CELLSCIENTIST: FROM EXECUTION FEEDBACK TO AUDITABLE MODEL-REVISION TRAJECTORIES FOR CELLULAR PERTURBATION PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Cellular perturbation-response modeling requires coordinated choices of representations, fusion mechanisms, objectives, and training procedures. Large language models (LLMs) can propose executable candidates, but unconstrained revision can produce invalid implementations, change task semantics, or discard useful components. We present **CellScientist**, a protocol-constrained workflow that converts execution and validation feedback into auditable model-revision trajectories. It records design states and outcomes, routes discrepancies to specific components, and applies local revisions under a fixed task contract. A matched-budget study fixes the candidate language, predictor, fitting, and evaluator: structured revision finds better held-out predictors at small budgets under two LLM backbones. Operational audits link history to fewer repeated proposals, contract checks to contained violations, and discrepancy routing to targeted repairs. Refits of two frozen designs on an independently acquired cohort evaluate external predictive utility. Open-workflow trajectories retain improvements, regressions, and failures, while transcriptomic and single-cell searches extend application to additional response spaces. CellScientist produces both a selected predictor and an inspectable record of its development.

1 INTRODUCTION

Cellular perturbation-response assays measure how chemical or genetic interventions reshape cellular states. Complementary readouts include Cell Painting morphology, transcriptomic profiles, and single-cell measurements (Stirling et al., 2021; Seal et al., 2025; Subramanian et al., 2017). Predicting these responses can extend observed effects to unmeasured perturbations and contexts, supporting compound characterization, mechanism-of-action analysis, and virtual-cell modeling.

Generalization remains difficult because measured responses combine perturbation-induced biology with assay- and context-specific variation. Morphological profiles depend on plate and processing factors (Arevalo et al., 2024), while transcriptomic responses vary across batches and cellular contexts (Zhou et al., 2025; Zhang et al., 2025). Effective predictors therefore require coordinated choices of perturbation and context representations, dose conditioning, feature interactions, objectives, and training procedures. Revising one component can change the behavior of others, making feedback attribution central to model development.

These dependencies create a feedback-attribution problem. A poor validation outcome does not identify whether the next change should target conditioning, representation, fusion, or the objective. Execution failures raise a different question: whether a candidate can be evaluated under the intended protocol. A useful search controller must distinguish these signals before converting them into a concrete revision.

AutoML systems search predefined model and hyperparameter spaces (Wang et al., 2021; Gijsbers et al., 2024), whereas LLM-based modeling agents can generate and revise executable programs (Jiang et al., 2025; Lu et al., 2024; Tang et al., 2025). This flexibility also requires control over how execution and validation outcomes guide subsequent changes. Revisions must preserve task semantics, target components implicated by the feedback, and retain the evidence behind each decision. We therefore

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.

ask how execution feedback can guide localized model-design revisions while preserving a fixed task contract.

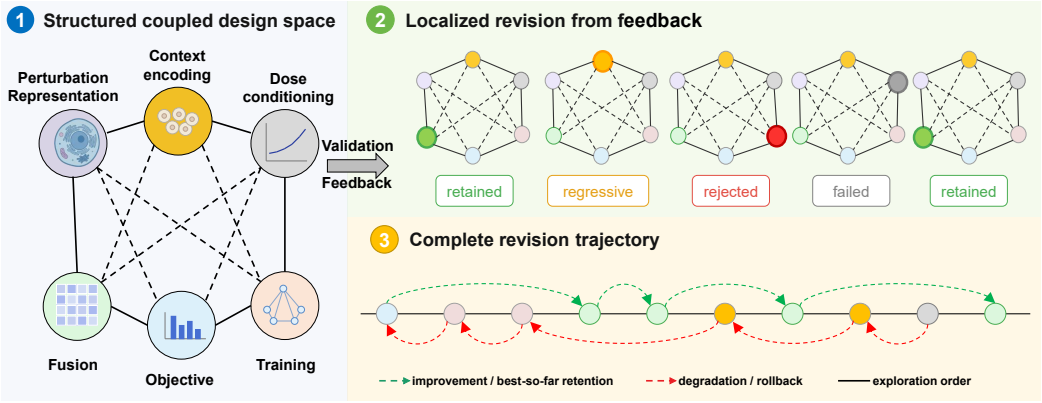


Figure 1: **Conceptual overview of CellScientist.** Revision operates over coupled choices of perturbation representation, context encoding, dose conditioning, fusion, objectives, and training strategy. Recorded feedback guides local updates, while the trajectory preserves improvements, regressions, rollbacks, rejected candidates, execution failures, and validation-retained states.

We propose **CellScientist**, a framework for structured and auditable model revision in cellular perturbation-response prediction. It represents the coupled design space, routes feedback to implicated components, and records successive revisions (Figure 1). Each state links a design, implementation, and outcome under a fixed task contract that preserves data, interface, and evaluation semantics.

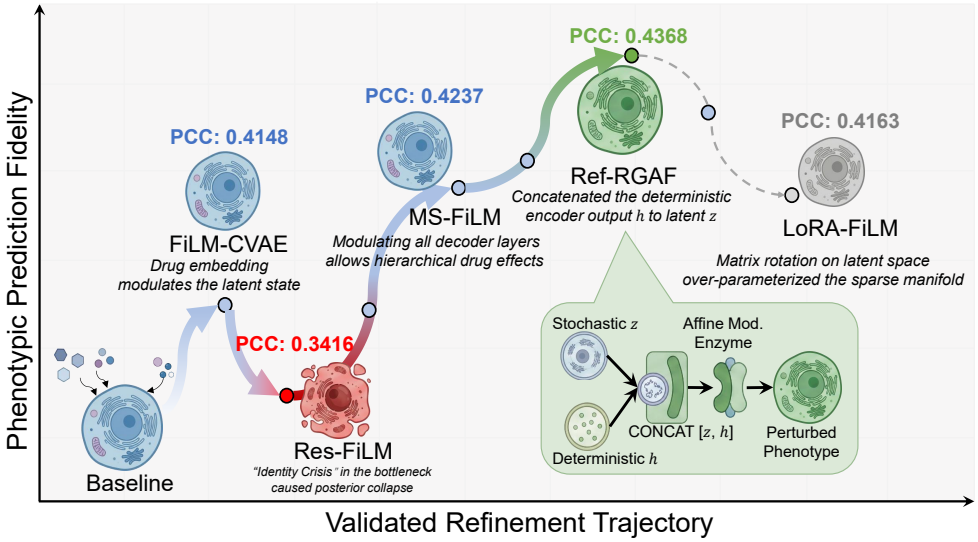


Figure 2: **Feedback-guided model-revision trajectory on BBBC047.** Recorded validation feedback guides successive revisions from modulation-based candidates to Ref-RGAF, which adds a deterministic control-reference encoding and attains the highest validation PCC in this trajectory. The annotations summarize working diagnostics recorded during the search. The retained record includes intermediate regressions and the later, lower-performing LoRA-FiLM candidate.

Figure 2 illustrates this process on BBBC047. Modulation-based candidates show both improvements and regressions, prompting a working diagnosis that the supplied control-reference information is underused. A localized revision introduces a deterministic reference encoding alongside the stochastic latent state, yielding the validation-selected Ref-RGAF candidate. The recorded later regression makes clear why the selected model differs from the last attempted candidate. Appendix B.2 gives the detailed trace.

CellScientist makes this transition explicit through three complementary functions. History-Aware Revision Tracking (HRT) maintains design states and evidence from previous attempts. Limited-Change Application (LCA) checks candidate eligibility and applies bounded repair. Performance-Discrepancy Refinement (PDR) links observed discrepancies to revision targets. Together, they connect the rationale for an edit to its implementation and subsequent validation, while preserving unsuccessful branches for inspection.

Our contributions are threefold. First, we formulate a *model-design revision-routing problem* over coupled components under a fixed task contract. Second, CellScientist links local revisions to auditable design states, implementations, and outcomes. Third, matched-budget comparisons under two LLM backbones measure revision efficiency, operational audits examine the component roles, and external refits evaluate the predictive utility of frozen discovered designs.

2 Related Work

Cellular perturbation-response modeling. Cell Painting models combine chemical structure, dose, and context to predict morphology (Stirling et al., 2021; Chandrasekaran et al., 2023; Seal et al., 2025). DeepCE, MultiDCP, and TranSiGen model expression (Pham et al., 2021; Wu et al., 2022; Tong et al., 2024); XPert uses biologically informed dual-branch modeling of cellular context and dose–time response (Guo et al., 2026). GEARS predicts unseen genetic perturbations (Roohani et al., 2024), while MAP uses knowledge-aligned representations for unprofiled drugs (Feng et al., 2026). Systema separates perturbation-specific signals from systematic variation (Viñas Torné et al., 2026). These approaches motivate representation, conditioning, and fusion as revision targets.

Representations and predictive evaluation. Hasanaj et al. (2025) compare perturbation embeddings with a fixed downstream learner; MVCBench, CHMR, and MORSE examine representation quality and multimodal integration (Li et al., 2026a;b; Zhang et al., 2026). Tabular foundation models are competitive across cell-level, pseudobulk, and organismal perturbation tasks (Palla et al., 2026). PerturBench highlights competitive simple models and perturbation-ranking metrics beyond reconstruction error (Wu et al., 2025). These findings motivate separating predictor quality from revision-policy efficiency: our controlled comparison fixes the candidate language, fitting procedure, and evaluator.

Automated model development. FLAML and AutoML benchmarks study predefined search spaces (Wang et al., 2021; Gijsbers et al., 2024); AIDE and scientific agents extend proposals to executable programs (Jiang et al., 2025; Lu et al., 2024; Turcan et al., 2025). MLAgentBench and MLE-bench evaluate end-to-end experimentation (Huang et al., 2024; Chan et al., 2025). MLE-STAR uses code-block ablations for targeted refinement (Nam et al., 2025). CellScientist evaluates diagnostic-ranked, history-filtered address routing under matched candidate budgets.

Biomedical agents and virtual cells. Virtual-cell LLM systems span direct predictors and tool-orchestrating agents (Li et al., 2025). Biomedical agents address spatial analysis (Wang et al., 2025a; Yu et al., 2025) and laboratory automation (Qiu et al., 2025; Wang et al., 2025b), with workflow evaluation studied by BioMLBench (Miller et al., 2025). AssayBench evaluates gene-hit ranking from assay descriptions (De Brouwer et al., 2026). CellForge studies agent-designed perturbation models (Tang et al., 2025); VCWorld integrates biological knowledge (Wei et al., 2026); HarmonyCell combines hierarchical search, execution repair, and solution reuse (Huang et al., 2026). CellScientist targets the revision decision within model development, linking feedback, component edits, and retained outcomes under protected task semantics.

3 Methodology

CellScientist is a protocol-constrained controller for iterative perturbation-response model search. It links a model-design space $\mathcal{H}$ to an executable-candidate space $\mathcal{I}$ through the loop "Design Specification $\rightarrow$ Executable Candidate $\rightarrow$ Validation Feedback $\rightarrow$ Local Revision," with $\mathbf{h}$ denoting a model-design state.

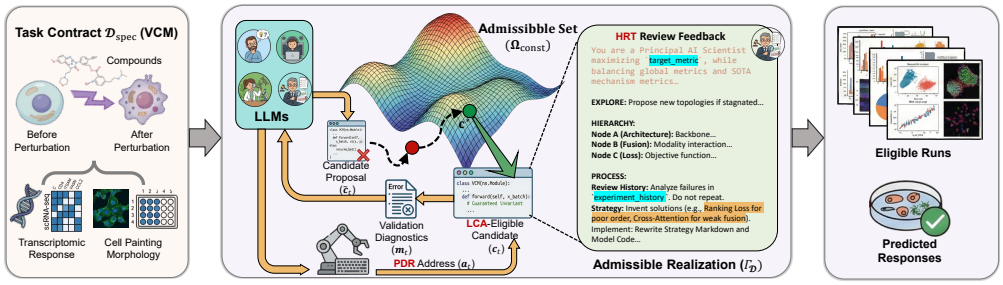


Figure 3: **Overview of the CellScientist framework.** The task contract $\mathcal{D}_{\text{spec}}$ fixes inputs, perturbation descriptors, response targets, data partitions, and evaluation semantics. HRT supplies the design state $\mathbf{h}_t$, task-state graph $\mathcal{G}_t$, and run history $\mathcal{B}_t$ for proposing $\tilde{\mathbf{c}}_t$. LCA applies configured checks and bounded repair to return an eligible candidate $\mathbf{c}_t$ or rejection $\perp$. Validation yields diagnostics $\mathbf{m}_t$, which PDR maps to a component address a_t for localized revision. Validation selection retains the best eligible candidate; the complete trajectory is recorded.

3.1 Task Contract and Iterative Search

Let $\mathbf{h}_t \in \mathcal{H}$ denote the model-design specification at iteration t and let $\mathbf{c}_t \in \mathcal{I}$ denote an executable candidate. A task is specified by $\mathcal{D}_{\text{spec}} = (\mathcal{X}, \mathcal{P}, \mathcal{Y}, \mathcal{S}, \mathcal{E})$, where $\mathcal{X}$ contains observed cellular context, $\mathcal{P}$ contains perturbation descriptors, $\mathcal{Y}$ contains response targets, $\mathcal{S}$ defines the data-partition protocol, and $\mathcal{E}$ defines the evaluation functional. The task contract remains fixed throughout a search trajectory.

The controller implements the following operational sequence:

$$\begin{aligned} \tilde{\mathbf{c}}_t &= \Phi(\mathbf{h}_t, \mathcal{G}_t, \mathcal{B}_t), \\ \mathbf{c}_t &= \Gamma_{\mathcal{D}}(\tilde{\mathbf{c}}_t; \mathbf{h}_t, \mathcal{G}_t, \mathcal{B}_t, R) \in \Omega_{\text{const}} \cup \{\perp\}, \\ \mathbf{m}_t &= \mathcal{F}(\mathbf{c}_t; \mathcal{D}_{\text{val}}, \mathcal{E}) \quad (\mathbf{c}_t \neq \perp), \\ \mathbf{h}_{t+1} &= \mathbf{h}_t \oplus \mathcal{T}_{\text{edit}}\big(\mathcal{A}(\mathbf{m}_t, \mathcal{G}_t, \mathcal{B}_t) \mid \mathbf{m}_t\big). \end{aligned} \tag{1}$$

Here Φ requests candidate code or a registered candidate ID from the LLM, and $\Gamma_{\mathcal{D}}$ applies the configured checks and bounded repairs. For $\mathbf{c}_t = \perp$, $\mathbf{m}_t$ contains failure status and error text, with validation metrics marked unavailable. The state $\mathcal{Z}_t = (\mathbf{h}_t, \mathcal{G}_t, \mathcal{B}_t)$ contains the design, structured task record, and attempted implementations with outcomes. Section 3.4 and Appendix C.4.4 specify controlled routing and edit instantiation.

3.2 HRT: History-Aware Revision Tracking

History-Aware Revision Tracking (HRT) represents the editable model-design state as a directed task-state graph $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t)$. Nodes denote components such as perturbation representation, cellular-context encoding, fusion, architecture, objective design, and training strategy; edges record configured dependencies or ordering relations. The graph provides structured context for candidate generation and is updated with task status and evidence from prior attempts.

Given a diagnostic, the controller records intended revision targets and supplies their descriptions to proposal generation. Graph edges organize task context; the mutable-region registry and configured checks govern application. In the controlled study the graph records dependencies, while the typed grammar and history define legal one-field edits. HRT makes previous attempts available for filtering and retention.

3.3 LCA: Limited-Change Application

Limited-Change Application (LCA) governs candidate application and eligibility. Open-code proposals are applied to configured mutable notebook regions, followed by runtime checks and bounded repair. Controlled proposals undergo candidate-domain, allow-list, pairing, duplicate, prediction-shape, and finite-metric checks. Appendix C.4.4 distinguishes these live checks from the separate registered contract-record audit. Failed attempts remain in the history and are excluded from selection.

Write $\chi_{\mathrm{I}} = 1$ when the configured checks pass. For the applied candidate $\mathbf{c}^{(0)}$ and at most R repair attempts, let r^* be the first passing attempt. Then

$$\Gamma_{\mathcal{D}}(\tilde{\mathbf{c}}_t; \mathbf{h}_t, \mathcal{G}_t, \mathcal{B}_t, R) = \begin{cases} \mathbf{c}^{(r^*)}, & \exists r \le R : \chi_{\mathrm{I}}(\mathbf{c}^{(r)}) = 1, \\ \perp, & \text{otherwise.} \end{cases} \tag{2}$$

The passing set is $\Omega_{\mathrm{const}} = \{\mathbf{c} : \chi_{\mathrm{I}}(\mathbf{c}) = 1\}$. Passing candidates enter validation selection, and all outcomes are appended to $\mathcal{B}_{t+1}$.

3.4 PDR: Performance-Discrepancy Refinement

Performance-Discrepancy Refinement (PDR) maps measured discrepancies to edit targets. Controlled feedback includes global PCC r, response PCC r_Δ, top-50 response PCC $r_{\Delta,50}$, and error metrics. For feedback incumbent $(\bar{\mathbf{h}}_t, \bar{\mathbf{m}}_t)$, routing forms an ordered list L_t and menu Q_t:

$$\begin{aligned} L_t &= \mathrm{FilterHistory}(\mathrm{Rank}(\bar{\mathbf{m}}_t, \bar{\mathbf{h}}_t, \mathcal{B}_t), \mathcal{B}_t), \\ Q_t &= \{(j, e_j(\bar{\mathbf{h}}_t, \mathcal{B}_t)) : j \in \mathrm{FirstTwo}(L_t)\}, \\ (a_t, \mathbf{h}_{t+1}) &= \mathrm{LLMSelect}(Q_t \mid \bar{\mathbf{m}}_t, \bar{\mathbf{h}}_t). \end{aligned} \tag{3}$$

The five addresses are conditioning, target, input rank, output rank, and regularization. Fixed priorities first address missing conditioning, then direct targets or $r_\Delta - r > 0.05$, then decoder discrepancy $r_{\Delta,50} < r_\Delta$, followed by capacity and fallback rules. History removes exhausted or unsuccessful addresses. Simultaneous diagnostics retain this order, with duplicates removed. Each e_j selects one unseen one-field edit by the shared preference order; the LLM chooses an allowed pair, with the first pair as fallback. Appendix C.4.4 specifies every rule and check.

Open-code revision. The open-code workflow uses a broader component vocabulary: the LLM proposes a focus and edits from diagnostics, graph context, and history; $\mathcal{T}_{\mathrm{edit}}$ records that revision request. An interface error can target an adapter, while suspected control-reference underuse can target an encoder or decoder. Appendix C.4.3 specifies this separate code-editing procedure.

In the BBBC047 example, the diagnosis becomes a decoder-input change: concatenate a deterministic reference encoding with the stochastic latent state. LCA checks the candidate, execution supplies its validation outcome, and HRT retains the edit and outcome, including the subsequent LoRA-FiLM regression. The record links a working diagnosis to an implementation change and the measured outcome used for selection.

