The Thinking Microscope: A Reinforcement Learning Framework for the Co-optimization of Computational and Generative Data Sparsification in Metagenomics

Arvid E. Gollwitzer[1,3*], Jakub Sienkiewicz[1*], Deepak Subramanian[3], Isaac Tucker[3], Joël Lindegger[1], Serghei Mangul[2], Onur Mutlu[1*], Giovanni Traverso[3*]

[1]Department of Information Technology and Electrical Engineering, ETH Zürich, Gloriastrasse 35, Zürich, 8092, Switzerland.
[2]Department of Clinical Pharmacy, University of Southern California, Los Angeles, CA, 90089, USA.
[3]Massachusetts Institute of Technology, Cambridge, MA, 02139, USA.

*Corresponding author(s). E-mail(s): arvidg@mit.edu; omutlu@ethz.ch; cgt20@mit.edu;

Abstract

The clinical utility of high-throughput sequencing is limited by a fundamental trade-off between computational cost and the accuracy of biomarker detection. We address this by formalizing analysis as a Partially Observable Markov Decision Process (POMDP). We prove that continuous strategy spaces can admit solutions with superior accuracy-cost profiles compared to approaches relying on a finite, discrete library of tools. We provide a theoretical foundation for solving this POMDP, including a discussion of policy optimization convergence and a martingale-based framework for error control during sequential testing. This allows for statistical validity despite the policy's adaptive data transformations. We instantiate this theory in *HighClass*, a deep reinforcement learning system where a single agent learns a unified policy to co-optimize computational sparsification, generative data sparsification, and optimal stopping. Validated on a diagnostic biomarker task (colorectal cancer screening), HighClass achieves an 8-fold speedup for positive cases and a 60-fold speedup for negative controls—often by analyzing less than 10% of the data—without loss of analytical accuracy. Its performance traces a Pareto-optimal frontier superior to a comprehensive

set of competitive baselines, empirically supporting our theoretical results. This work provides a mathematical foundation for adaptive biological data analysis, enabling the deployment of high-throughput sequencing as a time-critical tool for science and medicine.

Keywords: Metagenomics, Computational sparsification, Generative data sparsification, Early termination, Reinforcement learning, Bioinformatics, Precision medicine

1 Introduction

The exponential growth of high-throughput sequencing is revolutionizing precision medicine, enabling deep insights into the molecular basis of health and disease [1, 2, 24]. In metagenomics, this has unlocked the study of complex microbial communities from which we can derive critical biomarkers for diagnostics, prognostics, and treatment response [3]. Yet the sheer volume of sequencing data presents a fundamental bottleneck: the computational cost of analysis often conflicts with the need for timely and accurate results, a trade-off that is particularly acute in clinical diagnostics [4, 5, 17, 19, 22]. This challenge calls for the development of new computational paradigms to manage data complexity and accelerate analysis [18, 25].

Recent advances have focused on adaptive frameworks that dynamically allocate resources to optimize the trade-off between speed and accuracy [20, 21, 23]. Theoretical work has formalized this problem using the language of Partially Observable Markov Decision Processes (POMDPs), providing a rigorous foundation for sequential decision-making under uncertainty. Such studies suggest that policies optimized over continuous strategy spaces can outperform traditional approaches that rely on a discrete library of tools [26].

This work introduces a framework for adaptive metagenomic analysis, grounded in the theory of Partially Observable Markov Decision Processes (POMDPs) and continuous optimization. We formalize the trade-off between computational cost and analytical accuracy as a constrained optimization problem over a continuous space of data transformation strategies. We prove that this continuous formulation can admit superior solutions compared to a discrete approximation, with explicit bounds on the optimality gap.

We implement this concept through a deep reinforcement learning [13] system that learns to navigate a continuous manifold of analytical strategies. The system co-optimizes three interconnected components: (1) **Computational Sparsification**: adaptive selection of analysis algorithms based on the belief state; (2) **Generative Data Sparsification**: synthesis of optimal data transformation filters from a learned continuous latent space; and (3) **Optimal Stopping**: early termination based on sequential hypothesis testing with controlled error rates.

The theoretical foundation rests on three main results. First, we prove that the continuous strategy space, parameterized through a variational autoencoder, contains ϵ-optimal solutions for any discrete approximation with finitely many strategies. Second, we establish learning guarantees for our policy optimization algorithm, ensuring convergence to a locally optimal policy. Third, we derive finite-sample guarantees for our sequential testing procedure that maintain specified Type I and Type II error rates despite the adaptive, data-dependent analysis.

We validate our framework with **HighClass**, an example implementation of our approach. Through comprehensive experiments on a challenging diagnostic task (i.e., colorectal cancer screening), we show that the learned policies achieve order-of-magnitude computational savings while maintaining statistical power. This work establishes a mathematical foundation for adaptive biological data analysis, with potential implications for other domains that require efficient extraction of signal from high-dimensional sequential data.

2 Theoretical Framework

2.1 Modeling Choice: A Partially Observable Markov Decision Process

A key choice in our framework is the Partially Observable Markov Decision Process (POMDP) as the foundational model. A simpler approach might frame the problem as a series of myopic contextual bandit tasks, where the agent picks the best filter for the current data batch. Yet this fails to capture the long-term consequences of actions. An aggressive filtering decision made early on might offer a short-term speedup but irreversibly discard subtle information critical for a correct classification later. The POMDP framework explicitly models this trade-off between immediate reward and future value, making it an appropriate choice for strategic, long-horizon planning.

An alternative would be a fully observable MDP. In this case, we would assume that the true underlying biological state is known at each step. This is not a realistic assumption in (meta)genomics. The true microbial composition or the true signal-to-noise ratio of a sample is never directly accessible; it can only be inferred from the noisy, incomplete data observed in each sequencing batch. The state is inherently partially observed, making the POMDP the most appropriate and rigorous choice for this domain.

2.2 Problem Formulation as a Partially Observable Markov Decision Process

We formalize adaptive metagenomic analysis as a POMDP, an expressive model for sequential decision-making under uncertainty [11, 12]. A POMDP is defined by the tuple $\mathcal{P} = (\mathcal{S}, \mathcal{A}, T, R, \Omega, O, \gamma, \rho_0)$, where:

- $\mathcal{S} = \mathcal{G} \times \Xi \times [0, 1]$ is the state space, which is unobserved by the agent. It combines the true biological state ($\mathcal{G}$), technical parameters (Ξ), and analysis progress.
- $\mathcal{A} = \mathcal{A}_d \times \mathcal{Z}$ is the hybrid action space, combining a discrete choice of analysis tool with a continuous parameter vector $z \in \mathcal{Z} \subseteq \mathbb{R}^d$ for that tool.
- $T : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{S})$ is the state transition probability kernel.
- $R : \mathcal{S} \times \mathcal{A} \to [-R_{\max}, R_{\max}]$ is the bounded reward function.
- $\Omega = \mathbb{R}^{n_o}$ is the observation space. The agent observes $o \in \Omega$, which provides partial information about the state $s \in \mathcal{S}$.
- $O : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\Omega)$ is the observation probability kernel.
- $\gamma \in (0, 1)$ is the discount factor.
- $\rho_0 \in \mathcal{P}(\mathcal{S})$ is the initial state distribution.

Assumption 1 (Regularity Conditions). Our theoretical analysis relies on the following standard regularity conditions.

(A1) **Lipschitz Continuity in Latent Space**: For any fixed discrete action, the non-terminal reward function $R(s, a)$ is L_R-Lipschitz with respect to the continuous action component $z \in \mathcal{Z}$. This is a direct consequence of using a neural network decoder (see Appendix A), which is Lipschitz, to generate the sparsification pattern.

(A2) **Bounded Densities**: The transition and observation kernels have densities that are bounded.

(A3) **Identifiability**: For any two distinct states $s_1 \neq s_2$, there exists a sequence of actions that can produce statistically distinguishable observation sequences. This ensures the state is, in principle, discoverable from observations.

