L2L-GP: GRADIENT-GUIDED PLANNING OVER DYNAMICALLY REVEALED CRISPR-FEASIBLE EDIT GRAPHS

Anonymous authors
Paper under double-blind review

ABSTRACT

Every intended genome edit changes the sequence that defines which subsequent edits are executable. On RegEdit, a frozen set of 50 regulatory tasks, the median Jaccard overlap between consecutive action sets is 0.10; in 80% of tasks the trajectory selects a later action that did not exist at the root—one that a root-defined program could not have named.

We formulate this as planning with state-dependent executable action identities: a root-fixed program retains the original candidate list, while a dynamic program regenerates the executable set after every transition. In the native comparison the dynamic regime produces a higher model-predicted endpoint on all 50 tasks—yet under exposure-matched conditions the difference vanishes (bootstrap 95% CI crossing zero). The action space changes dramatically; the current search does not exploit them. The gap between reachable and reached defines the open problem: converting action-space turnover into utility. Within this regime, L2L-GP achieves the highest median gain at every tested native query scale.

1 INTRODUCTION

A natural approach to multi-edit regulatory design is to enumerate candidate edits at the native sequence, score them, and select a program. Sequence-to-function predictors supply the scoring (Avsec et al., 2021; Linder et al., 2025; Avsec et al., 2026); CRISPR design systems supply candidate enumeration under editor-, guide-, PAM-, and window-specific constraints (Concordet & Haeussler, 2018; Labun et al., 2019). For a single edit, root-level enumeration is sufficient to define the decision. For a multi-edit program, however, the first edit changes the sequence and coordinate map that determine which subsequent executable identities can be represented.

Consider a concrete example: a sequence insertion followed by a downstream edit. The insertion shifts downstream coordinates, can invalidate guides whose protospacers or PAMs cross the edited interval, and can expose sequence contexts that were absent at the root. We call the full specification of an executable edit—guide, PAM, editor window or cut site, protected-region checks, and coordinate replay—an *action identity*. A root-defined program cannot represent an identity materialized only after a transition; a dynamic program regenerates the executable action set in the post-edit state.

The central question is not *whether* the action space changes after an edit—on RegEdit the median Jaccard overlap between consecutive sets is 0.10—but whether the change matters for utility. We compare a root-fixed regime that retains original identities against a dynamic regime that regenerates them. In the native comparison the dynamic regime produces a higher model-predicted endpoint on all 50 tasks. Under exposure-matched conditions, however, the paired difference vanishes. The action space turns over dramatically, but the current search does not exploit the newly available identities.

This gap between what is reachable and what is reached motivates the present work. Our contributions:

- We formulate native-locus regulatory intervention as planning with state-dependent executable action identities, making the root-fixed, FrozenRoot, and Dynamic sequential regimes explicit.

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.

- On the RegEdit task set, we show that action-set turnover is large (median Jaccard 0.10, 80% root-absent selection) while the matched-exposure utility difference is indistinguishable from zero.
- We use L2L-GP versus strict No-Ascent as a one-factor proposal comparison inside the Dynamic regime. The comparison separates proposal-layer behavior from regeneration and reports descriptive operating-point variation rather than a universal optimizer ranking.

Code and accompanying materials for the anonymous submission are available at `https://anonymous.4open.science/r/anonymous-iclr-code-2026-0926-9C02/`.

2 Related Work

Sequence-to-function prediction and CRISPR design. Long-context models such as Enformer, Borzoi, and AlphaGenome provide frozen evaluators for counterfactual sequences (Avsec et al., 2021; Linder et al., 2025; Avsec et al., 2026). Guide-design systems encode PAM, editor, and activity constraints when ranking candidates (Concordet & Haeussler, 2018; Labun et al., 2019; Chuai et al., 2018; Fan et al., 2023). We use the former as a frozen endpoint and the latter to motivate our action identity contract.

Multiplex editing and sequence design. CRISPR systems support multiplex editing and programmed guide activation (Cong et al., 2013; Clarke et al., 2021). Differentiable sequence optimization and active combinatorial search propose high-value changes or combinations (de Almeida et al., 2022; Qin et al., 2025). Our setting differs: the executable action identity itself is state-dependent, with mixed edit families, ordered transitions, and explicit revalidation after each edit.

Planning with state-dependent actions. State-dependent action spaces arise in robot manipulation (Kroemer et al., 2021), molecular optimization (Zhou et al., 2019), and combinatorial optimization (Bengio et al., 2021). Regulatory editing exhibits a direct form: a sequence change can invalidate a guide/PAM contract and reveal identities absent in the prior state. Our formulation makes this explicit; L2L-GP occupies the proposal layer within it.