3.5 Trajectory Selection and Workflow

CellScientist stores each attempt in the run history. Let $\eta_t = \mathbb{I}\{\mathbf{c}_t \in \Omega_{\mathrm{const}}(\mathcal{D}_{\mathrm{spec}})\}$ indicate whether the proposed implementation passed the configured eligibility checks, and let $\mathcal{B}_{t+1} = \mathcal{B}_t \cup \{(\mathbf{h}_t, \tilde{\mathbf{c}}_t, \mathbf{c}_t, \mathbf{m}_t, a_t, \eta_t)\}$. The validation-selected candidate and its design state are given by

$$(\mathbf{h}_{\mathrm{best}}, \mathbf{c}_{\mathrm{best}}) = \mathrm{Select}_{\mathrm{val}}\left(\{(\mathbf{h}_i, \mathbf{c}_i, \mathbf{m}_i) : \eta_i = 1\}_{i=0}^{T-1}, \mathcal{E}\right), \tag{4}$$

which returns the best eligible model-design state and executable candidate under the configured validation rule.

4 Experiments

The central empirical question is whether recorded feedback helps a controller spend a limited revision budget more effectively. We first compare controllers under a shared candidate language and held-out evaluator, then examine the operational roles of HRT, LCA, and PDR. Finally, we test two frozen model designs on an independently acquired cohort and connect model selection to the recorded open-workflow trajectories.

RQ1: Revision efficiency. Under a shared candidate language, initialization, fitting, and evaluation budget, does structured revision find better predictors earlier, and does the pattern recur with a second LLM backbone? **RQ2: Operational mechanisms.** How do persistent history, contract checks, and discrepancy routing affect repeated proposals, invalid evaluations, and fault localization? **RQ3: Predictive utility and revision records.** Do frozen discovered designs retain predictive utility after external refitting, and what do their full trajectories reveal about model selection?

Table 1: **Five-seed matched-budget controller comparison on held-out global PCC.** Upper: mean±sample SD over five paired controller/fitting seeds. Lower: CellScientist-minus-comparator effects with four-setting 95% t intervals, descriptive across the two cohorts and their grouping protocols. Policies share initialization, candidate language, fitting, and evaluation budget, including h_0. Blue identifies CellScientist; **bold** marks the best mean.

Dataset	Split	Budget	Fixed h_0	Flat retry	Random routing	AIDE-style	CellScientist
BBBC036	Plate	3	$0.1664_{\pm 0.0005}$	$0.1828_{\pm 0.0005}$	$0.2250_{\pm 0.0493}$	$0.2244_{\pm 0.0276}$	$\mathbf{0.2650}_{\pm 0.0005}$
		5	$0.1664_{\pm 0.0005}$	$0.2073_{\pm 0.0056}$	$0.2277_{\pm 0.0502}$	$0.2477_{\pm 0.0294}$	$\mathbf{0.2674}_{\pm 0.0003}$
		10	$0.1664_{\pm 0.0005}$	$0.2175_{\pm 0.0061}$	$0.2626_{\pm 0.0077}$	$0.2562_{\pm 0.0205}$	$\mathbf{0.2693}_{\pm 0.0012}$
BBBC036	SMILES	3	$0.1547_{\pm 0.0005}$	$0.1583_{\pm 0.0046}$	$0.2052_{\pm 0.0413}$	$0.2058_{\pm 0.0230}$	$\mathbf{0.2336}_{\pm 0.0016}$
		5	$0.1547_{\pm 0.0005}$	$0.1882_{\pm 0.0129}$	$0.2049_{\pm 0.0417}$	$0.2231_{\pm 0.0169}$	$\mathbf{0.2346}_{\pm 0.0009}$
		10	$0.1547_{\pm 0.0005}$	$0.1948_{\pm 0.0005}$	$\mathbf{0.2347}_{\pm 0.0016}$	$0.2235_{\pm 0.0164}$	$0.2346_{\pm 0.0009}$
BBBC047	Plate	3	$0.2442_{\pm 0.0008}$	$0.2500_{\pm 0.0055}$	$0.3032_{\pm 0.0500}$	$0.3013_{\pm 0.0381}$	$\mathbf{0.3396}_{\pm 0.0004}$
		5	$0.2442_{\pm 0.0008}$	$0.2630_{\pm 0.0130}$	$0.3066_{\pm 0.0514}$	$0.3153_{\pm 0.0399}$	$\mathbf{0.3457}_{\pm 0.0004}$
		10	$0.2442_{\pm 0.0008}$	$0.2777_{\pm 0.0005}$	$0.3458_{\pm 0.0027}$	$0.3191_{\pm 0.0372}$	$\mathbf{0.3481}_{\pm 0.0009}$
BBBC047	SMILES	3	$0.2504_{\pm 0.0004}$	$0.2651_{\pm 0.0002}$	$0.3131_{\pm 0.0536}$	$0.3123_{\pm 0.0346}$	$\mathbf{0.3529}_{\pm 0.0010}$
		5	$0.2504_{\pm 0.0004}$	$0.2916_{\pm 0.0003}$	$0.3207_{\pm 0.0582}$	$0.3371_{\pm 0.0417}$	$\mathbf{0.3614}_{\pm 0.0003}$
		10	$0.2504_{\pm 0.0004}$	$0.2920_{\pm 0.0003}$	$0.3668_{\pm 0.0073}$	$0.3560_{\pm 0.0359}$	$\mathbf{0.3751}_{\pm 0.0017}$

CellScientist advantage: mean effect with paired-setting 95% CI

Budget	vs. Fixed h_0	vs. Flat	vs. Random	vs. AIDE-style
$B = 3$	$+0.0939$ $[+0.0773, +0.1104]$	$+0.0837$ $[+0.0734, +0.0940]$	$+0.0361$ $[+0.0275, +0.0448]$	$+0.0368$ $[+0.0270, +0.0466]$
$B = 5$	$+0.0984$ $[+0.0774, +0.1193]$	$+0.0648$ $[+0.0404, +0.0892]$	$+0.0373$ $[+0.0292, +0.0454]$	$+0.0215$ $[+0.0088, +0.0342]$
$B = 10$	$+0.1029$ $[+0.0737, +0.1320]$	$+0.0612$ $[+0.0306, +0.0919]$	$+0.0043$ $[-0.0018, +0.0104]$	$+0.0181$ $[+0.0052, +0.0309]$

4.1 Experimental Setup

Cellular response tasks. The main experiments use Cell Painting morphology profiles in CellProfiler feature space (Stirling et al., 2021). A predictor receives a vehicle-control reference, compound information, and dose, and predicts the treated morphology profile. BBBC036 and BBBC047 (Bray et al., 2017) supply the controller comparison under plate-based and exact-SMILES-based grouping. LKCP Batch 2 supplies the external evaluation (Natoli et al., 2021; Way et al., 2022). Global PCC is computed after flattening prediction and observation matrices; MSE and R^2 additionally measure prediction error and explained variation. Appendix C.1 describes the cohorts and preprocessing.

Matched controller protocol. All policies share the initial state h_0, candidate language, low-rank predictor, fitting procedure, evaluator, paired seeds, and retention rule; revising policies also share the candidate budget. The language crosses conditioning (4 choices), response target (2), input rank (4), output rank (4), and regularization (6), yielding $4 \times 2 \times 4 \times 4 \times 6 = 768$ candidates. Their one-edit neighborhoods contain $3, 1, 3, 3, 5$ alternatives before history filtering. This universe is fixed for the reported comparison; per-step shortlists are policy-specific. CellScientist selects registered candidates through address routing. Appendix C.4.4 specifies the domains, shared edit mapping, policy access, and fixed comparison protocol. Folds 1–3 fit candidates; disjoint fold-4 subsets provide feedback and selection scores; fold 5 reports held-out performance. Retention uses the mean feedback and selection PCC. Five paired controller/fitting seeds are evaluated at $B \in \{1, 3, 5, 10\}$, including h_0. Each paired seed initializes the same randomized fit across policies. Within each LLM backbone, decoding settings are fixed.

External design protocol. The DD-MIA and Ref-RGAF architectures selected on BBBC036 and BBBC047 are frozen and fitted afresh to LKCP. Compound-disjoint partitions contain 33 training

compounds, 15 validation compounds, and 16 test compounds. RealMLP (Holzmüller et al., 2024), RF-SKL (Breiman, 2001; Pedregosa et al., 2011), early-fusion MLP, and control-only MLP use the same target and split. Five neural refits assess training variation; RF-SKL uses one fixed-seed fit. Dataset construction, predictor settings, and training rules are given in Appendices C.1.1, C.2.1, and C.4.2.

4.2 Matched-Budget Revision Efficiency (RQ1)

Matched revision policies. Fixed h_0 performs no revision. Flat retry selects among up to 12 unseen one-step neighbors using validation feedback but without a typed edit address. Random routing samples uniformly over all addresses with an unseen legal neighbor, followed by the same deterministic address-to-edit mapping used by CellScientist. The AIDE-style shared-language policy retains draft-then-greedy search (Jiang et al., 2025): after h_0, it allocates up to five evaluations to independent drafts and the remaining budget to improvement. Each revising policy evaluates exactly B candidates. CellScientist ranks addresses using current diagnostics and HRT history, presents at most two eligible address–candidate pairs to Gemini 3 Pro, and applies LCA checks before evaluating the selected candidate. Capacity priorities use improvements over otherwise identical lower-rank candidates, following the rules in Appendix C.4.4. The comparison measures revision decisions within this shared executable language.

Useful candidates arrive earlier. Across the four task/split settings, CellScientist attains higher mean PCC than all comparison policies at $B = 3$ and $B = 5$ (Table 1). Against Random routing, the mean gains are 0.0361 and 0.0373. At $B = 3$, CellScientist SDs range from 0.0004 to 0.0016, versus 0.0413 to 0.0536 for Random routing. These comparisons evaluate the combined diagnostic-priority, history-filtering, and LLM-selection policy within the fixed language.

At $B = 10$, CellScientist exceeds Random routing in three settings and has a similar mean on BBBC036–SMILES (0.2346 versus 0.2347). Its average advantage over Random routing narrows to 0.0043, while the advantage over AIDE-style search remains 0.0181 (95% CI $[0.0052, 0.0309]$). Normalized frontier AUC summarizes predictive quality across the evaluation budget. CellScientist improves this measure over Fixed h_0, Flat retry, Random routing, and AIDE-style search by 0.0877, 0.0608, 0.0237, and 0.0215, with respective 95% CIs $[0.0680, 0.1074]$, $[0.0414, 0.0802]$, $[0.0172, 0.0303]$, and $[0.0117, 0.0314]$. Together, the budget curve and endpoint results identify the clearest benefit as faster discovery of useful candidates.

The budget pattern recurs across backbones. Replacing Gemini 3 Pro with Qwen2.5-0.5B-Instruct preserves the early-budget pattern (Table 2): gains are 0.0364, 0.0341, and 0.0012 at $B = 3, 5, 10$. Both backbones improve normalized frontier AUC. The four-setting intervals summarize configuration differences; Plate and SMILES reuse cohorts. Within-setting paired-seed intervals include zero at $B = 3, 5$ (Appendix D.1). Appendix C.4.5 records controller calls and fallback accounting.

Table 2: **The early-budget advantage recurs with an open-weight controller.** CellScientist minus Random routing: mean effect and descriptive four-setting 95% t interval, with five paired runs per setting. AUC is normalized frontier AUC.

Controller backbone	$B = 3$	$B = 5$	$B = 10$	AUC
Gemini 3 Pro	$+0.0361$ $[+0.0275, +0.0448]$	$+0.0373$ $[+0.0292, +0.0454]$	$+0.0043$ $[-0.0018, +0.0104]$	$+0.0237$ $[+0.0172, +0.0303]$
Qwen2.5-0.5B-Instruct	$+0.0364$ $[+0.0261, +0.0467]$	$+0.0341$ $[+0.0256, +0.0426]$	$+0.0012$ $[-0.0032, +0.0056]$	$+0.0217$ $[+0.0158, +0.0275]$

4.3 Operational Roles of History, Checks, and Routing (RQ2)

The three components act at different points in a revision: HRT makes previous attempts available to proposal generation, LCA determines whether a candidate remains eligible, and PDR assigns a discrepancy to a component to edit. We examine each operation with its corresponding measured outcome (Table 3).

History prevents repeated proposals. Twenty matched Gemini-controlled runs vary only persistent proposal history under the same trained-candidate budget. Without history, repeated IDs are recorded and replaced by legal unseen candidates before evaluation. HRT eliminates repeated proposals: all 170 proposals are unique, compared with 82 of 180 without history. This measures the use of prior attempts at proposal time while keeping the number of trained candidates fixed.

Contract checks protect evaluation. The LCA audit injects 40 registered contract-record violations alongside 20 legal controls. It repairs 15 records, blocks 15 at the pre-run gate, rejects 10 injected runtime-output records, and accepts all legal controls. Across these specified violation families, 30 requests are corrected or blocked before evaluation.

Discrepancy routing localizes repairs. The PDR audit uses five calibration cases and 15 held-out typed-state faults, with three variants per address and constructed metric records. Repair means restoring the registered configuration condition. Structured routing restores all 15 in one round, with no unrelated edits; Flat retry and Random routing restore 33.3% and 60.0%. Appendix E specifies the cases and repair criteria. Predictive revision efficiency is measured separately in RQ1.

Table 3: **Operational audits of HRT, LCA, and PDR. (a)** HRT: 20 matched Gemini-controlled runs, 200 evaluations, and 170/180 proposal steps with/without history. **(b)** LCA: 40 injected violations and 20 legal controls; 30 pre-dispatch interventions. **(c)** PDR: 15 held-out typed-state faults; rounds are capped at the five-round budget. Best values are **bolded**.

(a) HRT history.

Measure	HRT	w/o Hist.
Unique proposals ↑	**170**	82
Repeated proposals ↓	**0**	98
Repeat rate ↓	**0.0%**	54.4%
Unique ratio ↑	**100.0%**	45.6%

(b) LCA outcomes.

Outcome	Cases	Result
Repair	15	15/15 repaired
Pre-run block	15	15/15 blocked
Post-run block	10	10/10 blocked
Legal controls	20	20/20 accepted

(c) PDR held-out results.

Policy	Top-1	Repair	Edits	Rounds
Flat	13.3%	33.3%	3.20	4.13
Random	6.7%	60.0%	3.27	3.87
Structured	**100%**	**100%**	**0.00**	**1.00**

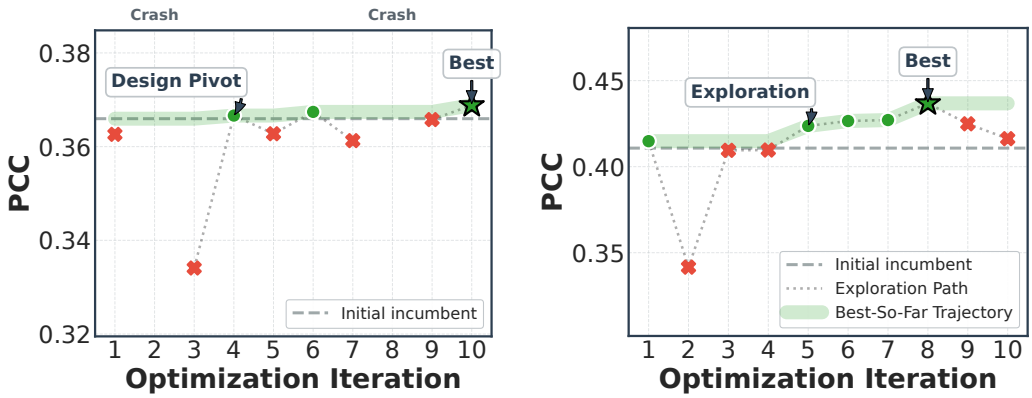
4.4 Frozen Designs and Recorded Revision Trajectories (RQ3)

External predictive utility. LKCP evaluates whether the frozen discovered architectures retain predictive utility after refitting on a separately acquired cohort with 1,671 morphology features. On its compound-disjoint test set, CS-model (BBBC047) attains the highest mean PCC among the evaluated predictors, 0.1766 versus 0.1496 for RealMLP. CS-model (BBBC036) attains the lowest mean MSE and highest mean R^2 (Table 4). Both frozen architectures exceed the early-fusion and control-only MLPs in mean PCC. The designs trade correlation against absolute error; seed-matched differences from RealMLP have intervals spanning zero (Appendix D.1).

From an attempted revision to a retained model. The open-workflow records show how these designs were reached (Figure 4). On BBBC036, attempts 2 and 8 crash and produce no PCC; the retained incumbent remains available for subsequent revisions. Several executable candidates also regress before the trajectory reaches DD-MIA. On BBBC047, the revision to a deterministic control-reference encoding produces Ref-RGAF, whereas a later LoRA-FiLM attempt performs worse. Retaining the best eligible state therefore gives a different outcome from taking the last attempted model. Appendices B.1 and B.2 link these changes to the recorded diagnostics and implementations.

Table 4: **External test of frozen CellScientist designs on LKCP Batch 2.** Compound-disjoint test results on 16 compounds (432 conditions). Entries are mean±sample SD over five refits; RF-SKL has one fixed-seed fit. CS-model (BBBC036/047) denotes the frozen DD-MIA/Ref-RGAF design, respectively. Best mean values among the evaluated methods are **bolded**.

Method	PCC↑	MSE↓	R^2↑
CS-model (BBBC036)	$0.1449_{\pm 0.0555}$	$\mathbf{4.4920}_{\pm 0.0696}$	$\mathbf{-0.0080}_{\pm 0.0082}$
CS-model (BBBC047)	$\mathbf{0.1766}_{\pm 0.0536}$	$4.5615_{\pm 0.1420}$	$-0.0561_{\pm 0.0424}$
RealMLP	$0.1496_{\pm 0.0582}$	$4.5032_{\pm 0.1076}$	$-0.0086_{\pm 0.0238}$
RF-SKL	0.0590	4.8402	−0.1034
Early-fusion MLP	$0.0860_{\pm 0.0020}$	$4.5630_{\pm 0.0022}$	$-0.0184_{\pm 0.0011}$
Control-only MLP	$0.0852_{\pm 0.0025}$	$4.5637_{\pm 0.0026}$	$-0.0184_{\pm 0.0014}$



(a) BBBC036. (b) BBBC047.

Figure 4: **High-freedom workflow trajectories.** Valid candidate PCC and retained best-so-far over ten revisions. BBBC036 crashes at steps 2 and 8 yield no PCC; separate crash markers denote these failures, and retained best remains unchanged. Dashed references mark the incoming incumbent.

The complete search outcomes are retained in Appendix F.1: morphology validation results cover BBBC036, BBBC047, and CPG0016, and independent task-specific searches cover LINCS2020 expression prediction and the Norman and Papalexi single-cell readouts. Appendix F.2 additionally records execution success, rule-compliant success, and runtime across cumulative workflow configurations. Together, these records document how the workflow proposes, evaluates, and retains models across cellular response tasks.

5 Conclusion

CellScientist converts execution and validation feedback into local model revisions under a fixed task contract. In the shared morphology candidate language, structured routing finds useful candidates earlier under two LLM backbones, highlighting the value of directing limited revision budgets toward components implicated by observed discrepancies. Operational audits further characterize the distinct roles of history, contract checks, and discrepancy routing by measuring repeated proposals, contract-record decisions, and registered fault repairs. External refits compare the predictive utility of two frozen designs on an independently acquired cohort, while complete revision trajectories retain failed, regressive, and validation-retained attempts behind model selection. Together, these results show how feedback-guided model development can produce not only an executable predictor, but also an inspectable record linking design decisions, implementation changes, and observed outcomes. Broader candidate languages, richer diagnostic signals, and cross-task reuse of revision knowledge are natural extensions of this formulation.