Definition 1 (State Space). The state $s_t = (g_t, \xi_t, \eta_t) \in \mathcal{S}$ is continuous and encodes:

- $g_t \in \mathcal{G}$: True genomic composition, represented as a probability vector over taxa, $g_t \in \Delta^{|\mathcal{T}|-1}$.
- $\xi_t \in \Xi$: Technical factors such as sequencing error model parameters.
- $\eta_t \in [0, 1]$: Fraction of the total data stream processed.

The transition dynamics model the evolution of this state. For in-vitro biomarker detection/measurement tasks, we assume the genomic composition is static ($T_{\text{bio}}(g_{t+1}|g_t) = \mathbb{I}[g_{t+1} = g_t]$). We recognize this is a simplification; extending the model to non-stationary biological states for longitudinal monitoring is an important direction for future work.

2.3 The Continuous Strategy Space and Its Advantage

We hypothesize that the continuous generation of analytical strategies is superior to selecting from a finite library.

Definition 2 (Strategy Spaces). A data transformation strategy is defined by a binary "sparsification pattern" $p \in \{0, 1\}^\ell$ that filters a sequence.

- A **discrete strategy space** $\mathcal{P}_n = \{p_1, \ldots, p_n\}$ is a finite library of n fixed patterns.
- A **continuous strategy space** is the image of a mapping $\mathcal{M}_\theta : \mathcal{Z} \to \{0, 1\}^\ell$ from a compact latent space $\mathcal{Z} \subset \mathbb{R}^d$ to the space of all possible patterns, parameterized by the decoder of a VAE [14] with weights θ.

Theorem 1 (Continuous Superiority). Let Π_n be the set of policies restricted to a discrete strategy space $\mathcal{P}_n$, and let Π_c be policies utilizing a continuous strategy space generated by $\mathcal{M}_\theta$. Assume the manifold of useful patterns has an intrinsic dimension $d_{int} \leq d$. Then for any finite n, there exists a distribution $\mathcal{D}$ over POMDP instances such that:

$$\mathbb{E}_{\mathcal{P}\sim\mathcal{D}}\left[\sup_{\pi\in\Pi_c} V^\pi(\rho_0) - \sup_{\pi\in\Pi_n} V^\pi(\rho_0)\right] \geq c_d \cdot n^{-1/d_{int}} - \epsilon_{\mathcal{M}}$$

where $V^\pi(\rho_0)$ is the expected value of policy π, $c_d > 0$ is a dimension-dependent constant, and $\epsilon_{\mathcal{M}}$ is the VAE's approximation error.

Proof Sketch. The full proof is in Appendix A. The key idea is that for any n-point discretization of a continuous space, "holes" must exist. We construct a POMDP instance where the reward function is sensitive to the precise structure of the sparsification pattern, creating a complex reward landscape over the latent space $\mathcal{Z}$. By placing higher reward in a region of this landscape that falls within a hole of the discrete library's coverage, the continuous policy achieves higher value. The rate $n^{-1/d_{int}}$

emerges from sphere packing bounds, which quantify the largest possible "hole" in any n-point discretization of a space with intrinsic dimension d_{int}. The constant c_d depends on the problem's Lipschitz constants (which can be bounded, see Appendix), while the VAE error $\epsilon_{\mathcal{M}}$ is empirically minimized during training. □

Relevance and Implications. This theorem proves a fundamental limitation of any analytical approach based on a finite toolkit. While the proof is an existence proof, its premise—a "peaky" reward landscape—is motivated by specific challenges in metagenomics. For example, a sample might be contaminated with a unique viral element or a specific primer-dimer artifact whose sequence is highly repetitive. An optimal filter would need to be precisely tailored to the repeat structure of that artifact to remove it without attenuating the nearby microbial signal. A slightly different filter might fail to remove the artifact or, worse, might begin to degrade the signal of interest. This creates a scenario where a filter can be highly sensitive to small changes in its structure. This motivates our theoretical exploration of continuous strategy generation. A finite library is unlikely to contain these bespoke optimal patterns for all samples. The continuous space allows the agent to generate them on demand. The concept of *intrinsic dimensionality* is one of our key contributions: even if the latent space is high-dimensional (e.g., $d = 16$), the manifold of patterns that are truly distinct and useful may have a much lower dimension, making the optimality gap practically significant. Theorem 1 suggests that a performance gap can exist for any finite library size n.

This approach may have implications beyond metagenomics. Any adaptive system operating in a high-dimensional strategy space—from robotic control to financial trading—could benefit from continuous parameterization.

2.4 Policy Optimization in Hybrid Action Spaces

The agent's policy, $\pi_\theta(a|h)$, maps the history of observations $h_t = (o_1, a_1, ..., o_t)$ to a distribution over actions. Our hybrid action space requires a factorized policy:

$$\pi_\theta(a|h_t) = \pi_{\theta,d}(a_d|h_t) \cdot \pi_{\theta,c}(a_c|h_t, a_d)$$

We use Proximal Policy Optimization (PPO) to train the policy network [10]. While a full theoretical analysis in the POMDP setting is complex, established convergence guarantees for PPO in simpler settings provide confidence in its stability. Under standard assumptions, such as the policy being parameterized by a neural network with bounded gradients and the satisfaction of our regularity conditions (A1-A3), the algorithm is expected to find a stationary point (a local optimum or saddle point) of the value function. The key insight for our application is that the PPO surrogate objective applies directly to our factorized policy, as it treats the action as a single sample from the compound policy distribution.

Theorem 2 (Convergence of PPO). Under the regularity conditions specified in Assumption 1 (A1-A3), the PPO algorithm is guaranteed to find a stationary point of the value function, where the gradient of the policy performance objective is zero.

The key insight for our application is that the PPO surrogate objective applies directly to our factorized policy, as it treats the action as a single sample from the

compound policy distribution. The proof is a standard result from the literature, sketched in Appendix A for completeness.

2.5 Rigorous Sequential Testing with Adaptive Policies

A core function of HighClass is to terminate analysis early. This is framed as a sequential hypothesis test [15] between H_0 (e.g., biomarker absent) and H_1 (e.g., biomarker present). A key challenge is that the policy π changes the data transformation at each step, which can violate the IID assumption of classical tests. We address this using martingale theory to provide error control.

Definition 3 (Transformed Likelihood Ratio). For a data batch X_t transformed by pattern p_{t-1} (generated from latent vector z_{t-1}), the log-likelihood ratio (LLR) is:

$$\ell_t(X_t, p_{t-1}) = \log \frac{f_1(\phi_{p_{t-1}}(X_i))}{f_0(\phi_{p_{t-1}}(X_i))}$$

The cumulative LLR is $L_t = \sum_{i=1}^{t} \ell_i$. The test stops when L_t crosses a boundary A or B.

Assumption 2 (Information Preservation). For any latent vector $z \in \mathcal{Z}$, the pattern $p = \mathcal{M}_\theta(z)$ it generates preserves a minimal amount of information: $D_{KL}(f_0(\phi_p(\cdot)) \| f_1(\phi_p(\cdot))) \geq \kappa > 0$. This is a constraint on the VAE to prevent it from generating null patterns that destroy all signal. This is enforced via the VAE reconstruction loss.

Theorem 3 (Error Control via Martingale Correction). For a sequential test with boundaries $A = \log(\frac{1-\beta}{\alpha})$ and $B = \log(\frac{\beta}{1-\alpha})$, the adaptive testing procedure, when using a learned estimator for the one-step bias, maintains the following error guarantees:

$$\mathbb{P}_{H_0}(\text{reject } H_0) \leq \alpha \cdot C_{\pi,\psi}$$
$$\mathbb{P}_{H_1}(\text{reject } H_1) \leq \beta \cdot C'_{\pi,\psi}$$

where $C_{\pi,\psi}$ and $C'_{\pi,\psi}$ are policy- and estimator-dependent correction factors that can be empirically verified to be close to 1 for a well-trained system. The full proof is in Appendix A.

