TODNA: PREFERENCE-PROGRAMMABLE TEST-TIME OPTIMIZATION FOR REGULATORY DNA DESIGN

Anonymous authors
Paper under double-blind review

ABSTRACT

Restricting gene expression to target cell types is a central goal in regulatory DNA design for gene therapy and synthetic biology. While genomic foundation models capture rich sequence regularities, their pre-trained priors do not explicitly enforce the differential activity across cellular contexts that defines cellular selectivity. Although reward-based post-training can encode such context-specific preferences into model parameters, adapting to new design objectives often necessitates costly re-training. To address this limitation, we introduce ToDNA, a novel test-time optimization framework that cleanly decouples a reusable regulatory sequence prior from cell-specific functional objectives. Specifically, ToDNA employs target-agnostic low-rank adaptation (LoRA) to specialize a base model (Evo 2) to specific regulatory-element classes, while leveraging a frozen multi-cell predictor to guide nucleotide-logit updates, regularized against a fixed local reference. This modular separation allows target-to-off-target preferences to be dynamically re-configured via reward functions without retraining the generative backbone. Evaluated across four enhancer and promoter design tasks, ToDNA demonstrates that candidate optimization substantially boosts enhancer design specificity, whereas promoter responsiveness exhibits more modest gains. Sequence-level analyses further reveal that stronger motif agreement does not necessarily align with higher predicted cell specificity. Crucially, we show that alternative reward configurations can redirect cell-activity preferences purely at inference time, though their effectiveness varies depending on the interplay between activity and sequence-level objectives. These findings establish the viability of reusable, inference-time control for regulatory DNA design, while highlighting objective formulation and predictor validity as key determinants of true cellular selectivity.

1 INTRODUCTION

Promoters and enhancers help determine where and how strongly genes are expressed. Their design offers a way to direct therapeutic expression and synthetic gene circuits toward intended cell types. Recent studies have designed and experimentally validated regulatory elements with cell-type preferences (Taskiran et al., 2024; Gosai et al., 2024). For such applications, a sequence with strong target-cell activity may still be unsuitable if it also drives expression in other cells.

Cell specificity depends on how regulatory motifs act together in a given cellular context (de Boer et al., 2020; Avsec et al., 2021). Massively parallel assays have expanded the available sequence–function data (de Boer et al., 2020; Agarwal et al., 2025), yet measurements cover only a small fraction of the 4^L sequences of length L (Vaishnav et al., 2022). Activity predictors make this space searchable, but a score for a complete sequence does not directly identify which nucleotide substitutions will improve selectivity. Search must also balance target activation against off-target expression, while accounting for uncertainty in predictions for sequences beyond the measured data.

Recent advances in generative modeling have unlocked new capabilities in decoding and designing functional regulatory sequences from large-scale genomic datasets. GAN-based approaches generate promoters and sequences with specified expression properties (Wang et al., 2020; Zrimec et al., 2022). Deep exploration networks jointly optimize predicted fitness and diversity (Linder et al., 2020), while diffusion models learn distributions over human promoter sequences (Avdeyev et al., 2023). These methods provide useful candidates, although their sequence priors are learned within

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.

particular regulatory settings. Genomic foundation models offer broader pretraining that can be reused for regulatory design (Nguyen et al., 2023; Brixi et al., 2026). For example, regLM uses activity labels to condition generation, TACO introduces functional rewards, and Ctrl-DNA uses constrained reinforcement learning (Lal et al., 2024; Yang et al., 2025; Chen et al., 2025). Such adaptation allows functional preferences to guide repeated sampling from the trained model. Changing an objective beyond its learned conditions, however, may require further training.

We study whether an adapted genomic model can support cell-specific design by optimizing its candidate sequences at inference. Predictor-guided sequence search and reward-guided diffusion refinement have established precedents for this approach (Linder & Seelig, 2021; Gosai et al., 2024; Uehara et al., 2025). The question here is how to reuse a genomic prior when the desired balance of target and off-target activities changes. A sequence model supplies candidate regulatory elements, whereas an activity predictor provides the cell-dependent information needed to distinguish these objectives. Keeping their roles separate allows the objective to change without retraining the generator.

We introduce ToDNA, which combines an adapted Evo 2 generator with regularized optimization of individual candidates (Figure 1). Low-rank adaptation specializes Evo 2 to enhancers or promoters without using a target-cell reward. The adapted model supplies starting sequences and a fixed local reference for their nucleotide probabilities. Straight-through updates then adjust candidate logits using a frozen multi-cell activity predictor, with a local KL penalty regularizing departure from that reference. Within the cell contexts supported by the predictor, different target preferences can be specified by replacing the reward function.

Our main contributions are as follows:

- We formulate cell-specific regulatory DNA design as candidate-level test-time optimization. A frozen predictor supplies nucleotide-level gradients from sparse sequence rewards, while a local genomic reference concentrates search in the exponential sequence space under complex target–off-target constraints.
- We develop ToDNA, an Evo 2 framework separating element-class LoRA adaptation from cell-specific inference. KL-regularized logit updates trade off the adapted regulatory prior and predictor-defined cell preference, allowing reward changes without model retraining.
- Across the four displayed tasks, ToDNA has higher observed enhancer-specificity means than PPO and GRPO and attains the largest observed motif-Pearson mean among the evaluated methods on all three enhancers.

2 Related Work

DNA Sequence Design Optimization. Predictor-guided enhancer design has produced cell-selective sequences with experimentally validated activity (Taskiran et al., 2024). CODA compares evolutionary search, simulated annealing, and gradient-based updates to optimize target activity relative to the strongest off-target (Gosai et al., 2024). ExpressionGAN instead optimizes generator inputs for specified expression levels (Zrimec et al., 2022). Deep exploration networks train generators for predicted fitness and diversity (Linder et al., 2020), whereas DRAKES fine-tunes discrete diffusion models with KL-regularized rewards (Wang et al., 2025). Conservative model-based optimization addresses promoter design with limited functional data (Reddy et al., 2024). These approaches optimize individual sequences, generator inputs, or model parameters, providing distinct ways to incorporate functional objectives into DNA design.

Test-time Optimization. Test-time optimization reuses pretrained models while adjusting generation to meet task-specific objectives. ReNO optimizes initial noise using preference rewards for image generation (Eyring et al., 2024), while DNO optimizes diffusion noise for motion-specific objectives (Karunratanakul et al., 2024). Kim et al. (2025) align diffusion sampling with rewards through sequential Monte Carlo and tempering. In language reasoning, Snell et al. (2025) study verifier-guided search and adapt inference compute allocation to prompt difficulty. Related DNA methods use straight-through updates (Linder & Seelig, 2021), diffusion refinement (Uehara et al., 2025), and, in recent workshop work, population sampling (Sarkar et al., 2026). ToDNA optimizes DNA candidate logits under regularization toward a fixed local reference from the adapted genomic model. Both

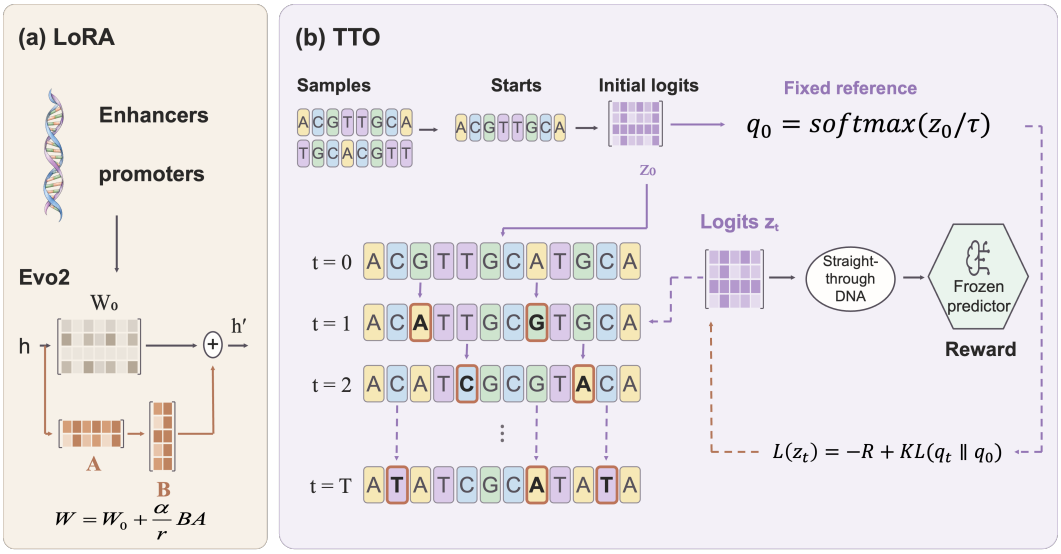

Figure 1: ToDNA framework. (a) Regulatory-DNA language modeling trains low-rank adapters with a frozen Evo 2 backbone. (b) Adapted Evo 2 samples candidates and initializes logits for Design-selected starts. A fixed reference supplies KL regularization; straight-through DNA connects each optimization step to the frozen Design oracle. Outlined bases illustrate changes, including possible reversions, rather than measured trajectories or irreversible commitments. In panel (b), t indexes optimization steps; R is the normalized mean contrast, and KL denotes the position-averaged penalty with unit weight used in the main comparison. The DNA illustration is schematic and does not depict experimental data.

the generator and activity predictor remain frozen while the reward specifies the target–off-target preference.