AI USE STATEMENT

CellScientist uses generative AI to propose and revise executable models and route diagnostics to local edits, as described in Section 3 and Appendix C.4. The experiments use Gemini 3 Pro and Qwen2.5-0.5B-Instruct. OpenAI Codex assisted with manuscript revision, language editing, citation formatting, and figure and layout preparation.

REPRODUCIBILITY STATEMENT

The operational algorithm and notation are provided in Appendix A. Dataset definitions, baseline protocols, evaluation metrics, implementation settings, and controller configurations are detailed in Appendices C.1–C.4. The case studies retain failed and regressive candidates alongside validation-selected outcomes, while the registered LCA and held-out PDR audits are documented in Appendix E.

REFERENCES

John Arevalo, Ellen Su, Jessica D. Ewald, Robert van Dijk, Anne E. Carpenter, and Shantanu Singh. Evaluating batch correction methods for image-based cell profiling. *Nature Communications*, 15: 6516, 2024. doi: 10.1038/s41467-024-50613-5.

Mark-Anthony Bray, Sigrun M. Gustafsdottir, Mohammad H. Rohban, Shantanu Singh, Vebjorn Ljosa, Katherine L. Sokolnicki, Joshua A. Bittker, et al. A dataset of images and morphological profiles of 30,000 small-molecule treatments using the Cell Painting assay. *GigaScience*, 6(12): 1–5, 2017. doi: 10.1093/gigascience/giw014.

Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. doi: 10.1023/A:1010933404324.

Jun Shern Chan, Neil Chowdhury, Oliver Jaffe, James Aung, Dane Sherburn, Evan Mays, Giulio Starace, Kevin Liu, Leon Maksin, Tejal Patwardhan, Lilian Weng, and Aleksander Madry. MLE-bench: Evaluating machine learning agents on machine learning engineering. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=6s5uXNWGIh.

Srinivas Niranj Chandrasekaran, Jeanelle Ackerman, Eric Alix, D Michael Ando, John Arevalo, Melissa Bennion, Nicolas Boisseau, Adriana Borowa, Justin D Boyd, Laurent Brino, et al. JUMP Cell Painting dataset: morphological impact of 136,000 chemical and genetic perturbations. *bioRxiv*, 2023. doi: 10.1101/2023.03.23.534023. URL https://www.biorxiv.org/content/10.1101/2023.03.23.534023v1.

Edward De Brouwer, Carl Edwards, Alexander Wu, Jenna Collier, Graham Heimberg, Xiner Li, Meena Subramaniam, Ehsan Hajiramezanali, David Richmond, Jan-Christian Hütter, Sara Mostafavi, and Gabriele Scalia. AssayBench: An assay-level virtual cell benchmark for LLMs and agents. *arXiv preprint arXiv:2605.10876*, 2026. doi: 10.48550/arXiv.2605.10876. URL https://arxiv.org/abs/2605.10876.

Jinghao Feng, Ziheng Zhao, Xiaoman Zhang, Mingfei Liu, Jingyi Chen, Xingran Quan, Boyang Fu, Jian Zhang, Yanfeng Wang, Ya Zhang, and Weidi Xie. A knowledge-driven framework for predicting single-cell responses for unprofiled drugs. *Nature Machine Intelligence*, 8(9):1478–1491, 2026. doi: 10.1038/s42256-026-01286-w. URL https://www.nature.com/articles/s42256-026-01286-w.

Pieter Gijsbers, Marcos LP Bueno, Stefan Coors, Erin LeDell, Sébastien Poirier, Janek Thomas, Bernd Bischl, and Joaquin Vanschoren. AMLB: an AutoML benchmark. *Journal of Machine Learning Research*, 25(101):1–65, 2024. URL https://www.jmlr.org/papers/v25/22-0493.html.

Yury Gorishniy, Ivan Rubachev, Nikolay Kartashev, Daniil Shlenskii, Akim Kotelnikov, and Artem Babenko. TabR: Tabular deep learning meets nearest neighbors. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=rhgIgTSSxW.

Yue Guo, Hao Zhang, Haitao Hu, Jialu Wu, Ji Cao, Chang-Yu Hsieh, and Bo Yang. Modelling drug-induced cellular perturbation responses with a biologically informed dual-branch transformer. *Nature Machine Intelligence*, 8(1):96–112, 2026. doi: 10.1038/s42256-025-01165-w. URL https://www.nature.com/articles/s42256-025-01165-w.

Euxhen Hasanaj, Elijah Cole, Shahin Mohammadi, Sohan Addagudi, Xingyi Zhang, Le Song, and Eric P. Xing. Multimodal benchmarking of foundation model representations for cellular perturbation response prediction. *bioRxiv*, 2025. doi: 10.1101/2025.06.26.661186. URL https://www.biorxiv.org/content/10.1101/2025.06.26.661186v2. Preprint, version 2.

David Holzmüller, Léo Grinsztajn, and Ingo Steinwart. Better by default: Strong pre-tuned MLPs and boosted trees on tabular data. In *Advances in Neural Information Processing Systems*, volume 37, pp. 26577–26658, 2024. doi: 10.52202/079017-0837. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/2ee1c87245956e3eaa71aaba5f5753eb-Abstract-Conference.html.

Qian Huang, Jian Vora, Percy Liang, and Jure Leskovec. MLAgentBench: Evaluating language agents on machine learning experimentation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 20271–20309. PMLR, 2024. URL https://proceedings.mlr.press/v235/huang24y.html.

Wenxuan Huang, Mingyu Tsoi, Yanhao Huang, Xinjie Mao, Xue Xia, Hao Wu, Jiaqi Wei, Yuejin Yang, Lang Yu, Cheng Tan, et al. HarmonyCell: Automating single-cell perturbation modeling under semantic and distribution shifts. *arXiv preprint arXiv:2603.01396*, 2026. URL https://arxiv.org/abs/2603.01396.

Zhengyao Jiang, Dominik Schmidt, Dhruv Srikanth, Dixing Xu, Ian Kaplan, Deniss Jacenko, and Yuxiang Wu. AIDE: AI-driven exploration in the space of code. *arXiv preprint arXiv:2502.13138*, 2025. URL https://arxiv.org/abs/2502.13138.

Bo Li, Qing Wang, Shihang Wang, Bob Zhang, Yuzhong Peng, Pinxian Zeng, Chengliang Liu, Mengran Li, Ziyang Tang, Xiaojun Yao, Chuxia Deng, and Qianqian Song. MVCBench: A multimodal benchmark for drug-induced virtual cell phenotypes. *bioRxiv*, 2026a. doi: 10.64898/2026.04.22.720110. URL https://www.biorxiv.org/content/10.64898/2026.04.22.720110v1. Preprint, version 1.

Krinos Li, Xianglu Xiao, Shenglong Deng, Lucas He, Zijun Zhong, Yuanjie Zou, Zhonghao Zhan, Zheng Hui, Weiye Bao, and Guang Yang. Large language models meet virtual cell: A survey. *arXiv preprint arXiv:2510.07706*, 2025. doi: 10.48550/arXiv.2510.07706. URL https://arxiv.org/abs/2510.07706.

Mengran Li, Zelin Zang, Wenbin Xing, Junzhou Chen, Ronghui Zhang, Jiebo Luo, and Stan Z. Li. Learning cell-aware hierarchical multi-modal representations for robust molecular modeling. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(1):623–631, 2026b. doi: 10.1609/aaai.v40i1.37027. URL https://ojs.aaai.org/index.php/AAAI/article/view/37027.

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. The AI scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*, 2024. URL https://arxiv.org/abs/2408.06292.

Henry E Miller, Matthew Greenig, Benjamin Tenmann, and Bo Wang. BioML-bench: Evaluation of AI agents for end-to-end biomedical ML. *bioRxiv*, 2025. doi: 10.1101/2025.09.01.673319. URL https://www.biorxiv.org/content/10.1101/2025.09.01.673319v2.

Jaehyun Nam, Jinsung Yoon, Jiefeng Chen, Jinwoo Shin, Sercan Arik, and Tomas Pfister. MLE-STAR: Machine learning engineering agent via search and targeted refinement. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-3889. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/a9619dd0f0d54a5cf7734add1dc38cd1-Abstract-Conference.html.

Ted Natoli, Gregory Way, Xiaodong Lu, Beth Cimini, David Logan, Kyle Karhohs, Juan Caicedo, Maria Alimova, Kate Hartland, Todd Golub, Anne Carpenter, Shantanu Singh, and Aravind Subramanian. broadinstitute/lincs-cell-painting: Full release of LINCS Cell Painting dataset. Zenodo, 2021. URL https://zenodo.org/records/5008187. Version v1.

Thomas M. Norman, Max A. Horlbeck, Joseph M. Replogle, Alex Y. Ge, Albert Xu, Marco Jost, Luke A. Gilbert, and Jonathan S. Weissman. Exploring genetic interaction manifolds constructed from rich single-cell phenotypes. *Science*, 365(6455):786–793, 2019. doi: 10.1126/science.aax4438.

Giovanni Palla, Alexander Hillsley, Yang-Joon Kim, and Loic A. Royer. Tabular foundation models are competitive cellular perturbation predictors across biological scales. *bioRxiv*, 2026. doi: 10.64898/2026.06.28.735106. URL https://www.biorxiv.org/content/10.64898/2026.06.28.735106v3. Preprint, version 3.

Efthymia Papalexi, Eleni P Mimitou, Andrew W Butler, Samantha Foster, Bernadette Bracken, William M. Mauck, III, Hans-Hermann Wessels, Yuhan Hao, Bertrand Z Yeung, Peter Smibert, et al. Characterizing the molecular regulation of inhibitory immune checkpoints with multimodal single-cell screens. *Nature Genetics*, 53(3):322–331, 2021. doi: 10.1038/s41588-021-00778-2.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12(85):2825–2830, 2011. URL https://www.jmlr.org/papers/v12/pedregosa11a.html.

Thai-Hoang Pham, Yue Qiu, Jucheng Zeng, Lei Xie, and Ping Zhang. A deep learning framework for high-throughput mechanism-driven phenotype compound screening and its application to COVID-19 drug repurposing. *Nature Machine Intelligence*, 3:247–257, 2021. doi: 10.1038/s42256-020-00285-9.

Thai-Hoang Pham, Yue Qiu, Jiahui Liu, Steven Zimmer, Eric O'Neill, Lei Xie, and Ping Zhang. Chemical-induced gene expression ranking and its application to pancreatic cancer drug repurposing. *Patterns*, 3(4):100441, 2022. doi: 10.1016/j.patter.2022.100441.

Yibo Qiu, Zan Huang, Zhiyu Wang, Handi Liu, Yiling Qiao, Yifeng Hu, Shu'ang Sun, Hangke Peng, Ronald X Xu, and Mingzhai Sun. BioMARS: A multi-agent robotic system for autonomous biological experiments. *arXiv preprint arXiv:2507.01485*, 2025. URL https://arxiv.org/abs/2507.01485.

Yusuf Roohani, Kexin Huang, and Jure Leskovec. Predicting transcriptional outcomes of novel multigene perturbations with GEARS. *Nature Biotechnology*, 42:927–935, 2024. doi: 10.1038/s41587-023-01905-6. URL https://www.nature.com/articles/s41587-023-01905-6.

Srijit Seal, Maria-Anna Trapotsi, Ola Spjuth, Shantanu Singh, Jordi Carreras-Puigvert, Nigel Greene, Andreas Bender, and Anne E Carpenter. Cell Painting: a decade of discovery and innovation in cellular imaging. *Nature Methods*, 22(2):254–268, 2025. doi: 10.1038/s41592-024-02528-8.

Vasileios Stathias, John Turner, Amar Koleti, Dusica Vidovic, Daniel Cooper, Mehdi Fazel-Najafabadi, Marcin Pilarczyk, Raymond Terryn, Caty Chung, Afoma Umeano, et al. LINCS data portal 2.0: next generation access point for perturbation-response signatures. *Nucleic Acids Research*, 48(D1):D431–D439, 2020. doi: 10.1093/nar/gkz1023.

David R Stirling, Madison J Swain-Bowden, Alice M Lucas, Anne E Carpenter, Beth A Cimini, and Allen Goodman. CellProfiler 4: improvements in speed, utility and usability. *BMC Bioinformatics*, 22(1):433, 2021. doi: 10.1186/s12859-021-04344-9.

Aravind Subramanian, Rajiv Narayan, Steven M Corsello, David D Peck, Ted E Natoli, Xiaodong Lu, Joshua Gould, John F Davis, Andrew A Tubelli, Jacob K Asiedu, et al. A next generation connectivity map: L1000 platform and the first 1,000,000 profiles. *Cell*, 171(6):1437–1452.e17, 2017. doi: 10.1016/j.cell.2017.10.049.

Xiangru Tang, Zhuoyun Yu, Jiapeng Chen, Yan Cui, Daniel Shao, Weixu Wang, Fang Wu, Yuchen Zhuang, Wenqi Shi, Zhi Huang, et al. CellForge: agentic design of virtual cell models. *arXiv preprint arXiv:2508.02276*, 2025. URL https://arxiv.org/abs/2508.02276.

Xiaochu Tong, Ning Qu, Xiangtai Kong, Shengkun Ni, Jingyi Zhou, Kun Wang, Lehan Zhang, Yiming Wen, Jiangshan Shi, Sulin Zhang, et al. Deep representation learning of chemical-induced transcriptional profile for phenotype-based drug discovery. *Nature Communications*, 15(1):5378, 2024. doi: 10.1038/s41467-024-49620-3.

Alistair Turcan, Kexin Huang, Lei Li, and Martin Jinye Zhang. TusoAI: Agentic optimization for scientific methods. *arXiv preprint arXiv:2509.23986*, 2025. URL https://arxiv.org/abs/2509.23986.

Ramon Viñas Torné, Maciej Wiatrak, Zoe Piran, Shuyang Fan, Liangze Jiang, Sarah A. Teichmann, Mor Nitzan, and Maria Brbić. Systema: a framework for evaluating genetic perturbation response prediction beyond systematic variation. *Nature Biotechnology*, 44(6):1050–1059, 2026. doi: 10.1038/s41587-025-02777-8. URL https://www.nature.com/articles/s41587-025-02777-8.

Chi Wang, Qingyun Wu, Markus Weimer, and Erkang Zhu. FLAML: A fast and lightweight AutoML library. In *Proceedings of Machine Learning and Systems*, volume 3, pp. 434–447, 2021. URL https://proceedings.mlsys.org/paper_files/paper/2021/hash/1ccc3bfa05cb37b917068778f3c4523a-Abstract.html.

Hanchen Wang, Yichun He, Paula P Coelho, Matthew Bucci, Abbas Nazir, Bob Chen, Linh Trinh, Serena Zhang, Kexin Huang, Vineethkrishna Chandrasekar, et al. SpatialAgent: An autonomous AI agent for spatial biology. *bioRxiv*, 2025a. doi: 10.1101/2025.04.03.646459. URL https://www.biorxiv.org/content/10.1101/2025.04.03.646459v1.

Wenbo Wang, Simran Swain, Jaeyong Lee, Zuwan Lin, Bradley Canales, Almir Aljović, Yaxuan Liu, Qiang Li, Arnau Marin-Llobet, Mai Liu, et al. Agentic lab: An agentic-physical AI system for cell and organoid experimentation and manufacturing. *bioRxiv*, 2025b. doi: 10.1101/2025.11.11.686354. URL https://www.biorxiv.org/content/10.1101/2025.11.11.686354v1.

Gregory P. Way, Ted Natoli, Adeniyi Adeboye, Lev Litichevskiy, Andrew Yang, Xiaodong Lu, Juan C. Caicedo, Beth A. Cimini, Kyle Karhohs, David J. Logan, Mohammad H. Rohban, Maria Kost-Alimova, Kate Hartland, Michael Bornholdt, Srinivas Niranj Chandrasekaran, Marzieh Haghighi, Erin Weisbart, Shantanu Singh, Aravind Subramanian, and Anne E. Carpenter. Morphology and gene expression profiling provide complementary information for mapping cell state. *Cell Systems*, 13(11):911–923.e9, 2022. doi: 10.1016/j.cels.2022.10.001.

Zhijian Wei, Runze Ma, Zichen Wang, Zhongmin Li, Shuotong Song, and Shuangjia Zheng. VCWorld: A biological world model for virtual cell simulation. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=hhq89Hs7T3.

Erin Weisbart, Ankur Kumar, John Arevalo, Anne E. Carpenter, Beth A. Cimini, and Shantanu Singh. Cell Painting Gallery: an open resource for image-based profiling. *Nature Methods*, 21(10):1775–1777, 2024. doi: 10.1038/s41592-024-02399-z.

Yan Wu, Esther Wershof, Sebastian M. Schmon, Marcel Nassar, Błażej Osiński, Ridvan Eksi, Zichao Yan, Rory Stark, Kun Zhang, and Thore Graepel. PerturBench: Benchmarking machine learning models for cellular perturbation analysis. In *Advances in Neural Information Processing Systems*, volume 38, pp. 106937–106977. Neural Information Processing Systems Foundation, Inc., 2025. doi: 10.52202/085713-3225. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/8aee537279a66ced96319dfca3c00002-Abstract-Datasets_and_Benchmarks_Track.html.

You Wu, Qiao Liu, Yue Qiu, and Lei Xie. Deep learning prediction of chemical-induced dose-dependent and context-specific multiplex phenotype responses and its application to personalized Alzheimer's disease drug repurposing. *PLOS Computational Biology*, 18(8):e1010367, 2022. doi: 10.1371/journal.pcbi.1010367.

Zhuorui Xie, Eryk Kropiwnicki, Megan L Wojciechowicz, Kathleen M Jagodnik, Ingrid Shu, Allison Bailey, Daniel JB Clarke, Minji Jeon, John Erol Evangelista, Maxim V. Kuleshov, et al. Getting started with LINCS datasets and tools. *Current Protocols*, 2(7):e487, 2022. doi: 10.1002/cpz1.487.

Xi Yu, Yang Yang, Qun Liu, Yonghua Du, Sean McSweeney, and Yuewei Lin. GenCellAgent: Generalizable, training-free cellular image segmentation via large language model agents. *arXiv preprint arXiv:2510.13896*, 2025. URL https://arxiv.org/abs/2510.13896.

Ronghui Zhang, Mengran Li, Wenbin Xing, Bo Li, Chengyang Zhang, Wenxuan Tu, Yongfu Li, and Ruxin Wang. MORSE: Molecular representation learning via structured semantic extraction across hierarchical and asymmetric biological modalities. *Pattern Recognition*, 174:112975, 2026. doi: 10.1016/j.patcog.2025.112975.

Zhaojun Zhang, Divij Mathew, Tristan L. Lim, Kaishu Mason, Clara Morral Martinez, Sijia Huang, E. John Wherry, Katalin Susztak, Andy J. Minn, Zongming Ma, et al. Recovery of biological signals lost in single-cell batch integration with CellANOVA. *Nature Biotechnology*, 43:1861–1877, 2025. doi: 10.1038/s41587-024-02463-1.

