From Retrieval to Reasoning: Agentic Mechanism Prediction from Cell Painting Profiles

Jiayuan Chen[1,2], Botao Yu[1]*, Tianyu Liu[3]*, Thai-Hoang Pham[1,2]*, Meng Wu[4], Ping Zhang[1,2]†
[1]Department of Computer Science and Engineering, The Ohio State University
[2]Department of Biomedical Informatics, The Ohio State University
[3]Department of Biostatistics, Yale University
[4]College of Pharmacy, The Ohio State University
{chen.12930,zhang.10631}@osu.edu

Abstract

Cell Painting is a high-content morphological profiling assay widely used for phenotype-based biological inference, with mechanism of action (MOA) prediction as a central application. Existing approaches largely formulate Cell Painting-based inference as representation matching, assigning predictions from nearby reference perturbations in morphological feature space. However, retrieved neighbors are often noisy and partially misleading evidence due to batch effects, non-specific cytotoxicity, phenotypic convergence, and source-dependent variability. We reformulate Cell Painting-based MOA prediction as a calibrated evidence reasoning problem, where retrieved neighbors are treated as uncertain observations that must be evaluated, compared, and sometimes rejected before supporting a mechanistic conclusion. We propose PhenoAIR, a reliability-aware multi-agent framework that maintains a candidate-centric evidence memory and performs controller-guided refinement over phenotype- and mechanism-side evidence. PhenoAIR uses offline reference-set calibration to weight evidence by source reliability, phenotype stability, and mechanism-level confusion. We evaluate PhenoAIR on a benchmark[3] constructed from JUMP Cell Painting profiles and annotations, covering controlled, realistic, and discovery-oriented open-world MOA prediction settings. PhenoAIR[4] outperforms representation-matching and LLM-based baselines across all settings.

1 Introduction

Cell Painting is a high-content screening assay that captures drug- or gene-induced changes in cellular morphology [3, 8], providing a scalable phenotypic readout for downstream biological inference tasks ranging from toxicity assessment [13] and lead prioritization to functional annotation of genetic perturbations [7, 17]. Across these applications, a common computational strategy is to encode perturbations into a morphological representation space and infer biological relationships from the geometry of that space, through clustering, retrieval, label transfer, or prototype matching. In this work, we focus on mechanism of action (MOA) prediction [4, 21], where existing methods commonly assign labels from the most similar reference perturbations, e.g., via top-K retrieval or distance-weighted voting. This assumes that proximity in phenotype space provides a reliable basis for biological inference. However, similarity alone does not indicate whether the retrieved evidence is biologically informative, reliable, or mechanistically consistent [8, 24].

*Equal contribution authors.
†Corresponding author.
[3]https://huggingface.co/datasets/Jerrychen229/PhenoAIR_MOA
[4]https://github.com/The-Real-JerryChen/PhenoAIR_NeurIPS26

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

In Cell Painting, morphological similarity is an indirect signal of shared mechanism rather than a mechanistic measurement itself. Similar profiles may reflect a common biological cause, but they may also arise from confounding experimental factors or convergent downstream cellular states. Thus, the core challenge is not retrieving morphologically similar perturbations, but deciding which similarities constitute reliable mechanistic evidence. This uncertainty creates three requirements that fixed retrieval pipelines do not satisfy. Retrieval quality is instance-dependent: for some queries, relevant mechanisms appear among top-ranked neighbors, whereas for others they are buried deep in the retrieval list, making any fixed retrieval depth inadequate [1]. Morphological profiles also show source-dependent variability, the same compound profiled across laboratories may yield inconsistent similarity patterns, and cross-source agreement or disagreement is itself an important reliability signal that single-profile retrieval cannot exploit [6, 27]. Finally, the MOA label space is mechanistically structured: polypharmacology and systematically confusable MOA pairs often require pharmacological context to distinguish mechanisms that appear similar in morphology. We therefore argue that **Cell Painting-based MOA prediction is better formulated not as a matching problem, but as an evidence-based reasoning problem.** Under this view, retrieved neighbors are uncertain observations that must be evaluated, compared, calibrated, and sometimes rejected before they can support a mechanistic conclusion. Importantly, our goal is not to replace Cell Painting with non-phenotypic annotations, but to make phenotypic evidence an explicit object of reasoning: final mechanisms should be accepted only when they are supported by reliable Cell Painting-derived evidence.

Recent LLM and agentic systems offer tool use [26], retrieval-augmented reasoning, and multi-agent collaboration [12], but they are not directly designed for this setting. Biomedical agents [19, 16] typically use tools for knowledge-centric queries or workflow execution, and generic RAG treats retrieved items as context to summarize. Debate-style multi-agent systems add multiple perspectives, but often rely on free-form consensus rather than persistent candidate-level evidence states. Cell Painting-centered MOA prediction instead requires reasoning over retrieved neighbors whose reliability is uncertain, motivating a task-specific agentic formulation that calibrates evidence, assigns it to explicit hypotheses, and refines decisions through controlled evidence-seeking actions.

To instantiate this evidence-based formulation, we propose PhenoAIR (**Pheno**typic **A**gentic **I**nference with **R**eliability-aware reasoning), a multi-agent framework for calibrated evidence reasoning over Cell Painting profiles. PhenoAIR does not treat retrieved neighbors as predictions. Instead, it treats them as uncertain evidence for candidate mechanisms and maintains a persistent candidate memory that records which hypotheses are supported, contradicted, suppressed, or still require validation across reasoning rounds. The inference process is organized around this memory. A phenotype-side reasoner interprets calibrated Cell Painting retrieval evidence, while a mechanism-side reasoner independently evaluates the structural and pharmacological plausibility of candidate mechanisms. Their outputs update the shared candidate memory rather than directly determining the final answer. An offline reliability layer further calibrates the evidence by summarizing global reference-set reliability, phenotype stability, and mechanism-level confusion patterns. A rule-based controller then uses the memory state to decide whether to stop, expand retrieval, inspect specific candidates, validate candidate groups, or request additional mechanism-side reasoning. Finally, an arbiter produces the prediction from the full evidence trajectory under explicit eligibility constraints. For open-set MOA inference, where the system must distinguish weak support for known mechanisms from evidence of a genuinely novel mechanism, we extend the controller and arbiter with an offline self-evolving procedure that refines decision rules from observed reasoning trajectories.

To evaluate our approach, we construct a benchmark from the public JUMP Cell Painting Consortium [6] and Drug Repurposing Hub [10], comprising three complementary settings. MoA-Verified provides a controlled setting with high-confidence cross-source profiles. MoA-Extended reflects realistic deployment conditions with heterogeneous and noisy data. MoA-Novel targets a discovery-oriented setting relevant to first-in-class drug discovery, where a query compound may act through a mechanism not represented in the reference set and methods must recognize when known MOA labels are insufficiently supported. Together, these settings evaluate not only classification accuracy, but also robustness to noisy retrieval evidence and the ability to identify unsupported known mechanisms. Across these settings, PhenoAIR outperforms representation-matching baselines and LLM-based agent/workflow baselines.

Our contributions are as follows:

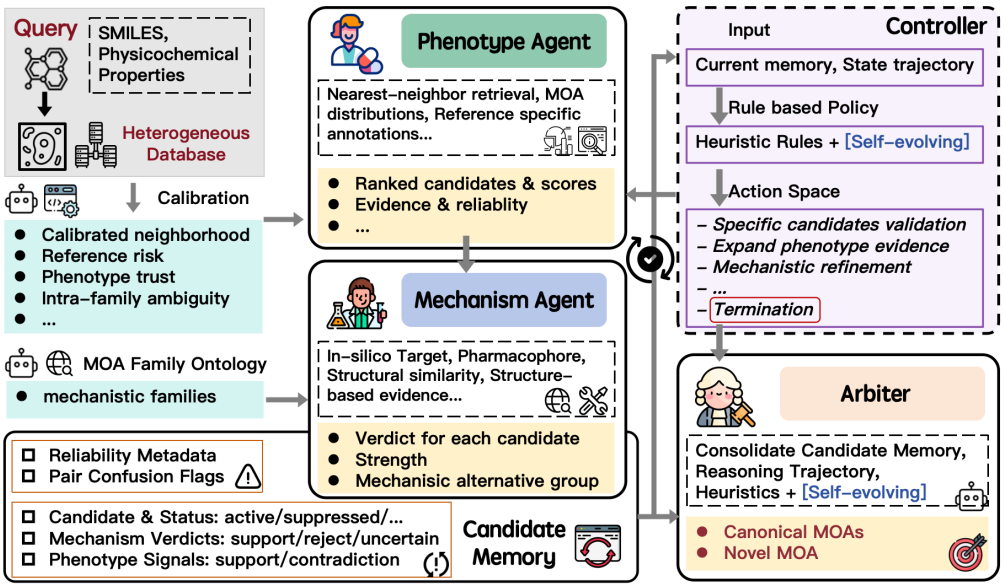


Figure 1: Architecture of PhenoAIR. The system consists of three stages: Externalized reliability calibration (left) precomputes a reliability-annotated phenotype neighborhood and a mechanistic family ontology from the multi-source reference database. Candidate-centric evidence reasoning (center) maintains a shared candidate memory updated by Phenotype Agent (calibrated retrieval) and a Mechanism Agent (structural and pharmacological verdicts). Controller-guided refinement (right) iteratively coordinates the two agents through a rule-based controller, terminating with an Arbiter that produces a canonical or novel MOA. Self-evolving extensions (in blue) handle open-set inference.

- We reformulate Cell Painting-based MOA prediction from representation matching to calibrated evidence reasoning, where retrieved phenotypic neighbors are treated as uncertain observations rather than direct answers.
- We propose PhenoAIR, a reliability-aware multi-agent framework that replaces free-form agent consensus with candidate-centric evidence memory, externalized reliability calibration, and controller-guided refinement over auditable evidence states.
- We construct a curated three-setting benchmark that evaluates MOA prediction under controlled, realistic, and discovery-oriented open-world conditions, and show that PhenoAIR improves over representation-matching and LLM-based baselines.

2 Method

2.1 Problem Formulation

We consider the task of predicting the mechanism of action (MOA) of a query compound from its Cell Painting morphological profile. Let $\mathcal{R} = \{(z_i, y_i)\}_{i=1}^{N}$ denote a reference set, where $z_i \in \mathbb{R}^d$ is the aggregated morphological profile of reference compound i, and $y_i \subseteq \mathcal{M}$ is its set of MOA labels drawn from a known label space $\mathcal{M}$. When a reference compound is profiled across multiple experimental sources, we aggregate its source-specific profiles into a consensus profile z_i for standard retrieval, while retaining the individual source profiles for cross-source reliability analysis. Given a query compound with Cell Painting profile z_q, a standard retrieval-based formulation obtains a set of nearest reference compounds $\mathcal{N}_K(q) \subseteq \mathcal{R}$ under a morphological distance function, and predicts $\hat{y}_q$ by aggregating the MOA labels of the retrieved neighbors, e.g., through top-K voting or distance-weighted aggregation.

We consider two evaluation settings. In the **closed-set** setting, the true MOA is assumed to be represented in $\mathcal{M}$, and the goal is to predict $\hat{y}_q \subseteq \mathcal{M}$. In the **open-set** setting, the query may correspond to a mechanism not represented by the known reference label space. The system must therefore either predict a supported known MOA $\hat{y}_q \subseteq \mathcal{M}$, or output the special label `novel` to indicate that the available evidence does not adequately support any candidate in $\mathcal{M}$.

2.2 Method Overview

Recent LLM-based inference systems often combine retrieval, tools, and multi-agent reasoning by treating external evidence as context for generation, discussion, or reflection [12, 26, 14]. In Cell Painting-based MOA prediction, however, retrieved evidence is not a set of reliable facts: nearest phenotypic neighbors may reflect batch effects, generic morphology, weak reference anchors, or mechanistically ambiguous profiles. PhenoAIR therefore treats retrieved neighbors as noisy observations whose reliability must be calibrated and whose candidate implications must be tracked.

PhenoAIR replaces label aggregation over nearest neighbors with reliable evidence integration over candidate MOA hypotheses. Given a query profile, the system maintains a candidate memory in which each candidate records phenotype support, phenotype contradiction, mechanism-side verdicts, reliability context, and eligibility state. The framework has three components. First, **externalized reliability calibration** constructs offline reliability resources and converts raw retrieval into calibrated phenotype evidence. Second, **candidate-centric evidence reasoning** uses role-separated phenotype and mechanism agents to update shared candidate states rather than conducting free-form debate. Third, **controller-guided refinement** adaptively selects evidence-collection actions and terminates when the memory state is sufficiently resolved. A final arbiter reasons over the completed memory and trajectory, followed by deterministic eligibility checks that return either a supported canonical MOA or `novel`. PhenoAIR differs from generic retrieval-augmented generation by treating retrieved neighbors as uncertain observations rather than static context for answer generation. It also differs from debate-style multi-agent systems: rather than relying on unconstrained agent consensus, retrieval, mechanism reasoning, and arbitration are connected through persistent candidate states and deterministic validity checks. Details are provided in Appendices C.1-C.3 and Algorithm 1.

2.3 Externalized Reliability Calibration

Raw phenotype similarity does not indicate whether a retrieved neighbor is biologically informative. PhenoAIR therefore separates evidence-quality assessment from query-time reasoning through an offline reliability layer. Before inference, it constructs a reference reliability memory $\mathcal{B}$ from the multi-source reference set. An LLM-assisted code-execution pipeline proposes analyses, computes reference-set measurements, and converts them into structured annotations describing cross-source consistency, reference-anchor replicability, intra-MOA stability, pairwise MOA separation, generic morphology risk, and recurring MOA-confusion patterns. This stage does not observe query outcomes and does not produce query-specific predictions; its outputs are cached before inference.

For a query q, the reliability layer produces a calibrated neighborhood $\tilde{\mathcal{N}}(q)$ and a query-level reliability context r_q. The calibrated neighborhood serves as ranked phenotype evidence, while r_q summarizes evidence concentration, source agreement, low-trust anchors, and potential MOA confusions as review-only context. The calibration layer does not introduce new mechanisms or modify the label space.

PhenoAIR also constructs an offline mechanistic family ontology over $\mathcal{M}$, grouping canonical MOA labels into coarse biological families using external biomedical evidence and label definitions. This ontology supports group-level reasoning when evidence localizes to a family but is insufficient to select a fine-grained canonical label. Final closed-set predictions remain restricted to labels in $\mathcal{M}$, with `novel` reserved for open-set cases.

2.4 Candidate-Centric Evidence Reasoning

Candidate memory. Inference maintains $\mathcal{H}_q = \{(c, s_c) : c \in \mathcal{C}_q\}$, where c is a candidate MOA and s_c is its evidence state. Canonical candidates belong to $\mathcal{M}$; optional family entries are used only for intermediate validation. Each state stores phenotype support and contradiction, mechanism verdicts (*support*, *reject*, or *uncertain*), candidate status, and reliability metadata. A candidate becomes final-eligible only through accumulated evidence, not merely because it appears in retrieval or is proposed by one agent.

Phenotype agent. The phenotype agent interprets Cell Painting evidence only. Given $\tilde{\mathcal{N}}(q)$, r_q, and the current memory, it summarizes local neighbors, broader MOA distributions, and retrieved reference-anchor annotations into candidate-level phenotype evidence. Reference-anchor annotations describe retrieved references, not the query compound, and are used only to assess anchor trustwor-

thiness. The phenotype agent does not use query-side structure, target predictions, or pharmacophore information.

Mechanism agent. The mechanism agent independently evaluates memory candidates using query-side chemical structure, in-silico target predictions, pharmacophore matches, and structural similarity to known compounds. It may propose a canonical candidate or mechanistic family when structure-side evidence supports an alternative, but such proposals are inserted as requiring phenotype grounding or group-level validation rather than becoming directly final-eligible.

Memory-mediated communication. The agents do not communicate through free-form dialogue. Each reads the current candidate memory and returns structured evidence updates. This makes cross-agent influence explicit: mechanism evidence can challenge a retrieved phenotype hypothesis, phenotype evidence can validate or reject a mechanism-side proposal, and unresolved conflicts remain visible to the controller.

