PROTOCOL-CERTIFIED WORKFLOW SEARCH FOR CELLULAR PERTURBATION-RESPONSE PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Language model agents now automate the whole modelling pipeline for cellular perturbation response, selecting among candidate workflows by a single validation score. We argue that this score is the wrong contract, because the protocol violations that matter most in this domain all *raise* it: leaking held out rows into selection, fitting normalization on the full corpus, admitting post treatment measurements as inputs, exploiting plate identity under a plate based split, selecting responsive genes with the model's own output, or requerying validation until the last edit looks good. None of these raises an exception, and validity is a property of the dependence structure among artifacts rather than of any single statement: execution feedback observes only crashes, and a local assertion observes only one statement, so neither reaches them. A search rewarded by that score therefore has a structural incentive to find them, and our audit confirms it: across nine violation categories the worst offender is not a weak agent but our own closed loop scaffold with certification disabled, at a 0.278 macro average violation rate against 0.169 for a generic AutoML pipeline. VERICELL moves admissibility out of the generator into a verifier. A taint labelled provenance graph collapses a heterogeneous family of protocol errors into one non interference property, discharged by a sound type and effect analysis composed with fail closed reference monitors; an audited reusable holdout bounds the adaptive generalization gap; and a claim checker ties every reported number to certified evidence. Certification cuts the macro average violation rate to 0.007 at verifier precision and recall above 0.93, and the validation to test gap from 0.120 to 0.016, while improving accuracy where signal exists (LINCS L1000 PCC $0.463 \rightarrow 0.557$, BBBC021 PCC $0.264 \rightarrow 0.355$) and costing nothing where it does not. Throughout, *certified* means machine checked protocol validity relative to a written contract and a bounded validation reuse gap, not biological correctness.

1 INTRODUCTION

Mapping how a cell responds to a chemical or genetic perturbation is the quantitative core of the virtual cell program. Cell Painting (Bray et al., 2016; Caicedo et al., 2017) and LINCS L1000 (Subramanian et al., 2017; Lamb et al., 2006) now produce measurements at a scale where exhaustive experimentation is infeasible, so a model that generalizes to unseen compounds, plates, or genetic targets has become the practical substitute for the next screen. The problem is hard on its own terms: the response is high dimensional, the assay carries strong batch structure, and the meaningful signal often sits in a few responsive features rather than the global profile.

What makes this domain unusual is not the difficulty of the model class but the fragility of the experimental semantics around it. A perturbation response benchmark is not a bare supervised dataset; it is a dataset plus a protocol. The protocol fixes which rows may be seen together, which measurements are inputs rather than targets, how normalization statistics are estimated (Caicedo et al., 2017), which features count as responsive, and how many times the validation partition may be consulted (Dwork et al., 2015b). Each encodes an experimental fact, and violating any yields a number that is arithmetically correct and scientifically void: the classical leakage failure (Kaufman et al., 2012), compounded here by batch structure strong enough that plate identity alone can carry the label (Way et al., 2022). None produces a runtime error: the pipeline completes, the metric improves, and the report reads well.

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.

Automated agents make this fragility consequential. An agent that writes, runs, and revises a modelling pipeline is steered by one scalar, the validation score, and every protocol violation listed above *increases* it. That a search optimizing a validation estimate inflates that estimate is long established (Cawley & Talbot, 2010; Varma & Simon, 2006); what is new is that an agent writing arbitrary code can reach violations no fixed search space contains. The search therefore has a structural incentive to discover them, and existing safeguards cannot see them. Execution feedback observes crashes, and a crash never occurs. Local assertions observe single statements, but leakage is a property of the dependence structure connecting many artifacts across a pipeline, and adaptive overfitting is a property of the entire sequence of runs rather than of any one run. Our audit makes the scale concrete: over five agent configurations and nine violation categories, macro average violation rates run from 0.169 for a generic AutoML pipeline to 0.278 for our own closed loop scaffold with certification disabled. The scaffold is the worst offender precisely because it searches hardest, which inverts the usual expectation that a stronger agent yields a better pipeline.

VERICELL responds by relocating admissibility from the generator to a verifier. A domain expert writes the protocol once as typed invariants; the agent then explores representations, architectures, and training strategies freely, and every candidate must carry a machine checked certificate before it may influence model selection. The technical core is that a heterogeneous family of protocol errors, unrelated at the level of code, collapses into a single property once execution is instrumented as a taint labelled provenance graph: non interference between a forbidden set of source labels and a class of sinks. That one property is discharged by a sound type and effect analysis over the data flow graph (Volpano et al., 1996; Cousot & Cousot, 1977), with the residual fragment depending on dynamic control flow deferred to fail closed runtime monitors (Schneider, 2000). Two further stages close the remaining gaps: an audited reusable holdout bounding the adaptive generalization gap under repeated validation (Dwork et al., 2015b), and a claim checker tying every quantitative statement in the agent's report back to a monitored evaluation node.

Our contributions are fourfold. First, we formalize protocol validity for perturbation response modelling as constraint algebra over typed invariants, and show that leakage, scope, shortcut, and circularity errors reduce to one non interference schema on a provenance graph. Second, we give a two stage certifier, sound by static over approximation and exact on the realized trace, with an audited validation mechanism carrying a high probability gap bound. Third, we quantify a problem the field lacks measurement for, auditing five agent configurations across nine categories on injected violation and open ended tasks. Fourth, we show certification costs nothing in accuracy: the macro average violation rate falls from 0.278 to 0.007 and the validation to test gap from 0.120 to 0.016, while PCC improves where signal exists and remains competitive where no method learns.

2 RELATED WORK

Perturbation response prediction. Transcriptomic response models including DeepCE (Pham et al., 2021), CIGER (Pham et al., 2022), MultiDCP (Wu et al., 2022), and TranSiGen (Tong et al., 2024) predict expression change from compound structure and cellular context; single cell methods (Lotfollahi et al., 2019; Bunne et al., 2023; Roohani et al., 2024) target genetic perturbations from pooled CRISPR screens (Dixit et al., 2016; Replogle et al., 2022); and morphology models operate on CellProfiler features (Carpenter et al., 2006). Language models are increasingly applied across this space (Jiang et al., 2025; Wang et al., 2025). All are fixed pipelines with protocols fixed by their authors. Our concern is orthogonal: we take the protocol as given and ask whether an automated search preserves it.

Automated machine learning and agentic synthesis. AutoML systems (Wang et al., 2021; Feurer et al., 2015; Bergstra & Bengio, 2012) optimize a validation objective over a fixed configuration space (He et al., 2021); language model agents extend this by writing arbitrary code. Both share the assumption we question, that the validation score is a sufficient contract, and our audit shows it fails for both, more severely for the more capable searcher: 0.169 against 0.278.

Leakage, information flow, and adaptive analysis. Leakage (Kaufman et al., 2012) and the selection bias from reusing a split for tuning and reporting (Cawley & Talbot, 2010; Varma & Simon, 2006; Krstajic et al., 2014) are both well documented, and both are usually addressed by checklists or by detectors written one per pattern. Non interference and taint tracking are standard in language based security (Sabelfeld & Myers, 2003; Volpano et al., 1996), and the reusable holdout

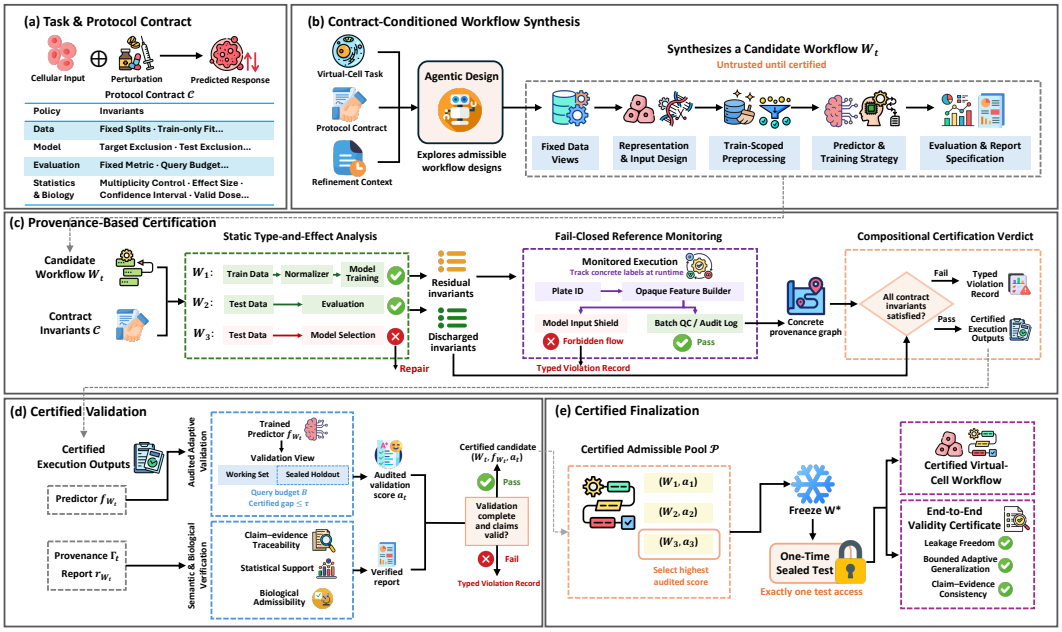


Figure 1: **VERICELL.** The agent proposes a candidate workflow conditioned on the contract $\mathcal{C}$, the run history $\mathcal{B}$, and prior typed violations. Static type and effect analysis computes the label fixpoint $\hat{\lambda}$ and discharges the covered invariants; residual invariants compile to fail closed monitors that wrap execution and emit the provenance graph Γ. Validation is answered only through the audited reusable holdout, which consumes overfitting budget B. The semantic checker ties each claim in the report r_W to a monitored evaluation node. Only candidates clearing all four stages enter the selection pool $\mathcal{P}$; violations return as typed repair signals rather than free form feedback. The sealed test partition D_{te} is queried once, for the pool best model.

restores a validity bound under repeated querying that degrades only poly logarithmically (Dwork et al., 2015b;a). Our contribution is to connect them to this domain: the protocol errors specific to perturbation assays, including plate shortcut and responsive feature circularity, are instances of the non interference schema, so one analysis discharges all of them, and the holdout becomes the validation interface of an agentic loop with an audited budget, which makes the resulting gap a certified quantity rather than an assumed one.

3 METHOD

VERICELL recasts autonomous virtual cell modelling as *protocol certified workflow synthesis*: the agent explores freely, but every candidate must carry a machine checked certificate that it preserves the experimental semantics of the task before it may influence model selection. The contribution is a single compositional validity certificate unifying three correctness notions prior agentic pipelines leave implicit: leakage freedom of the data flow, a bounded adaptive generalization gap under repeated validation, and consistency between the report's claims and its certified evidence. Unlike closed loop agents that treat feedback as free form code repair, VERICELL makes protocol violations first class semantic objects, so refinement is confined to a certified admissible region rather than merely encouraged toward it. Figure 1 gives the architecture.

