SPROUT: Steered Plant Promoter Editing via Rollout-Guided Utility Tilting of Edit Flows

Anonymous Authors[1]

Abstract

Engineering plant cis-regulatory elements (CREs) has high-impact applications in domains including agriculture, bioremediation, and bioengineering. Existing CRE engineering frameworks typically use either naive generative models, including random sequence generation or genetic algorithms, and/or reductive descriptions of expression context. The common formulation for cell-type specific expression – expression enrichment in one cell type over another – ignores all other cell types and axes of expression. For plant promoters, functional utility depends upon controlling a richer expression context across tissues, developmental stages, and environmental conditions. We thus propose **SPROUT**, an alternative formulation for CRE engineering based on **S**teered plant **P**romoter editing via **RO**llout-guided **U**tility **T**ilting of Edit Flows, pairing a recently introduced discrete flow matching framework for variable-length sequence editing with multi-objective optimization across oracles predicting expression context and strength. We evaluate SPROUT against the common formulations and show that our Edit Flow-based formulation produces more realistic sequences than random generation, while optimization with oracles trained on only two conditions does not fully constrain other expression axes.

1. Background and Related Works

Engineering cis-regulatory elements (CREs) holds great promise for augmenting plant capabilities in agricultural and biotechnological settings, since many relevant metabolic, developmental, or stress-responsive traits depend on precise regulatory control of gene expression. Studies of artificial intelligence (AI)-guided CRE generation in plants primarily intend to increase gene expression by utilizing genetic algorithms or random sequence generation paired with oracle scoring (Li et al., 2024; Deng et al., 2025; Jores et al., 2021; 2026). Although these generative methods are not considered state-of-the-art, the generated sequences nevertheless drive increased or otherwise modified expression of target genes (Deng et al., 2025; Jores et al., 2021), highlighting the feasibility of this task.

In non-plant organisms (commonly fly or human cell lines), recent works have extended CRE design to control not only expression strength but context as well. So-called "cell-type-specific" CREs are generated by training an oracle to predict expression enrichment in one cell type versus one to five others (Taskiran et al., 2024; de Almeida et al., 2024). Although promising, these methods ignore the possibility that the engineered CRE may additionally drive expression in a cell type that the oracle was not trained on. Further, oracles trained only on cell type cannot predict under which environmental conditions or developmental stages a CRE may be active within, even though these "axes" are essential for precise control of gene expression.

Recent works have developed discrete diffusion (DaSilva et al., 2026; Sarkar et al., 2024; Yang et al., 2026) or flow-matching (Davis et al., 2024; Stark et al., 2024) algorithms for DNA generation and design. These works often explore generation of two types of CREs: promoters, which are typically situated proximal to a gene and are essential for transcription initiation, and enhancers, which are more often distal to their target genes and fine-tune, or "enhance" gene expression (Andersson & Sandelin, 2020). For both CRE types, conditional design is formulated as generation of new sequences which closely match an "expression embedding" of a true promoter or enhancer. This vector is generated by the oracle Sei (Chen et al., 2022), and contains dense information on the expression context that the CRE drives. Although promising, these generated sequences have not been validated in any *in vivo* or *in vitro* experiments, and the contexts in which, and strength with which, these sequences drive expression is not immediately clear.

Herein, we introduce **SPROUT** (**S**teered plant **P**romoter editing via **RO**llout-guided **U**tility **T**ilting of Edit Flows),

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

Submitted to the 2026 Workshop on Generative and Agentic AI for Biology (ICML 2026). Do not distribute.

a multi-objective optimization framework for CRE design that we believe improves the tunability and interpretability of the designed sequences. SPROUT trains three "context" oracles on accessible chromatin regions from the model plant *Arabidopsis thaliana*, predicting the cell-type(s), developmental stage(s), or treatment(s) that the target gene is expressed in, alongside a "strength" oracle on expression readouts of a STARR-seq experiment (Jores et al., 2021) and a "feasibility" oracle trained to differentiate true CREs from fabricated or genic sequences. We pair these oracles with the discrete flow-matching model EditFlows (Havasi et al., 2025), capable of introducing insertions and deletions to DNA sequences and thereby expanding the possible sequence variants beyond simple substitutions, and steer editing using the rollout-guided multi-objective optimization framework introduced in pCoMole (Chen et al., 2026). Under SPROUT, we determine the extent to which CRE strength can be increased while maintaining a fixed expression context, and observe a Pareto-front between these two objectives. We further train a fourth context oracle on a restricted light/dark subset, mirroring the narrow-context design settings of prior work. While optimization with this limited oracle did not induce large off-target expression drift for the tested promoters, optimization with the complete context oracles generally yielded lower drift and more consistent preservation of broader expression context.

2. Problem Setup

We formulate the promoter-editing problem solved by SPROUT as an offline multi-objective optimization over variable-length DNA sequences. Let $\mathcal{T} = \{A, C, G, T\}$ denote the DNA alphabet, and let

$$\mathcal{X} = \bigcup_{n \geq 0} \mathcal{T}^n$$

denote the space of DNA sequences of arbitrary length. Given an initial promoter sequence $x_0 \in \mathcal{X}$, the goal of SPROUT is to generate edited sequences $x \in \mathcal{X}$ that improve predicted expression strength while preserving the broader expression context of the original promoter.