2.5 Controller-Guided Inference Refinement

Candidate memory records evidence state, but does not decide which evidence should be collected next. PhenoAIR uses a rule-based controller to select auditable refinement actions based on the current memory, reliability context, and trajectory history. The controller does not predict MOA labels; it only chooses the next evidence view. The process runs for query-dependent iterations $t = 0, \ldots, T$, where T is determined by the controller's STOP action.

Iterative refinement and controller policy. At iteration t, after phenotype- and mechanism-side updates, the controller selects

$$a^{(t)} = \pi\big(\mathcal{H}_q^{(t)}, r_q, \tau_q^{(t)}\big), \quad a^{(t)} \in \mathcal{A},$$

where $\tau_q^{(t)}$ records past states, agent outputs, and controller actions. The action space contains high-level operations such as expanding phenotype evidence, validating candidates or candidate groups, requesting mechanism-side family reasoning, inspecting risky reference anchors, and terminating inference. The policy uses interpretable signals such as phenotype reliability, phenotype-mechanism agreement, candidate support state, and whether previous actions changed the memory. This avoids both fixed-depth retrieval and free-form LLM planning.

Arbiter and final gate. When the controller stops, the arbiter produces a candidate-level decision:

$$y_q^{\mathrm{arb}} = \mathrm{Arbiter}\big(\mathcal{H}_q^{(T)}, \tau_q^{(T)}\big).$$

The arbiter is LLM-based but cannot call evidence-gathering tools. Its output is passed through deterministic eligibility checks:

$$\hat{y}_q = \mathrm{Gate}\big(y_q^{\mathrm{arb}}, \mathcal{H}_q^{(T)}\big).$$

The gate excludes suppressed candidates, requires adequate support, prevents selection from unsupported calibration metadata alone, and enforces canonical closed-set labels. If no canonical candidate is adequately supported, the output is `novel`.

Self-evolving refinement for open-set inference. For open-set inference, we use an offline self-evolving procedure over a held-out development set to refine the controller rule registry and the mutable arbiter heuristic section. The procedure does not update LLM parameters, feature extractors, tools, evidence sources, or the label space. Given development trajectories, an offline LLM clusters recurring failure patterns from compressed phenotype, mechanism, controller, memory, and arbiter traces, then proposes schema-constrained controller rules, arbiter heuristic revisions, or retention decisions. Proposed rules must use whitelisted runtime signals and allowed actions; invalid artifacts are blocked. All selected artifacts are fixed before final test evaluation.

3 Experiment

3.1 Benchmark Construction

We construct a benchmark for Cell Painting-based MOA prediction that under three progressively more challenging regimes: high-confidence closed-set inference, realistic noisy-reference inference, and discovery-oriented open-world inference. The benchmark is built from public Cell Painting

Table 1: Benchmark statistics. Our benchmark comprises three settings constructed from the JUMP Cell Painting Consortium and Drug Repurposing Hub.

Dataset	# Reference	# Test	# MOA classes	Avg Sources/Ref.	Avg Labels
MoA-Novel	63	100	26 (Known)	4.05	1.00
MoA-Verified	63	200	26	4.05	1.19
MoA-Extended	533	600	137	2.64	1.13

profiles and curated drug MOA annotations, allowing us to test not only predictive accuracy, but also robustness to reference quality, source heterogeneity, and incomplete mechanism coverage.

3.1.1 Data Curation

We use the cpg0016-jump dataset from the Cell Painting Gallery [6], which provides raw Cell Painting images of cells under chemical perturbations. Although CellProfiler well-level features are available, we use image-derived representations from pretrained encoders as the primary morphological profiles, and include feature-based baselines in our comparisons. To capture realistic experimental variability while keeping the benchmark computationally tractable, we select five sources covering approximately 115k unique compounds before MOA annotation filtering. These sources include both overlapping compounds profiled across laboratories and source-specific compounds, enabling evaluation under cross-source batch effects and incomplete source overlap. One source uses a lower perturbation concentration (0.625 μM) than the standard 10 μM used in the other sources, further introducing dosage-dependent phenotypic variation.

Ground-truth MOA annotations are obtained from the Broad Institute Drug Repurposing Hub [10]. We match compounds to the Hub and retain those with valid annotations. Since compounds may exhibit polypharmacology, we use curated multi-label MOA sets for evaluation. To ensure label quality, we remove redundant or hierarchically dependent annotations so that remaining labels correspond to distinct mechanistic hypotheses (details in Appendix D.1). In addition, each compound is assigned a representative MOA label for compatibility with standard retrieval-based baselines, while all primary evaluations also include the full multi-label annotation set.

3.1.2 Evaluation Settings

We define three complementary evaluation settings. **MoA-Verified** prioritizes reference reliability: the reference set contains compounds profiled across multiple sources, providing higher-confidence and more reproducible morphological evidence. This setting evaluates performance under relatively clean retrieval conditions. **MoA-Extended** prioritizes MOA coverage: it includes all available annotated compounds regardless of source count, expanding the label space while introducing heterogeneous reference quality, including single-source profiles that are more susceptible to batch effects. This setting evaluates robustness under realistic noisy retrieval conditions. In both settings, we retain MOA classes with at least two distinct reference compounds to reduce source-specific artifacts and support meaningful similarity-based evaluation. **MoA-Novel** evaluates discovery-oriented open-world inference. In this setting, a subset of test compounds (50%) is associated with mechanisms not represented in the reference label space. Models must therefore predict known MOAs for in-distribution compounds while recognizing cases where available reference MOAs are insufficiently supported, instead of forcing a nearest-neighbor label. The test set contains a balanced mixture of known-MOA and novel-MOA cases. Dataset statistics are reported in Table 1.

We use different metrics for closed-set and open-world settings. For MoA-Verified and MoA-Extended, we report **Top-1 accuracy** with respect to the representative MOA label of each compound, and **Any-match accuracy**, where a prediction is correct if it matches any label in the curated multi-label ground truth. For MoA-Novel, we report **overall accuracy**, treating `novel` as an additional output class, and **Novel detection F1**, which evaluates known-versus-novel discrimination as a binary classification task. For representation-based baselines (e.g., kNN), predictions are matched to ground-truth labels by exact string comparison. For LLM-based methods that will produce open-form textual outputs, we first apply string-based matching to map predictions to canonical MOA labels; unmatched cases are resolved using an LLM-as-judge, which we verify to be highly consistent with human expert annotations (details in Appendix D.3).

3.1.3 Baselines

We consider baselines spanning representation-based retrieval and LLM-based inference paradigms. **Representation-Based Retrieval.** We evaluate standard nearest-neighbor retrieval methods in morphological and molecular representation spaces. For Cell Painting, we use (i) CellProfiler features provided by the Broad Institute, and (ii) deep learning representations from foundation models including CellCLIP [22] . Predictions are obtained by distance-weighted voting over the top-K neighbors, following established practice in phenotypic profiling [4]. In addition, we include a structure-based retrieval baseline using molecular fingerprints to identify structurally similar compounds and transfer their MOA labels.

LLM-based inference. We evaluate LLM-based baselines that use chemical and phenotypic evidence. The **structure-only** baseline uses structure-driven information and external mechanistic signals without access to phenotypic profiles. It serves as a diagnostic control for non-phenotypic MOA signal, rather than as a Cell Painting analysis method. We further include a fixed **workflow** baseline that follows a predefined phenotype-then-mechanism reasoning sequence. The LLM first analyzes Cell Painting retrieval evidence and then incorporates structure-driven information to make a final prediction. Unlike PhenoAIR, this workflow does not maintain a persistent candidate memory, adaptively expand or validate evidence, or revise its reasoning through controller-guided refinement. Finally, we include a **single-agent** variant that has access to the same evidence tools as PhenoAIR but performs inference within a single agent. This baseline isolates the effect of reasoning decomposition, shared candidate memory, and controller-guided refinement.

3.1.4 Implementation Details

For representation-based retrieval, we use well-level Cell Painting features with post-processing to mitigate batch effects, followed by aggregation into compound-level representations across sources. We evaluate multiple aggregation strategies and report the best-performing variant. The kNN baselines use cosine distance for Cell Painting features and Tanimoto distance on Morgan fingerprints (radius 2, 2048 bits) for the ECFP baseline, and predict by inverse-distance-weighted voting over the $K = 5$ nearest reference compounds. For LLM-based methods, we evaluate both GPT-5.1 and Claude 4 Sonnet. We choose them because of their stable tool-use capability, reasoning ability, and biological knowledge, while reducing reliance on memorized associations between chemical structures and MOA labels. The goal is to evaluate inference from external evidence rather than recall of known structure-function mappings. The structure-only baseline has access to structure-driven and mechanism-side tools without phenotype information, while the fixed-workflow baseline uses the same evidence tools as PhenoAIR but follows a predefined inference pipeline. To reduce information leakage, we curate external knowledge sources and exclude trivial tool outputs that directly reveal ground-truth labels, such as exact target-gene matches with confidence 1.0. All LLM-based methods use consistent tool interfaces and comparable prompt settings. For stochastic LLM-based experiments, we report results averaged over five independent runs. Additional details on prompts, tool definitions, post-processing steps, computational cost, runtime, and API usage are provided in the Appendix B.1.3.

3.2 Main Results

Table 2 reports results on the three evaluation settings. LLM-based methods are averaged over five independent runs; deterministic retrieval baselines are reported without variance. Overall, representation-based retrieval methods perform substantially worse than LLM-based inference methods. Among kNN baselines, learned Cell Painting representations (CellCLIP) improve over CellProfiler features, while ECFP-based chemical retrieval is strongest within this group. However, all retrieval-only methods remain limited, indicating that neither morphological nor structural nearest-neighbor transfer is sufficient for reliable MOA prediction.

LLM-based baselines show that simply adding more evidence is not enough. Structure-only inference provides a strong baseline, reflecting the predictive value of chemical and mechanistic information. However, the fixed workflow that combines phenotype retrieval with structure-side evidence does not consistently improve over structure-only inference, especially on MoA-Extended, where retrieval evidence is noisier. This suggests that uncalibrated phenotype retrieval can introduce misleading evidence when incorporated through a fixed pipeline. The single-agent baseline also underperforms PhenoAIR, showing that the gains come not merely from tool access, but from role-separated multi-agent reasoning coupled with structured candidate memory and controller-guided refinement.

Table 2: Main results on three datasets. We report accuracy (Acc. %), hit rate (Any-match), and F1 measured against ground-truth MOA labels. All LLM-based methods operate on the same CellCLIP features. **Best** results are in bold; second-best are underlined. Our method is highlighted in blue.

	Dataset	MoA-Verified		MoA-Extended		MoA-Novel	
	Metrics	Acc	Any-match	Acc	Any-match	Acc	F1 (Novel)
kNN	Random	3.5 ± 1.4	12.3 ± 2.0	1.2 ± 0.3	3.2 ± 0.5	N/A	
	CellProfiler	9.5 ± 0.0	23.5 ± 0.0	6.8 ± 0.0	12.7 ± 0.0		
	CellCLIP	14.0 ± 0.0	26.5 ± 0.0	10.2 ± 0.0	16.0 ± 0.0		
	ECFP	23.0 ± 0.0	41.0 ± 0.0	30.0 ± 0.0	42.3 ± 0.0		
GPT 5.1	Structure-only	43.5 ± 1.4	59.0 ± 1.1	39.8 ± 0.3	52.9 ± 0.1	44.2 ± 3.4	53.3 ± 4.0
	Workflow	50.0 ± 1.2	56.0 ± 1.0	31.7 ± 0.5	35.4 ± 0.3	58.0 ± 0.3	61.3 ± 0.8
	Single Agent	39.1 ± 1.4	59.5 ± 1.1	30.9 ± 0.5	46.2 ± 0.5	48.8 ± 2.4	67.3 ± 1.1
	PhenoAIR	$\mathbf{62.6 \pm 0.7}$	$\mathbf{74.6 \pm 0.6}$	$\underline{45.6 \pm 0.2}$	$\underline{56.2 \pm 0.3}$	$\mathbf{79.4 \pm 2.2}$	$\mathbf{74.7 \pm 2.9}$
Claude 4	Structure-only	47.3 ± 0.8	64.5 ± 1.0	40.8 ± 0.3	55.7 ± 0.2	43.2 ± 1.1	50.6 ± 1.0
	Workflow	52.4 ± 0.9	60.0 ± 0.7	37.7 ± 0.6	42.2 ± 0.3	63.6 ± 1.5	70.2 ± 1.2
	Single Agent	40.5 ± 1.4	58.3 ± 0.9	32.1 ± 0.7	45.5 ± 0.5	52.0 ± 1.2	66.9 ± 1.1
	PhenoAIR	$\underline{61.5 \pm 0.5}$	$\underline{74.2 \pm 0.7}$	$\mathbf{45.8 \pm 0.3}$	$\mathbf{59.9 \pm 0.1}$	$\underline{73.6 \pm 3.1}$	$\underline{74.1 \pm 2.0}$

PhenoAIR achieves the best performance across all three settings and both LLM backbones. On MoA-Verified and MoA-Extended, it improves over both representation-matching baselines and LLM-based baselines, demonstrating the benefit of calibrated evidence reasoning for closed-set MOA prediction. The advantage is most pronounced in MoA-Novel: while retrieval baselines are not applicable because they must assign each query to an existing reference label, PhenoAIR substantially improves both overall accuracy and novel-detection F1. These results indicate that candidate-centric memory and reliability-aware arbitration help the model recognize when known MOA labels are insufficiently supported, rather than forcing a prediction from noisy or incomplete evidence.

3.3 Ablation Studies

3.3.1 Component Ablation

Table 3 reports an ablation that incrementally adds PhenoAIR components on top of a base LLM. Direct prompting with only SMILES information performs poorly on both benchmarks, indicating that parametric knowledge alone is insufficient for fine-grained MOA prediction. Adding the base PhenoAIR multi-agent workflow, without candidate memory or reliability calibration, substantially improves performance, showing that structured phenotype- and mechanism-side evidence reasoning is already much stronger than direct prompting.

Table 3: Effect of core components on two datasets. Variants are constructed by incrementally adding each PhenoAIR component to a base LLM, isolating the contribution of reliability calibration, candidate-centric memory with multi-agent reasoning. All variants use the same base LLM and CellCLIP features. Full results are in Appendix.

	Dataset	Verified	Extended
	Method	Acc	Acc
GPT 5.1	Base LLM	16.4	18.7
	PhenoAIR-Base	55.6	43.9
	+ Memo.	60.1	44.5
	+ Memo. & Cal.	62.6	45.6
Claude 4	Base LLM	26.7	23.3
	PhenoAIR-Base	55.0	42.3
	+ Memo.	59.3	44.2
	+ Memo. & Cal.	61.5	45.8

Candidate-centric memory provides consistent additional gains across both LLM backbones and datasets. This suggests that explicitly tracking candidate hypotheses and their supporting or contradicting evidence across rounds is beneficial beyond within-turn agent context. Adding reliability calibration further improves performance, especially on MoA-Verified, where the calibrated reference evidence is more stable and reliable. The smaller gain on MoA-Extended is consistent with its harder retrieval regime: many queries have weak or deeply ranked phenotype evidence, limiting what can be recovered from local retrieval even with calibrated reasoning.

3.3.2 Effect of Self-Evolving Refinement

Figure 2 evaluates self-evolving refinement, where controller rules and arbiter prompts are iteratively updated using trajectories from a small development set. On MoA-Novel, refinement improves both GPT-5.1 and Claude 4 Sonnet, suggesting that trajectory-driven updates help distinguish supported

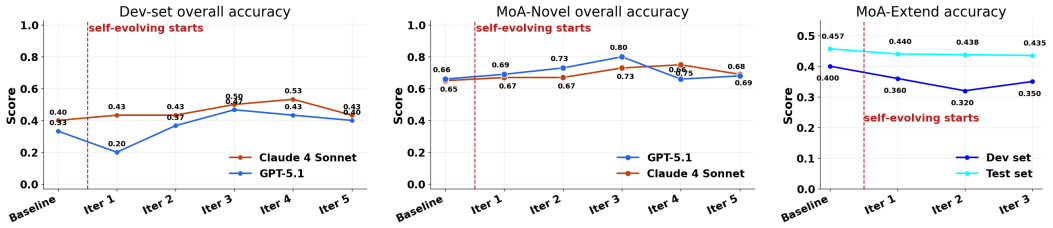


Figure 2: Self-evolving refinement curves across iterations. Controller rules and arbiter prompts are updated using development-set trajectories after the baseline step. Left two: development set and test set overall accuracy on MoA-Novel. Right: MoA-Extended accuracy on development and test sets.

known mechanisms from novel cases. The gains are not monotonic, indicating that later iterations can overfit to development trajectories rather than uniformly improve inference. On MoA-Extended, refinement does not improve accuracy and sometimes degrades performance, likely because its failure modes are broader and noisier, making rules from a small development set less transferable. We therefore restrict self-evolving refinement to the open-set MoA-Novel setting, where it targets the additional novel-detection decision, and do not apply it to the closed-set MoA-Verified or MoA-Extended settings to avoid dev-set-specific rule tuning for standard classification.