3.1 PROBLEM FORMULATION

A perturbation response task is a tuple $\mathcal{T} = (\mathcal{X}, \mathcal{U}, \mathcal{Y}, \mathcal{S}, \mathcal{E})$ with baseline cellular inputs $\mathcal{X}$, perturbation descriptors $\mathcal{U}$ (compound structure, dose, or genetic target), response targets $\mathcal{Y}$, a split protocol $\mathcal{S}$ inducing fixed train, validation, and sealed test views $D_{\text{tr}}, D_{\text{va}}, D_{\text{te}}$, and an evaluation suite $\mathcal{E}$ with a designated selection functional. An agent emits a typed workflow $W \in \mathcal{W}$ that ingests data, fits a predictor f_W, and returns predictions, metrics, and a report r_W. We seek the workflow

whose sealed test utility is maximal among those admitted by a contract $\mathcal{C}$, with the test partition queried exactly once:

$$W^\star \in \arg\max_{W \in \mathcal{W}} \mathcal{U}_{\mathrm{te}}(f_W) \qquad \text{subject to} \qquad \mathsf{Cert}_{\mathcal{C}}(W) = 1. \tag{1}$$

The predicate $\mathsf{Cert}_{\mathcal{C}}$ is evaluated by the certifier, not the agent, so admissibility is decided by verification rather than by the generator that has an incentive to overfit the protocol.

3.2 Protocol Contracts as a Constraint Algebra

A contract is a finite set of typed invariants $\mathcal{C} = \{\rho_1, \ldots, \rho_m\}$ in four policy layers: data policy (split immutability, normalization and reference scope, leakage freedom), model policy (admissible and forbidden input channels), evaluation policy (fixed selection functional, reporting metrics, query budget, sealed test), and statistics policy (multiple testing correction, effect size and interval reporting, the admissible source of the responsive feature mask, biological admissibility of dose and perturbation ranges). Each invariant is a predicate $\rho : \Gamma \to \{0, 1\}$ over a provenance object Γ defined in Section 3.3, and admissibility is their conjunction:

$$\mathsf{Cert}_{\mathcal{C}}(W) = \bigwedge_{\rho \in \mathcal{C}} \rho\big(\Gamma(W)\big). \tag{2}$$

Contracts form a bounded lattice under union and intersection of their invariant sets, ordered by the inclusion of the admissible sets they induce, so the empty contract is top and the full invariant set is bottom. Task level and dataset level policies therefore compose by meet, which is union of invariants, and strengthening a contract can only shrink the admissible set: a workflow certified under $\mathcal{C}' \supseteq \mathcal{C}$ is certified under $\mathcal{C}$. A contract is written once by a domain expert and reused across all candidates, encoding what must remain invariant while leaving how to model entirely to the agent. Full listings are in Appendix A.2.

3.3 Provenance Grounded Non Interference Semantics

We instrument execution to emit a provenance directed acyclic graph (Buneman et al., 2001; Moreau et al., 2011) $\Gamma = (V, E, \lambda, \theta)$, where vertices are data artifacts (tensors, tables, fitted transforms, metric values), edges encode data dependences induced by operations θ, and $\lambda : V \to 2^L$ assigns taint labels from the lattice $\big(2^L, \subseteq, \cup, \cap\big)$ with

$$L = \{\mathrm{tr}, \mathrm{va}, \mathrm{te}, \mathrm{post}, \mathrm{pid}, \mathrm{ctrl}, \mathrm{deg}, \mathrm{out}\}. \tag{3}$$

Labels denote train, validation, and test membership, the post treatment target, plate identity, the control reference, the ground truth responsive feature mask, and any artifact derived from the model's own predictions. The first seven are seeded at data sources; out is seeded at every prediction node, which is what allows the circularity invariant below to be stated in the same schema instead of as a separate check. All labels propagate forward by set union along edges. Protocol invariants are expressed uniformly as non interference constraints between a forbidden source label set $S \subseteq L$ and a sink class σ:

$$\mathsf{NI}(S, \sigma) \;\equiv\; \forall\, v \in \mathrm{sink}_\sigma(\Gamma) :\; \lambda(v) \cap S = \varnothing. \tag{4}$$

This schema absorbs errors that look unrelated in code. Split leakage is $\mathsf{NI}(\{\mathrm{te}\}, \text{selection})$; input leakage is $\mathsf{NI}(\{\mathrm{post}, \mathrm{te}\}, \text{model input})$; a normalization scope violation is any fitted transform whose parameters carry a label beyond tr; the plate shortcut prohibition is $\mathsf{NI}(\{\mathrm{pid}\}, \text{model input})$; control reference misuse is $\mathsf{NI}(\{\mathrm{ctrl}\}, \text{model input})$; and responsive feature circularity is $\mathsf{NI}(\{\mathrm{out}\}, \text{responsive mask definition})$, whose sink class collects every node that defines the deg mask. Four of the nine audited categories reduce to this one schema in full, namely split leakage, preprocessing leakage, target and perturbation leakage, and batch shortcut, as does the circularity component of biological invalidity. The two evaluation categories, metric drift and adaptive overfitting, are properties of which node the selection functional reads and of the sequence of validation queries rather than of a single flow, and are enforced by the evaluation monitors of Section 3.5; statistical misuse, provenance loss, and the dose component of biological invalidity are checked by the semantic stage of Section 3.6. Appendix A.9 gives the complete map.

3.4 Two Stage Certification of the Data Flow

The certifier has four stages in total, of which the two described here act on the data flow; Sections 3.5 and 3.6 add the remaining two. The first stage lifts label propagation to the abstract lattice and computes, by monotone abstract interpretation (Cousot & Cousot, 1977; Hankin et al., 2010) over the data flow graph, an over approximation $\hat{\lambda} \geq \lambda$ at the least fixpoint of the transfer system:

$$\frac{v \in \mathrm{src}(\Gamma)}{\hat{\lambda}(v) = \mathrm{seed}(v)} \qquad \frac{u_1, \ldots, u_k \to v \in E}{\hat{\lambda}(v) \supseteq \bigcup_{i=1}^{k} \hat{\lambda}(u_i)} \qquad \frac{v = \mathrm{fit}(u_1, \ldots, u_k)}{\hat{\lambda}(v) \supseteq \bigcup_{i=1}^{k} \hat{\lambda}(u_i)}. \tag{5}$$

An invariant $\mathsf{NI}(S, \sigma)$ is statically discharged when $\hat{\lambda}(v) \cap S = \varnothing$ for all sinks in class σ. Invariants depending on data dependent control flow, dynamic indexing, or opaque external calls are marked residual and compiled into runtime reference monitors (Schneider, 2000) that wrap tensor and table operations, propagate concrete labels, and abort on the first breach with a structured violation record. Monitors coalesce labels along linear pipeline segments, holding instrumentation cost to a constant factor per operation. Appendix A.5 gives the abstract domain, the iteration order, the convergence bound, and the set of library operations whose label semantics we model.

Proposition 1 (Static soundness)**.** *If the analysis discharges $\mathsf{NI}(S, \sigma)$ at the fixpoint $\hat{\lambda}$, every concrete execution of W satisfies $\mathsf{NI}(S, \sigma)$. The static stage produces no false negatives on the covered fragment.*

Proposition 2 (Monitor exactness on the realized trace)**.** *For any concrete execution path, the monitor reports a violation of $\mathsf{NI}(S, \sigma)$ if and only if that violation occurs on the path.*

The stages are complementary rather than redundant: the static analysis quantifies over all executions of a fragment, the monitor is exact on one path, and neither subsumes the other. Soundness is one directional in two senses. The analysis may reject a safe workflow, and Appendix A.4 puts static precision at 0.903 to 0.931, so between one alarm in ten and one in fourteen is false. And it is silent outside the covered fragment, which is why the single stage recall in Appendix A.4 is below one without contradicting Proposition 1: the missed positives are exactly the instances the analysis marks residual and hands to the monitors. Proofs of both propositions are in Appendix A.8. That asymmetry is the right one for a certifier, since a false alarm costs a repair iteration and a false negative costs an invalid result.

3.5 Certified Adaptive Validation

Multi round refinement issues many validation queries, and naive reuse turns the validation estimate into an optimistically biased proxy for test performance. This is invisible to leakage checking. VeriCell answers every query through an audited reusable holdout: the validation view splits into a working set and a sealed holdout of size n; a query returns the working set value unless it deviates from the noised holdout value by more than a threshold, in which case a noised holdout value is returned and one unit of an overfitting budget B is consumed. The test partition is never accessed during refinement.

Theorem 1 (Certified adaptive gap)**.** *Fix a failure probability $\beta \in (0, 1)$ and an overfitting budget B. Let the validation queries be an adaptively chosen sequence of K statistical queries $\phi_k : \mathcal{Z} \to [0, 1]$ with population values $\mu_k = \mathbb{E}_{z \sim P}\big[\phi_k(z)\big]$, answered by the audited holdout of size n. Then, with probability at least $1 - \beta$ and for as long as the budget is unexhausted, the answers a_k satisfy*

$$\max_{k \leq K} \big| a_k - \mu_k \big| \;\leq\; \tau, \qquad \tau = c \sqrt{\frac{\big(B + \log(1/\beta)\big) \log(K/\beta)}{n}} \tag{6}$$

for a universal constant c. A selection metric that is not a statistical query, PCC in particular, is issued as a fixed vector of five clipped moment queries and recovered by a Lipschitz functional, which multiplies K by five and τ by the Lipschitz constant (Appendix A.7).

The selection signal therefore tracks population utility, and accepted improvements are not artifacts of adaptive overfitting; Section 4.4 reports $\tau = 0.192$ at our operating point. The same interface also pins the selection functional, since a query is answered only for the metric the contract designates and the monitor records which node it read, which is how the metric drift category of Section 3.3 is enforced. Proofs of both theorems are in Appendix A.8.

3.6 Semantic and Biological Validity Verification

From the report r_W we extract claim tuples (subject, metric, relation, magnitude, scope) and verify each against certified evidence. A reported metric must trace to a monitored evaluation node computed under the fixed selection functional. A comparative claim must be backed by a paired significance test with multiple testing correction (Benjamini & Hochberg, 1995), plus a reported effect size and interval. Mechanistic language is downgraded to a validation supported inductive bias statement unless licensed by an admissible causal test (Pearl et al., 2003; Schölkopf et al., 2021). Biological admissibility additionally requires dose and perturbation settings within dataset supported ranges, and responsive feature selection computed from ground truth and control artifacts only, formalized as $\mathsf{NI}(\{\mathrm{out}\}, \mathrm{deg})$ in the labels of Eq. equation 3, which removes the circularity that otherwise inflates response sensitive metrics.