2.1. Edit Flows over DNA Sequences

SPROUT builds on the Edit Flows formulation of Havasi et al. (2025), in which sequence editing is realized as a continuous-time Markov chain (CTMC) on $\mathcal{X}$ with time-inhomogeneous rate field $u_t(\cdot \mid x)$. The CTMC is characterized by

$$\Pr(X_{t+h} = x \mid X_t = x_t) = \delta_{x_t}(x) + h\, u_t(x \mid x_t) + o(h),$$

with valid rates satisfying $u_t(x \mid x_t) \geq 0$ for $x \neq x_t$ and $\sum_{x \in \mathcal{X}} u_t(x \mid x_t) = 0$. The rate field is supported only on sequences that differ from x by a single local edit. For a sequence $x = (x_1, \ldots, x_{n(x)})$ and position i, the admissible operations are insertion $\mathrm{ins}(x, i, a)$, deletion $\mathrm{del}(x, i)$, and substitution $\mathrm{sub}(x, i, a)$ for $a \in \mathcal{T}$, with rates

$$\begin{aligned} u_t(\mathrm{ins}(x,i,a) \mid x) &= \lambda^{\mathrm{ins}}_{t,i}(x)\, Q^{\mathrm{ins}}_{t,i}(a \mid x), \\ u_t(\mathrm{del}(x,i) \mid x) &= \lambda^{\mathrm{del}}_{t,i}(x), \\ u_t(\mathrm{sub}(x,i,a) \mid x) &= \lambda^{\mathrm{sub}}_{t,i}(x)\, Q^{\mathrm{sub}}_{t,i}(a \mid x), \end{aligned}$$

where $\lambda^{\mathrm{ins}}_{t,i}$, $\lambda^{\mathrm{del}}_{t,i}$, and $\lambda^{\mathrm{sub}}_{t,i}$ are non-negative type-specific total rates and $Q^{\mathrm{ins}}_{t,i}, Q^{\mathrm{sub}}_{t,i}$ are normalized token distributions. A pre-trained EditFlows model with rates u_t starting from $X_0 = x_0$ therefore induces a base terminal distribution

$$q(\cdot \mid x_0)$$

over edited variants of x_0 at $t = 1$, rather than generating sequences *de novo*. SPROUT does not sample from this base editor directly. Instead, following the pCoMole perspective (Chen et al., 2026), we bias sampling toward sequences that optimize oracle-defined properties while remaining anchored to the dynamics of the pre-trained editor.

2.2. Oracle-Guided Promoter Editing

For an edited sequence x, let

$$f_{\mathrm{str}}(x)$$

denote the predicted expression strength, and let

$$f_{\mathrm{feas}}(x)$$

denote a feasibility score that distinguishes plausible promoter-like sequences from fabricated or genic negatives. SPROUT additionally trains three context oracles whose outputs are vector-valued predictions over tissue, developmental stage, and treatment labels:

$$c_{\mathrm{tis}}(x), \qquad c_{\mathrm{dev}}(x), \qquad c_{\mathrm{trt}}(x).$$

To quantify context preservation relative to the starting promoter x_0, we measure cosine similarity between the oracle output vectors of x and x_0:

$$\begin{aligned} s_{\mathrm{tis}}(x; x_0) &= \cos\big(c_{\mathrm{tis}}(x), c_{\mathrm{tis}}(x_0)\big), \\ s_{\mathrm{dev}}(x; x_0) &= \cos\big(c_{\mathrm{dev}}(x), c_{\mathrm{dev}}(x_0)\big), \\ s_{\mathrm{trt}}(x; x_0) &= \cos\big(c_{\mathrm{trt}}(x), c_{\mathrm{trt}}(x_0)\big). \end{aligned}$$

These quantities define the multi-objective promoter editing problem that SPROUT solves: we seek sequences that simultaneously improve strength and feasibility while preserving the original expression context. Following pCoMole, we cast this as the search for terminal samples $x_1 \sim q(\cdot \mid x_0)$ with improved objective vector $(f_1(x_1), \ldots, f_K(x_1))$, where each f_k is one of the strength, feasibility, or context-similarity scores defined above. Unlike the full pCoMole formulation, SPROUT does not impose explicit equality or inequality constraints $g_m(x) \leq 0$, $h_\ell(x) = 0$, and instead relies entirely on objective-guided sampling. Intermediate states visited by the editing process are not required to satisfy any feasibility threshold.

2.3. Full-Context Optimization Setting

The primary SPROUT setting considers the full panel of oracles. For a candidate edited promoter x, we define the objective vector

$$F_{\mathrm{full}}(x; x_0) = \Big(f_{\mathrm{str}}(x),\ f_{\mathrm{feas}}(x),\ s_{\mathrm{tis}}(x; x_0),\ s_{\mathrm{dev}}(x; x_0),\ s_{\mathrm{trt}}(x; x_0)\Big).$$

This setting corresponds to the central design problem considered in this work: increasing predicted expression strength while preserving a broad, multi-axis description of expression context. Rather than collapsing these objectives into a single fixed score, SPROUT varies the optimization preferences in order to characterize the trade-off between stronger expression and preserved context.

To summarize optimized sequences, we define the strength gain

$$\Delta_{\mathrm{str}}(x; x_0) = f_{\mathrm{str}}(x) - f_{\mathrm{str}}(x_0),$$

and the mean context shift

$$\Delta_{\mathrm{ctx}}(x; x_0) = 1 - \frac{s_{\mathrm{tis}}(x; x_0) + s_{\mathrm{dev}}(x; x_0) + s_{\mathrm{trt}}(x; x_0)}{3}.$$

We use these two quantities to visualize an empirical Pareto front over strength improvement and context preservation.