4 Analysis and Discussion

4.1 Confidence Calibration

To test whether PhenoAIR's gains rely on Cell Painting evidence rather than structure-side information, we shuffle one evidence stream across compounds while keeping the rest of the pipeline unchanged. Using GPT-5.1, we evaluate 100 MoA-Extended queries whose ground-truth MOA ranks highest in the calibrated phenotype neighborhood, i.e., where phenotype evidence is informative.

Table 4: Evidence-shuffling control on MoA-Extended (GPT-5.1, mean $\pm$ s.d. over five runs).

Setting	Phenotype	Structure side	Acc. (%)
Full PhenoAIR	matched	matched	80.0 ± 1.8
Shuffled structure side	matched	shuffled	63.2 ± 0.7
Shuffled phenotype	shuffled	matched	53.6 ± 1.0
Structure-only agent	removed	matched	50.2 ± 0.7

Shuffling only the phenotype evidence reduces accuracy by 26.4 points, to a level close to the structure-only agent, even though the structure-side inputs and reasoning pipeline are unchanged. This shows that PhenoAIR's gains depend on correctly matched, compound-specific phenotype evidence rather than on chemical structure alone. Conversely, PhenoAIR retains 63.2% accuracy when structure-side evidence is shuffled, indicating that the two evidence streams are complementary.

4.2 Robustness Across Image Features

We test whether PhenoAIR depends on a particular morphological feature space by comparing CellProfiler features with CellCLIP embeddings (Figure 3). Calibration yields only modest retrieval improvements, increasing Recall@10 from 0.545 to 0.570 for CellProfiler and from 0.575 to 0.605 for CellCLIP. This indicates that calibration is not mainly a re-ranking mechanism. The small change in Recall@10 also suggests that the downstream gains are not explained by improved neighbor ranking alone; they likely depend on the additional reliability context attached to the retrieved evidence. End-to-end, PhenoAIR reaches 60.0% accuracy with CellProfiler features and 62.6% with CellCLIP, compared with 9.5% and 14.0% for kNN on the same features. The gap between feature spaces is small relative to either configuration's margin over kNN, indicating that PhenoAIR's performance is not tied to a particular morphological representation.

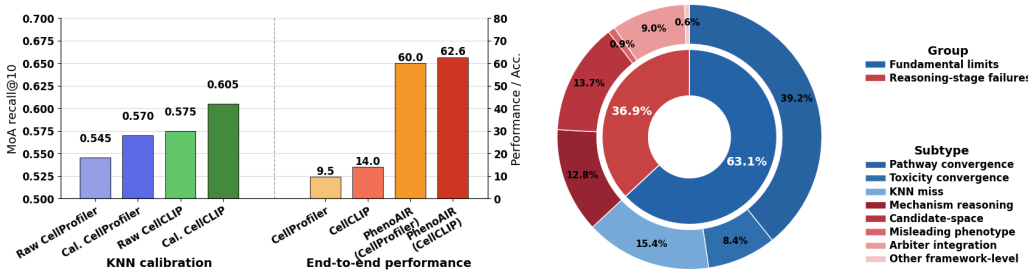


Figure 3: (Left) MoA Recall@10 with and without calibration, and accuracy comparison on MoA-Verified, both shown across CellProfiler and CellCLIP features. (Right) Failure mode distribution on MoA-Extended. Inner ring: high-level categories; outer ring: subtype breakdown.

4.3 Limitations and Failure Modes

We categorize failure cases on MoA-Extended in Figure 3. Most errors arise from fundamental evidence limitations, including pathway-level convergence, toxicity-driven convergence, and retrieval misses, where the local phenotype neighborhood either reflects non-specific cellular responses or lacks sufficient evidence for the correct MOA. This supports our central motivation: morphological similarity is not a mechanistic explanation, and Cell Painting-based MOA prediction requires explicit reasoning about evidence reliability. The remaining errors come from reasoning-stage failures, such as mechanism misinterpretation, candidate-space errors, misleading phenotype evidence, and imperfect arbiter integration. These failures indicate that PhenoAIR is not a complete solution to mechanistic interpretation, but they also identify actionable directions for future work, including stronger mechanism grounding, candidate validation, and arbitration under conflicting evidence.

5 Conclusion

In this paper, we argue that Cell Painting-based MOA prediction should be treated as an evidence reasoning problem rather than a direct representation-matching task. PhenoAIR instantiates this view by evaluating the reliability and consistency of retrieved phenotypic evidence before committing to a mechanistic prediction. Across controlled, realistic, and open-world settings, our results show that structured reasoning is most useful when similarity-based evidence is noisy, incomplete, or ambiguous. These findings suggest that future phenotype-based discovery systems should complement improved representations with explicit mechanisms for evidence evaluation and decision-making.

Acknowledgements

This work was funded in part by the National Science Foundation under award number IIS-2145625 and by the National Institutes of Health under award numbers R01AI188576 and R21EB030294.

References

[1] Arevalo, J., Su, E., Ewald, J. D., Van Dijk, R., Carpenter, A. E., and Singh, S. (2024). Evaluating batch correction methods for image-based cell profiling. *Nature Communications*, 15(1):6516.

[2] Averly, R., Baker, F. N., Watson, I. A., and Ning, X. (2025). Liddia: Language-based intelligent drug discovery agent. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 12015–12039.

[3] Bray, M.-A., Singh, S., Han, H., Davis, C. T., Borgeson, B., Hartland, C., Kost-Alimova, M., Gustafsdottir, S. M., Gibson, C. C., and Carpenter, A. E. (2016). Cell painting, a high-content image-based assay for morphological profiling using multiplexed fluorescent dyes. *Nature protocols*, 11(9):1757–1774.

[4] Caie, P. D., Walls, R. E., Ingleston-Orme, A., Daya, S., Houslay, T., Eagle, R., Roberts, M. E., and Carragher, N. O. (2010). High-content phenotypic profiling of drug response signatures across distinct cancer cells. *Molecular cancer therapeutics*, 9(6):1913–1926.

[5] Carpenter, A. E., Jones, T. R., Lamprecht, M. R., Clarke, C., Kang, I. H., Friman, O., Guertin, D. A., Chang, J. H., Lindquist, R. A., Moffat, J., et al. (2006). Cellprofiler: image analysis software for identifying and quantifying cell phenotypes. *Genome biology*, 7(10):R100.

[6] Chandrasekaran, S. N., Ackerman, J., Alix, E., Ando, D. M., Arevalo, J., Bennion, M., Boisseau, N., Borowa, A., Boyd, J. D., Brino, L., et al. (2023). Jump cell painting dataset: morphological impact of 136,000 chemical and genetic perturbations. *BioRxiv*, pages 2023–03.

[7] Chandrasekaran, S. N., Alix, E., Arevalo, J., Borowa, A., Byrne, P. J., Charles, W. G., Chen, Z. S., Cimini, B. A., Deng, B., Doench, J. G., et al. (2025). Morphological map of under-and overexpression of genes in human cells. *Nature methods*, 22(8):1742–1752.

[8] Chandrasekaran, S. N., Cimini, B. A., Goodale, A., Miller, L., Kost-Alimova, M., Jamali, N., Doench, J. G., Fritchman, B., Skepner, A., Melanson, M., et al. (2024). Three million images and morphological profiles of cells treated with matched chemical and genetic perturbations. *Nature Methods*, 21(6):1114–1121.

[9] Chen, J., Pham, T.-H., Wang, Y., and Zhang, P. (2025). Integrating biological knowledge for robust microscopy image profiling on de novo cell lines. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22846–22856.

[10] Corsello, S. M., Bittker, J. A., Liu, Z., Gould, J., McCarren, P., Hirschman, J. E., Johnston, S. E., Vrcic, A., Wong, B., Khan, M., et al. (2017). The drug repurposing hub: a next-generation drug library and information resource. *Nature medicine*, 23(4):405–408.

[11] Cui, H., Wang, C., Maan, H., Pang, K., Luo, F., Duan, N., and Wang, B. (2024). scgpt: toward building a foundation model for single-cell multi-omics using generative ai. *Nature methods*, 21(8):1470–1480.

[12] Du, Y., Li, S., Torralba, A., Tenenbaum, J. B., and Mordatch, I. (2023). Improving factuality and reasoning in language models through multiagent debate, 2023. *URL https://arxiv.org/abs/2305.14325*, 3.

[13] Ewald, J. D., Titterton, K. L., Bäuerle, A., Beatson, A., Boiko, D. A., Cabrera, Á. A., Cheah, J., Cimini, B. A., Gorissen, B. L., Harrill, J., et al. (2026). Cell painting for cytotoxicity and mode-of-action analysis in primary human hepatocytes. *Cell Systems*.

[14] Fernando, C., Banarse, D., Michalewski, H., Osindero, S., and Rocktäschel, T. (2023). Promptbreeder: Self-referential self-improvement via prompt evolution. *arXiv preprint arXiv:2309.16797*.

[15] Fradkin, P., Azadi, P., Suri, K., Wenkel, F., Bashashati, A., Sypetkowski, M., and Beaini, D. (2024). How molecules impact cells: Unlocking contrastive phenomolecular retrieval. *Advances in Neural Information Processing Systems*, 37:110667–110701.

[16] Huang, K., Zhang, S., Wang, H., Qu, Y., Lu, Y., Roohani, Y., Li, R., Qiu, L., Li, G., Zhang, J., et al. (2025). Biomni: A general-purpose biomedical ai agent. *biorxiv*.

[17] Kraus, O., Comitani, F., Urbanik, J., Kenyon-Dean, K., Arumugam, L., Saberian, S., Wognum, C., Celik, S., and Haque, I. S. (2025). Rxrx3-core: Benchmarking drug-target interactions in high-content microscopy. *arXiv preprint arXiv:2503.20158*.

[18] Kraus, O., Kenyon-Dean, K., Saberian, S., Fallah, M., McLean, P., Leung, J., Sharma, V., Khan, A., Balakrishnan, J., Celik, S., et al. (2024). Masked autoencoders for microscopy are scalable learners of cellular biology. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11757–11768.

[19] Li, M., Kilicoglu, H., Xu, H., and Zhang, R. (2025). Biomedrag: A retrieval augmented large language model for biomedicine. *Journal of Biomedical Informatics*, 162:104769.

[20] Liu, T., Xiao, Y., Luo, X., Xu, H., Zheng, W., and Zhao, H. (2024). Geneverse: A collection of open-source multimodal large language models for genomic and proteomic research. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 4819–4836.

[21] Ljosa, V., Sokolnicki, K. L., and Carpenter, A. E. (2012). Annotated high-throughput microscopy image sets for validation. *Nature methods*, 9(7):637.

[22] Lu, M., Weinberger, E., Kim, C., and Lee, S.-I. (2025). Cellclip-learning perturbation effects in cell painting via text-guided contrastive learning.

[23] Moshkov, N., Bornholdt, M., Benoit, S., Smith, M., McQuin, C., Goodman, A., Senft, R. A., Han, Y., Babadi, M., Horvath, P., et al. (2024). Learning representations for image-based profiling of perturbations. *Nature communications*, 15(1):1594.

[24] Ramezani, M., Weisbart, E., Bauman, J., Singh, A., Yong, J., Lozada, M., Way, G. P., Kavari, S. L., Diaz, C., Leardini, E., et al. (2025). A genome-wide atlas of human cell morphology. *Nature methods*, 22(3):621–633.

[25] Sanchez-Fernandez, A., Rumetshofer, E., Hochreiter, S., and Klambauer, G. (2023). Cloome: contrastive learning unlocks bioimaging databases for queries with chemical structures. *Nature Communications*, 14(1):7339.

[26] Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Hambro, E., Zettlemoyer, L., Cancedda, N., and Scialom, T. (2023). Toolformer: Language models can teach themselves to use tools. *Advances in neural information processing systems*, 36:68539–68551.

[27] Seal, S., Trapotsi, M.-A., Spjuth, O., Singh, S., Carreras-Puigvert, J., Greene, N., Bender, A., and Carpenter, A. E. (2025). Cell painting: a decade of discovery and innovation in cellular imaging. *Nature methods*, 22(2):254–268.

[28] Soman, K., Rose, P. W., Morris, J. H., Akbas, R. E., Smith, B., Peetoom, B., Villouta-Reyes, C., Cerono, G., Shi, Y., Rizk-Jackson, A., et al. (2024). Biomedical knowledge graph-optimized prompt generation for large language models. *Bioinformatics*, 40(9):btae560.

[29] Su, H., Long, W., and Zhang, Y. (2025). Biomaster: Multi-agent system for automated bioinformatics analysis workflow. *bioRxiv*, pages 2025–01.

[30] Swanson, K., Wu, W., Bulaong, N. L., Pak, J. E., and Zou, J. (2025). The virtual lab of ai agents designs new sars-cov-2 nanobodies. *Nature*, 646(8085):716–723.

[31] Tang, Q., Ratnayake, R., Seabra, G., Jiang, Z., Fang, R., Cui, L., Ding, Y., Kahveci, T., Bian, J., Li, C., et al. (2024). Morphological profiling for drug discovery in the era of deep learning. *Briefings in bioinformatics*, 25(4):bbae284.

[32] Wong, D. R., Logan, D. J., Hariharan, S., Stanton, R., Clevert, D.-A., and Kiruluta, A. (2023). Deep representation learning determines drug mechanism of action from cell painting images. *Digital Discovery*, 2(5):1354–1367.

[33] Yang, C., Lu, J., Wan, H., Yu, J., and Qin, F. (2025). From what to why: A multi-agent system for evidence-based chemical reaction condition reasoning. *arXiv preprint arXiv:2509.23768*.

Appendix

Section A reviews prior work on Cell Painting morphological profiling and the use of large language models for biological reasoning, contextualizing PhenoAIR within both lines of research.

Section B provides additional quantitative results and qualitative case analyses that complement the main paper.

Section C provides algorithmic details of each PhenoAIR component, including the construction of the externalized reliability layer, the candidate-centric memory schema, the controller's rule set, the arbiter's eligibility constraints, and the self-evolving refinement procedure.

Section D describes how the benchmark was constructed from raw data, how ground-truth MOA labels were curated, and how predictions are evaluated across the three settings.

Section E lists the full prompts used for each agent (Phenotype Agent, Mechanism Agent, Arbiter, and the offline calibration LLM), along with the schema definitions for all tools available to the agents during inference.

Algorithm Summary

Algorithm 1 summarizes the full inference procedure. We separate global pre-computation from query-time inference. The reliability memory, calibrated retrieval rule, and mechanistic family ontology are constructed once using the reference set and development split. At test time, a query only invokes the fixed calibrated retrieval module, the phenotype and mechanism agents, the controller, and the final arbiter. For open-set experiments, the controller rule registry and arbiter heuristic section may be selected by the offline self-evolving procedure described in Appendix C.3.4. These artifacts are fixed before final test evaluation and are not updated using test outcomes.

A Related Works

A.1 Cell Painting Profiling

High-content imaging assays such as Cell Painting enable large-scale characterization of cellular responses to chemical and genetic perturbations through multiplexed microscopy measurements of cellular morphology [3, 8, 6, 17]. Early work demonstrated that high-dimensional morphological profiles could capture biologically meaningful drug response signatures and support mechanism-of-action (MOA) analysis through clustering and nearest-neighbor similarity analysis [4].