Proof The sequence of likelihood ratios $\{e^{L_t}\}$ is not a martingale under adaptation. Yet we define a "correction process" $\{C_t = \sum_{i=1}^{t} c_i\}$ and show that the process $\{\exp(L_t - C_t)\}$ is a true martingale. By applying the Optional Stopping Theorem to this corrected process, we can bound the error probabilities. The factors C_π and C'_π emerge naturally from this bound. □

Relevance and Implications. This result provides a practical and verifiable error guarantee. The correction factors can be estimated online during analysis, providing a running measure of how much the adaptivity is "costing" in statistical confidence. A well-trained policy will learn to keep the one-step biases small on average, resulting in correction factors close to 1 and thus preserving the desired error rates α and β. This allows statistical control to be an objective for the reinforcement learning agent,

integrating it with policy optimization. Yet the practical estimation of the correction term is a significant challenge in itself, as we detail in Appendix A. This result provides a practical framework for adaptive testing.

2.6 Belief State Representation via Recurrent Networks

The optimal policy in a POMDP depends on the belief state $b_t = \mathbb{P}(s_t|h_t)$. For our continuous state space, this is intractable to compute. We follow standard practice in deep RL and use the hidden state of a recurrent neural network (RNN) as a sufficient statistic of the history. The policy and value functions are conditioned on this compact representation. Appendix B discusses the theoretical motivation for this choice, referencing the universal approximation capabilities of RNNs for belief state updates in simpler, finite POMDPs.

3 Results

To validate our theoretical framework, we performed computational experiments to determine if HighClass can reduce computational cost without sacrificing accuracy on realistic data streams. We specifically investigated whether its generative capability offers an advantage over fixed analytical libraries, as predicted by Theorem 1, and if the learned policy is robust to clinical confounders and adaptable to new data domains.

3.1 HighClass enables rapid, resource-efficient analysis without sacrificing accuracy

We first evaluate HighClass on a challenging diagnostic biomarker task: colorectal cancer (CRC) screening using fecal metagenomes from the Wirbel et al. dataset (n=122 samples; 59 CRC, 63 controls) [6]. To ensure our primary results reflect real-world conditions, all analyses reported in the main figures are performed using our most robust agent, which features a recurrent neural network architecture, operating on non-randomized (i.e., non-IID) data streams straight from the sequencer (see Methods).

HighClass achieves biomarker detection accuracy comparable to standard "profile-then-classify" pipelines (detailed in Section 3.3) while dramatically reducing computational cost. Measured by a "Cohort Turnaround Time Ratio," which accounts for the cost of re-analyzing ambiguous samples (see Methods), HighClass processes the entire test cohort 14 times faster than an exhaustive analysis. This aggregate gain is particularly pronounced for biomarker-negative samples, where the policy's early termination enables a median speedup of 60-fold. Biomarker-positive samples are also identified efficiently, with a median 8-fold speedup (Fig. 1c). These accelerations are a result of the RL controller's learned policy, which uses an early termination mechanism to analyze a median of only 12% of the total reads for positive cases and just 2% for negative controls (Fig. 1d). The end-to-end detection performance is visualized in Figure 1a,b.

Fig. 1 HighClass Performance, Architecture, and RL Control Unit. a,b, Receiver Operating Characteristic (ROC) and Precision-Recall (PR) curves for HighClass versus the exhaustive baseline on the CRC screening task. Shaded areas represent 95% confidence intervals computed via 1000 bootstrap replicates from the aggregated predictions of a 5-fold nested cross-validation. **c,d**, Distributions of wall-clock time speedup and percentage of dataset analyzed for HighClass relative to the baseline, stratified by final decision (Biomarker Positive n=59, Biomarker Negative n=63). Boxplots show median, interquartile range (IQR), and 1.5x IQR whiskers. **e**, A detailed data-flow diagram of the HighClass control loop for a single mini-batch of reads. The agent receives a history representation from its RNN, h_t, and its policy π outputs a compound action a_t. This action includes a termination decision and, if continuing, a latent vector z_t that is decoded by a pre-trained VAE into a sparsification pattern p_t. The pattern is applied to the next data batch, which is processed by the chosen analytical stage. This produces a new observation o_{t+1} and a reward signal R_{t+1}, which are used to update the RNN state to h_{t+1}. The agent's objective is to maximize cumulative future rewards, thereby co-optimizing speed, data complexity, and accuracy.

3.2 Comparison with State-of-the-Art Profiling Tools

Before evaluating the goal-directed, early-termination capabilities of our framework, we first assess the performance of HighClass as a stand-alone, complete metagenomic profiler. This initial analysis establishes the baseline analytical quality of our system when tasked with profiling an entire sample, without applying any early-stopping criteria (Table 1). We compare it against state-of-the-art tools: Kraken2 [7], Metalign [8], and MetaTrinity [9]. We evaluate performance on taxonomic classification accuracy (F1 score), abundance estimation error (L1 norm error), and computational resource usage (wall-clock time and peak RSS).

3.3 Goal-Directed Biomarker Detection

Having established the proficiency of our core profiler, we now evaluate the full, goal-directed HighClass system on the primary clinical task: biomarker-based classification. Our primary benchmark is the current standard paradigm in metagenomic analysis: a two-stage "profile-then-classify" pipeline. In this approach, a dedicated taxonomic

Table 1 Benchmarking against state-of-the-art taxonomic profilers for full sample analysis.

Tool	F1 Score (Species)	L1 Norm Error (Abundance)	Wall-Clock Time (h)	Peak RSS (GB)
CAMI Low Diversity Dataset				
HighClass (full profile)	0.25	1.18	0.50	80.0
Kraken2	0.02	1.78	0.10	25.0
Metalign	0.24	1.64	2.50	120.0
MetaTrinity	0.23	1.14	0.65	125.0
CAMI Medium Diversity Dataset				
HighClass (full profile)	0.27	1.15	0.60	85.0
Kraken2	0.02	1.67	0.10	28.0
Metalign	0.23	1.63	2.50	120.0
MetaTrinity	0.26	1.11	0.76	125.0
CAMI High Diversity Dataset				
HighClass (full profile)	0.32	0.95	0.70	90.0
Kraken2	0.00	1.83	0.10	25.0
Metalign	0.28	1.49	2.50	120.0
MetaTrinity	0.31	0.90	0.89	125.0
Real Dataset: MMSA Human Gut (SRX055381)				
HighClass (full profile)	0.13	1.32	0.50	75.0
Kraken2	0.08	0.76	0.20	45.0
Metalign	0.12	1.03	2.50	110.0
MetaTrinity	0.11	1.28	0.66	115.0
Real Dataset: Human Fecal Sample (SRX13554335)				
HighClass (full profile)	0.09	1.37	0.45	70.0
Kraken2	0.08	0.41	0.30	38.0
Metalign	0.08	1.33	2.50	105.0
MetaTrinity	0.06	1.33	0.57	110.0
Real Dataset: NIBSC GutMix (SRX8063904)				
HighClass (full profile)	0.21	1.19	0.60	80.0
Kraken2	0.10	0.53	0.55	65.0
Metalign	0.20	0.86	2.50	115.0
MetaTrinity	0.11	1.15	0.74	125.0

profiler first generates a static feature vector (i.e., relative abundances) from the entire sample. This vector is then used as input to a separate, high-performance machine learning classifier to render a decision. We implement this paradigm by creating a comprehensive baseline suite, pairing three state-of-the-art profilers (Kraken2, Metalign, MetaTrinity) with two accurate, state-of-the-art classifiers (Random Forest, XGBoost).

We show the results in Table 2. HighClass achieves an end-to-end accuracy that is statistically non-inferior to the strongest baseline combination (AUC-ROC of 0.93

vs. 0.92). This demonstrates that our adaptive, end-to-end framework sacrifices no analytical power while enabling the significant computational savings reported above.

Table 2 Biomarker detection with HighClass vs. standard profile-then-classify pipelines.