3.7 Certified Closed Loop Refinement

The agent proposes a candidate conditioned on the contract, the run history, and prior violation records. The certifier applies the static, dynamic, audited validation, and semantic stages in sequence, routes any violation as a typed repair signal to a bounded local edit rather than a full restart, and admits only candidates clearing all stages into the selection pool. After the iteration budget is spent, the pool best model by audited validation utility is frozen and evaluated once on the sealed test partition; its certificate records the discharged invariants, the consumed query budget, and the certified gap. Algorithm 1 in Appendix A.1 states the loop.

Theorem 2 (End to end validity certificate)**.** *Let $W^\star$ be the model returned by Algorithm 1 under contract $\mathcal{C}$. With probability at least $1-\beta$, $W^\star$ satisfies simultaneously: (i) leakage freedom on all covered non interference invariants of $\mathcal{C}$; (ii) an adaptive gap $\big|\widehat{\mathcal{U}}_{\mathrm{va}}(f_{W^\star}) - \mathcal{U}(f_{W^\star})\big| \le \tau$ against the population utility $\mathcal{U}$, with τ from Theorem 1 instantiated at failure probability $\beta/3$, hence $\big|\widehat{\mathcal{U}}_{\mathrm{va}}(f_{W^\star}) - \mathcal{U}_{\mathrm{te}}(f_{W^\star})\big| \le \tau + \varepsilon_{\mathrm{te}}$ with $\varepsilon_{\mathrm{te}} = O\big(\sqrt{\log(3/\beta)/n_{\mathrm{te}}}\big)$, the single query concentration term for the frozen $f_{W^\star}$ under the same moment reduction; and (iii) claim evidence consistency of every quantitative statement in $r_{W^\star}$.*

The certificate is relative to the contract and to the analysed fragment, and concerns protocol validity rather than biological correctness; Appendix A.16 states each boundary.

4 Experiments

4.1 Setup

Benchmarks. We evaluate on two perturbation response settings under a single frozen protocol each. For Cell Painting morphology we use four public high content screening datasets in CellProfiler feature space, BBBC021, BBBC036, and BBBC047 from the Broad Bioimage Benchmark Collection (Ljosa et al., 2012; Gustafsdottir et al., 2013) and CPG0016 from the JUMP consortium (Chandrasekaran et al., 2023; 2024), under paired SMILES (Weininger, 1988) based and plate based splits, reporting global PCC and R^2. The two splits probe different failure modes: SMILES based splitting tests chemical generalization, plate based splitting tests whether a model has learned batch identity instead of biology. For transcriptomic prediction we use LINCS L1000 (Subramanian et al., 2017) in a dense Seven Cell Lines regime with a chemical blind split and a large scale Full Data regime, reporting RMSE, PCC, and Positive and Negative P@100, the precision at rank 100 of the predicted up and down regulated gene lists against the measured ones. Compound structures enter as SMILES (Weininger, 1988) and extended connectivity fingerprints (Rogers & Hahn, 2010) computed with RDKit (Landrum, 2013). Because MultiDCP does not support the Full Data regime, we additionally report a Common Support Full subset on which every baseline runs, so no comparison is made across differing method sets. We attempted three single cell benchmarks; neither our system nor the vanilla coding agent baseline completed full training and prediction on them, and we report that outcome in Appendix A.16 rather than omitting it.

Baselines. Morphology baselines span fixed predictors, namely TabR (Gorishniy et al., 2023), RealMLP (Holzmüller et al., 2024), and a random forest at the tuned default configuration of Holzmüller et al. (2024) following the tuning guidance of Probst et al. (2019), denoted RF-TD,

and generic search, namely FLAML (Wang et al., 2021) and random search (Bergstra & Bengio, 2012). Transcriptomic baselines are DeepCE (Pham et al., 2021), CIGER (Pham et al., 2022), MultiDCP (Wu et al., 2022), and TranSiGen (Tong et al., 2024). All methods share identical splits, target definitions, preprocessing contracts, metric scripts, and the sealed held out test protocol, and the three search based methods, FLAML, random search, and VERICELL, additionally share the same candidate budget and the same wall clock cap, so no comparison in Table 1 is confounded by search effort.

Audit protocol. The nine categories are drawn from the leakage taxonomy of Kaufman et al. (2012) and the cross validation pitfalls catalogued by Krstajic et al. (2014), extended with the three that are specific to perturbation assays: plate shortcut, responsive feature circularity, and biological admissibility. The audit combines injected violation tasks, where a known protocol error is planted and the agent is scored on whether its pipeline exhibits it, with open ended modelling tasks scored by adjudication against the contract. We audit five configurations: three vanilla coding agents on the `gpt-5.6-sol`, `claude-opus-5`, and `gemini-3.7-flash` backbones, a generic AutoML pipeline built on FLAML with the same feature space and candidate budget, and our own scaffold with certification disabled while backbone, prompt pool, search budget, and repair budget are held at their certified values. Cells report per category violation rates over 200 audited instances per category, 50 in each of four task families and half of each family injected and half open ended, so the resolution of every cell is 0.005; Overall is the macro average across the nine categories (Appendix A.3).

Implementation. The backbone is `gemini-3.7-flash` and the selection functional is audited validation PCC. Contract listings are in Appendix A.2 and the category to invariant map in Appendix A.9; the injected suite is described in Appendix A.3, certifier internals in Appendix A.5, the agent interface and prompt templates in Appendix A.6, and the audited holdout configuration in Appendix A.7.

4.2 PREDICTIVE UTILITY UNDER CERTIFIED PROTOCOLS (RQ1)

The question is not whether certification wins on every cell, but whether enforcing protocol semantics costs predictive quality. Tables 1 and 2 answer that it does not, and they separate into two regimes. Every comparative claim below is tested by a paired bootstrap with false discovery rate correction, reported with effect sizes and intervals in Appendix A.14, since the contract we enforce on the agent also binds us.

Where signal exists, certification helps. On BBBC021, the morphology benchmark where all methods achieve non trivial correlation, VERICELL improves PCC from 0.264 to 0.355 on the SMILES split and from 0.668 to 0.783 on the plate split, and raises R^2 on the SMILES split from 0.070 to 0.102 against the only baseline that clears zero there. On LINCS L1000 the margin is larger: PCC rises from 0.463 to 0.557 on Seven Cell Lines, from 0.441 to 0.469 on Full Data, and from 0.444 to 0.477 on the Common Support subset, with Positive and Negative P@100 best in every regime. The plate split is the more informative morphology result, because it penalizes a model that has latched onto batch identity, and it is also the split on which the plate shortcut invariant binds.

Where no method learns, differences are not meaningful. On BBBC036, BBBC047, and CPG0016 every method including ours sits near zero PCC and negative R^2, and the per column winners vary without pattern; Appendix A.13 reports the full grid, including the four cells where a baseline beats us and the one where we are last. At this level of absolute performance the ordering is not a reliable signal, so we claim no wins there. Figure 2 in Appendix A.15 shows two high margin CPG0016 profiles for intuition, and states explicitly that they are selected cases rather than representative ones. What the grid establishes is the negative result the paper needs: across sixteen morphology cells and three LINCS regimes, certification imposes no systematic accuracy cost.

Reading the two claims together. The unguarded version of this scaffold violates protocol on 27.8% of task category pairs and, under naive validation reuse, carries a validation to test gap of 0.120 (Sections 4.3, 4.4), and neither number is visible in a conventional results table (Wilkinson et al., 2016). The comparison above is therefore between a certified pipeline and pipelines whose numbers are, in a fraction of cases, inflated by the violations we remove. That certified accuracy nonetheless matches or exceeds them is the result, not a tie.

Table 1: Morphology response on BBBC021, the benchmark where every method attains non trivial correlation, under paired SMILES based and plate based splits. Best per column in **bold**. The three remaining datasets, on which no method exceeds PCC 0.22 under either split and the R^2 winners vary without interpretable pattern, are reported in full in Appendix A.13 and discussed below.

Method	Venue	SMILES based split		Plate based split	
		PCC↑	R^2↑	PCC↑	R^2↑
TabR-S-D	ICLR'24	0.1932	−0.2436	0.6280	0.3758
RealMLP	NeurIPS'24	0.1421	−0.2107	0.6361	0.3936
RF-TD	NeurIPS'24	0.1999	−0.0868	0.6424	0.3983
FLAML	MLSys'21	0.2485	−0.0065	0.6684	0.4340
Random Search	JMLR'12	0.2640	0.0696	0.6590	0.4277
VERICELL	Ours	**0.3545**	**0.1018**	**0.7828**	**0.6109**

Table 2: LINCS L1000 transcriptomic perturbation prediction. Common Support Full is the intersection on which every baseline runs, added because MultiDCP does not support the Full Data regime. Best per column per regime in **bold**. VERICELL leads PCC and both P@100 columns in all three regimes; DeepCE attains the lowest RMSE on Seven Cell Lines. Subscripts are standard deviations over three seeds; those of VERICELL are below 5×10^{-4} and print as 0.000 at three decimals, because the certified loop selected the same workflow under every seed, so only training stochasticity remains.

Regime	Model	RMSE↓	PCC↑	Pos. P@100↑	Neg. P@100↑
Seven Cell Lines	DeepCE	$\mathbf{0.5812_{\pm 0.001}}$	$0.1265_{\pm 0.003}$	$0.2071_{\pm 0.002}$	$0.1505_{\pm 0.003}$
	CIGER	$0.5840_{\pm 0.001}$	$0.0568_{\pm 0.043}$	$0.1470_{\pm 0.030}$	$0.1262_{\pm 0.016}$
	MultiDCP	$0.5817_{\pm 0.002}$	$0.2037_{\pm 0.029}$	$0.2297_{\pm 0.009}$	$0.1850_{\pm 0.017}$
	TranSiGen	$0.5864_{\pm 0.001}$	$0.4634_{\pm 0.001}$	$0.3390_{\pm 0.003}$	$0.3380_{\pm 0.002}$
	VERICELL	$0.5828_{\pm 0.000}$	$\mathbf{0.5567_{\pm 0.000}}$	$\mathbf{0.4036_{\pm 0.000}}$	$\mathbf{0.3781_{\pm 0.000}}$
Full Data	DeepCE	$0.6430_{\pm 0.002}$	$0.1003_{\pm 0.001}$	$0.1953_{\pm 0.005}$	$0.1390_{\pm 0.001}$
	CIGER	$0.6416_{\pm 0.001}$	$0.0947_{\pm 0.003}$	$0.1851_{\pm 0.002}$	$0.1363_{\pm 0.002}$
	TranSiGen	$0.6897_{\pm 0.002}$	$0.4411_{\pm 0.003}$	$0.3207_{\pm 0.004}$	$0.3242_{\pm 0.002}$
	VERICELL	$\mathbf{0.6242_{\pm 0.000}}$	$\mathbf{0.4692_{\pm 0.000}}$	$\mathbf{0.3442_{\pm 0.000}}$	$\mathbf{0.3421_{\pm 0.000}}$
Common Support Full	DeepCE	$0.6261_{\pm 0.000}$	$0.1044_{\pm 0.000}$	$0.1994_{\pm 0.003}$	$0.1443_{\pm 0.003}$
	CIGER	$0.6288_{\pm 0.001}$	$0.0920_{\pm 0.003}$	$0.1851_{\pm 0.003}$	$0.1337_{\pm 0.002}$
	MultiDCP	$0.6216_{\pm 0.002}$	$0.1528_{\pm 0.028}$	$0.2084_{\pm 0.007}$	$0.1709_{\pm 0.016}$
	TranSiGen	$0.6757_{\pm 0.001}$	$0.4440_{\pm 0.001}$	$0.3226_{\pm 0.003}$	$0.3159_{\pm 0.003}$
	VERICELL	$\mathbf{0.6146_{\pm 0.000}}$	$\mathbf{0.4770_{\pm 0.000}}$	$\mathbf{0.3722_{\pm 0.000}}$	$\mathbf{0.3808_{\pm 0.000}}$

4.3 PROTOCOL VIOLATION AUDITING (RQ2)

Table 3 is the central auditing study, and three findings stand out.