Recent progress has focused heavily on representation learning for cellular morphology. Traditional approaches relied on hand-crafted features [5] extracted from cellular segmentation and intensity statistics, while more recent methods leverage vision foundation models and self-supervised learning to learn scalable microscopy representations. In particular, large-scale self-supervised microscopy foundation models based on masked autoencoding and Vision Transformers have substantially improved biological signal recovery and cross-dataset generalization [18, 9]. Contrastive multimodal learning has further extended this paradigm by aligning cellular phenotypes with molecular structure representations in a shared latent space [25]. Recent works such as MolPhenix [15] and CellCLIP [22] formulate cross-modal phenomolecular retrieval as a contrastive learning problem, enabling zero-shot retrieval of molecular structures from microscopy phenotypes. These learned representations have enabled a broad range of downstream applications, including MOA prediction, perturbation similarity analysis, target and pathway inference, virtual screening, and cross-modal retrieval between molecules and phenotypic profiles. However, despite substantial advances in representation quality, downstream inference in most existing pipelines still fundamentally relies on representation matching. Treatments are typically compared through nearest-neighbor retrieval, similarity ranking, or embedding-space aggregation, where biological conclusions are derived from phenotypic proximity in the learned representation space [23].

Recent work has increasingly recognized that this paradigm is heavily affected by confounding technical and biological variation. Multiple studies show that learned representations simultaneously capture both phenotypic signal and nuisance variation such as batch, source, and experimental artifacts, motivating extensive efforts on batch correction, representation disentanglement, and domain

generalization. In practice, realistic phenotypic neighborhoods frequently contain noisy or mechanistically heterogeneous neighbors due to polypharmacology, inactive perturbations, cytotoxicity, source-specific variation, and phenotypic convergence [24, 15, 1]. Nevertheless, most existing approaches address these challenges primarily through improved feature learning, retrieval calibration, or batch correction, while continuing to treat retrieved neighbors as direct evidence for prediction [31]. In contrast, our work argues that Cell Painting-based MOA prediction should be formulated not purely as a representation matching problem, but as an evidence reasoning problem over noisy phenotypic observations. Rather than directly aggregating nearest neighbors in embedding space, our framework treats retrieved phenotypic matches as uncertain and potentially conflicting evidence that must be actively evaluated, filtered, and integrated through agentic inference.

A.2 LLMs for Biological Reasoning

LLMs are increasingly being explored as reasoning systems for biological and biomedical research. Early efforts primarily focused on biomedical adaptation through domain-specific pretraining [11] and retrieval-augmented generation [28], enabling LLMs to access biological literature, pathway databases, and structured biomedical knowledge. More recent work has shifted toward agentic scientific systems that combine iterative reasoning, tool use, and autonomous workflow execution for complex biological tasks.

Recent systems [2, 33] formulate drug discovery as an autonomous reasoning process in which LLM agents iteratively explore chemical space and optimize molecular properties through external computational tools. The Virtual Lab [30] further extends this paradigm toward collaborative AI scientific teams, where multiple agents coordinate across hypothesis generation, protein modeling, and experimental planning to design novel SARS-CoV-2 nanobodies with experimental validation. Other recent frameworks [29, 16] investigate multi-agent architectures for automating bioinformatics workflows, tool orchestration, error recovery, and computational analysis pipelines. Collectively, these works suggest a broader transition from static prediction pipelines toward interactive and evidence-driven biological reasoning systems. In parallel, multimodal biological foundation models such as Geneverse [20] and related genomic-proteomic LLM systems have explored integrating heterogeneous biological modalities within unified representation and reasoning frameworks. These approaches demonstrate the growing interest in combining large language models with multimodal biological data and structured scientific workflows.

However, despite recent progress in biological agent systems and multimodal biomedical LLMs, Cell Painting pipelines remain largely dominated by representation matching approaches based on nearest-neighbor retrieval, similarity aggregation, and embedding-space ranking over phenotypic representations. Existing work primarily focuses on improving representation quality, retrieval performance, or batch correction, while explicit reasoning over noisy and potentially conflicting phenotypic evidence remains largely unexplored [27, 32]. In contrast, our work formulates Cell Painting-based MOA prediction as an evidence reasoning problem in which retrieved phenotypic neighbors must be iteratively evaluated, filtered, and integrated through coordinated phenotypic and mechanistic reasoning

B Extended Experimental Results

B.1 Additional Quantitative Results

B.1.1 Full Benchmark Results

Tables 5 and 6 report the full benchmark results across two backbone LLMs (GPT-5.1 and Claude 4), complementing the main results presented in Section 3.2. On MoA-Novel (Table 5), we evaluate Precision, Recall, and F1 of novel-MOA prediction in addition to Overall Accuracy, and include ablations of self-evolving refinement and reliability calibration to isolate their respective contributions to open-set inference. On MoA-Verified and MoA-Extended (Table 6), we provide a stepwise ablation that progressively introduces PhenoAIR's core components: multi-agent reasoning, candidate evidence memory, and reliability calibration onto the Base LLM. Across both tables, PhenoAIR consistently improves over its ablated variants and over non-agentic baselines on both backbones, with the most pronounced gains observed on MoA-Novel where the open-set decision benefits most from the full combination of components.

Table 5: Full results on the MoA-Novel benchmark across two backbone LLMs (GPT-5.1 and Claude 4). We report Overall Accuracy, and Precision, Recall, and F1 of the novel-MOA prediction. Two ablations of PhenoAIR are included: *w/o Self-evolving* (no offline rule refinement) and *w/o Calibration* (no externalized reliability calibration). Results are mean ± std over multiple runs with different random seeds.

	Method	Overall Acc.	Precision (Novel)	Recall (Novel)	F1 (Novel)
GPT 5.1	Structure-only	44.2 ± 3.4	61.5 ± 3.6	47.2 ± 5.0	53.3 ± 4.0
	Workflow	58.0 ± 0.3	72.4 ± 2.5	53.2 ± 1.8	61.3 ± 0.8
	Single Agent	48.8 ± 2.4	68.8 ± 2.0	66.0 ± 1.8	67.3 ± 1.1
	PhenoAIR	79.4 ± 2.2	99.0 ± 1.5	59.8 ± 3.5	74.7 ± 2.9
	w/o Self-evolving	66.0 ± 2.0	96.0 ± 3.0	48.0 ± 4.0	64.0 ± 3.0
	w/o Calibration	75.5 ± 2.5	93.5 ± 3.0	58.0 ± 4.0	71.5 ± 3.5
Claude 4	Structure-only	43.2 ± 1.1	69.0 ± 1.9	40.0 ± 1.4	50.6 ± 1.0
	Workflow	63.6 ± 1.5	78.0 ± 2.0	64.0 ± 2.0	70.2 ± 1.2
	Single Agent	52.0 ± 1.2	69.6 ± 1.8	64.0 ± 1.6	66.9 ± 1.1
	PhenoAIR	73.6 ± 3.1	96.8 ± 2.4	60.0 ± 2.8	74.1 ± 2.0
	w/o Self-evolving	65.6 ± 2.5	93.0 ± 4.0	51.0 ± 4.0	66.0 ± 3.0
	w/o Calibration	70.5 ± 3.2	91.5 ± 5.0	58.5 ± 3.5	71.5 ± 3.0

Table 6: Component ablation of PhenoAIR on MoA-Verified and MoA-Extended across two backbone LLMs. Variants are constructed incrementally: *Base LLM* predicts MOA from compound information alone; *PhenoAIR-Base* adds Cell Painting retrieval and multi-agent reasoning; *+Memo.* adds the candidate-centric evidence memory; *+Memo. & Cal.* additionally includes the externalized reliability calibration layer. We report Accuracy (Acc) and Any-match (Hit Rate).

	Dataset	MoA-Verified		MoA-Extended	
	Method	Acc	Any-match	Acc	Any-match
GPT 5.1	Base LLM	16.4	23.6	18.7	25.3
	PhenoAIR-Base	55.6	66.1	43.9	54.4
	+ Memo.	60.1	73.0	44.5	55.1
	+ Memo. & Cal.	62.6	74.6	45.6	56.2
Claude 4	Base LLM	26.7	31.0	23.3	30.2
	PhenoAIR-Base	55.0	69.1	42.3	54.8
	+ Memo.	59.3	72.5	44.2	56.1
	+ Memo. & Cal.	61.5	74.2	45.8	59.9

The main results in Table 2 and 6 show that structure-only inference can be a strong control in closed-set MOA prediction, which is expected because chemical structure and mechanism-side resources contain substantial MOA signal. This does not replace the Cell Painting analysis problem: structure-only inference does not evaluate whether retrieved phenotypic neighbors are reliable, misleading, or mechanistically supported. The ablations further show that PhenoAIR improves over a Base LLM through multi-agent phenotype-mechanism reasoning, candidate memory, and reliability calibration. Importantly, PhenoAIR enforces phenotype grounding for final canonical predictions: mechanism-side evidence may propose or prioritize candidates, but the final MOA must be supported by the Phenotype Agent using Cell Painting-derived evidence. Thus, our core claim is not that morphology alone dominates chemical evidence, but that noisy Cell Painting retrieval should be reasoned over as calibrated candidate-level evidence rather than consumed as direct nearest-neighbor labels.

B.1.2 Robustness Across Image Feature Representations

We analyze the effect of feature representations, batch-effect correction, and cross-source aggregation strategies on retrieval performance. Table 7 and Table 8 report retrieval-based results under different combinations of feature backbones and preprocessing settings. Across both MoA-Verified and MoA-Extended, deep learning representations generally outperform classical CellProfiler features, with CellCLIP achieving the strongest overall retrieval performance. Applying batch-effect correction

Table 7: Retrieval performance on the MoA-Verified benchmark under different feature representations, preprocessing strategies, and cross-source aggregation schemes. Raw features denote the original representations, while normalized features apply batch-effect correction using plate-level normalization and control-based whitening. Global aggregation constructs a single compound representation across sources, whereas per-source voting performs source-specific retrieval followed by cross-source score aggregation.

Aggregation	Model	Top-1 Acc	Top-5 Acc	F1	Any Overlap
Global	CellProfiler Raw	7.5	30.0	10.3	18.5
	CellProfiler Normalized	9.5	29.5	10.2	23.5
	CA-MAE Raw	7.5	28.0	9.2	19.5
	CA-MAE Normalized	13.0	31.5	10.6	23.5
	CellCLIP Raw	**14.5**	**37.5**	**12.8**	24.0
	CellCLIP Normalized	14.0	35.5	11.7	26.5
Per-Source	CellProfiler Raw	8.5	23.5	9.3	17.5
	CellProfiler Normalized	10.0	29.5	10.4	23.0
	CA-MAE Raw	8.0	29.5	8.8	21.0
	CA-MAE Normalized	11.5	30.5	10.1	21.0
	CellCLIP Raw	13.5	35.5	10.8	**27.0**
	CellCLIP Normalized	13.0	**37.5**	11.2	26.0

Table 8: Retrieval performance on the MoA-Extended benchmark under different feature representations, preprocessing strategies, and cross-source aggregation schemes.

Aggregation	Model	Top-1 Acc	Top-5 Acc	F1	Any Overlap
Global	CellProfiler Raw	3.3	10.2	3.7	7.5
	CellProfiler Normalized	6.8	16.5	6.3	12.7
	CA-MAE Raw	3.5	13.7	4.9	10.3
	CA-MAE Normalized	6.5	14.7	5.4	12.5
	CellCLIP Raw	5.5	16.2	5.7	11.0
	CellCLIP Normalized	10.2	**19.7**	**7.1**	**16.0**
Per-Source	CellProfiler Raw	3.2	11.5	3.7	9.0
	CellProfiler Normalized	7.7	17.0	5.4	13.7
	CA-MAE Raw	3.5	12.5	4.1	8.5
	CA-MAE Normalized	5.3	15.2	4.7	12.5
	CellCLIP Raw	5.8	16.8	4.6	12.8
	CellCLIP Normalized	**10.3**	19.0	5.5	15.5

consistently improves results for both handcrafted and learned representations, highlighting the importance of mitigating source-specific technical variation in large-scale Cell Painting datasets. The impact of aggregation strategy differs across settings. Under MoA-Verified, per-source voting improves Top-5 and overlap-based metrics in several cases, suggesting that preserving source-specific phenotype structure can improve retrieval diversity when reference profiles are relatively reliable. In contrast, under MoA-Extended, gains from per-source voting are less consistent due to increased heterogeneity and noisier retrieval neighborhoods.

Table 9 shows that PhenoAIR remains consistently effective across all feature backbones. While stronger representations lead to improved overall performance, the relative gains from agentic inference persist across both handcrafted and deep-learning-based features. This suggests that the benefits of PhenoAIR arise primarily from reliability-aware reasoning over retrieved evidence rather than dependence on a specific embedding space.

B.1.3 Runtime and API Usage

Table 10 reports per-query inference cost across all LLM-based methods, including the average number of LLM calls, cumulative input and output tokens across calls, and wall time. All methods use the same base model and identical hardware configurations.

Table 9: Performance of PhenoAIR across different Cell Painting feature representations.

Dataset	MoA-Verified		MoA-Extended		MoA-Novel	
Metrics	Acc	Any-match	Acc	Any-match	Acc	F1 (Novel)
CellCLIP	62.6 ± 0.7	74.6 ± 0.6	45.6 ± 0.2	56.2 ± 0.3	79.4 ± 2.2	74.7 ± 2.9
CA-MAE	62.0 ± 0.5	74.5 ± 0.4	42.5 ± 0.3	52.8 ± 0.3	77.0 ± 1.5	74.4 ± 1.9
CellProfiler	60.0 ± 0.6	74.1 ± 0.5	42.2 ± 0.3	53.9 ± 0.3	76.6 ± 1.8	72.5 ± 2.5

Table 10: Per-query inference cost across methods. We report the average number of LLM calls and the cumulative input tokens, output tokens, and wall time across all calls per query. All LLM-based methods use the same base model and identical hardware configurations.

Method	LLM calls/query	Input tokens/query	Output tokens/query	Wall time/query
Base LLM	1	0.3k	180	3.0s
Structure-only	1	0.7k	160	3.7s
Workflow	3	11k	800	12.8s
Singe Agent	10	65k	1.6k	31s
PhenoAIR	14	170k	8k	80s

PhenoAIR's cost is higher than non-agentic baselines, reflecting the multi-round reasoning required for evidence integration and verification. The cumulative 170k input tokens correspond to an average of approximately 12k tokens per LLM call across the 14 calls per query, comparable to the per-call prompt size of fixed workflow baselines (11k tokens per call). The increased total cost arises from the number of reasoning rounds, not from inflated single-call prompts.

Importantly, PhenoAIR's cost adapts to query difficulty rather than remaining fixed. Easy queries, those for which the ground-truth MOA appears among the top-ranked KNN neighbors, typically terminate within half the average compute, consuming approximately 90k cumulative input tokens, 3.5k output tokens, and fewer LLM calls. Hard queries that require iterative refinement, in contrast, incur correspondingly higher cost. This adaptation is enabled by the rule-based controller, which terminates inference once the candidate memory exhibits stable, high-confidence support, and continues refinement only when evidence remains weak or conflicting. Non-agentic baselines apply uniform compute regardless of query difficulty.

B.2 Decision Reconstruction Case Studies

B.2.1 Case Study: AZD8931

Overview. AZD8931 illustrates a successful subtype-resolution case within a mechanistically ambiguous receptor tyrosine kinase (RTK) family. Structural and target-based evidence broadly supported RTK inhibition but could not reliably distinguish EGFR from related RTK subclasses including FGFR, KIT, and VEGFR inhibition. After iterative refinement, the phenotype agent identified EGFR inhibitor as the only RTK subtype with localized support in the calibrated neighborhood, allowing the arbiter to resolve the subtype ambiguity.

Decision Reconstruction

Ground-truth MOAs: EGFR inhibitor
Compound annotation: EGFR / ERBB2 / ERBB3
Case type: Successful subtype disambiguation
Final prediction: EGFR inhibitor
Input: CNC(=O)CN1CCC(Oc2cc3c(=Nc4cccc(Cl)c4F)[nH]cnc3cc2OC)CC1

Decision reconstruction

Mechanism evidence. Structural neighbors and predicted targets broadly supported RTK-family inhibition, but could not reliably distinguish EGFR from FGFR, KIT, or VEGFR subclasses.

⇓

Phenotype evidence. Within the RTK family, EGFR inhibitor was the only subtype showing localized support in the calibrated top-20 phenotype neighborhood.

