration and property evaluation. Decoder-free therefore means that recursive latent transitions do not require a learned nucleotide decoder to reconstruct intermediate sequences. Property evaluation during imagined rollouts combines sequence-derived quantities with world-model predictions; implementation details are provided in Appendix A.4.

Risk-adjusted MPC. Each imagined branch is scored using predicted property improvement, future value, policy score, and ensemble disagreement. For a trajectory τ, the planner score is

$$S(\tau) = G_{\mathbf{w}}(\tau) + \beta_V V(\tau, \mathbf{w}) + \beta_\pi \ell_\pi(\tau) - \beta_U \mathcal{U}(\tau),$$

where $G_{\mathbf{w}}$ is cumulative preference-weighted property improvement, ℓ_π is the policy score associated with branch expansion, and $\mathcal{U}$ is obtained from disagreement of the property-change and future-value ensembles. Constraints and the exact experimental score are detailed in Appendix A.4.

Commit and replan. After searching up to H imagined steps, the planner selects the highest-scoring trajectory but commits only its first edit a_t^*. The resulting sequence x_{t+1} is re-evaluated and encoded as the new real state $z_{t+1} = h_\phi(s_{t+1})$. If the acceptance criterion is satisfied, planning restarts from z_{t+1}; otherwise optimization terminates. Committing only the first action before replanning limits accumulation of latent-model error. Optimization thus proceeds as closed-loop receding-horizon control rather than open-loop sequence generation.

Table 1: **Comparison with mRNA design methods across eight primary design metrics.** Bold cells with darker shading indicate the best result; lighter shading indicates the second-best result.

	Translation		*RNA Structure*		*Manufacturability*			*Sequence Composition*
Method	**CAI ↑**	**Kozak ↑**	**MFE/nt ↓**	**Ens. Div. ↔ 0.10**	**Homopolymer ↓**	**Window GC ↓**	**uAUG ↓**	**UG-rich ↓**
CodonTransformer	0.926	0.625	−0.392	0.096	0.772	0.328	0.0062	0.0212
CodonRL	0.928	0.625	−0.391	0.104	0.830	0.350	0.0062	0.0194
LinearDesign	0.747	0.618	−0.357	0.011	0.761	0.333	0.0065	0.0228
codonGPT	0.804	0.625	−0.357	0.131	0.782	0.334	0.0062	0.0261
Trias	0.854	0.625	−0.388	0.111	0.760	0.353	0.0062	0.0244
GEMORNA (CDS)	0.881	0.680	−0.384	0.166	0.820	0.330	0.0062	0.0233
mRNAWM (CDS)	**0.948**	**0.770**	−0.394	0.113	**0.644**	0.321	0.0062	**0.0097**
GEMORNA (Full)	0.874	0.622	−0.394	0.116	0.792	0.355	0.0061	0.0229
mRNAWM (Full)	0.945	0.677	**−0.401**	**0.098**	0.650	**0.314**	**0.0054**	0.0157

4 Experiments

4.1 Experimental Setup

Dataset and protein-cluster split. We construct the corpus from vertebrate-mammalian RefSeq transcripts (Goldfarb et al., 2025), each containing a target protein, CDS, and associated 5' and 3' UTRs. After validation and exact deduplication, transcripts encoding the same protein are grouped into exact protein-sequence clusters and assigned by cluster to an 80/10/10 train/validation/test split, preventing identical target proteins from crossing partitions. All methods use the frozen test cohort and native transcript references. CDS-only outputs are paired with the native UTRs for full-mRNA evaluation, whereas mRNAWM also permits constrained UTR substitutions while preserving the target protein. Structural quantities are computed with the same ViennaRNA (Lorenz et al., 2011) pipeline. Dataset and evaluation details are provided in Appendices B.1 and B.2.

Baselines. We compare mRNAWM with six recent mRNA design methods: CodonTransformer (Fallahpour et al., 2025), CodonRL (Du et al., 2026), GEMORNA (Zhang et al., 2025), LinearDesign (Zhang et al., 2023), codonGPT (Rajbanshi & Guruacharya, 2025), and Trias (Faizi et al., 2025). These methods cover naturalness-oriented generation, dynamic programming, generative sequence modeling, and multi-objective reinforcement learning. A detailed comparison of their design spaces and optimization mechanisms is provided in Appendix B.3.

Evaluation metrics. We evaluate eight metrics spanning four aspects of mRNA design. For **translation**, we measure codon adaptation index (CAI) (Sharp & Li, 1987) and Kozak-context score (Kozak, 1987). For **RNA structure**, we use full-transcript minimum free energy per nucleotide and ensemble diversity per nucleotide, the latter evaluated by proximity to 0.10. For **manufacturability**, we measure homopolymer burden, maximum window-GC deviation, and upstream AUG density; for **sequence composition**, we report UG-rich density. CAI and Kozak score are maximized, ensemble diversity targets 0.10, and all remaining metrics are minimized. We report metrics separately rather than combining them into a single score; full definitions are in Appendix B.4.

mRNAWM configuration. Experiments use the validation-selected checkpoint and the final preference vector selected in the ablation study. The final planner uses horizon $H = 5$, beam width $B = 5$, $K = 128$ proposals per branch, and an edit budget of 60. We set the minimum risk-adjusted improvement threshold to -0.003, $\beta_{\mathrm{CT}} = 0.02$, $\beta_U = 0.5$, $\beta_V = 0.5$, $\beta_\pi = 0.01$, scaffold-value coefficient 0.25, and discount $\gamma = 0.97$. The CDS is initialized from the CodonTransformer prior, and UTR optimization is limited to four substitutions in total, with at most two in each UTR. Full training and inference configurations are given in Appendix A.3.10 and Appendix A.4.11.

4.2 Comparison with Existing mRNA Design Methods

Table 1 compares mRNAWM with six existing mRNA design methods across eight metrics spanning translation, RNA structure, manufacturability, and sequence composition. We evaluate both CDS-only and full-mRNA operating modes of mRNAWM and, where available, the CDS-only and full-

length configurations of GEMORNA. Across the eight metrics, an mRNAWM configuration attains the best mean result on every metric. Notably, mRNAWM (Full) outperforms all external methods on seven of the eight metrics, while mRNAWM (CDS) achieves the strongest Kozak score.

For translation properties, mRNAWM (CDS) achieves the highest CAI (0.948) and Kozak score (0.770). The full-length variant retains a high CAI of 0.945 while showing stronger performance on transcript-level properties. In particular, mRNAWM (Full) obtains the lowest full-mRNA MFE per nucleotide (-0.401) and an ensemble diversity of 0.098, closest to the prespecified target of 0.10.

The two operating modes show complementary behavior on manufacturability and sequence-composition metrics. mRNAWM (CDS) achieves the lowest homopolymer burden (0.644) and UG-rich density (0.0097), whereas mRNAWM (Full) achieves the lowest window-GC deviation (0.314) and upstream-AUG density (0.0054). The full-length configuration also remains competitive in homopolymer burden and UG-rich density (0.650 and 0.0157, respectively), while substantially improving the RNA-structure metrics.

Overall, the two mRNAWM configurations occupy different points in the multi-objective design space. Restricting edits to the CDS favors several translation, manufacturability, and sequence-composition properties, whereas allowing full-length editing provides additional flexibility to improve transcript-level RNA structure, local GC composition, and 5'UTR-associated properties. This behavior is consistent with the goal of mRNAWM: rather than optimizing a single fixed sequence criterion, the model uses learned multi-step sequence dynamics to navigate trade-offs among coupled properties.

4.3 Consistent Design Quality Across Source Species

We further examine whether mRNAWM maintains consistent design quality across proteins originating from different species. Rather than aggregating over the test set, we report mRNAWM performance separately for each source-species stratum in Table 2. The complete species-stratified comparison against all baselines is provided in Appendix B.5.

mRNAWM exhibits a stable translation-oriented profile across all 13 source species. CAI remains consistently high, ranging from 0.9342 to 0.9659, while the rare-codon fraction is at most 2.5×10^{-4} in every stratum. The same behavior extends across proteins originating from primates, rodents, and other mammalian species, indicating that the strong codon adaptation is not specific to human-derived proteins. Sequence-composition and manufacturability properties are similarly consistent. CpG density ranges from 0.0107 to 0.0250, and UpA density from 0.0084 to 0.0149 across species. Repeat-6 burden also remains within a relatively narrow range of 0.288–0.382. Together, these results show that mRNAWM maintains a consistent multi-property design profile across diverse protein sources rather than relying on a particular source-species composition.

4.4 Ablation Studies

Planning horizon. Performance improves markedly with multi-step lookahead and rapidly saturates as the planning horizon increases (Fig. 3A). Most of the gain is obtained by $H = 2$–3, followed by a

Table 2: **Performance of mRNAWM across source species.**

Source species	CAI ↑	Rare codon frac. ↓	CpG density ↓	UpA density ↓	Repeat-6 burden ↓
Callithrix jacchus	0.9496	0.00013	0.0164	0.0116	0.3230
Camelus ferus	0.9464	0.00009	0.0141	0.0119	0.3460
Cricetulus griseus	0.9659	0.00000	0.0107	0.0084	0.3813
Homo sapiens	0.9419	0.00010	0.0155	0.0118	0.3653
Ictidomys tridecemlineatus	0.9413	0.00019	0.0190	0.0111	0.3746
Leptonychotes weddellii	0.9471	0.00025	0.0148	0.0119	0.3169
Macaca mulatta	0.9488	0.00000	0.0176	0.0102	0.3816
Microcebus murinus	0.9431	0.00006	0.0165	0.0115	0.3694
Mus musculus	0.9415	0.00017	0.0140	0.0119	0.3636
Ochotona princeps	0.9342	0.00000	0.0250	0.0103	0.3486
Peromyscus maniculatus bairdii	0.9478	0.00003	0.0159	0.0116	0.3450
Rattus norvegicus	0.9493	0.00016	0.0128	0.0149	0.3339
Sus scrofa	0.9413	0.00000	0.0117	0.0098	0.2876

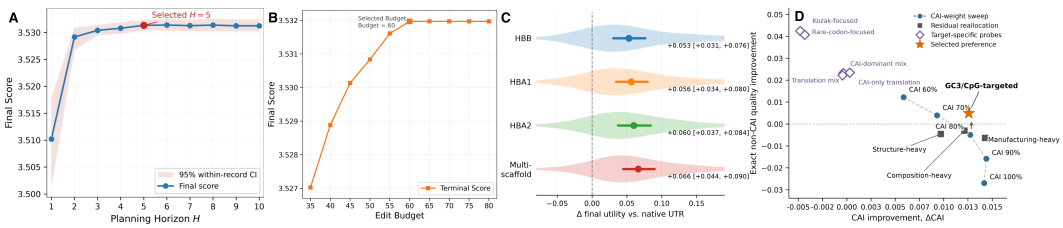


Figure 3: **Ablation studies of planning and inference-time design choices. (A)** Planning-horizon sweep, showing rapid gains from multi-step lookahead followed by saturation; $H = 5$ is selected. **(B)** Edit-budget sweep, where final performance saturates at a budget of 60. **(C)** Sensitivity to UTR initialization, reported as the change in final utility relative to native-UTR initialization for three globin scaffolds and target-specific multi-scaffold selection; points and horizontal bars denote mean improvements and 95% confidence intervals. **(D)** Inference-time preference calibration. Increasing CAI weight exposes a trade-off between CAI and the remaining objectives, while targeted redistribution of residual preference mass recovers non-CAI quality without sacrificing the CAI gain.

broad plateau from approximately $H = 5$ onward. The selected $H = 5$ configuration lies within this stable regime and improves over the shallower $H \leq 3$ settings, whereas deeper planning provides little additional benefit. We therefore use $H = 5$ as the default planning horizon.

Edit budget. Increasing the edit budget improves the final score (Fig. 3B), from 3.5270 at 35 edits to 3.5320 at 60, after which performance plateaus through 80 edits. We therefore use a budget of 60.

Robustness to UTR initialization. We test whether mRNAWM depends on a particular UTR initialization (Fig. 3C). Replacing native UTRs with RefSeq-derived globin scaffolds (Goldfarb et al., 2025) improves final utility by $+0.053$ for HBB (95% CI $[+0.031, +0.076]$), $+0.056$ for HBA1 ($[+0.034, +0.080]$), and $+0.060$ for HBA2 ($[+0.037, +0.084]$). The similar performance across fixed globin scaffolds indicates limited sensitivity to scaffold identity. Target-specific scaffold selection yields a further modest gain ($+0.066$, 95% CI $[+0.044, +0.090]$), selecting HBB, HBA1, and HBA2 for 26%, 26%, and 48% of targets, respectively. Overall, globin scaffolds provide a stronger initialization prior without making performance depend on a single scaffold.

Preference-vector calibration. Finally, we test whether mRNAWM can be steered at inference time by changing the preference vector while keeping the world model fixed (Fig. 3D). Increasing the CAI weight improves CAI but eventually degrades the remaining objectives, revealing a clear multi-objective trade-off. We therefore refine the CAI-80% setting by reallocating residual preference mass, and select a GC3/CpG-targeted vector that preserves the CAI gain while recovering positive non-CAI quality. This demonstrates controllable inference-time trade-offs without retraining.

4.5 Visualizing World-Model Planning

The central distinction of mRNAWM is that optimization proceeds through a learned world model rather than greedy mutation scoring. At each step, candidate edit sequences are imagined in latent

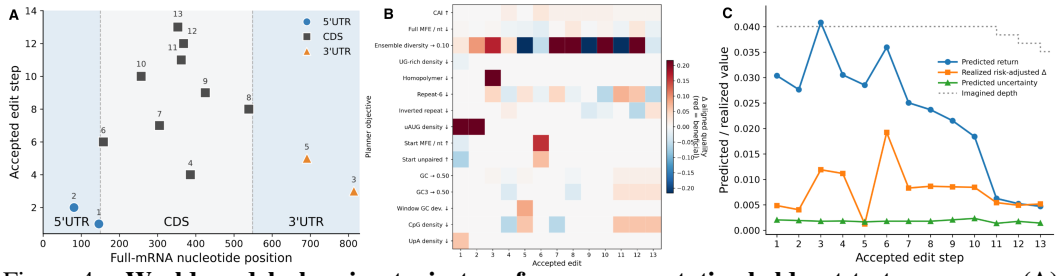


Figure 4: **World-model planning trajectory for a representative held-out test sequence. (A)** Accepted edits across the 5'UTR, CDS, and 3'UTR over the selected optimization trajectory. **(B)** Objective-aligned property changes induced by each edit; red indicates a beneficial change relative to the corresponding optimization direction, blue indicates an adverse change, and near-white indicates little change. **(C)** World-model planning signals for the same trajectory, showing predicted return, realized risk-adjusted improvement, predicted uncertainty, and imagined stopping depth.

Table 3: **Zero-shot transfer to tumor-associated antigens.** Metrics span codon adaptation, sequence composition, manufacturability, and translation-initiation accessibility.

Group	Metric	mRNAWM	CT	CodonRL	GEMORNA	Linear	codonGPT	Trias
Translation	CAI ↑	**0.9458**	0.9321	0.9334	0.8821	0.7467	0.7983	0.8861
Sequence Composition	CpG density ↓	0.0176	**0.0140**	0.0572	0.0587	0.0674	0.0463	0.0481
Manufacturability	Homopolymer burden ↓	**0.5457**	0.6079	0.6883	0.6581	0.6024	0.6296	0.6048
RNA Structure	Start unpaired fraction ↑	0.4657	0.4446	0.4633	**0.4680**	0.4119	0.4603	0.4469

space, ranked by predicted return and uncertainty, and only the first edit of the selected rollout is executed before replanning (Fig. 4). Figure 4A shows the resulting edit trajectory across the 5'UTR, CDS, and 3'UTR, illustrating that planning acts over the full transcript rather than a single region. Panel B shows the corresponding objective-aligned effects of each accepted edit. Individual edits need not improve every property simultaneously; instead, local gains and losses are balanced across steps, consistent with preference-aware multi-step planning. Panel C exposes the associated planning signals. Predicted return fluctuates across early edits before declining toward the endpoint, realized risk-adjusted improvements remain positive but vary across steps, and predicted uncertainty stays low throughout. The imagined stopping depth also shortens near termination as fewer beneficial future edits remain. Together, these panels illustrate the receding-horizon control loop in which imagined futures guide each edit and the plan is updated after every execution.