Yang Zhou, Qiongyu Sheng, Guohua Wang, Li Xu, and Shuilin Jin. Quantifying batch effects for individual genes in single-cell data. *Nature Computational Science*, 5:612–620, 2025. doi: 10.1038/s43588-025-00824-7.

Jie Zhu, Jingxiang Wang, Xin Wang, Mingjing Gao, Bingbing Guo, Miaomiao Gao, Jiarui Liu, Yanqiu Yu, Liang Wang, Weikaixin Kong, et al. Prediction of drug efficacy from transcriptional profiles with deep learning. *Nature Biotechnology*, 39(11):1444–1452, 2021. doi: 10.1038/s41587-021-00946-z.

A ALGORITHM AND NOTATION

Coupled model-design revision. Perturbation-response models combine interdependent choices in perturbation representation, cellular-context encoding, dose conditioning, feature interaction, objective design, and optimization. Changing one component can alter the behavior of others. Model revision therefore requires associating observed outcomes with candidate explanations and possible edit targets. An explicit record of component dependencies, accumulated evidence, and edit scope guides this process.

Hypothesis–implementation alignment. Evaluating a model-design hypothesis requires an executable implementation that preserves the task's data, prediction target, interface, and evaluation semantics. Execution and validation outcomes then inform which design assumptions and components merit reconsideration. We formulate this as a *model-design revision-routing problem*: mapping observed discrepancies to local design revisions while maintaining a fixed task contract.

Algorithm 1: CellScientist Workflow

Require: Fixed task specification $\mathcal{D}_{\text{spec}}$, initial model-design state $\mathbf{h}_0$, task-state graph $\mathcal{G}_0$, iteration budget T, repair budget R

Initialize run history $\mathcal{B}_0 \leftarrow \emptyset$ and best eligible state $(\mathbf{h}_{\text{best}}, \mathbf{c}_{\text{best}}) \leftarrow (\mathbf{h}_0, \perp)$
for $t = 0$ to $T - 1$ **do**
  Summarize the current model-design state, task-state graph, and run history
  Request a focus area and candidate edits $\tilde{\mathbf{c}}_t \leftarrow \Phi(\mathbf{h}_t, \mathcal{G}_t, \mathcal{B}_t)$
  Apply edits and configured checks with at most R repair attempts: $\mathbf{c}_t \leftarrow \Gamma_{\mathcal{D}}(\tilde{\mathbf{c}}_t; \mathbf{h}_t, \mathcal{G}_t, \mathcal{B}_t, R)$
  if $\mathbf{c}_t \neq \perp$ **then**
    Execute $\mathbf{c}_t$ under the validation view of $\mathcal{D}_{\text{spec}}$
    Record execution status, configured rule checks, and validation feedback $\mathbf{m}_t$
    Set $\eta_t \leftarrow 1$
  else
    Record execution or required-output failure in $\mathbf{m}_t$ and set $\eta_t \leftarrow 0$
  end if
  Route the diagnostic to an edit address $a_t \leftarrow \mathcal{A}(\mathbf{m}_t, \mathcal{G}_t, \mathcal{B}_t)$
  Form the next design record $\mathbf{h}_{t+1} \leftarrow \mathbf{h}_t \oplus \mathcal{T}_{\text{edit}}(a_t \mid \mathbf{m}_t)$
  Append the addressed evidence and status to obtain $\mathcal{G}_{t+1}$
  Update run history $\mathcal{B}_{t+1} \leftarrow \mathcal{B}_t \cup \{(\mathbf{h}_t, \tilde{\mathbf{c}}_t, \mathbf{c}_t, \mathbf{m}_t, a_t, \eta_t)\}$
  if $\eta_t = 1$ and $\mathbf{c}_t$ improves the configured validation score **then**
    $(\mathbf{h}_{\text{best}}, \mathbf{c}_{\text{best}}) \leftarrow (\mathbf{h}_t, \mathbf{c}_t)$
  end if
end for
return $(\mathbf{h}_{\text{best}}, \mathbf{c}_{\text{best}})$ and the recorded trajectory $(\mathcal{G}_T, \mathcal{B}_T)$

Table 5: **Notation used in CellScientist.**

Symbol	Description
$\mathcal{H}$	Model-design space containing task-specific representation, fusion, objective, and training choices
$\mathcal{I}$	Space of executable candidate programs
$\mathcal{D}_{\text{spec}}$	Fixed task specification $(\mathcal{X}, \mathcal{P}, \mathcal{Y}, \mathcal{S}, \mathcal{E})$
$\mathcal{X}, \mathcal{P}, \mathcal{Y}$	Cellular inputs, perturbation descriptors, and response targets
$\mathcal{S}$	Data-partition protocol used by the configured task
$\mathcal{E}$	Evaluation functional used for candidate scoring and selection
$\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{val}}, \mathcal{D}_{\text{test}}$	Training, controller-feedback/selection, and reporting views induced by $\mathcal{S}$
$\mathbf{h}_t$	Model-design record at iteration t
$\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t)$	Structured task-state graph used for routing context and audit
$\mathcal{V}_t, \mathcal{E}_t$	Task nodes and directed dependency edges
$\mathcal{B}_t$	Run history storing attempted designs, implementations, and outcomes
$\mathcal{Z}_t = (\mathbf{h}_t, \mathcal{G}_t, \mathcal{B}_t)$	Complete controller state at iteration t
T, R	Candidate-iteration budget and bounded LCA repair budget
Φ	LLM-conditioned candidate-proposal operator
$\Gamma_{\mathcal{D}}$	Code-application, configured-check, bounded-repair, and rejection procedure
$\tilde{\mathbf{c}}_t$	Proposed candidate edits before execution checks
$\mathbf{c}_t$	Candidate produced after code application and checks
$\perp$	Candidate that failed configured execution or required-output checks
η_t	Indicator that $\mathbf{c}_t$ passed the configured eligibility checks
Ω_{const}	Candidates that pass the implemented checks under $\mathcal{D}_{\text{spec}}$
$\chi_{\text{I}}, \mathcal{C}_{\text{I}}$	Eligibility indicator and its configured implementation-rule set
$\mathcal{F}$	Feedback extraction operator
$\mathbf{m}_t$	Structured feedback vector at iteration t
$\bar{\mathbf{h}}_t, \bar{\mathbf{m}}_t$	Feedback-best candidate configuration and its diagnostic record
L_t, Q_t, e_j	Ordered addresses, allow-listed pair menu, and deterministic one-field edit at address j
$\mathcal{A}, a_t$	Diagnostic-to-task association procedure and returned task address
$\mathcal{T}_{\text{edit}}$	Revision-request operator conditioned on recorded feedback
$\oplus$	Application of a revision to a model-design record
$\text{Select}_{\text{val}}$	Selection rule over eligible validation runs
$(\mathbf{h}_{\text{best}}, \mathbf{c}_{\text{best}})$	Validation-selected design record and executable candidate

B Detailed Case Studies of Auditable Model-Revision Trajectories

We reconstruct two model-revision trajectories on morphology-based perturbation-response tasks. Each record preserves execution errors, rule checks, validation outcomes, rejected candidates, and retained candidates under a fixed evaluation protocol. Phase 1–3 follow the pipeline logs: diagnosis and task setup, initial candidate generation, and subsequent review. HRT, LCA, and PDR operate throughout these phases.

On BBBC036, CellScientist retained DD-MIA (Dose-Disentangled Mechanism-Intensity Attention) after evaluating representation and fusion variants. Its exported implementation uses Morgan fingerprints, virtual-token cross-attention, fold-label adversarial regularization, and hidden-space chemical–dose gating. The architecture name records the design proposal; the executable operations are specified in Appendix C.4.1.

On BBBC047, modulation and VAE-style candidates showed both gains and regressions. The trajectory recorded possible underuse of the supplied control-reference profile as a working diagnostic. CellScientist subsequently retained Ref-RGAF, which adds a deterministic control-reference encoding to the decoder. The record links this diagnostic to a local revision and its validation outcome.

These cases make CellScientist's selection process inspectable: the records identify attempted assumptions, failed or regressive branches, and validation-selected executable candidates. They complement the quantitative results by tracing local revisions and selection decisions under a fixed protocol.

B.1 Detailed Case Study A

Case A: Auditable Revision Toward DD-MIA

Trajectory including negative results, crashes, regressions, and retained candidates.

Case Study Overview

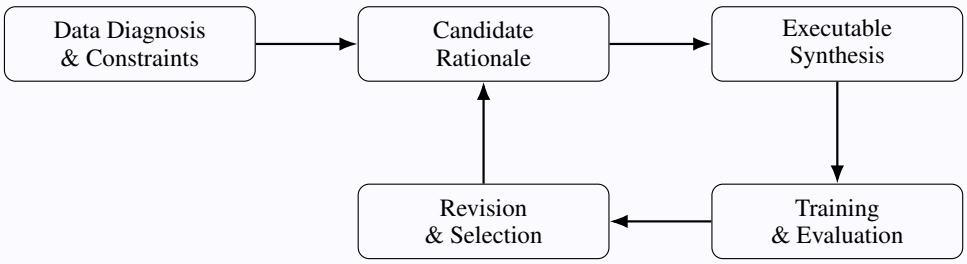
This section reconstructs the model-revision trajectory of **CellScientist** on BBBC036 under the SMILES-based split. The trajectory moves from simple feature-based modeling toward **DD-MIA**, whose learned chemical and dose gates modulate an attention update in hidden space.

What this case study illustrates. The trace records how CellScientist diagnosed data and batch-related issues, proposed design rationales, implemented candidates, and retained the highest-scoring eligible candidate under the validation protocol.

Trajectory evidence. The trajectory documents how recorded diagnostics and validation outcomes were associated with local design revisions and empirical candidate selection.

Closed-Loop Revision Trajectory

Auditable closed-loop design record:
diagnosis → executable candidate → validation feedback



Interpretation. The loop records a practical model-design process: *diagnose → propose → implement → test → revise*.

(A) From Representation Choice to Design Rationale The trajectory is non-monotonic: several candidates underperformed despite valid execution, and other candidates failed during execution. The complete record shows which components were retained, revised, or rejected at each step.

Table 9 summarizes representative scored candidates from the trajectory. The initial search retained a graph-based candidate, while several more complex fusion variants regressed or failed.

Key Observation

The trajectory preserves both validation-retained candidates and negative evidence. Several higher-capacity variants regressed, while the validation rule retained a more structured candidate. This record makes the empirical selection process auditable.

(B) Auditable Model-Revision Trajectory We report the diagnosis and revision phases, including failures and regressions. These records show how the validation rule selected the retained candidate.

Phase 1: Problem Formulation and Data Diagnosis. Before architecture revision, the system defines the prediction target and records operational constraints. Table 6 summarizes the data checks that informed later candidates.

Table 6: **Phase 1 Execution Trace: Diagnosis and Constraint Identification.**

Step	Activity	Artifact	Design / Operational Insight
1	Data Ingestion & Pairing	HDF5 Database	**Operational repair:** Resolved HDF5 sizing mismatch by enforcing type and shape alignment. Established the $f(\text{ControlRef}, \text{Chem}, \text{Dose}) \rightarrow \text{TreatedResponse}$ mapping used by later candidates.
2	Statistical Diagnosis	ANOVA/PCA	**Operational repair:** Corrected a syntax error in the statistical constraint formula. **Data cue:** Observed a measurable plate-related signal in the feature space, motivating later checks for batch-related shortcuts.
3	Dose-Response Profiling	Volcano Plot	**Response cue:** Identified features with strong perturbation response and observed non-linear, heterogeneous dose behavior. This suggested that a purely linear response model may be insufficient.
4	Technical-Factor Inspection	Assay-Variable Schematic	**Constraint definition:** The persistence of plate effects after normalization motivated the use of batch-related regularization in later candidates.

Key Observation

From diagnosis to design. ANOVA/PCA diagnostics associated part of the feature variation with technical factors. This recorded observation motivated batch-related regularization in later candidates.

Phase 2: Generate and Execute. Phase 2 evaluated candidate representations and recorded setup errors and unstable training (Table 7). The highest-scoring eligible candidate was recorded as a graph-based variant and exceeded the initial flat-vector candidate in validation PCC. Later revisions also changed the compound encoder; the selected DD-MIA export uses fingerprints. Phase 2 uses an exploratory baseline fit; Phase 3 aggregate gains use the benchmark reference.

Table 7: **Phase 2 Execution Trace: Structural Representation Trial.**

Iter	Strategy	Outcome	PCC	Design Analysis & Operational Event
1	Initial Debug / Baseline[a]	Baseline	0.3172	**Operational:** Pipeline stabilized through a setup-node repair. **Baseline:** Established a reference PCC using standard flat-vector features.
2	**EquiMorph-GAT**	**Retained**	**0.3660**	**Design cue:** Explicit molecular topology encoding improved over the initial flat-vector candidate in this trace, suggesting that compound-structure representation is useful for this task.
3	MD_AdvN (Adversarial)	Regression	0.3378	**Negative result:** Adversarial normalization alone obtained a lower validation PCC and was rejected by the validation rule.
4	Innovation Variant	Regression	0.3656	**Validation check:** A graph-based variant produced similar performance across a second implementation attempt.
5	Baseline Re-test	Regression	0.2827	**Regression:** The simplified baseline obtained a lower validation PCC in this trace; subsequent candidates retained the structured representation.

[a]Exploratory Phase 2 baseline; Phase 3 gains use the separately refit Phase 3 reference shown in Table 9.

Key Observation

Representation cue. The graph-based candidate improved over simplified flat-vector baselines within this recorded trajectory, providing task-specific validation evidence for retaining compound-structure encoding.

Phase 3: Review and Revise. Phase 3 evaluated how to combine the control-reference profile, compound representation, and dose information. Table 8 reports all iterations, including crashes and candidates rejected for lower validation performance.

Table 8: **Phase 3 Full Iteration Trace (Iter 1–10).**

Iter	Strategy	Outcome	PCC	Design Analysis
1	Chem-Modulated FiLM	Regression	0.3626	Candidate assumption: affine modulation is sufficient. Result: validation PCC was lower than the retained graph-based model.
2	Res-EquiMorph + Token	Crash	N/A	Candidate assumption: tokenization may improve interaction modeling. Execution failed with a CUDA class-index assertion; no valid PCC was produced.
3	Tokenized Adversarial	Regression	0.3341	Candidate assumption: finer granularity would improve robustness. Result: the tokenized variant obtained lower validation PCC.
4	DGAN (Gated-Adv)	Improved	0.3666	Design pivot: adversarial regularization was restricted to the morphology branch, while chemistry was used to gate the perturbation response.
5	Res-DGAN	Regression	0.3627	Negative add-on: an additional residual skip produced a lower validation PCC in this run.
6	Attn-DGAN (Attention)	Improved	0.3674	Refinement: cross-attention provided a more structured interaction between morphology and compound features.
7	Dose-Gated Attention	Regression	0.3613	Negative result: directly gating attention by dose reduced validation PCC in this candidate.
8	Dose-Conditioned Res	Crash	N/A	Execution failed with a CUDA class-index assertion; no valid PCC was produced.
9	Projected-Token Attn	Regression	0.3658	The candidate executed successfully but remained below the retained PCC of 0.3674.
10	**DD-MIA**	**Retained**	**0.3688**	Selected design: separates compound-structure representation from dose-dependent intensity scaling using multiplicative fusion.

Key Observation

Why failures matter. The record preserves crashes and regressions from direct dose injection into attention and more complex tokenized variants. These outcomes expose the sequence of attempted revisions and validation decisions.

(C) Validation-Selected Design: DD-MIA The retained **DD-MIA** candidate combines three implementation choices:
(1) Structural compound representation. The selected export encodes a 2,048-bit Morgan fingerprint with an MLP, then reshapes chemical and context embeddings into 16 virtual tokens of width 32 for four-head cross-attention.
(2) Fold-label adversarial regularization. Technical-variation diagnostics motivated a six-way adversarial head on the context encoding. Its labels are recorded fold IDs, with gradient-reversal strength 0.5 and cross-entropy weight 0.1.

Failure Mode / Design Risk

The diagnostic motivated regularization against technical variation; the implemented supervision is the fold label. This identifies the training objective separately from its intended effect.

(3) Chemistry–dose gating of the attention update. Let h_m, h_c, h_d encode context, fingerprint, and dose; a is the flattened context-to-chemical attention output. The selected predictor computes

$$u = \mathrm{LN}(h_m + a \odot \sigma(W_c h_c + b_c) \odot \sigma(W_d h_d + b_d)), \qquad \widehat{y} = D([u, h_c, h_d]). \tag{5}$$

The product acts on a 512-dimensional hidden update; the decoder returns the treated profile. Appendix C.4.1 gives all dimensions and losses.

Design Rationale Encoded as Architecture

The hidden gates encode a task-specific preference for multiplicative chemical–dose interaction. Retention follows the validation evidence recorded in this trajectory.

(D) Evaluation and Trajectory Audit The retained DD-MIA candidate achieved a five-fold validation aggregate PCC of 0.3688, corresponding to CS-model on BBBC036 in Table 28. The trajectory records the failed candidates, regressions, and retained components leading to this design.

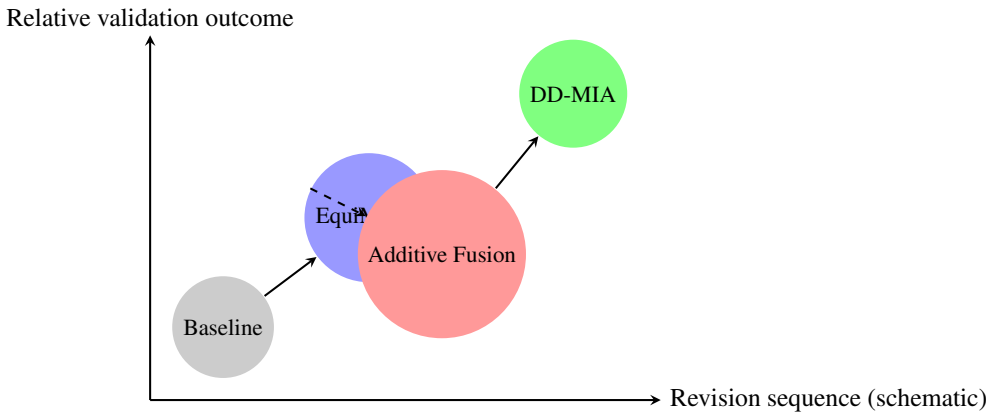


Figure 5: Auditable model-revision trajectory: CellScientist follows a non-monotonic path through candidate architectures, retaining validated components while recording underperforming branches.

Table 9: **Trajectory of Candidate Performance.**

Activity Step	Design Rationale	PCC	Δ vs Base	Status
Baseline[a]	MLP (Flat Vectors)	0.3396	0.00%	Ref
Initial Search	EquiMorph-GAT	0.3660	+7.77%	Retained candidate
Iter 1	Additive FiLM	0.3626	+6.77%	Regression
Iter 6	Attn-DGAN	0.3674	+8.18%	Improved
Iter 10	DD-MIA	**0.3688**	**+8.60%**	Retained candidate

[a]Phase 3 reference fit, separately refit from the exploratory Phase 2 baseline (0.3172); all Phase 3 gains in this table use 0.3396.

Key Observation

The trajectory records both validation-retained improvements and negative evidence. DD-MIA is the candidate retained for this benchmark under the reported validation-selection rule.