⇓

Final arbitration. The arbiter selected EGFR inhibitor because phenotype evidence provided the only subtype-specific support signal within the RTK family.

Iterative refinement trajectory

Round	Top candidate	Key transition
0	PI3K inhibitor	Initial phenotype neighborhood was dominated by noisy kinase-related signals.
1	JAK inhibitor	Reliability calibration penalized unstable retrieval anchors and reduced confidence in early kinase attractors.
2	EGFR inhibitor	Localized RTK subtype evidence emerged after group-level inspection of calibrated neighborhood support.

Decision-relevant evidence

- **Calibration.** EGFR was globally flagged as phenotype-unreliable, but remained locally visible after calibration rather than being fully suppressed. In contrast, early PI3K and JAK signals were associated with unstable or high-risk retrieval anchors.
- **Phenotype Agent (P).** EGFR inhibitor was the only RTK subtype with direct localized support in the calibrated top-20 neighborhood. FGFR, KIT, and VEGFR remained mechanistically plausible but showed no phenotype support.
- **Mechanism Agent (M).** Predicted targets, pharmacophore similarity, and structure neighbors broadly supported RTK-family inhibition. However, mechanism evidence alone could not reliably distinguish EGFR from related RTK subclasses.
- **Arbiter (A).** The final decision relied on phenotype evidence as the subtype tie-breaker inside a mechanistically overlapping RTK family.

Baseline comparison

Method	Prediction	Correct
Structure-only	tyrosine kinase inhibitor	×
Single-agent	VEGFR inhibitor	×
Workflow	SRC inhibitor	×
PhenoAIR	EGFR inhibitor	✓

Takeaway. This example illustrates the intended division of labor in PhenoAIR. Mechanism evidence defines the plausible RTK family, while localized phenotype evidence resolves the subtype ambiguity. The case also demonstrates how iterative refinement can recover from initially misleading kinase-related phenotype signals rather than committing to the strongest early retrieval attractor.

B.2.2 Case Study: IT1t

Overview. IT1t illustrates a mechanism-guided hypothesis expansion case in which the correct MOA does not emerge from the initial phenotype neighborhood. Early phenotype retrieval was dominated by noisy dopamine-, kinase-, and PDE-related signals. During iterative refinement, the mechanism agent introduced a non-obvious CXCR4-driven hypothesis that was not among the dominant initial phenotype candidates. The phenotype agent subsequently re-inspected this candidate within the calibrated neighborhood and identified sparse but coherent support, allowing the arbiter to accept the CC chemokine receptor antagonist hypothesis with low confidence rather than discarding it due to weak initial retrieval evidence.

Decision Reconstruction

Ground-truth MOAs: CC chemokine receptor antagonist
Compound annotation: CXCR4
Case type: Mechanism-guided candidate expansion
Final prediction: CC chemokine receptor antagonist
Input: CC1(C)CN2C(CSC(=NC3CCCCC3)NC3CCCCC3)=CSC2=N1
Decision reconstruction

Initial phenotype retrieval. The calibrated phenotype neighborhood was dominated by dopamine-, kinase-, and PDE-related candidates with weak or unstable retrieval anchors.

⇓

Mechanism-guided hypothesis expansion. Mechanism evidence introduced a strong and specific CXCR4 signal, proposing CC chemokine receptor antagonism as a new active hypothesis.

⇓

Phenotype reinspection and arbitration. The phenotype agent identified sparse but coherent local support for the CC chemokine hypothesis, and the arbiter accepted the mechanism-led explanation despite globally weak phenotype evidence.

Iterative refinement trajectory

Round	Top candidate	Key transition
0	dopamine receptor antagonist	Initial phenotype retrieval dominated by noisy GPCR- and kinase-related signals.
1	mTOR inhibitor	Calibration reduced confidence in unstable retrieval anchors, but no coherent phenotype explanation emerged.
2	CC chemokine receptor antagonist	Mechanism-guided CXCR4 hypothesis was re-inspected against the calibrated neighborhood and retained after sparse but non-contradictory phenotype support was identified.

Decision-relevant evidence

- **Calibration.** Several dominant phenotype neighbors, including dopamine- and PDE-related candidates, were associated with weak or high-risk retrieval anchors. In contrast, the correct CC chemokine label was sparse but remained inspectable rather than being suppressed.
- **Phenotype Agent (P).** After candidate-specific reinspection, the phenotype agent identified one calibrated supporting reference for CC chemokine receptor antagonism at rank 1 within the candidate check. No contradictory phenotype evidence was observed.
- **Mechanism Agent (M).** Mechanism evidence strongly supported GPCR-directed activity with a dominant CXCR4 signal and rejected dopamine-, kinase-, and mTOR-related alternatives. No competing mechanistic family received comparable support.
- **Arbiter (A).** The final decision accepted a mechanism-introduced candidate only after phenotype validation confirmed sparse but coherent local support.

Baseline comparison

Method	Prediction	Correct
Structure-only	CXCR4 antagonist	×
Workflow	KIT inhibitor	×
Singel Agent	histamine receptor antagonist	×
PhenoAIR	CC chemokine receptor antagonist	✓

Takeaway. This example illustrates one of the central behaviors of PhenoAIR: mechanism evidence can introduce a previously inactive candidate hypothesis, but the system does not accept that hypothesis blindly. Instead, the phenotype agent re-checks the proposed mechanism against calibrated local evidence before arbitration. The final prediction therefore emerges from iterative hypothesis generation and evidence verification rather than from static top-k retrieval ranking.

B.2.3 Case Study: CH5132799

Overview. CH5132799 illustrates an upper-bound ambiguity case in which PhenoAIR correctly identifies the PI3K–mTOR signaling axis but cannot reliably determine the benchmark primary label between two tightly coupled pathway siblings. Both phenotype and mechanism evidence strongly support PI3K and mTOR inhibition, and the calibrated retrieval signals explicitly flag this pair as highly confusable in Cell Painting space. Importantly, the system remains confined to the correct mechanistic family throughout the entire reasoning trajectory and retains the ground-truth PI3K inhibitor label as an active candidate.

Decision Reconstruction

Ground-truth MOAs: PI3K inhibitor
Compound annotation: MTOR / PIK3CA / PIK3CB / PIK3CD / PIK3CG
Case type: Upper-bound pathway-level ambiguity
Final prediction: mTOR inhibitor
Other candidates: PI3K inhibitor
Input: CS(=O)(=O)N1CCc2c(-c3cnc(=N)[nH]c3)nc(N3CCOCC3)nc21

Decision reconstruction

Shared pathway evidence. Both phenotype and mechanism evidence consistently mapped the compound to the PI3K–mTOR signaling axis.

⇓

Persistent pathway-level ambiguity. Calibrated phenotype evidence showed strong support for both PI3K and mTOR inhibition, while mechanism evidence also supported both labels without a decisive tie-breaker.

⇓

Final arbitration. The arbiter selected mTOR inhibitor because the phenotype trajectory remained slightly more stable for mTOR across rounds, while PI3K inhibitor remained an active secondary candidate throughout the run.

Iterative refinement trajectory

Round	Top candidate	Key transition
0	mTOR inhibitor	Initial calibrated phenotype neighborhood strongly supported both PI3K and mTOR inhibition, with slightly denser local support for mTOR.
1	mTOR inhibitor	Group-constrained reinspection confirmed persistent PI3K–mTOR pathway ambiguity; both labels remained highly supported without contradiction.

Decision-relevant evidence

- **Calibration.** Both PI3K inhibitor and mTOR inhibitor received high-trust calibrated support and were explicitly flagged with pair-confusion warnings, indicating that the retrieval space could not reliably separate the two pathway labels.
- **Phenotype Agent (P).** The phenotype agent identified strong support for both PI3K and mTOR inhibition. mTOR had slightly more exact local hits in the top-20 neighborhood, while PI3K contained the best calibrated individual anchor. No contradictory phenotype evidence was observed for either label.
- **Mechanism Agent (M).** Mechanism evidence strongly supported the PI3K–mTOR axis, including pharmacophore similarity, structural neighbors, and in-silico target predictions involving both MTOR and multiple PI3K isoforms. No alternative mechanistic family received meaningful support.
- **Arbiter (A).** The final decision remained confined to the biologically correct pathway family. The arbiter selected mTOR inhibitor because the phenotype trajectory remained slightly more stable for mTOR, while PI3K inhibitor was retained as an active alternative explanation.

Takeaway. This case is an upper-bound pathway ambiguity example rather than uncontrolled reasoning failure. PhenoAIR successfully localizes the compound to the correct PI3K–mTOR

signaling axis and retains the ground-truth PI3K inhibitor label throughout the reasoning trajectory. The strict primary error arises because both phenotype and mechanism evidence strongly support two closely related pathway siblings whose Cell Painting profiles are difficult to separate even after calibrated reinspection.

B.3 Self-evolving Case Studies

We analyze one self-evolution trace on the MoA-Novel benchmark. The goal of the evolving scaffold is not to update the prediction model directly, but to revise controller rules and Arbiter decision heuristics based on clustered development-set trajectories. The trace in Table 11 shows that the evolving process first proposes candidate interventions, then uses result changes and cross-round failures to retain, disable, or refine them.

Table 11: Self-evolving trace on the MoA-Novel benchmark. The evolving scaffold does not simply add rules monotonically; it proposes, evaluates, disables, and refines controller and Arbiter heuristics based on cross-round trajectory changes.

Round	Diagnosed pattern	Proposed / retained update	Evolution decision
0	Novel cases missed because the ground-truth MOA was never surfaced; verified cases also showed false novel overcalls and P/M reinforcement around wrong known labels.	Generated an early STOP rule for low-reliability phenotype cases with a strong mechanistic alternative; also proposed a canonical-survivor Arbiter fallback.	Initial proposal accepted for testing.
1	The STOP rule over-triggered and caused success-to-failure regressions in both core and novel buckets.	Added phenotype expansion rules for rejected canonical candidates and medium-reliability P/M-supported wrong labels.	Disabled the harmful STOP rule and kept the Arbiter fallback temporarily.
2	Expansion alone did not resolve true novel cases because the Arbiter still blocked novel outputs when canonical candidates remained visible.	Introduced final-state novel diagnostics: no supported canonical candidate, top-3 all rejected, and cumulative rejection rate.	Disabled the canonical-survivor fallback and kept the useful medium-reliability expansion rule.
3–4	Cross-round analysis confirmed that correct novel calls were blocked by missing final-state diagnostics and by overly conservative canonical fallback.	Retained final-state novel signals and disabled rules/prompts that prematurely forced canonical outputs or stopped refinement.	Final policy separates weak canonical visibility from genuine canonical support.

Table 12: Policy-level lessons produced by the self-evolving scaffold.

Evolved principle	Failure mode addressed	Operational signal / action
Avoid premature STOP	A STOP rule intended for unresolved novel cases caused regressions by halting recoverable core and novel success cases.	Disable over-triggered STOP rules; prefer additional phenotype inspection when evidence remains ambiguous.
Separate canonical visibility from canonical support	The Arbiter fallback treated any visible canonical candidate as enough to suppress novel outputs.	Allow novel only using terminal diagnostics rather than candidate visibility alone.
Expose final-state novel diagnostics	Per-round signals were insufficient because true novel patterns only became clear after full refinement.	Use final-state signals such as no supported canonical candidate, top-3 all rejected, and high cumulative rejection rate.

C PhenoAIR Framework Details

C.1 Externalized Reliability Calibration

This section provides additional details on the offline reliability layer used by PhenoAIR. The goal of this layer is not to solve the query directly, but to pre-compute reusable evidence-quality annotations over the reference set. These annotations are then used to condition how retrieved Cell Painting neighbors are interpreted at inference time. We describe three components: reference memory construction, the mechanistic family ontology, and calibrated retrieval.

C.1.1 Reference Memory Construction

The reference memory is constructed offline from the reference Cell Painting feature library, reference MOA labels, source metadata, and compound-level metadata. Let $\mathcal{R} = \{(z_i, y_i, m_i)\}_{i=1}^{n}$ denote the reference set, where z_i is the Cell Painting representation of a reference compound, $y_i \in \mathcal{M}$ is its canonical primary MOA label, and m_i contains metadata such as compound identity, source, and available replicate information. In our multi-source setting, the same compound may be profiled across multiple experimental sources. We therefore summarize well-level features into source-specific compound profiles by taking the median over wells belonging to the same compound and source: $z_{d,s} = \mathrm{median}\{z_w : w \in (d, s)\}$, where d denotes a reference compound and s denotes an experimental source.

Reference memory construction is implemented as an offline LLM-assisted analysis pipeline with code execution. The LLM is not used to predict query MOAs. Instead, it acts as an analysis planner and interpreter over the reference set. Given the available feature matrices, reference labels, source metadata, and candidate evidence schemas, the LLM proposes executable analysis procedures, writes code to compute reference-set measurements, executes the code, inspects the outputs, and converts the resulting measurements into compact structured annotations. This process allows the reliability layer to capture both predefined quantities, such as cross-source consistency or neighborhood purity, and dataset-specific failure patterns discovered from the reference statistics, such as unstable mechanism classes, risky reference anchors, or recurring intra-family confusions.

The analyses are organized at several granularities. At the source level, the pipeline evaluates whether a source provides discriminative and comparable phenotypic evidence by analyzing quantities such as global MOA discriminability, neighbor-label purity, cross-source agreement, and distance-distribution shift. At the MOA level, it evaluates whether a canonical MOA has a stable and specific morphological signature by analyzing compactness, separability, neighbor purity, and cross-source stability. At the reference-anchor level, it evaluates whether an individual compound is a reliable retrieval anchor, using evidence such as within-MOA consistency, hubness, generic morphology risk, and cross-source replicability. At the MOA-pair level, it analyzes whether two MOAs are empirically difficult to separate in phenotype space, especially when they are mechanistically related.

The prompt used for this offline analysis provides: (i) the available reference statistics and the data objects from which they can be computed; (ii) the analysis goal for the current memory type; (iii) the required JSON output schema; and (iv) constraints that the output must be based on computed evidence rather than direct query-time prediction. We do not include the full prompt text, since the method depends on the functional interface and output schema rather than prompt-specific wording. The generated analyses, code outputs, and cached annotations are stored and reused across queries, making the reliability memory inspectable and decoupled from test-time reasoning.

The resulting reference memory contains the following annotation types:

$$\mathcal{B} = \{\mathcal{B}_{\mathrm{src}}, \mathcal{B}_{\mathrm{moa}}, \mathcal{B}_{\mathrm{ref}}, \mathcal{B}_{\mathrm{pair}}\}.$$

Each item contains a categorical reliability state, a risk level, a compact evidence summary, and query-time usage guidance.

Source reliability. A source-level memory item summarizes whether an experimental source should be treated as globally reliable, weakly discriminative, or affected by a known context mismatch. Its functional signature is:

$\mathrm{SourceMemory}(s) \rightarrow$ (`reliability_status`, `risk_level`, `evidence`, `query_usage_guidance`).

Source-level annotations are exposed to downstream agents as reliability context, but are not used as direct multiplicative factors in the current deterministic reranking rule.

MOA phenotype reliability. An MOA-level memory item summarizes whether a canonical MOA has a stable and specific Cell Painting signature:

$$\mathrm{MOAMemory}(y) \rightarrow (\texttt{reliability_status}, \texttt{risk_level}, \texttt{bad_anchor_refs}, \texttt{pair_confusion_refs}, \texttt{evidence}).$$

This annotation is used both by the phenotype agent and by calibrated retrieval. For example, MOAs whose reference profiles show weak or unstable phenotype structure are treated as lower-confidence evidence unless the query itself provides strong local support.

Reference-anchor reliability. A reference-anchor memory item summarizes whether a specific reference compound is a reliable nearest-neighbor anchor:

$$\mathrm{RefMemory}(d) \rightarrow (\texttt{anchor_status}, \texttt{risk_level}, \texttt{primary_moas}, \texttt{evidence}).$$

This captures cases where a compound may frequently appear as a neighbor because of generic morphology, hubness, poor within-MOA consistency, or unstable cross-source behavior. This annotation is directly used in calibrated retrieval through a reference-anchor reliability factor.