4.6 Zero-Shot Transfer to Tumor-Associated Antigens

We further evaluate zero-shot transfer on tumor-associated antigens (TAAs) curated from CEDAR (Koşaloğlu-Yalçın et al., 2023), providing a cancer-vaccine-relevant domain shift. Unlike the general benchmark, this setting tests whether a model trained on the broader mRNA design corpus can optimize previously unseen antigen targets without TAA-specific adaptation. Data curation and homology auditing are detailed in Appendix B.10. Without TAA-specific fine-tuning, mRNAWM achieves the highest CAI among all evaluated methods, improving from 0.9321 for CT initialization to 0.9458, while maintaining favorable composition. Its CpG density remains close to CT (0.0176 vs. 0.0140) and below the remaining generative and optimization baselines. mRNAWM also achieves the lowest homopolymer burden (0.5457), improving over CT (0.6079) and the strongest external baseline (0.6024), while increasing the start-unpaired fraction from 0.4446 to 0.4657.

These results show more than robustness to distribution shift: mRNAWM can transfer its learned design strategy to unseen, cancer-relevant antigen sequences while preserving the target protein and improving multiple mRNA properties without retraining. This target-conditioned transfer is particularly relevant to mRNA cancer vaccine design, where newly selected antigen sequences must often be converted rapidly into optimized mRNA constructs.

5 Conclusion

We introduced mRNAWM, a protein-conditioned latent world model for multi-objective full-length mRNA design. By learning action-conditioned latent dynamics and planning over imagined edits, mRNAWM moves beyond one-step mutation scoring and direct sequence generation. Across RefSeq evaluation, it achieves strong performance across translation, RNA structure, manufacturability, and sequence composition, while supporting inference-time preference control and transfer across source species and tumor-associated antigens.

This study remains computational and relies on sequence- and structure-based proxies rather than direct measurements of expression, stability, immunogenicity, or therapeutic efficacy. Moreover, the curated TAA collection contains few mutation-specific targets, so the zero-shot analysis primarily evaluates germline tumor-associated antigens rather than patient-specific neoantigens. Future work should incorporate experimental outcomes and broader mutation-specific antigen datasets to test whether world-model planning can optimize biologically realized mRNA behavior.

More broadly, our results frame biological sequence engineering as model-based decision making over coupled objectives. Learning reusable predictive dynamics while adapting control to design preferences provides a flexible path toward experimentally grounded, personalized mRNA design.

AI USE STATEMENT

In this work, we used generative AI tools to assist with interpretation of experimental results, manuscript drafting and editing, literature search, and partial code editing and command-line debugging. All AI-assisted outputs were reviewed by the authors before use. Result interpretations were checked against the underlying quantitative results, AI-assisted code was reviewed and tested for correctness, and literature-related outputs were verified against the original sources. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work is computational and does not involve new human-subject recruitment, clinical intervention, or wet-lab experimentation. Our experiments use public RefSeq transcript records and the public CEDAR resource. The reported results concern computational mRNA design properties and should not be interpreted as evidence of clinical efficacy, safety, or therapeutic benefit. Although the present study focuses on protein-preserving optimization of predefined therapeutic mRNA targets, more general sequence-optimization systems may raise dual-use concerns and should be accompanied by appropriate target selection, experimental validation, and biosafety review. Further discussion is provided in Appendix D.

REPRODUCIBILITY STATEMENT

We provide detailed information to support reproduction of both training and evaluation. Appendix A specifies the complete mRNAWM formulation, including property definitions and normalization, model architecture, offline trajectory construction, training objectives and hyperparameters, and the full inference and planning procedure. Appendix B describes dataset construction, the protein-level train/validation/test split, baseline configurations, and the shared cross-method evaluation protocol. All reported methods are evaluated with the same downstream property definitions and RNA-structure pipeline, and the major sources of randomness are controlled through fixed seeds or record-deterministic procedures. The final training configuration and checkpoint-selection criteria are reported in Appendix A.3, while the final inference and planner hyperparameters are reported in Appendix A.4.

REFERENCES

Mahmoud Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba, Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-jepa 2: Self-supervised video models enable understanding, prediction and planning, 2025. URL https://arxiv.org/abs/2506.09985.

Shiyi Du, Gün Kaynar, Jiayi Li, Zhaoyi You, Shijie Tang, and Carl Kingsford. CodonRL: Multi-objective codon sequence optimization using demonstration-guided reinforcement learning. *bioRxiv*, 2026.

Marjan Faizi, Helen Sakharova, and Liana F Lareau. A generative language model decodes contextual constraints on codon choice for mRNA design. *bioRxiv*, 2025.

Adibvafa Fallahpour, Vincent Gureghian, Guillaume J Filion, Ariel B Lindner, and Amir Pandi. CodonTransformer: a multispecies codon optimizer using context-aware neural networks. *Nature Communications*, 16(1):3205, 2025.

Tamara Goldfarb, Vamsi K Kodali, Shashikant Pujar, Vyacheslav Brover, Barbara Robbertse, Catherine M Farrell, Dong-Ha Oh, Alexander Astashyn, Olga Ermolaeva, Diana Haddad, et al. NCBI RefSeq: reference sequence standards through 25 years of curation and annotation. *Nucleic Acids Research*, 53(D1):D243–D257, 2025.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640(8059):647–653, 2025.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations (ICLR)*, 2024.

Adnan Khattak, Matteo S Carlino, Tarek Meniawy, George Ansstas, Matthew H Taylor, Kevin B Kim, Meredith McKean, Georgina V Long, Ryan J Sullivan, Mark Faries, et al. Intismeran autogene plus pembrolizumab versus pembrolizumab alone in high-risk resected melanoma: 5-year update of the randomized phase IIb KEYNOTE-942 study. *Journal of Clinical Oncology*, 44 (26):2482–2489, 2026. doi: 10.1200/JCO-26-00835.

Zeynep Koşaloğlu-Yalçın, Nina Blazeska, Randi Vita, Hannah Carter, Morten Nielsen, Stephen Schoenberger, Alessandro Sette, and Bjoern Peters. The cancer epitope database and analysis resource (CEDAR). *Nucleic Acids Research*, 51(D1):D845–D852, 2023.

Marilyn Kozak. An analysis of 5'-noncoding sequences from 699 vertebrate messenger RNAs. *Nucleic Acids Research*, 15(20):8125–8148, 1987.

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, 2011.

Mihir Metkar, Christopher S Pepin, and Melissa J Moore. Tailor made: the art of therapeutic mRNA design. *Nature Reviews Drug Discovery*, 23(1):67–83, 2024.

Norbert Pardi and Florian Krammer. mRNA vaccines for infectious diseases—advances, challenges and opportunities. *Nature Reviews Drug Discovery*, 23(11):838–861, 2024.

Yuan Pu, Yazhe Niu, Zhenjie Yang, Jiyuan Ren, Hongsheng Li, and Yu Liu. UniZero: Generalized and efficient planning with scalable latent world models. *Transactions on Machine Learning Research*, 2025.

Julian Quevedo, Ansh Kumar Sharma, Yixiang Sun, Varad Suryavanshi, Percy Liang, and Sherry Yang. WorldGym: World model as an environment for policy evaluation. In *International Conference on Learning Representations*, 2026.

Binita Rajbanshi and Anuj Guruacharya. codonGPT: reinforcement learning on a generative language model enables scalable mRNA design. *Nucleic Acids Research*, 53(22):gkaf1345, 2025.

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.

Martin Steinegger and Johannes Söding. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature Biotechnology*, 35(11):1026–1028, 2017.

UniProt Consortium. UniProt: the universal protein knowledgebase in 2025. *Nucleic Acids Research*, 53(D1):D609–D617, 2025. doi: 10.1093/nar/gkae1010.

U.S. Food and Drug Administration. MRESVIA. U.S. Food and Drug Administration, 2024. Respiratory Syncytial Virus Vaccine, original approval May 31, 2024.

He Zhang, Liang Zhang, Ang Lin, Congcong Xu, Ziyu Li, Kaibo Liu, Boxiang Liu, Xiaopin Ma, Fanfan Zhao, Huiling Jiang, et al. Algorithm for optimized mRNA design improves stability and immunogenicity. *Nature*, 621(7978):396–403, 2023.

He Zhang, Hailong Liu, Yushan Xu, Haoran Huang, Yiming Liu, Jia Wang, Yan Qin, Haiyan Wang, Lili Ma, Zhiyuan Xun, et al. Deep generative models design mRNA sequences with enhanced translational capacity and stability. *Science*, 390(6773):eadr8470, 2025.

Xiangcheng Zhang and Yilun Du. World action planner: Generalizable decision-making with action-conditioned world models. *arXiv preprint arXiv:2607.27599*, 2026.

Fangqi Zhu, Hongtao Wu, Song Guo, Yuxiao Liu, Chilam Cheang, and Tao Kong. IRASim: A fine-grained world model for robot manipulation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9834–9844. IEEE, 2025.

APPENDIX CONTENTS

A Additional Method Details 14

A.1 Property Definitions, Normalization, and Preferences 14

A.2 Architecture Details 16

A.3 Offline Trajectory Construction and Training 18

A.4 Inference and Planner Details 21

A.5 Complete mRNAWM workflow 23

B Additional Experimental Details 25

B.1 Dataset Construction and Protein-Cluster Splits 25

B.2 Cross-Method Evaluation Protocol 25

B.3 Baseline Methods and Configurations 26

B.4 Primary Evaluation Metrics 27

B.5 Complete Cross-Species Comparison 27

B.6 Additional Analysis of Planning Horizon 28

B.7 Effect of the Edit Budget 30

B.8 Additional Analysis of UTR Initialization 31

B.9 Preference-Vector Calibration 32

B.10 Zero-Shot Evaluation on Tumor-Associated Antigens 34

B.11 Additional Qualitative Analysis of World-Model Planning 37

C Limitations and Future Work 40

D Ethical Considerations 42

E Broader Impact 43

Table 4: The 18 properties used by mRNAWM. The first 13 are deterministic sequence statistics and are eligible as state inputs. The five RNA-structural properties are prediction targets only and are never supplied to the state encoder.

Property	Definition	Objective	State input
CAI	Codon adaptation index computed from human relative synonymous codon usage weights, excluding the terminal stop codon.	max	yes
GC	Fraction of G/C nucleotides over the complete 5'UTR–CDS–3'UTR sequence.	target 0.50	yes
GC3	GC fraction at the third position of coding codons, excluding the stop codon.	target 0.50	yes
Rare-codon fraction	Fraction of coding codons whose human relative synonymous codon-usage weight is below 0.2.	min	yes
CpG density	Number of CG dinucleotides divided by the number of dinucleotide positions in the full transcript.	min	yes
UpA density	Number of TA dinucleotides in the internal DNA representation, corresponding to UpA after transcription.	min	yes
UG-rich density	Fraction of length-8 windows containing at least five U/T bases and at least one G.	min	yes
Homopolymer burden	Maximum homopolymer-run length divided by 10 and clipped at 1.	min	yes
Repeat-6 burden	Excess occurrences of repeated 6-mers divided by the number of 6-mer positions.	min	yes
Window-GC deviation	Maximum absolute deviation from 0.50 GC over length-60 sliding windows.	min	yes
Inverted-repeat burden	Fraction of length-8 windows whose reverse complement occurs elsewhere in the sequence.	min	yes
uAUG density	Number of upstream ATG triplets in the 5'UTR divided by the number of possible triplet positions.	min	yes
Kozak score	0.5 if the nucleotide at position -3 relative to the start codon is A/G, plus 0.5 if the $+4$ nucleotide is G.	max	yes
MFE per nt	ViennaRNA minimum free energy of the complete transcript divided by transcript length.	min	no
Start MFE per nt	Minimum free energy per nucleotide around the translation-initiation region.	max	no
Ensemble free energy per nt	ViennaRNA partition-function ensemble free energy divided by transcript length.	min	no
Ensemble diversity per nt	Mean base-pair distance from the ViennaRNA ensemble, normalized by transcript length.	target 0.10	no
Start unpaired fraction	Fraction of unpaired nucleotides in the predicted minimum-free-energy structure around the translation-initiation region.	max	no

A Additional Method Details

This appendix provides implementation details omitted from the main text, including the property definitions and normalization, model architecture, offline trajectory construction and losses, and the exact inference procedure. The notation follows Section 3.

A.1 Property Definitions, Normalization, and Preferences

A.1.1 Property set

mRNAWM models $D = 18$ sequence-derived properties. Internally, sequences are represented in the DNA alphabet $\{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$; RNA folding replaces T with U before calling ViennaRNA (Lorenz et al., 2011). Table 4 lists the properties, their optimization direction, and whether they are provided to the state encoder.

Table 5: Property centers and scales used for standardization.

Property	c_i	s_i	Property	c_i	s_i
CAI	0.700	0.200	GC	0.500	0.200
GC3	0.500	0.250	Rare codon	0.050	0.100
CpG	0.015	0.020	UpA	0.015	0.020
UG-rich	0.050	0.080	Homopolymer	0.500	0.300
Repeat-6	0.050	0.080	Window-GC deviation	0.200	0.200
Inverted repeat	0.050	0.080	uAUG	0.010	0.020
Kozak	0.500	0.500	MFE / nt	-0.250	0.200
Start MFE / nt	-0.200	0.200	Ensemble FE / nt	-0.250	0.200
Ensemble diversity	0.100	0.100	Start unpaired	0.500	0.250

A.1.2 RNA STRUCTURAL PROPERTIES

Let $x = U^{5'}CU^{3'}$ denote the complete nucleotide sequence. For the full-transcript structural quantities, ViennaRNA is applied to x. For initiation-region quantities, we use a window of 120 nucleotides (nt) whenever available, consisting of the last 60 nt of the 5'UTR and the first 60 nt of the CDS.

The minimum free energy is normalized by sequence length. When partition-function evaluation is enabled, the ensemble free energy and mean base-pair distance are also normalized by transcript length. The initiation-region unpaired fraction is the fraction of "." characters in the ViennaRNA minimum-free-energy dot-bracket structure.

Full-transcript folding is capped at 5,000 nt. When a structural target is unavailable, it is marked invalid rather than replaced by zero.

A.1.3 PROPERTY STANDARDIZATION

Prediction heads operate on standardized properties. For property i with raw value p_i, center c_i, and scale s_i, we use

$$\tilde{p}_i = \mathrm{clip}\left(\frac{p_i - c_i}{s_i}, -8, 8\right).$$

The constants are given in Table 5.

A.1.4 DIRECTION-ALIGNED QUALITY FEATURES

Preferences operate on direction-aligned quality features rather than directly on standardized prediction targets. Let $d_i \in \{-1, 0, +1\}$ denote whether property i is minimized, targeted, or maximized, respectively, and let t_i denote the target value when $d_i = 0$. We define

$$q_i(p_i) = \mathrm{clip}\left(\begin{cases} d_i p_i / s_i, & d_i \in \{-1, +1\}, \\ -|p_i - t_i|/s_i, & d_i = 0, \end{cases} - 20, 20\right).$$

Thus, larger q_i always denotes a more desirable value. The scalar utility used to construct behavior and value targets is $U_{\mathbf{w}}(x) = \sum_i w_i q_i(p_i(x))$.

If a property is unavailable for an entire training trajectory, its preference weight is masked and the remaining weights are renormalized:

$$\bar{w}_i = \frac{w_i m_i}{\sum_j w_j m_j},$$

where $m_i = 1$ only when the property is available. This prevents policy and value targets from implicitly optimizing an unobserved structural endpoint.