2.4. Restricted-Context Evaluation Setting

To mirror prior narrow-context CRE design settings, we also consider a reduced SPROUT regime in which the optimizer is provided only a strength oracle, a feasibility oracle, and a restricted context oracle trained on a light/dark subset. Let

$$s_{\mathrm{ld}}(x; x_0)$$

denote similarity under this reduced light/dark oracle. The corresponding objective vector is

$$F_{\mathrm{restricted}}(x; x_0) = \Big(f_{\mathrm{str}}(x),\ f_{\mathrm{feas}}(x),\ s_{\mathrm{ld}}(x; x_0)\Big).$$

This setting is not intended as a complete model of promoter context. Instead, it serves as a controlled evaluation of whether optimization under a narrow context description is sufficient to preserve broader expression behavior. Sequences generated under this restricted objective are therefore rescored with the full tissue, developmental stage, and treatment oracles, allowing off-target context drift to be measured along axes ignored during optimization.

2.5. Preference-Tilted Terminal Distribution

To bias the base editor toward promoter variants with favorable objective values, SPROUT defines a terminal preference function $G : \mathcal{X} \to \mathbb{R}_{\geq 0}$ constructed from the objective vector $F(x; x_0) \in \{F_{\mathrm{full}}, F_{\mathrm{restricted}}\}$. Following pCoMole, we use an augmented Tchebycheff scalarization

$$U(x; x_0) = \min_k \omega_k\big(f_k(x) - r_k\big) + \rho \sum_{k=1}^{K} \omega_k\big(f_k(x) - r_k\big),$$

where $\omega \in \mathbb{R}^K_{>0}$ is a preference weight vector over the K components of F, $r \in \mathbb{R}^K$ is a reference point, and $\rho > 0$ is an augmentation coefficient. The associated terminal preference is

$$G(x; x_0) = \exp\big(\beta\, U(x; x_0)\big),$$

with inverse temperature $\beta > 0$. Because SPROUT does not impose hard constraints, no feasibility indicator multiplies G. The induced target terminal distribution is the tilted law

$$q_G(x_1 \mid x_0) \ \propto\ q(x_1 \mid x_0)\, G(x_1; x_0),$$

which reweights base EditFlows samples by their multi-objective preference. Varying ω sweeps out distinct points along the strength–context trade-off, allowing the same pre-trained editor to be steered toward different regions of the Pareto front.

2.6. Guided Sampling

In practice, SPROUT adopts the rollout-guided editing strategy introduced in pCoMole, which approximates Doob-h guidance under q_G using short Monte Carlo continuations. The associated harmonic function

$$h_t(x) = \mathbb{E}[G(X_1; x_0) \mid X_t = x]$$

is intractable to compute exactly. Following pCoMole, we estimate it with R short rollouts under the base rates u_t, yielding

$$\widehat{h}_t(x) = \frac{1}{R} \sum_{r=1}^{R} G\Big(X_1^{(r)}(t, x);\ x_0\Big).$$

At each editing step, the pre-trained EditFlows model proposes C candidate edits $\{y_c\}_{c=1}^{C}$ sampled from $u_t(\cdot \mid x_t)$; each candidate is scored by short rollouts $\widehat{h}_t(y_c)$; and the next state is drawn according to the approximate guided rule

$$\widetilde{u}_t(y_c \mid x_t) \ \propto\ u_t(y_c \mid x_t)\, \widehat{h}_t(y_c).$$

In this way, editing remains grounded in the learned sequence prior while being steered toward promoter variants with improved oracle scores. The SPROUT implementation uses objective-guided sampling only; unlike the full pCoMole formulation, we do not impose explicit equality or inequality constraints at generation time, and we do not maintain an incumbent feasible terminal beyond tracking the best preference value observed across rollouts.

3. Methods

The SPROUT pipeline consists of four trained oracles (strength, feasibility, and three multilabel context oracles), an EditFlows base model, and a pCoMole-style rollout-guided sampler that ties them together. We describe each component below.

3.1. ATAC-seq data collection and promoter assignment

To construct training data for the SPROUT context oracles, we collected publicly available *Arabidopsis thaliana* ATAC-seq experiments from AraENCODE (Wang et al., 2023). ATAC-seq measures chromatin accessibility, which is necessary, though not sufficient, for transcription initiation and gene expression. In total, we collected 242 experiments spanning diverse tissues, developmental stages, and growth treatments, and restricted the dataset to non-mutant lines to reduce background confounding.

Open chromatin regions were aligned to the *A. thaliana* reference genome from TAIR (Reiser et al., 2024). We defined a promoter-associated accessible region as any peak whose summit fell within $[-500, +100]$ bp of a transcription start site. Because the SPROUT sequence editor operates on variable-length sequences whereas STARR-seq provides fixed-length promoter fragments of 170 bp, we restricted promoter-associated sequences to lengths between 100 and 500 bp to limit distribution shift between context-oracle and strength-oracle inputs.

Metadata were harmonized into three multilabel context categories: tissue, developmental stage, and treatment. Because source annotations mixed levels of specificity, labels were normalized to broad biological categories. Tissue labels were mapped to an organ-level ontology; developmental stages were grouped into “seedling”, “vegetative”, and “reproductive”; and treatment labels were normalized to broad perturbation classes, separating biological perturbations from controls and assay-specific metadata. Each promoter-associated sequence was labeled by the set of tissues, developmental stages, and treatments in which it was observed accessible.