(E) Summary The BBBC036 trajectory connects representation choices, batch-related diagnostics, and fusion revisions to recorded execution and validation outcomes. CellScientist retained DD-MIA under the fixed selection rule while preserving failed and regressive alternatives.

B.2 Detailed Case Study B

The introduction summarizes the BBBC047 trajectory in Figure 2. The detailed record below links its working diagnostics, local revisions, validation outcomes, and retained candidate.

Case B: Auditable Revision Toward Ref-RGAF

Trajectory including negative results, crashes, regressions, and retained candidates.

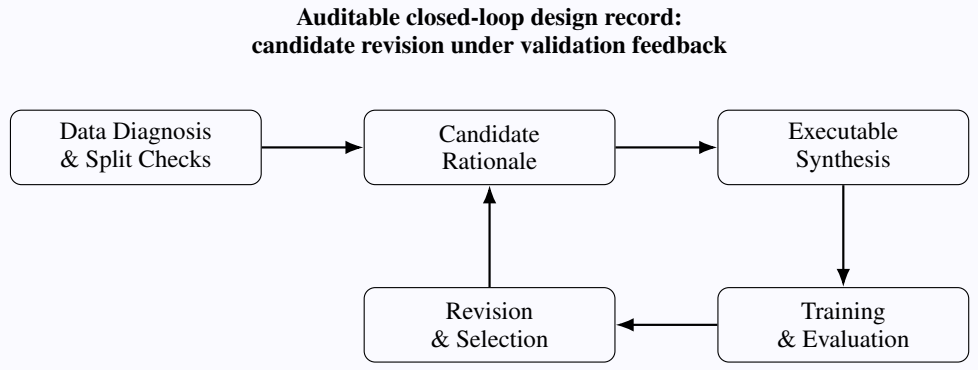
Case Study Overview

This section reconstructs the model-revision trajectory of **CellScientist** on BBBC047 under the SMILES-based split. The trajectory moves from standard generative baselines toward **Ref-RGAF** (Reference-Gated Affine Fusion), a candidate architecture that adds a deterministic control-reference path to the decoder.

What this case study illustrates. The trace records how the system evaluated modulation-based candidates, observed validation regressions in several VAE-style variants, and retained a reference-aware revision under the fixed validation protocol.

Trajectory evidence. The record links observed validation outcomes to subsequent local revisions and the candidate retained by the selection rule.

Closed-Loop Revision Trajectory



(A) From Regression to Reference-Aware Generation Generative candidates in the BBBC047 trajectory showed both validation gains and regressions. The run recorded possible underuse of the supplied control-reference information as a working diagnostic for subsequent candidate generation.

Key Observation

Observed pivot. The retained revision injects a deterministic encoding h of the supplied control-reference profile alongside the stochastic latent variable in the decoder. The observed validation result supports retaining this reference-aware design.

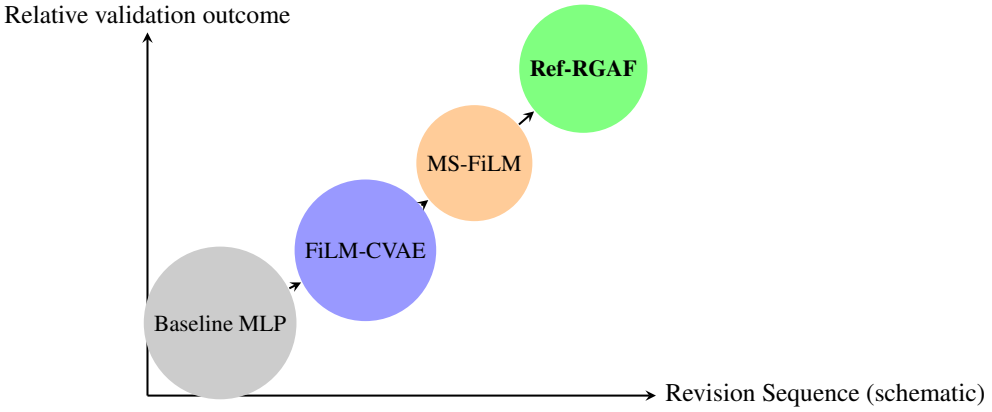


Figure 6: Auditable model-revision trajectory: the system moves from global modulation to multi-scale modulation and then to a reference-aware decoder.

(B) Auditable Model-Revision Trajectory We report the complete trajectory, including Phase 2 instability and Phase 3 revisions, to make validation-based candidate retention inspectable. **Phase 1: Diagnosis and Constraint Identification.** The system identified data-quality issues and recorded preprocessing checks and split alignment before candidate evaluation (Table 10).

Table 10: **Phase 1 Execution Trace: Diagnosis and Constraint Identification.**

Step	Activity	Artifact	Design / Operational Insight
1	Data Ingestion	HDF5/Pandas	**Operational repair:** Fixed a `TypeError` caused by mixed numeric and string columns in the raw CSV.
2	Numerical Diagnosis	Feature Scan	**Constraint definition:** Identified invalid-value warnings and added strict NaN checks and feature filtering to prevent training failures.
3	Split Alignment	Grouped CV	**Protocol choice:** Grouped samples by exact SMILES identity, matching the SMILES-based split used for the reported controller and trajectory scores.

Phase 2: Generate and Execute. Phase 2 recorded frequent code-generation and training failures. The baseline MLP had the highest validation PCC among eligible candidates and provided the reference for subsequent revisions (Table 11). This Phase 2 value is an exploratory setup-trace fit; the Phase 3 aggregate gains use a separate Phase 3 reference fit.

Table 11: **Phase 2 Execution Trace: Establishing the Baseline.**

Iter	Strategy	Outcome	PCC	Design Analysis
1	Init	Crash	N/A	Syntax error in setup node.
2	**Baseline MLP**[a]	**Retained**	**0.4118**	**Anchor candidate:** A standard MLP using median-normalized features provided a stable reference point.
3	Innovation Var.	Regression	0.3834	**Negative result:** A more complex candidate obtained a lower validation PCC and was rejected by the validation rule.
4	Training Loop	Repaired / Regression	0.3707	**Instability:** Training-node failure was repaired, but the candidate degraded.
5	Final Attempt	Repaired / Regression	0.3468	**Regression:** Severe instability led the system to preserve Iteration 2 as the validation-selected anchor.

[a]Exploratory Phase 2 baseline; the Phase 3 trajectory uses the separately refit reference of 0.4120 shown in Table 13.

Phase 3: Review and Revise. Phase 3 evaluated modulation, gating, and reference-aware variants (Table 12). The retained Ref-RGAF candidate adds a deterministic control-reference path to the generative architecture.

Table 12: **Phase 3 Full Iteration Trace: Revision Logic.** "Improved" denotes an increase from the immediately preceding candidate; "Retained" denotes the best validation-selected candidate in this trace.

Iter	Strategy	Outcome	PCC	Design Analysis
1	**FiLM-CVAE**	Improved	0.4148	**Pivot:** Replaced a rigid ODE-style candidate with FiLM modulation, where drug embedding modulates the latent state.
2	Res-FiLM	Regression	0.3416	**Negative result:** The residual VAE obtained a lower validation PCC; the trace used possible underuse of control-reference information as a working diagnostic for later revisions.
3	Dose-Gated	Improved	0.4095	Candidate assumption: simple dose gating may be sufficient. Result: the candidate obtained a lower validation PCC than the retained anchor.
4	TransMod	Improved	0.4097	The transformer-style fusion candidate obtained a lower validation PCC than the retained anchor.
5	**MS-FiLM**	Improved	**0.4237**	**Design cue:** The multi-layer modulation candidate exceeded the bottleneck-only candidate in validation PCC.
6	Gated-Attn	Improved	0.4265	Multiplicative gating improved validation PCC relative to the preceding candidates in this trace.
7	RGAF-FiLM	Improved	0.4271	Added affine fusion and a residual hypernetwork.
8	**Ref-RGAF**	**Retained**	**0.4368**	**Key revision:** Concatenated deterministic control-reference encoding h with latent z; this candidate obtained the highest validation PCC in the trace.
9	State-Adaptive	Regression	0.4249	The state-adaptive fusion candidate obtained a lower validation PCC than Ref-RGAF.
10	LoRA-FiLM	Regression	0.4163	The LoRA-style latent-rotation candidate obtained a lower validation PCC than Ref-RGAF.

Failure Mode / Design Risk

Working diagnostic. The residual VAE candidate used $y = x + \mathrm{Dec}(z)$ and obtained a lower validation PCC. The run record proposed possible underuse of the supplied control-reference information as a working diagnostic and used it to motivate reference injection.

(C) Validation-Selected Design: Reference-Gated Affine Fusion The retained **Ref-RGAF** candidate combines three implementation choices:
(1) Control-reference conditioning. Ref-RGAF adds a deterministic reference path:

$$v_0 = [z_{\text{stochastic}}, h_{\text{deterministic}}]. \tag{6}$$

The joint state v_0 is affinely modulated before decoding, preserving a deterministic encoding of the supplied plate-matched DMSO control profile alongside the sampled latent state.
(2) Affine dose modulation. The drug effect is represented as a dose-conditioned affine transformation:

$$\text{DrugFeat} = (E_{\text{chem}} \odot \sigma(\text{Scale}(d))) + \text{Shift}(d). \tag{7}$$

Layer normalization and SiLU transform this intermediate feature before a hypernetwork supplies scale and shift vectors for three decoder stages.
(3) Multi-scale modulation. Drug modulation is applied at multiple decoder layers, allowing perturbation effects to enter at several representation levels.

Design Rationale Encoded as Architecture

Design rationale. The retained architecture reflects a task-specific modeling preference for preserving deterministic control-reference information while combining it with compound and dose features.

(D) Validation Performance and Response-Sensitive Metrics The retained Ref-RGAF candidate achieved a five-fold validation aggregate PCC of 0.4368 (Table 13), corresponding to CS-model on BBBC047 in Table 28. This reference-aware revision was retained by the validation-selection rule.

Table 13: **Trajectory of Candidate Performance (Phase 3).**

Iter	Architecture Strategy	PCC	Δ vs Base	RF-RMSE ↓	Status
Ref[a]	Baseline MLP	0.4120	0.00%	5.13	Reference
1	FiLM-CVAE	0.4148	+0.68%	5.07	Early gain
5	MS-FiLM	0.4237	+2.84%	5.01	Improved
8	**Ref-RGAF**	**0.4368**	**+6.02%**	**4.96**	**Retained**

[a]Phase 3 reference fit, separately refit from the exploratory Phase 2 baseline (0.4118); all Phase 3 gains in this table use 0.4120.

Key Observation

Interpreting RF-RMSE. Response-feature RMSE measures prediction error on the most perturbation-responsive morphology features, complementing global PCC when evaluating the retained candidate.

(E) Summary On BBBC047, CellScientist used a working diagnostic about control-reference use to propose the Ref-RGAF design. The trajectory preserves valid but underperforming candidates and links the reference-aware revision to execution results and validation-based selection.

C Experimental Details

This appendix describes the datasets, baselines, metrics, and training and evaluation protocols used in Section 4.1.

C.1 Benchmark Tasks and Datasets

C.1.1 Cell Painting Morphology Prediction

We evaluate model discovery on BBBC036, BBBC047, and CPG0016, and transfer two frozen designs to LKCP Batch 2. Prediction targets are CellProfiler features (Stirling et al., 2021) describing cellular size, shape, texture, and fluorescence intensity. Each task uses an assay-context reference derived from vehicle controls according to its preprocessing protocol.

BBBC036 and BBBC047 derive from the Cell Painting small-molecule collection of Bray et al. (Bray et al., 2017). CPG0016 is part of the JUMP Cell Painting resource (Chandrasekaran et al., 2023; Arevalo et al., 2024). All three use U2OS cells and span progressively larger sets of chemical perturbations. Table 14 reports source-profile counts for the processed inputs, distinguishing raw profiles from treated observations used for response prediction. Raw profiles include vehicle controls; treated profiles contain only retained non-DMSO observations. Exact non-control SMILES are counted within each benchmark, and raw feature counts precede split-specific filtering.

The discovery benchmarks use two group-based holdout settings. The SMILES-based setting groups records by exact recorded SMILES, and the plate-based setting groups records by experimental plate. Complete groups are assigned to a common fold.

Table 14: **Source-level statistics of the processed Cell Painting inputs used in our experiments.** Raw profiles include vehicle controls. Treated profiles are non-DMSO observations retained for response prediction. Exact non-control SMILES are counted within each benchmark, and raw Cell Painting feature counts precede split-specific feature filtering.

Dataset	Cell line	Raw profiles	DMSO controls	Treated profiles	Exact non-control SMILES	Plates (all / treated)	Raw CP features
BBBC036	U2OS	18,540	3,528	15,012	1,916	55 / 47	602
BBBC047	U2OS	59,275	16,302	42,973	11,415	241 / 229	894
CPG0016	U2OS	235,307	32,837	202,470	82,282	407 / 392	737

LKCP Batch 2 comes from the LINCS Cell Painting release (Natoli et al., 2021; Way et al., 2022) in the Cell Painting Gallery (Weisbart et al., 2024). We use a processed subset of the December 5, 2017 batch containing 64 compounds, 1,728 conditions, and 1,671 morphology features. Compound-level hashing assigns 33 compounds (891 conditions) to training, 15 (405 conditions) to validation, and 16 (432 conditions) to testing. Inputs are a same-plate median vehicle-control profile, a 2,048-bit radius-2 Morgan fingerprint, and log dose. The treated-scale target is reconstructed by adding the condition-median treated-minus-control profile to the condition-median control profile.

C.1.2 LINCS2020 Transcriptomic Perturbation Prediction

For transcriptomic perturbation prediction, we use gene-expression profiles from the next-generation Connectivity Map (CMap) project, which measures large-scale transcriptional responses with the L1000 platform (Subramanian et al., 2017). Raw and processed profiles are obtained through the LINCS data portal (Stathias et al., 2020), following recommended access and preprocessing protocols (Xie et al., 2022). We construct two benchmark variants to evaluate different generalization regimes: a controlled **Seven Cell Lines** subset and a large-scale **Full Data** benchmark.

The **Seven Cell Lines** subset contains 355 compounds profiled across seven frequently assayed cell lines: A375, HELA, PC3, MCF7, HT29, YAPC, and HA1E. This dense subspace provides high compound–cell coverage and enables a chemical-blind split, where scored compounds are unseen in the corresponding training folds. It therefore measures generalization to held-out compound identities. The **Full Data** benchmark contains 8,316 compounds across 164 cell lines, yielding 78,569 compound–cell expression profiles. This setting is substantially sparser and more heterogeneous and measures performance under long-tail cell-line coverage and diverse perturbation contexts. Dataset statistics are summarized in Table 15.

Table 15: **Statistics of LINCS L1000 benchmarks.**

Dataset	# Compounds	# Cell Lines	# Samples	Characteristics
Seven Cell Lines	355	7	∼2,485	Dense; chemical-blind
Full Data	8,316	164	78,569	Sparse; long-tail

C.1.3 Single-cell Perturbation Prediction

We further evaluate the CellScientist workflow on public single-cell benchmarks covering genetic perturbation and multimodal perturbation responses. These datasets complement morphology and L1000 evaluations with measurements at single-cell resolution and response-sensitive prediction targets.

Norman (Norman et al., 2019) is a pooled CRISPRa scRNA-seq dataset designed to study the effects of single and combinatorial gene activation. It is commonly used to evaluate transcriptional-response prediction for genetic interventions, including combinations. **Papalexi** (Papalexi et al., 2021) is a CRISPR–Cas9 ECCITE-seq dataset that jointly profiles RNA and surface-protein responses after genetic perturbation, enabling evaluation on both transcriptomic and antibody-derived-tag readouts. Table 16 summarizes the processed benchmark statistics used in the corresponding evaluations.

We report task-specific workflow results on Norman, Papalexi RNA, and Papalexi Protein in Appendix F.4. Each readout is evaluated under its dataset-specific protocol.

Table 16: **Statistics of single-cell benchmarks used in our evaluations.** Readout dimensions are reported where the evaluation uses a fixed feature matrix; Papalexi RNA and protein are evaluated as separate readouts.

Benchmark	Assay / task	Cells	Readout and condition statistics
Norman	CRISPRa scRNA-seq	91,205	5,045 RNA genes; 284 conditions (152 single-gene, 131 two-gene, and 1 control)
Papalexi	CRISPR–Cas9 ECCITE-seq	20,729	RNA and surface-protein ADT readouts; 25 targeted genes plus non-targeting control and 98 perturbation labels

C.2 Baselines

C.2.1 Morphology Baselines

For morphology prediction, we organize baselines into three comparison groups used consistently in the result tables.

Existing reference methods include strong feature-space predictors:

- **TabR** (Gorishniy et al., 2024): A tabular prediction model with a retrieval component that uses nearest-neighbor information at inference time.
- **RealMLP** (Holzmüller et al., 2024): A carefully tuned MLP baseline designed to provide strong and stable performance on tabular benchmarks.
- **RF-TD** (Holzmüller et al., 2024): A Random Forest baseline with tuned default settings for competitive tabular prediction.

Search-based baselines provide generic model-selection reference points under the corresponding benchmark split, target, and metric definitions:

- **FLAML** (Wang et al., 2021): An AutoML framework that searches over model and hyperparameter configurations under a configured budget.
- **Random Search** (Gijsbers et al., 2024): A generic search baseline that samples candidate configurations independently.

Agent-designed baselines provide a comparison with an existing automated modeling workflow:

- **CellForge** (Tang et al., 2025): A multi-agent automated modeling framework for proposing, training, and selecting predictive models.

The LKCP comparison uses RealMLP, RF-SKL, early-fusion MLP, and control-only MLP. RF-SKL is a scikit-learn random forest (Breiman, 2001; Pedregosa et al., 2011), whereas the discovery benchmarks use the tuned-default RF-TD variant. RealMLP, RF-SKL, and early-fusion MLP concatenate context, fingerprint, and dose. Control-only MLP uses the same MLP with chemical and dose inputs fixed to zero. The two transferred CS-models preserve the DD-MIA and Ref-RGAF architectures specified in Appendix C.4.1, adapting their morphology input and output dimensions to LKCP.

C.2.2 Transcriptomic and Single-cell Baselines

For LINCS2020 transcriptomic perturbation prediction, we compare against representative domain-specific baselines:

- **DLEPS** (Zhu et al., 2021): A deep-learning predictor of chemically induced transcriptional changes for drug screening.
- **DeepCE** (Pham et al., 2021): A graph neural network framework that combines chemical substructure representations with cell-line features.
- **CIGER** (Pham et al., 2022): An attention-based model that predicts gene-expression rankings from chemical structures.

- **MultiDCP** (Wu et al., 2022): A multi-task model of dose-dependent, context-specific gene-expression and cell-viability responses.
- **TranSiGen** (Tong et al., 2024): A VAE-based generative model that learns latent representations for denoising and reconstructing perturbation profiles.

For the workflow-instantiation results on Norman and Papalexi, we compare against **CellForge** (Tang et al., 2025) under the dataset-specific evaluation procedures.