A.1.5 STATE-CONDITIONING PROPERTIES

All 18 property slots and an 18-dimensional validity mask are represented in the state-fusion interface, but structural information is explicitly removed before encoding. For the first 13 properties,

Table 6: Final GC3/CpG-targeted preference vector used for the benchmark configuration. The CAI weight is fixed at 0.80, while the residual preference mass is allocated to preserve composition, manufacturability, and structural objectives, with additional emphasis on GC3 and CpG density.

Property	Weight	Property	Weight
CAI	0.8000	GC	0.0150
GC3	0.0350	Rare codon fraction	0.0000
CpG density	0.0350	UpA density	0.0150
UG-rich density	0.0115	Homopolymer burden	0.0115
Repeat-6 burden	0.0104	Window-GC deviation	0.0058
Inverted-repeat burden	0.0104	uAUG density	0.0104
Kozak score	0.0000	MFE per nt	0.0178
Start MFE per nt	0.0059	Ensemble FE per nt	0.0000
Ensemble diversity	0.0104	Start unpaired fraction	0.0059

available standardized values are retained. For the final five structural properties, both the conditioning mask and the property value supplied to the representation network are forced to zero. Consequently, structural labels can supervise the prediction heads without leaking directly into z_t.

A.1.6 Final Preference Vector

Unless otherwise stated, the final benchmark configuration uses the **GC3/CpG-targeted** preference in Table 6. This preference was obtained by first selecting a CAI allocation of 0.80 and then redistributing the remaining preference mass to reduce the composition degradation observed during CAI optimization, with particular emphasis on GC3 and CpG density. The weights sum to one exactly; the implementation additionally normalizes the preference vector before planning.

A.2 Architecture Details

A.2.1 Cached CodonTransformer features

CodonTransformer is frozen and is not part of the trainable mRNAWM backbone. For each unique target protein, it is run once and its outputs are cached. The cache stores (i) a 128-dimensional contextual feature at each amino-acid position and (ii) a distribution over the 64 possible codons at each position.

The contextual representation is consumed by the state encoder. The codon distribution is not included in z_t and is used only for initial CDS construction and as a weak proposal prior during inference. Consequently, mRNAWM training does not repeatedly execute the CodonTransformer backbone.

A.2.2 Protein-conditioned CDS encoder

At each codon position, the CDS encoder combines:

- the cached 128-dimensional protein-context feature;
- a 64-dimensional embedding of the current codon; and
- a two-dimensional positional feature $[r, 1 - r]$.

The protein context is projected to 192 dimensions and summed with a projected positional representation. This is concatenated with the current codon embedding and passed through a 192-dimensional fusion block. Masked mean and max pooling across codon positions are concatenated and projected to a 192-dimensional CDS representation.

The target protein context therefore remains fixed across a trajectory, whereas the codon embeddings change after synonymous edits.

A.2.3 JUNCTION UTR ENCODER

The same junction encoder is shared by the 5' and 3' UTRs. An orientation embedding distinguishes the start and stop boundaries.

The 5' branch receives the visible 5'UTR together with the first 48 nt of the CDS, while the 3' branch receives the last 48 nucleotides (nt) of the CDS followed by the visible 3'UTR. The maximum retained UTR lengths are 256 nt for the 5'UTR and 512 nt for the 3'UTR.

For UTRs exceeding these limits, the encoder retains both ends rather than a single contiguous crop. For the 5'UTR, 64 nt are retained from the distal end and 192 nt from the CDS-proximal end. For the 3'UTR, 384 nt are retained from the CDS-proximal end and 128 nt from the distal end. Eight explicit gap tokens are inserted between retained fragments. Local convolutions are applied independently on the two sides of a crop seam, preventing artificial motifs from being created by concatenating originally non-adjacent nucleotides.

Each junction token receives the sum of:

- nucleotide or gap-token embedding;
- segment embedding distinguishing 5'UTR, CDS, and 3'UTR;
- start/stop orientation embedding;
- sinusoidal absolute-coordinate encoding; and
- four learned positional features derived from normalized absolute position, signed distance to the junction, direction, and junction proximity.

A kernel-7 depthwise convolution followed by a pointwise convolution provides local motif processing. The representation is then downsampled with stride 4 before a two-layer, four-head Transformer. Mean pooling and learned attention pooling are concatenated and projected to 192 dimensions.

A.2.4 GLOBAL UTR ENCODER

The global branch retains coarse information from the entire UTR at fixed computational cost. Each complete UTR is divided into up to 128 ordered bins. Every bin contains the empirical frequencies of A, C, G, T, and N in that interval.

The five input channels are projected to 48 channels and processed by a kernel-5 depthwise convolution followed by a pointwise convolution. Mean pooling, max pooling, and a normalized log-length feature $\log(1 + L)/10$ are concatenated and projected to 192 dimensions. The junction and global representations are concatenated and fused through a linear–LayerNorm–SiLU block to produce one 192-dimensional representation for each UTR.

A.2.5 STATE FUSION

The state-fusion network receives

$$192_{\mathrm{CDS}} + 192_{5'\mathrm{UTR}} + 192_{3'\mathrm{UTR}} + 18_{\mathrm{property}} + 18_{\mathrm{mask}} = 612$$

input dimensions. It applies $612 \to 512 \to 384$ with LayerNorm, SiLU activation, and dropout, followed by a final LayerNorm. The resulting 384-dimensional vector is the latent state z_t.

A.2.6 ACTION ENCODING AND LATENT TRANSITION

Actions contain an edit type, normalized position, normalized sequence length, old symbol, and replacement symbol. There are three edit types and 68 possible symbols: 64 codons and four nucleotide symbols.

The normalized position is $(i + 0.5)/L$, and the length feature is $\min(\log(1 + L)/10, 1)$. Edit type, old symbol, and new symbol are each embedded into 64 dimensions. The concatenated representation is projected to a 384-dimensional action embedding.

Table 7: Prediction-head dimensions.

Module	Input dim.	Output dim.	Heads
Observation O	384	18	5
Property change R	384+384	18	5
Future value V	384+18	1	5
Policy trunk	384+18	512	1
Policy region	512	3	1
Policy position	512	128	1
Policy symbol	512	68	1
Inverse-action head	384+384	3	1

The residual dynamics concatenate the 384-dimensional latent and 384-dimensional action representation, pass them through a $768 \to 512 \to 384$ MLP, add the predicted residual to the previous latent, and apply LayerNorm.

A.2.7 Prediction heads

The observation, property-change, and future-value predictors each use $M = 5$ independently parameterized ensemble heads. Each head is a two-layer MLP with hidden width 512 and dropout 0.05.

For an ensemble prediction $\{y^{(m)}\}_{m=1}^{M}$, the predictive mean and disagreement are

$$\bar{y} = \frac{1}{M} \sum_{m=1}^{M} y^{(m)}, \qquad \sigma_y = \mathrm{std}_{m=1}^{M} \left[y^{(m)} \right],$$

using the population standard deviation.

The policy factorizes an action into region, position bin, and replacement symbol. Its log-probability for a legal action is the sum of the three corresponding log-softmax terms.

A.3 Offline Trajectory Construction and Training

A.3.1 Preference sampling

For each offline trajectory, an 18-dimensional non-negative preference vector is sampled independently as $g_i \sim \mathrm{Gamma}(\alpha, 1)$ with $\alpha = 0.35$, followed by normalization,

$$w_i = \frac{g_i}{\sum_j g_j}.$$

The small concentration parameter produces diverse and often sparse objective combinations.

When a trajectory does not contain structural evaluator supervision, structural preference entries are set to zero and the remaining weights are renormalized. For MFE-only trajectories, objectives requiring the partition function are similarly disabled.

A.3.2 Behavior-trajectory generation

Offline trajectories contain a mixture of goal-directed and exploratory legal edits. At each state, edit regions are sampled with probabilities 0.50, 0.25, and 0.25 for CDS, 5'UTR, and 3'UTR, respectively.

For the standard training trajectory, up to eight unique candidate edits are sampled. With probability 0.5, the candidate having the highest utility under the sampled preference is selected; otherwise a candidate is selected randomly. For trajectories requiring ViennaRNA supervision, the candidate set is capped at two to limit folding cost.

The dynamics-supervision horizon is five edits. An additional five behavior steps are generated only for future-value supervision. These tail states are not passed through the sequence encoder as part of the latent rollout window.

Table 8: Sparse structural ViennaRNA sampling. The partition-function fraction is a subset of the total ViennaRNA fraction; remaining structurally supervised trajectories receive MFE-based targets without ensemble free-energy/diversity targets.

	Training	**Validation**
Any ViennaRNA supervision	0.002	0.020
Partition-function supervision	0.0005	0.005
Structural-loss weight	16	1

A.3.3 Sparse multi-fidelity structural supervision

Structural evaluator assignment is deterministic per transcript identifier and random seed. The final training configuration uses the sampling fractions in Table 8.

For unsupervised structural entries, the target mask is false. For structurally supervised training trajectories, valid structural loss terms are upweighted by a factor of 16. This reweighting increases the influence of scarce structural labels without duplicating complete encoder forward passes.

A.3.4 Latent consistency loss

Let z_{t+k} be the re-encoded real latent and $\hat{z}_{t+k}$ the recursively imagined latent. The exact dynamics loss is

$$\mathcal{L}_{\text{latent}} = \frac{1}{H}\sum_{k=1}^{H} \frac{\text{SmoothL1}\left(\hat{z}_{t+k}, \text{sg}(z_{t+k})\right) + 0.25\left[1 - \cos\left(\hat{z}_{t+k}, \text{sg}(z_{t+k})\right)\right]}{k}.$$

The $1/k$ weighting emphasizes short-horizon transition accuracy while still supervising recursive rollouts.

A.3.5 Observation and rollout losses

Each observation head predicts standardized properties. The observation ensemble is supervised on both re-encoded real latents and recursively imagined latents:

$$\mathcal{L}_{\text{obs}} = \mathcal{H}_{\text{ens}}\left(O(z), \tilde{\mathbf{p}}\right), \qquad \mathcal{L}_{\text{rollout}} = \mathcal{H}_{\text{ens}}\left(O(\hat{z}), \tilde{\mathbf{p}}\right),$$

where $\mathcal{H}_{\text{ens}}$ denotes masked Smooth-L1 loss across ensemble members.

A deterministic record-dependent bootstrap mask assigns approximately 80% of examples to each ensemble member. This gives the heads different effective training subsets while preserving fully reproducible data assignment.

A.3.6 Property-change loss

The property-change target is defined in standardized property space. For consecutive states,

$$\mathbf{y}_t^{\Delta} = \kappa_{\Delta}\left(\tilde{\mathbf{p}}_{t+1} - \tilde{\mathbf{p}}_t\right), \qquad \kappa_{\Delta} = 50.$$

The property-change ensemble is trained with masked ensemble Smooth-L1 loss:

$$\mathcal{L}_{\Delta p} = \mathcal{H}_{\text{ens}}\left(R(z_t, a_t), \mathbf{y}_t^{\Delta}\right).$$

A property is valid for this loss only when it is available at both ends of the transition.

A.3.7 Future-value targets

Let $r_t = U_{\mathbf{w}}(x_{t+1}) - U_{\mathbf{w}}(x_t)$. The value target for state t is

$$G_t = \sum_{j=t}^{T-1} \gamma^{j-t} r_j, \qquad \gamma = 0.97,$$

where T extends beyond the five-step latent-training window using the additional behavior tail described in Appendix A.3.2. The value ensemble is trained with bootstrap-masked Smooth-L1 loss.

The additional tail avoids treating the end of the dynamics-supervision window as an environmental terminal state.

Table 9: Training hyperparameters for the final mRNAWM model.

Training parameter	**Value**
Optimizer	AdamW
Learning rate	2×10^{-4}
Weight decay	0.01
Warmup	1,000 optimizer steps
Post-warmup schedule	cosine decay
Training batch size	12
Validation batch size	16
Gradient accumulation	2
Gradient clipping	1.0
Precision	bfloat16
Dropout	0.05
Training latent horizon	5
Value-only tail horizon	5
Behavior candidates	8
Greedy behavior probability	0.5
Preference concentration α	0.35
Random seed	3407

A.3.8 Policy and auxiliary supervision

The structured policy predicts three categorical targets: edit region, normalized position bin, and replacement symbol. With 128 position bins, its loss is

$$\mathcal{L}_\pi = \mathrm{CE}(\pi_{\mathrm{region}}, y_{\mathrm{region}}) + \mathrm{CE}(\pi_{\mathrm{position}}, y_{\mathrm{position}}) + \mathrm{CE}(\pi_{\mathrm{symbol}}, y_{\mathrm{symbol}}).$$

An auxiliary inverse-action classifier receives $[z_t, \mathrm{sg}(z_{t+1})]$ and predicts which of the three edit regions produced the transition. Its loss is denoted $\mathcal{L}_{\mathrm{inv}}$. A continuation head is implemented but receives zero weight in the final training configuration.

A.3.9 Full training objective

The final model is trained with

$$\begin{aligned}\mathcal{L} = {} & 2.0\,\mathcal{L}_{\mathrm{latent}} + 1.0\,\mathcal{L}_{\mathrm{obs}} + 1.0\,\mathcal{L}_{\mathrm{rollout}} + 2.0\,\mathcal{L}_{\Delta p} \\ & + 1.0\,\mathcal{L}_V + 0.5\,\mathcal{L}_\pi + 0.1\,\mathcal{L}_{\mathrm{inv}}.\end{aligned}$$

The continuation-loss coefficient is zero in the final model.

A.3.10 Optimization hyperparameters

Table 9 summarizes the optimization and sampling hyperparameters used to train the final mRNAWM model. Unless otherwise stated, these settings are fixed across the reported training runs.

A.3.11 Coverage-aware streaming

The final training pipeline uses epoch-aware coverage sampling over the frozen training split. Each data-loader worker retains its original byte-range partition, but the starting location within that partition is cyclically shifted between epochs. This changes only which records are encountered first; trajectory generation, feature lookup, preference sampling, structural-oracle sampling, and model supervision are otherwise unchanged.

The training launcher consumes at most 5,000 batches of size 12 per epoch, corresponding to 60,000 trajectory examples per epoch. Coverage offsets are deterministic under the training seed, enabling exact resume behavior and auditing of which base records were visited.

A.3.12 Checkpoint selection

Validation tracks both the complete multi-task loss and structural prediction accuracy. In addition to the overall best checkpoint, training stores a `best_structure` checkpoint according to the mean of the real-state and rollout structural-property MAE. The final mRNAWM experiments use the selected structural checkpoint.

A.4 Inference and Planner Details

A.4.1 CDS and UTR initialization

For the final configuration, CDS initialization uses the cached CodonTransformer prior. At each protein position, the highest-probability synonymous codon under the cached prior is selected.

The planner supports either a supplied UTR pair or a library of sequence-verified UTR scaffolds. In scaffold-library mode, each scaffold is combined with the initialized CDS and evaluated before edit planning. The figure in the main text depicts this scaffold-initialization mode.

A.4.2 Scaffold ranking

For a real scaffold state, available properties are computed directly and unavailable structural values are completed using the observation-ensemble mean. Let

$$A(x) = \mathbf{w}^\top q(\bar{\mathbf{p}}(x))$$

denote the resulting analytic utility. Uncertainty associated with unavailable properties is

$$\mathcal{U}_O(x) = \sum_i |w_i|\, \sigma_{O,i}(z)\mathbf{1}[m_i = 0].$$

Initial scaffolds are ranked using

$$S_{\text{scaffold}}(x) = A(x) + \beta_{\text{scaffold}}\bar{V}(z, \mathbf{w}) - \beta_U \left[\mathcal{U}_O(x) + \beta_{\text{scaffold}}\sigma_V(z, \mathbf{w})\right].$$

Feasible scaffolds are preferred to infeasible ones; infeasible scaffolds are ranked by their distance to the feasible constraint set. The final configuration uses $\beta_{\text{scaffold}} = 0.25$.