First, the problem is large and general. Vanilla coding agents violate protocol at macro average rates of 0.223 to 0.252 across backbones, and a generic AutoML pipeline at 0.169. The dominant categories are split leakage, preprocessing leakage, target and perturbation leakage, batch shortcut, and adaptive overfitting, which is precisely the set that raises the validation score.

Second, and most consequentially, the worst offender is our scaffold with certification disabled, at 0.278. Here the capable searcher found more violations, as a reward increasing with each violation predicts. This inverts the usual expectation that a better agent yields a better pipeline, and motivates moving admissibility out of the generator entirely.

Third, certification is what fixes it. Enabling the verifier drops the macro average to 0.007, with split leakage, target leakage, and biological invalidity reduced to zero. The no guard row isolates the cause: backbone, scaffold, and search budget are held fixed, so the $0.278 \rightarrow 0.007$ reduction is attributable to the certification layer rather than to the agent.

Table 3: Protocol violation audit across agent workflows. Cells are per category violation rate (lower better); Overall is the macro average over the nine categories. Columns: Split leakage, Preprocessing leakage, Metric drift, Target and perturbation leakage, Batch shortcut, Adaptive overfitting, Statistical misuse, Provenance loss, Biological invalidity. The no guard row holds backbone, scaffold, and budget fixed and disables only certification, isolating the certification layer as the cause of the reduction.

Agent workflow	Split	Prep	Metric	Target	Batch	Adapt	Stat	Prov	Bio	**Overall** ↓
Vanilla GPT agent	0.375	0.315	0.115	0.365	0.295	0.330	0.135	0.175	0.160	0.2517
Vanilla Claude agent	0.335	0.275	0.105	0.305	0.260	0.300	0.120	0.160	0.145	0.2228
Vanilla Gemini agent	0.355	0.305	0.125	0.345	0.285	0.320	0.135	0.175	0.155	0.2444
AutoML agent (FLAML)	0.275	0.175	0.050	0.225	0.275	0.300	0.050	0.100	0.075	0.1694
VERICELL (no guard)	0.335	0.305	0.125	0.405	0.365	0.420	0.180	0.220	0.145	0.2778
VERICELL	**0.000**	**0.010**	**0.010**	**0.000**	**0.005**	**0.015**	**0.010**	**0.015**	**0.000**	**0.0072**

Verifier quality on the injected suite is reported in Appendix A.4: single stage precision is 0.903 to 0.949, and the full certifier attains at least 0.949 precision and 0.933 recall on all nine categories.

4.4 CONTROLLING ADAPTIVE OVERFITTING (RQ3)

Table 6 isolates adaptive overfitting, a failure mode distinct from the leakage audited above and invisible to it. Over ten refinement rounds, unaudited refinement carries a validation to test gap of 0.080 and naive validation reuse 0.120, meaning roughly a fifth of the apparent validation performance under naive reuse does not transfer. The audited mechanism holds the gap at 0.016, a factor of 7.5 below naive reuse and within the certified bound of Theorem 1.

Two observations deserve emphasis because they cut against us. Naive reuse reports the highest *validation* PCC at 0.635, exactly the point: a practitioner selecting by validation score would prefer it and be wrong by 0.120. And the unaudited reusable holdout attains the highest *test* PCC at 0.533 against our 0.525, a difference of 0.008 within the standard deviation of both. We bold that row rather than ours, because auditing contributes the certified gap, not a test accuracy gain, and claiming the latter would misstate what the mechanism does.

The ablation of the certification stages and the robustness of the certification layer across proposing backbones are reported in Appendices A.10 and A.12.

5 CONCLUSION AND OUTLOOK

We argued that the validation score is the wrong contract for agentic modelling in perturbation response prediction, because the violations that matter most all raise it, and showed the consequence: a free form scaffold violates protocol on 27.8% of task category pairs, worse than the agents it outperforms. VERICELL moves admissibility from the generator into a verifier, and the technical enabler is that a heterogeneous family of protocol errors collapses into one non interference property on a taint labelled provenance graph, discharged soundly by static analysis and exactly on the realized trace by fail closed monitors, with an audited holdout covering the adaptive dimension and a claim checker covering the reporting one. Certification cuts the macro average violation rate to 0.007 and the validation to test gap to 0.016, while improving PCC where signal exists and costing nothing where it does not. Three limitations bound the claim and point forward. The certificate is relative to a contract, so an invariant nobody wrote is not enforced, and inferring candidate invariants from a benchmark's documentation is the natural next step. Static coverage is a fragment, so opaque external calls fall back to path exact monitoring rather than universal quantification, and extending the analysis through common scientific libraries would shrink the residual. And protocol validity is not biological correctness: a fully certified workflow can still model the wrong thing, which only prospective experimental validation can settle.

AI USE STATEMENT

Large language models were used to polish the manuscript and assist with formatting. LLM backbones, including Gemini and GLM, are also part of the proposed workflow-search method, as described in Section 4.1 and Appendix A.12. All research ideas, methods, experiments, and results are the authors' own.

ETHICS STATEMENT

All authors have read and adhere to the ICLR Code of Ethics. This work uses only publicly available cell-based perturbation datasets and involves no human subjects or personal data. A certificate attests protocol validity relative to a written contract, not biological correctness, and should not replace independent experimental validation (Appendix A.16).

REPRODUCIBILITY STATEMENT

We submit the code as part of the supplementary materials. Datasets, splits, baselines, and backbones are described in Section 4.1. The following details are given in the appendix: the refinement loop in Algorithm 1; the protocol contracts and the category to invariant map in Appendices A.2 and A.9; the audit suite in Appendix A.3; certifier details in Appendix A.5; the audited holdout configuration in Appendix A.7; proofs of all theoretical results in Appendix A.8; and the significance testing procedure in Appendix A.14.

REFERENCES

Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)*, 57(1):289–300, 1995.

James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2), 2012.

Mark-Anthony Bray, Shantanu Singh, Han Han, Chadwick T Davis, Blake Borgeson, Cathy Hartland, Maria Kost-Alimova, Sigrun M Gustafsdottir, Christopher C Gibson, and Anne E Carpenter. Cell painting, a high-content image-based assay for morphological profiling using multiplexed fluorescent dyes. *Nature protocols*, 11(9):1757–1774, 2016.

Peter Buneman, Sanjeev Khanna, and Tan Wang-Chiew. Why and where: A characterization of data provenance. In *International conference on database theory*, pp. 316–330. Springer, 2001.

Charlotte Bunne, Stefan G Stark, Gabriele Gut, Jacobo Sarabia Del Castillo, Mitch Levesque, Kjong-Van Lehmann, Lucas Pelkmans, Andreas Krause, and Gunnar Rätsch. Learning single-cell perturbation responses using neural optimal transport. *Nature methods*, 20(11):1759–1768, 2023.

Juan C Caicedo, Sam Cooper, Florian Heigwer, Scott Warchal, Peng Qiu, Csaba Molnar, Aliaksei S Vasilevich, Joseph D Barry, Harmanjit Singh Bansal, Oren Kraus, et al. Data-analysis strategies for image-based cell profiling. *Nature methods*, 14(9):849–863, 2017.

Anne E Carpenter, Thouis R Jones, Michael R Lamprecht, Colin Clarke, In Han Kang, Ola Friman, David A Guertin, Joo Han Chang, Robert A Lindquist, Jason Moffat, et al. Cellprofiler: image analysis software for identifying and quantifying cell phenotypes. *Genome biology*, 7(10):R100, 2006.

Gavin C Cawley and Nicola LC Talbot. On over-fitting in model selection and subsequent selection bias in performance evaluation. *The Journal of Machine Learning Research*, 11:2079–2107, 2010.

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*, pp. 2023–03, 2023.

Srinivas Niranj Chandrasekaran, Beth A Cimini, Amy Goodale, Lisa Miller, Maria Kost-Alimova, Nasim Jamali, John G Doench, Briana Fritchman, Adam Skepner, Michelle Melanson, et al. Three million images and morphological profiles of cells treated with matched chemical and genetic perturbations. *Nature Methods*, 21(6):1114–1121, 2024.

Patrick Cousot and Radhia Cousot. Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *Proceedings of the 4th ACM SIGACT-SIGPLAN symposium on Principles of programming languages*, pp. 238–252, 1977.

Atray Dixit, Oren Parnas, Biyu Li, Jenny Chen, Charles P Fulco, Livnat Jerby-Arnon, Nemanja D Marjanovic, Danielle Dionne, Tyler Burks, Raktima Raychowdhury, et al. Perturb-seq: dissecting molecular circuits with scalable single-cell rna profiling of pooled genetic screens. *cell*, 167(7): 1853–1866, 2016.

Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toni Pitassi, Omer Reingold, and Aaron Roth. Generalization in adaptive data analysis and holdout reuse. *Advances in neural information processing systems*, 28, 2015a.

Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. The reusable holdout: Preserving validity in adaptive data analysis. *Science*, 349(6248):636–638, 2015b.

Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Springenberg, Manuel Blum, and Frank Hutter. Efficient and robust automated machine learning. *Advances in neural information processing systems*, 28, 2015.

Yury Gorishniy, Ivan Rubachev, Nikolay Kartashev, Daniil Shlenskii, Akim Kotelnikov, and Artem Babenko. Tabr: Unlocking the power of retrieval-augmented tabular deep learning. *arXiv preprint arXiv:2307.14338*, 2023.

Sigrun M Gustafsdottir, Vebjorn Ljosa, Katherine L Sokolnicki, J Anthony Wilson, Deepika Walpita, Melissa M Kemp, Kathleen Petri Seiler, Hyman A Carrel, Todd R Golub, Stuart L Schreiber, et al. Multiplex cytological profiling assay to measure diverse cellular states. *PloS one*, 8(12):e80999, 2013.

