
000 001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 Emergent Biological Realism in RL-Trained DNA Language Models

Anonymous Authors¹

Abstract

Reinforcement learning has driven the mass adoption of large language models by unlocking unexpected emergent capabilities, yet this approach remains largely underexplored for generative DNA models. We investigate whether similar post-training techniques can induce emergent biological realism in DNA language models, using plasmid generation as a testbed due to plasmids' relative simplicity, well-characterized functional constraints, and ubiquity in biotechnology. Using Group Relative Policy Optimization with a reward function based on constraints from engineered biology, our model achieves 97% quality control pass rate compared to 6% for the pretrained baseline. Remarkably, beyond explicitly optimized features, the model exhibits emergent biological parallels: generated sequences match natural plasmids in thermodynamic stability, codon usage patterns, and ORF length distributions, properties not directly encoded in the reward function. These results suggest that RL post-training can steer DNA language models toward biologically coherent regions of sequence space, analogous to how such techniques unlock emergent capabilities in natural language models.

1. Introduction

Plasmids are extrachromosomal DNA sequences, often found in bacteria, capable of replication independent of a host genome (Lederberg, 1952). These genetic elements are ubiquitous in biotechnology, serving as the primary vectors for protein expression, gene editing, and emerging DNA therapeutics (Kutzler & Weiner, 2008; Prather et al., 2003). Despite their widespread utility, plasmid engineering remains a complex, high-dimensional optimization problem. Traditional workflows are cost intensive and heuristic

driven, often requiring iterative cycles of manual sequence editing and experimental validation to resolve structural instabilities (Oliveira et al., 2009; Meng & Ellis, 2015). Suboptimal plasmid architectures, plagued by incompatible regulatory elements or unstable repeat regions, frequently lead to metabolic burden, reduced expression efficiency, and manufacturing bottlenecks (Brophy & Voigt, 2011; Wu et al., 2016).

Current approaches to plasmid design rely heavily on tacit domain knowledge and piecemeal assembly of genetic parts. Designers must simultaneously optimize for competing objectives such as copy number, transcriptional output, and host viability while navigating the strict biophysical constraints of DNA folding and context dependent regulatory interactions (Deng et al., 2025; Fung et al., 2025).

The dramatic success of reinforcement learning post-training in natural language processing, such as improved reasoning, instruction following, and unexpected generalization highlights the potential of RL post training in DNA language models (Wei et al., 2022; Berti et al., 2025). We investigate this using plasmid generation as a testbed, applying Group Relative Policy Optimization to the PlasmidGPT foundation model (Shao et al., 2024a). Beyond dramatically improving quality control pass rates (97% vs. 6% baseline), we observe emergent properties not explicitly optimized: generated sequences match natural plasmids in thermodynamic stability, codon usage patterns, and ORF length distributions. These results suggest that RL post-training may unlock similar emergent capabilities in genomic models as it has in language models, steering generation toward biologically coherent sequence space regions through reward-guided optimization.

2. Background

2.1. DNA Language Models

Natural language processing and genomics have developed in parallel, often with algorithms developed for bioinformatics used in NLP and vice versa (Durbin et al., 1998). Natural language models pretrained on massive amounts of data have displayed emergent capabilities and remarkable utility by understanding the structure of language as a whole (Wei et al., 2022). These capabilities have vastly

¹Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

increased the utility of language models in a wide range of tasks, and are one of the primary reasons why large language model adoption has skyrocketed in the last few years. This has inspired the application of many similar techniques to genomic language models, resulting in models capable of impressive performance on biological tasks including variant prediction (Ji et al., 2021), transcription factor binding site identification (Nguyen et al., 2023), and even whole-genome generation (Nguyen et al., 2024).

Plasmid DNA has received relatively little attention compared to other sequence types despite its importance in biomanufacturing and research. OriGen (Martinson et al., 2025) introduces a generative model to produce previously undiscovered origins of replication (ORIs) but does not model whole sequences. PlasmidGPT (Shao et al., 2024a; Cunningham et al., 2025) uses modern language modeling techniques to develop a generative model for whole plasmid sequences, and later work expands on this by synthesizing whole plasmids generated using a fine tuned version of the PlasmidGPT model (Cunningham et al., 2025). We use this as a base model and apply post-training techniques to improve results significantly.

2.2. Plasmid Design

Lab-designed plasmids are short circular DNA molecules (typically 2–15 kb) that must contain multiple functional components arranged in precise configurations. At minimum, a viable plasmid requires: (i) an origin of replication to enable autonomous replication, (ii) a selection marker (e.g., antibiotic resistance gene) for identifying successfully transformed cells, and (iii) a cloning site where genes of interest can be inserted.

The search space of valid plasmids is massive due to combinatorial explosion across components. There are many types of each component on the scale of hundreds to thousands per component, and additional regulatory elements (promoters, terminators, enhancers) and reporters may be required depending on the application. Multiple instances of certain components may be necessary, and the ordering and spacing of these elements significantly impacts function. Beyond sheer combinatorics, designers must navigate complex biological constraints including compatibility requirements (specific ORIs only function in certain hosts), physical stability issues (repeat regions can fold and bind to each other), and other design challenges.

3. Methods

We follow an established method that starts with a base model, fine-tunes it using supervised fine-tuning (SFT), and then applies reinforcement learning (RL) to optimize for specific attributes in the output (Figure 1).

3.1. Supervised Fine-Tuning

Supervised fine-tuning (SFT) was performed on a curated corpus of *E. coli* plasmid sequences assembled from PlasmidScope and Addgene. After deduplication and quality filtering, approximately 15k circular plasmids (≤ 30 kb) were retained, excluding linear entries, fragments, and incomplete records. Sequences were tokenized using the original PlasmidGPT byte-pair DNA tokenizer. The pretrained PlasmidGPT model was fine-tuned using an autoregressive next-token prediction objective with gradient accumulation and learning-rate warmup over three epochs.