3 Method

3.1 Problem formulation

Let $x = (x_1, \ldots, x_L) \in \mathcal{V}^L$ be a fixed-length regulatory sequence, where $\mathcal{V} = \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$. For element class e, the target-agnostic generator p_{θ,η_e} supplies candidates, with θ denoting the Evo 2 backbone and η_e its class-specific adapter. A frozen, differentiable predictor f_ψ returns an activity estimate $a_c(x)$ for each cell $c \in \mathcal{C}$, where $|\mathcal{C}| \geq 2$.

To place cell-wise outputs on a common study scale, we use fixed reference values $\ell_c < u_c$:

$$s_c(x) = \frac{a_c(x) - \ell_c}{u_c - \ell_c}, \tag{1}$$

where values outside $[0, 1]$ are retained. The reference ranges are given in Appendix A.1; these scores are not experimentally calibrated activity measurements.

For target cell $t \in \mathcal{C}$, let $\mathcal{O}_t = \mathcal{C} \setminus \{t\}$ denote the off-target cells. We use the mean target–off-target contrast

$$R_t(x) \equiv \Delta R_{\mathrm{mean}}(x; t) = s_t(x) - \frac{1}{|\mathcal{O}_t|} \sum_{o \in \mathcal{O}_t} s_o(x), \tag{2}$$

and define the corresponding ideal design as

$$x_t^\star \in \arg\max_{x \in \mathcal{V}^L} R_t(x). \tag{3}$$

We additionally report the worst-off-target margin

$$\Delta R_{\mathrm{min}}(x; t) = s_t(x) - \max_{o \in \mathcal{O}_t} s_o(x), \tag{4}$$

which is positive only when the target exceeds every evaluated off-target. Thus, ΔR_{mean} measures average contrast, whereas $\Delta R_{\min}$ reveals failure against the strongest off-target. Section 3.3 performs finite-step, reference-regularized search using R_t; this search does not guarantee global optimality, hard off-target constraints, or biological activity.

3.2 DNA Sequence Prior Distribution Learning

Sequence prior initialization. Genomic foundation models learn reusable sequence distributions from genomic corpora without requiring cell-activity labels (Nguyen et al., 2023; Dalla-Torre et al., 2025; Nguyen et al., 2024; Brixi et al., 2026). For a DNA sequence $x = (x_1, \ldots, x_L)$ with $x_i \in \mathcal{V}$, an autoregressive model factorizes this distribution as

$$p_\theta(x) = \prod_{i=1}^{L} p_\theta(x_i \mid x_{<i}). \tag{5}$$

Next-nucleotide pretraining estimates each conditional distribution from the sequence contexts represented in the training corpus. The resulting model supports sampling, sequence scoring, and downstream adaptation, but remains a statistical prior rather than a cell-specific regulatory objective.

We initialize p_θ with the pretrained 7B-parameter Evo 2 checkpoint rather than learning a genomic model from scratch. Evo 2 is a single-nucleotide autoregressive model built on the StripedHyena 2 multi-hybrid architecture. Its OpenGenome2 pretraining corpus spans bacteria, archaea, eukaryotes, and bacteriophages (Brixi et al., 2026). The causal interface maps each context $x_{<i}$ to next-nucleotide logits in the native vocabulary. The four-base local reference used by ToDNA is normalized separately from these native-vocabulary outputs.

This pretrained distribution provides broad genomic context but contains neither a requested target cell nor the target–off-target contrast in Equation 2. Evo 2 was also developed for substantially longer contexts than the 200–250-bp sequences studied here. Our experiments use its conditional distributions at the task sequence length and do not isolate an advantage from its long-context capacity. The remaining mismatch is therefore one of sequence domain: a genome-scale prior need not concentrate probability on short enhancers or promoters.

Adapting the sequence prior. We use low-rank adaptation (LoRA) to shift the pretrained distribution toward the regulatory-element class of interest while preserving a shared backbone (Hu et al., 2022). For a frozen backbone matrix $W_0 \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, the effective class-specific weight at inference is

$$W_e = W_0 + \frac{\alpha}{r} B_e A_e, \qquad A_e \in \mathbb{R}^{r \times d_{\text{in}}}, \quad B_e \in \mathbb{R}^{d_{\text{out}} \times r}. \tag{6}$$

Here $r \ll \min(d_{\text{in}}, d_{\text{out}})$ controls the update rank, and α sets its scale. The backbone weights remain fixed while the adapter parameters η_e collect the (A_e, B_e) pair for each selected matrix. Each adapted matrix introduces $r(d_{\text{in}} + d_{\text{out}})$ trainable parameters, compared with $d_{\text{in}} d_{\text{out}}$ under a full update. Separate enhancer and promoter adapters can therefore specialize the prior while sharing θ.

For a fixed-length regulatory-sequence corpus $\mathcal{D}_e$, with $L_e \geq 2$, we learn η_e by minimizing the next-token mean cross-entropy

$$\mathcal{L}_{\text{adapt}}(\eta_e) = -\mathbb{E}_{x \sim \mathcal{D}_e, \xi} \left[\frac{1}{L_e - 1} \sum_{i=2}^{L_e} \log p^{\xi}_{\theta, \eta_e}(x_i \mid x_{<i}) \right], \qquad \theta \text{ fixed.} \tag{7}$$

Here, ξ denotes adapter-dropout masks. The model conditions on the first nucleotide and predicts positions 2 through L_e in the native output vocabulary. Adapter dropout is disabled at inference, and held-out language-model loss guides selection of separate enhancer and promoter adapters. Neither the requested target cell nor an activity reward enters this objective. Thus, p_{θ,η_e} models an element-class distribution while cell preference remains an inference-time signal.

The adapted prior connects to test-time optimization through two interfaces (Figure 1). Autoregressive sampling from p_{θ,η_e} determines the candidate pool and hence the regions of sequence space available to the search. Teacher-forced prediction at a selected candidate x^0 supplies initial nucleotide logits z^0. Vocabulary alignment and a selected-base margin ensure that hard decoding recovers x^0. The resulting

$q_i^0 = \mathrm{softmax}(z_i^0/\tau)$ is fixed during optimization and defines the local penalty for redistributing nucleotide probabilities. Its construction is specified in Appendix A.4.

Once adaptation is complete, θ and η_e remain fixed for every cell target within an element class. The same adapted distribution can therefore support different cell-specific rewards without further model training. LoRA restricts the parameterization and changes the sequence prior; it does not by itself establish regulatory function or preserve every capability of the pretrained model.

3.3 DNA Sequence Optimization under Cell-Specific Constraints

Section 3.2 yields a regulatory-element prior that is shared across target cells. Cell identity enters only at inference through the reward R_t. To favor cell-selective activity without allowing reward-driven search to move arbitrarily far from the adapted prior, ToDNA balances two soft design pressures: R_t promotes the target relative to the evaluated off-target cells, whereas a start-specific KL term limits local departure from the prior-derived reference. Accordingly, "constraints" here denotes terms in the optimization objective, not hard restrictions defining a feasible set.

Reference-regularized sequence objective. For each starting sequence x^0 selected from a candidate pool sampled from p_{θ,η_e}, the teacher-forced construction in Section 3.2 supplies initial logits z^0 and a fixed local reference $q^0 = (q_1^0, \ldots, q_L^0)$. We optimize nucleotide logits $z \in \mathbb{R}^{L\times 4}$ initialized at z^0, and define the hard decoder $h_i(z) = \arg\max_{b\in\mathcal{V}} z_{i,b}$ using fixed tie-breaking. Because this discrete map is non-differentiable, we introduce, for temperature $\tau > 0$, the relaxation

$$q_i(z) = \mathrm{softmax}(z_i/\tau), \qquad \tilde{q}_i = \mathrm{sg}[\mathrm{onehot}(h_i(z)) - q_i(z)] + q_i(z), \tag{8}$$

where sg is the identity in the forward pass and has zero backward derivative. We write $q(z) = (q_1(z), \ldots, q_L(z))$, $h(z) = (h_1(z), \ldots, h_L(z))$, and $\tilde{q} = (\tilde{q}_1, \ldots, \tilde{q}_L)$. The predictor therefore receives the one-hot representation of $h(z)$ in the forward pass, while differentiation uses the surrogate Jacobian $\partial\tilde{q}_i/\partial z_i = (\mathrm{diag}(q_i(z)) - q_i(z)q_i(z)^\top)/\tau$.