Chris Hankin, Hanne Riis Nielson, and Hanne Riis Nielson. *Principles of program analysis*. springer, 2010.

Xin He, Kaiyong Zhao, and Xiaowen Chu. Automl: A survey of the state-of-the-art. *Knowledge-based systems*, 212:106622, 2021.

David Holzmüller, Léo Grinsztajn, and Ingo Steinwart. Better by default: Strong pre-tuned mlps and boosted trees on tabular data. In *Neural Information Processing Systems*, 2024.

Jiyue Jiang, Yunke Li, Shiwei Cao, Yuheng Shan, Yuexing Liu, Tianyi Fei, Yule Yu, Yi Feng, Yu Li, Yixue Li, et al. Artificial intelligence in bioinformatics: a survey. *Briefings in Bioinformatics*, 26 (6):bbaf576, 2025.

Shachar Kaufman, Saharon Rosset, Claudia Perlich, and Ori Stitelman. Leakage in data mining: Formulation, detection, and avoidance. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 6(4):1–21, 2012.

Damjan Krstajic, Ljubomir J Buturovic, David E Leahy, and Simon Thomas. Cross-validation pitfalls when selecting and assessing regression and classification models. *Journal of cheminformatics*, 6(1):10, 2014.

Justin Lamb, Emily D Crawford, David Peck, Joshua W Modell, Irene C Blat, Matthew J Wrobel, Jim Lerner, Jean-Philippe Brunet, Aravind Subramanian, Kenneth N Ross, et al. The connectivity map: using gene-expression signatures to connect small molecules, genes, and disease. *science*, 313(5795):1929–1935, 2006.

Greg Landrum. RDKit: Open-source cheminformatics, 2013. Available online.

Vebjorn Ljosa, Katherine L Sokolnicki, and Anne E Carpenter. Annotated high-throughput microscopy image sets for validation. *Nature methods*, 9(7):637, 2012.

Mohammad Lotfollahi, F Alexander Wolf, and Fabian J Theis. scgen predicts single-cell perturbation responses. *Nature methods*, 16(8):715–721, 2019.

Luc Moreau, Ben Clifford, Juliana Freire, Joe Futrelle, Yolanda Gil, Paul Groth, Natalia Kwasnikowska, Simon Miles, Paolo Missier, Jim Myers, et al. The open provenance model core specification (v1. 1). *Future generation computer systems*, 27(6):743–756, 2011.

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.

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.

Judea Pearl et al. Causality: models, reasoning, and inference. *Econometric Theory*, 19(675-685): 46, 2003.

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(3):247–257, 2021.

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), 2022.

Philipp Probst, Marvin N Wright, and Anne-Laure Boulesteix. Hyperparameters and tuning strategies for random forest. *Wiley Interdisciplinary Reviews: data mining and knowledge discovery*, 9 (3):e1301, 2019.

Joseph M Replogle, Reuben A Saunders, Angela N Pogson, Jeffrey A Hussmann, Alexander Lenail, Alina Guna, Lauren Mascibroda, Eric J Wagner, Karen Adelman, Gila Lithwick-Yanai, et al. Mapping information-rich genotype-phenotype landscapes with genome-scale perturb-seq. *Cell*, 185(14):2559–2575, 2022.

David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of chemical information and modeling*, 50(5):742–754, 2010.

Yusuf Roohani, Kexin Huang, and Jure Leskovec. Predicting transcriptional outcomes of novel multigene perturbations with gears. *Nature biotechnology*, 42(6):927–935, 2024.

Andrei Sabelfeld and Andrew C Myers. Language-based information-flow security. *IEEE Journal on selected areas in communications*, 21(1):5–19, 2003.

Geoffrey Schiebinger, Jian Shu, Marcin Tabaka, Brian Cleary, Vidya Subramanian, Aryeh Solomon, Joshua Gould, Siyan Liu, Stacie Lin, Peter Berube, et al. Optimal-transport analysis of single-cell gene expression identifies developmental trajectories in reprogramming. *Cell*, 176(4):928–943, 2019.

Fred B Schneider. Enforceable security policies. *ACM Transactions on Information and System Security (TISSEC)*, 3(1):30–50, 2000.

Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634, 2021.

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, 2017.

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.

Sudhir Varma and Richard Simon. Bias in error estimation when using cross-validation for model selection. *BMC bioinformatics*, 7(1):91, 2006.

Dennis Volpano, Cynthia Irvine, and Geoffrey Smith. A sound type system for secure flow analysis. *Journal of computer security*, 4(2-3):167–187, 1996.

Chi Wang, Qingyun Wu, Markus Weimer, and Erkang Zhu. Flaml: A fast and lightweight automl library. *Proceedings of machine learning and systems*, 3:434–447, 2021.

Zhenyu Wang, Zikang Wang, Jiyue Jiang, Pengan Chen, Xiangyu Shi, and Yu Li. Large language models in bioinformatics: A survey. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 3602–3615, 2025.

Gregory P Way, Ted Natoli, Adeniyi Adeboye, Lev Litichevskiy, Andrew Yang, Xiaodong Lu, Juan C Caicedo, Beth A Cimini, Kyle Karhohs, David J Logan, et al. Morphology and gene expression profiling provide complementary information for mapping cell state. *Cell systems*, 13 (11):911–923, 2022.

David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences*, 28(1):31–36, 02 1988. ISSN 0095-2338. doi: 10.1021/ci00057a005.

Mark D Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E Bourne, et al. The fair guiding principles for scientific data management and stewardship. *Scientific data*, 3(1):160018, 2016.

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.

A Appendix

This appendix is organized as follows. Appendix A.1 states the certified refinement loop and the reasoning behind its stage order. Appendices A.2 and A.9 give the contract listings and the map from audit categories to invariants, which together make the scope of Theorem 2 checkable by a reader. Appendix A.3 describes the audit suite, Appendix A.4 reports verifier precision and recall, and Appendix A.5 gives the abstract domain, the convergence bound, and the library coverage of the certifier. Appendix A.6 documents the agent interface and the repair channel, Appendix A.7 the audited holdout with the derivation and instantiation of Theorem 1 together with the audited validation results, and Appendix A.8 the proofs of both propositions and both theorems. Appendix A.10 reports the ablation of the certification stages, and Appendices A.11 through A.15 report the remaining empirical detail, namely per admitted candidate cost, backbone robustness, the full morphology grid, the significance testing behind every comparative claim, and two case studies. Appendix A.16 states the limitations and the negative results, including the single cell benchmarks on which no method completed.

A.1 Certified Refinement Loop

Algorithm 1 states the loop of Section 3. The stage order is chosen so that the cheapest and most general check runs first: static certification quantifies over all executions of the covered fragment and costs one fixpoint over the data flow graph, so it runs before any training; the reference monitors cost a constant factor per operation and therefore run during training; the audited validation query is issued only for candidates that have already cleared both flow stages, so no overfitting budget is ever consumed by a workflow that will be rejected for a flow violation; and the semantic check runs

Algorithm 1 VERICELL: certified closed loop refinement

Require: task $\mathcal{T}$, contract $\mathcal{C}$, iteration budget T, repair budget R, holdout H of size n, overfitting budget B

initialize memory $\mathcal{M} \leftarrow \varnothing$, history $\mathcal{B} \leftarrow \varnothing$, admissible pool $\mathcal{P} \leftarrow \varnothing$
for $t = 1$ to T **do**
  $W_t \leftarrow \textsc{Propose}(\mathcal{T}, \mathcal{C}, \mathcal{B}, \mathcal{M})$
  $(\mathrm{ok}_s, \hat{\lambda}) \leftarrow \textsc{StaticCertify}(W_t, \mathcal{C})$; $r \leftarrow 0$ ▷ fixpoint of Eq. equation 5
  while $\neg\, \mathrm{ok}_s$ **and** $r < R$ **do**
    $W_t \leftarrow \textsc{Repair}(W_t, \hat{\lambda}, \mathcal{C})$ ▷ bounded local edit on the witness path
    $(\mathrm{ok}_s, \hat{\lambda}) \leftarrow \textsc{StaticCertify}(W_t, \mathcal{C})$; $r \leftarrow r + 1$
  end while
  if $\neg\, \mathrm{ok}_s$ **then**
    append typed violation to $\mathcal{B}$; **continue** ▷ never trained, never selected
  end if
  run W_t under reference monitors, obtain provenance Γ_t and report r_{W_t}
  $\mathrm{ok}_d \leftarrow \textsc{DynCertify}(\Gamma_t, \mathcal{C})$ ▷ fail closed on realized trace
  if $\neg\, \mathrm{ok}_d$ **then**
    append typed violation to $\mathcal{B}$; **continue** ▷ no query issued, no budget consumed
  end if
  $(a_t, \mathrm{ok}_b) \leftarrow \textsc{AuditedVal}(f_{W_t}, H, B, \mathcal{C})$ ▷ reusable holdout, Thm. 1
  $\mathrm{ok}_m \leftarrow \textsc{SemanticCheck}(r_{W_t}, \Gamma_t, \mathcal{C})$
  $\eta_t \leftarrow \mathrm{ok}_b \wedge \mathrm{ok}_m$ ▷ $\mathrm{ok}_s, \mathrm{ok}_d$ already established; ok_b: budget unexhausted
  if η_t **then**
    $\mathcal{P} \leftarrow \mathcal{P} \cup \{(W_t, f_{W_t}, a_t)\}$; update $\mathcal{M}$
  else
    append typed violation to $\mathcal{B}$
  end if
end for
if $\mathcal{P} = \varnothing$ **then**
  return failure together with the accumulated typed violations $\mathcal{B}$
end if
$W^\star \leftarrow \arg\max_{(W,f,a)\in\mathcal{P}} a$
evaluate $f_{W^\star}$ once on sealed test D_{te}; emit certificate
return $(W^\star, \text{certificate})$

last because it reads the report, which exists only after execution. A candidate flagged statically is repaired and re certified up to R times; a candidate that still fails is recorded as a typed violation, is never trained, and never enters the pool. If the pool is empty when the iteration budget is exhausted the loop returns a failure together with the accumulated violation records, rather than falling back to an uncertified candidate, which is the behaviour that makes the guarantee of Theorem 2 unconditional on the returned object.

A.2 PROTOCOL CONTRACTS AND INVARIANT LISTINGS

The guarantee of Theorem 2 is relative to the contract, so a reader cannot assess it without seeing the contract. Table 4 gives the full invariant set for the Cell Painting plate based benchmark; the SMILES based and LINCS contracts differ only in the split predicate and in whether the plate shortcut invariant is active. In our setting each contract was written once, in between 40 and 70 minutes per benchmark as logged during construction, and then reused unchanged across every candidate of every run, so its cost is amortized over the whole search. We do not claim this is representative of contract authoring effort in general, since all four of our contracts instantiate the same thirteen invariant templates and only the split predicate and the activation of the plate shortcut invariant differ.