3 Problem Formulation

We model native-locus intervention as planning with state-dependent executable action identities. This formulation captures the observation that every intended edit changes the sequence, and the sequence determines which subsequent actions exist.

An ordered intervention is a sequence of intended computational edits applied to a native-locus sequence. The state at depth t is

$$s_t = (x_t, h_t, m_t, b_t),$$

where x_t is the current sequence, h_t is the intended-edit history, m_t is the coordinate map induced by that history, and b_t is the remaining logical-call budget.

The state-dependent action generator returns

$$\mathcal{A}(s_t) = G(x_t, m_t, h_t).$$

An action identity is sequence-conditioned rather than coordinate-only. Its targeting fields specify the edit family, applicable guide and PAM evidence, editor window or cut site, and protected-region checks. The record also stores the edit payload and coordinate replay required at its parent state. An action is legal when it passes this implemented in-silico targeting and replay contract in the current state.

Applying an action produces an intended-edit transition

$$s_{t+1} = T(s_t, a_t), \qquad a_t \in \mathcal{A}(s_t).$$

The transition updates the sequence and coordinate map and therefore can change the next action set. It is an intended-edit replay operator that constructs the specified sequence state for computational comparison.

We distinguish three computational regimes. A single-step or StaticRoot program selects one action from the root set $\mathcal{A}(s_0)$. A FrozenRoot program permits multiple intended edits, but restricts every later action to an identity in the root snapshot and revalidates that identity through the current-state replay and legality checks:

$$\mathcal{A}_{\mathrm{FR}}(s_t) = \{a \in \mathcal{A}(s_0) : \mathrm{LegalReplay}(a, s_t) = 1\}.$$

A Dynamic sequential program regenerates $\mathcal{A}(s_t)$ after every transition. FrozenRoot is therefore a multi-edit identity-freezing regime, whereas Dynamic sequential is the regime that can represent an action identity revealed by an earlier edit.

Given a frozen model-predicted endpoint $U(x_H)$, the query-budgeted objective is

$$\max_{\tau=(a_0,\ldots,a_{H-1})} U(x_H) \quad \text{subject to} \quad a_t \in \mathcal{A}(s_t), \qquad \sum_{t=0}^{H-1} q_t \le B.$$

Here q_t counts logical endpoint requests under the declared accounting contract. Logical requests and physical model calls are recorded separately because caching can reduce physical execution without reducing the number of requests made by the controller.

4 METHOD

Section 3 defines the planning regimes by when executable action identities are determined. The method separates the mechanism that exposes the current action space from the proposal mechanism that searches within it. Dynamic regeneration is the problem-level operation; L2L-GP is one proposal mechanism inside the Dynamic regime.

4.1 DYNAMIC ACTION REGENERATION

At a visited state s_t, the generator materializes the current executable records and forms a controlled candidate pool. A controller proposes candidate actions from this pool, materializes their intended sequence states, and evaluates them with the frozen endpoint under the declared frontier, lookahead, and query budget. The selected intended edit is then committed, updating the sequence and coordinate map; the sequence-conditioned action geometry is reconstructed for s_{t+1}. Thus regeneration occurs after every selected intended edit rather than once at the root.

4.2 LATENT-TO-LEGAL GRADIENT PROPOSAL

L2L-GP navigates the regenerated action space through a continuous waypoint and a discrete legal projection. At state s_t, it starts from a controlled candidate pool

$$\mathcal{C}_K(s_t) \subseteq \mathcal{A}(s_t).$$

Each candidate action a is encoded as an 18-dimensional representation $e_t(a) \in \mathbb{R}^{18}$. The representation combines edit-mechanism indicators, position and span, guide/PAM composition, targeting-window features, an efficiency prior, and edit burden; coordinate definitions remain in Appendix A.2. A deterministic state encoder supplies $z_t = h(s_t)$.

The training-free proposal proxy defines a structured field over this engineered action space:

$$\widetilde{J}(z_t, u) = J_{\mathrm{proxy}}(z_t, u) - \lambda_{\mathrm{valid}} \min_{a \in \mathcal{C}_K(s_t)} \|u - e_t(a)\|_2^2 - \lambda_{\mathrm{norm}} \|u\|_2^2.$$

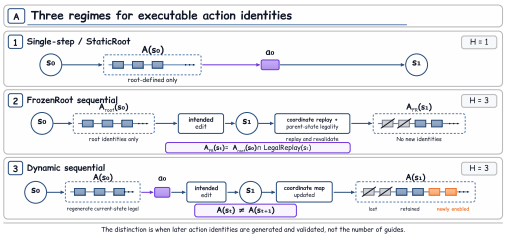
The first term scores the proposal in the proxy space. The validity term keeps the waypoint near the current legal candidate geometry, while the norm term regularizes the waypoint. The proxy proposes directions; it is not the AlphaGenome endpoint evaluator.