3.2. Reinforcement Learning with GRPO

We implement a configurable reinforcement learning pipeline for plasmid design that uses Group Relative Policy Optimization (GRPO) (Shao et al., 2024b) with a domain-specific reward function described below. At each training iteration, the model generates a batch of candidate plasmids via autoregressive rollouts conditioned on short nucleotide prompts. Prompts are either stochastic (4–25 bp random seeds, excluding "ATG", used to promote rollout diversity) or structured (partial "cassette" seeds encoding canonical marker genes such as antibiotic-resistance or fluorescent reporters). Each candidate sequence is evaluated via our reward function, which captures structural plausibility, cassette organization, repeat content, and other biologically motivated constraints. GRPO is then applied to update the model parameters using these sequence-level rewards, enabling the policy to progressively shift toward generating plasmids with higher predicted validity.

3.3. Reward Function Design

The reward function scores each generated plasmid according to its structural plausibility and expected stability. It is composed of three conceptual components:

Functional annotation scoring: Lightweight annotations identify origins of replication, promoters, terminators, coding sequences (CDS), and selectable markers, which are then scored according to a configuration designed with subject matter expert (SME) input to reflect biologically reasonable quantities (e.g., exactly one origin of replication and at least one selectable marker). CDS regions are identified using Pyrodigal (Hyatt et al., 2010), a gene prediction tool that detects coding sequences based on statistical patterns rather than homology search. To encourage coherent gene cassettes, this component also includes a location-aware bonus for promoter → CDS → terminator arrangements that appear in the correct order and within a reasonable proximity window.

Length prior: A length prior favors plasmid sizes within typical experimental ranges (5–15 kb) preferred for plas-

110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164

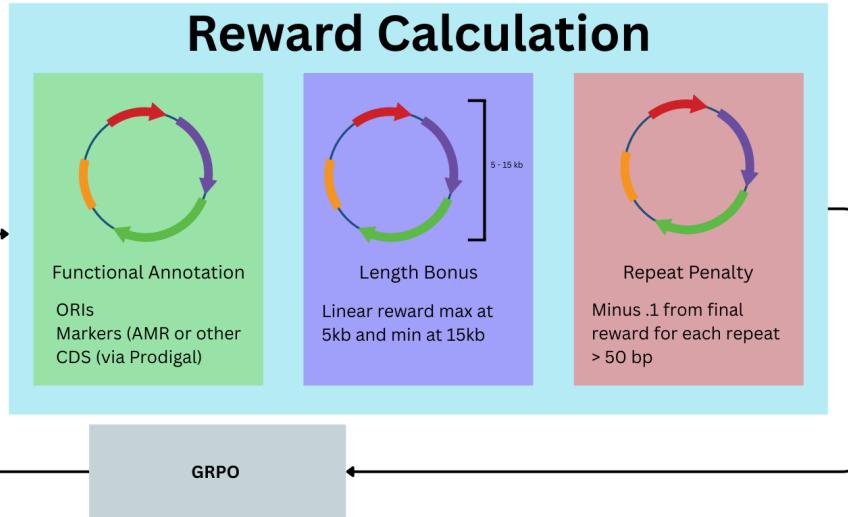


Figure 1. Overview of the Plasmid-RL training pipeline. Starting from the pretrained PlasmidGPT base model, we apply supervised fine-tuning on curated plasmid sequences, followed by reinforcement learning with Group Relative Policy Optimization using a biologically-motivated reward function that evaluates functional annotations, length constraints, and repeat content.

mid construction by giving a linearly increasing reward for lengths between 5 and 15 kb with a maximum reward at 5 kb and no reward for sequences longer than 15 kb.

Repeat penalty: A repeat penalty down-weights sequences containing long exact repeats that are associated with instability or recombination, specifically penalizing .1 reward for each repeat of length 50 bp or greater.

These terms are combined into a single scalar in [0, 1], yielding a fast and interpretable proxy for “plasmid-likeness” during reinforcement learning.

4. Experiments

4.1. Plasmid Quality Control and Uniqueness

We evaluate three model variants on held-out prompts not seen during training. For each model, we sampled 50 rollouts with two prompts: (i) a minimal prompt (single ATG codon) to test unconditional generation capability, and (ii) a structured prompt containing a complete GFP expression cassette to test the model’s ability to build around provided components. These prompts were deliberately excluded from the training corpus to ensure evaluation reflects generalization rather than memorization.

4.1.1. VALIDITY ASSESSMENT

In silico plasmid validity was assessed using a bioinformatics quality-control pipeline that leverages BLAST based tools, requiring exactly one origin of replication ($\geq 95\%$ identity and coverage), one or two antimicrobial resistance genes ($\geq 99\%$ identity and coverage), and no internal repeats longer than 50 bp (Altschul et al., 1990). This pipeline has been validated as a reliable proxy for experimental synthesis success (Cunningham et al., 2025). While we do not perform wet-lab validation in this work, our focus is on establishing that RL post-training can successfully navigate the plasmid design space in silico, laying groundwork for future conditional generation systems where user-specified designs can be experimentally validated.

4.1.2. UNIQUENESS ASSESSMENT

To assess whether generated plasmids represent genuinely new designs rather than minor variants of existing constructs, we compute similarity to known sequences using the NCBI BLASTn API. Each generated plasmid is assigned to one of three categories based on identity and query-coverage thresholds: sequences with $\geq 99\%$ identity and $\geq 95\%$ coverage are classified as **Exists**; those with $\geq 95\%$ identity and

165
166 **Table 1.** Novelty and diversity metrics across model variants. RL
167 achieves dramatically higher quality (97% pass rate) with reduced
168 diversity in samples.