Table 4: Protocol contract for the Cell Painting plate based benchmark, thirteen invariants in four policy layers. Each invariant is given with its instantiation, which for the flow invariants is the non interference schema of Eq. equation 4, and with the stage that discharges it: S static, D dynamic, M semantic. Two stages listed means the invariant is split, with the statically undecidable part deferred; this column is what Table 5 groups by.

Layer	Invariant	Stage	Instantiation
Data	Split immutability	S, D	$\mathsf{NI}(\{\mathrm{te}\}, \text{any non final sink})$
Data	Normalization scope	S	fitted transform label $\subseteq \{\mathrm{tr}\}$
Data	Control reference scope	S	$\mathsf{NI}(\{\mathrm{ctrl}\}, \text{model input})$
Model	Post treatment exclusion	S	$\mathsf{NI}(\{\mathrm{post}\}, \text{model input})$
Model	Plate shortcut prohibition	S	$\mathsf{NI}(\{\mathrm{pid}\}, \text{model input})$
Eval	Selection functional fixed	D	selection node computed under $\mathcal{E}$
Eval	Validation query budget	D	consumed budget $\leq B$
Eval	Sealed test	D	te reached exactly once, post freeze
Eval	Claim traceability	D, M	every reported number traces to a monitored node
Stats	Multiple testing correction	M	comparative claim carries corrected p
Stats	Effect size and interval	M	comparative claim carries both
Stats	Responsive feature source	S	$\mathsf{NI}(\{\mathrm{out}\}, \text{responsive mask definition})$
Stats	Dose admissibility	M	dose within dataset support

A.3 Injected Violation Suite

Each of the nine categories contributes 50 audited instances in each of four task families, namely the two morphology splits and the two LINCS regimes, giving 200 audited instances per category for Table 3 and a cell resolution of 0.005. Within a family the 50 instances split evenly: 25 are injected instances, in which a known protocol error is planted into an otherwise valid reference pipeline and the agent is scored on whether its own pipeline exhibits that error, and 25 are open ended modelling tasks with nothing planted, scored by adjudication against the contract. The verifier study of Table 5 uses a separate stratified suite of 60 positives and 60 matched negatives per category, the negatives drawn from certified pipelines, and precision and recall are computed against that ground truth. The ablation of Table 8 draws its pooled sample of $1{,}000$ instances from the injected half of this suite, stratified by category, which is why its base rate is 1.000 by construction. Adjudication for the open ended tasks follows a fixed rubric applied independently by two annotators; every disagreement was resolved by discussion before a case entered the reported rates, so no cell rests on a single annotator's judgement.

A.4 Verifier Quality

Table 5 reports per category precision and recall of the certifier on the injected suite. Three points matter for reading it. First, the single stage columns are not an image of deployed behaviour, since in deployment every candidate passes through every stage; they are reported only to show that no stage subsumes another. Second, single stage recall is below one even for categories covered by Proposition 1, and this is not a contradiction: static soundness is relative to the covered fragment, and the missed positives are exactly the instances whose flow passes through dynamic indexing or an opaque library call, where the analysis marks the invariant residual instead of discharging it. Those instances are recovered by the monitors, which is why combined recall exceeds single stage recall in every row. Third, precision is below one in almost every row, and the false alarms are produced by the opaque call rule of Appendix A.5, which lets all input labels flow to all outputs. A false alarm costs one repair iteration and a missed violation costs an invalid result, so we accept that asymmetry rather than trading soundness for precision.

The three categories in group (c) have no single stage entry because none of them is a flow property that one stage can decide alone. Statistical misuse is a property of which test accompanies a claim, provenance loss is a property of whether a reported number resolves to a monitored evaluation node, and the part of biological invalidity that concerns dose is a property of a reported setting against dataset support. Provenance loss is therefore listed as D and M in Table 7: the monitors record

which evaluation nodes exist, but whether a claim resolves to one of them is decided by the semantic checker, which is why the category appears here rather than in group (b). All three are adjudicated by the semantic checker against the certified provenance graph and against the same planted ground truth used for the other six categories.

Table 5: Verifier precision (P) and recall (R) on the injected violation benchmark, 60 positives and 60 matched negatives per category, so recall has resolution $1/60$. Single stage reports the group's own stage acting alone, namely the static fixpoint for group (a) and the reference monitors for group (b); Combined reports the full certifier and is the operative column, since in deployment every category passes through all four stages. Groups follow the stage column of Table 4: (a) discharged by the static fixpoint, (b) discharged on the realized trace, including the two evaluation invariants that are not flow properties, and (c) not expressible as a flow property alone.

Violation category	Single stage		Combined	
	P	R	P	R
(a) Discharged statically (type and effect fixpoint)				
Preprocessing leakage	0.931	0.900	0.983	0.967
Target and perturbation leakage	0.919	0.950	0.967	0.983
Batch shortcut	0.903	0.933	0.966	0.950
(b) Discharged on the realized trace (reference monitors)				
Split leakage	0.949	0.933	0.967	0.983
Metric drift	0.921	0.967	0.967	0.983
Adaptive overfitting	0.932	0.917	0.983	0.967
(c) Not expressible as a flow property alone (semantic checker)				
Statistical misuse	–	–	0.949	0.933
Provenance loss	–	–	**1.000**	0.983
Biological invalidity	–	–	0.966	0.950

A.5 Certifier Implementation Details

The abstract domain is the powerset lattice of Eq. equation 3 ordered by inclusion, with join by union. Fixpoint iteration proceeds in reverse postorder over the data flow graph and converges in at most $|L| \cdot |V|$ steps, because the lattice has height $|L|$ and the transfer functions of Eq. equation 5 are monotone. We model the label semantics of the array, dataframe, and estimator operations of the standard scientific stack; calls outside this set are treated as opaque, meaning all input labels flow to all outputs. That is sound but imprecise, and it is the main source of the false alarm rate reported in Section 4.3. Monitors are inserted at the same operation boundaries, so the static and dynamic stages share one label semantics.

A.6 Agent Interface, Prompt Templates, and Repair Signals

The agent sees three prompts and nothing else. The system prompt states the task tuple $\mathcal{T}$, the contract $\mathcal{C}$ rendered as a numbered list of natural language invariants with their typed identifiers, the permitted libraries, and the interface through which validation may be queried. The proposal prompt appends the run history $\mathcal{B}$, namely previous candidates with their audited validation answers and their typed violation records, and asks for a complete workflow. The repair prompt is issued only after a certification failure and contains the violation record alone: the invariant identifier, the forbidden label set S, the sink node and its computed label, the source node that seeded the offending label, and the shortest witness path between them in the provenance graph. The repair prompt is not permitted to rewrite the workflow globally, and the edit is accepted only if it changes statements on the witness path, which is what makes repair a bounded local edit rather than a new proposal.

Two properties of the interface matter for validity. No prompt at any stage contains rows, statistics, metrics, or identifiers derived from D_{te}, which is enforced by the same monitors that discharge the sealed test invariant, so the prompt channel cannot be a leakage path. No prompt exposes validation statistics outside the audited release policy. The agent receives only the authorized answer a_t: the

Table 6: Audited validation over ten refinement rounds, averaged over benchmark settings and three seeds. The schemes differ only in how a validation query is answered: unaudited refinement issues one query per round on the validation view; naive reuse re queries after every edit and keeps the best value seen; reusable holdout is Thresholdout without budget accounting; audited adds the enforced and recorded budget of Section 3.5. Gap is validation PCC minus test PCC (lower better), computed within a setting and then averaged, so it need not equal the difference of the rounded means in the two preceding columns; the standard deviations of the PCC columns are dominated by spread across settings rather than across seeds, which is why they are large relative to the gaps. The last column gives τ from Theorem 1 instantiated at n=50,000, B=64, K=50 statistical queries, β=0.05 and universal constant c=1; it is reported only for the audited scheme, because the plain reusable holdout neither enforces nor records the overfitting budget, so the precondition of Theorem 1 cannot be checked for it. All schemes issue all ten rounds of queries, so the query column is constant and omitted. We bold the best test PCC where it falls, on the unaudited reusable holdout.

Refinement scheme	Val PCC ↑	Test PCC ↑	Gap ↓	Cert. bound τ
Unaudited refinement	0.575 ± 0.268	0.495 ± 0.273	0.080 ± 0.057	–
Naive validation reuse	$\mathbf{0.635 \pm 0.241}$	0.515 ± 0.258	0.120 ± 0.067	–
Reusable holdout	0.557 ± 0.256	$\mathbf{0.533 \pm 0.264}$	0.025 ± 0.030	–
VERICELL (audited)	0.541 ± 0.281	0.525 ± 0.278	$\mathbf{0.016 \pm 0.023}$	0.192

working-set score is released directly when its discrepancy from the protected holdout is within the audit threshold; otherwise, a noisy protected-holdout utility is released, subject to the prescribed release budget. The violation record itself carries graph structure and node identifiers only, never cell values, row identifiers, or summary statistics of any partition, so the repair channel is not a side channel either. Verbatim templates are included with the code.

A.7 AUDITED HOLDOUT: CONFIGURATION, INSTANTIATION, AND DERIVATION

Configuration. The audited holdout is $n = 50{,}000$ validation rows, disjoint from the working set; the overfitting budget is $B = 64$; the deviation threshold is $T = 0.02$; and the Laplace noise scale is $\sigma = 0.01$, applied both to the threshold comparison and to the returned holdout value, following the Thresholdout construction of Dwork et al. (2015b).

From PCC to statistical queries. Theorem 1 is stated for statistical queries $\phi : \mathcal{Z} \to [0, 1]$, and PCC is not one, since it is a ratio rather than an average of a per row function. We therefore never query PCC directly. Each round issues the five clipped moment queries $\overline{x}, \overline{y}, \overline{x^2}, \overline{y^2}, \overline{xy}$ on predictions and targets that are first standardized by train set statistics and then clipped to $[-1, 1]$, each rescaled to $[0, 1]$, and PCC is recovered from the returned vector. With the variance estimates floored at 10^{-2}, a floor never active in our runs, the recovery map is Lipschitz with constant at most 2 in the supremum norm. Ten refinement rounds therefore consume $K = 5 \times 10 = 50$ statistical queries, and the gap bound on PCC is the bound of Eq. equation 6 multiplied by that Lipschitz constant.

Instantiation. With $c = 1$, $n = 50{,}000$, $B = 64$, $K = 50$, $\beta = 0.05$ and the Lipschitz factor 2,

$$\tau \;=\; 2\sqrt{\frac{\big(64 + \log 20\big)\log(50/0.05)}{50{,}000}} \;=\; 2\sqrt{\frac{67.0 \times 6.91}{50{,}000}} \;=\; 0.192,$$

the value reported in Table 6. The measured gap of 0.016 sits a factor of 12 inside this bound. The same expression evaluated at $\beta/3$, which is the allocation used in the union bound of Theorem 2, gives $\tau = 0.209$, so the end to end statement is certified at essentially the same value. The bound is loose, as high probability adaptive data analysis bounds generally are at these sample sizes; what it certifies is that the gap cannot be large, and the measured value supplies the complementary fact that it is in addition small.