The continuous waypoint is initialized from the highest-valued proxy candidate and updated by gradient ascent,

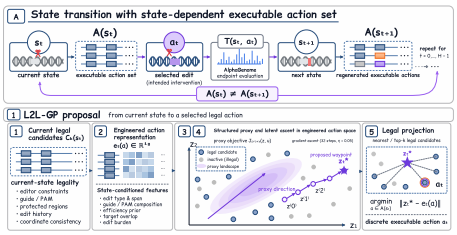
$$u^{(j+1)} = u^{(j)} + \eta \nabla_u \widetilde{J}(z_t, u^{(j)}).$$

The implementation uses 32 ascent steps, learning rate 0.05, and $\lambda_{\mathrm{valid}} = 0.1$. After ascent, the waypoint is projected back to current executable actions:

$$\mathcal{P}_K(u_t^*) = \mathrm{TopK}_{a \in \mathcal{C}_K(s_t)} \left[-\|u_t^* - e_t(a)\|_2^2 \right].$$



(a) Three executable intervention regimes



(b) L2L-GP latent-to-legal proposal pipeline

Figure 1: Overview of state-dependent executable action planning. (a) StaticRoot, FrozenRoot, and Dynamic sequential regimes differ in whether later action identities remain root-defined or are regenerated after each intended transition. (b) L2L-GP encodes current legal candidates, performs latent ascent in the engineered action space, and projects the waypoint back to executable candidates.

The projected legal actions are materialized as intended sequence states and evaluated with the frozen AlphaGenome endpoint. The controller selects an action under the declared frontier and tie-break, commits the intended transition, and regenerates $\mathcal{A}(s_{t+1})$. Proposal and endpoint evaluation therefore remain separate: L2L-GP shapes candidate exposure, while AlphaGenome evaluates the materialized state.

No-Ascent attribution control. Strict No-Ascent preserves the initialization, action representation, legal candidate exposure, projection, frontier, lookahead, endpoint evaluator, tie-breaking, and logical/physical call accounting. It changes only the number of ascent steps, which is set to zero. This makes the comparison a one-factor attribution control for the latent ascent operator. A direct proxy-ranking controller changes the proposal rule itself and answers a different question, so it is not a substitute for No-Ascent.

5 Experimental Setup

5.1 RegEdit benchmark and computational scope

RegEdit is a frozen set of 50 native-locus regulatory intervention tasks. The current primary tasks share the INCREASE direction, a deterministic intended-edit transition contract, and a frozen AlphaGenome model-predicted endpoint. The statistical unit is the task. Methods are joined by exact task identity, and state, action, and endpoint rows remain subordinate records rather than independent samples.

The primary endpoint is the absolute target-transcript gene-body change,

$$\Delta_{\mathrm{gb}} = F_{\mathrm{gb}}(x_H) - F_{\mathrm{gb}}(x_0).$$

Relative gene-body change, TSS and exon tracks, and whole-track summaries are secondary views. The endpoint is a model prediction for the materialized sequence state. The action contract labels records as computationally legal or model-supported candidates. Experimental validation is outside this computational labeling scheme.

5.2 Part I comparison contracts

The core comparison uses a three-edit horizon and a shared $K = 128$, $B = 384$ logical-call contract for Dynamic Greedy, FrozenRoot, L2L-GP, and strict No-Ascent. StaticRootBudget is a root-only reference with $H = 1$, $K = 384$, and a maximum of 384 logical requests. The root pool exceeds the operating budget, so StaticRootBudget samples a root subset rather than exhausting it.

The main regeneration comparison is exposure-matched Dynamic Greedy versus FrozenRoot. The intended control is to match candidate exposure at every task and depth while changing whether the candidate identities are regenerated from the current state or retained from the root snapshot. The exact per-task exposure-matching schedule will be specified with the formal receipts. Native no-padding FrozenRoot is a complementary system condition that preserves the natural survivor

Table 1: Comparison contracts. All methods share the RegEdit task set, deterministic intended-edit replay, and frozen AlphaGenome endpoint. Each reported comparison uses task records that pass consistency checks on the manifest, replay trace, action identity, endpoint alignment, and call ledger.

Method	Regime	Action universe	H	K	Budget
StaticRootBudget	Single-step	Root only	1	384	384
Dynamic Greedy	Dynamic seq.	Regenerated	3	128	384
FrozenRoot native	FrozenRoot	Root survivors	3	128	384
FrozenRoot matched	FrozenRoot	Matched/depth	3	—	—
L2L-GP	Dynamic seq.	Regenerated	3	128	384
No-Ascent	Dynamic seq.	Regenerated	3	128	384

depletion, shortage, and unused-allowance behavior of a fixed root pool. FrozenRoot may select only root identities after current-state replay, while Dynamic may select newly materialized identities.