MODEL	QC PASS RATE	% NOVEL	DIVERSITY
BASE	6%	95.5%	0.926
SFT	11%	100%	0.886
RL	97%	88%	0.391

175 $\geq 80\%$ coverage are **Similar**; and all others are classified as
176 **Novel**.

177 This categorization is based on large scale data curation
178 efforts such as PLSDB that use these thresholds of similarity
179 to attempt to de-duplicate plasmids for not adding any
180 additional value to a dataset (Galata et al., 2018).

182 4.1.3. DIVERSITY ASSESSMENT

184 To detect and prevent model collapse, we attempt to measure
185 the diversity of the many samples of the model from the
186 same prompt. Due to the lack of utility of traditional NLP
187 metrics on this task, we use the mean Pairwise Jaccard
188 distance of the 21-mers of each sequence. Diversity of a
189 group of rollouts is calculated as follows:

$$191 \quad 192 \quad 193 \quad 194 \quad D = 1 - \frac{1}{\binom{n}{2}} \sum_{i=1}^n \sum_{j=i+1}^n J(S_i, S_j)$$

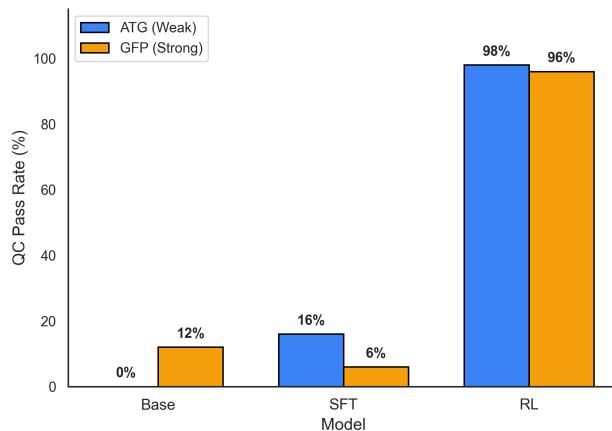
195 where $J(S_i, S_j)$ is the Jaccard similarity between MinHash
196 sketches of sequences i and j , and n is the number of
197 sequences in the group.

198 The diversity metric (pairwise Jaccard distance) serves pri-
199 marily as a model collapse detector rather than a biolog-
200 ical validity measure. The RL model achieves 0.391 diver-
201 sity compared to 0.926 for the base model, indicating that
202 while the RL model concentrates probability mass on higher-
203 quality regions of sequence space, it does not collapse to
204 identical outputs. This is consistent with RL optimization
205 finding conserved “successful motifs” (e.g., proven origins
206 of replication, reliable resistance markers) while still main-
207 taining sequence-level uniqueness.

209 4.1.4. RESULTS

211 Reinforcement learning substantially increases the proba-
212 bility of generating plasmids that pass our bioinformatics
213 quality control (QC) pipeline, while supervised fine-tuning
214 provides modest improvements. Figure 2 shows pass rates
215 by prompt type, and Table 1 summarizes novelty and
216 diversity metrics across models.

217 When prompted with the weak ATG prompt, the base model
218 never produces a valid plasmid (0%), whereas SFT increases



181
182 **Figure 2.** Summary of QC outcomes by prompt. Base model is
183 only able to generate functional plasmids with a strong prompt.
184 SFT allows the model to overcome this limitation on occasion but
185 still often fails QC. Adding RL to the training process improves
186 pass rate dramatically.

187 the pass rate to 16%, and RL further increases it to 96%. A
188 similar trend holds with the stronger GFP-cassette prompt:
189 the base model achieves 12%, SFT drops to 4%, but the
190 RL-optimized model reaches 76%.

191 Aggregated across strong and weak prompts, the overall
192 QC pass rate rises from 6% with the base model to 11%
193 with the SFT model and finally to 97% with the RL model,
194 representing more than an order-of-magnitude improvement
195 in validity relative to the pretrained baseline.

196 Importantly, this increase does not come from the RL mod-
197 els repeating previously known, high scoring sequences.
198 Among passing sequences, the proportion classified as novel
199 remains substantial for all models (Figure 3). Using our
200 novelty thresholds, the RL model produces 88% novel plasmids
201 among its QC-passing samples, compared to 95.5% for the
202 base model. The SFT model generated 100% novel se-
203 quences. When normalized over all 100 rollouts per model,
204 the RL model produces 85.4% sequences that are both QC-
205 valid and novel, compared to 11% for SFT and 6% for the
206 base model.

207 Taken together, these results show that RL post-training not
208 only dramatically improves biological plausibility but also
209 preserves meaningful novelty relative to published plasmids.
210 The model learns to satisfy constraints without collapsing to
211 memorized or trivial constructs, suggesting that sequence-
212 level RL can push generation toward realistic design regions
213 while still exploring new areas of plasmid space.

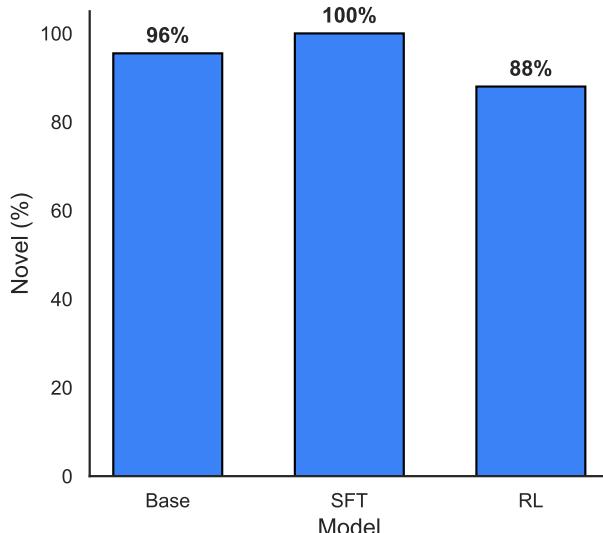


Figure 3. Summary of plasmid novelty as measured by comparison to NCBI database. SFT improves novelty of generated sequences from both base model and RL post-trained model.