C.3 Evaluation Metrics

C.3.1 Morphology Metrics

For morphology prediction, we evaluate global profile fidelity and perturbation-sensitive feature recovery. Given prediction and target matrices $\widehat{\mathbf{Y}}$ and $\mathbf{Y}$, global PCC is $\mathrm{corr}(\mathrm{vec}(\widehat{\mathbf{Y}}), \mathrm{vec}(\mathbf{Y}))$. MSE measures numerical reconstruction error, PCC measures profile association, and R^2 measures explained variation relative to feature means. The discovery benchmarks summarize validation scores across five folds; the controller audit reports the validation-selected candidate on its reporting fold. LKCP uses global PCC, entry-wise MSE, and mean feature-wise R^2 on the fixed external test partition, summarized as mean±sample SD over five neural-model refits and a point estimate for the single RF-SKL fit. Negative feature-wise R^2 indicates greater squared error than predicting that feature's test-set mean; PCC is insensitive to affine scale and offset. Appendix D.1 reports seed-matched differences from RealMLP.

To further assess whether a model captures the most responsive morphology changes, we compute response-feature metrics on the top-K perturbed features. For each sample i, let $\mathbf{y}_{i,\mathrm{gt}}$ be the ground-truth response vector and $\boldsymbol{\mu}_{\mathrm{ctrl}}$ the mean control profile. We select responsive features using only ground-truth deviation from control:

$$\mathcal{F}_i^{(K)} = \mathrm{topk}_j \left(\left| \mathbf{y}_{i,\mathrm{gt}}^{(j)} - \boldsymbol{\mu}_{\mathrm{ctrl}}^{(j)} \right| \right), \qquad K \in \{20, 50\}.$$

We then compute $\mathrm{RMSE}_{\text{RF-}K}$ and $\mathrm{PCC}_{\text{RF-}K}$ within $\mathcal{F}_i^{(K)}$, where RF denotes response feature. The feature subset is determined from ground truth and control profiles. These metrics complement the global PCC used for discovery and controller retention.

C.3.2 Transcriptomic and Single-cell Metrics

For LINCS2020 transcriptomic prediction, RMSE and PCC measure global expression-profile fidelity. Positive and Negative P@100 measure recovery of the most strongly upregulated and downregulated genes, respectively. These metrics complement whole-profile accuracy with the recovery of strongly responding genes.

For single-cell perturbation prediction, we report PCC and $\mathrm{PCC}_{\mathrm{DE}}$. PCC measures global agreement between predicted and observed response profiles, while $\mathrm{PCC}_{\mathrm{DE}}$ evaluates the designated differentially expressed subset under each dataset-specific protocol. Together, these metrics separate overall profile reconstruction from recovery of key response signals.

C.4 Implementation Details

C.4.1 Frozen Predictor Implementations

The frozen predictors use the following architectures. Let $x \in \mathbb{R}^p$ be the supplied control profile, $c \in \{0,1\}^{2048}$ the radius-2 Morgan fingerprint, and $d = \log_{10}(\mathrm{dose} + 10^{-6})$. External LKCP refits use $p = 1671$ and millimolar dose. The architectures are retained while input/output dimensions adapt to the cohort.

DD-MIA. The context and chemical encoders map $p \to 512$ and $2048 \to 512$, respectively, each with BatchNorm, ReLU, and two residual blocks. A residual block computes $u + \mathrm{Dropout}_{0.2}(\mathrm{ReLU}(\mathrm{BN}(Wu + b)))$. The dose MLP is $1 \to 64 \to 64$ with an intermediate ReLU. The context and chemical 512-vectors are reshaped into 16×32 tokens; four-head attention uses context queries and chemical keys/values. Its flattened output $a \in \mathbb{R}^{512}$ is multiplied by separate

chemical and dose sigmoid gates, then added to the context embedding and layer-normalized. The predictor consumes the concatenated $[u, h_c, h_d] \in \mathbb{R}^{1088}$ through $1088 \to 512 \to 256 \to p$, with BatchNorm/ReLU/dropout 0.2 after the first layer and ReLU after the second. It directly outputs the treated profile.

A fold classifier applies gradient reversal of strength 0.5 to h_m, followed by $512 \to 128 \to 6$ with ReLU/dropout 0.2. The training objective is

$$\mathcal{L}_{\mathrm{DD}} = \mathrm{MSE}(\widehat{y}, y) + 0.1\,\mathrm{CE}(C(\mathrm{GRL}_{0.5}(h_m)), f - 1),$$

where f is the recorded fold ID. The classifier minimizes cross-entropy, while gradient reversal opposes its gradient into the context encoder. The six-way head is retained on LKCP, whose training data use three fold labels.

Ref-RGAF. The context encoder is $p \to 1024 \to 256$, with BatchNorm and SiLU after each layer, giving deterministic state h. Two $256 \to 64$ heads yield μ and $\log \sigma^2$; $z = \mu + \sigma \odot \epsilon$, with $\epsilon \sim \mathcal{N}(0, I)$. A $2048 \to 256$ chemical encoder uses LayerNorm and SiLU. Separate $1 \to 64 \to 256$ dose MLPs with SiLU produce scale $S(d)$ and shift $B(d)$:

$$q = \mathrm{SiLU}(\mathrm{LN}(E_c(c) \odot \sigma(S(d)) + B(d))).$$

A hypernetwork maps $256 \to 512$, applies SiLU and two residual blocks, then maps $512 \to 3200$. Each block contains two linear–LayerNorm–SiLU–dropout(0.1) layers with a residual addition. The output supplies scale/shift pairs at widths 320, 256, and 1024. The initial state $[z, h]$ has width 320. Decoder dimensions are $320 \to 256 \to 1024 \to p$, with BatchNorm/SiLU at the two hidden layers. Modulation $v \mapsto v \odot (1 + \gamma_j) + \beta_j$ occurs before decoding and after each hidden activation. The final layer is linear, and the objective is

$$\mathcal{L}_{\mathrm{Ref}} = \mathrm{MSE}(\widehat{y}, y) + 0.001\mathcal{L}_{\mathrm{KL}}, \qquad \mathcal{L}_{\mathrm{KL}} = -\tfrac{1}{2}\,\mathrm{mean}(1 + \log \sigma^2 - \mu^2 - \sigma^2).$$

The KL mean covers batch and latent entries. Sampling remains active during validation and testing. The fixed refit seed determines evaluation random states: seed+100000+epoch during validation, seed+300000 for saved-checkpoint validation, and seed+500000 for test evaluation.

C.4.2 Implementation and Execution Configuration

Table 17 reports the environments used for the high-freedom workflow and the controlled audit. Methods compared within each experiment use the same hardware and software environment. Table 18 separates open-workflow settings from those used to compare controllers within the fixed candidate language.

For LKCP refitting, DD-MIA uses training-standardized context, Adam at 10^{-4}, batch size 32, and its adversarial objective. Ref-RGAF uses unscaled context, Adam at 10^{-3}, batch size 64, and MSE plus 0.001 times KL loss. Both use at most 500 epochs and validation-MSE early stopping with patience 50; Ref-RGAF retains latent sampling with fixed seed-derived random states at evaluation. DD-MIA's validation loss averages mini-batch MSEs equally; Ref-RGAF uses entry-wise MSE. RealMLP uses the PyTabKit tuned-default configuration with at most 500 epochs, batch size 32, and early stopping. RF-SKL uses 200 trees. Context and response scalers for the tabular predictors are fitted on training data, and predictions are returned to the target scale. Neural refits use seeds 1729–1733; RF-SKL uses one fixed seed. Model designs remain fixed throughout external refitting.

Table 17: **Computational infrastructure and software environment.**

Component	Specification
High-freedom workflow GPU	NVIDIA RTX 5880 Ada Generation (48GB VRAM)
High-freedom workflow software	Python 3.11.14; PyTorch 2.0.1+cu118
Controlled-audit GPU	NVIDIA H100 (80GB HBM3)
Controlled-audit software	Python 3.10.20; PyTorch 2.3.1+cu121
Operating system	Linux

Table 18: **CellScientist controller and execution settings.**

Configuration	Value	Role in the workflow
Open-workflow backbone	Gemini 3 Pro	Default backbone for high-freedom model exploration
Open-workflow context length	40,000 tokens	Maintained interaction context limit
Open-workflow temperatures	0.5 analysis; 0.7 proposal	Sampling settings for analysis and generative revision
Open-workflow prompt views	4	Robust scaling, PCA, feature-network, and enrichment views
Open-workflow iteration budget	10 revisions	Maximum number of high-freedom revision steps
Open-workflow repair budget	3 rounds	Configured debugging budget for runtime or interface errors
Controlled-audit backbones	Gemini 3 Pro; Qwen2.5-0.5B-Instruct	Manuscript-default and open-weight reproduction
Controlled-audit decoding	Temperature 0	Fixed within every matched controller comparison
Controlled-audit candidate language	768 executable candidates	Shared predictor configurations available to every policy
Controlled-audit fitting	Seeded randomized low-rank fitting	Identical fitting seed for every policy within each paired run
Controlled-audit budgets	$B \in \{1, 3, 5, 10\}$	Trained and evaluated candidates, including h_0
Validation selection metric	Global PCC	Metric used by $\mathrm{Select}_{\mathrm{val}}$ in the controlled audit
Controlled-audit retention view	Mean feedback/selection PCC	Identical candidate-retention rule for every matched policy

C.4.3 Protocol Controls and Routing Details

This section specifies evaluation semantics and feedback routing for the high-freedom workflow. Within each benchmark, methods use the reported split, target, and metric definitions. Fixed predictors follow their model protocols, search baselines use configured search procedures, and CellForge follows its agent workflow. CellScientist uses 10 revision steps with up to 3 local implementation repairs per step. Candidates are eligible for selection only after passing the configured checks and completing scoring; rejected and failed candidates remain in the run history.

PDR associates execution and validation diagnostics with proposed component edits. The open-code prompt contains baseline/candidate metric JSON, mutable and read-only notebook context, history, and a task-graph summary. Deterministic text flags improved PCC with worse MSE as a scale/bias discrepancy, improved MSE with worse PCC as conservative fitting, and worsening top-response-feature error as a weighting issue. The first two flags are mutually exclusive; the feature-error flag is independent. The LLM returns a focus, task names, and cell-indexed edits. Active tasks are resolved by returned-name matching, then focus-label matching, then root children; matching is case-insensitive and allows substrings. Initial edits are applied only to registered mutable cells. Graph dependencies supply context and attribution rather than an additional edit-validity test. Runtime repair uses error context and a bounded retry prompt. Table 19 summarizes diagnostic-to-edit mappings; Listing 1 summarizes the early diagnostic and selected design in one recorded trajectory.

Table 19: **Representative PDR mappings.** PDR maps diagnostic signals to proposed local edit targets; LCA checks the resulting edits against protected evaluation semantics.

Diagnostic	Editable region	Protected semantics
Interface failure	Data adapter or implementation interface	Split, target, and metric definitions
Metric-reporting failure	Evaluation adapter	Metric formulas and fold protocol
Validation collapse	Fusion, decoder, objective, or training node	Dataset split, target definition, and output space
Weak perturbation sensitivity	Perturbation encoding, dose encoding, or fusion edge	Evaluation code and response-feature definition
Suspected underuse of control reference	Encoder state, decoder input, or fusion component	Preprocessing, split, target, and metrics
Batch-shortcut risk	Representation or training-regularization component	Configured split and evaluation rules


```
{
  "dataset": "BBBC047",
  "iteration": 2,
  "diagnostic": {
    "type": "suspected_control_reference_underuse",
    "candidate": "Res-FiLM",
    "execution_status": "valid",
    "rule_status": "passed",
    "selection_metric": "PCC",
    "candidate_score": 0.3416,
    "best_previous_score": 0.4148,
    "verdict": "valid_but_regressed",
    "summary": "Validation PCC dropped after a residual VAE-style
        update; control-reference use was recorded as a working
        diagnostic."
  },
  "routed_address": {
    "address_type": "proposed_edit_targets",
    "editable_components": [
      "model_architecture",
      "encoder_state_representation",
      "decoder_input",
      "fusion_module"
    ],
    "routing_reason": "valid execution with degraded validation
        performance and a working diagnostic about control-reference use"
  },
  "allowed_edit": {
    "operation": "test_reference_aware_decoder_path",
    "protected_components": [
      "dataset_split",
      "target_definition",
      "feature_preprocessing",
      "validation_protocol",
      "evaluation_metrics",
      "fold_protocol"
    ]
  },
  "revised_candidate": {
    "iteration": 8,
    "name": "Ref-RGAF",
    "design_change": "concatenate deterministic control-reference
        encoding h with stochastic latent code z before decoding",
```

```
    "intended_effect": "retain supplied control-reference information
        while allowing compound-dose modulation"
  },
  "lca_check": [
    "interface_check",
    "shape_check",
    "configured_rule_check",
    "metric_completeness"
  ]
}
```

Listing 1: Structured summary of the BBBC047 trajectory: the iteration-2 diagnostic and the iteration-8 Ref-RGAF candidate.

C.4.4 Shared Candidate Language and Policy Access

Domains and one-edit neighborhoods. Each candidate is a tuple (u, v, r_x, r_y, α) specifying the input blocks, response target, input and output PCA dimensions, and regularization strength of the common low-rank predictor. Let c denote the supplied control-reference profile, p the hashed SMILES descriptor, and a log dose. Direct prediction fits y; response prediction fits $y - c$ and adds c back at inference. The full Cartesian product contains $4 \times 2 \times 4 \times 4 \times 6 = 768$ distinct candidate IDs. The common initial candidate is $([c, a], y - c, 32, 32, 10)$. Requested PCA dimensions are capped by the available sample and feature dimensions during fitting.

Table 20: **The shared 768-candidate language.** The listed order is also the fixed preference order used by CellScientist and Random routing to instantiate an edit within an already selected address. Alternatives count one-field changes from an incumbent before previously evaluated IDs are removed.

Address	Allowed values, in edit-preference order	Choices	Alternatives
Conditioning	$[c, p, a]$, $[c, p]$, $[c, a]$, c	4	3
Response target	$y - c$, y	2	1
Input representation	64, 128, 32, 16	4	3
Output decoder	64, 128, 32, 16	4	3
Regularization	1, 0.1, 0.01, 10, 100, 1000	6	5

An atomic transition changes exactly one tuple coordinate. Every state therefore has $3+1+3+3+5 = 15$ one-edit neighbors before history filtering. These are local neighborhood counts, not disjoint subsets of the global language. For incumbent h and previously evaluated IDs S, the available address set is

$$\mathcal{A}_{\text{avail}}(h, S) = \{j : \mathcal{N}_j(h) \setminus S \neq \varnothing\},$$

where $\mathcal{N}_j(h)$ contains the one-field alternatives at address j. Random routing samples uniformly from this set. CellScientist ranks addresses within it using diagnostics and history, then exposes the first two ranked addresses. Both instantiate one unseen candidate at an exposed address using the same preference order in Table 20, with candidate ID as the tie-breaker. For identical (h, S, j), the instantiated edit is identical under both policies. Allow-list and pairing checks constrain proposals before evaluation; the runner records transition admissibility.

What each policy receives. The shared object is the candidate universe and its evaluator. The per-step candidate menu is produced by the search policy, as specified in Table 21. CellScientist's controlled implementation routes among existing configurations; candidate generation in the open-code workflow is evaluated separately.

Menus contain fewer than 12 candidates when fewer unseen choices remain. The local parent is the candidate with highest feedback PCC, retaining the earliest observation on ties; the common final retention rule uses mean feedback/selection PCC. CellScientist orders its diagnostic priorities as follows: incomplete input conditioning; direct targets or response PCC exceeding global PCC by more than 0.05; top-50 response PCC below response PCC; and input/output capacity below 64, or capacity

Table 21: **Policy-specific access to the common candidate universe.** Every revising policy uses the same h_0, fitting procedure, feedback/selection/test partitions, and total candidate budget. Unseen means not evaluated earlier in that trajectory.

Policy	Candidate menu	Decision rule
Fixed h_0	Common initial candidate	Evaluate once; retain as the unrevised reference.
Flat retry	First 12 unseen one-edit neighbors, sorted by candidate ID	LLM selects an ID from the local menu.
Random routing	One canonical edit at a uniformly sampled available address	LLM returns the pre-instantiated ID/address pair.
AIDE-style draft	12 uniformly sampled unseen IDs from the full language	LLM selects an independent root candidate.
AIDE-style improve	First 12 unseen neighbors of the best-feedback node, sorted by ID	LLM selects a one-edit improvement.
CellScientist	One canonical edit at each of at most two ranked addresses	LLM selects an allow-listed ID/address pair.

at 64 when the incumbent outperforms an otherwise identical lower-rank history entry. The decoder-error diagnostic can prioritize that address independently of the capacity-based rule. Regularization, target, and conditioning form the fallback priority. Duplicate addresses, exhausted neighborhoods, and addresses already attempted without improvement since the incumbent are removed; the first two remaining addresses form the shortlist. Simultaneous triggers keep this order. The LLM is prompted to prefer the earlier pair unless feedback favors the second; malformed, out-of-menu, or mismatched responses use the first pair. These rules and value preferences are fixed during the reported runs.

Diagnostic schema and graph record. The evaluator returns global PCC, sample-wise PCC, MSE, pooled R^2 with feature-wise centering, centered PCC, response PCC, response sample-wise PCC, response MSE, and top-20/top-50 response PCC. Responses subtract each row's control profile. Global and response PCC flatten their matrices; top-K PCC averages row-wise correlations over the K largest absolute observed responses. Ranking reads global, response, and top-50 response PCC, plus configuration and history. Fixed dependency edges are conditioning $\rightarrow$ representation/target, target $\rightarrow$ decoder, representation $\rightarrow$ decoder/regularization, and decoder $\rightarrow$ regularization. These edges are recorded for inspection; legal transitions are determined by the configuration domains and one-field neighborhoods. For a selected pair (j, h'), $\mathcal{T}_{\text{edit}}$ replaces field j of the incumbent by its value in h'. The deterministic preference order is shared with Random routing, so their difference is address selection rather than a different within-address candidate generator.

Executable checks. Before fitting, the live controlled runner validates domain membership, response allow lists, address–ID pairing, and duplicate IDs. After prediction it checks shape and finite outputs/metrics. Atomic-transition status is recorded with the trial. Proposal/execution exceptions invoke an unseen one-field recovery neighbor, deterministically indexed by seed and step; the common candidate budget remains in force. The separate LCA record audit additionally tests protected-state mutation, exact interface fields, non-atomic edits, route consistency, and repair-budget exhaustion. Its repair actions restore protected fields, remove extra interface fields, or correct the routed address; output-shape and numerical failures are rejected. Thus the registered audit and live-search checks have explicit, distinct scopes.

Language and comparison scope. All 768 configurations instantiate the same low-rank fitting pipeline; input blocks and ranks change fitting cost and representational capacity. The controlled contrast evaluates the combined routing policy within this developed design space. Deterministic ranking and LLM pair selection are evaluated jointly; the no-history audit separately measures proposal repetition.

AIDE-style correspondence. The adaptation retains the reference algorithm's initial drafting and subsequent greedy improvement of the best valid node (Jiang et al., 2025). After the common h_0, it creates five new root drafts when budget permits, then improves the best-feedback node; all

evaluations count toward B. The executable candidate language supplies the configurations for both stages. The reference algorithm's debugging branch is conditional on buggy code, whereas these configurations are executable by construction. The recorded reference is the official implementation at commit `5d66a21`. This implements its draft-then-improve policy under the common predictor and evaluation budget.