The proposal comparison is L2L-GP versus strict No-Ascent. The two methods share the task set, state transitions, candidate exposure, frontier, lookahead, legal projection, endpoint path, tie-break, and call accounting. The only declared proposal change is the ascent operator. Any result from a direct proxy-ranking controller is interpreted separately because it changes more than that single factor.

5.3 Metrics and admissibility

Primary comparisons report the task-level paired median difference in absolute gene-body delta, a task bootstrap 95% confidence interval, and win/tie/loss counts. Means are reported as descriptive summaries. Cost and failure summaries include logical requests, physical calls, duplicate proposals, shortage, unused allowance, and runtime when available. Relative gains and secondary tracks are not used to replace the primary absolute endpoint.

A task enters a primary paired comparison only when its manifest, replay trace, action identity, endpoint alignment, and call ledger pass consistency checks. Historical rows are labeled separately from fresh Part I rows. A primary Dynamic/FrozenRoot or No-Ascent claim requires same-task records under the declared method contract. Turnover and parent-state legality are separate computational evidence. Utility attribution requires the corresponding paired comparison.

The endpoint is a model prediction for the materialized computational state, not a biological measurement. This boundary allows Results to attach each claim to its measured comparison.

6 Results

6.1 State-dependent action identities turn over after an intended edit

The root action pool is large: 591 to 999 legal identities per RegEdit task (median 715.5).

How much of this pool survives a single intended edit? The median exact-identity Jaccard overlap between the root and next-state sets is 0.0998.

The tested trajectory selects from this reshaped pool. Thirty-two of 50 tasks select a root-absent second action. At the third action the count rises to 36 of 50. Across any later depth, 40 of 50 tasks contain at least one root-absent identity. The audited root-absent identities pass the guide/PAM, editor-window, and coordinate-replay checks in their direct parent states. Figure 2 and Table 2 separate set-level turnover, policy selection, and parent-state legality as independent evidence layers.

6.2 Native regime context and matched regeneration

Low overlap establishes that the action set changes; the next question is whether this matters for endpoint utility. At their native operating points, the three regimes differ in how they handle the post-edit action set: StaticRootBudget samples once from the root set, FrozenRoot retains root identities under current-state replay, and Dynamic Greedy regenerates after each transition.

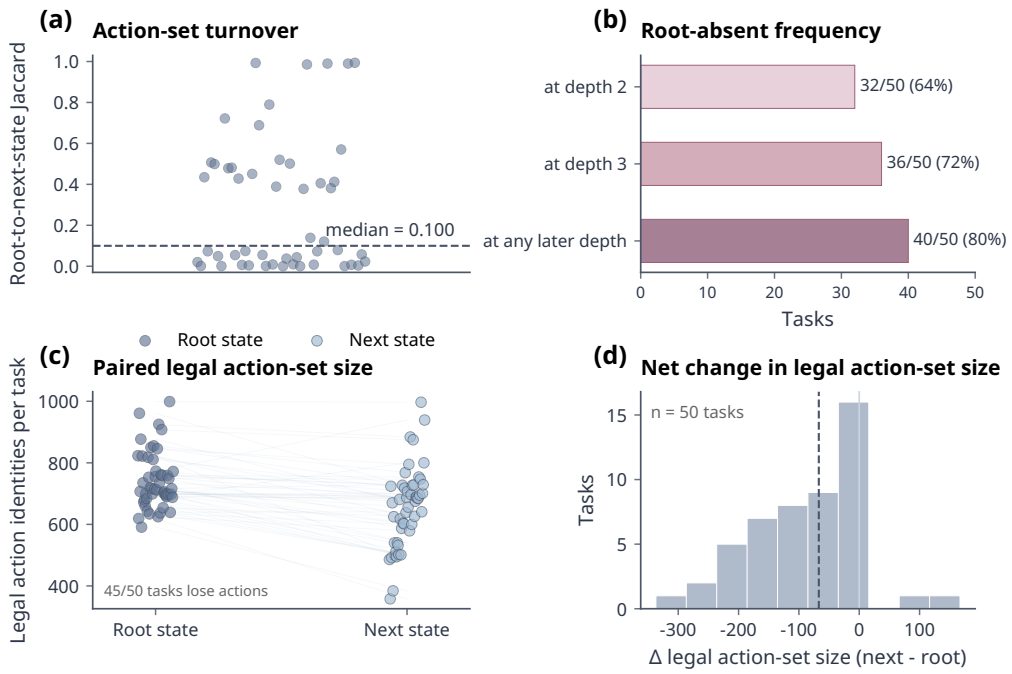


Figure 2: Action-space turnover after an intended-edit transition. Panels report exact root-to-next-state overlap, root-absent selections at later depths, paired legal-set sizes, and net size change across the 50 RegEdit tasks. Root-absent selection is policy-dependent; direct-parent legality is audited separately in Table 2.