Profiler (Feature Extractor)	Classifier	AUC-ROC (95% CI)	Mean Speedup	Speedup (Negatives)
Kraken2	Random Forest	0.88 (0.85 - 0.91)	1x	1x
	XGBoost	0.89 (0.86 - 0.92)	1x	1x
Metalign	Random Forest	0.90 (0.87 - 0.93)	1x	1x
	XGBoost	0.91 (0.88 - 0.94)	1x	1x
MetaTrinity	Random Forest	0.91 (0.88 - 0.94)	1x	1x
	XGBoost	0.92 (0.90 - 0.94)	1x	1x
HighClass (end-to-end)	**-**	**0.93 (0.91 - 0.95)**	**∼14x**	**∼60x**

3.4 Evaluating the generative policy against discrete tool selection

We propose that the space of possible analytical filters constitutes a continuous, high-dimensional strategy space. Any finite library of pre-defined filters, no matter how large, represents a discrete and therefore incomplete sampling of this space. An optimal strategy for a given analytical state may not exist in this library but can be generated by an agent that navigates the continuous latent space of all filters. This hypothesis is formalized in Theorem 1.

To test this hypothesis, we benchmark the full, generative HighClass agent against a series of powerful non-generative RL agents. These agents are identical to HighClass in every respect (recurrent architecture, state space, training protocol) but are limited to a discrete action space, allowing them to select only from fixed libraries of the top-performing, pre-defined sparsification patterns of varying sizes (N=10, N=50, and N=100).

The results are consistent with our theory (Fig. 2). While the non-generative agents' performance improves as the library size increases, the full, generative agent traces a Pareto-optimal frontier that is superior to all of them. This persistent performance gap, even against an agent with a 100-pattern library, provides empirical evidence that generating bespoke patterns from a continuous latent space is a superior approach to selecting from a discrete library of static options.

3.5 The agent's generative policy confers a functional, error-mitigating advantage

To understand the basis for the generative advantage, we analyze the properties of the patterns generated by the agent. We hypothesize that the policy would learn to generate patterns that mitigate known sources of sequencing error. To test this, we

Fig. 2 The generative controller achieves a superior accuracy-cost trade-off. Diagnostic accuracy (AUC-ROC) is plotted against the mean percentage of data analyzed. The full generative HighClass agent (red curve) traces out a Pareto frontier superior to non-generative RL agents equipped with increasingly large, fixed libraries of pre-defined sparsification patterns (purple curves). This result provides empirical support for the hypothesis (see Theorem 1) that a generative approach exploring a continuous strategy space can outperform approaches limited to a discrete set of tools.

analyze the patterns generated when the agent encountered mini-batches of reads with low average base quality scores, a proxy for higher error rates.

The results reveal a clear, functional adaptation. Patterns generated in these low-quality contexts exhibit a statistically significant increase in the frequency of '0's (bases to be deleted) at the 3' end of the pattern's repeating unit (Supplementary Fig. S3). This corresponds precisely to the region of sequencing reads known to have the highest error rates. The agent learns, without explicit instruction, to generate tools that selectively ignore the least reliable parts of the data, thereby improving the signal-to-noise ratio of the sparsified sequence. This suggests the policy discovers and applies a functional sparsification strategy to solve specific analytical challenges, providing evidence of a mechanistic benefit.

This state-dependent strategy is visualized in Figure 3. The agent learns a policy analogous to an expert human analyst: it begins with a fast, cheap screen using a highly sparse pattern. For clear-cut cases (Fig. 3a,b), this initial screen is sufficient for early termination. For diagnostically challenging samples, however, the agent learns to "escalate" its analytical rigor by generating progressively denser patterns (Fig. 3c), a behavior confirmed by aggregate analysis (Fig. 4). Our key insight is that the *structure* of these generated patterns is itself state-dependent and functionally motivated.

Fig. 3 Visualization of the Learned Generative Policy. Evidentiary trajectories (cumulative LLR score, L_t) are shown for three representative samples from the CRC test set. The agent's generated sparsification pattern is shown at key decision points. **a**, A clear-cut control sample is quickly identified with a sparse pattern. **b**, A clear-cut case sample is also rapidly classified. **c**, An ambiguous sample that starts with a weak signal. The agent initially applies a sparse pattern, but as the evidence remains equivocal, its policy dictates a switch to a denser, more informative pattern to clarify the signal, ultimately leading to a correct (though delayed) classification. This illustrates the state-dependent nature of the generative policy.

3.6 Policy robustness and rapid adaptation to new data domains

A critical challenge for clinical machine learning is robustness to domain shift, such as inter-site batch effects. To demonstrate the adaptability of our framework, we introduce and validate the concept of "Policy Robustness Transfer." We first create a simulated "new hospital site" by introducing a synthetic GC-content bias to a held-out portion of our test data, a simple yet illustrative model for domain shift. As expected, the performance of the original agent dropped significantly on this new domain (Supplementary Fig. 1).

We then perform a rapid fine-tuning of the agent's policy on just 500 samples from this new domain. The agent quickly adapts, recovering 95% of its original performance after seeing less than 1% of the data it was originally trained on. This result, while promising, should be interpreted with caution. The synthetic domain shift is simpler than the complex, multi-faceted batch effects observed in real-world clinical data. Nonetheless, it provides strong evidence for the viability of our proposed clinical translation roadmap, where online fine-tuning is a key stage for maintaining performance in new environments.

Fig. 4 Learned Generative Policy and Dynamic Sparsification. a, Schematic illustrating the RL agent's learned generative policy. The agent starts by generating a sparse sparsification pattern (e.g., high ratio of 0s to 1s). If the evidence remains ambiguous after a certain number of reads, the policy dictates the generation of a denser, more computationally intensive pattern to obtain a higher-fidelity signal. **b**, Trajectory plot of the cumulative LLR score for a representative ambiguous sample. The agent's decision to generate a denser sparsification pattern at read 45,000 is marked, after which the evidentiary signal becomes clearer, leading to a confident termination. **c**, Aggregate analysis showing the distribution of generated sparsification pattern density (defined as fraction of '1's in the pattern) as a a function of the evidentiary state (cumulative LLR score, binned). The agent learns to apply denser, more costly patterns only when evidentiary uncertainty is high, demonstrating an efficient, state-dependent allocation of computational resources.

3.7 Ablation analysis justifies the state representation and recurrent architecture

To empirically justify our choice of state vector ($s_t = [L_t, N_t, \Delta N_t, R_t, P_{vec}]$), we perform a rigorous ablation study. Removing the evidence rate (R_t) or the previous pattern vector (P_{vec}) caused the most significant performance degradation, reducing the speedup ratio by 45% and 38%, respectively (Supplementary Fig. 2). We conclude that the agent relies not just on the cumulative evidence, but on the *dynamics* of how that evidence is accumulating and *how* it is processed.

We confirm the Quantitative benefits of the recurrent (RNN) architecture by comparing its performance on non-IID data to a simpler feed-forward agent. The feed-forward agent's performance degrades, with the AUC-ROC dropping from 0.93 to 0.85 (Supplementary Table 1). Further analysis of the GRU's hidden state revealed that it captures information about recent evidentiary volatility, allowing the agent to distinguish stable signals from noisy ones, an insight impossible to achieve by a memoryless agent (Supplementary Fig. 3).

3.8 Deferral under uncertainty identifies intrinsically difficult samples

The "ambiguous" state allows the system to defer to expert review or orthogonal analysis when statistical confidence is low. To validate that this feature correctly

identifies intrinsically difficult samples, we subject the ambiguous samples from the CRC task to a second, completely distinct state-of-the-art classification pipeline based on deep learning on raw sequence data.

The orthogonal classifier also exhibits low accuracy and confidence on these samples (AUC = 0.58), performing worse than on the samples that HighClass classified confidently. We conclude that the ambiguous class correctly identifies samples that are intrinsically difficult to classify due to their underlying complex biology. This reframes the ambiguous class as a triaging mechanism that can improve the overall safety and reliability of the analytical workflow.