4.2. Distribution Comparison

We compute several statistics known to be relevant to the performance of DNA from the raw sequences of both the generated plasmids and a small subset of real plasmids used for protein expression, genome editing, and other applications. See full details of plasmids used in the appendix. We calculate the following summary statistics: sequence length distribution, GC content, longest open reading frame (ORF; calculated two ways), Jensen-Shannon divergence of the codon distribution, and Gibbs free energy (as calculated by ViennaRNA (Lorenz et al., 2011)).

Figure 4 shows that the RL post-trained model's samples much more closely match the distributions of the real plasmids than the pretrained and supervised models, even when the metric is not directly encoded by the reward function. Kolmogorov-Smirnov statistics comparing model-generated distributions to real plasmids demonstrate this quantitatively: for GC content, RL achieves a KS statistic of 0.51 (closest to real at 0.52) compared to base (0.48) and SFT (0.50); for codon usage measured by Jensen-Shannon divergence, RL achieves 0.06, identical to real plasmids and substantially better than base (0.10) and SFT (0.09). See Appendix for complete distribution comparison metrics.

Sequence length is directly encoded by the reward function, with optimal reward determined by a parameter sweep for training stability. GC content is not directly encoded, but regions selected for by the reward function likely have GC content distributed similarly to real plasmids. Despite gener-

Table 2. Average log-probability on held-out continuation task. Higher values indicate better next-token prediction. RL shows unexpected improvement despite not being optimized for this task.

MODEL	MEAN LOG-PROB	STD DEV
BASE	-12.449	6.144
RL	-11.148	2.977

ally making up a smaller percentage of the whole sequence, GC content is partially encoded by the reward function, as regions with higher GC content are more likely to be rewarded.

ORF length and codon distribution are not factored into the reward function directly. We calculate ORF length two ways: (1) maximum length of codons that are not stop codons on a single strand, and (2) longest stretch of non-stop codons after the presence of a start codon on either strand. These methods are disjoint from the method used to account for ORF in the reward function, which uses Prodigal (Hyatt et al., 2010) to predict and reward correctly placed ORFs. The ORF length, measured by either method, converges closely to the distribution of the real plasmids.

The same pattern holds for codon distribution and Gibbs free energy. While not accounted for in the reward function, the RL model learns to generate tokens with a much more similar codon distribution to real plasmids than the two models that have seen the correct distribution in the training data. The similarity in the distribution of free energy measurements is particularly remarkable given that not only is free energy not optimized for directly, but no structural components are factored in at all excluding the weakly correlated repeat penalty originally included to solve recombination issues.

4.3. Held-Out Continuation

To evaluate how RL training affects nucleotide-level predictive performance, we measure how well each model can predict future bases given a real plasmid prefix. For each sequence, we provide the first 400 nucleotides as a prompt, have each model produce the next 100 nucleotides, and then compute the average log-probability of the true next 100 bases under each model. This allows us to compare the Base and RL models against real plasmids in terms of next-token prediction accuracy.

The RL model shows improved log-probability compared to the base model (Figure 5), with the standard deviation shrinking substantially from 6.144 to 2.977. This improvement is unexpected, as RL generally makes language models worse at next-token prediction tasks, a phenomenon known as the "alignment tax" (Lin et al., 2023).