Pair and intra-family ambiguity. Pair-level memory items summarize empirical ambiguity between two MOA labels:

$$\mathrm{PairMemory}(y_a, y_b) \rightarrow (\texttt{confusion_status}, \texttt{risk_level}, \texttt{shared_families}, \texttt{evidence}).$$

These annotations are especially useful when two labels are biologically related and difficult to distinguish morphologically. In the current implementation, pair and intra-family ambiguity annotations are provided to the agents as review-only context and are not directly multiplied into the calibrated similarity score. This design avoids hard-coding pairwise penalties while still allowing the agents and controller to recognize ambiguous neighborhoods. Overall, the reference memory separates two roles. First, it provides structured reliability annotations that can be used to calibrate raw retrieval evidence. Second, it provides interpretable context for the phenotype agent, mechanism agent, and controller. This separation is important because not all reliability evidence should be reduced to a scalar score.

C.1.2 Mechanistic Family Ontology

The mechanistic family ontology groups canonical MOA labels into coarse biological families. Its purpose is not to replace the canonical label space, but to support reasoning when evidence is strong at the family level but insufficient to distinguish a fine-grained MOA. Each family is represented as:

$$g = (\texttt{family_name}, \texttt{canonical_moa_set}, \texttt{family_summary}, \texttt{intra_family_differences}).$$

The field `canonical_moa_set` contains the original MOA labels in $\mathcal{M}$ and therefore serves as the join key between the ontology and the benchmark label space. The family summary describes the shared biological theme, while the intra-family difference field describes how member MOAs differ mechanistically.

The ontology is constructed offline from the full canonical MOA label list. The construction prompt provides the complete label vocabulary and asks the LLM to retrieve external biomedical background through web search, including MOA ontology information, definitions of individual MOA labels, and known relationships among targets, pathways, and pharmacological mechanisms. Based on this evidence, the LLM organizes the labels into biologically coherent mechanistic families and produces concise descriptions of both shared family mechanisms and intra-family distinctions. The resulting JSON ontology is manually inspectable and can be reused across all queries. Importantly, the ontology does not introduce new prediction labels. Final closed-set predictions must still be canonical MOAs in $\mathcal{M}$, and open-set predictions use the reserved output `novel`. The ontology only provides an intermediate structure for grouping related labels and interpreting ambiguous evidence.

The ontology is used in three places. First, it defines candidate groups when the mechanism agent finds evidence for a mechanistic family rather than a single canonical MOA. For example, structure-side evidence may support a kinase inhibitor family without resolving whether the best canonical label is *SRC inhibitor*, *RAF inhibitor*, or another kinase-related MOA. Second, it allows the phenotype agent and controller to perform group-level validation. Rather than validating only one fine-grained label, the controller can ask whether a broader family is supported in the calibrated phenotype neighborhood. Third, it restricts pairwise ambiguity analysis to biologically meaningful comparisons. If two MOAs

are empirically close in Cell Painting space and belong to the same family, the ambiguity is treated as intra-family uncertainty rather than arbitrary retrieval noise. Formally, let $\mathcal{G} = \{g_1, \ldots, g_K\}$ denote the family ontology and let

$$F(y) = \{g \in \mathcal{G} : y \in g.\texttt{canonical_moa_set}\}$$

be the family membership map for MOA y. For a pair (y_a, y_b), the offline memory marks it as an intra-family pair if

$$F(y_a) \cap F(y_b) \neq \emptyset$$

and the empirical pair-separation analysis indicates medium or high confusion risk. These intra-family ambiguity items are cached and supplied to downstream reasoning. They do not directly change the candidate space, but they help distinguish two qualitatively different situations: an unsupported noisy neighbor versus a plausible but fine-grained intra-family ambiguity.

C.1.3 Calibrated Retrieval

Calibrated retrieval converts a raw KNN neighborhood into a reliability-aware neighborhood. Given a query q, raw retrieval returns

$$\mathcal{N}(q) = \{(d_j, y_j, \rho_j, k_j)\}_{j=1}^{K},$$

where d_j is a reference compound, y_j is its primary MOA, ρ_j is the raw similarity, and k_j is the raw rank. The goal is to produce a calibrated neighborhood

$$\tilde{\mathcal{N}}(q) = \{(d_j, y_j, \tilde{\rho}_j, \tilde{k}_j, e_j)\}_{j=1}^{K},$$

where $\tilde{\rho}_j$ is a calibrated similarity, $\tilde{k}_j$ is the reranked position, and e_j records the evidence supporting the calibration decision. The calibration rule is designed as a conservative reliability-aware scoring procedure. It does not infer new labels, introduce query-specific free-form reasoning, or allow unreliable global annotations to override strong local evidence by themselves. Instead, it attenuates raw similarity only when the reference memory indicates that a retrieved neighbor should be treated as lower-confidence evidence. The scoring rule and its operating parameters are selected on a held-out development split through calibration sweeps, and are then fixed for test-time evaluation.

For each neighbor, the calibration rule retrieves two memory items:

$$b_j^{\mathrm{ref}} = \mathrm{RefMemory}(d_j), \qquad b_j^{\mathrm{moa}} = \mathrm{MOAMemory}(y_j).$$

These memory states are mapped to a reference-anchor reliability factor α_j^{ref} and an MOA phenotype reliability factor α_j^{moa}. Both factors are bounded and conservative: reliable evidence remains close to the raw retrieval score, while evidence associated with unreliable anchors or unstable phenotype classes is attenuated. The initial combined reliability factor is $\alpha_j = \alpha_j^{\mathrm{ref}} \alpha_j^{\mathrm{moa}}$.

To avoid calibration artifacts, the scoring rule includes several safeguards. First, if an individual reference-anchor penalty and an MOA-level penalty refer to the same known bad anchor, a no-double-penalty rule removes the redundant penalty. Second, a rank-preservation floor limits how much a highly ranked raw neighbor can be demoted solely from global reliability annotations. Third, a query-local support guard protects MOAs that receive repeated support within the raw neighborhood, since multiple locally consistent neighbors provide evidence that should not be erased by global priors. Finally, a small set of development-selected protection floors can be used to prevent systematic over-attenuation of labels whose raw evidence is empirically useful but whose global reliability annotations are pessimistic.

Let $\ell(k_j)$ be the rank-preservation floor and $u_j(q)$ be the query-local support floor. The final reliability factor is:

$$\bar{\alpha}_j = \max\{\alpha_j^{\mathrm{ref}} \alpha_j^{\mathrm{moa}}, \ell(k_j), u_j(q), \alpha_{\min}\},$$

with the no-double-penalty adjustment applied before the maximum. For nonnegative raw similarity, the calibrated similarity is:

$$\tilde{\rho}_j = \rho_j \bar{\alpha}_j.$$

For negative raw similarity, the penalty is applied in the opposite direction so that unreliable neighbors move farther down the ranking rather than being artificially improved. The references are then reranked by calibrated similarity:

$$\tilde{k}_j = \mathrm{rank}_{d_j}\big(-\tilde{\rho}_j,\ k_j,\ d_j\big).$$

Each calibrated neighbor stores the raw rank, raw similarity, reference-anchor factor, MOA reliability factor, final combined factor, calibrated similarity, calibrated rank, and a list of human-readable calibration reasons. These reasons make the calibrated neighborhood auditable: downstream agents can inspect not only which neighbors are retrieved, but also whether a neighbor was preserved, attenuated, or protected because of a specific reliability signal.

The current deterministic calibration directly uses three classes of information: reference-anchor quality, MOA-level phenotype reliability, and query-local support statistics. Source-level reliability, source-pair agreement, distance-distribution shifts, and intra-family pair ambiguity are computed and exposed to the agents as reliability context, but are not directly used as multiplicative factors in the calibrated similarity. This distinction keeps the retrieval calibration conservative. The deterministic step only changes the raw KNN ranking through local and directly attributable reliability evidence, while richer contextual annotations remain available for candidate-centric reasoning during agent inference.

C.2 Candidate-Centric Evidence Reasoning

PhenoAIR organizes inference around a shared candidate memory rather than direct dialogue between agents. This section describes the candidate memory schema, the phenotype and mechanism agent interfaces, and the update rules used to integrate their outputs. The goal is to make cross-agent influence explicit: agents do not overwrite each other's conclusions, but instead contribute structured evidence to candidate-level states.

C.2.1 Candidate Memory Schema

For a query q, PhenoAIR maintains a candidate memory

$$\mathcal{H}_q = \{(c, s_c) : c \in \mathcal{C}_q\},$$

where c is a candidate MOA and s_c is its current evidence state. Each candidate state contains five groups of fields.

Candidate identity and status. Each entry stores the candidate label c, whether it is a canonical label in the task label space $\mathcal{M}$, and a discrete status. The main statuses are:

`active`, `needs_phenotype_support`, `family_only`, `suppressed`.

An `active` candidate is visible to downstream reasoning. A `needs_phenotype_support` candidate is usually introduced by the mechanism agent and must be validated by phenotype evidence before it can be selected. A `family_only` entry records evidence for a mechanistic group when the evidence is not yet specific enough to identify a canonical MOA. Suppressed entries are retained for auditability but are not directly eligible for final prediction.

Phenotype evidence. The phenotype fields record whether Cell Painting retrieval supports or contradicts the candidate. They include a categorical support level,

`strong`, `moderate`, `weak`, `absent`, `uncertain`,

a contradiction level, the latest phenotype score, the best phenotype score observed across rounds, and short evidence notes. These fields are written only from phenotype-agent outputs.

Mechanism evidence. The mechanism fields record whether structure-side evidence supports, rejects, or cannot resolve the candidate. They include a mechanism verdict,

`support`, `reject`, `uncertain`,

a verdict strength, a suppression recommendation, and mechanism-side evidence notes. Mechanism-supported alternatives that are not already phenotype-grounded are inserted as `needs_phenotype_support`, rather than being made immediately eligible for final prediction.

Reliability context. Candidate entries may also attach reliability metadata from the externalized calibration layer, such as calibrated rank exposure, MOA reliability status, anchor risk, or pair-confusion flags. These fields are treated as context for evidence interpretation. They do not by themselves create candidates or make a candidate final-eligible.

Eligibility and audit fields. Finally, each entry stores derived fields such as whether the candidate is currently final-eligible, whether it has ever been ranked by the phenotype agent, whether it has ever received mechanism support, and why it was suppressed, promoted, or retained. These fields allow the controller and arbiter to distinguish several cases that would be collapsed by standard retrieval: a candidate may be strongly retrieved but mechanistically implausible, weakly retrieved but mechanistically supported, or visible only as part of a broader mechanistic family. This memory design enforces a candidate-centric invariant: final decisions must be grounded in the accumulated state of a candidate, rather than in a single agent output or a single retrieved neighbor.

C.2.2 Phenotype Agent

The phenotype agent P interprets Cell Painting evidence only. Its input contains the calibrated neighborhood $\tilde{\mathcal{N}}(q)$, the query-level reliability context r_q, the current candidate memory $\mathcal{H}_q$, the fixed task label set $\mathcal{M}$, and the current controller request. It is not allowed to use query-side structure, target predictions, pharmacophore matches, or other mechanism-side information as independent evidence. The phenotype agent has access to phenotype-side tools that expose local and broader views of the calibrated retrieval neighborhood. Functionally, these tools support the following operations:

$$\begin{aligned}
&\mathrm{LocalView}(q, k),\\
&\mathrm{MOADistribution}(q, k),\\
&\mathrm{CheckCandidate}(q, c, k),\\
&\mathrm{CheckCandidateSet}(q, \mathcal{C}, k),\\
&\mathrm{CheckFamilyGroup}(q, g, \mathcal{C}_g, k),\\
&\mathrm{InspectReferenceAnchor}(d).
\end{aligned}$$

These operations allow P to inspect the top calibrated neighbors, broader rank-band MOA distributions, evidence for a specific candidate, evidence for a candidate set, evidence for a mechanism-defined family group, and annotations of retrieved reference anchors. Reference-anchor information is used only to assess the reliability of phenotype evidence, not to infer mechanisms from query-side chemical structure. At each round, P returns a structured output:

$$P(q, \mathcal{H}_q, r_q) \rightarrow$$

$(\texttt{ranked_candidates}, \texttt{candidate_evidence}, \texttt{retrieval_reliability}, \texttt{flags}, \texttt{phenotype_summary})$.

The `ranked_candidates` field contains exact canonical labels from $\mathcal{M}$ and phenotype-side scores. The `candidate_evidence` field provides candidate-level support and contradiction judgments, together with the evidence scope used to make the judgment. The reliability and flag fields summarize whether the retrieval neighborhood is concentrated, ambiguous, unstable, or potentially confounded.

The phenotype agent is intentionally conservative. Weak candidates may remain visible with low score if the phenotype evidence is uncertain, while candidates with strong phenotype contradiction are marked for validation rather than automatically removed. When the controller requests a group-level check, P restricts its analysis to the specified candidate group instead of reopening the full label space.

C.2.3 Mechanism Agent

The mechanism agent M provides an independent structure-side assessment of the current candidates. Its input contains the query identity and structure, the current phenotype output, the candidate memory, and the fixed task label set. It does not inspect phenotype KNN neighborhoods directly and does not make the final prediction.

The mechanism-side tools expose complementary evidence sources:

$$\begin{aligned}
&\mathrm{CompoundInfo}(q),\\
&\mathrm{PharmacophoreSearch}(q),\\
&\mathrm{InSilicoTargets}(q),\\
&\mathrm{StructureKNN}(q, k),\\
&\mathrm{ReferenceDrugDetails}(d),\\
&\mathrm{FamilyLookup}(g).
\end{aligned}$$

These tools provide chemical properties, known or predicted targets, pharmacophore-level evidence, structural similarity to known compounds, and access to the mechanistic family ontology. At each round, M returns:

$$M(q, \mathcal{H}_q, P_q) \rightarrow$$

$$(\texttt{candidate_assessment}, \texttt{mechanistic_candidate_group}, \texttt{overall_agreement}, \texttt{mechanism_summary}).$$

For each candidate under consideration, `candidate_assessment` assigns a verdict of `support`, `reject`, or `uncertain`, together with a strength level and a suppression recommendation. A `mechanistic_candidate` is used when mechanism evidence supports a single canonical MOA. A `mechanistic_candidate_group` is used when the evidence supports a broader mechanistic family but cannot reliably select a fine-grained label. Crucially, mechanism-side proposals are not directly eligible for final selection unless they receive phenotype-side grounding. If M proposes a new canonical candidate, the memory marks it as `needs_phenotype_support`. If M proposes a family group, the controller can request a phenotype-side group inspection. This prevents structure-side evidence from bypassing the Cell Painting evidence, while still allowing mechanism reasoning to challenge or refine retrieval-derived hypotheses.

C.2.4 Candidate Memory Update Rules

Candidate memory updates are applied by deterministic host-side rules after each agent returns structured output. The agents propose evidence; the update rules decide how that evidence changes the memory state.

Phenotype updates. For every label in P's ranked candidate list, the system ensures that a candidate entry exists in $\mathcal{H}_q$. The entry records the current phenotype score, updates the best score observed across rounds, and marks whether the candidate has ever been ranked or ranked first by P. For each candidate-level evidence item, the memory updates phenotype support, phenotype contradiction, and evidence notes. If P marks a candidate as supported and no strong phenotype contradiction is present, the candidate can remain or become `active`. If the phenotype evidence strongly contradicts a candidate, the candidate is marked as requiring validation rather than immediately hard-suppressed.

Mechanism updates. For each mechanism assessment, the memory records the mechanism verdict, verdict strength, suppression recommendation, and mechanism-side evidence notes. A supported candidate receives a mechanism-support flag. A rejected candidate receives contradiction evidence, but mechanism rejection alone does not remove a phenotype-visible candidate. New single-label mechanism proposals are inserted as `needs_phenotype_support`. Family-level proposals create or update entries for the candidate labels in the family subset and attach the corresponding family name.

Promotion and suppression. The update rules distinguish soft disagreement from hard suppression. A candidate can be promoted from `needs_phenotype_support` to `active` when it receives phenotype support or when mechanism support is consistent with non-contradictory phenotype evidence. Conversely, hard suppression requires both strong phenotype contradiction and strong mechanism rejection. In simplified form, hard suppression is applied only when

$$\begin{aligned}
&\texttt{phenotype_contradiction} = \texttt{strong},\\
&\texttt{mechanism_verdict} = \texttt{reject},\\
&\texttt{mechanism_strength} = \texttt{strong},\\
&\texttt{suppression_recommendation} = \texttt{suppress}.
\end{aligned}$$

This rule prevents either agent from unilaterally deleting a plausible candidate.