Finite-step optimization and decoding. Starting from z^0, we apply Adam for a finite number of steps to the straight-through gradient of

$$\mathcal{L}_t(z) = -R_t(\tilde{q}) + \lambda_{\mathrm{KL}} \frac{1}{L} \sum_{i=1}^{L} D_{\mathrm{KL}}(q_i(z) \| q_i^0), \tag{9}$$

where $\lambda_{\mathrm{KL}} \geq 0$ controls the strength of reference regularization. The reward term increases the normalized target score relative to the mean off-target score in Equation 2; here $R_t(\tilde{q})$ denotes evaluation through the predictor's one-hot input interface and equals $R_t(h(z))$ in the forward pass. The KL term penalizes positionwise probability shifts away from q^0. Equation 9 uses exact categorical KL; the implementation's probability-floor approximation is specified in Appendix A.4. Because q^0 is a teacher-forced, position-factorized reference, this regularizer is neither the autoregressive likelihood of the optimized sequence nor a certificate of biological validity.

Throughout optimization, the backbone parameters θ, adapter parameters η_e, predictor parameters ψ, and reference q^0 remain fixed; only z is updated. The final logits $\hat{z}_t(x^0)$ yield the sequence $\hat{x}_t = h(\hat{z}_t(x^0))$, with $\hat{x}_{t,i} = \arg\max_{b\in\mathcal{V}} \hat{z}_{t,i,b}$. The realized within-start gain is $G_t(x^0) = R_t(\hat{x}_t) - R_t(x^0)$. Hard decoding guarantees only the prescribed sequence length and nucleotide alphabet. The finite-step search does not certify a global maximizer or impose hard thresholds on target or off-target activity, GC content, homopolymer length, or edit burden; these properties are evaluated separately. Search hyperparameters and budget accounting are provided in Appendix A; the complete candidate-update procedure is summarized in Algorithm 1.

4 Theoretical Analysis

Test-time DNA optimization updates candidate logits while keeping the sequence model, adapter, and predictor fixed. For a fixed start x^0 and reference q^0, the target enters Equation 9 only through R_t. This makes the search target-modular, but does not guarantee optimization success. We analyze three links: objective improvement to reward gain, local KL to nucleotide edits, and mean reward

to strict selectivity. All results are conditional; none proves convergence or global optimality. KL statements use exact categorical probabilities and apply to the implementation only when probability flooring is inactive. Proofs are in Appendix D.

Optimization-space certificate. Let $\Omega(q, q^0) = L^{-1} \sum_i D_{\mathrm{KL}}(q_i \| q_i^0)$ and $J_t(z) = R_t(h(z)) - \lambda_{\mathrm{KL}} \Omega(q(z), q^0)$, the hard-forward value of $-\mathcal{L}_t(z)$. Since $q(z^0) = q^0$ and $h(z^0) = x^0$, any iterate with $J_t(z) \geq J_t(z^0)$ satisfies

$$R_t(h(z)) - R_t(x^0) \geq \lambda_{\mathrm{KL}} \Omega(q(z), q^0). \tag{10}$$

For $\lambda_{\mathrm{KL}} > 0$, reward gain must compensate for departure from the reference. This is an a posteriori certificate, not an optimizer guarantee: the straight-through direction is not the gradient of the piecewise-constant hard objective. Finite-step Adam therefore need not improve J_t monotonically.

Proposition 1 (Prior divergence bounds nucleotide substitutions). Assume each q_i^0 has full support and a unique maximizer x_i^0. Define $\gamma_i = q_{i,x_i^0}^0 - \max_{b \neq x_i^0} q_{i,b}^0 > 0$, $\mathcal{I}(z) = \{i : h_i(z) \neq x_i^0\}$, and $\gamma_{\min} = \min_i \gamma_i$. Then

$$\sum_{i \in \mathcal{I}(z)} \gamma_i^2 \leq 2L\Omega(q(z), q^0), \qquad \frac{d_H(h(z), x^0)}{L} \leq \frac{2\Omega(q(z), q^0)}{\gamma_{\min}^2}. \tag{11}$$

Each substitution therefore incurs at least $\gamma_i^2/2$ in its positionwise categorical KL. Writing $\Delta R_t(z) = R_t(h(z)) - R_t(x^0)$, any iterate with $J_t(z) \geq J_t(z^0)$ and $\lambda_{\mathrm{KL}} > 0$ further satisfies

$$\sum_{i \in \mathcal{I}(z)} \gamma_i^2 \leq \frac{2L\Delta R_t(z)}{\lambda_{\mathrm{KL}}}, \qquad \frac{d_H(h(z), x^0)}{L} \leq \frac{2\Delta R_t(z)}{\lambda_{\mathrm{KL}} \gamma_{\min}^2}. \tag{12}$$

These a posteriori bounds can be weak at nearly tied positions. They neither make λ_{KL} a hard edit constraint nor imply monotonic edit counts across separately optimized runs. Moreover, Ω is a positionwise divergence from the adjusted reference, not a joint-sequence KL or a bound on Evo 2 likelihood.

Proposition 2 (Mean reward does not ensure strict selectivity). Let $\bar{s}_{\mathcal{O}_t}(x)$ be the mean off-target score and define $D_t(x) = \max_{o \in \mathcal{O}_t} s_o(x) - \bar{s}_{\mathcal{O}_t}(x) \geq 0$. Then

$$\Delta R_{\min}(x; t) = R_t(x) - D_t(x). \tag{13}$$

Thus, strict selectivity on the normalized predictor scale requires $R_t(x) > D_t(x)$, not merely $R_t(x) > 0$. For a TTO output, its worst-off-target gain equals $G_t(x^0) - [D_t(\hat{x}_t) - D_t(x^0)]$. Mean-reward improvement can therefore leave one off-target dominant. This is a limitation of the chosen reward, rather than of test-time optimization in general.

5 Experiments

5.1 Experimental setup

Datasets. We use the public human enhancer and promoter datasets described in Chen et al. (2025). These datasets pair regulatory sequences with activity measurements from massively parallel reporter assays (MPRAs). The enhancer dataset contains 200-bp sequences measured in HepG2, K562, and SK-N-SH. The promoter dataset contains 250-bp sequences measured across three promoter cell channels. The displayed promoter target is K562; the remaining measured channels define its off-target context.

Evaluation Metrics. We assess predicted cell preference using normalized Evaluation ΔR_{mean} and $\Delta R_{\min}$ (Equations 2 and 4), which compare target activity with the mean and strongest off-target activity, respectively. Motif Pearson and Spearman quantify linear and rank agreement, respectively, between generated and task-matched natural motif-frequency profiles, while 3-mer Pearson compares their 64-component nucleotide 3-mer count profiles. These sequence-reference metrics do not measure regulatory function. Motif profiles use JASPAR 2024 and FIMO at $p \leq 10^{-4}$ (Rauluseviciute et al., 2024; Grant et al., 2011). Sequence-level observations are aggregated within runs; trajectory summaries are computed separately at each step. We report means and sample SDs across three runs, without significance claims.

Models and Baselines. ToDNA uses Evo 2-7B with separate enhancer and promoter LoRA adapters, each shared across target cells. We compare it with constrained reinforcement learning in Ctrl-DNA and LoRA policy updates using PPO and GRPO (Chen et al., 2025; Schulman et al., 2017; Shao et al., 2024). Four Evo 2 variants examine alternative test-time strategies: LM-Evo uses guided mutation and selection, Infilling resamples short spans, and Evo2-CEM updates categorical proposals through cross-entropy selection. Evo2-CREsted adapts differentiable design (Kempynck et al., 2026) and is not a reproduction of the original pipeline. Each method receives the same budget for evaluating sequences with the activity predictor during optimization and returns the same number of sequences. Candidate generation and scoring outside this stage incur additional costs, so the comparison does not match total computation (Appendix A.5).

Reward Models. Frozen Enformer-based predictors (Avsec et al., 2021) map each sequence to three cell-activity estimates within its regulatory-element family. The Design predictor supplies the normalized target-minus-mean-off-target reward in Equation 2. A separately trained Evaluation predictor assesses generated sequences. The predictors use separate sequence-group partitions but share architecture and pretraining, limiting their independence. Changing the target or reward leaves predictor weights fixed, with activity normalization specified in Appendix A.1.