Table 2: Task-level action-space turnover and identity-level legality audit on RegEdit.

Evidence layer	Quantity	Definition	Value
Set-level turnover	Root action count	$\lvert\mathcal{A}(s_0)\rvert$	median 715.5; range 591–999
	Root-to-next Jaccard	$\mathrm{Jaccard}(\mathcal{A}(s_0), \mathcal{A}(s_1))$	median 0.0998
Policy-selected identity	Root-absent at depth 2	$a_1 \notin \mathcal{A}(s_0)$	32/50 (64%)
	Root-absent at depth 3	$a_2 \notin \mathcal{A}(s_0)$	36/50 (72%)
	Root-absent at any later depth	$\exists\, t > 0 :\ a_t \notin \mathcal{A}(s_0)$	40/50 (80%)
Parent-state legality	Audited root-absent identity	$a_t \in \mathcal{A}(s_t)$	all audited identities pass

At the native operating point, Dynamic Greedy has a higher model-predicted endpoint than Frozen-Root on all 50 tasks, with mean and median paired absolute differences of 0.0204 and 0.0046 (Figure 3b).

Under the exposure-matched contract, the same task set, horizon, candidate exposure, and query budget are equalized at each task and depth. The mean paired difference is -0.000178, the median is 0, the task-bootstrap 95% interval is $[-0.0005,\ 0.0001]$, and the win/tie/loss counts are $7/32/11$ (Figure 3d). The native advantage does not survive exposure matching: action-set turnover is large, but newly available identities do not yield detectable additional utility under the matched contract.

6.3 Proposal operating point and attribution

Within the regenerated action space, how much does the proposal mechanism contribute? The attribution comparison is a one-factor test: L2L-GP versus strict No-Ascent, with all other controller and accounting conditions fixed, including task set, transitions, candidate exposure, legal projection, and endpoint evaluation. Because only the ascent operator changes, any paired difference under this matched contract attributes to the proposal layer.

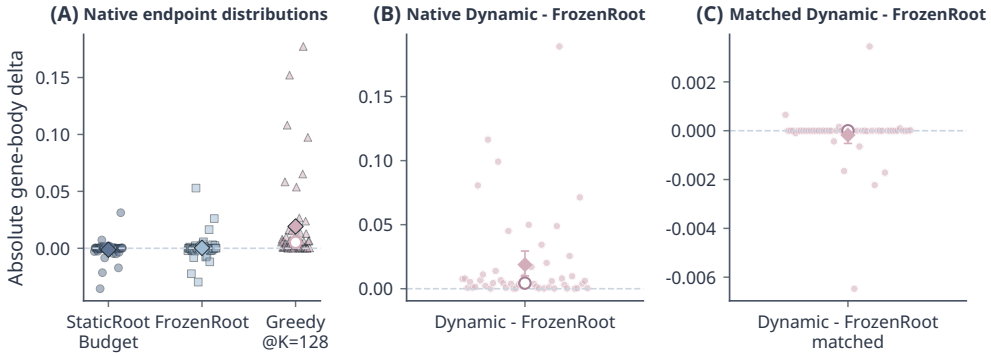


Figure 3: Regeneration comparisons on RegEdit. (A) Task-level endpoint distributions for StaticRootBudget, native FrozenRoot, and native Dynamic Greedy. (B) Native Dynamic minus FrozenRoot paired differences. (C) Native Dynamic minus StaticRoot. (D) Exposure-matched Dynamic minus FrozenRoot: mean -0.000178, median 0, bootstrap 95% CI $[-0.0005,\ 0.0001]$, W/T/L $7/32/11$.

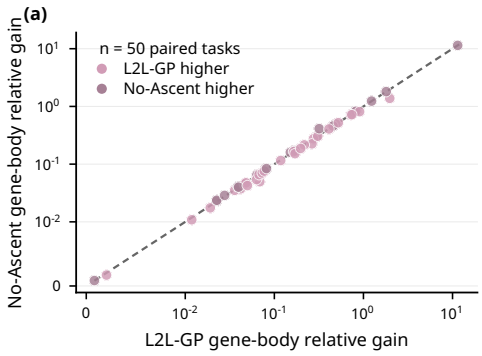


Figure 4: L2L-GP versus No-Ascent at the native operating point. Each point is one paired task; the diagonal is the equal-gain reference.

At the native $K = 128$, logical-cap $= 1792$ operating point, L2L-GP has a positive descriptive mean difference over No-Ascent of 0.01396 relative-gain units; the median difference is 0.000607.

At the task level, L2L-GP is higher on 30 tasks and No-Ascent is higher on 20.