3.2. STARR-seq data collection and strength labels

While ATAC-seq identifies accessible chromatin, it does not quantify how strongly a promoter fragment drives expression. We therefore used publicly available *Arabidopsis* STARR-seq data from Jores et al. (Jores et al., 2021), in which promoter fragments were assayed under light and dark conditions. The STARR-seq readout served as the target for the SPROUT strength oracle.

Following the original study, we trained the primary strength oracle to predict expression under dark conditions. We chose this setting because the authors likewise reported a CNN trained on dark-condition values, and this regime trained stably in our setting. The resulting oracle was used as the main quantitative objective during SPROUT optimization.

3.3. Feasibility-oracle dataset construction

To guide SPROUT toward plausible promoter-like sequences, we trained a feasibility oracle to distinguish true promoter-associated sequences from fabricated or non-promoter negatives. Positive examples were the ATAC-derived promoter sequences described above. For each positive sequence, we generated one negative example from each of the following classes: (i) a dinucleotide-preserving shuffle of the promoter, (ii) a non-promoter genomic fragment drawn from genic or intergenic background regions, and (iii) a random sequence matched for length and single-nucleotide composition. This yielded a balanced negative set spanning both realistic genomic background and clearly implausible synthetic controls. The feasibility task was framed as binary classification.

3.4. Oracle architectures and training

We compared several neural architectures for the context and strength oracles, including multilayer perceptrons, transformers, standard convolutional neural networks (CNNs), and reverse-complement-aware CNNs (“Jores et al. CNN”). Input sequences were represented either with frozen embeddings from the Genomic Pre-trained Network (GPN) (`songlab/gpn-brassicales`) (Benegas et al., 2023) or, where supported, simple one-hot encodings. After observing consistently stronger performance from CNN-based models, we considered only these architectures for the feasibility oracle. For all tasks, train/validation/test splits were performed by chromosome to reduce sequence leakage across related genomic regions, following recent work (Benegas et al., 2023).

The tissue, developmental-stage, and treatment tasks were framed as multilabel classification problems; the strength task as regression on STARR-seq expression values; and the feasibility task as binary classification. Final models were selected by validation performance, using macro F1 for the context oracles, Spearman correlation for the strength oracle, and AUPRC for the feasibility oracle. The strongest-performing CNN variants were then used as oracles in downstream SPROUT optimization experiments.

3.5. EditFlows base model

The SPROUT sequence editor is based on EditFlows (Havasi et al., 2025), a discrete flow-matching model defined over variable-length biological sequences. EditFlows specifies a continuous-time Markov chain on $\mathcal{X}$ with a learned rate field $u_t^\theta(\cdot \mid x)$, parameterized so that all probability mass is placed on single-edit transitions corresponding to insertion, deletion, or substitution operations. For a sequence $x = (x_1, \dots, x_{n(x)})$, each edit has a type-specific total rate and, where applicable, a token distribution:

$$\begin{aligned} u_t^\theta(\mathrm{ins}(x,i,a) \mid x) &= \lambda_{t,i}^{\mathrm{ins}}(x)\, Q_{t,i}^{\mathrm{ins}}(a \mid x), \\ u_t^\theta(\mathrm{del}(x,i) \mid x) &= \lambda_{t,i}^{\mathrm{del}}(x), \\ u_t^\theta(\mathrm{sub}(x,i,a) \mid x) &= \lambda_{t,i}^{\mathrm{sub}}(x)\, Q_{t,i}^{\mathrm{sub}}(a \mid x), \end{aligned}$$

with the diagonal term fixed by the rate conditions. Training follows the alignment-augmented objective of Havasi et al. (2025): pairs (x_0, x_1) are augmented with token-wise alignments (z_0, z_1) over an extended alphabet that includes a blank symbol, intermediate alignments z_t are sampled along a mixture path with schedule κ_t, and the rate field is supervised on the projected sequence x_t according to

$$\mathcal{L}(\theta) = \mathbb{E}\left[\sum_{x \neq x_t} u_t^\theta(x \mid x_t) - w_t \sum_{i \in \Delta_t} \log u_t^\theta(x_t^{(i)} \mid x_t)\right],$$

where $w_t = \dot{\kappa}_t/(1-\kappa_t)$, Δ_t indexes mismatched alignment positions, and $x_t^{(i)}$ is the projected sequence after applying the alignment-implied edit at position i. This formulation permits direct editing of an incumbent promoter sequence rather than requiring *de novo* generation from a fixed-length latent representation.

To assess whether EditFlows captured promoter-like structure beyond trivial sequence statistics, we compared unconditional EditFlows generations against a random-sequence baseline. Evaluation included GC-related composition and feasibility-oracle score.

3.6. SPROUT sampling: rollout-guided multi-objective editing

The SPROUT sampler combines the EditFlows base model with the rollout-guided objective steering of pCoMole (Chen et al., 2026), which biases the pre-trained edit process toward sequences with improved downstream objective values while remaining grounded in the learned editing dynamics. Given an objective vector $F(x; x_0)$ assembled from the SPROUT strength, feasibility, and context-similarity oracles introduced in the Problem Setup, we construct a terminal preference using the augmented Tchebycheff scalarization

$$U(x; x_0) = \min_k \omega_k\big(f_k(x) - r_k\big) + \rho \sum_{k=1}^{K} \omega_k\big(f_k(x) - r_k\big),$$

with preference weights $\omega \in \mathbb{R}_{>0}^K$, reference point $r \in \mathbb{R}^K$, and augmentation coefficient $\rho > 0$. The terminal preference is

$$G(x; x_0) = \exp\big(\beta\, U(x; x_0)\big),$$

with inverse temperature $\beta > 0$, and the corresponding target distribution is the tilted law $q_G(x_1 \mid x_0) \propto q(x_1 \mid x_0)\, G(x_1; x_0)$. The augmented Tchebycheff form is strictly monotone with respect to Pareto dominance whenever $\omega > 0$, so sweeping ω traces out distinct points along the strength–context trade-off rather than collapsing onto a single scalar criterion. In SPROUT, generation is guided only through these objective functions; we do not impose explicit equality or inequality constraints at generation time.