Use of reliability metadata. Reliability metadata from the calibration layer can attach soft warnings to a candidate, for example if it is supported mainly by low-trust anchors or belongs to a weak phenotype class. However, reliability metadata alone does not create a candidate, suppress a candidate, or make a candidate final-eligible. It only modulates how P, M, the controller, and the arbiter interpret the candidate's evidence state.

Final eligibility. After each update, the system recomputes whether each candidate is visible to the controller and eligible for final prediction. A final-eligible candidate must be canonical, not hard-suppressed, and supported by adequate evidence in the memory. Candidates introduced only by mechanism-side family reasoning, noncanonical labels, and candidates supported only by calibration

metadata are not final-eligible. If no candidate remains adequately supported, the final gate may output `novel` rather than forcing selection from a weak known-MOA candidate.

These update rules are intentionally asymmetric. Phenotype evidence initializes and grounds candidates, mechanism evidence critiques and proposes alternatives, and reliability metadata contextualizes evidence quality. The shared memory therefore acts as the interface through which heterogeneous evidence sources are integrated without collapsing them into a single retrieval score or a free-form agent vote.

C.3 Controller-Guided Inference Refinement

PhenoAIR uses a controller to decide what evidence should be collected next and an arbiter to convert the final candidate memory into a prediction. The controller is deliberately separated from the agents: the phenotype and mechanism agents produce structured evidence, while the controller only selects the next refinement operation based on the current memory state and trajectory. This section describes the controller action space, the controller decision logic, the arbiter logic, and the offline self-evolving procedure used for open-set inference.

C.3.1 Controller Action Space

At refinement step t, the controller selects an action $a^{(t)} \in \mathcal{A}$ based on the current candidate memory $\mathcal{H}_q^{(t)}$, reliability context r_q, and trajectory summary $\tau_q^{(t)}$. The action space contains a small set of evidence-collection and termination operations. We group them into four macro families. **Phenotype expansion.** These actions request additional phenotype-side evidence when the current retrieval neighborhood is ambiguous, low-confidence, or insufficiently broad:

`EXPAND_K`, `GET_BROADER_SUMMARY`.

The former increases the retrieval depth, while the latter asks the phenotype agent to summarize broader MOA distributions beyond the current local view.

Candidate validation. These actions ask the phenotype agent to validate specific hypotheses already present in the candidate memory:

`CHECK_CANDIDATE_IN_NEIGHBORHOOD`, `CHECK_ACTIVE_CANDIDATE_SET`.

They are used when a candidate is visible but its phenotype grounding is insufficient, or when several active candidates need to be compared under the same retrieval context.

Group-level validation and mechanism reframing. When mechanism-side evidence points to a family rather than a single canonical MOA, the controller can request group-level phenotype inspection:

`CHECK_GROUP_IN_NEIGHBORHOOD`.

It can also request that the mechanism agent return a family-level assessment:

`REQUEST_GROUP_MODE_FROM_M`.

These actions are useful when the system needs to distinguish fine-grained label confusion from broader mechanistic support.

Reference inspection and termination. The controller may inspect a retrieved reference anchor when a candidate depends heavily on one or a few potentially risky anchors: `INSPECT_REFERENCE`.

Finally, it may terminate refinement: `STOP`. Termination does not itself choose the final prediction; it only passes the final memory and trajectory to the arbiter. This action space is intentionally coarse-grained. The controller does not directly edit predictions or candidate labels. It only decides which evidence view should be collected next.

C.3.2 Controller Decision Logic

The controller policy is implemented as an interpretable signal-driven function:

$$a^{(t)} = \pi\big(\mathcal{H}_q^{(t)}, r_q, \tau_q^{(t)}\big).$$

Rather than relying on free-form LLM deliberation, the controller operates over a compact set of runtime signals derived from the current phenotype output, mechanism output, candidate memory, and action history. Examples include the reliability of the phenotype neighborhood, the agreement between phenotype and mechanism evidence, whether the top phenotype candidate has been rejected by mechanism evidence, whether candidate validation has already been performed, and whether the trajectory has already explored broader phenotype evidence.

The decision logic follows a simple priority structure. First, terminal guards prevent unproductive refinement, such as repeated action cycles or exhausting the round budget. Second, if a promising but insufficiently grounded candidate or candidate group exists, the controller prioritizes validation. Third, if the phenotype evidence is weak, unstable, or based on a narrow neighborhood, the controller requests broader phenotype evidence. Fourth, if the mechanism agent suggests that evidence is better interpreted at the family level, the controller requests group-level validation or mechanism-side reframing. If no additional evidence operation is expected to change the memory state, the controller stops.

The controller can be viewed as selecting among evidence views rather than making predictions. A typical decision has the form:

$$\mathrm{condition}(\mathcal{H}_q^{(t)}, r_q, \tau_q^{(t)}) \Rightarrow \mathrm{action}(\mathrm{arguments}),$$

where the condition is defined over exposed runtime signals and the action is one of the allowed operations above. We use this rule-based form for two reasons. First, it makes the inference protocol auditable: each refinement step can be traced to a controller action and a small set of triggering signals. Second, it prevents the controller from directly introducing unsupported labels or overriding agent evidence.

Importantly, the controller is not designed as a large hand-tuned cascade. Its rules express generic refinement behaviors: validate unresolved candidates, expand unreliable phenotype evidence, inspect risky anchors, request family-level reasoning when the mechanism evidence is coarse, and stop when additional evidence is unlikely to alter the candidate memory. These behaviors are applied uniformly across queries.

C.3.3 Arbiter Logic

After the controller stops at step T, the arbiter produces the final output:

$$\hat{y}_q = \mathrm{Arbiter}\big(\mathcal{H}_q^{(T)}, \tau_q^{(T)}\big).$$

The arbiter is an LLM-based component, but it is not an evidence-gathering agent. It cannot call retrieval, structure, target-prediction, or reference inspection tools. Its input is restricted to the final candidate memory, the trajectory summary, and the structured outputs already produced by the phenotype and mechanism agents. The arbiter follows three principles. First, it selects from candidate states, not from raw retrieved labels. A candidate must be present in the memory and must have an auditable evidence history. Second, it distinguishes support from visibility. A label may appear in the retrieval neighborhood or in mechanism reasoning, but it is not final-eligible unless it receives adequate grounding in the candidate memory. Third, it respects deterministic eligibility checks after the LLM decision. The deterministic gate filters the arbiter output before evaluation. A selected canonical candidate must satisfy the following high-level requirements:

the candidate is in the task label space $\mathcal{M}$,
the candidate is not suppressed,
the candidate is supported by phenotype or mechanism evidence,
the candidate does not have unresolved strong contradiction.

If the arbiter selects a candidate that fails these checks, the system falls back to the best eligible candidate when one exists. If no canonical candidate is adequately supported, the final gate returns `novel`. Thus, `novel` is not treated as a free-form label produced by the arbiter, but as the outcome of a final eligibility test over known-MOA candidates. This separation is important for open-set MOA inference. The arbiter can reason over the full trajectory, but the final decision is still constrained by candidate memory and deterministic validity checks.

C.3.4 Self-Evolving Procedure

For open-set inference, the main challenge is deciding when weak evidence for known MOAs should be interpreted as insufficient support for any known MOA. To improve this decision boundary, we use an offline self-evolving procedure over a held-out development set. This procedure does not update LLM parameters, does not modify the phenotype or mechanism tools, and does not use the test set. It only updates two bounded artifacts: the controller rule registry and the mutable decision-heuristic section of the arbiter. The self-evolving loop operates over development trajectories generated by the current multi-agent system. Each trajectory contains the query metadata, final prediction, phenotype-agent summaries, mechanism-agent assessments, controller action history, candidate memory states, and final arbiter output. Since full trajectories are long, they are compressed into case packets that retain the key signals needed for diagnosis, such as the final candidate memory, the sequence of controller decisions, top phenotype candidates across rounds, mechanism verdicts, and whether known-MOA candidates remained adequately supported. Each evolution round has two LLM-assisted steps.

Step 1: failure pattern analysis. The first step diagnoses systematic errors. Given compact trajectory packets from failure cases and a small set of success guardrails, the LLM clusters failures by recurring root cause. The output is a structured JSON summary of failure clusters. Each cluster contains a description, example cases, a distinguishing signature over phenotype behavior, mechanism behavior, controller trajectory, memory state, and final arbitration behavior, and an assessment of which evolvable surface is implicated:

$$\{\texttt{controller_rule_gap},\ \texttt{arbiter_heuristic_gap},\ \texttt{agent_interaction_gap},\ \texttt{data_limit_or_one_off}\}.$$

Clusters that are rare or appear to be one-off data limitations are not treated as primary targets for artifact updates. Novel-case failures are analyzed separately from closed-set failures so that open-set detection errors are not confused with ordinary wrong-label errors.

Step 2: artifact proposal. For selected systematic clusters, a second LLM call proposes bounded artifact updates. The allowed outputs are:

$$\{\text{formal controller rules},\ \text{arbiter heuristic updates},\ \text{runtime signal extension requests},\ \text{retention decisions}\}.$$

Controller rules must follow a fixed schema. Each rule has a rule identifier, priority, description, conditions over whitelisted runtime signals, and one allowed controller action. Rules cannot access arbitrary hidden state or directly edit candidate memory. If the LLM identifies a useful trigger that is not exposed as a runtime signal, it must output a signal-extension request rather than an executable rule. Arbiter updates are similarly constrained. The arbiter prompt is divided into immutable sections and one mutable decision-heuristic section. The LLM may propose a replacement for the mutable heuristic section, but cannot change the arbiter role, input format, output schema, or immutable constraints. This keeps self-evolution focused on decision criteria rather than redefining the task.

Validation and materialization. Before an artifact is used, it is validated against schema and semantic constraints. Controller rules must use allowed condition roots, allowed operators, whitelisted runtime signals, and allowed actions with the required arguments. Invalid or non-executable rules are blocked and saved for inspection. Valid artifacts are materialized into the next development run.

Cross-round retention. After the first evolution rounds, the procedure also compares outcomes across rounds. It tracks which cases improved, which regressed, and which changed without affecting correctness. Subsequent LLM analysis can then recommend keeping, disabling, or revising prior artifacts. This cross-round step is used to reduce the chance that a rule which fixes one failure cluster harms existing success cases.

The self-evolving procedure can therefore be summarized as offline, schema-constrained artifact evolution:

$$\text{trajectories} \rightarrow \text{failure clusters} \rightarrow \text{validated controller/arbiter artifacts} \rightarrow \text{development rerun}.$$

The final test evaluation uses the selected artifacts fixed in advance. This makes the procedure different from test-time reflection: the system does not adapt to individual test examples using their outcomes, and all evolution is performed before final evaluation.

D Benchmark and Evaluation Details

We describe how the benchmark was constructed from raw data, how ground-truth MOA labels were curated, and how predictions are evaluated across the three settings.

D.1 Benchmark Construction

The morphological profiles in our benchmark are drawn from the `cpg16` dataset of the JUMP Cell Painting Consortium [6], with profiles aggregated across five independent laboratory sources (`source_1`, `source_2`, `source_3`, `source_5`, and `source_7`). MOA annotations are obtained from the Broad Institute Drug Repurposing Hub [10], which provides hand-curated MOA, target, and clinical-phase annotations for over 6,000 compounds. Compounds in the Cell Painting dataset are matched to MOA annotations via JCP2022 identifiers. Well-level profiles undergo MAD-based feature selection following the standard JUMP processing pipeline[5]. Profiles are aggregated to the treatment level via per-source median across replicate wells and plates, and to the global consensus via cross-source median. Per-source treatment-level profiles are retained for downstream multi-source analysis.

Drug Repurposing Hub annotations are multi-label and use a free-text format with multiple MOA strings concatenated by "|" (e.g., "*Bcr-Abl kinase inhibitor|SRC inhibitor|tyrosine kinase inhibitor*"). We parse these into atomic labels and apply two curation steps to ensure clean evaluation:

- Hierarchical redundancy removal. A compound annotated with both a specific MOA and its parent category (e.g., *EGFR inhibitor* and *tyrosine kinase inhibitor*) introduces label-set redundancy that artificially inflates multi-label evaluation. We identify and remove compounds whose label sets contain such hierarchical subsumption relationships.
- Representative MOA designation. For Top-1 evaluation and retrieval-quality analyses, we designate the first-listed label as the representative MOA. Because hierarchical redundancies are removed in the previous step, the remaining multi-label annotations are independent and non-hierarchical, making the representative-label choice inconsequential to the primary evaluation metrics.

We construct three settings:

- **MoA-Verified.** The reference set includes compounds profiled across multiple experimental sources, providing more reliable and reproducible phenotype evidence. MOA classes containing fewer than two distinct reference compounds are excluded to reduce instability from singleton classes and ensure that retrieval reflects reproducible mechanistic structure rather than isolated compounds.
- **MoA-Extended.** The reference set pools all available compounds regardless of source count, substantially expanding MOA coverage at the cost of increased heterogeneity and noisier phenotype evidence. As in MoA-Verified, only MOA classes with at least two distinct reference compounds are retained.
- **MoA-Novel.** An open-set evaluation setting constructed using the same reference set as MoA-Verified, but with query compounds whose MOA classes are absent from the reference label space. A test compound is labeled novel only if none of its curated MOA labels appears in the reference label space after synonym normalization and hierarchical redundancy removal. This setting evaluates whether the system can recognize when available evidence is insufficient to support any known mechanism and correctly classify the query as novel.

We standardize compounds by InChIKey and ensure that no identical compound appears in both reference and test sets. To assess potential analog leakage, we compute ECFP-based Tanimoto similarity between each test compound and its nearest reference compound. We screen for potential near-duplicate or same-active-series overlap by the maximum ECFP Tanimoto similarity from each test compound to the reference set. Test compounds whose nearest reference exceeds a predefined high-similarity threshold are flagged as potential analog-overlap cases. Our curated benchmarks

[5] https://cellpainting-gallery.s3.amazonaws.com/index.html#cpg0016-jump/

are smaller than the full JUMP compound collection because MOA prediction requires reliable mechanism annotations, non-redundant label curation, and sufficient reference anchors per class. We therefore prioritize reference quality and controlled evaluation over maximizing dataset size. This choice is particularly important for Cell Painting MOA prediction, where noisy or hierarchically redundant labels can inflate apparent performance or obscure whether a method is reasoning over reliable phenotypic evidence. MoA-Extended provides a larger and noisier setting with broader MOA coverage, while MoA-Verified and MoA-Novel are intentionally compact settings for controlled evaluation of cross-source reliability and open-set recognition. We view further scaling to larger MOA ontologies and additional split constructions as an important direction for future work.

Leakage controls. Several controls are applied to prevent label and structure leakage:

- **No compound overlap between reference and test sets.** Reference and test compounds (identified by JCP2022 IDs) are strictly disjoint in all evaluation settings.
- **Novel queries are mechanistically out-of-scope.** For MoA-Novel, we additionally verify that novel query MOAs do not belong to mechanistic families represented in the reference label space, preventing trivial lexical or hierarchical overlap with known classes.
- **Self-evolving uses a held-out development set.** Self-evolving refinement is performed using a fixed development split of 30 compounds that does not overlap with the final test set.
- **External tool outputs are filtered for direct leakage.** Information returned by external tools is carefully screened to remove fields that directly reveal ground-truth MOA annotations or deterministic identifier-based matches, ensuring that predictions rely on inference over retrieved evidence rather than explicit label exposure.

D.2 Evaluation Protocol

Metrics. We report two primary metrics on MoA-Verified and MoA-Extended:

- **Top-1 Accuracy.** A prediction is correct if the primary predicted MOA is semantically equivalent to the representative ground-truth MOA associated with the compound.
- **Hit Rate (Any-match).** A prediction is correct if any predicted MOA (primary or secondary) is semantically equivalent to any MOA in the ground-truth label set.