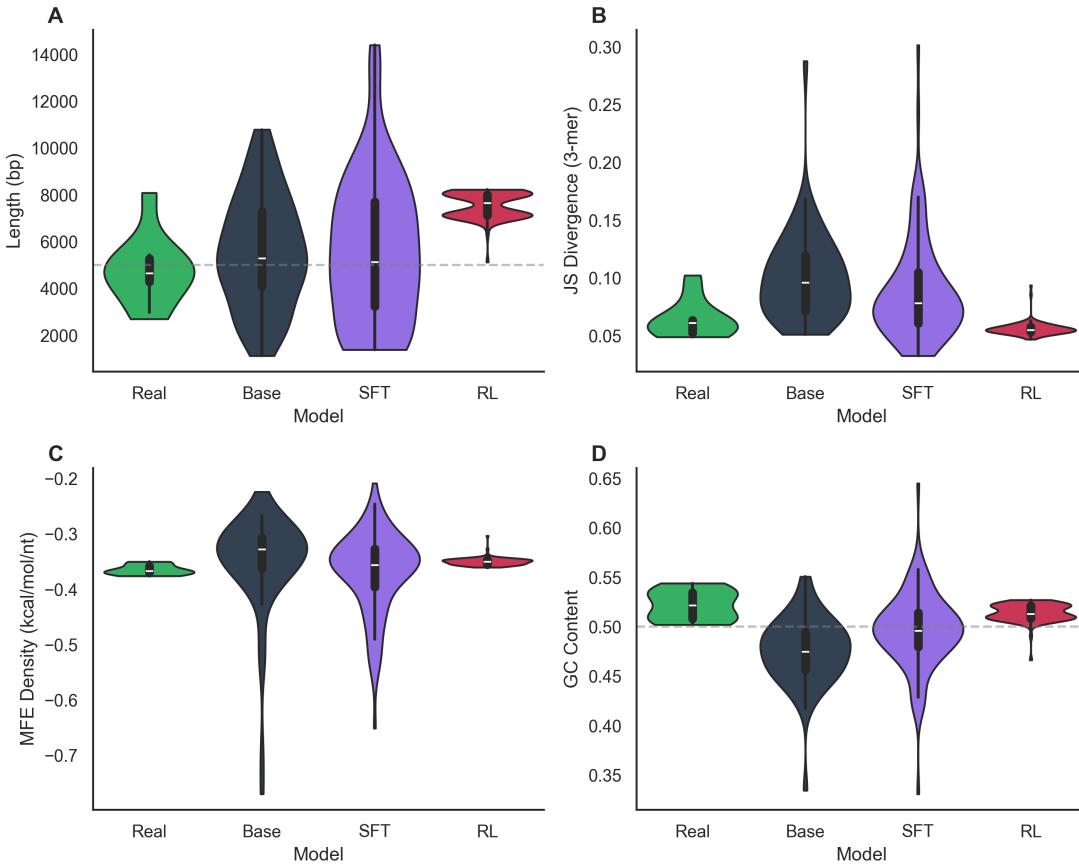


Figure 4. Distribution comparison across key biophysical metrics. The RL post-trained model (red) closely matches real plasmid distributions (green) across sequence length, GC content, ORF length, codon usage (Jensen-Shannon divergence), and thermodynamic stability (Gibbs free energy), while the base model (blue) and SFT model (purple) show substantial deviations. Notably, RL optimization produces realistic distributions even for metrics not explicitly encoded in the reward function.

4.4. Coding Sequence Surprisal Analysis

The reward function evaluates CDS regions using Prodigal rather than bioinformatics token-level approaches, reducing the risk of token-level information leakage. Because Prodigal predicts coding regions based only on generic statistical patterns and not on specific plasmid sequences, the reward signal is biologically grounded but distribution-agnostic. Interestingly, the RL-trained model achieves lower surprisal on real plasmids than the pretrained model (Figure 6), suggesting that the Prodigal-based reward sharpens the model's understanding of natural plasmid structure.

5. Discussion

5.1. Summary: Emergent Biological Realism in RL-Trained Models

Our key finding is that reinforcement learning post-training induces emergent biological realism in DNA language models. Beyond the dramatic improvement in quality control

pass rates (97% vs. 6%), the RL model exhibits properties not explicitly encoded in the reward function: realistic thermodynamic stability (Gibbs free energy distributions matching natural plasmids), natural codon usage patterns (Jensen-Shannon divergence from real plasmids significantly lower than base model), and appropriate ORF length distributions. The emergence of biologically realistic properties, reflecting those of real plasmids, mirrors evolutionary processes that produce complex, correlated traits as byproducts of selecting for primary fitness criteria.

Furthermore, the RL model shows unexpected improvement on next-token prediction tasks, reversing the typical “alignment tax” observed in language models. This suggests the model is not simply memorizing valid solutions but rather learning general principles of plasmid structure. The steering vector hypothesis offers a potential explanation: RL may be moving the model's latent space toward regions that are not only valid by our reward criteria but also broadly biologically coherent, explaining why unoptimized properties improve alongside explicitly rewarded features.

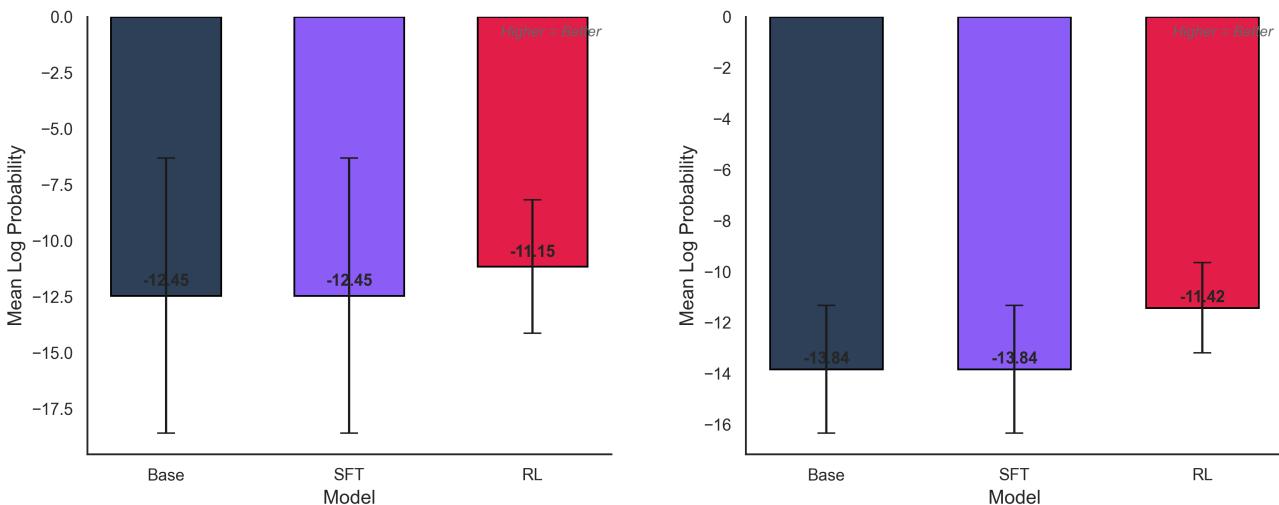


Figure 5. Held-out continuation performance across models. The RL model shows improved log-probability on real plasmid sequences, demonstrating better alignment with natural plasmid structure despite not being explicitly optimized for next-token prediction.

5.2. Implications for Computational Biology and Genomics

This work demonstrates that modern post-training techniques from natural language processing can transfer to DNA language models with similar success. Just as instruction tuning and RLHF unlocked the practical utility of large language models, RL post-training may represent a key step toward practical DNA generative models. The success of our approach suggests that appropriate reward function design, even without perfect biological accuracy (e.g., using lightweight bioinformatics rather than wet-lab validation), can guide models toward broadly realistic sequence spaces.