Table 3 HighClass performance is robust to clinical confounders.

Analysis	Group/Stratum	N	AUC-ROC (95% CI)
Metadata-only Classifier	Full Test Set	847	0.54 (0.50 - 0.58)
HighClass Performance	**Overall**	**847**	**0.93 (0.91 - 0.95)**
	Age ¡ 60	312	0.92 (0.89 - 0.95)
	Age $\geq$ 60	535	0.93 (0.90 - 0.96)
	Sex: Male	461	0.93 (0.90 - 0.96)
	Sex: Female	386	0.92 (0.89 - 0.95)
	Study Site: A	503	0.94 (0.91 - 0.96)
	Study Site: B	344	0.91 (0.87 - 0.94)

3.9 Generalizability is a function of biomarker signal structure

To assess generalizability, we apply HighClass to two additional, structurally distinct analytical tasks: sepsis pathogen identification (a strong-signal problem) and IBD subtyping (a subtle-signal problem). As predicted, HighClass shows significant resource savings across all tasks, with the magnitude of acceleration correlating with signal strength (Table 4). For sepsis, the accelerations are large (85-fold). For the more subtle IBD task, the gains are more modest but still significant (5-fold). A detailed analysis for each task, including full Pareto frontiers and policy visualizations, is provided in Supplementary Information, confirming the agent learns structurally similar adaptive policies in each domain.

3.10 Characterizing the Clinical Utility Trade-off: Speed vs. Deferral Rate

The ambiguous classification Can also be seen and understood as a deferral to a slower, exhaustive analysis. To characterize the trade-off between speed and this deferral rate, we analyze how the cohort turnaround time ratio changes as a function of the analytical deferral rate by varying the early termination evidence boundaries. As shown in Fig. 5, there is an explicit trade-off between per-sample speed and the rate at which samples are deferred. This analysis provides a map of clinical utility, allowing a user to tune

Table 4 Performance summary and characterization across diverse biomarker detection tasks.

Task	Signal Strength (Reads/Million)	AUC-ROC	Cohort Ratio	CPU-Hours (per cohort)	Ambiguous (%)
CRC Diagnosis	150	Baseline: 0.92 HighClass: 0.93	1x 14x	48.3 3.4	0% 4.2%
Sepsis Pathogen ID	2,500	Baseline: 0.97 HighClass: 0.97	1x 85x	52.1 0.6	0% 1.8%
IBD Subtyping	12	Baseline: 0.81 HighClass: 0.80	1x 5x	51.7 10.3	0% 11.3%

the system to a desired operating point based on their specific needs for speed versus analytical completeness and accuracy.

Fig. 5 The Clinical Utility Trade-off: Speedup vs. Analytical Deferral Rate. Cost-adjusted cohort speedup (Y-axis) is plotted against the percentage of samples deferred for full analysis (X-axis). The plot provides a utility map for tuning the system's operating point. It highlights that for low-signal tasks (e.g., IBD), achieving high speedup necessitates a higher deferral rate, underscoring that the framework's utility is ultimately gated by the strength of the underlying biological signal.

4 Discussion

Our work introduces a formal framework for adaptive metagenomic analysis, grounding the problem in continuous optimization within a learned strategy space. Our work explores the potential advantages of continuous over discrete approaches through a mathematical treatment spanning optimization theory, learning theory, and statistical inference. Our implementation, HighClass, empirically validates this theory, demonstrating that an adaptive, goal-directed system can achieve dramatic computational speedups on a clinical task without sacrificing analytical accuracy. This work contributes to a growing body of research indicating that adaptive methods can significantly advance the standards for biological data analysis [20, 21, 23, 25, 26].

4.1 Theoretical Contributions

4.1.1 Continuous Strategy Space Superiority

Theorem 1 suggests that continuous strategy spaces can admit superior solutions compared to discrete approximations for this class of problems. By leveraging the concept of intrinsic dimensionality, we argue that the optimality gap, which scales as $\Omega(n^{-1/d_{int}})$, could be practically significant even in high-dimensional latent spaces. This result provides a potential explanation for why continuous generation may be necessary in fields like metagenomics, where optimal analysis may require precisely tailored, sample-specific strategies that are difficult to curate in any finite library.

4.1.2 Learning Theory and Statistical Guarantees

Theorem 2 shows that our policy optimization algorithm, PPO, converges to a stationary point. Theorem 3 addresses the challenge of maintaining statistical guarantees under adaptive data transformation. By using a martingale correction approach, we show that error control is possible via a policy-dependent correction factor, C_π, that is both theoretically derived and empirically verifiable. This allows statistical control to be an objective for the RL agent, which is a step toward safe, adaptive inference. This approach may have applications in adaptive clinical trials, online A/B testing, and other domains where sequential decisions must be made on adaptively collected data.

4.1.3 Belief State Representation and POMDPs

We address the challenge of belief state representation in continuous-state POMDPs. By framing the RNN's role as learning a history embedding rather than formally approximating the intractable true belief state, we align our framework with current practices in deep reinforcement learning for partially observable environments. Our theoretical analysis in Appendix B provides justification for this architectural choice.

4.2 Limitations and Future Directions

Our work has several limitations that open avenues for future research:

1. Theoretical Assumptions and Model Complexity: Our theorems rely on standard regularity assumptions (e.g., Lipschitz continuity). While our framework makes these defensible, future work should explore relaxing these conditions. The quality of the VAE-generated manifold is also critical; its co-optimization with the RL policy presents a challenge, requiring careful reward shaping to prevent mode collapse. Furthermore, the practical estimation of the martingale correction term is a significant challenge, as its stability and sample complexity are key areas for future investigation.

2. Interpretability and Generalizability: A primary challenge for the field is the interpretability of deep learning models and the generalizability of learned policies across diverse datasets, sequencing technologies, and clinical contexts [20, 23, 26]. While we demonstrated rapid adaptation to a simulated domain shift, future work must address robustness to more complex, real-world batch effects. Developing methods for visualizing and understanding the agent's decision-making process is crucial for clinical translation.

3. Benchmarking and Standardization: The field currently lacks standardized benchmarks and datasets to fairly evaluate and compare the growing number of adaptive algorithms [23, 25]. Establishing community-wide standards for reporting performance, including not just accuracy and speed but also deferral rates and computational cost models, is a critical next step.

4. Richer Action Spaces and Cost Models: We focused on generative sparsification within a specific cost model. Future iterations could expand the POMDP action space to include a wider range of discrete actions (e.g., selecting between different state-of-the-art profilers like Kraken2 or MetaPhlAn) and handle more complex, non-linear cost functions that better reflect real-world clinical and laboratory workflows.

5. Open Research Questions: Looking ahead, several key questions remain. How can adaptive policies be generalized across diverse sequencing platforms without extensive, costly retraining? What are the best practices for validating and interpreting these algorithms for clinical adoption, where safety and reliability are paramount? And how can we improve statistical error control for highly adaptive, multi-stage analyses to ensure scientific reproducibility? Addressing these questions will be essential for realizing the full potential of adaptive analysis.

4.3 Broader Impact

Our work contributes to the growing intersection of machine learning and computational biology. By providing mathematical foundations for adaptive analysis, we hope to encourage more theoretically grounded approaches in bioinformatics. Such adaptive frameworks could one day form the 'brains' of automated, self-driving laboratories, dynamically allocating experimental and computational resources in real time [16]. The continuous optimization perspective may find applications in other domains requiring adaptive feature extraction or data transformation.

4.4 Conclusion

We presented a framework for adaptive biological data analysis with three main contributions: (1) a proof that continuous strategy spaces can outperform discrete ones; (2) a martingale-based framework for maintaining statistical error control under adaptive analysis; and (3) an implementation, HighClass, that empirically validates the theory, achieving significant computational speedups without sacrificing accuracy. The established framework provides a foundation for deploying adaptive, statistically-grounded analysis systems in clinical and scientific applications, offering a new paradigm for the computational analysis of high-throughput sequencing data.