To realize the tilted distribution while preserving the local edit structure of the base process, SPROUT applies a Doob-h transform with harmonic function $h_t(x) = \mathbb{E}[G(X_1; x_0) \mid X_t = x]$, giving guided rates $u_t^G(y \mid x) = u_t(y \mid x)\, h_t(y)/h_t(x)$ for $y \neq x$. Because h_t is intractable in the EditFlows state space, we estimate it with short Monte Carlo rollouts under the base rates: at each editing step (t, x_t), we sample C candidate next states $\{y_c\}_{c=1}^C$ from $u_t(\cdot \mid x_t)$, simulate R base-rate continuations from each candidate to terminal time, and form

$$\widehat{h}_t(y_c) = \frac{1}{R}\sum_{r=1}^{R} G\Big(X_1^{(c,r)}; x_0\Big).$$

The next state is then drawn from the candidate set according to the approximate guided rule

$$\widetilde{u}_t(y_c \mid x_t) \;\propto\; u_t(y_c \mid x_t)\, \widehat{h}_t(y_c),$$

in which u_t proposes locally plausible edits and $\widehat{h}_t$ favors candidates whose downstream continuations achieve higher terminal preference. SPROUT follows pCoMole in pruning candidates whose immediate scalarized score does not improve upon the current state before allocating rollouts, and in tracking the best terminal preference observed across the entire sampling process. EditFlows and the underlying pCoMole guidance framework are described in more detail in their respective manuscripts (Havasi et al., 2025; Chen et al., 2026).

3.7. Full-context SPROUT optimization

The primary SPROUT setting considered the joint objectives of increasing promoter strength and feasibility while preserving the broader expression context of the starting promoter. Specifically, we optimized promoter sequences with respect to the strength oracle, the feasibility oracle, and the three context oracles for tissue, developmental stage, and treatment, instantiating $F(x; x_0) = F_{\mathrm{full}}(x; x_0)$ in the

SPROUT preference G. Promoters for this study were selected from chromosome 5, which was held out during oracle training. Candidate promoters were required to appear in both the ATAC-seq and STARR-seq datasets, to have treatment labels consistent with the target condition, and to have low but measurable STARR-seq expression so that there remained room for improvement.

To evaluate the importance of full-context guidance, we compared this setting against a reduced SPROUT objective that optimized only strength and feasibility. For each starting promoter, we additionally swept the preference weights ω in order to trace different points along the strength–context trade-off rather than committing to a single scalarization. Generated sequences from both settings were rescored with the full oracle panel, and their predicted strength gain and mean context shift from the starting promoter were used to trace an empirical Pareto trade-off between increased expression and preserved expression context.

3.8. Restricted-context SPROUT evaluation

To evaluate whether narrow-context guidance is sufficient to preserve broader promoter behavior, we introduced a restricted-context SPROUT setting based only on light and dark conditions. In this setting, we trained an additional context oracle on a light/dark subset of the data and optimized sequences with $F(x;x_0) = F_{\text{restricted}}(x;x_0)$, substituting the light/dark similarity s_{ld} for the full tissue, developmental-stage, and treatment context oracles inside the SPROUT preference. Starting promoters were again taken from held-out chromosome 5 and were required to be represented in both the ATAC-seq and STARR-seq datasets, to have STARR-seq measurements in both light and dark, to show a substantial difference between light and dark expression, and to be predicted confidently by the restricted oracle.

Sequences generated under this restricted objective were then rescored with the full tissue, developmental-stage, and treatment oracles in order to quantify off-target drift in the broader expression context. In this way, the restricted-context setting serves as a direct comparison to narrow-context CRE design, while the full-context setting serves as the SPROUT formulation we propose.

4. Results

4.1. Generative model and oracle training

We first evaluated whether the EditFlows base model used by SPROUT learned promoter-like sequence structure beyond simple nucleotide composition. Relative to a random generation baseline, EditFlows produced sequences with substantially higher feasibility-oracle scores while maintaining realistic G/C composition (Table 1). Because random

Table 1. Comparison of random generation and EditFlows-generated sequences. Higher feasibility is better, while the G/C ratio should remain close to 1.