For MoA-Novel, we additionally report:

- **Overall Accuracy.** Whether the prediction, including a possible `novel` output, is correct.
- **Precision, Recall, F1 (Novel).** Novel-detection metrics treating `novel` prediction as a binary classification task over out-of-scope and in-scope queries.

LLM-as-judge for semantic equivalence. LLM-based methods may produce free-text MOA descriptions that differ lexically from canonical labels (e.g., "histone deacetylase inhibitor" vs. "HDAC inhibitor"). We therefore use an LLM-as-judge to evaluate semantic equivalence between predicted and ground-truth MOA labels for both Top-1 and Any-match evaluation. We validate judge reliability against human annotations on a randomly sampled subset of prediction-ground-truth pairs and observe 98% agreement.

D.3 Implementation and Reproducibility

LLM backbones. We evaluate PhenoAIR using GPT-5.1 (Azure OpenAI) and Claude 4 Sonnet (Anthropic). Lightweight inference configurations are used for evaluation-only utilities such as semantic judging, while agentic components use reasoning-oriented configurations for multi-step tool-based inference.

Cheminformatics tools. We use **RDKit** for chemical structure parsing, fingerprint computation (Morgan fingerprints with radius 2 and 1024 bits), Tanimoto similarity, and pharmacophore-pattern detection. SMILES strings are canonicalized with RDKit prior to all structural operations.

External knowledge resources. The Mechanism Agent and offline calibration pipeline use several public bioinformatics resources under a leakage-controlled access protocol. For reference compounds, we use external annotations to characterize retrieved anchors and support offline reliability calibration. For query/test compounds, however, we do not expose known targets, known MOA annotations, pharmacological descriptions, or database entries that could directly reveal the ground-truth mechanism. Query-side access is restricted to structure-derived inputs, including SMILES/InChIKey identifiers, basic physicochemical descriptors, structural similarity, molecular fingerprints, pharmacophore matches, and controlled in-silico target predictions. The resources are used as follows:

- **ChEMBL**[6]: bioactivity data and structurally similar analogs of query compounds.
- **DrugBank** [7]: known drug targets and pharmacological annotations.
- **PubChem** [8]: compound metadata and cross-database identifier resolution.
- **SwissTargetPrediction** [9]: in-silico target prediction based on structural similarity to known bioactive compounds.

Cell Painting features. We evaluate multiple morphological feature backbones, including CellProfiler [5], CA-MAE [18], and CellCLIP [22]. For CA-MAE and CellCLIP, we use ViT-Base encoders pretrained on raw Cell Painting images from the five selected sources, covering approximately 3.3 million images. Each model is pretrained for 100 epochs on H100 GPUs, requiring approximately 120 GPU hours. Profile preprocessing and aggregation pipelines are applied consistently across feature representations unless otherwise specified.

Inference budget. The maximum number of refinement rounds is capped at $T_{\max} = 8$. Self-evolving refinement terminates when no further rule updates are produced or when development-set performance no longer improves. Runtime statistics and API usage summaries are provided in Appendix B.1.3.

E Prompt Templates

This section provides representative prompt templates used in PhenoAIR. The prompts are designed to enforce role-specific reasoning constraints, separate phenotype and mechanism evidence, and support iterative refinement through controller-guided interaction. For brevity, low-level orchestration details and implementation-specific tool-call formatting are omitted.

Phenotype Agent Prompt

You are the Phenotype Agent (P) in a multi-agent system for drug mechanism of action (MOA) prediction. Your role is to interpret phenotype-side Cell Painting retrieval evidence. Phenotype retrieval results include calibrated reliability information in addition to raw similarity scores. Retrieved reference compounds should be treated as phenotype anchors for interpreting the local neighborhood of the query compound, rather than as direct labels.
You are allowed to:

- inspect nearest-neighbor retrieval results,
- inspect broader MOA distributions,
- verify whether specific MOAs are supported by the phenotype neighborhood,
- inspect retrieved reference compounds for phenotype interpretation.

Your job is to:

- propose ranked MOA candidates based on phenotype evidence,
- assess retrieval reliability and neighborhood consistency,
- identify ambiguity, instability, or potential confounding patterns,
- revise candidate rankings when additional phenotype evidence becomes available.

Important constraints:

[6] https://www.ebi.ac.uk/chembl/
[7] https://go.drugbank.com/
[8] https://pubchem.ncbi.nlm.nih.gov/
[9] https://www.swisstargetprediction.ch/

- Do not use structure-based evidence, pharmacophore evidence, or in-silico target predictions as independent decision criteria.
- Do not assume that the top-ranked neighbor directly determines the answer.
- Predictions must remain grounded in phenotype-side retrieval evidence and the structure of the local neighborhood.
- Retrieved reference compounds may be used to interpret phenotype patterns, but should not act as standalone mechanistic proof.
- When phenotype evidence is weak or mixed, maintain a broader candidate set rather than collapsing to a single hypothesis prematurely.
- When the controller provides a constrained candidate set, perform phenotype-side discrimination only within that set.

Candidate memory discipline:
You will receive a memory summary containing previously proposed candidates, supporting evidence, and suppression signals from prior rounds. The memory supports iterative refinement and consistency tracking, but should not be treated as a final decision oracle. Label discipline:

- Only output labels from the predefined benchmark MOA label space.
- Do not invent new labels, paraphrase canonical labels, or produce free-form mechanistic descriptions.
- Do not output `novel moa`; novelty decisions are reserved for the final Arbiter.
- When evidence is insufficient, lower confidence and retain uncertainty rather than generating unsupported labels.

Mechanism Agent Prompt

You are the Mechanism Agent (M) in a multi-agent system for drug mechanism of action (MOA) prediction. Your role is to evaluate phenotype-derived MOA candidates using structure-driven mechanistic evidence. You are allowed to:

- inspect compound structure and physicochemical properties,
- inspect pharmacophore patterns,
- inspect in-silico target predictions,
- inspect structurally similar compounds and their mechanistic annotations,
- inspect known reference compounds and associated mechanism information.

You are instructed to:

- evaluate MOA candidates proposed by the Phenotype Agent,
- determine whether mechanism-side evidence supports, rejects, or remains uncertain about each candidate,
- summarize overall agreement with phenotype-side reasoning,
- determine the appropriate granularity of mechanistic conclusions.

Mechanistic conclusions may operate at different levels of specificity:

- If the evidence strongly supports a specific canonical MOA label, return a single-candidate conclusion.
- If the evidence only supports a mechanistic family or a small subset of related labels, return a constrained candidate group rather than forcing a fine-grained label decision.

Important constraints:

- Do not inspect raw Cell Painting retrieval results or phenotype neighborhoods directly.
- Do not output a full ranking over all MOA labels.
- Do not act as the final decision-maker.
- Treat mechanistic evidence as complementary evidence rather than a standalone oracle overriding phenotype observations.
- If mechanism-side evidence strongly supports an allowed MOA absent from the current phenotype candidate list, you may still propose it for phenotype-side validation.

- Very weak structural similarity alone should not override stronger target-level or pharmacological evidence.

You receive a memory summary containing previously proposed candidates, supporting evidence, and suppression signals from prior rounds. The memory is intended to support iterative refinement and consistency tracking rather than directly determine final predictions.
Label discipline:

- Only output labels from the predefined benchmark MOA label space.
- Do not invent new labels, paraphrase canonical labels, or generate free-form mechanistic descriptions as candidate labels.
- If evidence supports only a broader mechanistic family, map the evidence to the corresponding canonical benchmark labels rather than introducing new family names.
- Do not output `novel moa`; novelty decisions are reserved for the final Arbiter.

Arbiter Agent Prompt

You are the Arbiter Agent (A) in a multi-agent system for drug mechanism of action (MOA) prediction. Your role is to make the final decision based on structured outputs from the Phenotype Agent, the Mechanism Agent, candidate memory, and controller history. Your main value is meta-level confidence calibration and generating an interpretable final rationale, rather than redoing the phenotype or mechanism analysis from scratch.
You are instructed to:

- determine the final predicted MOA,
- calibrate confidence based on the full refinement trajectory,
- include alternative candidates when evidence remains ambiguous,
- output `novel moa` when no task-allowed canonical label is adequately supported,
- provide a concise rationale that downstream researchers can audit.

Confidence calibration:

- Assign high confidence when phenotype retrieval is reliable, mechanism evidence supports the selected candidate, and the candidate is stable across refinement rounds.
- Assign medium confidence when evidence improves after refinement but remains partially ambiguous, when mechanism evidence is uncertain, or when mechanism evidence supports only a candidate group rather than a specific label.
- Assign low confidence when candidate rankings oscillate, retrieval reliability remains limited, mechanism and phenotype evidence remain in conflict, or the final choice relies on marginal support.

Mechanistic resolution:

- If the Mechanism Agent supports a specific canonical MOA, treat this as strong evidence only when it is not contradicted by phenotype evidence.
- If the Mechanism Agent supports only a candidate group or mechanistic family, treat this as weaker evidence and do not upgrade confidence based on mechanism evidence alone.
- If mechanism evidence suggests a non-canonical or overly fine-grained label, use it only as supporting evidence for the nearest canonical candidate already present in memory.

Final-choice rules:

- Candidate memory records which hypotheses appeared during refinement and which were supported, challenged, or suppressed. It should not be treated as a ranking by itself.
- Do not choose solely by phenotype score when the phenotype trajectory is weak, sparse, or non-specific.
- Do not choose solely by mechanism support when a phenotype-supported candidate is stable and not specifically rejected.
- Prefer a mechanism-supported candidate over a phenotype-favored candidate only when mechanism evidence is specific, the candidate is canonical, and phenotype evidence does not strongly contradict it.
- For same-family candidate labels, prefer the label most specifically supported by mechanism evidence only when phenotype evidence cannot distinguish among them.

- Output exactly `novel moa` when the integrated evidence indicates that no task-allowed canonical candidate is sufficiently supported.

Important constraints:

- Do not reinterpret the task from scratch.
- Do not use external tools.
- Do not invent candidates that were not introduced by the Phenotype Agent, Mechanism Agent, controller, or candidate memory.
- The final `primary_moa` must be either a task-allowed canonical label or exactly `novel moa`.
- Do not output `ABSTAIN`, `no_supported_moa`, or an empty string as the final label.
- Do not select archived or hard-suppressed candidates unless no viable canonical alternative remains.

Before submitting the final prediction, perform a brief decision audit:

- Is the selected label canonical and not hard-suppressed?
- Was it supported by phenotype evidence, mechanism evidence, or both?
- Was it strongly rejected by either agent, and if so, why is it still selected?
- Are stronger alternatives suppressed or unsupported?
- Is the evidence resolved at the canonical-label level, or only at the family level?

Use this audit only for calibration and final choice. Do not redo the Phenotype Agent or Mechanism Agent analysis from scratch.

Algorithm 1 PhenoAIR Inference with Global Reliability Calibration

Require: Reference set $\mathcal{R}$ with Cell Painting profiles and MOA labels; query compound q with Cell Painting profile and structure; canonical MOA label space $\mathcal{M}$; maximum refinement rounds $T_{\max}$

Ensure: Final prediction $\hat{y}_q \in \mathcal{M} \cup \{\texttt{novel}\}$

Global pre-computation before query-time inference

Build source-specific reference profiles from well-level features.

Construct reference reliability memory $\mathcal{B}$:
use offline LLM-assisted code execution to analyze source-level, MOA-level, reference-anchor-level, and MOA-pair reliability.

Construct mechanistic family ontology $\mathcal{G}$:
group canonical MOA labels into coarse mechanistic families using external biomedical evidence.

Define a fixed calibrated retrieval rule $\mathrm{Calib}(\cdot)$ using the reference reliability memory $\mathcal{B}$.

▷ The above steps are global and are not repeated for each test query.

Query-time inference for query q

Retrieve raw phenotype neighborhood $\mathcal{N}(q)$ from the reference set.

Compute calibrated neighborhood

$$\tilde{\mathcal{N}}(q) \leftarrow \mathrm{Calib}(\mathcal{N}(q), \mathcal{B}).$$

Build query-level reliability context r_q from calibrated retrieval and reference memory.

Initialize candidate memory $\mathcal{H}_q^{(0)} \leftarrow \emptyset$.

Initialize trajectory log $\tau_q^{(0)} \leftarrow \emptyset$.

for $t = 0, 1, \ldots, T_{\max}$ **do**

  Run phenotype agent:

$$P_q^{(t)} \leftarrow P(\tilde{\mathcal{N}}(q), r_q, \mathcal{H}_q^{(t)}, \tau_q^{(t)}).$$

  Update candidate memory with phenotype evidence:

$$\mathcal{H}_q^{(t+\frac{1}{3})} \leftarrow \mathrm{UpdateP}(\mathcal{H}_q^{(t)}, P_q^{(t)}).$$

  Run mechanism agent:

$$M_q^{(t)} \leftarrow M(q, \mathcal{G}, \mathcal{H}_q^{(t+\frac{1}{3})}, P_q^{(t)}).$$

  Update candidate memory with mechanism evidence:

$$\mathcal{H}_q^{(t+\frac{2}{3})} \leftarrow \mathrm{UpdateM}(\mathcal{H}_q^{(t+\frac{1}{3})}, M_q^{(t)}).$$

  Select controller action:

$$a^{(t)} \leftarrow \pi(\mathcal{H}_q^{(t+\frac{2}{3})}, r_q, \tau_q^{(t)}).$$

  Append $(P_q^{(t)}, M_q^{(t)}, a^{(t)}, \mathcal{H}_q^{(t+\frac{2}{3})})$ to trajectory log τ_q.

  if $a^{(t)} = \texttt{STOP}$ **then**

    $\mathcal{H}_q^{(T)} \leftarrow \mathcal{H}_q^{(t+\frac{2}{3})}$.

    break

  else

    Execute controller action $a^{(t)}$ to obtain additional evidence view.

    Examples: expand phenotype neighborhood, validate candidate, validate candidate group, inspect reference anchor, or request mechanism group mode.

    Update available evidence context for the next round.

    $\mathcal{H}_q^{(t+1)} \leftarrow \mathcal{H}_q^{(t+\frac{2}{3})}$.

  end if

end for

Run arbiter over final memory and trajectory:

$$y_q^{\mathrm{arb}} \leftarrow A(\mathcal{H}_q^{(T)}, \tau_q^{(T)}).$$

Apply deterministic final eligibility gate:

$$\hat{y}_q \leftarrow \mathrm{FinalGate}(y_q^{\mathrm{arb}}, \mathcal{H}_q^{(T)}).$$

return $\hat{y}_q$.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the paper's scope as Cell Painting-based MOA prediction and frame the main contributions as task reformulation, the multi-agent framework, and a novel benchmark. The claims are supported by the experimental results and ablation studies reported in the main paper.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 4.2 discusses the limitations of the proposed approach through an analysis of model failure modes.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper does not include theoretical results or formal proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Sections 3 and 4 describe the benchmark construction, baselines, evaluation metrics, implementation details, and ablation settings needed to reproduce the main experimental results.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The experiments are based on publicly available Cell Painting profiles and curated MOA annotations, and the paper describes the benchmark construction and evaluation protocol. We do not provide a full code release at submission time.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4 describes the experimental settings, including benchmark splits, baselines, evaluation metrics, LLM configurations, and ablation settings needed to interpret the results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The main results report averages over multiple runs where stochastic LLM-based methods are used, while deterministic retrieval baselines are reported once; the evaluation protocol and sources of variability are described in Section 4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section 4 and Appendix reports the compute environment and approximate runtime used for the experiments, including LLM inference settings and retrieval/evaluation workloads.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research uses publicly available biological profiling data and curated annotations, does not involve human subjects or private personal data, and conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: The paper discusses broader impacts, including potential benefits for drug mechanism discovery and risks from over-interpreting model-generated mechanistic hypotheses without experimental validation.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper does not release high-risk models, scraped datasets, or other assets requiring special misuse safeguards.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The paper uses publicly available datasets and existing tools/models, and credits the original sources with citations and usage descriptions in the data and implementation sections.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The paper introduces curated benchmark splits derived from public datasets, and documents their construction, evaluation protocol, and intended use in the benchmark section.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing experiments or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: LLMs are a core component of the proposed method, and the paper describes their roles in phenotype analysis, mechanism evaluation, and final arbitration, as well as the model configurations and prompting/evaluation protocol.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.