6.4 Prefix continuation

Does the multi-edit trajectory continue to change the computational endpoint after the first committed edit? In 47 of 50 tasks the first transition increased the model-predicted endpoint; in the remaining 3 it decreased. By the third state, all 50 exceed the first ($F_3 > F_1$).

First-state-relative continuation averaged 30.73%. The median is 6.52%. The gap between mean and median reflects wide variation in continuation across the 50 tasks (Figure 5).

6.5 Query-scale sensitivity

The preceding comparisons use a single query scale. Does the observed ranking persist at other budget levels? Table 3 and Figure 6 compare historical native operating points at $K = 128$, 192, and 256. The rows report relative gene-body gain for BestSingle, Dynamic Greedy, L2L-GP, and Beam.

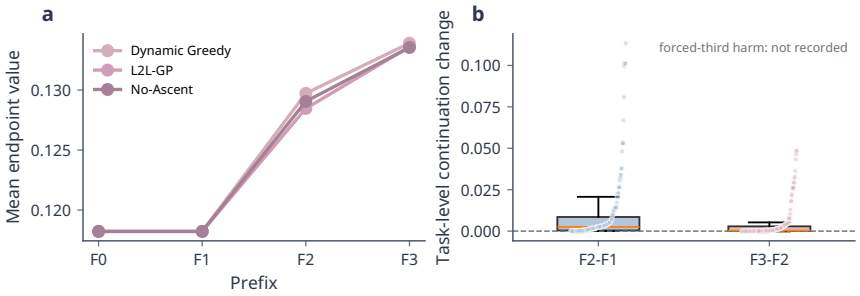


Figure 5: Prefix continuation across RegEdit. Each curve traces one task's model-predicted endpoint over continuation depth.

Table 3: Historical native K-scale operating points on RegEdit. Relative gene-body gains are normalized to the unedited state. BestSingle selects the single highest-scoring root action (depth 1). Dynamic Greedy† and Beam are historical native controllers. †The historical Greedy controller uses a dynamic sequential operating point.

K	Controller	Mean relative gain (%)	Median relative gain (%)
128	BestSingle	12.74	6.18
	Dynamic Greedy†	27.65	9.16
	L2L-GP	52.39	13.45
	Beam	33.97	9.10
192	BestSingle	16.46	6.69
	Dynamic Greedy†	59.95	11.21
	L2L-GP	60.14	12.08
	Beam	53.95	11.51
256	BestSingle	19.63	6.93
	Dynamic Greedy†	57.15	11.09
	L2L-GP	62.33	14.06
	Beam	58.70	13.63

L2L-GP has the highest median relative gain at each tested scale: 13.45% at $K = 128$, 12.08% at $K = 192$, and 14.06% at $K = 256$. At $K = 128$, L2L-GP and Dynamic Greedy are separated by 4.29 percentage points in median gain; at $K = 192$ the gap narrows to 0.87 points. The ranking is stable across these operating points, but the margin is not.

6.6 Cost and accounting

Logical requests, physical endpoint calls, duplicate proposals, candidate shortage, unused allowance, and runtime are recorded for each method. Detailed accounting is reported in the Appendix.

7 Discussion

7.1 Action-space turnover does not translate to utility

An intended edit reshapes the executable action universe: the post-edit set has low overlap with the root set, and the tested trajectories select identities that were absent at the root. The direct-parent legality audit confirms these identities are executable at the states where they appear. This supports a formulation in which action identity is state-dependent rather than fixed at the root.

The native comparison shows a consistent Dynamic advantage on all 50 tasks. The exposure-matched comparison shows no detectable difference. The action space turns over, but the current search does not convert the newly available identities into utility. This separation—large turnover,

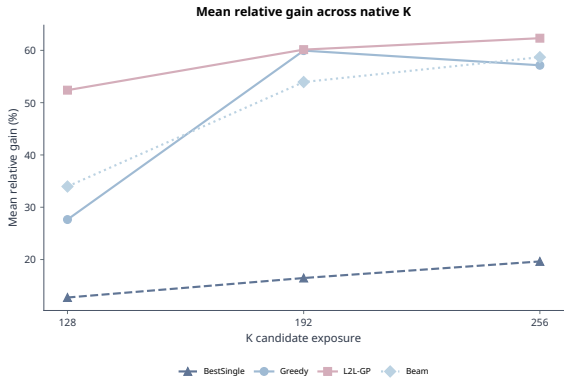


Figure 6: Historical native query-scale sensitivity at $K = 128$, 192, and 256.

null matched utility—is the central empirical finding. It implies that the bottleneck is not whether new actions exist, but whether the proposal mechanism can identify the valuable ones among them.