Method	G/C Ratio	Feasibility
Random	1.023	0.008
EditFlows	**0.994**	**0.136**

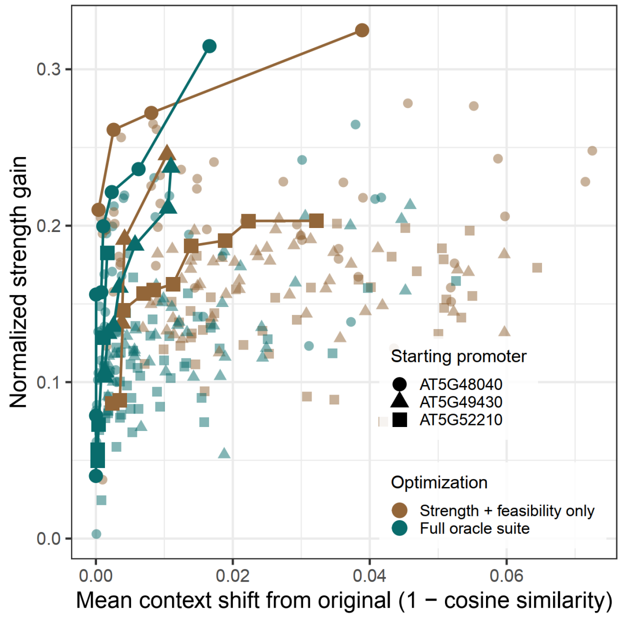

Figure 1. **Full-context promoter optimization with SPROUT for three held-out promoters.** Each point denotes an optimized sequence. Larger points and connecting lines denote the empirical Pareto frontier for each optimization setting.

sequences can remain compositionally plausible while failing to resemble true promoters, we interpret this result as a minimal sanity check that the learned editor captures non-trivial promoter-associated structure. We include random generation as a baseline, although trivial, as it remains commonly used in works on plant CRE generation (Li et al., 2024; Deng et al., 2025).

We next evaluated the oracle models used for downstream SPROUT optimization. Across the tissue, developmental-stage, and treatment tasks, all oracle families achieved strong performance, with CNN-based models consistently outperforming MLP and transformer baselines (Table 2) – all oracle results are presented in Appendix A. In addition, frozen GPN embeddings generally improved performance relative to one-hot inputs. This observable and consistent performance improvements is in contrast to recent works demonstrating genomic language models achieve no improvements over simple baselines (Patel et al., 2024). Among the context models, the Jores et al. CNN gave the strongest overall results, achieving the best or tied-best GPN performance on all three tasks. The strength oracle was more challenging than the classification tasks, but CNN-based models again performed best, with the standard CNN achieving the strongest overall regression performance under GPN embeddings (Table 3). The feasibility oracle likewise achieved strong classification performance (Table 4), indicating that true promoter-associated sequences can be distinguished reliably from shuffled, random, and genomic background negatives.

Oracle performance in our setting was substantially stronger than that reported in some prior guided DNA-generation studies (Davis et al., 2024), although such comparisons should be interpreted cautiously because the underlying tasks differ in assay modality, label-space size, and prediction formulation. One plausible explanation is that our context tasks overall involve fewer classes. Taken together, these results suggest that both the EditFlows editor and the learned oracle panel are sufficiently trained to support downstream promoter optimization with SPROUT.

4.2. SPROUT reveals a trade-off between strength and context preservation under full-context guidance

We next evaluated whether SPROUT can increase promoter strength while preserving the broader expression context of the starting sequence. For three held-out promoters, we compared SPROUT optimization under the full oracle suite against a reduced setting using only the strength and feasibility oracles (Figure 1). In all three cases, both optimization settings produced variants with improved predicted strength, but they differed markedly in the amount of context shift induced. Sequences generated without context guidance more frequently achieved high strength gains at the cost of larger changes in the tissue-, developmental-stage-, and treatment-oracle outputs, whereas the full SPROUT oracle suite concentrated samples near low context shift while still preserving substantial room for strength improvement. This produced an empirical Pareto front between normalized strength gain and context preservation, suggesting that increased promoter activity can be achieved without fully discarding the original expression context, but that stronger gains are often associated with larger context changes when broader context is left unconstrained.

4.3. Restricted-context SPROUT guidance only partially preserves broader expression context

We next asked whether SPROUT requires the full context oracle suite, or whether a narrow context description is sufficient to preserve broader promoter behavior. To do so, we optimized three held-out promoters using either the full SPROUT oracle suite or a restricted objective consisting only of the strength oracle, feasibility oracle, and a light/dark context oracle, then rescored generated sequences with the full context panel (Figure 2). Off-target drift remained modest in both settings, indicating that light/dark optimization alone does not necessarily induce large changes in broader expression context. However, the full SPROUT oracle suite generally produced lower drift and a tighter distribution of outcomes across promoters. This effect was promoter-dependent and was most apparent for the sequences with higher drift under the restricted setting. Taken together, these results suggest that narrow-context optimization can provide only partial control over promoter behavior: for some promoters, restricted guidance remains close to the original context manifold, whereas for others, broader context preservation benefits from explicitly modeling additional expression axes within SPROUT.

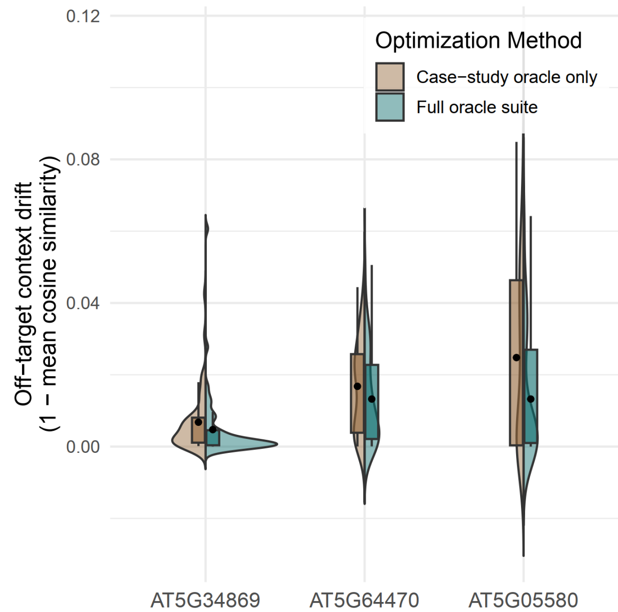

Figure 2. **Restricted-context SPROUT evaluation for three held-out promoters.** Promoters were optimized with SPROUT under either the full oracle suite or a restricted light/dark objective, then rescored with the full context panel to quantify off-target drift. Split violin plots show the distribution of drift values, with black points marking group means.