A.4.3 Legal action space

CDS actions enumerate synonymous codon replacements at mutable protein positions. UTR actions enumerate single-nucleotide substitutions. Insertions and deletions are not permitted.

For an initialized UTR scaffold, the final setting permits at most four UTR substitutions in total, with at most two in the 5'UTR and two in the 3'UTR. The Hamming distance to the original scaffold is recomputed after every candidate edit, so an edit that reverts a previous substitution can decrease the distance.

For UTRs longer than the visible junction-encoder crop, editable positions are restricted to positions retained by the encoder. This avoids proposing an action whose edited nucleotide is completely absent from the local sequence representation.

A.4.4 Structured proposal score

All legal actions are first scored by the structured policy. For action a with region $r(a)$, position bin $b(a)$, and replacement symbol $s(a)$,

$$\ell_{\text{policy}}(a) = \log \pi_r(r(a) \mid z, \mathbf{w}) + \log \pi_b(b(a) \mid z, \mathbf{w}) + \log \pi_s(s(a) \mid z, \mathbf{w}).$$

For synonymous CDS edits, the CodonTransformer prior contributes an additional weak term:

$$\ell_{\text{prop}}(a) = \ell_{\text{policy}}(a) + \mathbf{1}[a \in \text{CDS}]\beta_{\text{CT}} \log p_{\text{CT}}(c_{\text{new}} \mid P, i).$$

We use $\beta_{\text{CT}} = 0.02$.

A small region-balanced quota is reserved before filling the remaining proposal budget globally. This prevents the much larger synonymous-CDS action space from eliminating all 5'UTR or 3'UTR proposals before MPC evaluation.

A.4.5 MULTI-FIDELITY IMAGINED TRANSITION

For an imagined transition, the property-change ensemble first predicts the standardized change. With ensemble mean $\bar{R}$,

$$\tilde{\mathbf{p}}' = \tilde{\mathbf{p}} + \frac{\bar{R}(z, a)}{\kappa_\Delta}, \qquad \kappa_\Delta = 50.$$

The candidate action also defines an explicit edited nucleotide sequence. The first 13 deterministic properties are therefore recomputed exactly and overwrite the corresponding entries of $\tilde{\mathbf{p}}'$. Only the five structural properties remain model-predicted during latent search.

This exact overwrite is also used for hard constraints, so constraint feasibility never depends on a learned approximation of these deterministic properties.

A.4.6 EXACT RISK-ADJUSTED BRANCH SCORE

Suppose a partial imagined trajectory contains d edits. Its discounted cumulative utility gain is

$$G_d = \sum_{k=0}^{d-1} \gamma^k \Delta U_k, \qquad \Delta U_k = \mathbf{w}^\top \left[q(\mathbf{p}_{k+1}) - q(\mathbf{p}_k) \right].$$

For property-change uncertainty we accumulate

$$\Sigma_d^\Delta = \sum_{k=0}^{d-1} \sum_i |w_i| \frac{\sigma_{R,k,i}}{\kappa_\Delta}.$$

The terminal value uncertainty is

$$\Sigma_d^V = \gamma^d \beta_V \sigma_V(\hat{z}_d, \mathbf{w}),$$

and the total uncertainty penalty is $\Sigma_d = \Sigma_d^\Delta + \Sigma_d^V$.

The exact branch-ranking score is

$$S_d = G_d + \gamma^d \beta_V \bar{V}(\hat{z}_d, \mathbf{w}) + \beta_\pi \ell_{\text{prop}}(a_{d-1}) - \beta_U \Sigma_d - \beta_C C_d,$$

where C_d is the normalized hard-constraint violation distance at the candidate state.

The final configuration uses $\gamma = 0.97$, $\beta_V = 0.5$, $\beta_\pi = 0.01$, $\beta_U = 0.5$, and $\beta_C = 10$.

Notice that the policy term is the proposal score for the action that extends the current branch; it is not an accumulated log-probability over the entire trajectory.

A.4.7 CONSTRAINT HANDLING

Hard constraints are applied only to exactly computable properties. For property i with scale s_i and bounds (l_i, u_i), the normalized violation is

$$c_i(p_i) = \frac{[l_i - p_i]_+}{s_i} + \frac{[p_i - u_i]_+}{s_i}, \qquad C = \sum_i c_i(p_i).$$

The default benchmark constraints are

$$0.30 \le \text{GC} \le 0.70, \qquad \text{homopolymer burden} \le 0.8, \qquad \text{uAUG density} \le 0.05.$$

If the current branch is feasible, an imagined successor must remain feasible. If the current branch is infeasible, only successors that strictly reduce the constraint-violation distance are retained.

A.4.8 BEAM SEARCH AND VARIABLE IMAGINED DEPTH

For each retained branch, up to K proposals are expanded and scored. The top B successors are retained after each depth. A branch is prevented from revisiting the same (CDS, 5'UTR, 3'UTR) sequence state.

Importantly, planning is not forced to use exactly H imagined edits. The retained beams at every depth $d \in \{1, \ldots, H\}$ are eligible terminal candidates. The planner therefore selects the highest-scoring partial trajectory over all searched depths.

Table 10: Final inference configuration. The primary search parameters were selected through the ablation studies; fixed implementation parameters are reported for completeness.

Parameter	Final setting
Planning horizon H	5
Beam width B	5
Proposal count K	128
Maximum edit budget	60
Minimum risk-adjusted improvement ϵ	-0.003
CodonTransformer prior β_{CT}	0.02
Uncertainty penalty β_U	0.50
Future-value coefficient β_V	0.50
Policy coefficient β_π	0.01
Scaffold-value coefficient $\beta_{\mathrm{scaffold}}$	0.25
Discount γ	0.97
ORF initialization	CodonTransformer prior
Maximum total UTR edits	4
Maximum 5'UTR edits	2
Maximum 3'UTR edits	2
Constraint-violation coefficient β_C	10.0

A.4.9 Real-state acceptance and stopping

Only the first edit of the selected imagined branch is applied to the real sequence. The resulting state is re-encoded and receives a risk-adjusted real-state score

$$S_{\mathrm{real}}(x) = A(x) - \beta_U \mathcal{U}_O(x).$$

Let $\Delta S_{\mathrm{real}} = S_{\mathrm{real}}(x_{t+1}) - S_{\mathrm{real}}(x_t)$. A proposed edit is committed if either

1. the current state is infeasible and the edit strictly decreases its constraint-violation distance; or
2. the new state satisfies all hard constraints and $\Delta S_{\mathrm{real}} > \epsilon$.

For the final configuration, $\epsilon = -0.003$. Thus, the controller may accept a small locally negative risk-adjusted change when the multi-step planner predicts that it leads to a better downstream trajectory. If neither condition is met, optimization terminates.

A.4.10 Final ViennaRNA validation

The state committed most recently is not automatically returned as the final design. All accepted states, including the unedited initialization, are retained. At the end of a scaffold run, these states are re-evaluated with ViennaRNA. The best feasible validated state is selected according to its terminal risk-adjusted utility.

When multiple UTR scaffolds are searched, final selection is performed across the best validated state from each scaffold run. This prevents the learned structural surrogate from determining the final reported sequence without oracle verification.

A.4.11 Final planner configuration

The final planner configuration was selected from the ablation studies reported in the main experiments and is summarized in Table 10.

A.5 Complete mRNAWM workflow

Algorithm 2 summarizes the complete mRNAWM workflow, including protein-conditioned feature caching, offline world-model learning, and closed-loop inference. The key separation is between preference-independent predictive learning (h_ϕ, f_θ, O, R) and preference-conditioned control (V, π): the former learns how protein-preserving edits change the mRNA design state, whereas the latter determines how the learned dynamics are exploited for a particular objective trade-off.

Algorithm 2 Complete mRNAWM Training and Inference Workflow

Require: Training transcripts with target proteins; property evaluator $\mathbf{p}(\cdot)$; planning preference $\mathbf{w}$; rollout horizon H; proposal count K; beam width B

Phase I: Cache protein-conditioned features

for each unique target protein P **do**
  Run frozen CodonTransformer once
  Cache contextual protein features $c(P)$
  Cache synonymous-codon prior $\rho(\cdot \mid P)$
end for

Phase II: Construct offline edit trajectories

for each training transcript $x_0 = (U_0^5, C_0, U_0^3)$ **do**
  Compute sequence properties $\mathbf{p}_0 = \mathbf{p}(x_0)$
  Construct a trajectory of legal protein-preserving edits $\{a_t\}_{t=0}^{T-1}$
  for $t = 0, \ldots, T-1$ **do**
    Apply a_t to obtain the real edited sequence x_{t+1}
    Recompute available properties $\mathbf{p}_{t+1} = \mathbf{p}(x_{t+1})$
    Store $(x_t, a_t, x_{t+1}, \mathbf{p}_t, \mathbf{p}_{t+1})$
  end for
end for

Phase III: Learn the predictive world model

for each training minibatch **do**
  Sample trajectory windows $(x_t, a_t, \ldots, a_{t+H-1}, x_{t+H})$
  Encode real states $z_{t+k} = h_\phi(s_{t+k})$, $k = 0, \ldots, H$
  Initialize imagined state $\hat{z}_t \leftarrow z_t$
  for $k = 0, \ldots, H-1$ **do**
    Encode action $e_{t+k} \leftarrow e(a_{t+k})$
    Predict residual transition $\Delta\hat{z}_{t+k} \leftarrow f_\theta(\hat{z}_{t+k}, e_{t+k})$
    Roll forward $\hat{z}_{t+k+1} \leftarrow \mathrm{LN}(\hat{z}_{t+k} + \Delta\hat{z}_{t+k})$
  end for
  Supervise latent dynamics by matching $\hat{z}_{t+1:t+H}$ to $\mathrm{sg}(z_{t+1:t+H})$
  Train O to predict properties from real and imagined latents
  Train R to predict vector-valued property changes of recorded edits
  Sample a preference vector $\mathbf{w}$
  Compute realized utility improvements $\delta_j^{\mathbf{w}} \leftarrow U_{\mathbf{w}}(x_{j+1}) - U_{\mathbf{w}}(x_j)$
  Construct discounted continuation targets for V
  Train $V(z_j, \mathbf{w})$ on continuation improvement
  Train $\pi(a_j \mid z_j, \mathbf{w})$ on the selected legal edits
  Update trainable mRNAWM parameters with the joint objective
end for

Phase IV: Closed-loop mRNA optimization

Require: Target protein P and optional initial transcript x_0

if an initial CDS is not provided **then**
  Initialize CDS using cached codon prior $\rho(\cdot \mid P)$
end if
Initialize 5' and 3' UTRs from valid scaffolds if needed
Form the initial full-length transcript x_0
Evaluate $\mathbf{p}(x_0)$ and encode $z_0 = h_\phi(s_0)$
for $t = 0, 1, \ldots$ until termination **do**
  Initialize beam $\mathcal{B} \leftarrow \{(x_t, z_t)\}$
  for $h = 1, \ldots, H$ **do**
    for each branch $(x, z) \in \mathcal{B}$ **do**
      Propose legal edits using $\pi(a \mid z, \mathbf{w})$ and cached codon prior
      Retain up to K candidate actions
      for each candidate edit a **do**
        Apply a to obtain explicit candidate sequence x'
        Imagine $\hat{z}' \leftarrow \mathrm{LN}(z + f_\theta(z, e(a)))$

Evaluate candidate sequence properties and world-model predictions
Obtain $R(z, a)$, $V(\hat{z}', \mathbf{w})$, policy score, and ensemble disagreement
Compute risk-adjusted trajectory score
end for
end for
Retain the top B partial trajectories
end for
Select the highest-scoring imagined trajectory
Let a_t^* be its first edit and apply it to the real sequence
Obtain x_{t+1} and recompute its properties
if the real-state acceptance criterion fails **then**
break
end if
Re-encode the real state $z_{t+1} \leftarrow h_\phi(s_{t+1})$
end for
return optimized full-length mRNA x_T

B Additional Experimental Details

B.1 Dataset Construction and Protein-Cluster Splits

The general mRNA corpus is constructed from human and broad vertebrate-mammalian RefSeq (Goldfarb et al., 2025) transcript records. Each normalized record contains a protein sequence, its corresponding coding sequence (CDS), a 5'UTR, a 3'UTR, and source metadata. Records are retained only when the CDS can be parsed consistently with the reported protein sequence, and exact duplicate protein–CDS records are removed during corpus construction.

To prevent leakage of identical design targets across data partitions, we split the corpus by *exact protein sequence* rather than by transcript record. Specifically, all transcripts encoding the same amino-acid sequence are assigned to a single protein cluster, and complete clusters are assigned to the train, validation, or test partition. We use an 80/10/10 split with random seed 3407. Thus, no exact target-protein sequence occurs in more than one partition.

The same frozen split is used throughout model training, hyperparameter experiments, baseline generation, and final evaluation. The validation split is used for checkpoint selection, while the held-out test split is reserved for the reported cross-method comparisons.

B.2 Cross-Method Evaluation Protocol

All methods are evaluated using a common post-generation evaluator. We do not reuse method-specific scores or objective values when reporting cross-method results. Instead, the generated nucleotide sequences are independently rescored using the same property definitions, codon-usage table, and RNA-structure pipeline.

Construction of full transcripts. Most comparison methods optimize only the coding sequence. For a CDS-only method producing $\hat{C}$, we construct the evaluated transcript as

$$\hat{x} = \left(U^{5'}_{\text{native}}, \hat{C}, U^{3'}_{\text{native}}\right),$$

Table 11: Summary of the frozen RefSeq mRNA dataset splits used in the final experiments. The dataset contains 1,923,364 records in total and is partitioned approximately 80/10/10 into training, validation, and test sets. Protein clusters denote the grouping used for the exact-protein split.

Split	Records	Protein clusters	Human records	Fraction
Train	1,538,006	1,428,914	85,463	0.7996
Validation	192,387	178,614	10,804	0.1000
Test	192,971	178,614	10,868	0.1003

Table 12: Methods and inference configurations used in the cross-method comparison. "Designed region" refers specifically to the component optimized in our experiments.

Method	Designed region	Optimization	Configuration
mRNAWM	CDS / CDS + UTR	Latent MPC	final preference, $H = 5$, $B = 5$, $K = 128$
CodonTransformer	CDS	Conditional generation	*H. sapiens*
CodonRL	CDS	Multi-objective RL	soup55 checkpoint
GEMORNA	CDS / CDS + UTR	Generative model	released CDS and full-length models
LinearDesign	CDS	Dynamic programming	released implementation
codonGPT	CDS	Autoregressive + RL	$T = 1.0$, top-$p = 0.9$
Trias	CDS	Conditional generation	*H. sapiens*, greedy

where the UTRs are taken from the corresponding frozen RefSeq test record. For mRNAWM, UTR substitutions are part of the optimization problem, and the evaluated sequence is therefore

$$\hat{x}_{\mathrm{mRNAWM}} = \left(\hat{U}^{5'}, \hat{C}, \hat{U}^{3'}\right).$$

This convention allows CDS-only methods to be evaluated on full-transcript properties without attributing arbitrary UTR sequences to them. For the primary results in Table 1, mRNAWM (Full) is initialized with the same native 5' and 3' UTRs and then jointly optimizes synonymous CDS choices and constrained UTR substitutions.

Shared property computation. Codon-usage statistics are recomputed using a single human codon-usage table. Sequence-composition and motif-based quantities are recomputed directly from the generated transcript. RNA-structure properties are recomputed using the same ViennaRNA implementation and settings for every method.

Because partition-function evaluation is substantially more expensive than sequence-level statistics, ViennaRNA-dependent quantities are evaluated on a deterministic record-ID-hashed subset shared by all methods. The subset membership is independent of method output and therefore identical across compared methods.

Aggregation. For every reported property, we first compute the metric independently for each test transcript and then report the arithmetic mean over the corresponding evaluation cohort. The same metric direction is fixed for every method before comparison. No post hoc reweighting or composite cross-method score is used in the primary results.