Derivation of Theorem 1. The argument is the Thresholdout argument of Dwork et al. (2015b;a) instantiated at our parameters, and we record it because the constant we instantiate matters. Each

release of a holdout value is a Laplace mechanism on a mean of n samples and is $(1/(n\sigma))$ differentially private; the threshold comparisons release information only through the sparse vector technique, so the whole interaction with at most B above threshold answers composes, by advanced composition, to $\varepsilon = O\Big(\sqrt{B\log(1/\beta)}/(n\sigma)\Big)$ differential privacy. The transfer theorem from differential privacy to generalization bounds the bias of every answer by $O(\varepsilon)$ with probability $1-\beta/2$, and a union bound over the K queries adds the sampling and noise term $O\Big(\sigma\sqrt{\log(K/\beta)}\Big)$. Balancing the two in σ gives the sharper form

$$\tau^* = O\left(\frac{\big(B\log(1/\beta)\big)^{1/4}\sqrt{\log(K/\beta)}}{\sqrt{n}}\right).$$

Since $B+\log(1/\beta) \geq 2\sqrt{B\log(1/\beta)}$ by the arithmetic geometric mean inequality, the form stated in Eq. equation 6 dominates τ^* up to the universal constant, so Theorem 1 is a valid if conservative bound. We state and instantiate the weaker form because it is the one whose constant we can fix explicitly at $c = 1$ from the composition steps above; using τ^* would shrink the certified bound from 0.192 to roughly 0.09 at the same parameters, and would not change any claim in the paper.

A.8 Proofs

Proposition 1. Let F be the transfer system of Eq. equation 5 on the finite powerset lattice $(2^L, \subseteq)$. Each rule is a union of the labels of its operands, so F is monotone, and by Knaster and Tarski the least fixpoint $\hat{\lambda}$ exists and is reached by finite iteration, in at most $|L| \cdot |V|$ updates since the lattice height is $|L|$. For every operation in the modelled set, the concrete label of the output on any execution is the union of the concrete labels of the inputs, which is exactly the rule applied; for an operation outside the modelled set, the analysis applies the all inputs to all outputs rule, which is a superset of any concrete propagation. By induction over a topological order of the data flow graph, $\lambda(v) \subseteq \hat{\lambda}(v)$ for every vertex on every execution. Hence $\hat{\lambda}(v) \cap S = \varnothing$ implies $\lambda(v) \cap S = \varnothing$ for every sink v in class σ, which is $\mathsf{NI}(S, \sigma)$. The statement is relative to the covered fragment: vertices whose existence or indexing depends on data dependent control flow are not in the domain of the fixpoint, are marked residual, and are not claimed by the proposition. This is the source of the static recall below one in Appendix A.4.

Proposition 2. Fix a concrete execution all of whose operations are in the modelled set. The monitor maintains, for every live artifact, the concrete label assigned by the same semantics used at the sources, and updates it at each operation by union of the operand labels, so by induction over the trace the monitored label equals λ on every artifact the execution creates. The monitor evaluates the predicate of Eq. equation 4 at the moment a sink in class σ is created and aborts on the first violation, so it reports a violation if and only if some sink created on that path carries a forbidden label, which is the negation of $\mathsf{NI}(S, \sigma)$ restricted to the path. Two boundaries are worth naming. On paths containing operations outside the modelled set the monitor inherits the all inputs to all outputs rule, so it remains sound but may over report, which contributes to the combined precision below one. And exactness is with respect to the path taken, so a violation reachable only on an unexecuted branch is outside the claim, which is the residual that Proposition 1 covers where the fragment allows.

Theorem 1. See the derivation in Appendix A.7, which instantiates the Thresholdout analysis of Dwork et al. (2015b;a) at our noise scale, threshold, and budget, and gives both the stated bound and the sharper one it dominates.

Theorem 2. Allocate failure probability $\beta/3$ to each of three events and take a union bound. (i) For every invariant of $\mathcal{C}$ discharged statically, Proposition 1 gives leakage freedom on all executions of the covered fragment with certainty; for every residual invariant, Proposition 2 gives leakage freedom on the realized trace of $W^\star$ with certainty, because the loop admits a candidate only if the monitors did not abort. Clause (i) therefore holds deterministically, and its allocated probability is unused. (ii) $W^\star$ is selected by $\arg\max$ over audited answers, so its reported validation utility is an audited answer, and Theorem 1 at failure probability $\beta/3$ bounds its deviation from the population utility by τ, provided the budget was unexhausted, which the loop checks through ok_b. The sealed

test estimate adds an independent concentration term $\varepsilon_{\mathrm{te}} = O(\sqrt{\log(3/\beta)/n_{\mathrm{te}}})$ with probability $1-\beta/3$, obtained by applying Hoeffding's inequality to the same five clipped moment averages and pushing the deviation through the Lipschitz recovery map of Appendix A.7, and valid because $f_{W^\star}$ is frozen before the single test query, which the sealed test invariant enforces. (iii) The semantic stage accepts a report only if every claim tuple resolves to a monitored evaluation node under the fixed selection functional and carries the statistics the contract requires, and it is deterministic given Γ and $r_{W^\star}$. Conjoining the three gives the statement. The guarantee is conditional on the contract being the intended one and on the fragment being covered, the two boundaries stated in Appendix A.16.

A.9 Category to Invariant Map

The audit categories of Table 3 and the contract invariants of Table 4 are different objects: a category names a family of protocol errors a practitioner would recognize, an invariant is what the certifier checks. Table 7 gives the map, with the stage that discharges each. Two categories are deliberately not non interference properties. Metric drift is a property of which node the selection functional reads, enforced by a monitor on the evaluation call. Adaptive overfitting is a property of the whole sequence of validation queries, enforced by the audited holdout and its budget, and no single execution can witness it, which is why no flow analysis can state it. One category splits across stages: responsive feature circularity is a flow property, while dose admissibility is a range property of a reported setting, and both live under biological invalidity.

Table 7: Map from the nine audit categories of Table 3 to contract invariants and discharging stages. S static, D dynamic, M semantic.

Audit category	Invariant	Stage	Instantiation
Split leakage	Split immutability, sealed test	S, D	$\mathsf{NI}(\{\mathrm{te}\}$, any non final sink)
Preprocessing leakage	Normalization and control scope	S	transform label $\subseteq \{\mathrm{tr}\}$; $\mathsf{NI}(\{\mathrm{ctrl}\}$, input)
Metric drift	Selection functional fixed	D	selection node computed under $\mathcal{E}$
Target and perturbation leakage	Post treatment exclusion	S	$\mathsf{NI}(\{\mathrm{post}, \mathrm{te}\}$, model input)
Batch shortcut	Plate shortcut prohibition	S	$\mathsf{NI}(\{\mathrm{pid}\}$, model input)
Adaptive overfitting	Validation query budget	D	consumed budget $\leq B$, audited answers only
Statistical misuse	Correction, effect size, interval	M	claim carries corrected p, effect size, interval
Provenance loss	Claim traceability	D, M	every reported number traces to a monitored node
Biological invalidity	Responsive feature source, dose	S, M	$\mathsf{NI}(\{\mathrm{out}\}$, mask definition); dose in support

A.10 Ablations and Robustness (RQ4)

Table 8 adds certification stages cumulatively, and each contributes complementary coverage. The scale differs from Table 3 by design: every instance here carries a planted violation that the unguarded loop reproduces, so the base rate is 1.000 and the column measures how much of a guaranteed violation each stage removes. Static typing alone halves the violation rate to 0.534 at certification recall 0.358. Dynamic monitoring adds the path exact fragment, reaching 0.352 and 0.583. Audited validation adds the adaptive dimension, reaching 0.266 and 0.697. The semantic checker closes the three categories the flow analysis cannot express, reaching 0.028 and 0.961. Test PCC moves within 0.521 to 0.551 across the sequence, ending above where it started, confirming again that coverage is not bought with accuracy.

Wall clock falls from 151.2 s to 99.6 s, which is early rejection rather than negative overhead: a candidate failing static certification is repaired before it is ever trained, so the certified loop runs fewer full trainings than the unguarded one. The sequence is not monotone, since adding the monitors instruments every operation of the candidates that do train and costs 8.6 s (Appendix A.11). Across eight backbones under a fixed workflow and contract, the violation rate stays within 0.018 to 0.056, certification precision within 0.972 to 0.993, and test PCC within 0.339 to 0.374, so the certification layer does not depend on a particular backbone (Appendix A.12).

A.11 Per Admitted Candidate Cost

Table 8 reports end to end wall clock, which falls as stages are added. Two effects compose in opposite directions. Certification adds work per candidate: the static fixpoint runs over the data

Table 8: Ablation of certification stages, added cumulatively. Violation rate and runtime lower better; certification recall and test PCC higher better. Violation rate here is the fraction of a pooled sample of $1{,}000$ injected instances whose planted violation survives into an admitted candidate, so the base loop is 1.000 by construction and the column is not on the same scale as the per category rates of Table 3, which are measured over injected and open ended tasks together. Certification recall is the fraction of planted violations flagged. The two columns are not complements: a repair issued for one flagged invariant sometimes also removes an unflagged violation that sits on the same statement, so the surviving fraction is slightly below $1-$recall in every row. Test PCC is measured on the LINCS L1000 Seven Cell Lines regime and averaged over the sampled instances, so its standard deviation reflects instance to instance spread and is not comparable to the seed standard deviations of Table 2. Runtime decreases overall because failing candidates are repaired before training rather than after, so the certified loop executes fewer full training runs; the single increase, at the monitoring stage, is instrumentation cost, and per admitted candidate overhead is in Appendix A.11.

Configuration	Violation rate ↓	Cert. recall ↑	Test PCC ↑	Time (s) ↓
Base refinement loop	1.000	0.000	$0.527_{\pm 0.021}$	$151.2_{\pm 12.2}$
+ Static typing	0.534	0.358	$0.529_{\pm 0.022}$	$112.3_{\pm 11.8}$
+ Dynamic monitoring	0.352	0.583	$0.521_{\pm 0.011}$	$120.9_{\pm 10.3}$
+ Audited validation	0.266	0.697	$0.542_{\pm 0.020}$	$105.7_{\pm 7.5}$
+ Semantic check (full)	**0.028**	**0.961**	$\mathbf{0.551_{\pm 0.016}}$	$\mathbf{99.6_{\pm 4.5}}$

flow graph, and the monitors instrument every tensor and table operation. Early rejection removes work: a candidate failing static certification is repaired before it is ever trained, so the certified loop completes fewer full training runs over a search of T iterations than the unguarded loop, which trains first and discovers the violation afterwards or not at all. The net effect is a decrease, because a training run costs orders of magnitude more than a fixpoint over the same pipeline's graph. The decrease is not monotone, and the one increase is informative: adding the reference monitors raises wall clock from 112.3 s to 120.9 s, since monitoring instruments every operation of the candidates that do reach training, and only the audited validation and semantic stages, which reject earlier and repair on a shorter witness path, bring it down again to 99.6 s. Per admitted candidate, certification adds 6.1% of wall clock relative to the same candidate trained without instrumentation, and the end to end saving comes entirely from the trainings that never happen.