5. Discussion and Limitations

SPROUT is, to our knowledge, the first framework to formulate plant promoter optimization as a multi-objective sequence-editing problem. Rather than treating promoter design as the search for stronger expression alone, or as enrichment in one narrowly defined context over another, SPROUT makes it possible to study the trade-off between expression strength and a broader description of expression context. In this setting, promoter optimization becomes more interpretable: one can directly examine whether increasing predicted strength requires context drift, or whether some promoters admit improvements while largely preserving their original tissue, developmental-stage, and treatment profiles. More broadly, this formulation opens the possibility of studying which context axes are easier to modulate, which are more tightly coupled, and whether some context transitions are systematically more accessible than others.

We believe SPROUT offers a step toward more transparent and controllable CRE engineering. If sufficiently accurate oracles can be trained, the same formulation could in principle be used not only to preserve context while strengthening a promoter, but also to deliberately modify where, when, and under which conditions a promoter is active. In the longer term, this may enable the design of promoters with expression patterns not observed in nature. Although such applications remain far beyond the scope of the present study, our results provide a proof of concept that multi-objective, oracle-guided sequence editing is a viable framework for plant promoter design.

SPROUT also has several limitations. First, we did not benchmark against existing DNA diffusion or flow-matching methods, and therefore cannot yet claim superiority over contemporary generative baselines beyond the random-generation sanity check included here. Second, oracle quality remains a central bottleneck for SPROUT. Although the oracle results were strong in our setting, alternative training strategies, including contrastive or representation-learning approaches, may yield better context predictors. Third, all models are limited by the quantity and diversity of available plant regulatory data. Public plant ATAC-seq and STARR-seq resources remain much smaller and less standardized than analogous datasets in mammalian systems, which constrains both model scale and label richness. Finally, all results presented here are computational. The generated promoters were not experimentally validated, and the biological meaning of oracle-predicted context similarity therefore remains provisional. Expanding available plant regulatory datasets, exploring stronger oracle-training strategies, and validating designed promoters experimentally will all be important next steps for advancing SPROUT toward experimental deployment.

Impact Statement

This work presents a framework for cis-regulatory element design. We envision and are optimistic that it may support applications such as engineering precise control of agricultural traits, and future bioremediation efforts. It also offers an alternative formulation to CRE design that may help researchers study gene regulation across all life forms. However, methods for biological sequence design can carry risks of misuse – risks which grow as the accuracy and scope of these methods increase. Advances in generative design could eventually contribute to the creation of biological sequences with harmful societal, agricultural, or ecological effects if deployed irresponsibly. We therefore believe that progress in biological sequence design should be accompanied by careful oversight, clear and widely agreed-upon usage policies, and experimental validation practices that consider biosafety at every stage.

References

Andersson, R. and Sandelin, A. Determinants of enhancer and promoter activities of regulatory elements. *Nature Reviews Genetics*, 21(2):71–87, 2020.

Benegas, G., Batra, S. S., and Song, Y. S. Dna language models are powerful predictors of genome-wide variant effects. *Proceedings of the National Academy of Sciences*, 120(44):e2311219120, 2023.

Chen, K. M., Wong, A. K., Troyanskaya, O. G., and Zhou, J. A sequence-based global map of regulatory activity for deciphering human genetics. *Nature genetics*, 54(7): 940–949, 2022.

Chen, T., Holsman, M., Zhao, L., Zhang, Y., and Chatterjee, P. pcomole: Pareto-constrained molecule editing with discrete flows. In *ICLR 2026 2nd Workshop on Deep Generative Model in Machine Learning: Theory, Principle and Efficacy*, 2026. URL https://openreview.net/forum?id=tTILzscPs4.

DaSilva, L. F., Senan, S., Kribelbauer-Swietek, J. F., Patel, Z. M., Louis, L. K., Reddy, A. J., Gabbita, S., Rosen, J. D., Nussbaum, Z., Córdova, C. M. V., et al. Designing synthetic regulatory elements using the generative ai framework dna-diffusion. *Nature Genetics*, 58(1):180–194, 2026.

Davis, O., Kessler, S., Petrache, M., Ceylan, İ. İ., Bronstein, M., and Bose, A. J. Fisher flow matching for generative modeling over discrete data. *Advances in Neural Information Processing Systems*, 37:139054–139084, 2024.

de Almeida, B. P., Schaub, C., Pagani, M., Secchia, S., Furlong, E. E., and Stark, A. Targeted design of synthetic enhancers for selected tissues in the drosophila embryo. *Nature*, 626(7997):207–211, 2024.

Deng, Y., Zhao, W., Xiong, Y., Naeem, M., Lu, S., Zhou, X., Zhao, L., and Zhang, L. Decoding tissue-specific enhancers in plants using massively parallel assays and deep learning. *The Plant Cell*, 37(10):koaf236, 2025.

Havasi, M., Karrer, B., Gat, I., and Chen, R. T. Edit flows: Flow matching with edit operations. *arXiv preprint arXiv:2506.09018*, 2025.