5.2 ToDNA captures motif distributions under cell-type-specific constraints

The main comparison examines how well cell-specific optimization retains the motif-frequency structure of natural regulatory sequences (Table 1). ToDNA achieves the highest mean motif Pearson correlation among the evaluated methods on all three enhancer tasks: 0.844 ± 0.020 for HepG2, 0.865 ± 0.012 for K562, and 0.759 ± 0.030 for SK-N-SH. It also exceeds PPO and GRPO in predicted enhancer specificity, with ΔR_{mean} values of 0.282, 0.348, and 0.193, respectively. Under the evaluated settings, candidate-level optimization therefore improves both readouts relative to these policy post-training baselines. The results support reusing the adapted generator while adjusting individual candidates, although the comparison does not isolate which optimization component accounts for the advantage.

Comparisons with other inference-time strategies clarify this result. On HepG2, Infilling approaches ToDNA in specificity (0.248 versus 0.282), but has lower motif agreement (0.332 versus 0.844). On K562 enhancers, Infilling attains higher specificity (0.379 versus 0.348), again with lower motif agreement (0.472 versus 0.865). Thus, similar activity contrasts can arise from sequences with substantially different motif profiles. Resemblance to regulatory sequence profiles alone is therefore insufficient to direct a generator toward a chosen cellular context.

ToDNA combines strong enhancer motif agreement with positive predicted cell preference, but does not maximize both measures across the benchmark. Ctrl-DNA leads specificity on HepG2 and K562 enhancers, and LM-Evo leads on SK-N-SH. For K562 promoters, ToDNA retains high motif agreement (0.865 ± 0.004), while specificity remains negative for every evaluated method. This contrast locates the demonstrated benefit in motif-profile fidelity during cell-directed search, rather than uniform superiority in functional control. Motif Pearson summarizes frequency agreement and cannot establish motif syntax or binding activity. The cell-specific objective likewise provides a soft prediction-based preference, not a verified biological constraint.

5.3 The DNA Optimization Trajectories of ToDNA

Endpoint comparisons leave open whether improved scores emerge through candidate refinement or final selection. We therefore follow the same starts across optimization steps, tracking hard-decoded sequences, composition, and the Design reward. These diagnostics use separate family-specific adapters and are interpreted within their own protocol (Appendix B).

Figure 2 connects the discrete sequence changes to score evolution. Hard-decoded sequences are initially unchanged, followed by a rapid increase in the fraction of positions differing from the start. Later, this fraction changes more slowly while enhancer Design scores continue to rise. This temporal separation is consistent with continued refinement after the initial sequence displacement: increasing distance from the start is not required for every subsequent score gain. Because argmax decoding changes only when nucleotide rankings cross, continuous logit updates need not produce a substitution at each step. However, Hamming distance cannot distinguish new edits from replacements

or reversions at previously edited positions. Its slower growth therefore does not establish convergence, and the non-monotonic individual example shows that aggregate trends do not guarantee improvement at every step.

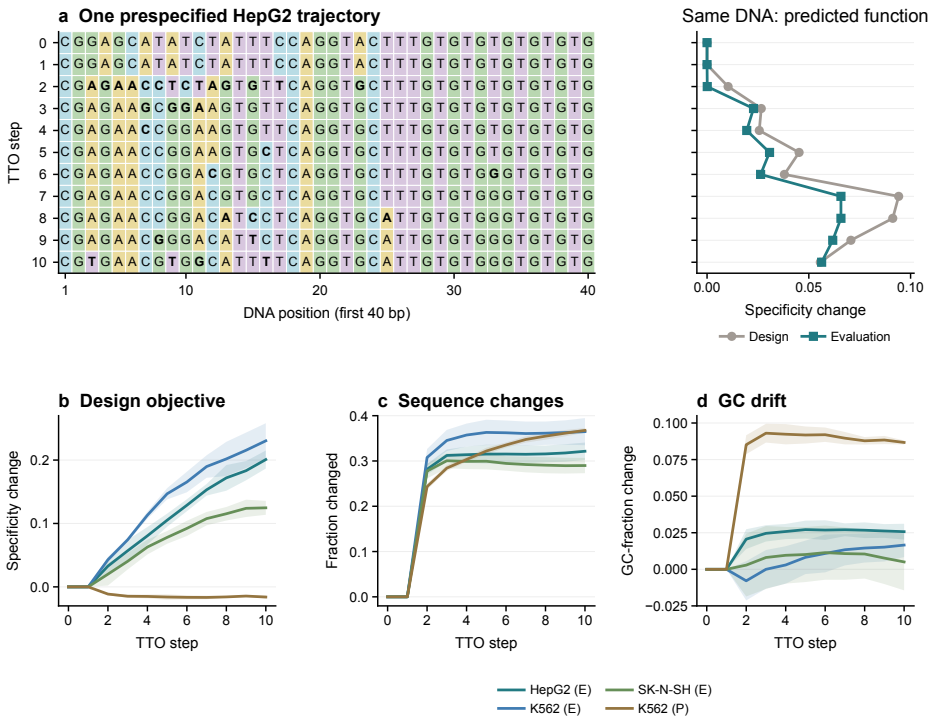

Figure 2: Sequence changes during optimization. (a) A 40-bp segment of a prespecified HepG2 start, with bold bases marking changes from the preceding step; adjacent curves score the full sequence. (b–d) Changes in Design specificity, cumulative edit fraction, and GC fraction. Edit fraction is Hamming distance from the initial sequence, divided by its length. Scores in (a) describe the full sequence and do not isolate the effects of the displayed substitutions. Lines and bands show mean and sample SD across three run means, each averaging 32 starts. E: enhancer; P: promoter.

These observations motivate examining how reference regularization and adapted logits shape candidate refinement in the ablations below.

5.4 Ablation Study

The trajectory analysis motivates separating two roles of the sequence prior: constraining departure during search and specifying the local nucleotide preferences from which search begins. Paired HepG2 ablations remove the reference penalty or replace adapted logits with base-model logits, holding candidate pools, starts, and search budgets fixed (Table 2). This retrospective study followed evaluation-informed development and uses different predictor checkpoints. Its raw MinGap scores are therefore interpreted within the ablation study, rather than compared directly with the normalized benchmark.

Removing regularization increases the number of substitutions by 10.417 ± 0.888, but decreases Evaluation and FullMPRA MinGap by 0.147 ± 0.020 and 0.237 ± 0.042, respectively. Mean 3-mer Pearson also falls from 0.874 to 0.862. Greater freedom to leave the reference thus produces more sequence change without improving specificity under either assessment predictor. These paired outcomes suggest that reference restraint can benefit assessed specificity as well as sequence-profile agreement. They do not identify whether the benefit arises from avoiding predictor exploitation, improving optimization stability, or another mechanism.

Table 1: Condensed strict-independent comparison on the four displayed tasks with Evo2 test-time baselines. Each entry is mean ± sample SD over three independent seeds; 256 locked sequences are nested within each seed. All locks precede Evaluation and post-hoc sequence metrics. Evaluation MinGap and entropy are omitted here. The four added test-time baselines use the same Evo2 7B family-specific adapter, split, seed set, query budget, and Top-256 lock protocol as ToDNA. Motif metrics use the JASPAR 2024 CORE vertebrates non-redundant collection with FIMO 5.5.7 at $p \leq 10^{-4}$; 3-mer metrics use all 64 count profiles against the task-matched frozen reference. The added test-time baselines are method-idea adaptations rather than byte-for-byte reproductions of the cited pipelines. ToDNA is marked as the authors' method in the final column. Bold marks the highest mean for each directional metric within a task, not statistical significance.