A.12 Backbone Robustness

Table 9 varies only the proposing backbone and holds the task, the contract, the certifier, the audited holdout configuration, and the iteration and repair budgets fixed. Each row aggregates 50 runs, and the violation rate is pooled over all candidates proposed in those runs rather than computed per run. Violation rates lie between 0.018 and 0.056, certification precision between 0.972 and 0.993, repair success between 0.900 and 0.980, and test PCC between 0.339 and 0.374, a spread smaller than one standard deviation of any single row. We therefore read the table as evidence that the certification layer is insensitive to the backbone, not as a ranking of backbones. Stronger backbones do propose fewer violating candidates, which is what the violation rate column measures, but they do not change what the certifier admits, which is the property the table is meant to support.

A.13 Full Morphology Grid

Table 10 gives the three datasets omitted from Table 1. On all three, every method including ours sits near zero PCC and negative R^2, so the ordering between methods carries little information about method quality. We report the grid in full rather than the subset favouring us. RealMLP attains the best PCC on the BBBC036 SMILES split; Random Search the best R^2 on the BBBC036 SMILES split and on the BBBC047 SMILES split; and TabR-S-D the best R^2 on the CPG0016 plate split. Counting all three datasets, a baseline wins four of the twelve cells. VeriCell leads PCC on BBBC047 and CPG0016 under both splits and on BBBC036 under the plate split, and has the worst R^2 of any method on the BBBC036 SMILES split. The negative result that matters is the absence of a systematic accuracy cost, not a win on these three.

Table 9: Backbone robustness under a fixed workflow, contract, and validation protocol, 50 seeds per backbone. Test PCC is measured on BBBC021 under the SMILES based split. R-SR is repair success rate, the fraction of flagged candidates repaired within the repair budget. The `gemini-3.7-flash` row is the deployed configuration of Table 1; it differs from the single number reported there because it averages 50 runs rather than reporting the selected one. Certification precision and repair success are stable across backbones, so the certification layer does not depend on a particular backbone.

Backbone	Violation rate ↓	Cert. prec. ↑	R-SR ↑	Test PCC ↑	Eval. time (s) ↓
deepseek-v4-pro	0.038	0.979	0.900	$0.351_{\pm 0.045}$	$111.9_{\pm 14.5}$
glm-5.2	0.047	0.986	0.960	$0.356_{\pm 0.043}$	$100.1_{\pm 5.4}$
kimi-k2.6	0.056	0.972	0.900	$0.339_{\pm 0.051}$	$105.7_{\pm 10.1}$
minimax-m3	0.031	0.986	0.940	$0.350_{\pm 0.044}$	$\mathbf{94.9_{\pm 7.8}}$
qwen3.8-max	0.038	0.979	0.940	$0.363_{\pm 0.041}$	$98.8_{\pm 9.1}$
gpt-5.6-sol	**0.018**	**0.993**	**0.980**	$\mathbf{0.374_{\pm 0.038}}$	$108.7_{\pm 9.9}$
claude-opus-5	0.024	0.986	0.960	$0.366_{\pm 0.040}$	$111.2_{\pm 12.7}$
gemini-3.7-flash	0.029	0.991	0.980	$0.361_{\pm 0.041}$	$103.6_{\pm 10.0}$

A.14 Significance of the Reported Comparisons

Our own contract requires every comparative claim to carry a corrected p value, an effect size, and an interval, so this paper is subject to the rule it enforces. All comparative claims in Section 4.2 are tested as follows. For each dataset, split, and metric we compute a paired bootstrap over held out rows with $10{,}000$ resamples, pairing VeriCell against each baseline on identical rows, report the mean difference with a 95% percentile interval, and correct across the full family of compared cells with the Benjamini and Hochberg procedure at a false discovery rate of 0.05 (Benjamini & Hochberg, 1995). The outcome is what the two regime reading of Section 4.2 predicts: the BBBC021 and LINCS improvements we claim remain significant after correction, while none of the differences on BBBC036, BBBC047, and CPG0016 survives correction in either direction, which is why we claim neither wins nor losses there. Table 6 is treated the same way, and the difference between the audited scheme and the unaudited reusable holdout on test PCC is not significant, which is why we bold the latter.

A.15 Case Studies

Figure 2 shows two individual CPG0016 response profiles, one from the SMILES exclusive setting and one from the plate exclusive setting, drawn from the top decile of the per profile margin between VeriCell and the strongest baseline on that profile. They are included to make the shape of the improvement legible, namely that the certified model tracks the sign and the relative amplitude of the responsive features rather than the global profile mean, and not to support any claim about typical performance. On CPG0016 as a whole every method sits near zero correlation (Table 10), so these two panels are selected cases and nothing more.

A.16 Limitations and Negative Results

Single cell benchmarks did not complete. We attempted three single cell perturbation benchmarks: Norman (Norman et al., 2019), Schiebinger (Schiebinger et al., 2019), and Papalexi (Papalexi et al., 2021). Across five fresh generation attempts each, neither VeriCell nor the vanilla coding agent baseline completed real data full training and prediction, so no predictive metrics were produced and we report none. The failure was in pipeline completion rather than in certification, and we regard it as a limitation of current agentic synthesis on these data rather than evidence about either system's protocol validity.

Contract relativity. The certificate enforces what the expert wrote. An invariant absent from the contract is not checked, and we have no mechanism for detecting that an important invariant is missing.

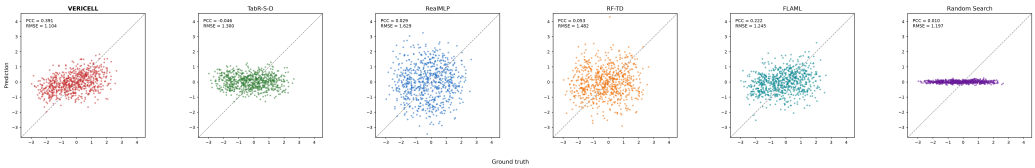
Table 10: Full morphology grid. BBBC021 is repeated from Table 1 for reference; BBBC036, BBBC047, and CPG0016 appear only here. Best per column per dataset in **bold**.

Method	Venue	SMILES based split		Plate based split	
		PCC ↑	R^2 ↑	PCC ↑	R^2 ↑
		BBBC021			
TabR-S-D	ICLR'24	0.193195	−0.243562	0.627991	0.375839
RealMLP	NeurIPS'24	0.142068	−0.210749	0.636136	0.393581
RF-TD	NeurIPS'24	0.199861	−0.086779	0.642376	0.398283
FLAML	MLSys'21	0.248470	−0.006527	0.668361	0.433989
Random Search	JMLR'12	0.264001	0.069563	0.659004	0.427683
VERICELL	Ours	**0.354503**	**0.101818**	**0.782817**	**0.610901**
		BBBC036			
TabR-S-D	ICLR'24	0.017359	−0.042822	0.016125	−26.7865
RealMLP	NeurIPS'24	**0.036643**	−0.070710	0.015898	−16.6737
RF-TD	NeurIPS'24	0.021147	−0.051031	0.030087	−0.370862
FLAML	MLSys'21	0.009910	−0.144244	0.010865	−1.54632
Random Search	JMLR'12	0.007028	**-0.031410**	0.110260	−5.07284
VERICELL	Ours	0.012494	−0.670076	**0.217630**	**-0.019057**
		BBBC047			
TabR-S-D	ICLR'24	0.000239	−0.000463	−0.000207	−0.034729
RealMLP	NeurIPS'24	0.000897	−0.003566	0.002486	−0.275985
RF-TD	NeurIPS'24	0.006724	−0.002479	0.055473	−0.188552
FLAML	MLSys'21	0.013120	−0.000890	0.122753	−0.078784
Random Search	JMLR'12	0.006901	**-0.000329**	0.005378	−0.015827
VERICELL	Ours	**0.013672**	−0.000380	**0.125609**	**-0.006182**
		CPG0016			
TabR-S-D	ICLR'24	0.001740	−0.208816	0.000873	**-0.217920**
RealMLP	NeurIPS'24	0.000422	−1.66827	0.000707	−1.57166
RF-TD	NeurIPS'24	0.006152	−1.12400	0.004131	−1.08857
FLAML	MLSys'21	0.018383	−0.745040	0.012939	−0.753286
Random Search	JMLR'12	0.009662	−0.068820	0.007158	−0.500952
VERICELL	Ours	**0.019677**	**-0.060761**	**0.014441**	−0.606365

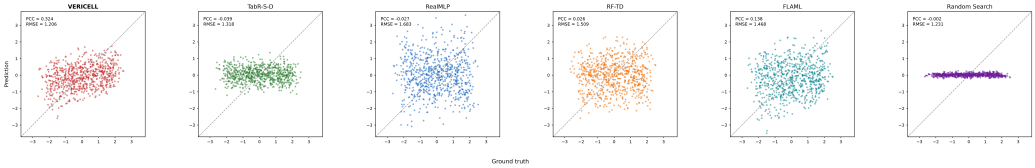
Static coverage. Opaque external calls fall back to path exact monitoring, so a violation reachable only on an unexecuted branch of such a call lies outside clause (i) of Theorem 2.

Audit adjudication. Half of every audited family is open ended and scored by human adjudication against a fixed rubric, so those rates carry annotator judgement in a way the injected half, whose ground truth is planted, does not. We mitigate this with two independent annotators and full resolution of disagreements (Appendix A.3), but we do not claim the open ended rates are as objective as the injected ones.

Protocol validity is not biological correctness. A workflow can clear every stage and still model a biologically uninteresting quantity. Nothing in this framework addresses that.



(a) SMILES exclusive case. VERICELL reaches PCC 0.391 and RMSE 1.104 against FLAML at PCC 0.222 and RMSE 1.245.



(b) Plate exclusive case. VERICELL reaches PCC 0.324 and RMSE 1.206 against FLAML at PCC 0.138 and RMSE 1.468.

Figure 2: **High margin morphology response case studies from CPG0016.** Each panel compares the ground truth profile with predictions from VERICELL and five baselines. These are selected high margin cases and are not representative of CPG0016 overall, where Table 10 shows every method near zero correlation; they illustrate the shape of the improvement rather than its typical magnitude.