The parallel between emergent capabilities in LLMs and DNA models has broader implications: it suggests that genomic foundation models may benefit from similar training paradigms that have proven successful in other domains. This opens avenues for applying other techniques from the NLP toolkit (e.g., chain-of-thought prompting, mixture of experts, parameter-efficient fine-tuning) to genomic tasks. Moreover, the emergent properties we observe suggest that DNA language models may be capturing fundamental biological constraints implicit in natural sequence data, much as LLMs capture linguistic structure.

5.3. Limitations and Critiques

Bioinformatics-only evaluation: Our training and evaluation process relies almost entirely on bioinformatics, which

Figure 6. Coding sequence surprisal analysis. The RL model achieves lower surprisal on real plasmid coding sequences compared to the base and SFT models, indicating better capture of natural coding patterns despite using only Prodigal-based structural rewards.

depends on libraries of known regions. These libraries are naturally incomplete. Therefore, if our model generates a novel functional sequence (e.g., a never-before-seen ORI), it will not receive reward, sharply limiting how creative the model can be. This contrasts with other domains where RL has succeeded: natural language processing uses preference models that can evaluate novel outputs, and protein design uses biophysical simulators. Without wet-lab validation, we cannot definitively claim our sequences are experimentally viable, though our pipeline has been validated as a proxy for synthesis success.

Diversity trade-offs: The post-trained model shows decreased sequence diversity (0.391 vs. 0.926 for base model), a known effect of RL optimization. While the diversity metric primarily serves as a collapse detector rather than a biological validity measure, there is a meaningful reduction in functional diversity. For example, when sampling 500 sequences from each model, the base model uses 10 unique ORIs while the RL model uses only 7. The model converges toward conserved, reliable solutions at the expense of exploring less common but potentially viable alternatives. This is consistent with RL finding “successful motifs” but may limit the model’s utility for applications requiring high functional diversity.

Assumptions about reward function design: Our reward function design assumes that structural annotations (promoters, ORIs, etc.) and simple composition metrics can capture essential aspects of plasmid validity. While emergent prop-

385 erties suggest this assumption is partially validated, we may
386 be missing important biological constraints not easily en-
387 coded in lightweight bioinformatics. The simplification is
388 pragmatic for efficient training but may become limiting as
389 we scale to more complex genomic systems.
390

391 **5.4. Future Work**

392 This work further supports the ongoing hypothesis that mod-
393 ern language modeling techniques can be applied to DNA
394 language models in a similar fashion to how they have been
395 applied to natural language in the past. The real utility of
396 natural language models came from instruction tuning them
397 to respond to questions and follow directions. This work
398 shows that post-training techniques can be applied to DNA
399 language models to achieve similar results.
400

401 We hope to build out a dataset designed for conditional
402 generation where the user can prompt the model with the
403 specifics of the plasmid they want, and the model will de-
404 velop it from there. This would enable practical use cases
405 such as “design a plasmid for expressing protein X in *E. coli*
406 with high copy number” or “create a mammalian expression
407 vector with constitutive GFP expression.”
408

409 In addition to being far more practical than unconditional
410 generation, the conditioning will promote more diversity of
411 samples and make evaluation of the model’s performance
412 on the task of plasmid design much more understandable.
413

414 **6. Conclusion**

415 We demonstrate that reinforcement learning can mimic dis-
416 tributions of real plasmids across several key features, even
417 when not explicitly optimized for these features. Our model
418 generates structurally valid plasmids at a 97% rate (com-
419 pared to 6% for the base model), but more significantly,
420 exhibits biological realism not directly encoded in the re-
421 ward function: realistic thermodynamic stability, natural
422 codon usage patterns, and appropriate ORF length distri-
423 butions. These parallels to natural plasmids emerge from
424 optimizing only functional annotations, length constraints,
425 and repeat penalties.
426

427 This finding suggests that RL post-training operates simi-
428 larly across modalities by steering models toward coherent,
429 naturalistic regions of their respective sequence spaces. Just
430 as RL unlocked reasoning and generalization in language
431 models beyond their pretraining objectives, it guides DNA
432 models toward biologically realistic sequences beyond their
433 explicit rewards. The unexpected improvement in next-
434 token prediction further supports this interpretation: the
435 model learns general principles of plasmid structure rather
436 than memorizing specific solutions.
437

438 These findings open several directions for future work. First,

extending to conditional generation (e.g., “express protein X in *E. coli* with high copy number”) would enable practical applications and experimental validation of model-generated sequences. Second, applying RL post-training to other genomic models could test whether emergent biological realism generalizes across applications. Finally, investigating what reward function properties induce such emergence could provide principles for designing effective biological reward functions, potentially accelerating the broader application of RL to computational biology.

Impact Statement

This paper presents work whose goal is to advance the field of computational biology and machine learning for biological sequence design. The ability to generate novel, valid plasmid sequences could accelerate research in synthetic biology, biomanufacturing, and therapeutic development. While the immediate applications are primarily beneficial for scientific research, we acknowledge the dual-use nature of synthetic biology tools. The methods described here are limited to generating plasmid sequences based on patterns in existing databases and do not enable the design of harmful organisms without substantial additional effort and expertise. We believe the benefits to research efficiency and accessibility outweigh the potential risks, particularly given existing biosafety regulations and oversight mechanisms in synthetic biology research.

Code Availability

Code for model training and evaluation will be released upon publication. Training data is derived from PlasmidScope and Addgene databases, available under their respective licenses.

Acknowledgements

We thank the PlasmidGPT team for providing the base model and the computational biology community for making plasmid sequence databases publicly available. This work was supported by Pillar VC’s Encode Fellowship, which is backed by Pillar VC and powered by ARIA and DSIT.