Task	Metric	Ctrl-DNA	PPO	GRPO	LM-Evo	Infilling	Evo2-CEM	Evo2-CREsted	ToDNA (ours)
HepG2 (enh.)	Evaluation $\Delta R_{\text{mean}} \uparrow$	$\mathbf{0.381 \pm 0.021}$	0.024 ± 0.003	0.032 ± 0.002	0.221 ± 0.033	0.248 ± 0.068	0.214 ± 0.015	0.020 ± 0.005	0.282 ± 0.005
	Motif Pearson ↑	0.535 ± 0.091	0.651 ± 0.013	0.610 ± 0.011	0.673 ± 0.072	0.332 ± 0.107	0.514 ± 0.045	0.446 ± 0.087	$\mathbf{0.844 \pm 0.020}$
	Motif Spearman ↑	0.625 ± 0.076	0.784 ± 0.014	0.744 ± 0.010	0.713 ± 0.091	0.423 ± 0.060	0.657 ± 0.014	0.564 ± 0.042	$\mathbf{0.821 \pm 0.037}$
	3-mer Pearson ↑	0.521 ± 0.097	0.678 ± 0.009	0.449 ± 0.019	0.491 ± 0.187	0.125 ± 0.300	0.279 ± 0.188	0.011 ± 0.264	$\mathbf{0.770 \pm 0.057}$
K562 (enh.)	Evaluation $\Delta R_{\text{mean}} \uparrow$	$\mathbf{0.498 \pm 0.010}$	0.084 ± 0.011	0.138 ± 0.019	0.245 ± 0.041	0.379 ± 0.067	0.323 ± 0.177	0.043 ± 0.020	0.348 ± 0.026
	Motif Pearson ↑	0.581 ± 0.079	0.723 ± 0.011	0.686 ± 0.014	0.685 ± 0.107	0.472 ± 0.114	0.632 ± 0.117	0.581 ± 0.175	$\mathbf{0.865 \pm 0.012}$
	Motif Spearman ↑	0.618 ± 0.014	0.687 ± 0.017	0.633 ± 0.016	0.764 ± 0.069	0.497 ± 0.064	0.665 ± 0.024	0.658 ± 0.048	$\mathbf{0.789 \pm 0.017}$
	3-mer Pearson ↑	0.312 ± 0.038	0.080 ± 0.040	-0.058 ± 0.008	0.359 ± 0.258	0.209 ± 0.334	0.275 ± 0.054	0.076 ± 0.211	$\mathbf{0.667 \pm 0.018}$
SK-N-SH (enh.)	Evaluation $\Delta R_{\text{mean}} \uparrow$	0.220 ± 0.026	0.022 ± 0.001	0.022 ± 0.000	$\mathbf{0.263 \pm 0.004}$	0.204 ± 0.072	0.225 ± 0.036	0.021 ± 0.007	0.193 ± 0.009
	Motif Pearson ↑	0.329 ± 0.138	0.382 ± 0.052	0.349 ± 0.022	0.439 ± 0.078	0.372 ± 0.177	0.188 ± 0.035	0.521 ± 0.136	$\mathbf{0.759 \pm 0.030}$
	Motif Spearman ↑	0.421 ± 0.184	0.534 ± 0.034	0.492 ± 0.034	0.734 ± 0.029	0.473 ± 0.060	0.637 ± 0.045	0.612 ± 0.036	$\mathbf{0.792 \pm 0.032}$
	3-mer Pearson ↑	0.381 ± 0.139	0.657 ± 0.030	0.626 ± 0.008	0.512 ± 0.099	0.159 ± 0.205	0.240 ± 0.162	0.004 ± 0.180	$\mathbf{0.716 \pm 0.056}$
K562 (prom.)	Evaluation $\Delta R_{\text{mean}} \uparrow$	-0.192 ± 0.002	-0.180 ± 0.003	-0.154 ± 0.001	-0.154 ± 0.004	-0.152 ± 0.015	$\mathbf{-0.151 \pm 0.002}$	-0.172 ± 0.002	-0.171 ± 0.000
	Motif Pearson ↑	0.424 ± 0.050	0.141 ± 0.005	0.218 ± 0.008	0.711 ± 0.051	0.476 ± 0.241	0.273 ± 0.071	0.807 ± 0.047	$\mathbf{0.865 \pm 0.004}$
	Motif Spearman ↑	0.460 ± 0.033	0.432 ± 0.021	0.534 ± 0.021	0.620 ± 0.007	0.584 ± 0.027	0.564 ± 0.013	0.623 ± 0.003	$\mathbf{0.632 \pm 0.014}$
	3-mer Pearson ↑	0.106 ± 0.071	0.161 ± 0.007	0.546 ± 0.040	0.786 ± 0.065	0.685 ± 0.128	0.612 ± 0.123	0.842 ± 0.022	$\mathbf{0.905 \pm 0.003}$

Replacing adapted logits produces a different pattern. Mean 3-mer Pearson decreases from 0.874 to 0.857, whereas the mean specificity reductions are smaller and reverse direction in one run. Adapted local preferences therefore contribute to sequence-profile agreement, while their specificity benefit is less consistent in this study. Because the pools remain fixed, this ablation assesses initialization and reference construction, not the full effect of adaptation on candidate generation. Together, the comparisons distinguish the contribution of reference regularization from the information supplied by the adapted reference. The evidence is limited to three runs on one enhancer task and does not establish that either component guarantees biological selectivity or the exact-KL assumptions required by the theoretical analysis.

Table 2: HepG2 ablations of reference regularization and adapted logits with paired starts, pools, and query budgets. Values are mean ± sample SD across three runs. MinGap is a raw predictor score, distinct from the main normalized metric. Base logits replaces the adapted initialization and reference while retaining the candidate pools. No arm enforces composition filters or a hard edit cap. Bold marks the best displayed mean, not significance.

Method	Evaluation MinGap ↑	FullMPRA MinGap ↑	3-mer Pearson ↑	Edits ↓
ToDNA	$\mathbf{6.037 \pm 0.114}$	$\mathbf{6.601 \pm 0.183}$	$\mathbf{0.874 \pm 0.013}$	$\mathbf{67.749 \pm 1.037}$
w/o KL	5.890 ± 0.132	6.364 ± 0.145	0.862 ± 0.019	78.165 ± 1.410
Base logits	5.976 ± 0.250	6.543 ± 0.263	0.857 ± 0.021	68.456 ± 2.180

6 Discussion and Conclusion

ToDNA separates element-class adaptation from cell-specific candidate optimization, allowing a fixed Evo 2 prior to support multiple objectives. Enhancer Design trajectories show continued score gains, but the weaker promoter response and objective-specific trade-offs delimit current control. The results support test-time optimization as a reusable inference-layer mechanism, not biological validation: mean-off-target rewards can mask one off-target response, motif agreement is not functional evidence, and related predictors may share errors. Given three runs and no assays, target/off-target MPRAs and end-to-end compute comparisons remain necessary.

AI USE STATEMENT

OpenAI Codex assisted with figure preparation, literature checking, and manuscript drafting and editing. The authors remain responsible for the accuracy of the reported results, citations, and conclusions.

Reproducibility and responsible use. Appendix A documents settings and baselines. Aggregate results and scripts are provided, but unavailable checkpoints, sequences, and reward settings preclude full reproduction. Experimental use requires biosafety oversight.

REFERENCES

Vikram Agarwal et al. Massively parallel characterization of transcriptional regulatory elements. *Nature*, 639(8054):411–420, 2025. doi: 10.1038/s41586-024-08430-9.

Pavel Avdeyev, Chenlai Shi, Yuhao Tan, Kseniia Dudnyk, and Jian Zhou. Dirichlet diffusion score model for biological sequence generation. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 1276–1301. PMLR, 2023. URL https://proceedings.mlr.press/v202/avdeyev23a.html.

Žiga Avsec, Vikram Agarwal, Daniel Visentin, Joseph R. Ledsam, Agnieszka Grabska-Barwinska, Kyle R. Taylor, Yannis Assael, John Jumper, Pushmeet Kohli, and David R. Kelley. Effective gene expression prediction from sequence by integrating long-range interactions. *Nature Methods*, 18 (10):1196–1203, 2021. doi: 10.1038/s41592-021-01252-x.

Garyk Brixi, Matthew G. Durrant, Jerome Ku, et al. Genome modelling and design across all domains of life with Evo 2. *Nature*, 652:1349–1361, 2026. doi: 10.1038/s41586-026-10176-5.

Xingyu Chen, Shihao Ma, Runsheng Lin, Jiecong Lin, and Bo Wang. Ctrl-DNA: Controllable cell-type-specific regulatory DNA design via constrained RL. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-0472. URL https://openreview.net/forum?id=JeXkIy0JyM.

Hugo Dalla-Torre, Liam Gonzalez, Javier Mendoza-Revilla, Nicolas Lopez Carranza, Adam Henryk Grzywaczewski, Francesco Oteri, Christian Dallago, Evan Trop, Bernardo P. de Almeida, Hassan Sirelkhatim, Guillaume Richard, Marcin Skwark, Karim Beguir, Marie Lopez, and Thomas Pierrot. Nucleotide transformer: building and evaluating robust foundation models for human genomics. *Nature Methods*, 22(2):287–297, 2025. doi: 10.1038/s41592-024-02523-z.

Carl G. de Boer, Eeshit Dhaval Vaishnav, Ronen Sadeh, Esteban Luis Abeyta, Nir Friedman, and Aviv Regev. Deciphering eukaryotic gene-regulatory logic with 100 million random promoters. *Nature Biotechnology*, 38(1):56–65, 2020. doi: 10.1038/s41587-019-0315-8.

Luca Eyring, Shyamgopal Karthik, Karsten Roth, Alexey Dosovitskiy, and Zeynep Akata. ReNO: Enhancing one-step text-to-image models through reward-based noise optimization. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-3987. URL https://papers.neurips.cc/paper_files/paper/2024/hash/e31bdea0a93741c2157eea705dd219eb-Abstract-Conference.html.