B.3 Baseline Methods and Configurations

We use the official implementations and released pretrained models whenever available. Table 12 summarizes the sequence-design space and inference configuration used in our experiments. The table describes how each method is used in *our evaluation*; it is not intended as an exhaustive summary of every capability provided by the original software.

CodonTransformer. We use the released pretrained CodonTransformer model and condition generation on *Homo sapiens*. Its generated synonymous CDS is evaluated using the common evaluation protocol in Appendix B.2.

CodonRL. We use the released `soup55` checkpoint through the official CodonRL inference pipeline. No mRNAWM-specific objective or reranking is applied to its generated sequence.

GEMORNA. GEMORNA provides models for both CDS and full-length mRNA generation. We evaluate both released configurations in our comparison. For GEMORNA (CDS), we use its protein-conditioned CDS-generation component and pair the generated CDS with the native UTRs when computing full-mRNA metrics, consistent with the evaluation protocol for CDS-only methods. For GEMORNA (Full), we directly evaluate the full-length sequence generated by its released full-mRNA model.

Table 13: Eight primary properties used in the cross-method comparison. Detailed definitions are given in Appendix A.1.1.

Category	Metric	Region	Direction
Translation	CAI	CDS	maximize
Translation	Kozak score	start junction	maximize
RNA structure	Full-mRNA MFE / nt	full transcript	minimize
RNA structure	Ensemble diversity / nt	full transcript	target 0.10
Manufacturability	Homopolymer burden	full transcript	minimize
Manufacturability	Window-GC deviation	full transcript	minimize
Manufacturability	Upstream AUG density	5'UTR	minimize
Sequence composition	UG-rich density	full transcript	minimize

LinearDesign. We use the released LinearDesign implementation to optimize synonymous CDS choices under its native codon-usage and RNA-structure objective. We evaluate the resulting sequence independently using our common evaluator.

codonGPT. We use the released codonGPT checkpoint with synonymous-codon constraints. Following the released inference configuration, generation uses temperature 1.0 and top-p 0.9 sampling. We use random seed 3407 for reproducibility.

Trias. We use the released Trias checkpoint conditioned on *Homo sapiens*. Decoding is deterministic and greedy. During evaluation, each predicted codon is restricted to synonymous codons compatible with the target amino acid sequence.

B.4 Primary Evaluation Metrics

The complete set of sequence properties modeled by mRNAWM is defined in Appendix A.1.1. For the primary cross-method comparison, we select eight metrics spanning four distinct aspects of mRNA sequence design, rather than reporting multiple highly correlated variants of the same property.

Ensemble-diversity target. Unlike the remaining structural metric, ensemble diversity is treated as a targeted rather than monotonic objective. The target value 0.10 is fixed before test evaluation and is used consistently in training, planning, and evaluation. Performance is therefore compared using $|p_{\mathrm{div}} - 0.10|$. We do not interpret 0.10 as a universal biological optimum; it is the prespecified operating point used by the model objective.

B.5 Complete Cross-Species Comparison

We provide the complete method comparison underlying the cross-species analysis in Sec. 4.3. The source-species label denotes the biological origin of the encoded protein; all methods are evaluated using the same human codon-usage reference and metric definitions. All entries are means within each source-species stratum. Unlike the main-text table, which focuses on the consistency of mRNAWM itself, Tables 14–16 report the corresponding baseline results for every source-species stratum. Bold cells with darker shading indicate the best result within each species; lighter shading indicates the second-best result.

Across species, mRNAWM attains the highest CAI in every source-species stratum. Its rare-codon fraction is also uniformly small, generally ranking immediately behind CodonRL and remaining substantially below GEMORNA, LinearDesign, codonGPT, and Trias. These results indicate that the high CAI achieved by mRNAWM is consistent across proteins originating from diverse mammalian species rather than being driven by a single dominant stratum.

The sequence-composition results show that mRNAWM and CodonTransformer consistently produce the lowest CpG densities. CodonRL most often achieves the lowest UpA density, while mRNAWM is typically the second-best method and is best for *C. griseus* and *S. scrofa*. The updated LinearDesign results do not alter the leading methods for either composition metric.

Table 14: **Cross-species comparison of translation-related properties.** Results are reported independently within each source-species stratum. ↑ and ↓ denote preferred directions.

Metric	Source species	mRNAWM	CT	CodonRL	GEMORNA	LinearDesign	codonGPT	Trias
CAI ↑	*C. jacchus*	**0.9496**	0.9209	0.9240	0.8712	0.7486	0.8045	0.8437
	C. ferus	**0.9464**	0.9222	0.9237	0.8724	0.7458	0.8044	0.8533
	C. griseus	**0.9659**	0.9270	0.9233	0.8664	0.7601	0.8131	0.8433
	H. sapiens	**0.9419**	0.9213	0.9240	0.8718	0.7492	0.8044	0.8458
	I. tridecemlineatus	**0.9413**	0.9214	0.9240	0.8701	0.7484	0.8030	0.8447
	L. weddellii	**0.9471**	0.9230	0.9237	0.8723	0.7482	0.8037	0.8502
	M. mulatta	**0.9488**	0.9216	0.9237	0.8742	0.7477	0.8045	0.8529
	M. murinus	**0.9431**	0.9216	0.9239	0.8714	0.7489	0.8053	0.8450
	M. musculus	**0.9415**	0.9234	0.9238	0.8765	0.7499	0.8086	0.8593
	O. princeps	**0.9342**	0.9242	0.9187	0.8163	0.7654	0.8465	0.7986
	P. m. bairdii	**0.9478**	0.9223	0.9240	0.8741	0.7486	0.8055	0.8540
	R. norvegicus	**0.9493**	0.9242	0.9237	0.8854	0.7522	0.8058	0.8644
	S. scrofa	**0.9413**	0.9297	0.9234	0.8790	0.7503	0.8041	0.8631
Rare-codon fraction ↓	*C. jacchus*	0.00013	0.00020	**0.00000**	0.00536	0.00475	0.00846	0.00842
	C. ferus	0.00009	0.00017	**0.00000**	0.00471	0.00516	0.00794	0.00613
	C. griseus	**0.00000**	0.00025	**0.00000**	0.00658	0.00640	0.00961	0.01250
	H. sapiens	0.00010	0.00021	**0.00000**	0.00555	0.00476	0.00859	0.00865
	I. tridecemlineatus	0.00019	0.00021	**0.00000**	0.00557	0.00505	0.00872	0.00818
	L. weddellii	0.00025	0.00020	**0.00000**	0.00507	0.00518	0.00844	0.00692
	M. mulatta	**0.00000**	0.00019	**0.00000**	0.00474	0.00485	0.00825	0.00650
	M. murinus	0.00006	0.00020	**0.00000**	0.00517	0.00497	0.00838	0.00766
	M. musculus	0.00017	0.00015	**0.00000**	0.00506	0.00478	0.00815	0.00563
	O. princeps	**0.00000**	**0.00000**	**0.00000**	0.01295	0.00518	**0.00000**	**0.00000**
	P. m. bairdii	0.00003	0.00020	**0.00000**	0.00524	0.00479	0.00833	0.00680
	R. norvegicus	0.00016	0.00020	**0.00000**	0.00404	0.00551	0.00899	0.00603
	S. scrofa	**0.00000**	0.00013	**0.00000**	0.00462	0.00494	0.00820	0.00324

For maximum homopolymer run, mRNAWM achieves the best result in seven source-species strata. LinearDesign is best or tied-best in five strata, while Trias is best or tied-best in two. Together with its consistently high CAI and low rare-codon usage, these results show that mRNAWM maintains competitive sequence-composition and homopolymer characteristics across species.

B.6 Additional Analysis of Planning Horizon

We provide additional diagnostics for the planning-horizon ablation on the validation set, focusing on the final objective values, editing behavior, and the stability of the resulting designs.

Detailed scores and editing behavior. Table 17 reports the complete results for $H \in \{1, \ldots, 10\}$. The Final Score increases from 3.510211 at $H = 1$ to 3.529165 at $H = 2$, followed by progressively smaller improvements as the planning depth increases. The score reaches 3.531323 at $H = 5$ and remains nearly unchanged for deeper horizons. Although $H = 6$ gives the numerically highest mean score (3.531389), the variation among $H = 5$–10 is very small.

The editing statistics exhibit a similar saturation pattern. The mean number of selected edits increases substantially from 3.28 at $H = 1$ to 7.62 at $H = 2$, but remains close to eight for all larger horizons. The fraction of examples for which the planner modifies the initial design also stabilizes after $H = 2$. Thus, increasing the horizon beyond short multi-step planning does not simply cause the planner to make progressively more edits; instead, the overall editing magnitude becomes stable while the planner refines which edits and trajectories are selected.

Stability of the final designs. We further examine whether the saturation in Final Score is accompanied by stability in the actual sequences selected by the planner. Table 18 reports the fraction of final designs that are identical to those produced with the default $H = 5$ configuration. Agreement increases substantially as the comparison moves from shallow planning toward moderate lookahead: only 38% of the $H = 1$ outputs match the $H = 5$ designs, compared with 70% at $H = 2$, 78% at $H = 3$, and 86% at $H = 4$.

The neighboring deeper horizon $H = 6$ produces exactly the same final design as $H = 5$ in 92% of cases. Agreement remains high for still deeper horizons, ranging from 88% to 90% for $H = 7$–10. This shows that the plateau in objective value is also associated with substantial convergence in the

Table 15: **Cross-species comparison of sequence-composition properties.** Lower CpG and UpA densities are preferred.

Metric	Source species	mRNAWM	CT	CodonRL	GEMORNA	LinearDesign	codonGPT	Trias
CpG density ↓	*C. jacchus*	0.0164	**0.0156**	0.0695	0.0678	0.0807	0.0555	0.0543
	C. ferus	**0.0141**	0.0158	0.0704	0.0722	0.0827	0.0581	0.0615
	C. griseus	**0.0107**	0.0122	0.0630	0.0553	0.0721	0.0527	0.0393
	H. sapiens	0.0155	**0.0151**	0.0689	0.0665	0.0799	0.0545	0.0532
	I. tridecemlineatus	0.0190	**0.0152**	0.0689	0.0673	0.0799	0.0543	0.0547
	L. weddellii	**0.0148**	0.0153	0.0697	0.0700	0.0810	0.0560	0.0596
	M. mulatta	0.0176	**0.0156**	0.0698	0.0702	0.0815	0.0564	0.0582
	M. murinus	0.0165	**0.0154**	0.0699	0.0685	0.0808	0.0560	0.0571
	M. musculus	**0.0140**	0.0147	0.0681	0.0665	0.0795	0.0560	0.0563
	O. princeps	**0.0250**	**0.0250**	0.0873	0.0951	0.0761	0.0905	0.1190
	P. m. bairdii	0.0159	**0.0149**	0.0682	0.0667	0.0796	0.0542	0.0535
	R. norvegicus	**0.0128**	0.0142	0.0671	0.0675	0.0772	0.0532	0.0565
	S. scrofa	**0.0117**	0.0139	0.0672	0.0691	0.0797	0.0551	0.0577
UpA density ↓	*C. jacchus*	0.0116	0.0125	**0.0093**	0.0191	0.0243	0.0301	0.0209
	C. ferus	0.0119	0.0117	**0.0094**	0.0178	0.0239	0.0291	0.0178
	C. griseus	**0.0084**	0.0119	0.0101	0.0235	0.0285	0.0326	0.0260
	H. sapiens	0.0118	0.0126	**0.0096**	0.0196	0.0248	0.0307	0.0212
	I. tridecemlineatus	0.0111	0.0126	**0.0097**	0.0198	0.0252	0.0311	0.0208
	L. weddellii	0.0119	0.0121	**0.0097**	0.0188	0.0245	0.0303	0.0191
	M. mulatta	0.0102	0.0120	**0.0093**	0.0180	0.0239	0.0295	0.0186
	M. murinus	0.0115	0.0126	**0.0099**	0.0196	0.0253	0.0306	0.0207
	M. musculus	0.0119	0.0129	**0.0104**	0.0196	0.0254	0.0305	0.0183
	O. princeps	0.0103	0.0103	**0.0095**	0.0233	0.0233	0.0112	0.0103
	P. m. bairdii	0.0116	0.0128	**0.0098**	0.0192	0.0253	0.0309	0.0193
	R. norvegicus	0.0149	0.0129	**0.0108**	0.0185	0.0267	0.0328	0.0191
	S. scrofa	**0.0098**	0.0120	0.0105	0.0184	0.0260	0.0318	0.0171

Table 16: **Cross-species comparison of maximum homopolymer run.** Lower values are preferred.

Metric	Source species	mRNAWM	CT	CodonRL	GEMORNA	LinearDesign	codonGPT	Trias
Max homopolymer run ↓	*C. jacchus*	**5.70**	6.38	8.45	7.10	5.89	6.33	6.00
	C. ferus	**5.83**	6.35	8.28	6.93	6.00	6.35	5.90
	C. griseus	**5.00**	6.00	7.11	7.00	5.89	6.67	6.22
	H. sapiens	5.96	6.35	8.32	7.06	**5.89**	6.33	5.99
	I. tridecemlineatus	6.18	6.36	8.28	7.09	**5.88**	6.32	5.99
	L. weddellii	5.92	6.28	8.18	6.91	5.88	6.28	**5.86**
	M. mulatta	**5.52**	6.35	8.40	7.03	5.91	6.32	5.99
	M. murinus	6.02	6.30	8.14	6.96	**5.88**	6.29	5.89
	M. musculus	**5.73**	6.14	7.93	6.82	5.82	6.23	5.76
	O. princeps	11.00	11.00	6.00	6.00	**5.00**	6.00	**5.00**
	P. m. bairdii	**5.70**	6.22	8.09	6.98	5.82	6.24	5.87
	R. norvegicus	6.00	5.89	7.18	6.49	**5.73**	6.18	5.79
	S. scrofa	**5.33**	5.76	7.11	6.15	5.66	5.80	5.48

Table 17: Detailed planning-horizon ablation. Final Score is the objective value of the final selected design. Mean and median edits summarize the selected editing trajectory.

H	Final Score ↑	Mean Edits	Median Edits	Changed Fraction
1	3.510211	3.28	2.0	0.60
2	3.529165	7.62	4.0	0.62
3	3.530391	8.10	5.0	0.62
4	3.530798	8.18	5.0	0.62
5	3.531323	8.32	5.0	0.62
6	**3.531389**	8.28	4.5	0.62
7	3.531268	8.24	5.0	0.62
8	3.531372	8.32	4.5	0.62
9	3.531243	8.34	4.5	0.62
10	3.531243	8.34	4.5	0.62

Table 18: Agreement between final designs produced with each planning horizon and the default $H = 5$ configuration. Two outputs are considered identical only when the complete designed sequence is unchanged.

Comparison	Identical Final Designs
$H = 1$ vs. $H = 5$	38%
$H = 2$ vs. $H = 5$	70%
$H = 3$ vs. $H = 5$	78%
$H = 4$ vs. $H = 5$	86%
$H = 6$ vs. $H = 5$	**92%**
$H = 7$ vs. $H = 5$	90%
$H = 8$ vs. $H = 5$	90%
$H = 9$ vs. $H = 5$	88%
$H = 10$ vs. $H = 5$	88%

planner's final decisions. At larger horizons, the planner occasionally selects alternative sequences, but these changes occur within a very similar Final Score regime rather than yielding consistent additional improvement.

Summary. Together, these results show that the main benefit of increasing the planning horizon arises when moving from shallow to short multi-step lookahead. At moderate horizons, the Final Score, editing magnitude, and selected final designs all become comparatively stable. Although $H = 6$ yields the numerically highest mean Final Score, its difference from $H = 5$ is negligible and not clearly supported by the paired bootstrap analysis. We therefore use $H = 5$ as the default planning horizon, providing sufficient lookahead without unnecessary additional rollout depth.