Jores, T., Tonnies, J., Wrightsman, T., Buckler, E. S., Cuperus, J. T., Fields, S., and Queitsch, C. Synthetic promoter designs enabled by a comprehensive analysis of plant core promoters. *Nature Plants*, 7(6):842–855, 2021.

Jores, T., Mueth, N. A., Gorjifard, S., Triesch, S., Schirmer, D., Tonnies, J., Bubb, K. L., Cuperus, J. T., Fields, S., and Queitsch, C. Massively parallel characterization and deep learning of enhancers in plant genomes. *bioRxiv*, pp. 2026–04, 2026.

Li, T., Xu, H., Teng, S., Suo, M., Bahitwa, R., Xu, M., Qian, Y., Ramstein, G. P., Song, B., Buckler, E. S., et al. Modeling 0.6 million genes for the rational design of functional cis-regulatory variants and de novo design of cis-regulatory sequences. *Proceedings of the National Academy of Sciences*, 121(26):e2319811121, 2024.

Patel, A., Singhal, A., Wang, A., Pampari, A., Kasowski, M., and Kundaje, A. Dart-eval: A comprehensive dna language model evaluation benchmark on regulatory dna. *Advances in Neural Information Processing Systems*, 37:62024–62061, 2024.

Reiser, L., Bakker, E., Subramaniam, S., Chen, X., Sawant, S., Khosa, K., Prithvi, T., and Berardini, T. Z. The arabidopsis information resource in 2024. *Genetics*, 227(1):iyae027, 2024.

Sarkar, A., Duran, A., Yu, Y., Lin, D.-W., Kang, Y., Somia, N., Mantilla, P., Zhou, J., Nagai, M., Tang, Z., et al. Designing dna with tunable regulatory activity using discrete diffusion. *bioRxiv*, pp. 2024–05, 2024.

Stark, H., Jing, B., Wang, C., Corso, G., Berger, B., Barzilay, R., and Jaakkola, T. Dirichlet flow matching with applications to dna sequence design. *arXiv preprint arXiv:2402.05841*, 2024.

Taskiran, I. I., Spanier, K. I., Dickmänken, H., Kempynck, N., Pančíková, A., Ekşi, E. C., Hulselmans, G., Ismail, J. N., Theunis, K., Vandepoel, R., et al. Cell-type-directed design of synthetic enhancers. *Nature*, 626(7997):212–220, 2024.

Wang, Z., Liu, M., Lai, F., Fu, Q., Xie, L., Fang, Y., Zhou, Q., and Li, G. Araencode: a comprehensive epigenomic database of arabidopsis thaliana. *Molecular Plant*, 16(7):1113–1116, 2023.

Yang, Z., Liu, H., Cao, C., and Su, B. D3lm: A discrete dna diffusion language model for bidirectional dna understanding and generation. *arXiv preprint arXiv:2603.01780*, 2026.

A. Appendix A: Oracle Results

Tables 2–4 detail outcomes of oracle training.

Table 2. Context-oracle performance across tasks, model architectures, and input embeddings. Higher macro and micro F1 are better, while lower hamming loss is better. Embedding types are labeled as OH = One-Hot, GPN = Genomic Pre-trained Network.

Task	Model	Macro F1		Micro F1		Hamming Loss	
		OH	GPN	OH	GPN	OH	GPN
Tissue	MLP	0.548	0.803	0.691	0.820	0.323	0.195
Tissue	CNN	0.795	0.810	0.815	0.828	0.194	**0.183**
Tissue	Transformer	0.778	0.796	0.786	0.813	0.236	0.204
Tissue	Jores et al. CNN	0.775	**0.814**	0.810	**0.829**	0.198	**0.183**
Dev. stage	MLP	0.694	0.824	0.734	0.839	0.320	0.191
Dev. stage	CNN	0.830	0.836	0.841	0.849	0.190	0.179
Dev. stage	Transformer	0.822	0.827	0.831	0.840	0.205	0.189
Dev. stage	Jores et al. CNN	0.804	**0.840**	0.822	**0.852**	0.203	**0.175**
Treatment	MLP	0.417	0.714	0.617	0.784	0.305	0.187
Treatment	CNN	0.690	0.728	0.785	0.797	0.186	0.176
Treatment	Transformer	0.651	0.676	0.776	0.778	0.207	0.185
Treatment	Jores et al. CNN	0.685	**0.738**	0.785	**0.802**	0.187	**0.169**

Table 3. Strength oracle performance across model architectures and input embeddings.

	Spearman ρ		RMSE		MAE		R^2	
Model	One-hot	GPN	One-hot	GPN	One-hot	GPN	One-hot	GPN
MLP	0.313	0.542	1.126	0.967	0.912	0.771	0.110	0.343
CNN	0.696	**0.719**	0.836	**0.800**	0.655	**0.624**	0.510	**0.551**
Transformer	0.329	0.644	1.121	0.874	0.913	0.689	0.118	0.465
Jores et al. CNN	0.663	0.704	0.884	0.815	0.697	0.638	0.452	0.534

Table 4. Feasibility oracle performance across model architectures and input embeddings.

	Macro F1		Weighted F1		Balanced Accuracy	
Model	One-hot	GPN	One-hot	GPN	One-hot	GPN
CNN	0.685	**0.772**	0.838	**0.883**	0.795	**0.856**
Jores et al. CNN	0.695	0.757	0.824	0.876	0.824	0.844