Sager J. Gosai, Rodrigo I. Castro, Natalia Fuentes, John C. Butts, Kousuke Mouri, Michael Alasoadura, Susan Kales, Thanh Thanh L. Nguyen, Ramil R. Noche, Arya S. Rao, Mary T. Joy, Pardis C. Sabeti, Steven K. Reilly, and Ryan Tewhey. Machine-guided design of cell-type-targeting cis-regulatory elements. *Nature*, 634(8036):1211–1220, 2024. doi: 10.1038/s41586-024-08070-z.

Charles E. Grant, Timothy L. Bailey, and William Stafford Noble. FIMO: Scanning for occurrences of a given motif. *Bioinformatics*, 27(7):1017–1018, 2011. doi: 10.1093/bioinformatics/btr064.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Korrawe Karunratanakul, Konpat Preechakul, Emre Aksan, Thabo Beeler, Supasorn Suwajanakorn, and Siyu Tang. Optimizing diffusion noise can serve as universal motion priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1334–1345, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Karunratanakul_Optimizing_Diffusion_Noise_Can_Serve_As_Universal_Motion_Priors_CVPR_2024_paper.html.

Niklas Kempynck, Seppe De Winter, Casper H. Blaauw, et al. CREsted: modeling genomic and synthetic cell-type-specific enhancers across tissues and species. *Nature Methods*, 23(5):946–959, 2026. doi: 10.1038/s41592-026-03057-2.

Sunwoo Kim, Minkyu Kim, and Dongmin Park. Test-time alignment of diffusion models without reward over-optimization. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=vi3DjUhFVm.

Avantika Lal, David Garfield, Tommaso Biancalani, and Gokcen Eraslan. Designing realistic regulatory DNA with autoregressive language models. *Genome Research*, 34(9):1411–1420, 2024. doi: 10.1101/gr.279142.124.

Johannes Linder and Georg Seelig. Fast activation maximization for molecular sequence design. *BMC Bioinformatics*, 22:510, 2021. doi: 10.1186/s12859-021-04437-5.

Johannes Linder, Nicholas Bogard, Alexander B. Rosenberg, and Georg Seelig. A generative neural network for maximizing fitness and diversity of synthetic DNA and protein sequences. *Cell Systems*, 11(1):49–62.e16, 2020. doi: 10.1016/j.cels.2020.05.007.

Eric Nguyen, Michael Poli, Marjan Faizi, Armin W. Thomas, Callum Birch-Sykes, Michael Wornow, Aman Patel, Clayton M. Rabideau, Stefano Massaroli, Yoshua Bengio, Stefano Ermon, Stephen A. Baccus, and Christopher Ré. HyenaDNA: Long-range genomic sequence modeling at single nucleotide resolution. In *Advances in Neural Information Processing Systems*, volume 36, pp. 43177–43201, 2023. doi: 10.52202/075280-1872. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/86ab6927ee4ae9bde4247793c46797c7-Paper-Conference.pdf.

Eric Nguyen, Michael Poli, Matthew G. Durrant, Brian Kang, Dhruva Katrekar, David B. Li, Liam J. Bartie, Armin W. Thomas, Samuel H. King, Garyk Brixi, Jeremy Sullivan, Madelena Y. Ng, Ashley Lewis, Aaron Lou, Stefano Ermon, Stephen A. Baccus, Tina Hernandez-Boussard, Christopher Ré, Patrick D. Hsu, and Brian L. Hie. Sequence modeling and design from molecular to genome scale with Evo. *Science*, 386(6723):eado9336, 2024. doi: 10.1126/science.ado9336.

Ieva Rauluseviciute, Rafael Riudavets-Puig, Romain Blanc-Mathieu, Jaime A. Castro-Mondragon, Katalin Ferenc, Vipin Kumar, Roza Berhanu Lemma, Jérémy Lucas, Jeanne Chèneby, Damir Baranasic, Aziz Khan, Oriol Fornes, Sveinung Gundersen, Morten Johansen, Eivind Hovig, Boris Lenhard, Albin Sandelin, Wyeth W. Wasserman, François Parcy, and Anthony Mathelier. JASPAR 2024: 20th anniversary of the open-access database of transcription factor binding profiles. *Nucleic Acids Research*, 52(D1):D174–D182, 2024. doi: 10.1093/nar/gkad1059.

Aniketh Janardhan Reddy, Xinyang Geng, Michael H. Herschl, Sathvik Kolli, Aviral Kumar, Patrick D. Hsu, Sergey Levine, and Nilah M. Ioannidis. Designing cell-type-specific promoter sequences using conservative model-based optimization. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://openreview.net/forum?id=F8DWffLkYG.

Anirban Sarkar, Alejandra Duran, and Peter K. Koo. GPA: Generative population annealing for test-time sequence design. In *Workshop on Generative and Agentic AI for Biology at ICML*, 2026. URL https://openreview.net/forum?id=4JTkToC9Ox.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. doi: 10.48550/arXiv.1707.06347.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. doi: 10.48550/arXiv.2402.03300.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=4FWAwZtd2n.

Ibrahim I. Taskiran, Katina I. Spanier, Hannah Dickmänken, et al. Cell-type-directed design of synthetic enhancers. *Nature*, 626:212–220, 2024. doi: 10.1038/s41586-023-06936-2.

Masatoshi Uehara, Xingyu Su, Yulai Zhao, Xiner Li, Aviv Regev, Shuiwang Ji, Sergey Levine, and Tommaso Biancalani. Reward-guided iterative refinement in diffusion models at test-time with applications to protein and DNA design. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 60515–60529. PMLR, 2025. URL https://proceedings.mlr.press/v267/uehara25a.html.

Eeshit Dhaval Vaishnav, Carl G. de Boer, Jennifer Molinet, Moran Yassour, Lin Fan, Xian Adiconis, Dawn A. Thompson, Joshua Z. Levin, Francisco A. Cubillos, and Aviv Regev. The evolution, evolvability and engineering of gene regulatory DNA. *Nature*, 603(7901):455–463, 2022. doi: 10.1038/s41586-022-04506-6.

Chenyu Wang, Masatoshi Uehara, Yichun He, Amy Wang, Avantika Lal, Tommi Jaakkola, Sergey Levine, Aviv Regev, Hanchen Wang, and Tommaso Biancalani. Fine-tuning discrete diffusion models via reward optimization with applications to DNA and protein design. In *International Conference on Learning Representations*, 2025.

Ye Wang, Haochen Wang, Lei Wei, Shuailin Li, Liyang Liu, and Xiaowo Wang. Synthetic promoter design in Escherichia coli based on a deep generative network. *Nucleic Acids Research*, 48(12): 6403–6412, 2020. doi: 10.1093/nar/gkaa325.

Zhao Yang, Bing Su, Chuan Cao, and Ji-Rong Wen. Regulatory DNA sequence design with reinforcement learning. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=F4IMiNhim1.

Jan Zrimec, Xiaozhi Fu, Azam Sheikh Muhammad, Christos Skrekas, Vykintas Jauniskis, Nora K. Speicher, Christoph S. Börlin, Vilhelm Verendel, Morteza Haghir Chehreghani, Devdatt Dubhashi, Verena Siewers, Florian David, Jens Nielsen, and Aleksej Zelezniak. Controlling gene expression with deep generative design of regulatory DNA. *Nature Communications*, 13:5099, 2022. doi: 10.1038/s41467-022-32818-8.

A Implementation and Evaluation Details

This appendix provides the implementation details, optimization trajectories, reward comparisons, and proofs supporting cell-specific DNA design with a fixed genomic generator. Unless stated otherwise, results are means and sample standard deviations across three runs. Functional scores are predictor outputs; sequence-level observations are aggregated within runs before uncertainty is calculated.

A.1 Tasks

The main-text comparison and this task summary use four displayed target tasks, matching Table 1: HepG2 (E), K562 (E), SK-N-SH (E), and K562 (P). The enhancer tasks use 200-bp sequences and the promoter task uses 250-bp sequences. For each target task, the other measured channels in the same regulatory-element family define the off-target context. These channels therefore occur only as predictor channels for K562 (P), not as additional displayed target tasks. Promoters are centered in zero-padded inputs of length 2,114. Within each family, Design and Evaluation checkpoints are trained on separate sequence-group partitions and frozen during candidate optimization. Their shared architecture and pretraining limit the independence of their predictions. Independence of generated sequences from all predictor training data has not been established.

We normalize activity using $s_c = (a_c - \ell_c)/(u_c - \ell_c)$ with the fixed bounds in Table 3. Values are not clipped to $[0, 1]$. These study-specific bounds lack a documented derivation and do not represent calibrated biological activity. The component study reports unnormalized MinGap; its absolute scores are not comparable with the main normalized results.

Table 3: Fixed activity-normalization bounds for the four displayed target tasks. Bounds for promoter off-target channels used internally by the K562 (P) reward are omitted from this table.