7.2 Proposal mechanisms operate inside the dynamic graph

L2L-GP is a proposal mechanism, not the definition of the planning problem. Its native comparison with No-Ascent is task-dependent: the mean difference is positive, the median is near zero, and task-level signs are mixed. The one-factor contract isolates proposal-layer contribution from regeneration. The formulation remains meaningful with a different proposal rule; a better proposal would strengthen the search without redefining the computational object.

7.3 Computational scope

All endpoints are predictions of a frozen evaluator applied to deterministic intended-edit states. The replay does not model allele mixtures, repair distributions, delivery, or phenotype. Both observations are computational; biological efficacy requires a separate experimental chain. A controller should expose the current action graph after each edit, retain identity provenance, and account for logical requests separately from physical calls.

The primary evidence is a 50-task, INCREASE-direction computational study under one frozen evaluator; the results establish the planning phenomenon and its matched operating-point behavior, not broad cross-task or biological generalization.

8 Conclusion

We formulated multi-edit native-locus regulatory intervention as planning over state-dependent executable action identities. On the frozen 50-task RegEdit benchmark, intended edits produce low-overlap successor action sets, and later selected identities can be absent from the root universe while satisfying the direct-parent legality contract. Prefix continuation and query-scale analyses show that the predicted endpoint keeps changing across depths and budget levels.

The core distinction is between exposing the current action graph and proposing actions within it. Dynamic regeneration addresses the former; L2L-GP is one mechanism for the latter. Native comparisons provide operating-point context, while matched Dynamic–FrozenRoot and L2L-GP–No-Ascent contracts determine the corresponding utility and proposal attributions under the frozen endpoint and intended-edit replay setting. The broader point is that any multi-edit controller must represent the fact that its own actions change the space of future actions.The L2L-GP–No-Ascent contract provides one proposal-layer test within that dynamic graph; the remaining challenge is to convert newly reachable action identities into utility under a fixed exposure contract.

REFERENCES

Ziga 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. URL https://doi.org/10.1038/s41592-021-01252-x.

Ziga Avsec, Natasha Latysheva, Jun Cheng, Guido Novati, Kyle R. Taylor, Tom Ward, Clare Bycroft, Lauren Nicolaisen, Eirini Arvaniti, Joshua Pan, Raina Thomas, Vincent Dutordoir, Matteo Perino, Soham De, Alexander Karollus, Adam Gayoso, Toby Sargeant, Anne Mottram, Lai Hong Wong, Pavol Drotar, Adam Kosiorek, Andrew Senior, Richard Tanburn, Taylor Applebaum, Souradeep Basu, Demis Hassabis, and Pushmeet Kohli. Advancing regulatory variant effect prediction with alphagenome. *Nature*, 649(8099):1206–1218, 2026. doi: 10.1038/s41586-025-10014-0. URL https://doi.org/10.1038/s41586-025-10014-0.

Yoshua Bengio, Andrea Lodi, and Antoine Prouvost. Machine learning for combinatorial optimization: A methodological tour d'horizon. *European Journal of Operational Research*, 290(2): 405–421, 2021. doi: 10.1016/j.ejor.2020.07.063. URL https://doi.org/10.1016/j.ejor.2020.07.063.

Guohui Chuai, Hanhui Ma, Jifang Yan, Ming Chen, Nanfang Hong, Dongyu Xue, Chi Zhou, Chenyu Zhu, Ke Chen, Bin Duan, Feng Gu, Sheng Qu, Deshuang Huang, Jia Wei, and Qi Liu. Deepcrispr: optimized crispr guide rna design by deep learning. *Genome Biology*, 19(1):1459, 2018. doi: 10.1186/s13059-018-1459-4. URL https://doi.org/10.1186/s13059-018-1459-4.

Ryan Clarke, Alexander R. Terry, Hannah Pennington, Cody Hasty, Matthew S. MacDougall, Maureen Regan, and Bradley J. Merrill. Sequential activation of guide rnas to enable successive crispr-cas9 activities. *Molecular Cell*, 81(2):226–238.e5, 2021. doi: 10.1016/j.molcel.2020.12.003. URL https://doi.org/10.1016/j.molcel.2020.12.003.

Jean-Paul Concordet and Maximilian Haeussler. Crispor: intuitive guide selection for crispr/cas9 genome editing experiments and screens. *Nucleic Acids Research*, 46(W1):W242–W245, 2018. doi: 10.1093/nar/gky354. URL https://doi.org/10.1093/nar/gky354.

Le Cong, F. Ann Ran, David Cox, Shuailiang Lin, Robert Barretto, Naomi Habib, Patrick D. Hsu, Xuebing Wu, Wenyan Jiang, Luciano A. Marraffini, and Feng Zhang. Multiplex genome engineering using crispr/cas systems. *Science*, 339(6121):819–823, 2013. doi: 10.1126/science.1231143. URL https://doi.org/10.1126/science.1231143.