440 A. Training Configuration Details

441 A.1. Supervised Fine-Tuning Hyperparameters

- 442 • **Batch size:** 1
- 443 • **Learning rate:** 5×10^{-5} with 500 warmup steps
- 444 • **Optimizer:** AdamW
- 445 • **Epochs:** 3
- 446 • **Gradient accumulation steps:** 8
- 447 • **Hardware:** [single/multi] NVIDIA L4 GPU(s)

448 A.2. Reinforcement Learning (GRPO)

449 Hyperparameters

- 450 • **Rollout batch size:** 50
- 451 • **Policy learning rate:** 3.55×10^{-5} (found via hyperparameter sweep)
- 452 • **GRPO group size:** 16
- 453 • **Training steps:** Varies between 1000-2500 (scheduled for 2500 but rarely reached due to early convergence)
- 454 • **Convergence criteria:** Reward plateau or dip indicating collapse
- 455 • **Prompt types:** Random 4–25bp seeds (excluding ATG) and structured cassette seeds
- 456 • **Total training time:** ~10–20 hours on NVIDIA L4 GPU

457 A.3. Reward Function Configuration

- 458 • **Origin of replication (exactly 1):** Weight = 1.0
- 459 • **Selectable markers (≥ 1):** Weight = 1.5
- 460 • **Promoter→CDS→terminator bonus:** Weight = 1.5
- 461 • **Length prior (2–15kb):** Weight = 0.6, linear with maximum reward at 5kb and minimum at 15kb, zero reward beyond 15kb
- 462 • **Repeat penalty (>50bp):** -0.1 from total reward for each repeat > 50bp

463 A.4. Distribution Comparison Metrics

464 Table 3 presents Kolmogorov-Smirnov statistics comparing
465 the distributions of generated plasmids to real plasmids
466 across key biophysical metrics. Lower values indicate closer
467 match to the real plasmid distribution.

468 **Table 3.** Kolmogorov-Smirnov statistics for distribution comparison
469 against real plasmids. RL model achieves closest match for
470 GC content and codon usage (JS divergence).

METRIC	REAL	BASE	SFT	RL
LENGTH (BP)	4737	5548	5647	7551
GC CONTENT	0.52	0.48	0.50	0.51
JS DIVERGENCE	0.06	0.10	0.09	0.06
MFE DENSITY	-0.36	-0.35	-0.37	-0.35

471 A.5. Reference Plasmids for Distribution Comparison

472 The following common laboratory plasmids were used for
473 distribution comparison and benchmarking analyses in Section 4.2: pUC19 (Yanisch-Perron et al., 1985), pBluescript
474 (Short et al., 1988), pBR322 (Bolivar et al., 1977), pACYC184 (Chang & Cohen, 1978), pBAD24 (Guzman et al.,
475 1994), pEGFP (Clontech Laboratories, 1999), pGEX-4T-1 (GE Healthcare, 2000), pET-28a (Novagen, 2005), pcDNA3
476 (Kaufman et al., 1991), and px330 (Cong et al., 2013). These
477 plasmids represent widely-used vectors spanning diverse
478 applications and size ranges (2.7–8.1 kb), providing a representative
479 benchmark for evaluating the biological realism of generated sequences.

References

- 480 Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and
481 Lipman, D. J. Basic local alignment search tool. *Journal
482 of Molecular Biology*, 215(3):403–410, 1990.
- 483 Berti, L., Giorgi, F., and Kasneci, G. Emergent abilities in
484 large language models: A survey, 2025. URL <https://arxiv.org/abs/2503.05788>.
- 485 Bolivar, F., Rodriguez, R. L., Greene, P. J., Betlach, M. C.,
486 Heyneker, H. L., Boyer, H. W., Crosa, J. H., and Falkow,
487 S. Construction and characterization of new cloning
488 vehicles. II. A multipurpose cloning system. *Gene*, 2(2):
489 95–113, 1977.
- 490 Brophy, J. A. and Voigt, C. A. Plasmid design for tunable
491 gene expression in bacteria. *Methods in Enzymology*, 497:
492 371–388, 2011.
- 493 Chang, A. C. and Cohen, S. N. Construction and characteriza-
494 tion of new cloning vehicles. V. Mobility and coding
495 functions of pbr322 and pbr325 'derived' plasmids. *Jour-
496 nal of Bacteriology*, 134(3):1141–1156, 1978.
- 497 Clontech Laboratories. pEGFP vector information. Product
498 Manual, 1999. Catalog #6077-1.
- 499 Cong, L., Ran, F. A., Cox, D., Lin, S., Barretto, R., Habib,
500 N., Hsu, P. D., Wu, X., Jiang, W., Marraffini, L. A.,