Family	Cell channel	Lower bound	Upper bound
Enhancer	HepG2	-6.051336	10.992575
Enhancer	K562	-5.857445	10.781755
Enhancer	SK-N-SH	-7.283977	12.888308
Promoter	K562	-4.088671	10.781755

A.2 Hyperparameters

Enhancers and promoters use separate Evo 2 adapters trained on their respective Design partitions. Both adapters use rank 8, scale 16, and 2,000 training steps. Each adapter remains fixed across target cells within its sequence family. Table 4 lists the candidate-search settings for the main comparison.

Table 4: Hyperparameters for the main ToDNA comparison.

Parameter	Value
Initial pool / selected starts	768 / 128
Search attempts per start	12
Optimization-stage Design evaluations	15,360
Output sequences per run	256
Adam steps / learning rate	10 / 0.1
Softmax temperature / KL weight	0.7 / 1.0
Gradient clipping norm / initial logit margin	5.0 / 0.25
Hard edit cap / GC and homopolymer rejection	None / disabled

The budget counts optimization-stage predictor calls and excludes pool scoring, initial and final scoring, and diagnostics. It therefore does not establish equal total compute across methods. The main comparison uses repeated attempts without explicit logit perturbations or enforced deterministic execution. Their variability remains unexplained, so attempts are not treated as independent replicates. Reward-extension studies instead introduce stochastic logit perturbations (Appendix C).

A.3 INFERENCE-TIME SEQUENCE OPTIMIZATION

The following pseudocode makes explicit the separation between pool selection, candidate updates, and final locking. It follows the objective and straight-through estimator in Section 3.3; it does not add hard biological constraints.

Algorithm 1 ToDNA cell-specific test-time optimization

Require: Adapted generator p_{θ,η_e}, target cell t, and frozen Design predictor f_ψ
Require: Search settings $(K, T, M, \tau, \lambda_{\mathrm{KL}})$
Ensure: Locked candidate set $\widehat{\mathcal{X}}_t$ with post-lock predictor scores

```
Sample a candidate pool 𝒫 ~ p_{θ,η_e} and score it with f_ψ
Select K starts {x_k^0}_{k=1}^K from 𝒫
for k = 1, ..., K do
    Teacher-force x_k^0 to obtain margin-adjusted logits z_k^0 and reference q_k^0
    Set z_k ← z_k^0
    for r = 1, ..., T do
        q_{k,i} ← softmax(z_{k,i}/τ) and x_{k,i} ← arg max_{b∈𝒱} z_{k,i,b} for i = 1, ..., L
        q̃_{k,i} ← sg(onehot(x_{k,i}) − q_{k,i}) + q_{k,i} for i = 1, ..., L
        ℒ_k ← −R_t(q̃_k) + λ_KL L^{-1} Σ_{i=1}^L D_KL(q_{k,i} ‖ q_{k,i}^0)
        Take one Adam step on z_k to minimize ℒ_k; apply configured gradient clipping
    end for
    Decode x̂_k ← h(z_k) and compute its final predictor scores
end for
Lock the top M candidates using the prespecified output-selection rule
```

A.4 LOCAL REFERENCE CONSTRUCTION

Teacher-forced Evo 2 predictions at a start x^0 supply A/C/G/T logits $v \in \mathbb{R}^{L\times 4}$. We shift predictions to align with their next-nucleotide targets and set $v_{1,b} = 0$. The adjusted initial logits are

$$z^0_{i,b} = v_{i,b} + \mathbf{1}_{\{b=x^0_i\}} \left[\max_{c\in\mathcal{V}} v_{i,c} - v_{i,x^0_i} + m\right]_+, \tag{14}$$

where $[u]_+ = \max(u, 0)$ and $m > 0$ is the logit margin in Table 4. In exact arithmetic, this adjustment ensures that hard decoding recovers x^0, and $q^0_i = \mathrm{softmax}(z^0_i/\tau)$ remains fixed throughout search. The probability margin in Proposition 1 depends on all four logits and τ, rather than equaling the added logit margin. This positionwise reference is conditioned on the starting sequence; its KL penalty is not the autoregressive likelihood of an edited sequence.

The supplied implementation floors probabilities before evaluating the regularizer. For floating-point epsilon $\varepsilon_{\mathrm{fp}} > 0$, it uses

$$\Omega_\varepsilon(q, q^0) = \frac{1}{L}\sum_{i=1}^{L}\sum_{b\in\mathcal{V}} \bar{q}_{i,b} \log \frac{\bar{q}_{i,b}}{\bar{q}^0_{i,b}}, \qquad \bar{q}_{i,b} = \max(q_{i,b}, \varepsilon_{\mathrm{fp}}), \quad \bar{q}^0_{i,b} = \max(q^0_{i,b}, \varepsilon_{\mathrm{fp}}). \tag{15}$$

The floored arrays are not renormalized. This equals exact KL when the floor is inactive, but otherwise need not be nonnegative or obey Proposition 1. Available summaries do not establish whether flooring was active during the reported searches; the exact-KL analysis is therefore not an unconditional guarantee for this numerical implementation.

A.5 BASELINE IMPLEMENTATIONS

All compared methods receive the search budget in Table 4 for each task and run. They differ in proposal generation, model calls, and final candidate generation, so this budget controls only search-stage predictor use.

Ctrl-DNA. We use the public implementation with the same Design and Evaluation predictors, candidate pools, and selected starts as ToDNA. Its constrained reinforcement-learning search receives

the matched optimization-stage budget. The predictor checkpoints from its published results are unavailable, so this comparison assesses a reconstruction under common predictors.

PPO and GRPO. Both initialize the same family-specific LoRA adapter as ToDNA and update its weights while retaining the frozen Evo 2 backbone (Schulman et al., 2017; Shao et al., 2024). Autoregressive rollouts are organized into groups of four sequences. PPO uses a learned value head and GAE with $\gamma = 1$ and $\lambda = 0.95$; GRPO standardizes rewards within each group. Both use clipping 0.2, reference KL weight 0.05, learning rate 10^{-5}, and two update epochs. Each final policy generates a candidate pool of the size in Table 4 for deduplication and Design ranking.

Alternative test-time strategies. LM-Evo uses language-model-guided point mutation and selection; Infilling resamples short spans from causal left-context logits. Evo2-CEM updates categorical proposals through cross-entropy selection. Evo2-CREsted uses differentiable candidate-logit optimization with a frozen Evo 2 prior, adapting the design principle of CREsted (Kempynck et al., 2026). These four variants share the adapter, data splits, paired runs, query budget, and final candidate-pool size. Evo2-CREsted is a method adaptation, not a reproduction of the original pipeline.

B Optimization Trajectories: Protocol

The trajectory analysis in Section 5.3 follows 32 starts per task and run for ten optimization steps. Starts are chosen before observing optimization outcomes. Design scores, hard decodes, and sequence statistics are recorded during search, and Evaluation scores each state without providing optimization feedback. Curves first average starts within each run, then report the mean and sample SD across three runs. Only aggregate trajectories are available; complete sequence and logit trajectories are unavailable.

The enhancer adapter uses an ENCODE-derived corpus, rank 16, scale 32, and 400 training steps at learning rate 2×10^{-4}. The promoter adapter uses rank 8, scale 16, and 2,000 steps at learning rate 10^{-4}. Both apply dropout 0.05 and use held-out language-model loss for selection. These adapters differ from those in the main benchmark. Trajectory results are therefore interpreted within this diagnostic and are not pooled with the benchmark's selected outputs.

C ToDNA Adapts to Different Biological Properties

We next test whether different design objectives can be specified by changing the reward without retraining the generator or its LoRA adapter. Separate enhancer studies examine activity contrast and sequence properties across three target cells. Reward arms share pools, starts, and search budgets within each study. Unlike the main benchmark, these searches perturb initial logits, so absolute scores are compared only within each study.

C.1 Activity and specificity require different rewards

Target-only optimization yields the highest FullMPRA target activity for every cell, whereas all three contrast rewards yield larger raw MinGap (Table 5). FullMPRA is an assessment predictor, and MinGap is its target activity minus the strongest off-target activity. For K562, replacing target-only with normalized mean contrast reduces target activity from 8.352 ± 0.080 to 6.930 ± 0.162, while increasing MinGap from 2.670 ± 0.211 to 5.316 ± 0.292. Mean contrast gives the largest mean MinGap for HepG2 and K562, whereas raw smooth-MinGap leads for SK-N-SH. These results distinguish increasing target activity from controlling relative cell preference, with no single contrast formulation leading across all targets.

The reward-comparison runner applies a fixed affine transformation $(R - c_R)/\sigma_R$ to each base reward, using its pool median c_R and positive robust scale σ_R. This preserves reward ordering but changes the reward-to-KL trade-off and is distinct from activity normalization.