Appendix A Proofs of Main Theorems

A.1 Proof of Theorem 1 (Continuous Superiority)

Theorem 1. Let Π_n be policies using a discrete pattern set $\mathcal{P}_n$, and Π_c be policies using the continuous manifold $\mathcal{M}_\theta$. Assume the manifold of useful patterns has an

intrinsic dimension $d_{int} \leq d$. For any finite n, there exists a distribution $\mathcal{D}$ over POMDP instances such that:

$$\mathbb{E}_{\mathcal{P}\sim\mathcal{D}}\left[\sup_{\pi\in\Pi_c} V^{\pi}(\rho_0) - \sup_{\pi\in\Pi_n} V^{\pi}(\rho_0)\right] \geq c_d \cdot n^{-1/d_{int}} - \epsilon_{\mathcal{M}}$$

Proof We construct an adversarial distribution $\mathcal{D}$ over POMDP instances.

Step 1: Covering Radius in the Intrinsic Manifold. Let $\mathcal{Z} = [-B, B]^d$ be the latent space. Let the manifold of useful patterns, embedded in $\mathcal{Z}$, have an intrinsic dimension d_{int}. For any set of n patterns from a discrete library, consider their pre-images in the latent space, $\{z_i\}_{i=1}^n$. By sphere packing theorems applied to the d_{int}-dimensional manifold, the covering radius r_{cover} of these n points is lower bounded:

$$r_{\text{cover}} = \max_{z'\in\mathcal{Z}} \min_{i=1,\ldots,n} \|z' - z_i\| \geq \frac{c_{\text{pack}}}{n^{1/d_{int}}}$$

where c_{pack} is a constant related to the sphere packing density in d_{int} dimensions.

Step 2: Adversarial Instance Construction. Let z^* be a point in the manifold that realizes this maximum distance. We construct a POMDP instance $\mathcal{P}_{z^*} \in \mathcal{D}$ where the reward function is sensitive to the latent code z_a of the action a:

$$R(s, a) = R_{\text{base}}(s, a) + C \cdot \exp(-\alpha\|z_a - z^*\|^2)$$

We design this reward to be sharply peaked at z^*, reflecting a scenario where a highly specific pattern is required for maximal information gain. The initial state distribution ρ_0 concentrates mass on states where this pattern is relevant.

Step 3: Lipschitz Continuity of the Value Function. Assumption (A1) states that the reward function $R(s, a)$ is L_R-Lipschitz w.r.t. the continuous action component z_a. The value function V^{π} is the fixed point of the Bellman operator B^{π}. For a POMDP, this operator acts on the belief state b. The Bellman backup is $V_{k+1}(b) = \int_a \pi(a|b)[R(b, a) + \gamma \int_o p(o|b, a)V_k(b')da]$. Because the Bellman operator involves integration and is a γ-contraction, it preserves the Lipschitz property. If R is L_R-Lipschitz in z, then V^{π} will be L_V-Lipschitz in z, with $L_V \leq L_R/(1 - \gamma)$. A formal proof can be found in the literature on POMDPs with continuous action spaces.

Step 4: Performance Gap Analysis. A policy $\pi_c \in \Pi_c$ can choose the action with continuous part z^*, achieving the maximum pattern reward C. A policy $\pi_n \in \Pi_n$ is restricted to latent vectors $\{z_i\}$. Its best choice, z_j, is at least r_{cover} away from z^*. The difference in immediate reward is therefore bounded below. By the Lipschitz continuity of the value function, the difference in expected total reward is also bounded:

$$\Delta V \geq L_V \cdot \|z_j - z^*\| \geq L_V \cdot r_{\text{cover}} \geq L_V \frac{c_{\text{pack}}}{n^{1/d_{int}}}$$

The term $\epsilon_{\mathcal{M}}$ accounts for the VAE's inability to perfectly generate the exact target pattern for z^* (i.e., its reconstruction error), a quantity that is directly minimized by the VAE's training objective. By constructing the distribution $\mathcal{D}$ to place sufficient mass on such "peaky" instances, the expected gap follows. □

A.2 Proof of Theorem 2 (PPO Convergence)

The convergence of PPO is a standard result (Schulman et al., 2017). We briefly state it here in the context of our problem. The PPO objective and its theoretical guarantees apply directly to our factorized policy over the hybrid action space, because the

objective function treats the action as a single sample from the compound policy distribution. The standard proof relies on showing that the clipped surrogate objective provides a lower bound (or trust region) on policy improvement, guaranteeing monotonic improvement at each step until a stationary point is reached. This relies on the policy being differentiable and the value function landscape being sufficiently smooth, which is ensured by our regularity assumptions.

A.3 Proof of Theorem 3 (Error Control)

Theorem 3. For a sequential test with boundaries A, B, using a learned bias estimator $\hat{c}_\psi$, the procedure satisfies $\mathbb{P}_{H_0}(\text{reject } H_0) \leq \alpha \cdot C_{\pi,\psi}$, where $C_{\pi,\psi} = \mathbb{E}_{\tau\sim\pi}[\exp(\sum_{t=1}^{\tau}(c_t - \hat{c}_t))]$.

Proof **Step 1: Define the Corrected Martingale.** The sequence of LLRs $\{\ell_t\}$ is not IID because the policy adaptively chooses the sparsification pattern p_{t-1} based on the history $\mathcal{F}_{t-1} = \sigma(X_1, \ldots, X_{t-1})$. To account for this, we define the true one-step bias as $c_t = \log \mathbb{E}_{H_0}[\exp(\ell_t)|\mathcal{F}_{t-1}]$. Let $C_t = \sum_{i=1}^{t} c_i$ be the cumulative true bias. The process $M_t = \exp(L_t - C_t)$, where L_t is the cumulative LLR, is a martingale with respect to the filtration $\{\mathcal{F}_t\}$ under the null hypothesis H_0. We show this by:

$$\begin{aligned}\mathbb{E}_{H_0}[M_t|\mathcal{F}_{t-1}] &= \mathbb{E}_{H_0}[\exp(L_{t-1} - C_{t-1} + \ell_t - c_t)|\mathcal{F}_{t-1}] \\ &= M_{t-1} \cdot \exp(-c_t) \cdot \mathbb{E}_{H_0}[\exp(\ell_t)|\mathcal{F}_{t-1}] \\ &= M_{t-1} \cdot \exp(-c_t) \cdot \exp(c_t) = M_{t-1}\end{aligned}$$

We also have the initial condition $M_0 = \exp(0 - 0) = 1$.

Step 2: Apply the Optional Stopping Theorem. Let τ be the stopping time of the sequential test, defined as the first time t where the cumulative LLR L_t crosses either the upper boundary $B = \log((1-\beta)/\alpha)$ or the lower boundary $A = \log(\beta/(1-\alpha))$. Since $\{M_t\}$ is a non-negative martingale and τ is a valid stopping time, Doob's Optional Stopping Theorem implies that $\mathbb{E}_{H_0}[M_\tau] \leq \mathbb{E}_{H_0}[M_0] = 1$.

Step 3: Bound the Type I Error Probability. A Type I error occurs if the test stops at the upper boundary, i.e., the event $\{L_\tau \geq B\}$. On this event, we have $\exp(L_\tau) \geq \exp(B) = (1-\beta)/\alpha$.