Fixed execution protocol. After development, the finalized language and controller code are fixed for the reported five-seed comparisons. Before formal execution, the runner verifies a lock containing the configuration, dataset, and source-file hashes, and rejects a mismatch. Each run records the verified protocol and source-bundle hashes. The reported Gemini and Qwen runs share the same source bundle, including the candidate domains, neighborhood construction, and policy implementations; the backbone-specific configurations have separate locks. This records execution under the finalized protocol.

C.4.5 LLM BACKBONES

The matched-budget audit uses Gemini 3 Pro for the manuscript-default runs and repeats the controller comparison with the open-weight Qwen2.5-0.5B-Instruct model. Within each backbone, every controller receives the same decoding configuration and candidate budget. Table 22 records the public model identities used in this audit.

Controller-call accounting. Gemini completed 720 controller decisions with valid allow-listed responses and no fallback. Qwen completed 720 decisions; invalid field pairings triggered the registered legal fallback in 10/180 CellScientist decisions and 70/180 AIDE-style decisions. Flat and Random routing used no fallback. All decisions count toward the reported candidate budgets.

Table 22: **LLM backbones used in the controlled framework audit.** Gemini 3 Pro is the manuscript-default backbone; Qwen2.5-0.5B-Instruct provides an open-weight reproduction.

Main-text Label	Full Backbone Identity	Provider / Family	Source Link
Gemini 3 Pro	Gemini 3 Pro	Google Gemini	https://blog.google/products/gemini/gemini-3
Qwen 0.5B	Qwen2.5-0.5B-Instruct	Qwen	https://huggingface.co/Qwen/Qwen2.5-0.5B-Instruct

D CONTROLLER-BACKBONE REPRODUCTION

Table 23: **Task-level reproduction of structured routing across controller backbones.** The upper panel reports mean±sample SD in held-out global PCC over five paired controller/fitting seeds. The lower panel reports CellScientist-minus-Random effects with descriptive four-setting 95% t intervals, computed from unrounded results. CS denotes CellScientist. Both backbones use the same task settings, candidate language, fitting, and budgets. Bold marks the better mean within each backbone, setting, and budget.

Backbone	Dataset	Split	$B=3$		$B=5$		$B=10$	
			Random	**CS**	**Random**	**CS**	**Random**	**CS**
Gemini 3 Pro	BBBC036	Plate	0.2250 ±0.0493	**0.2650** ±0.0005	0.2277 ±0.0502	**0.2674** ±0.0003	0.2626 ±0.0077	**0.2693** ±0.0012
	BBBC036	SMILES	0.2052 ±0.0413	**0.2336** ±0.0016	0.2049 ±0.0417	**0.2346** ±0.0009	**0.2347** ±0.0016	0.2346 ±0.0009
	BBBC047	Plate	0.3032 ±0.0500	**0.3396** ±0.0004	0.3066 ±0.0514	**0.3457** ±0.0004	0.3458 ±0.0027	**0.3481** ±0.0009
	BBBC047	SMILES	0.3131 ±0.0536	**0.3529** ±0.0010	0.3207 ±0.0582	**0.3614** ±0.0003	0.3668 ±0.0073	**0.3751** ±0.0017
Qwen2.5-0.5B	BBBC036	Plate	0.2250 ±0.0493	**0.2566** ±0.0024	0.2277 ±0.0502	**0.2566** ±0.0024	0.2626 ±0.0077	**0.2657** ±0.0035
	BBBC036	SMILES	0.2052 ±0.0413	**0.2357** ±0.0004	0.2049 ±0.0417	**0.2357** ±0.0004	**0.2347** ±0.0016	0.2346 ±0.0009
	BBBC047	Plate	0.3032 ±0.0500	**0.3429** ±0.0003	0.3066 ±0.0514	**0.3425** ±0.0009	**0.3458** ±0.0027	0.3438 ±0.0018
	BBBC047	SMILES	0.3131 ±0.0536	**0.3570** ±0.0005	0.3207 ±0.0582	**0.3615** ±0.0005	0.3668 ±0.0073	**0.3707** ±0.0002

CellScientist − Random: mean effect with paired-setting 95% CI

Backbone	Contrast	$B=3$	$B=5$	$B=10$
Gemini 3 Pro	CellScientist − Random	+0.0361 [+0.0275, +0.0448]	+0.0373 [+0.0292, +0.0454]	+0.0043 [−0.0018, +0.0104]
Qwen2.5-0.5B	CellScientist − Random	+0.0364 [+0.0261, +0.0467]	+0.0341 [+0.0256, +0.0426]	+0.0012 [−0.0032, +0.0056]

D.1 PAIRED-RUN UNCERTAINTY

For each fixed setting, $d_s = \mathrm{PCC}_{\mathrm{CS},s} - \mathrm{PCC}_{\mathrm{Random},s}$ pairs seeds $s \in \{11, 22, 33, 44, 55\}$. Table 24 reports $\bar{d} \pm s_d$ and the nominal interval $\bar{d} \pm t_{0.975,4} s_d/\sqrt{5}$, assuming independent approximately normal seed-level differences conditional on the fixed partition and candidate language. Table 1 instead summarizes the four effects obtained after averaging seeds within each setting. The settings share two cohorts, budgets are repeated checkpoints of each run, and both backbone comparisons use the same Random-routing results. We therefore interpret the four-setting intervals descriptively and report seed-level uncertainty separately. Intervals are unadjusted across metrics and budgets. The accompanying CSVs retain all policy/checkpoint values and paired differences. For LKCP, seeds 1729–1733 are matched by identifier on the same compound-disjoint test partition (16 compounds and 432 conditions). Table 25 describes refit variation conditional on these compounds. Positive differences favor the CS-model for PCC and R^2; negative differences favor it for MSE. Seed matching preserves the run protocol, while architectures retain their own stochastic operations.

Table 24: **Paired-run CellScientist-minus-Random PCC differences.** Each cell gives mean±sample SD across five runs, with a nominal 95% paired-t interval below, conditional on the fixed task partition.

Backbone	Dataset	Split	$B = 3$	$B = 5$	$B = 10$
Gemini 3 Pro	BBBC036	Plate	$+0.0400 \pm 0.0491$ $[-0.0210, +0.1010]$	$+0.0397 \pm 0.0503$ $[-0.0228, +0.1022]$	$+0.0066 \pm 0.0067$ $[-0.0017, +0.0149]$
Gemini 3 Pro	BBBC036	SMILES	$+0.0284 \pm 0.0406$ $[-0.0221, +0.0788]$	$+0.0297 \pm 0.0414$ $[-0.0217, +0.0811]$	-0.0001 ± 0.0019 $[-0.0024, +0.0023]$
Gemini 3 Pro	BBBC047	Plate	$+0.0364 \pm 0.0498$ $[-0.0254, +0.0983]$	$+0.0391 \pm 0.0514$ $[-0.0247, +0.1028]$	$+0.0023 \pm 0.0033$ $[-0.0017, +0.0064]$
Gemini 3 Pro	BBBC047	SMILES	$+0.0398 \pm 0.0534$ $[-0.0265, +0.1061]$	$+0.0407 \pm 0.0583$ $[-0.0317, +0.1130]$	$+0.0083 \pm 0.0061$ $[+0.0007, +0.0159]$
Qwen2.5-0.5B	BBBC036	Plate	$+0.0316 \pm 0.0504$ $[-0.0309, +0.0942]$	$+0.0289 \pm 0.0510$ $[-0.0345, +0.0922]$	$+0.0031 \pm 0.0088$ $[-0.0078, +0.0140]$
Qwen2.5-0.5B	BBBC036	SMILES	$+0.0304 \pm 0.0410$ $[-0.0205, +0.0813]$	$+0.0308 \pm 0.0415$ $[-0.0207, +0.0823]$	-0.0001 ± 0.0019 $[-0.0024, +0.0023]$
Qwen2.5-0.5B	BBBC047	Plate	$+0.0397 \pm 0.0501$ $[-0.0225, +0.1020]$	$+0.0359 \pm 0.0518$ $[-0.0285, +0.1002]$	-0.0020 ± 0.0039 $[-0.0068, +0.0028]$
Qwen2.5-0.5B	BBBC047	SMILES	$+0.0439 \pm 0.0534$ $[-0.0225, +0.1102]$	$+0.0407 \pm 0.0582$ $[-0.0315, +0.1130]$	$+0.0039 \pm 0.0075$ $[-0.0054, +0.0132]$

Table 25: **LKCP seed-matched CS-model-minus-RealMLP differences.** Mean±sample SD and nominal 95% paired-t intervals use five refits on the fixed test partition.

Frozen design	ΔPCC	ΔMSE	ΔR^2
CS-model (BBBC036)	-0.0046 ± 0.0999 $[-0.1286, +0.1194]$	-0.0112 ± 0.1604 $[-0.2104, +0.1880]$	$+0.0006 \pm 0.0299$ $[-0.0365, +0.0377]$
CS-model (BBBC047)	$+0.0270 \pm 0.0672$ $[-0.0565, +0.1105]$	$+0.0583 \pm 0.1996$ $[-0.1896, +0.3061]$	-0.0475 ± 0.0564 $[-0.1175, +0.0225]$

E Detailed LCA and PDR Audit Results

Table 3 summarizes the HRT, LCA, and PDR results in the main text. Here we report LCA outcomes and PDR results by component address, including registered case counts, decisions, localization, repair success, unrelated edits, and repair rounds.

E.1 Registered LCA Outcome Audit

The LCA audit constructs 40 single-violation request records and 20 legal controls. The validator repairs 15 records, blocks 15 at the pre-run gate, and rejects 10 injected runtime-output records. Legal controls remain unchanged, yielding 60/60 correct decisions on these registered cases. The 30 pre-run interventions count requests corrected or stopped before dispatch in this audit, rather than timed model-training jobs. Its registered families test the implemented checker predicates.

E.2 Held-Out Registered-Fault PDR Audit

The PDR audit constructs four variants of each of five typed-state fault families, with one variant per family assigned to calibration and three to reporting. The controller receives a candidate configuration and constructed metric values; the diagnostic label is retained for scoring. Repair is the restoration of the registered configuration condition, checked without refitting: required conditioning, response target, input/output capacity, or regularization. Flat retry, Random routing, and structured routing receive the same cases. Structured routing restores every held-out state with 0 unrelated edits and 1.00 mean round. These are localization and state-restoration measurements within known fault families.

Table 26: **Registered LCA audit grouped by outcome.** Each positive request record contains one injected violation; legal controls contain none. "Eligible" denotes acceptance after record repair, and "Avoided" counts requests corrected or stopped at the pre-run gate. Runtime records are injected for post-run checks. Across these registered cases, violation recall, exact-code accuracy, legal-case specificity, and repair-or-safe-block containment are all 100%.

LCA outcome	Registered cases	Stage	Cases	Detected / accepted	Observed action	Eligible	Avoided
Repair	Protected-state mutation; interface mismatch; route mismatch	Pre	15	**15/15 detected**	**15/15 repaired**	**15/15**	15
Pre-run block	Out-of-domain candidate; non-atomic edit; repair budget exhausted	Pre	15	**15/15 detected**	**15/15 blocked**	0/15	15
Post-run block	Output-shape mismatch; non-finite output	Post	10	**10/10 detected**	**10/10 blocked**	0/10	0
Legal controls	Contract-preserving candidates across registered interfaces and component domains	Pre	20	**20/20 accepted**	Unchanged	**20/20**	0
Overall	All registered violation and legal-control cases		**60**	**60/60 correct**	**40/40 contained**	**35/60 eligible**	**30**

Table 27: **Held-out registered-fault PDR audit by component address.** Five injected typed-state cases are used for calibration and 15 for reporting, with three reporting variants per address. Structured PDR uses Gemini 3 Pro to select an allow-listed pair from candidate-state and metric feedback. Repair denotes restoration of the registered field condition. The repair budget is five rounds; displayed means assign five rounds to unrepaired cases. Best values within each address are **bolded**; ties share emphasis.

Routing policy	Expected address	Cases	Top-1 accuracy ↑	Top-2 recall ↑	Repair success ↑	Mean unrelated edits ↓	Mean repair rounds ↓
Flat retry	Conditioning	3	66.7%	N/A	0.0%	4.33	5.00
	Response parameterization	3	0.0%	N/A	**100%**	1.00	2.00
	Representation	3	0.0%	N/A	0.0%	4.00	5.00
	Decoder	3	0.0%	N/A	0.0%	4.33	5.00
	Regularization	3	0.0%	N/A	66.7%	2.33	3.67
	All held-out	15	13.3%	N/A	33.3%	3.20	4.13
Random routing	Conditioning	3	0.0%	0.0%	0.0%	5.00	5.00
	Response parameterization	3	33.3%	33.3%	66.7%	2.67	3.33
	Representation	3	0.0%	0.0%	66.7%	3.33	4.00
	Decoder	3	0.0%	33.3%	**100%**	2.33	3.33
	Regularization	3	0.0%	0.0%	66.7%	3.00	3.67
	All held-out	15	6.7%	13.3%	60.0%	3.27	3.87
Structured PDR	Conditioning	3	**100%**	**100%**	**100%**	**0.00**	**1.00**
	Response parameterization	3	**100%**	**100%**	**100%**	**0.00**	**1.00**
	Representation	3	**100%**	**100%**	**100%**	**0.00**	**1.00**
	Decoder	3	**100%**	**100%**	**100%**	**0.00**	**1.00**
	Regularization	3	**100%**	**100%**	**100%**	**0.00**	**1.00**
	All held-out	15	**100%**	**100%**	**100%**	**0.00**	**1.00**

Table 28: **Validation-selected morphology-response models.** CS-model denotes the task-specific predictor selected by CellScientist. Entries are five-fold validation mean±sample SD under SMILES- and plate-based splits. Blue shading identifies CellScientist models; **bold** marks the best mean within each comparison.

Method	Venue	SMILES-based Split			Plate-based Split		
		MSE↓	PCC↑	R^2↑	MSE↓	PCC↑	R^2↑
BBBC036							
# **Existing reference methods**							
TabR	ICLR'24	$3.5518_{\pm 0.24}$	$0.2747_{\pm 0.03}$	$0.0119_{\pm 0.02}$	$2.8759_{\pm 0.13}$	$0.5065_{\pm 0.03}$	$0.1855_{\pm 0.02}$
RealMLP	NeurIPS'24	$3.5305_{\pm 0.16}$	$0.3301_{\pm 0.01}$	$0.0049_{\pm 0.02}$	$2.5740_{\pm 0.16}$	$0.5856_{\pm 0.02}$	$0.2530_{\pm 0.03}$
RF-TD	NeurIPS'24	$3.5951_{\pm 0.19}$	$0.2447_{\pm 0.01}$	$0.0035_{\pm 0.01}$	$3.5978_{\pm 0.14}$	$0.2466_{\pm 0.02}$	$0.0152_{\pm 0.01}$
□ **Search-based baselines**							
FLAML	MLSys'21	$3.4217_{\pm 0.18}$	$0.3486_{\pm 0.02}$	$0.0379_{\pm 0.02}$	$2.4918_{\pm 0.14}$	$0.5987_{\pm 0.02}$	$0.2784_{\pm 0.03}$
Random Search	JMLR'24	$3.4763_{\pm 0.21}$	$0.3392_{\pm 0.02}$	$0.0246_{\pm 0.02}$	$2.5335_{\pm 0.15}$	$0.5904_{\pm 0.02}$	$0.2649_{\pm 0.03}$
△ **Agent-designed methods**							
CellForge	arXiv'25	$3.4568_{\pm 0.19}$	$0.3415_{\pm 0.03}$	$0.0464_{\pm 0.02}$	$2.5187_{\pm 0.14}$	$0.6013_{\pm 0.02}$	$0.2878_{\pm 0.03}$
CS-model	Ours	$\mathbf{3.3115}_{\pm 0.20}$	$\mathbf{0.3688}_{\pm 0.03}$	$\mathbf{0.0717}_{\pm 0.02}$	$\mathbf{2.3662}_{\pm 0.15}$	$\mathbf{0.6212}_{\pm 0.02}$	$\mathbf{0.3165}_{\pm 0.03}$
BBBC047							
# **Existing reference methods**							
TabR	ICLR'24	$2.9382_{\pm 0.04}$	$0.3552_{\pm 0.01}$	$0.0059_{\pm 0.01}$	$2.7930_{\pm 0.14}$	$0.4094_{\pm 0.01}$	$0.0568_{\pm 0.03}$
RealMLP	NeurIPS'24	$2.8441_{\pm 0.03}$	$0.3985_{\pm 0.01}$	$0.0285_{\pm 0.01}$	$2.6475_{\pm 0.10}$	$0.4597_{\pm 0.01}$	$0.1332_{\pm 0.02}$
RF-TD	NeurIPS'24	$2.8850_{\pm 0.05}$	$0.3746_{\pm 0.01}$	$0.0281_{\pm 0.01}$	$2.8174_{\pm 0.16}$	$0.4005_{\pm 0.01}$	$0.0615_{\pm 0.01}$
□ **Search-based baselines**							
FLAML	MLSys'21	$2.8126_{\pm 0.06}$	$0.4127_{\pm 0.01}$	$0.0435_{\pm 0.01}$	$2.6158_{\pm 0.11}$	$0.4698_{\pm 0.01}$	$0.1407_{\pm 0.02}$
Random Search	JMLR'24	$2.8314_{\pm 0.07}$	$0.4059_{\pm 0.01}$	$0.0358_{\pm 0.01}$	$2.6342_{\pm 0.12}$	$0.4635_{\pm 0.01}$	$0.1362_{\pm 0.02}$
△ **Agent-designed methods**							
CellForge	arXiv'25	$2.7869_{\pm 0.09}$	$0.4154_{\pm 0.01}$	$0.0476_{\pm 0.01}$	$2.5908_{\pm 0.10}$	$0.4772_{\pm 0.02}$	$0.1471_{\pm 0.02}$
CS-model	Ours	$\mathbf{2.7612}_{\pm 0.11}$	$\mathbf{0.4368}_{\pm 0.01}$	$\mathbf{0.0687}_{\pm 0.01}$	$\mathbf{2.5704}_{\pm 0.11}$	$\mathbf{0.4844}_{\pm 0.02}$	$\mathbf{0.1538}_{\pm 0.03}$
CPG0016							
# **Existing reference methods**							
TabR	ICLR'24	$2.2622_{\pm 0.12}$	$0.0770_{\pm 0.01}$	$-0.5516_{\pm 0.04}$	$2.0667_{\pm 0.04}$	$0.1953_{\pm 0.01}$	$-0.4128_{\pm 0.02}$
RealMLP	NeurIPS'24	$1.4174_{\pm 0.08}$	$0.2616_{\pm 0.02}$	$0.0297_{\pm 0.01}$	$1.2105_{\pm 0.01}$	$0.4418_{\pm 0.01}$	$0.1717_{\pm 0.01}$
RF-TD	NeurIPS'24	$1.4673_{\pm 0.09}$	$0.1354_{\pm 0.01}$	$-0.0035_{\pm 0.01}$	$1.3400_{\pm 0.01}$	$0.3213_{\pm 0.01}$	$0.0849_{\pm 0.01}$
□ **Search-based baselines**							
FLAML	MLSys'21	$1.2865_{\pm 0.05}$	$0.3748_{\pm 0.01}$	$0.1183_{\pm 0.01}$	$1.1206_{\pm 0.03}$	$0.5037_{\pm 0.02}$	$0.2331_{\pm 0.02}$
Random Search	JMLR'24	$1.3428_{\pm 0.06}$	$0.3189_{\pm 0.02}$	$0.0747_{\pm 0.02}$	$1.1674_{\pm 0.03}$	$0.4668_{\pm 0.02}$	$0.1994_{\pm 0.02}$
△ **Agent-designed methods**							
CellForge	arXiv'25	$1.2849_{\pm 0.03}$	$0.3716_{\pm 0.01}$	$0.1272_{\pm 0.01}$	$1.0458_{\pm 0.02}$	$0.5187_{\pm 0.03}$	$0.2376_{\pm 0.03}$
CS-model	Ours	$\mathbf{1.1113}_{\pm 0.01}$	$\mathbf{0.5093}_{\pm 0.01}$	$\mathbf{0.2383}_{\pm 0.01}$	$\mathbf{0.9937}_{\pm 0.02}$	$\mathbf{0.5761}_{\pm 0.04}$	$\mathbf{0.3117}_{\pm 0.04}$

F Additional Experimental Analysis

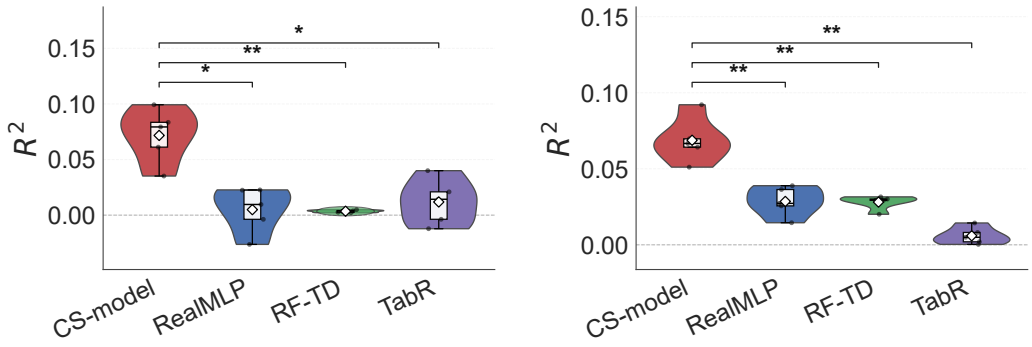
This appendix characterizes the high-freedom workflow, evaluates recovery of responsive morphology features, and reports additional single-cell results. These analyses complement the main results under their respective evaluation protocols.