- 495 et al. Multiplex genome engineering using CRISPR/Cas
496 systems. *Science*, 339(6121):819–823, 2013.
- 497 Cunningham, A. G., Dekker, L., Shcherbakova, A., and
498 Barnes, C. P. Generative design and construction
499 of functional plasmids with a dna language model.
500 *bioRxiv*, 2025. doi: 10.64898/2025.12.06.692736.
501 URL <https://www.biorxiv.org/content/early/2025/12/07/2025.12.06.692736>.
- 502 Deng, Y., Maurais, H. E., Etheridge, K., and Sarpeshkar,
503 R. Gene syntaxes modulate gene expression and
504 circuit behavior on plasmids. *Journal of Biological
505 Engineering*, 19(1):25, 2025. doi: 10.1186/s13036-025-00493-0. URL <https://doi.org/10.1186/s13036-025-00493-0>.
- 506 Durbin, R., Eddy, S. R., Krogh, A., and Mitchison, G. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press,
507 1998. ISBN 9780521629713.
- 508 Fung, V., Tiwade, P. B., and Fenton, O. S. Clonefast:
509 A simple plasmid design and construction guide
510 for labs venturing into synthetic biology. *STAR
Protocols*, 6(3):104025, 2025. ISSN 2666-1667.
511 doi: <https://doi.org/10.1016/j.xpro.2025.104025>.
512 URL <https://www.sciencedirect.com/science/article/pii/S2666166725004319>.
- 513 Galata, V., Fehlmann, T., Backes, C., and Keller, A. Plsdb:
514 a resource of complete bacterial plasmids. *Nucleic Acids Research*, 47(D1):D195–D202, 10 2018. ISSN 0305-
515 1048. doi: 10.1093/nar/gky1050. URL <https://doi.org/10.1093/nar/gky1050>.
- 516 GE Healthcare. pGEX vectors. Product Manual, 2000. GST
517 Gene Fusion System.
- 518 Guzman, L.-M., Belin, D., Carson, M. J., and Beckwith, J.
519 Making transcriptional silencers work in Escherichia coli.
520 *Molecular Microbiology*, 13(4):655–662, 1994.
- 521 Hyatt, D., Chen, G.-L., LoCascio, P. F., Land, M. L.,
522 Larimer, F. W., and Hauser, L. J. Prodigal: prokaryotic
523 gene recognition and translation initiation site identifi-
524 cation. *BMC Bioinformatics*, 11(1):1–11, 2010. doi:
525 10.1186/1471-2105-11-119.
- 526 Ji, Y., Zhou, Z., Liu, H., and Davuluri, R. V. DNABERT:
527 pre-trained bidirectional encoder representations from
528 transformers model for DNA-language in genome. *Bioin-
529 formatics*, 37(15):2112–2120, 2021.
- 530 Kaufman, R. J., Davies, M. V., Pathak, V. K., and Hershey,
531 J. W. A mammalian expression vector for the expression
532 of complete open reading frames. *Nucleic Acids Research*,
533 19(16):4485–4490, 1991.
- 534 Kutzler, M. A. and Weiner, D. B. Plasmid DNA vaccines:
535 an overview. *Vaccine*, 26:S59–S75, 2008.
- 536 Lederberg, J. Cell genetics and hereditary symbiosis. *Phys-
537 iological Reviews*, 32(4):403–430, 1952. Seminal paper
538 defining the term “plasmid”.
- 539 Lin, Y., Dong, H., Wang, H., Zhang, J., Liu, J., Pi, R.,
540 Pan, R., Zhang, H., Hu, W., Zhao, H., et al. Mit-
541 igating the alignment tax of RLHF. *arXiv preprint arXiv:2309.06256*, 2023.
- 542 Lorenz, R., Bernhart, S. H., Höner zu Siederdissen, C., Tafer,
543 H., Flamm, C., Stadler, P. F., and Hofacker, I. L. Vien-
544 narna package 2.0. *Algorithms for Molecular Biology*, 6(1):26, 2011.
- 545 Martinson, J. N. V. et al. Generating functional plasmid
546 origins with OriGen. *Nucleic Acids Research*, 53(22):
547 gkaf1198, 2025. doi: 10.1093/nar/gkaf1198.
- 548 Meng, F. and Ellis, T. Challenges in rational design of
549 synthetic promoters. *New Biotechnology*, 32(3):337–344,
550 2015.
- 551 Nguyen, E., Poli, M., Faizi, M., Thomas, A., Birch-Sykes,
552 C., Wornow, M., Patel, A., Rabideau, C., Massaroli, S.,
553 Bengio, Y., et al. HyenaDNA: Long-range genomic se-
554 quence modeling at single nucleotide resolution. *arXiv
555 preprint arXiv:2306.15794*, 2023.
- 556 Nguyen, E., Poli, M., Durrant, M. G., Kang, B., Katrekar,
557 D., Li, D. B., Bartie, A., Thomas, A. W., King, S. H.,
558 Bifulco, G., et al. Sequence modeling and design from
559 molecular to genome scale with Evo. *bioRxiv*, 2024. doi:
560 10.1101/2024.02.27.582234.
- 561 Novagen. pET system manual. Product Manual, 11th Edi-
562 tion, 2005. TB055.
- 563 Oliveira, P. H., Prather, K. L., Prazeres, D. M., and Monteiro,
564 G. A. Structural instability of plasmid biopharmaceuti-
565 cals: challenges and implications. *Trends in Biotechnol-
566 ogy*, 27(9):503–511, 2009.
- 567 Prather, K. J., Sagar, S., Murphy, J., and Chartrain, M. In-
568 dustrial scale production of plasmid DNA for vaccine and
569 gene therapy: plasmid design, production, and purifica-
570 tion. *Enzyme and Microbial Technology*, 33(7):865–883,
571 2003.
- 572 Shao, L. et al. PlasmidGPT: a generative framework for
573 plasmid design and annotation. *bioRxiv*, 2024a. doi:
574 10.1101/2024.09.30.615762. Preprint.
- 575 Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Zhang, M.,
576 Li, Y. K., Wu, Y., and Guo, D. DeepSeekMath: Pushing
577 the limits of mathematical reasoning in open language
578 models. *arXiv preprint arXiv:2402.03300*, 2024b.

550 Short, J., Fernandez, J., Sorge, J., and Huse, W. Bluescript:
551 a high copy number Escherichia coli plasmid vector. *Nu-*
552 *ucleic Acids Research*, 16(15):7583–7600, 1988.

553
554 Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B.,
555 Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D.,
556 Metzler, D., Chi, E. H., Hashimoto, T., Vinyals, O.,
557 Liang, P., Dean, J., and Fedus, W. Emergent abili-
558 ties of large language models, 2022. URL <https://arxiv.org/abs/2206.07682>.

560 Wu, G., Yan, Q., Jones, J. A., Tang, Y. J., Fong, S. S.,
561 and Koffas, M. A. Metabolic burden: cornerstones in
562 synthetic biology and metabolic engineering applications.
563 *Trends in Biotechnology*, 34(8):652–664, 2016.

564 Yanisch-Perron, C., Vieira, J., and Messing, J. New M13
565 vectors for cloning. *Gene*, 33(1):103–119, 1985.

567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604