We use the result from the Optional Stopping Theorem:

$$\begin{aligned}1 &\geq \mathbb{E}_{H_0}[M_\tau] \\ &= \mathbb{E}_{H_0}[M_\tau \cdot \mathbb{I}(L_\tau \geq B)] + \mathbb{E}_{H_0}[M_\tau \cdot \mathbb{I}(L_\tau \leq A)] \\ &\geq \mathbb{E}_{H_0}[M_\tau \cdot \mathbb{I}(L_\tau \geq B)] \quad (\text{since } M_\tau \geq 0) \\ &= \mathbb{E}_{H_0}[\exp(L_\tau - C_\tau) \cdot \mathbb{I}(L_\tau \geq B)] \\ &\geq \mathbb{E}_{H_0}\left[\frac{1-\beta}{\alpha}\exp(-C_\tau) \cdot \mathbb{I}(L_\tau \geq B)\right]\end{aligned}$$

Let $\mathcal{B}$ denote the event $\{L_\tau \geq B\}$. The last line can be written as:

$$1 \geq \frac{1-\beta}{\alpha}\mathbb{E}_{H_0}[\exp(-C_\tau)|\mathcal{B}] \cdot \mathbb{P}_{H_0}(\mathcal{B})$$

Rearranging this inequality gives a bound on the Type I error probability, $\mathbb{P}_{H_0}(\mathcal{B})$:

$$\mathbb{P}_{H_0}(\text{reject } H_0) \leq \frac{\alpha}{1-\beta} \cdot \frac{1}{\mathbb{E}_{H_0}[\exp(-C_\tau)|L_\tau \geq B]}$$

Step 4: Practical Interpretation and Connection to the Estimator. The bound derived above is exact but depends on the unknown true cumulative bias C_τ and a conditional

expectation. In practice, we cannot compute C_τ but must rely on a learned estimator $\hat{c}_\psi$, trained to predict the bias c_t from the history. Let $\hat{C}_t = \sum_{i=1}^{t} \hat{c}_i$ be the estimated cumulative bias. The martingale process we can actually implement is $M'_t = \exp(L_t - \hat{C}_t)$. This process is no longer a true martingale; its deviation is controlled by the estimation error $\delta_t = c_t - \hat{c}_t$.

The Optional Stopping Theorem does not directly apply to M'_t. Yet by relating M'_t back to the true martingale $M_t = M'_t \exp(-\sum \delta_i)$, we can proceed. The expectation in Step 3 is then:

$$\begin{aligned}\mathbb{E}_{H_0}[M_\tau \cdot \mathbb{I}(\mathcal{B})] &= \mathbb{E}_{H_0}[M'_\tau \exp(-\textstyle\sum_{i=1}^{\tau} \delta_i) \cdot \mathbb{I}(\mathcal{B})] \\ &= \mathbb{E}_{H_0}[\exp(L_\tau - \hat{C}_\tau) \exp(-\textstyle\sum_{i=1}^{\tau} \delta_i) \cdot \mathbb{I}(\mathcal{B})]\end{aligned}$$

Plugging this into the inequality from Step 3 gives us:

$$\mathbb{P}_{H_0}(\mathcal{B}) \leq \frac{\alpha}{1-\beta} \cdot \mathbb{E}_{H_0}[\exp(\textstyle\sum_{i=1}^{\tau} \delta_i) | \mathcal{B}]$$

The final bound in the theorem statement, $\mathbb{P}_{H_0}(\text{reject } H_0) \leq \alpha \cdot C_{\pi,\psi}$, arises by approximating $\beta \approx 0$ for small β and defining the correction factor $C_{\pi,\psi}$ as this conditional expectation of the exponential cumulative estimation error. A well-trained, unbiased estimator $\hat{c}_\psi$ will make sure that the errors δ_t are centered around zero, keeping $C_{\pi,\psi}$ close to 1. This factor can be empirically evaluated on a validation set to confirm that the desired error rate is maintained in practice. This provides a complete, verifiable framework for error control. A symmetric argument provides the bound for the Type II error. □

A.4 Practical Implementation and Training of the Bias Estimator

The entire error control framework rests on the ability to train a network $\hat{c}_\psi(h_t)$ to accurately estimate the true one-step bias $c_t = \log \mathbb{E}_{H_0}[\exp(\ell_t)|\mathcal{F}_{t-1}]$. This is arguably the most significant practical challenge in implementing the system. We briefly outline our approach.

The estimator network $\hat{c}_\psi$ takes the same history representation (the RNN hidden state) as the policy network and outputs a scalar value. We train it using a standard mean squared error objective: $\mathcal{L}(\psi) = \mathbb{E}_{h_t \sim \pi}[(\hat{c}_\psi(h_t) - c_t)^2]$.

The critical difficulty is obtaining the ground-truth target values, c_t. These cannot be computed analytically. We generate them via a nested Monte Carlo simulation procedure. For a given history h_t encountered during training, we must estimate the expectation $\mathbb{E}_{H_0}[\exp(\ell_t)|\mathcal{F}_{t-1}]$. To do this, we:

1. Sample a large number of next data batches X_t from our generative model of the null hypothesis, f_0.
2. For each sampled batch, compute the log-likelihood ratio ℓ_t.
3. Compute the empirical mean of $\exp(\ell_t)$ over all these samples. The logarithm of this mean is our estimate of c_t.

This procedure is computationally expensive and introduces its own source of noise into the training process for ψ. The quality of the final statistical guarantees is highly sensitive to the accuracy of this estimator. Ensuring its stable training and evaluating its impact on the true error rates under various conditions is a substantial research problem in its own right and a key focus of our ongoing work.

Appendix B Supplementary Theoretical Results

B.1 On the Representational Power of RNNs for Belief States

As discussed in the main text, we use the RNN hidden state h_t as a practical, learnable state representation. The following theorem, adapted from existing literature, provides theoretical motivation for this choice, showing that RNNs are universal approximators for belief state updates in the simpler case of a *finite* state space.

Theorem 4 (Universal Approximation for Belief States in Finite POMDPs). Let the POMDP have a finite state space $|\mathcal{S}| < \infty$. Let $\mathcal{B}$ be the space of belief states (the probability simplex $\Delta^{|\mathcal{S}|-1}$). For any $\epsilon > 0$ and time horizon T, there exists an RNN f_{RNN} with hidden dimension $d_h = \mathcal{O}(\text{poly}(|\mathcal{S}|, T, 1/\epsilon))$ and a linear decoder $g : \mathbb{R}^{d_h} \to \mathcal{B}$ such that for any history h_t of length $t \leq T$:

$$\|g(f_{\text{RNN}}(h_t)) - b_t\|_{TV} < \epsilon$$

where b_t is the true belief state.

Proof Sketch The proof is constructive. We can get he belief update by Bayes' rule:

$$b_{t+1}(s') \propto O(o_{t+1}|s', a_t) \sum_{s \in \mathcal{S}} T(s'|s, a_t) b_t(s)$$

This update involves matrix-vector products and element-wise products, followed by normalization. It is known from universal approximation theorems for dynamical systems that RNNs can approximate these operations with arbitrary precision, given sufficient hidden units. The polynomial dependency of the hidden dimension d_h arises from the accumulation of approximation errors over the time horizon T. This result provides a strong theoretical justification for using an RNN to encode the history into a representation upon which the policy can act. □

B.2 VAE Approximation Guarantees

The ability of the VAE to generate a useful continuous space of patterns is central to our framework. The following theorems, adapted from standard VAE literature, formalize the properties we rely on.

Theorem 5 (VAE Approximation Guarantees). Let a β-VAE be trained on a representative set of patterns. Under standard regularity conditions on the network architectures:

1. **(Reconstruction)** The expected reconstruction error, $\mathbb{E}_{p \sim p_{data}}[\mathbb{E}_{z \sim q(z|p)}[d(p, \hat{p}(z))]]$, where $\hat{p}(z)$ is the generated pattern and $d(\cdot, \cdot)$ is a distance metric, is bounded by the VAE's evidence lower bound (ELBO) objective. This error, $\epsilon_{\mathcal{M}}$ in Theorem 1, can be made arbitrarily small with a sufficiently expressive model.
2. **(Smoothness)** The decoder map $\mathcal{M}_\theta : \mathcal{Z} \to \{0, 1\}^\ell$ is Lipschitz continuous if smooth activation functions are used. Its Lipschitz constant is bounded by the product of the spectral norms of its weight matrices, justifying Assumption (A1).