Bernardo P. de Almeida, Franziska Reiter, Michaela Pagani, and Alexander Stark. Deepstarr predicts enhancer activity from dna sequence and enables the de novo design of synthetic enhancers. *Nature Genetics*, 54(5):613–624, 2022. doi: 10.1038/s41588-022-01048-5. URL https://doi.org/10.1038/s41588-022-01048-5.

Jibiao Fan, Leisheng Shi, Qi Liu, Zhipeng Zhu, Fan Wang, Runxian Song, Jimeng Su, Degui Zhou, Xiao Chen, Kailong Li, Lixiang Xue, Lichao Sun, and Fengbiao Mao. Annotation and evaluation of base editing outcomes in multiple cell types using crisprbase. *Nucleic Acids Research*, 51(D1): D1249–D1256, 2023. doi: 10.1093/nar/gkac967. URL https://doi.org/10.1093/nar/gkac967.

Oliver Kroemer, Scott Niekum, and George Konidaris. A review of robot learning for manipulation: Challenges, representations, and algorithms. *Journal of Machine Learning Research*, 22(30): 1–82, 2021. URL https://www.jmlr.org/papers/v22/19-804.html.

Kornel Labun, Tessa G. Montague, Maximilian Krause, Yamila N. Torres Cleuren, Hakon Tjeldnes, and Eivind Valen. Chopchop v3: expanding the crispr web toolbox beyond genome editing. *Nucleic Acids Research*, 47(W1):W171–W174, 2019. doi: 10.1093/nar/gkz365. URL https://doi.org/10.1093/nar/gkz365.

Johannes Linder, Divyanshi Srivastava, Han Yuan, Vikram Agarwal, and David R. Kelley. Predicting rna-seq coverage from dna sequence as a unifying model of gene regulation. *Nature Genetics*, 57(4):949–961, 2025. doi: 10.1038/s41588-024-02053-6. URL https://doi.org/10.1038/s41588-024-02053-6.

Jason Qin, Hans-Hermann Wessels, Carlos Fernandez-Granda, and Yuhan Hao. Active learning for efficient discovery of optimal combinatorial perturbations. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 50363–50383. PMLR, 2025. URL https://proceedings.mlr.press/v267/qin25g.html.

Zhenpeng Zhou, Steven Kearnes, Li Li, Richard N. Zare, and Patrick Riley. Optimization of molecules via deep reinforcement learning. *Scientific Reports*, 9(1):10752, 2019. doi: 10.1038/s41598-019-47148-x. URL https://doi.org/10.1038/s41598-019-47148-x.

A Implementation Details

A.1 State replay and audit trail

Every materialized action records its parent state, identity payload, legality result, coordinate replay status, and root-presence status. Selected trajectories retain the action identity, parent legality, endpoint-before and endpoint-after values, and cumulative logical and physical calls. FrozenRoot additionally records root-pool survivors, survivor ratios, rejected new identities, shortage, and unused logical allowance.

The accounting layer distinguishes logical requests, cache hits, physical model calls, duplicate proposals, and candidate shortage. No-padding is part of the contract: when the current legal pool is smaller than the requested exposure, the run records the shortage and unused allowance rather than creating virtual actions. Together, these records support state- and task-level replay.

A.2 Training-free proxy definition

At state s_t, the action encoder produces an 18-dimensional representation $e_t(a)$ for each current legal action a. Its coordinates combine edit-mechanism indicators, position and span scaled to the locus, guide/PAM composition, an efficiency prior, target-window overlap, and edit burden. A deterministic state encoder supplies $z_t = h(s_t)$. The training-free value proxy uses this context to define a proposal field over the engineered action space.

L2L-GP initializes the continuous waypoint from the highest-valued proxy action and optimizes

$$\widetilde{J}(z_t, u) = J_{\text{proxy}}(z_t, u) - \lambda_{\text{valid}} \min_{a \in \mathcal{C}_K(s_t)} \|u - e_t(a)\|_2^2 - \lambda_{\text{norm}} \|u\|_2^2.$$

L2L-GP uses 32 ascent steps with learning rate 0.05 and $\lambda_{\text{valid}} = 0.1$. The latent-to-legal operator then projects the waypoint onto the current legal candidate set,

$$\mathcal{P}_K(u_t^*) = \text{TopK}_{a \in \mathcal{C}_K(s_t)} \left[-\|u_t^* - e_t(a)\|_2^2 \right].$$

The projected actions are materialized and evaluated with the frozen endpoint. After the selected intended transition, the action set and its engineered geometry are reconstructed for the new state. No-Ascent preserves the same initialization, representation, projection, frontier, lookahead, endpoint path, and accounting contract while setting the number of ascent steps to zero.