F.1 Validation-Selected Discovery Outcomes

The high-freedom workflow selects task-specific predictors for morphology and transcriptomic response tasks. Table 28 reports five-fold validation mean and sample SD under exact-SMILES and plate grouping. CS-model has the best mean MSE, PCC, and R^2 among the listed methods in these comparisons. Figure 7 shows the corresponding fold-level R^2 values and paired-test markers; response-feature metrics follow in Appendix F.3.

The morphology baselines are TabR (Gorishniy et al., 2024), RealMLP and RF-TD (Holzmüller et al., 2024), FLAML (Wang et al., 2021), Random Search (Gijsbers et al., 2024), and CellForge (Tang et al., 2025). In addition to BBBC036 and BBBC047, CPG0016 extends the morphology search to a larger compound collection (Chandrasekaran et al., 2023; Arevalo et al., 2024).

Additional response spaces. Table 29 reports task-specific searches on the Seven Cell Lines and Full Data settings of LINCS2020 L1000 (Subramanian et al., 2017; Xie et al., 2022). The listed predictors include DLEPS (Zhu et al., 2021), DeepCE (Pham et al., 2021), CIGER (Pham et al., 2022), MultiDCP (Wu et al., 2022), and TranSiGen (Tong et al., 2024). Metrics cover RMSE, PCC, and positive/negative P@100. Appendix F.4 gives results for the Norman and Papalexi single-cell



(a) BBBC036. (b) BBBC047.

Figure 7: **Fold-level comparison on BBBC036 and BBBC047.** Violins show five-fold R^2; boxes, points, and diamonds denote quartiles, folds, and means. Stars: uncorrected paired t-tests, $^{*}p < 0.05$, $^{**}p < 0.01$, $^{***}p < 0.001$.

benchmarks (Norman et al., 2019; Papalexi et al., 2021), including separate Papalexi RNA and protein readouts. Each task uses its own search and training run.

Table 29: **LINCS2020 transcriptomic perturbation prediction. CS-model** achieves the best mean results among the listed methods in both the Seven Cell Lines and Full Data settings. Each uses an independent task-specific search. Best means are **bolded**.

Dataset	Model	RMSE ↓	PCC ↑	Positive P@100 ↑	Negative P@100 ↑
Seven Cell	DLEPS	$1.4954_{\pm 0.04}$	$0.4268_{\pm 0.02}$	$0.2460_{\pm 0.01}$	$0.3087_{\pm 0.01}$
	DeepCE	$1.7728_{\pm 0.03}$	$0.4121_{\pm 0.01}$	$0.2282_{\pm 0.02}$	$0.2713_{\pm 0.02}$
	CIGER	$4.4646_{\pm 1.06}$	$0.4165_{\pm 0.01}$	$0.2337_{\pm 0.02}$	$0.2849_{\pm 0.02}$
	MultiDCP	$1.7697_{\pm 0.03}$	$0.4461_{\pm 0.01}$	$0.2378_{\pm 0.02}$	$0.2872_{\pm 0.01}$
	TranSiGen	$0.6903_{\pm 0.02}$	$0.5332_{\pm 0.02}$	$0.3785_{\pm 0.01}$	$0.3886_{\pm 0.02}$
	CS-model	$\mathbf{0.5088}_{\pm 0.02}$	$\mathbf{0.6183}_{\pm 0.02}$	$\mathbf{0.4450}_{\pm 0.02}$	$\mathbf{0.4481}_{\pm 0.02}$
Full Data	TranSiGen	$0.5186_{\pm 0.01}$	$0.6076_{\pm 0.01}$	$0.4353_{\pm 0.01}$	$0.4403_{\pm 0.01}$
	CS-model	$\mathbf{0.4967}_{\pm 0.01}$	$\mathbf{0.6368}_{\pm 0.01}$	$\mathbf{0.4583}_{\pm 0.01}$	$\mathbf{0.4652}_{\pm 0.01}$

F.2 End-to-End Open-Workflow Characterization

Open-workflow evidence. Tables 28 and 29 report task-specific predictors selected by the high-freedom CellScientist workflow under morphology and transcriptomic protocols. Figures 7 and 4 show fold-level comparisons and non-monotonic revision trajectories, respectively. These results assess predictor quality; the main-text RQ1 comparison separately compares controller policies within a fixed candidate language.

The RQ1 study compares controller policies within a fixed executable language. Table 30 complements that audit with an end-to-end comparison in which implementations and selected model families can vary. Cumulative configurations share the task specification, split, validation metric, iteration budget, repair budget, execution timeout, and environment. The comparison records execution success, rule-compliant success, predictive quality, and runtime as structured task state, configured checks, and history-conditioned revision are added.

On both datasets, M4 achieved the highest R-SR, Avg PCC, and Best PCC, and the lowest runtime among the tested open-workflow configurations. Figure 4 records exploratory regressions and validation-retained improvements, linking these aggregate results to the case-study trajectories.

Table 30: **End-to-end open-workflow characterization.** SR denotes execution success, R-SR denotes rule-compliant success, and Avg/Best PCC summarize validation performance over eligible runs. Time reports wall-clock runtime. This high-freedom characterization complements the fixed-language comparisons in RQ1.

ID	Method Configuration	BBBC036 (SMILES-based Split)					BBBC047 (SMILES-based Split)				
		SR↑	R-SR↑	Avg PCC↑	Best PCC↑	Time (h)↓	SR↑	R-SR↑	Avg PCC↑	Best PCC↑	Time (h)↓
M0	Prompt + LLM + Code Executor	0.526	0.526	0.1357	0.1682	10.25	0.368	0.368	0.1422	0.1752	15.58
M1	M0 + Re-prompting (Naive)	0.263	0.263	0.2156	0.2485	17.96	0.210	0.210	0.2267	0.2581	27.29
M2	M1 + Structured Task State (HRT)	0.947	0.684	0.2771	0.3050	10.34	0.941	0.588	0.2472	0.2775	15.96
M3	M2 + Protected Editing / Checks (LCA)	**1.000**	0.737	0.3183	0.3214	8.46	**1.000**	0.684	0.3315	0.3962	13.38
M4	**M3 + History-Conditioned Revision (PDR)**	**1.000**	**0.789**	**0.3413**	**0.3688**	**7.93**	**1.000**	**0.714**	**0.3539**	**0.4368**	**12.58**

Table 31: **Response-sensitive morphology metrics.** Response-feature metrics evaluate prediction quality on the top-K most responsive morphology features. CS-model has the best mean values among the listed methods under both SMILES-based and plate-based splits in this table. Best mean values are **bolded**.

Method	Venue	SMILES-based Split				Plate-based Split			
		$\text{RMSE}_{\text{RF-20}}\downarrow$	$\text{RMSE}_{\text{RF-50}}\downarrow$	$\text{PCC}_{\text{RF-20}}\uparrow$	$\text{PCC}_{\text{RF-50}}\uparrow$	$\text{RMSE}_{\text{RF-20}}\downarrow$	$\text{RMSE}_{\text{RF-50}}\downarrow$	$\text{PCC}_{\text{RF-20}}\uparrow$	$\text{PCC}_{\text{RF-50}}\uparrow$
BBBC036									
# Existing reference methods									
TabR	ICLR'24	$4.1306_{\pm 0.03}$	$3.3883_{\pm 0.04}$	$0.2742_{\pm 0.01}$	$0.2584_{\pm 0.01}$	$3.8974_{\pm 0.12}$	$3.1610_{\pm 0.07}$	$0.3784_{\pm 0.05}$	$0.3783_{\pm 0.05}$
RealMLP	NeurIPS'24	$3.9910_{\pm 0.03}$	$3.2980_{\pm 0.03}$	$0.3170_{\pm 0.01}$	$0.2929_{\pm 0.01}$	$3.5303_{\pm 0.13}$	$2.8547_{\pm 0.08}$	$0.4636_{\pm 0.03}$	$0.4681_{\pm 0.02}$
RF-TD	NeurIPS'24	$4.1499_{\pm 0.02}$	$3.4356_{\pm 0.03}$	$0.2445_{\pm 0.01}$	$0.2227_{\pm 0.01}$	$3.9590_{\pm 0.13}$	$3.2927_{\pm 0.07}$	$0.5268_{\pm 0.05}$	$0.4649_{\pm 0.05}$
□ Search-based baselines									
FLAML	MLSys'21	$3.9892_{\pm 0.04}$	$3.2897_{\pm 0.04}$	$0.3297_{\pm 0.01}$	$0.3079_{\pm 0.01}$	$3.3658_{\pm 0.11}$	$2.7702_{\pm 0.08}$	$0.6486_{\pm 0.03}$	$0.6705_{\pm 0.03}$
Random Search	JMLR'24	$3.9948_{\pm 0.05}$	$3.2964_{\pm 0.05}$	$0.3218_{\pm 0.01}$	$0.2978_{\pm 0.01}$	$3.4316_{\pm 0.12}$	$2.8357_{\pm 0.08}$	$0.6235_{\pm 0.04}$	$0.6468_{\pm 0.04}$
△ Agent-designed methods									
CellForge	arXiv'25	$3.9903_{\pm 0.05}$	$3.2928_{\pm 0.05}$	$0.3262_{\pm 0.01}$	$0.3034_{\pm 0.01}$	$3.3869_{\pm 0.10}$	$2.7883_{\pm 0.08}$	$0.6574_{\pm 0.03}$	$0.6813_{\pm 0.03}$
CS-model	Ours	$\mathbf{3.9865}_{\pm 0.09}$	$\mathbf{3.2849}_{\pm 0.08}$	$\mathbf{0.3339}_{\pm 0.01}$	$\mathbf{0.3139}_{\pm 0.01}$	$\mathbf{3.2817}_{\pm 0.11}$	$\mathbf{2.7046}_{\pm 0.09}$	$\mathbf{0.6856}_{\pm 0.02}$	$\mathbf{0.7082}_{\pm 0.02}$
BBBC047									
# Existing reference methods									
TabR	ICLR'24	$6.2132_{\pm 0.02}$	$4.6263_{\pm 0.01}$	$0.3355_{\pm 0.01}$	$0.3898_{\pm 0.01}$	$6.2120_{\pm 0.05}$	$4.6035_{\pm 0.04}$	$0.3398_{\pm 0.01}$	$0.4009_{\pm 0.01}$
RealMLP	NeurIPS'24	$5.8479_{\pm 0.02}$	$4.4168_{\pm 0.01}$	$0.3678_{\pm 0.01}$	$0.4851_{\pm 0.01}$	$6.1791_{\pm 0.06}$	$4.5301_{\pm 0.04}$	$0.3836_{\pm 0.02}$	$0.4270_{\pm 0.01}$
RF-TD	NeurIPS'24	$6.0601_{\pm 0.03}$	$4.5652_{\pm 0.02}$	$0.3742_{\pm 0.01}$	$0.4153_{\pm 0.01}$	$6.0934_{\pm 0.05}$	$4.5715_{\pm 0.04}$	$0.3591_{\pm 0.02}$	$0.4063_{\pm 0.01}$
□ Search-based baselines									
FLAML	MLSys'21	$5.2847_{\pm 0.04}$	$4.0209_{\pm 0.03}$	$0.3685_{\pm 0.01}$	$0.4979_{\pm 0.01}$	$5.5128_{\pm 0.05}$	$4.0879_{\pm 0.04}$	$0.3892_{\pm 0.02}$	$0.5096_{\pm 0.01}$
Random Search	JMLR'24	$5.4362_{\pm 0.05}$	$4.1328_{\pm 0.04}$	$0.3601_{\pm 0.01}$	$0.4887_{\pm 0.01}$	$5.6315_{\pm 0.06}$	$4.1746_{\pm 0.04}$	$0.3838_{\pm 0.02}$	$0.4983_{\pm 0.01}$
△ Agent-designed methods									
CellForge	arXiv'25	$5.1164_{\pm 0.04}$	$3.8946_{\pm 0.03}$	$0.3748_{\pm 0.01}$	$0.5096_{\pm 0.01}$	$5.3846_{\pm 0.05}$	$3.9907_{\pm 0.04}$	$0.3967_{\pm 0.02}$	$0.5208_{\pm 0.01}$
CS-model	Ours	$\mathbf{4.9589}_{\pm 0.03}$	$\mathbf{3.7581}_{\pm 0.03}$	$\mathbf{0.3796}_{\pm 0.01}$	$\mathbf{0.5172}_{\pm 0.01}$	$\mathbf{5.2274}_{\pm 0.04}$	$\mathbf{3.8788}_{\pm 0.04}$	$\mathbf{0.4025}_{\pm 0.02}$	$\mathbf{0.5312}_{\pm 0.01}$
CPG0016									
# Existing reference methods									
TabR	ICLR'24	$2.6335_{\pm 0.05}$	$2.3924_{\pm 0.05}$	$0.0896_{\pm 0.01}$	$0.0865_{\pm 0.01}$	$2.5259_{\pm 0.01}$	$2.2874_{\pm 0.01}$	$0.1330_{\pm 0.01}$	$0.1309_{\pm 0.01}$
RealMLP	NeurIPS'24	$2.5152_{\pm 0.04}$	$2.2456_{\pm 0.04}$	$0.3776_{\pm 0.02}$	$0.3579_{\pm 0.02}$	$2.3456_{\pm 0.01}$	$2.0846_{\pm 0.01}$	$0.4658_{\pm 0.01}$	$0.4504_{\pm 0.01}$
RF-TD	NeurIPS'24	$2.5885_{\pm 0.05}$	$2.3130_{\pm 0.04}$	$0.0170_{\pm 0.01}$	$0.0224_{\pm 0.01}$	$2.4955_{\pm 0.01}$	$2.2221_{\pm 0.01}$	$0.0758_{\pm 0.01}$	$0.0835_{\pm 0.01}$
□ Search-based baselines									
FLAML	MLSys'21	$2.1126_{\pm 0.03}$	$1.9075_{\pm 0.03}$	$0.5062_{\pm 0.01}$	$0.4974_{\pm 0.01}$	$2.0249_{\pm 0.02}$	$1.8186_{\pm 0.02}$	$0.5614_{\pm 0.02}$	$0.5537_{\pm 0.02}$
Random Search	JMLR'24	$2.2484_{\pm 0.04}$	$2.0293_{\pm 0.04}$	$0.4385_{\pm 0.02}$	$0.4262_{\pm 0.02}$	$2.1227_{\pm 0.02}$	$1.9084_{\pm 0.02}$	$0.5126_{\pm 0.02}$	$0.5028_{\pm 0.02}$
△ Agent-designed methods									
CellForge	arXiv'25	$2.1038_{\pm 0.03}$	$1.8937_{\pm 0.03}$	$0.4978_{\pm 0.01}$	$0.4896_{\pm 0.01}$	$1.9426_{\pm 0.02}$	$1.7438_{\pm 0.02}$	$0.5867_{\pm 0.03}$	$0.5789_{\pm 0.03}$
CS-model	Ours	$\mathbf{1.8127}_{\pm 0.02}$	$\mathbf{1.6328}_{\pm 0.01}$	$\mathbf{0.5924}_{\pm 0.01}$	$\mathbf{0.5890}_{\pm 0.01}$	$\mathbf{1.7303}_{\pm 0.02}$	$\mathbf{1.5517}_{\pm 0.02}$	$\mathbf{0.6529}_{\pm 0.03}$	$\mathbf{0.6518}_{\pm 0.03}$

F.3 Response-sensitive Morphology Metrics

Table 31 describes the observed method differences on the most perturbation-responsive morphology features. We report response-feature RMSE and PCC over the top-K responsive features on BBBC036, BBBC047, and CPG0016 under both SMILES-based and plate-based splits.

F.4 Single-cell Perturbation Prediction

Table 32 reports task-specific workflow results on Norman and Papalexi, extending the evaluation beyond morphology and LINCS-style readouts. Norman evaluates transcriptional responses to gene activation; Papalexi evaluates RNA and protein responses as separate prediction tasks. Each comparison uses its dataset-specific protocol and reports PCC and PCC_{DE}.

Table 32: **Single-cell perturbation prediction.** We compare task-specific CS-model predictors with CellForge on genetic and multimodal single-cell benchmarks using PCC and $\mathrm{PCC}_{\mathrm{DE}}$. Entries are point estimates under the dataset-specific protocols. Best values are **bolded**.

Dataset	Method	PCC ↑	$\mathrm{PCC}_{\mathrm{DE}}$ ↑
CRISPRa Gene-Activation scRNA-seq Dataset			
Norman	CellForge	0.9846	0.8109
	CS-model	**0.9871**	**0.9060**
CRISPR–Cas9 ECCITE-seq RNA Dataset			
Papalexi RNA	CellForge	0.6935	0.6406
	CS-model	**0.8193**	**0.8311**
CRISPR–Cas9 ECCITE-seq Protein Dataset			
Papalexi Protein	CellForge	0.7495	0.7409
	CS-model	**0.7941**	**0.8275**