B.7 Effect of the Edit Budget

We further analyze the effect of the maximum number of executed MPC edits. The edit budget limits the number of real sequence modifications that may be committed during optimization; it is distinct from the planning horizon, which controls how many future edits are imagined when selecting each next action. For this study, all other planner settings are fixed, and only the maximum number of executed edits is varied.

Table 19 reports both the final optimization quality and the behavior of the search trajectory. Increasing the edit budget from 35 to 60 consistently improves the final terminal score, from 3.5270 to 3.5320. The gain becomes progressively smaller as the budget increases: relative to a budget of 80, the mean terminal-score gap decreases from 4.94×10^{-3} at a budget of 35 to 3.60×10^{-4} at a budget of 55. At a budget of 60, this gap reaches zero, and all evaluated examples select the same final solution as with a budget of 80. Increasing the budget further to 65–80 therefore provides no additional improvement in the selected designs.

Importantly, the maximum edit budget should not be interpreted as the number of edits required by a typical design. The median depth of the selected best state is only 13, and the mean depth

Table 19: **Detailed edit-budget analysis.** *Best depth* denotes the position of the ultimately selected state along the accepted optimization trajectory. *Accepted* is the mean number of accepted edits, and *Hit budget* is the fraction of examples whose search terminates after reaching the maximum allowed number of edits. Δ vs. 80 is the difference in mean terminal score relative to the largest tested budget, and *Same as 80* is the fraction of examples whose selected final score is identical to that obtained with a budget of 80.

Budget	Best depth	Median	Accepted	Hit budget	Terminal	Δ vs. 80	Same as 80
35	12.622	13	23.649	67.6%	3.527025	−0.004944	86.5%
40	13.162	13	27.027	67.6%	3.528885	−0.003084	89.2%
45	13.703	13	30.405	67.6%	3.530132	−0.001836	94.6%
50	13.973	13	33.784	67.6%	3.530836	−0.001133	94.6%
55	14.243	13	37.162	67.6%	3.531609	−0.000360	94.6%
60	**14.432**	**13**	**40.541**	**67.6%**	**3.531969**	**0.000000**	**100.0%**
65	14.432	13	43.919	67.6%	3.531969	0.000000	100.0%
70	14.432	13	47.297	67.6%	3.531969	0.000000	100.0%
75	14.432	13	50.676	67.6%	3.531969	0.000000	100.0%
80	14.432	13	54.054	67.6%	3.531969	0.000000	100.0%

saturates at approximately 14.4 once the budget reaches 60. In contrast, the number of accepted intermediate edits continues to grow with the available budget, from 40.5 at a budget of 60 to 54.1 at a budget of 80. This distinction arises because the planner may continue exploring after visiting its best state, while the final output is selected retrospectively from the feasible states encountered along the trajectory rather than being restricted to the final iterate. Consequently, the larger edit budget primarily gives difficult examples more opportunity to explore alternative trajectories without forcing every design to undergo a correspondingly large number of modifications.

The fraction of trajectories that exhaust the available budget remains 67.6% throughout this range. However, this does not imply that the selected solutions are still budget-limited: from a budget of 60 onward, both the mean terminal score and the selected final states are unchanged. Together, these results indicate that an edit budget of 60 is sufficient to remove observable truncation effects on final design quality while avoiding the unnecessary additional search performed with larger budgets. We therefore use a maximum edit budget of 60 in the final configuration.

B.8 Additional Analysis of UTR Initialization

We provide additional analyses of the UTR-initialization experiment in Fig. 3C. All conditions use the same CDS initialization and planner configuration; only the initial UTR scaffold is changed. In particular, we fix the preference vector, planning horizon $H = 5$, beam width 5, 128 proposals per planning step, an edit budget of 60, and a minimum risk-adjusted improvement threshold of -0.003.

The planner continues to refine globin-initialized sequences. The improvement obtained with globin scaffolds is not explained solely by starting from a favorable fixed UTR. As shown in Table 20, mRNAWM obtains a positive utility gain relative to its own initialization under every scaffold condition. Moreover, the planner makes multiple sequence edits after initialization, including UTR substitutions, with larger refinement gains for HBA1 and HBA2 than for the native or HBB initializations. Thus, the globin scaffold acts as an initialization prior rather than as the final solution itself.

Sensitivity among fixed globin scaffolds is limited. To quantify dependence on the identity of the globin scaffold independently of the native-UTR comparison, we compute, for each target, the range of final utilities obtained from HBB, HBA1, and HBA2 initialization. The mean per-target range is only 0.0201, with a median of 0.0157 (Table 21), indicating that the three fixed globin initializations typically lead to similar final solutions despite target-specific variation.

Multi-scaffold selection provides a smaller additional gain beyond the strongest fixed scaffold in aggregate. Relative to HBA2, which has the highest mean utility among the three fixed-scaffold conditions, multi-scaffold selection improves final utility by $+0.0065$ on average (95% CI

Table 20: Planner refinement under different UTR initializations. ΔU_{plan} denotes final utility minus the utility of the corresponding initial state. "Selected edits" is the median edit count of the returned design, and "UTR edits" is the mean number of UTR substitutions relative to that condition's own initialization.

UTR initialization	$\Delta U_{\mathrm{plan}} \uparrow$	Selected edits	UTR edits
Native transcript	$+0.0238$	4.5	2.04
HBB	$+0.0252$	5.5	2.42
HBA1	$+0.0325$	6.0	3.08
HBA2	$+0.0354$	8.0	3.32
Multi-scaffold	$+0.0406$	8.0	3.20

Table 21: Sensitivity to the identity of the fixed globin scaffold and the incremental benefit of multi-scaffold selection. The fixed-scaffold range is computed per target as $\max(U_{\mathrm{HBB}}, U_{\mathrm{HBA1}}, U_{\mathrm{HBA2}}) - \min(U_{\mathrm{HBB}}, U_{\mathrm{HBA1}}, U_{\mathrm{HBA2}})$.

Statistic	Value
Mean fixed-scaffold utility range	0.0201
Median fixed-scaffold utility range	0.0157
90th percentile range	0.0412
Maximum range	0.0749
Multi-scaffold − HBA2, mean	$+0.0065$
95% CI	$[+0.0032, +0.0107]$
Targets improved	52%
Targets tied	48%
Targets worsened	0%

$[+0.0032, +0.0107]$). It improves 52% of targets and matches HBA2 on the remaining 48%, with no target performing worse under the selection rule. This result is consistent with target-dependent scaffold preference: the benefit of multi-scaffold initialization comes from selecting among complementary starting points rather than from a single universally dominant globin scaffold.

Compute interpretation. The multi-scaffold condition selects among independently optimized HBB-, HBA1-, and HBA2-initialized candidates for each target and therefore uses a larger initialization/search budget than a single fixed-scaffold condition. We consequently interpret it as an initialization-selection analysis rather than a matched-compute comparison against HBB, HBA1, or HBA2 alone.

B.9 Preference-Vector Calibration

mRNAWM supports inference-time control through a preference vector over the predicted sequence properties. We calibrate this vector in two stages: we first determine how much preference mass should be assigned to CAI, and then redistribute the remaining mass among non-CAI objectives. Unless otherwise stated, the final preference sweeps use the selected planner configuration with planning horizon $H = 5$, edit budget 60, proposal count 128, and minimum risk-adjusted improvement -0.003.

CAI-weight sweep. We first vary the total preference mass assigned to CAI while preserving the relative allocation among the remaining objectives. Figure 3D shows the resulting trajectory for CAI weights between 0.60 and 1.00. Increasing the CAI preference consistently increases $\Delta\mathrm{CAI}$, confirming that the planner responds to changes in the inference-time objective. However, this improvement is accompanied by progressively worse behavior on the remaining sequence-quality objectives. In particular, moving from the CAI-dominant regime to CAI-only optimization provides only a modest additional CAI gain while causing a pronounced decrease in non-CAI quality. We therefore use a CAI weight of 0.80 as the starting point for the second stage of preference calibration.

Table 22: **Residual preference-vector calibration at fixed $w_{\mathrm{CAI}} = 0.80$.** Composition comprises GC, GC3, CpG density, and UpA density; manufacturability comprises windowed-GC deviation, UG-rich density, homopolymer burden, 6-mer repeat burden, inverted-repeat burden, and upstream-AUG density. Structure comprises start-region MFE, start-region unpaired fraction, global MFE, and ensemble diversity. All reported outcome statistics are computed on the same common-support records. Larger $\Delta\mathrm{CAI}$, ΔQ_{comp}, $\Delta Q_{\mathrm{manuf}}$, and $\Delta Q_{\mathrm{non\text{-}CAI}}$ are preferred.

	Preference mass				Observed improvement			
Preference	**CAI**	**Comp.**	**Manuf.**	**Struct.**	**ΔCAI**	**ΔQ_{comp}**	**$\Delta Q_{\mathrm{manuf}}$**	**$\Delta Q_{\mathrm{non\text{-}CAI}}$**
Base	0.80	0.029	0.084	0.087	+0.01258	-0.04611	+0.03621	-0.00495
Manufacturing-heavy	0.80	0.020	0.130	0.050	+0.01425	-0.05659	+0.04406	-0.00626
Structure-heavy	0.80	0.020	0.050	0.130	+0.00969	-0.02888	+0.01979	-0.00455
Composition-heavy	0.80	0.080	0.060	0.060	+0.01212	-0.03212	+0.02611	-0.00301
GC3/CpG-targeted	**0.80**	**0.100**	**0.060**	**0.040**	**+0.01272**	**-0.01873**	**+0.02855**	**+0.00491**

To summarize the non-CAI trade-off, we define

$$\Delta Q_{\mathrm{non\text{-}CAI}} = \frac{1}{2}\Delta Q_{\mathrm{comp}} + \frac{1}{2}\Delta Q_{\mathrm{manuf}}, \tag{1}$$

where ΔQ_{comp} is the mean normalized quality improvement over GC, GC3, CpG density, and UpA density, and $\Delta Q_{\mathrm{manuf}}$ is the corresponding mean over windowed-GC deviation, UG-rich density, homopolymer burden, 6-mer repeat burden, inverted-repeat burden, and upstream-AUG density. GC and GC3 are evaluated by movement toward the target value 0.50; all remaining properties are oriented such that positive values indicate improvement. The two groups are weighted equally so that manufacturability does not receive additional influence merely because it contains more individual metrics.

Residual preference reallocation. We next fix $w_{\mathrm{CAI}} = 0.80$ and redistribute the remaining 0.20 preference mass among composition, manufacturability, and structural objectives. Table 22 summarizes both the group-level allocations and their resulting exact-property behavior.

The different reallocations expose clear objective-specific trade-offs. The manufacturing-heavy preference provides the strongest manufacturability improvement, but also produces the largest degradation in composition quality. Conversely, the composition-heavy preference substantially reduces this degradation, while giving up part of the manufacturability gain. Increasing the structural preference reduces the CAI gain to $+0.00969$ and does not yield sufficiently consistent improvements across the exact structural measurements to compensate for this loss. Because exact initial structural quantities are unavailable for all records, structural properties are not included in Eq. 1; we instead compare the exact final structural properties on common-support sequences.

These results motivate a more targeted residual allocation. The GC3/CpG-targeted preference increases the weights specifically assigned to GC3 and CpG protection rather than increasing the entire composition group uniformly. It retains essentially the full CAI improvement of the base CAI-80% configuration: $\Delta\mathrm{CAI} = +0.01272$ versus $+0.01258$ for the base allocation. At the same time, the mean GC3 increase is reduced from $+0.01473$ to $+0.01073$, and the increase in CpG density is reduced from $+0.00232$ to $+0.00081$. UpA density also changes from a slight increase ($+0.00007$) to a decrease (-0.00018). Importantly, manufacturability improvement remains positive ($\Delta Q_{\mathrm{manuf}} = +0.02855$). Consequently, the aggregate non-CAI quality changes from -0.00495 under the base allocation to $+0.00491$ under the targeted allocation while preserving nearly the same CAI gain.

The final preference vector is therefore

$$\begin{aligned}
w_{\mathrm{CAI}} &= 0.8000, & w_{\mathrm{GC}} &= 0.0150, & w_{\mathrm{GC3}} &= 0.0350, & w_{\mathrm{CpG}} &= 0.0350,\\
w_{\mathrm{UpA}} &= 0.0150, & w_{\mathrm{windowGC}} &= 0.0058, & w_{\mathrm{UG}} &= 0.0115, & w_{\mathrm{homopolymer}} &= 0.0115,\\
w_{\mathrm{repeat6}} &= 0.0104, & w_{\mathrm{inverted}} &= 0.0104, & w_{\mathrm{uAUG}} &= 0.0104, & w_{\mathrm{startMFE}} &= 0.0059,\\
w_{\mathrm{startUnpaired}} &= 0.0059, & w_{\mathrm{MFE}} &= 0.0178, & w_{\mathrm{ensembleDiv}} &= 0.0104.
\end{aligned}$$

Rare-codon fraction, Kozak score, and ensemble free energy receive zero weight in the final preference. The resulting weights sum to one up to rounding and are normalized by the planner before use.

Target-specific preference probes. The hollow-diamond points in Fig. 3D show additional target-specific probes performed during preference calibration. These include translation mixtures with progressively stronger CAI emphasis, as well as rare-codon-focused and Kozak-focused preferences. They are intended to test directional controllability rather than to provide a quantitative ranking against the final planner configuration. The translation probes show that increasing the relative CAI emphasis shifts the generated sequences toward greater CAI improvement. In contrast, rare-codon-focused optimization provides little direct leverage because the initial rare-codon fraction is already close to its lower bound, while the discrete Kozak objective changes only for a small subset of sequences. Together with the final-config preference sweep, these experiments show that the same trained world model can express distinct optimization behaviors through inference-time preference conditioning, without retraining.

B.10 Zero-Shot Evaluation on Tumor-Associated Antigens

Dataset construction. We construct a tumor-antigen collection from the public CEDAR resource to evaluate zero-shot transfer of the frozen mRNA world model to a cancer-vaccine-relevant target domain. The curation pipeline proceeds from CEDAR cancer evidence to full-length source antigens, validated antigen constructs, human-expression CDSs, and finally full-length mRNA contexts.

The starting CEDAR resource contains 69,742 antigen records and 1,596,681 epitope records. Using the available neoantigen/germline annotations, we collect 428,336 candidate records and retain entries with (i) a cancer-associated context, (ii) a curated human source antigen, (iii) positive experimental evidence, and (iv) a non-viral antigen source. After filtering, the collection contains 192,816 evidence records associated with 7,501 antigen design targets and 7,459 unique parent proteins. Disease context, epitope evidence, assay information, and PubMed references are retained for each target.

Canonical full-length parent proteins are retrieved preferentially from UniProt (UniProt Consortium, 2025), with NCBI/GenPept used as a fallback. We recover 7,205 of the 7,459 parent proteins (96.6%). Germline tumor-associated antigens use the canonical wild-type protein, whereas supported amino-acid substitutions are introduced for mutation-specific targets. Constructs are validated for mutation consistency, reference residues, epitope matching, amino-acid validity, and the current model length limit of 1022 aa.

Among 7,501 candidate targets, 7,110 pass sequence validation. After excluding valid proteins exceeding the model length limit, 5,235 model-ready antigen targets remain, comprising 5,231 germline TAAs and 4 mutation-specific targets. Their median and mean protein lengths are 484 and 505 aa, respectively, with a 95th percentile of 930 aa and a maximum of 1022 aa. Table 23 summarizes the curation process.

Sequence diversity and Stage-1 exposure audit. To characterize protein-level diversity, the model-ready antigens are clustered with MMseqs2 (Steinegger & Söding, 2017) using 50% minimum sequence identity and 80% coverage, producing 2,738 homology clusters. Stage-1 RefSeq exposure is audited at the same similarity threshold, identifying 131 clusters without detectable Stage-1 training exposure. In the present experiment, no TAA-specific optimization, fine-tuning, or model selection is performed: the general mRNAWM checkpoint is frozen and directly applied to the cancer-antigen evaluation collection.