Table 5: Changing the activity reward trades target activation against cell specificity. FullMPRA target activity and raw MinGap are predictions. Values are mean ± sample SD across three runs. Bold marks the largest mean within each target and metric, not statistical significance.

Target	Reward	Target activity	MinGap
K562	Target-only	$\mathbf{8.352 \pm 0.080}$	2.670 ± 0.211
	Raw smooth-MinGap	6.466 ± 0.254	5.202 ± 0.248
	Normalized smooth-MinGap	6.818 ± 0.131	5.226 ± 0.172
	Normalized mean margin	6.930 ± 0.162	$\mathbf{5.316 \pm 0.292}$
HepG2	Target-only	$\mathbf{7.127 \pm 0.135}$	0.721 ± 0.252
	Raw smooth-MinGap	5.220 ± 0.116	4.218 ± 0.205
	Normalized smooth-MinGap	5.437 ± 0.161	4.275 ± 0.222
	Normalized mean margin	5.468 ± 0.113	$\mathbf{4.340 \pm 0.210}$
SK-N-SH	Target-only	$\mathbf{8.398 \pm 0.212}$	1.739 ± 0.047
	Raw smooth-MinGap	5.409 ± 0.162	$\mathbf{3.334 \pm 0.163}$
	Normalized smooth-MinGap	5.200 ± 0.349	3.288 ± 0.177
	Normalized mean margin	5.959 ± 0.289	3.180 ± 0.100

C.2 Property rewards expose competing design requirements

Table 6 compares accessibility, soft motif, 3-mer, compliance, and multi-objective rewards. Accessibility optimization yields predicted probabilities of 0.994–0.996, but 3-mer correlations range from -0.039 to 0.310. Conversely, 3-mer optimization yields correlations of 0.940–0.957, while raw MinGap remains negative for every target. The compliance reward produces outputs with no homopolymer longer than six bases, but does not consistently produce positive MinGap. These outcomes show that optimizing a sequence property does not by itself specify the desired target-versus-off-target relationship.

The multi-objective reward combines activity contrast with all four property terms. It yields the largest mean MinGap and motif Pearson within this study, while retaining 3-mer correlations of 0.819–0.889. Its specificity advantage cannot be attributed to synergy among property terms, because the single-property arms omit activity contrast. The soft motif reward also differs from distributional motif agreement: it scores selected motifs, rather than matching their frequency profile. Accessibility uses the same predictor during search and assessment, so its high scores do not constitute independent validation.

The supplied implementation matches differentiable expected 3-mer frequencies to the reference using a smooth L1 loss. Compliance penalizes departures from GC bounds and long homopolymers. The composite uses equal weights after pool-based robust standardization of its five components.

D Theoretical Proofs

The following derivations establish the statements in Section 4. All KL divergences use natural logarithms.

D.1 Conditional objective improvement

At initialization, $\Omega(q(z^0), q^0) = 0$ and $h(z^0) = x^0$, so $J_t(z^0) = R_t(x^0)$. Rearranging $J_t(z) \geq J_t(z^0)$ gives Equation 10. This premise is not guaranteed by straight-through optimization: its backward pass differentiates a soft relaxation rather than the piecewise-constant hard decoder. Finite-step Adam and a positive KL penalty therefore need not produce monotonic improvement of the hard-decoded objective.

D.2 Proof of Proposition 1: decoding-margin bound

For a changed position i, write $a = x_i^0$ and $b = h_i(z) \neq a$. Since $q_{i,b} \geq q_{i,a}$ and $q_{i,a}^0 - q_{i,b}^0 \geq \gamma_i$,

$$\gamma_i \leq (q_{i,a}^0 - q_{i,a}) + (q_{i,b} - q_{i,b}^0) \leq \|q_i - q_i^0\|_1. \tag{16}$$

Table 6: Sequence-property rewards produce different output profiles. Values are mean ± sample SD across three runs. Accessibility is scored by the same predictor used in its reward. Motif and 3-mer Pearson compare output profiles with natural references; MinGap uses FullMPRA. The multi-objective arm also includes activity contrast. Bold marks the largest mean within each target and metric, not statistical significance.

Target	Reward	Accessibility probability	Motif Pearson	3-mer Pearson	MinGap
HepG2	Accessibility	$\mathbf{0.9943 \pm 0.0004}$	0.726 ± 0.032	0.310 ± 0.087	2.105 ± 0.326
	Soft motif	0.1636 ± 0.0344	0.562 ± 0.066	0.901 ± 0.034	0.359 ± 0.154
	3-mer	0.1075 ± 0.0095	0.547 ± 0.042	$\mathbf{0.956 \pm 0.008}$	-0.089 ± 0.052
	Compliance	0.1771 ± 0.0064	0.614 ± 0.019	0.835 ± 0.013	-0.009 ± 0.038
	Multi-objective	0.9787 ± 0.0020	$\mathbf{0.773 \pm 0.027}$	0.883 ± 0.023	$\mathbf{6.470 \pm 0.050}$
K562	Accessibility	$\mathbf{0.9959 \pm 0.0005}$	0.754 ± 0.026	0.001 ± 0.041	1.530 ± 0.141
	Soft motif	0.1963 ± 0.0287	0.593 ± 0.037	0.918 ± 0.037	0.473 ± 0.146
	3-mer	0.1180 ± 0.0052	0.556 ± 0.022	$\mathbf{0.940 \pm 0.003}$	-0.073 ± 0.020
	Compliance	0.2155 ± 0.0118	0.669 ± 0.032	0.870 ± 0.031	0.267 ± 0.133
	Multi-objective	0.9763 ± 0.0018	$\mathbf{0.858 \pm 0.031}$	0.819 ± 0.048	$\mathbf{6.913 \pm 0.176}$
SK-N-SH	Accessibility	$\mathbf{0.9939 \pm 0.0016}$	0.702 ± 0.060	-0.039 ± 0.093	1.246 ± 0.250
	Soft motif	0.1807 ± 0.0307	0.598 ± 0.013	0.915 ± 0.023	0.185 ± 0.110
	3-mer	0.1536 ± 0.0083	0.603 ± 0.014	$\mathbf{0.957 \pm 0.003}$	-0.184 ± 0.017
	Compliance	0.2026 ± 0.0237	0.685 ± 0.048	0.794 ± 0.022	0.157 ± 0.062
	Multi-objective	0.9274 ± 0.0238	$\mathbf{0.748 \pm 0.055}$	0.889 ± 0.018	$\mathbf{4.363 \pm 0.512}$

The categorical inequality $D_{\mathrm{KL}}(p\|q) \geq \frac{1}{2}\|p-q\|_1^2$ then gives $D_{\mathrm{KL}}(q_i\|q_i^0) \geq \gamma_i^2/2$. To verify the inequality, group categories with $p_b \geq q_b$ and apply the log-sum inequality. The case $p = q$ is immediate; otherwise, full support of q makes the reference group's mass v lie in $(0, 1)$. The resulting binary probabilities satisfy $u - v = \|p-q\|_1/2$. Binary KL has value and first derivative zero at $u = v$, and second derivative $1/[u(1-u)] \geq 4$, yielding the lower bound $2(u-v)^2$; boundary values of u follow by continuity.

Summing over changed positions gives $\sum_{i\in\mathcal{I}} \gamma_i^2 \leq 2L\Omega(q, q^0)$. Using $\sum_{i\in\mathcal{I}} \gamma_i^2 \geq |\mathcal{I}|\gamma_{\min}^2$ proves the Hamming bound in Equation 11. The proof holds under the fixed tie-breaking rule used by h, while q^0 must have a unique maximizer. Without a positive reference margin, arbitrarily small probability changes can alter the decoded base. The result assumes exact probabilities; numerical underflow requires separate implementation checks.

D.3 Proof of Proposition 2: off-target dispersion

Subtracting the two specificity definitions gives

$$R_t(x) - \Delta R_{\min}(x; t) = \max_{o\in\mathcal{O}_t} s_o(x) - \bar{s}_{\mathcal{O}_t}(x) = D_t(x). \tag{17}$$

For an optimized output $\hat{x}_t$,

$$\Delta R_{\min}(\hat{x}_t; t) - \Delta R_{\min}(x^0; t) = G_t(x^0) - [D_t(\hat{x}_t) - D_t(x^0)]. \tag{18}$$

With two off-targets, $D_t = |s_{o_1} - s_{o_2}|/2$. Thus, mean-margin improvement increases worst-off-target specificity only when it exceeds the increase in dispersion. As a numerical illustration, $(s_t, s_{o_1}, s_{o_2}) = (0.6, 0.8, 0.0)$ gives $R_t = 0.2$ but $\Delta R_{\min} = -0.2$.