The zero-shot evaluation used here contains 97 antigen targets from the homology-audited evaluation collection. Each target is paired with three UTR scaffold contexts, yielding 291 context-specific mRNA design problems. The evaluated antigens are germline TAAs; accordingly, this experiment tests transfer to unseen TAA protein families rather than generalization to patient-specific neoantigens.

Cancer-type coverage. The curated model-ready collection retains at least one CEDAR disease annotation for every target and spans a broad range of tumor contexts. Melanoma-related evidence is associated with 3,751 targets (71.65%), breast-related evidence with 3,396 targets (64.87%), pancreatic ductal adenocarcinoma (PDAC) with 1,613 targets (30.81%), and any pancreatic cancer annotation with 1,645 targets (31.42%). Frequently represented disease labels include melanoma, breast adenocarcinoma, hepatocellular carcinoma, brain cancer, glioblastoma, lung adenocarcinoma, colon

Table 23: **Summary of the tumor-antigen curation pipeline.** The train/validation/test partition used in separate Stage-2 adaptation experiments is not used for model fitting or selection in the zero-shot experiment reported here.

Dataset statistic	Count
CEDAR antigen records	69,742
CEDAR epitope records	1,596,681
Candidate cancer-antigen records	428,336
Retained evidence records	192,816
Candidate antigen targets	7,501
Unique parent proteins	7,459
Retrieved parent proteins	7,205
Sequence-valid targets	7,110
Model-ready targets	5,235
Germline TAA targets	5,231
Mutation-specific targets	4
Protein homology clusters	2,738
Stage-1-clean homology clusters	131
Successful human-expression CDSs	5,235
Full mRNA contexts after UTR assembly	15,705

adenocarcinoma, and PDAC. These categories are not mutually exclusive, because one antigen may be supported by evidence from multiple tumor contexts.

CDS and mRNA construction. A human-expression CDS is generated for every model-ready antigen while preserving the encoded amino-acid sequence. CDS generation succeeds for all 5,235 targets with no duplicate target IDs, and the resulting dataset is frozen using SHA-256 checksums. Each CDS is subsequently paired with three RefSeq-derived globin UTR scaffolds, HBA1, HBA2, and HBB.

For the CDS-only baselines, one coding sequence is generated for each antigen and is subsequently evaluated under each of the same three UTR contexts. mRNAWM instead performs context-specific planning and can modify both synonymous codons and the bounded subset of UTR positions allowed by the planner. Thus, CDS-level properties compare the generated coding sequences directly, whereas full-mRNA properties additionally capture the effect of context-specific UTR optimization.

Evaluation protocol. All methods are evaluated with the same downstream sequence-property pipeline. To avoid treating multiple UTR contexts associated with the same antigen as independent samples, we first average each property across the three scaffold contexts for a target and then aggregate across antigen targets.

CodonTransformer (CT) serves both as a baseline and as the CDS initialization used by mRNAWM. We additionally compare with CodonRL, codonGPT, Trias, GEMORNA, and LinearDesign. Protein translation is checked explicitly for every generated sequence. RNA structure properties are recomputed using the same folding pipeline for all methods.

Complete zero-shot results. Table 24 reports the complete set of properties available for all compared methods. The main text reports a smaller subset chosen to summarize codon adaptation, sequence composition, manufacturability, and initiation-region accessibility.

The full comparison reveals several complementary optimization regimes. mRNAWM attains the highest CAI while maintaining CpG density close to its CT initialization and substantially below all external baselines. It also produces the lowest homopolymer burden. In contrast, several specialized baselines achieve stronger values on individual properties: LinearDesign strongly optimizes global folding energy and repeat-6 burden, codonGPT produces GC and GC3 values closest to the 0.50 target and the lowest inverted-repeat burden, and GEMORNA obtains the highest start-unpaired fraction. The mRNAWM result is therefore characterized by coordinated improvement across multiple objectives rather than domination of every individual metric.

Table 24: **Complete zero-shot comparison on tumor-associated antigens.** Metrics are first averaged over the three UTR contexts associated with each antigen and then aggregated across antigen targets. Bold shaded cells indicate the best mean result and lightly shaded cells indicate the second-best result. ↑ and ↓ denote preferred directions. GC and GC3 are evaluated by proximity to the target 0.50, and ensemble diversity by proximity to the target 0.10.

Group	Metric	mRNAWM	CT	CodonRL	GEMORNA	Linear	codonGPT	Trias
Translation	CAI ↑	**0.9458**	0.9321	0.9334	0.8821	0.7467	0.7983	0.8861
Sequence Composition	GC ↔ 0.50	0.5868	0.5850	0.6325	0.6014	0.5812	**0.5418**	0.6013
	GC3 ↔ 0.50	0.8166	0.8002	0.9665	0.8506	0.7753	**0.6426**	0.8560
	CpG density ↓	0.0176	**0.0140**	0.0572	0.0587	0.0674	0.0463	0.0481
	UpA density ↓	0.0157	0.0152	**0.0142**	0.0209	0.0266	0.0326	0.0188
Manufacturability	Homopolymer burden ↓	**0.5457**	0.6079	0.6883	0.6581	0.6024	0.6296	0.6048
	Repeat-6 burden ↓	0.2680	0.2737	0.2937	0.2317	**0.1631**	0.1750	0.2718
	Inverted-repeat burden ↓	0.0454	0.0500	0.0424	0.0438	0.1087	**0.0323**	0.0584
RNA Structure	Full-mRNA MFE/nt ↓	−0.3928	−0.3915	−0.3887	−0.3798	−0.6089	−0.3412	−0.3914
	Start MFE/nt ↑	−0.2722	−0.2906	−0.2737	−0.2822	−0.3262	−0.2708	−0.2896
	Ensemble diversity/nt ↔ 0.10	0.1915	0.2225	0.2137	0.2304	**0.0449**	0.2260	0.2166
	Start unpaired fraction ↑	0.4657	0.4446	0.4633	**0.4680**	0.4119	0.4603	0.4469

Table 25: **Validity of zero-shot TAA designs.** All evaluated generation methods preserve the target protein sequence. Constraint statistics are additionally reported for the CT-to-mRNAWM planning trajectory.

Method	Success	Valid ORF	Protein match
CodonRL	1.000	1.000	1.000
codonGPT	1.000	1.000	1.000
Trias	1.000	1.000	1.000
GEMORNA	1.000	1.000	1.000
LinearDesign	1.000	1.000	1.000
mRNAWM	1.000	1.000	1.000

Relative to its CT initialization, mRNAWM increases CAI from 0.9321 to 0.9458 and reduces homopolymer burden from 0.6079 to 0.5457, repeat-6 burden from 0.2737 to 0.2680, and inverted-repeat burden from 0.0500 to 0.0454. The start-region MFE becomes less negative (-0.2906 to -0.2722), the start-unpaired fraction increases from 0.4446 to 0.4657, and ensemble diversity moves toward its target of 0.10 (0.2225 to 0.1915). These improvements are accompanied by modest trade-offs in GC3 and CpG density, consistent with the CAI-dominant optimization preference used for this experiment.

Validity and constraint satisfaction. All external baseline methods successfully generate valid, protein-preserving ORFs for the complete evaluation collection, as summarized in Table 25. mRNAWM likewise preserves the encoded protein for every generated sequence. In addition, the planner increases hard-constraint satisfaction relative to its CT initialization, from 96.9% to 98.6%. Approximately 1.7% of contexts that violate a hard constraint at initialization are repaired during planning, while the constraint-fallback rate is 1.4%.

Planner behavior. The zero-shot planner does not simply exhaust the available edit budget. Although optimization permits up to 60 edits, the selected designs use 8.24 edits on average and 8 edits at the median. Only 2.4% of inputs retain the initialization without an accepted edit. This indicates that the improvements above arise from relatively sparse, selective interventions rather than wholesale sequence rewriting.

Overall, the TAA study complements the in-distribution benchmark by testing whether a frozen mRNA world model remains useful after a therapeutically meaningful shift in encoded protein targets. The results show that the planner preserves exact protein identity while improving multiple sequence and structure properties from a strong CT initialization, without any TAA-specific fine-tuning. Because the evaluated collection contains germline TAAs, these results should be interpreted

Table 26: **Constraint behavior of mRNAWM during zero-shot TAA optimization.**

Statistic	Value
CT initialization feasible	0.969
mRNAWM final feasible	0.986
Initially infeasible contexts repaired	0.017
Constraint fallback	0.014

as evidence of zero-shot robustness to tumor-associated antigen proteins rather than as evidence of generalization to patient-specific neoantigens.

B.11 Additional Qualitative Analysis of World-Model Planning

We provide additional trajectory-level analyses complementing Section 4.5. The main-text example exposes the interaction between imagined return, executed edits, and realized multi-objective changes within a single trajectory. Here, we use additional held-out sequences to examine complementary aspects of the same controller: changes in the local RNA structural ensemble, non-greedy action selection, asynchronous evolution of multiple sequence properties, and planning over a substantially longer transcript. These examples are intended as qualitative views of the learned world-model controller rather than as additional quantitative performance comparisons. In every trajectory, mRNAWM encodes the current full-mRNA state, imagines multi-step futures under candidate edit sequences, executes the first action of the selected rollout, and replans from the resulting sequence state.

Additional structural analysis of XM_082670439.1:cds:2. The main-text analysis of XM_082670439.1:cds:2 focuses on the spatial distribution of edits, objective-aligned property changes, and the planning signals produced by the world model. Figure 5 provides a complementary structural view by tracking nucleotide-level unpaired probabilities around the translation start site throughout the same 13-edit trajectory.

The accessibility pattern does not evolve as a uniform drift from the initial to the final state. Instead, the heatmap contains several distinct structural regimes, with pronounced changes appearing after particular edits, including transitions around the middle and later portions of the trajectory. These changes extend across many positions within the start-region window even though each executed action is itself a discrete nucleotide substitution or synonymous codon edit. Thus, an executed edit changes not only the sequence symbol at its action location but also the structural state on which the next round of world-model imagination is conditioned. This provides a nucleotide-level view of the state transitions underlying the replanning process shown in the main text.

XM_081097192.1:cds:2: non-greedy world-model planning. XM_081097192.1:cds:2 provides a useful example of the distinction between multi-step model-based planning and one-step mutation ranking. Figure 6 shows the planning signals over the selected 18-edit trajectory. The world model initially predicts substantial remaining return, which decreases rapidly as optimization progresses. At the same time, the imagined stopping depth changes with the current sequence state rather than remaining fixed throughout the trajectory.

More importantly, the selected trajectory contains an accepted action with a slightly negative immediate realized risk-adjusted improvement while the predicted return of the corresponding imagined continuation remains positive. Subsequent replanning from the modified sequence produces later actions with positive realized improvement. This observation should not be interpreted as showing that the locally unfavorable edit is beneficial in isolation. Instead, it illustrates the decision rule used by mRNAWM: candidate actions are evaluated as the first transitions of imagined multi-step futures rather than solely according to their immediate realized gain.

After the selected action is executed, the remainder of the imagined rollout is not followed open-loop. The resulting sequence is encoded again, a new set of futures is imagined, and the next action is selected from the updated state. The example therefore visualizes the interaction between model-based lookahead and closed-loop replanning that is absent from a purely greedy sequence search.

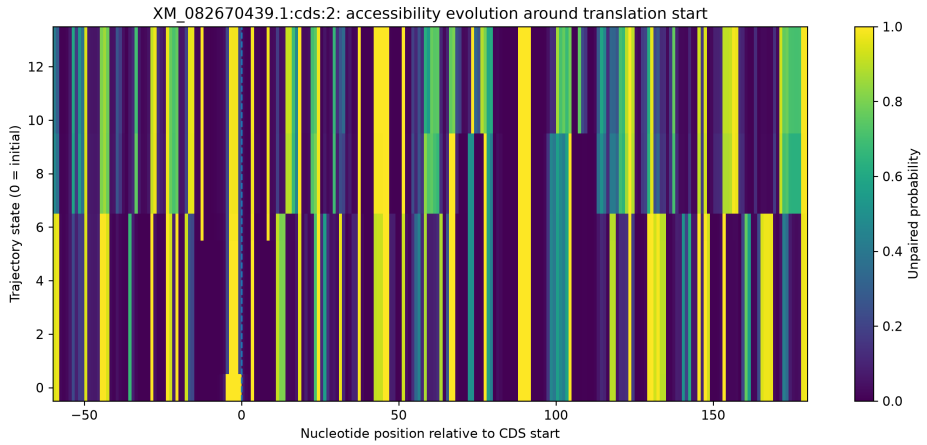


Figure 5: **Start-region accessibility trajectory for XM_082670439.1:cds:2.** Colors denote nucleotide-level unpaired probability around the CDS start across the initial state and 13 accepted edits. Position zero denotes the CDS start.

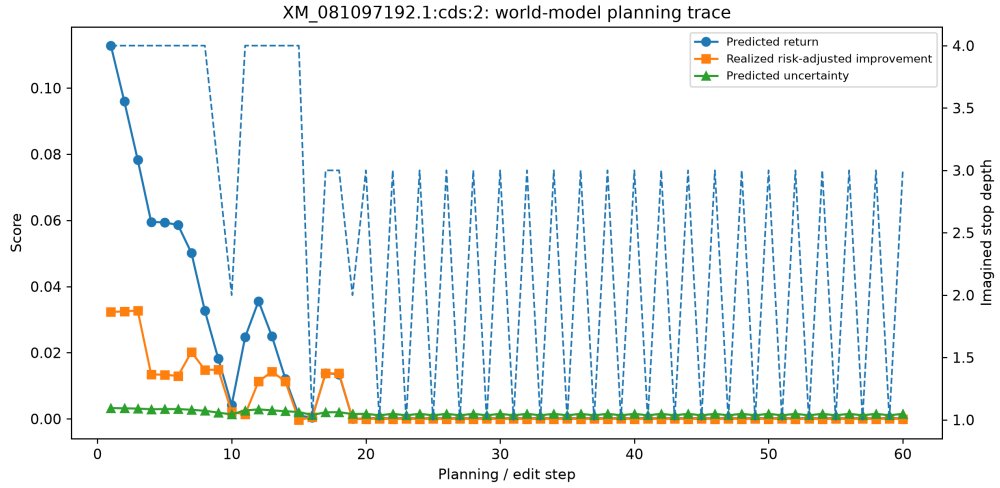


Figure 6: **World-model planning signals for XM_081097192.1:cds:2.** Predicted return, realized risk-adjusted improvement, predicted uncertainty, and imagined stopping depth are shown across the edit trajectory.

XM_054321017.1:cds:2: asynchronous multi-property evolution. Figure 7 provides a complementary view of the optimization process in property space. Rather than displaying the complete objective-aligned vector used in Figure 4, this visualization tracks changes from the initial state for a representative set of sequence and structural properties, with sign alignment applied where appropriate.

The individual properties evolve on markedly different schedules during the selected 23-edit trajectory. CAI increases progressively over successive CDS edits, whereas start-region accessibility changes in several discrete stages. The GC3 and start-MFE components undergo comparatively large changes in the opposite plotted direction, while the homopolymer component changes abruptly near the middle of the trajectory. Thus, the state cannot be described as moving monotonically along a single scalar property axis: different components respond at different steps and with different magnitudes.

This behavior is consistent with the role of the learned world model in mRNAWM. Each executed action produces a new multidimensional sequence state, and the subsequent rollout is imagined from

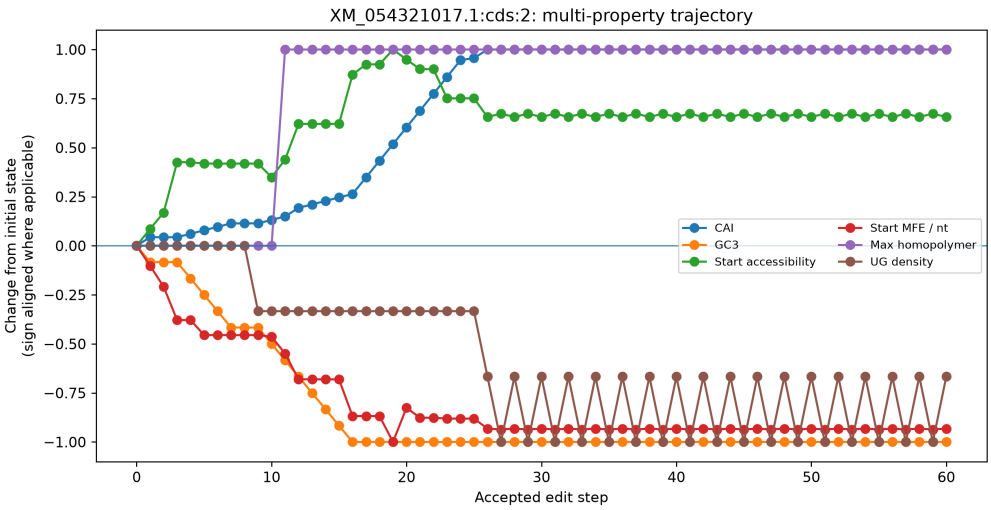


Figure 7: **Multi-property trajectory for XM_054321017.1:cds:2.** The curves show changes from the initial state for selected codon-, sequence-, and structure-related properties across the edit trajectory; signs are aligned where applicable.

that updated state under the complete preference vector. The planner can therefore move through intermediate states in which some measured properties have already changed substantially while others remain near their initial values. The trajectory in Figure 7 makes this asynchronous evolution visible directly.

XM_082344848.1:cds:2: an alternative accessibility trajectory. The start-region structural evolution of XM_082344848.1:cds:2 is shown in Figure 8. Within its selected 20-edit trajectory, the unpaired-probability profile passes through several visibly distinct regimes rather than changing smoothly. Early states, intermediate states, and the states near the selected endpoint exhibit different accessibility patterns across a broad portion of the window surrounding the CDS start.

The trajectory is also qualitatively different from that of XM_082670439.1:cds:2 in Figure 5. The comparison is not intended to imply that a particular edit has the same structural effect across transcripts; rather, it highlights that structural responses are strongly dependent on the current sequence context. In the world-model formulation, these sequence-specific structural states form part of the current condition from which subsequent transitions are imagined. Consequently, repeated replanning can encounter different structural regimes for different transcripts even under the same global optimization configuration.

XM_045998867.1:cds:2: planning over a long full-mRNA sequence. XM_045998867.1:cds:2 provides a substantially longer spatial planning problem, with a full-mRNA length of approximately 1.7 kb in Figure 9. The accepted actions span the 5'UTR, CDS, and the early portion of the 3'UTR, with edited positions separated by several hundred nucleotides. Consecutive planning decisions are therefore not restricted to neighboring sequence locations: the controller can move between widely separated parts of the transcript as the current state and predicted future return change.

This example also clarifies what receding-horizon planning means for a long sequence. mRNAWM does not construct a single fixed edit path covering the entire transcript at the beginning of optimization. At each iteration it considers candidate futures from the current encoded state, executes one selected edit, and then solves the planning problem again. As a result, physical distance along the mRNA does not determine the temporal order of actions. An edit in a UTR can be followed by a distant CDS edit, or vice versa, whenever the newly imagined rollout assigns the latter action higher future utility. Figure 9 visualizes this state-dependent action selection over a substantially larger nucleotide coordinate range than the main-text example.

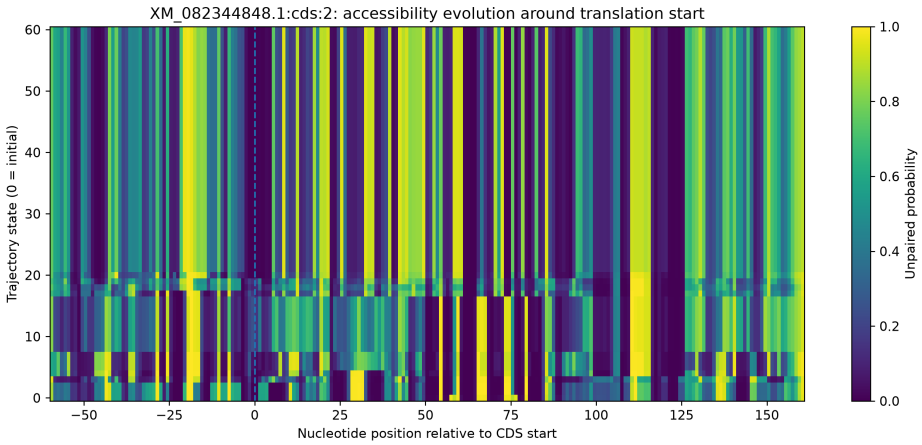


Figure 8: **Start-region accessibility trajectory for XM_082344848.1:cds:2.** Colors denote nucleotide-level unpaired probability around the CDS start across the initial state and the edit trajectory. Position zero denotes the CDS start.

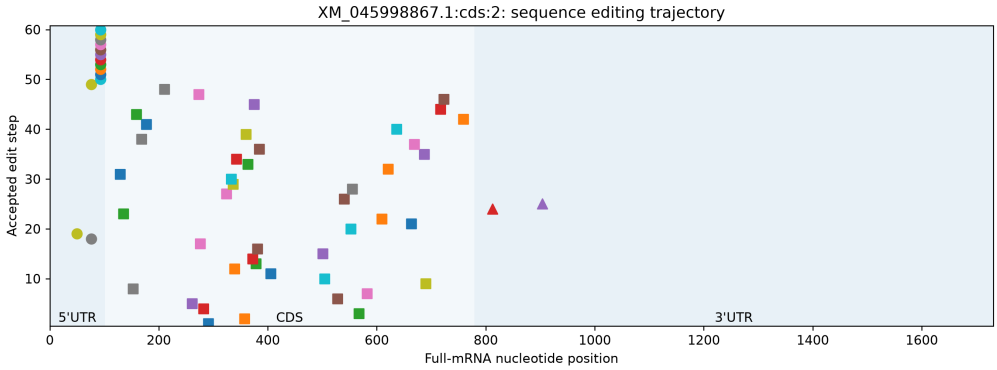


Figure 9: **Full-mRNA editing trajectory for XM_045998867.1:cds:2.** Accepted edits are shown by full-mRNA nucleotide position and execution step, with marker shape indicating the edited region.

C Limitations and Future Work

The present study establishes mRNAWM as a computational framework for multi-objective full-length mRNA design. Its current limitations are primarily limitations of supervision, search scope, and available evaluation data, rather than restrictions inherent to the world-model formulation itself. Each therefore suggests a direct path for extending the framework.

From computational objectives to experimentally measured function. The current model is trained and evaluated using sequence-derived and RNA-structure-derived properties, including codon adaptation, nucleotide composition, motif burdens, and ViennaRNA-based structural quantities. These objectives provide reproducible and scalable supervision for developing the world-model and planner, but they do not directly measure cellular expression, transcript half-life, innate immune activation, protein yield, or therapeutic efficacy. A natural next step is to replace or augment individual computational properties with experimentally measured endpoints. Because mRNAWM separates its latent dynamics from preference-conditioned control, experimentally observed measurements can be incorporated as additional prediction targets without changing the overall planning formulation. This makes the framework well suited to iterative design–build–test cycles in which newly measured outcomes are used to refine the learned transition and value models.

Richer structural supervision. RNA-structural objectives in the present work are derived from thermodynamic folding calculations and are supervised sparsely during training to control computational cost. The resulting multi-fidelity design allows the model to learn structural effects while retaining tractable training and inference, but more informative structural supervision could further improve the fidelity of imagined trajectories. Future extensions could incorporate larger-scale partition-function supervision, alternative folding models, experimentally informed probing data, or learned structural representations. The same uncertainty-aware planner can naturally accommodate such richer structural predictors, since uncertainty is already propagated into branch scoring rather than treating every predicted property as equally reliable.

Broader tumor-antigen and somatic-mutation coverage. The tumor-associated-antigen experiment evaluates zero-shot transfer to a therapeutically relevant target domain without TAA-specific fitting. The current curated collection, however, contains very few mutation-specific targets; consequently, the reported zero-shot analysis primarily measures transfer to germline tumor-associated antigens rather than the full diversity of patient-specific neoantigens. This limitation is principally one of available curated sequence evidence. Expanding the evaluation set with larger collections of validated somatic mutations, HLA-linked neoantigen annotations, and patient-specific antigen sequences would enable a more direct test of individualized cancer-vaccine design. Importantly, such an extension does not require a change to the protein- preserving planning formulation: a mutated antigen sequence can simply be treated as the fixed target protein for which an optimized mRNA is designed.

Expanding the edit space. The current planner deliberately uses a conservative action space consisting of synonymous CDS substitutions and single-nucleotide substitutions in the 5' and 3' UTRs. All edits preserve sequence length, and CDS edits preserve the encoded protein by construction. This restricted action space simplifies validity guarantees and makes action-conditioned transition learning substantially easier. Future versions could extend the same framework to motif-level substitutions, short UTR insertions or deletions, poly(A)-related design variables, or structured multi-nucleotide edits. A hierarchical action representation would be particularly attractive: high-level actions could first select a region or design operation, while lower-level actions instantiate a valid nucleotide-level modification. Such an extension would enlarge the accessible design space while retaining the model-based planning principle used here.

Longer sequences and more scalable representations. The present implementation imposes practical sequence-length limits to keep training, cached protein representations, and RNA folding tractable. These limits are implementation choices rather than conceptual constraints of the latent-dynamics formulation. Longer proteins and transcripts could be supported through chunked or hierarchical protein encoders, more aggressive multiscale UTR representations, and sparse or linear-time sequence models. Similarly, expensive full-transcript structural calculations could be replaced during search by increasingly accurate learned surrogates, with exact oracle evaluation retained only for final candidate validation.

Preference specification and adaptive objectives. In the current experiments, the planner is controlled by an explicitly specified preference vector. The preference-calibration experiments show that the same trained world model can express different design behaviors without retraining, which provides a useful interface for multi-objective control. A further extension would be to infer preferences from experimental or application-specific constraints rather than specifying them manually. For example, preference vectors could be selected from desired property intervals, learned from historical successful constructs, adapted to cell-type-specific measurements, or optimized jointly with a downstream experimental objective. Because the transition model itself is preference-independent, such changes can be introduced at the control layer while reusing the same learned dynamics.

Closed-loop model refinement. The current planner performs closed-loop replanning within a fixed learned world model: after every committed sequence edit, the resulting sequence is re-encoded and planning is repeated from the updated state. An analogous loop can be introduced across experimental rounds. Candidate mRNAs proposed by the planner could be synthesized and measured, their outcomes added to the training set, and the world model periodically updated with new observations. This would allow uncertainty to guide not only sequence optimization but also

data acquisition, concentrating experimental effort on regions of sequence space where the model is uncertain yet predicts substantial potential gain.

Toward experimentally grounded mRNA world models. Taken together, these directions suggest a progression from the present computational model toward an experimentally grounded mRNA world model. The core architecture—a protein-conditioned state representation, action-conditioned latent dynamics, vector property prediction, preference-conditioned value estimation, and uncertainty-aware receding- horizon planning—does not depend on any particular choice of property oracle. As richer measurements become available, the same framework can therefore be extended from computational sequence optimization toward predictive models of experimentally observed mRNA behavior, while preserving the ability to reason over multiple coupled objectives before committing sequence changes.

D Ethical Considerations

This work develops a computational framework for mRNA sequence design and does not involve new human-subject recruitment, clinical intervention, or wet-lab experimentation. The general-domain experiments are constructed from public RefSeq transcript records, and the tumor-antigen analysis uses curated evidence from the public CEDAR resource. The reported results should therefore be interpreted as computational evidence about sequence-design behavior rather than evidence of clinical efficacy, safety, or therapeutic benefit.

Clinical interpretation. mRNAWM optimizes computational properties associated with translation, sequence composition, manufacturability, and RNA structure. These quantities are useful design proxies, but they do not directly measure protein expression in vivo, transcript half-life, immune activation, toxicity, or therapeutic outcome. Accordingly, sequences produced by the model should not be regarded as clinically validated constructs without appropriate experimental validation. This distinction is especially important for the tumor-associated-antigen study, which evaluates transfer to germline TAAs rather than individualized patient-specific neoantigens.

Responsible use of sequence optimization. Like other sequence-design methods, a sufficiently general optimization system could potentially be applied outside the benign therapeutic settings studied here. The present work focuses on protein-preserving optimization of predefined mRNA targets and evaluates sequence-level design properties rather than biological pathogenicity or virulence. Nevertheless, deployment of more capable future versions should be accompanied by appropriate controls on target selection, experimental validation, and domain-specific biosafety review. In particular, computational improvement of generic sequence objectives should not be treated as sufficient justification for synthesizing or experimentally testing a construct whose biological function has not been independently assessed.

Transparency and reproducibility. We report the model architecture, property definitions, training objectives, planner configuration, baseline settings, and evaluation procedures in detail throughout the appendices. Data splits are defined at the exact protein-sequence-cluster level, and the major sources of randomness are controlled through fixed seeds or record-deterministic procedures. These choices are intended to make the scope and limitations of the reported claims auditable and to reduce the risk that apparent gains arise from undocumented preprocessing or evaluation differences.

Scope of the tumor-antigen analysis. The curated tumor-antigen collection contains predominantly germline tumor-associated antigens, with only a small number of mutation-specific targets in the underlying collection. We therefore avoid interpreting the present zero-shot experiment as evidence for patient-specific neoantigen design. Expanding evaluation to larger and better characterized somatic-mutation datasets, together with experimental validation, is an important prerequisite for drawing conclusions in personalized cancer-vaccine settings.

E BROADER IMPACT

mRNA therapeutics require the simultaneous consideration of many interacting sequence properties, and current design workflows often separate these objectives into different optimization stages or rely on fixed scalarized criteria. mRNAWM explores an alternative formulation in which the consequences of sequence edits are modeled explicitly and candidate edit trajectories are evaluated before changes are committed. If further validated experimentally, this type of model-based design framework could help make computational mRNA engineering more systematic, controllable, and reusable across different design objectives.

One potential benefit is improved *design efficiency*. A learned world model can reuse predictive dynamics across many preference settings, allowing multiple objective trade-offs to be explored without training a separate sequence generator or optimization policy for every new design specification. Receding-horizon planning may also reduce the need to exhaustively evaluate large numbers of candidate sequences with expensive structural or experimental oracles, since the model can prioritize promising regions of the design space before high-cost validation.

A second potential impact is improved *controllability*. Because the transition model is preference-independent while the policy and value functions are preference-conditioned, one trained model can express different optimization behaviors at inference time. This separation may be useful in practical settings where design priorities vary across applications—for example, when different constraints on translation, RNA structure, sequence composition, or manufacturability must be balanced for different targets or development stages. The preference-calibration results in this work provide an initial demonstration of such post-training control.

More broadly, the framework illustrates how biological sequence engineering can be viewed as a model-based decision problem rather than solely as direct sequence generation. Many biological design tasks involve deterministic sequence edits whose downstream functional consequences are difficult to predict and whose effects interact across multiple objectives. Learning reusable action-conditioned dynamics and planning over predicted future states may therefore be applicable beyond mRNA optimization, including other settings in which local sequence modifications produce coupled, long-range changes in molecular properties.

The practical impact of such systems will ultimately depend on the quality of their biological supervision. The present study relies primarily on computational sequence and RNA-structure objectives; future integration of experimentally measured expression, stability, immune-response, and other functional endpoints could make the learned world model substantially more biologically grounded. In this sense, we view mRNAWM primarily as a computational foundation for closed-loop sequence design, where predictive modeling, uncertainty-aware planning, and experimental feedback can be progressively integrated rather than as a replacement for experimental validation.