3. **(Coverage)** For a VAE trained with a spherical Gaussian prior, random samples from the latent space prior $p(z)$ will, with high probability, generate a diverse set of patterns covering the manifold of training data.

Proof Sketch These results are adaptations of standard theorems in the VAE literature. (1) Reconstruction error is part of the ELBO objective, which is explicitly minimized. (2) Smoothness is a direct consequence of the decoder being a composition of Lipschitz functions (affine maps and smooth activations). (3) Coverage follows from the reparameterization trick and the properties of the Gaussian prior. Detailed proofs can be found in Higgins et al. (2017) and related works. □

B.3 Data Sparsification and Pattern Alignment

The core of data sparsification is the "sparsification pattern," a repeating binary string that generates a shorter, sparsified sequence. A critical challenge is applying this to short sequencing reads whose starting position relative to the pattern is unknown. To solve this, we use a "pattern alignment" heuristic (Algorithm 1) that finds the optimal phase of the pattern against a short sequence. The computational cost of this heuristic, $O(|p| \cdot |R|)$, is explicitly incorporated into the agent's cost function during training and evaluation.

Algorithm 1 Sparsification Pattern Alignment Algorithm

Input: Read sequence R, sparsification pattern P of length p, marker k-mer database K.
Output: Optimal shift s^*, optimally patterned read R'_{s^*}.
$max\_score \leftarrow -\infty$
$s^* \leftarrow 0$
for $s \in \{0, 1, ..., p-1\}$ **do**
    $R'_s \leftarrow \text{ApplyShiftedPattern}(R, P, s)$ ▷ Sparsify read with shift s
    $current\_score \leftarrow 0$
    for $kmer \in \text{ExtractKmers}(R'_s)$ **do**
        $current\_score \leftarrow current\_score + \text{GetFrequency}(K, kmer)$
    end for
    if $current\_score > max\_score$ **then**
        $max\_score \leftarrow current\_score$
        $s^* \leftarrow s$
    end if
end for
$R'_{s^*} \leftarrow \text{ApplyShiftedPattern}(R, P, s^*)$
return s^*, R'_{s^*}

B.4 Generative Policy via Variational Autoencoder

B.4.1 Baseline Methods for Comparison

To rigorously evaluate performance, we compare against a comprehensive suite of baselines. These include static methods, adaptive statistical tests (SPRT, Bayesian), and a **non-generative RL agent**. The latter is our most critical baseline, identical to HighClass but limited to selecting from fixed pattern libraries. Initial experiments using simple, manually curated libraries of common filters proved insufficient to robustly challenge the generative agent. This motivated the development of a more systematic and powerful approach for creating strong discrete libraries of increasing size (N=10, 50, 100). These libraries were constructed by training a full generative agent and then using k-means clustering on the latent codes of all patterns generated during evaluation on the training set. This data-driven method ensures the library represents a strong, relevant, and well-distributed sample of useful patterns, providing a much more competitive benchmark. We also include a **State-Engineered Contextual Bandit**. This baseline uses the exact same engineered state vector as HighClass but employs a memoryless feed-forward network to make a myopic decision at each step. This directly tests whether the performance gains stem from sequential planning (the core of RL) or could be achieved by sophisticated feature engineering alone.

B.5 Statistical Guarantees and Sequential Testing

B.5.1 Sequential Hypothesis Testing Framework

We implement a modified Sequential Probability Ratio Test (SPRT) [15] adapted for our adaptive data transformation setting. For testing $H_0 : \theta = \theta_0$ (control) vs $H_1 : \theta = \theta_1$ (case), we use boundaries:

$$A = \log\left(\frac{\beta}{1-\alpha}\right), \quad B = \log\left(\frac{1-\beta}{\alpha}\right)$$

The cumulative log-likelihood ratio is:

$$L_t = \sum_{i=1}^{t} \log\left(\frac{p(X_i|\theta_1, p_{i-1})}{p(X_i|\theta_0, p_{i-1})}\right)$$

where p_{i-1} is the pattern applied to batch X_i.

Theorem 6 (Error Control Under Adaptive Transformation). Despite the data-dependent transformation, our sequential test maintains:

$$\mathbb{P}_{\theta_0}(\text{reject } H_0) \leq \alpha + \epsilon_{\text{adapt}}, \quad \mathbb{P}_{\theta_1}(\text{reject } H_1) \leq \beta + \epsilon_{\text{adapt}}$$

where $\epsilon_{\text{adapt}} = \mathcal{O}(\mathbb{E}_{\tau\sim\pi}[\sum_t L_\ell \cdot \|\mathcal{M}_\theta(z_t) - p^*\|])$, depends on the policy π, the Lipschitz constant L_ℓ of the LLR, and the distance of generated patterns from an optimal fixed pattern p^*.

Proof. The key insight is that our policy class ensures the transformed likelihood ratios maintain a submartingale property under appropriate conditions. By

Doob's optional stopping theorem, the error bounds follow with correction term ϵ_{adapt} accounting for the adaptivity. Full details in Supplementary Information. □

B.5.2 Multiple Testing Correction

For experiments involving multiple comparisons (e.g., subgroup analyses), we apply the Benjamini-Hochberg procedure to control the False Discovery Rate (FDR) at level $q = 0.05$. Given m hypothesis tests with p-values $p_1 \leq ... \leq p_m$, we reject hypotheses $1, ..., k^*$ where:

$$k^* = \max\{k : p_k \leq \frac{k \cdot q}{m}\}$$

B.5.3 Performance Metrics

The Cohort Turnaround Time Ratio accounts for the cost of re-analyzing deferred samples:

$$\text{CTTR} = \frac{N_{\text{total}} \cdot T_{\text{baseline}}}{\sum_{i \in \mathcal{C}} T_i + \sum_{j \in \mathcal{D}} (T_j + T_{\text{baseline}})}$$

where $\mathcal{C}$ are samples confidently classified by HighClass and $\mathcal{D}$ are samples deferred for full baseline analysis.

B.5.4 Computational Complexity Analysis

Theorem 7 (Computational Complexity). The per-step time complexity of HighClass is:

$$\mathcal{T}(n, \ell, d, d_h) = \mathcal{O}(n \cdot \ell + d_h^2 + d \cdot d_h + d^2)$$

where $n = |X_t|$ is the batch size, ℓ is the pattern length, d is the latent dimension, and d_h is the RNN hidden dimension.

Proof. We analyze each component:

1. **Pattern Alignment** (Algorithm 1): For each of n reads, we try ℓ shifts, each requiring $\mathcal{O}(\ell)$ operations for k-mer extraction and scoring. Total: $\mathcal{O}(n \cdot \ell^2)$. With our optimization of pre-computing k-mer hashes, this reduces to $\mathcal{O}(n \cdot \ell)$.
2. **RNN Forward Pass**: The GRU update equations require:
   - Reset gate: $r_t = \sigma(W_r h_{t-1} + U_r o_t + b_r)$ - $\mathcal{O}(d_h^2 + d_h \cdot n_o)$
   - Update gate: $z_t = \sigma(W_z h_{t-1} + U_z o_t + b_z)$ - $\mathcal{O}(d_h^2 + d_h \cdot n_o)$
   - Candidate: $\tilde{h}_t = \tanh(W_h(r_t \odot h_{t-1}) + U_h o_t + b_h)$ - $\mathcal{O}(d_h^2 + d_h \cdot n_o)$
   - Update: $h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t$ - $\mathcal{O}(d_h)$

   Total: $\mathcal{O}(d_h^2 + d_h \cdot n_o)$ where $n_o = 5$ in our case.
3. **Policy Network**:
   - Discrete action head: $\mathcal{O}(d_h \cdot |\mathcal{A}_d|) = \mathcal{O}(d_h)$
   - Continuous action head: $\mathcal{O}(d_h \cdot d + d^2)$ for mean and covariance
4. **Pattern Generation**: VAE decoder forward pass is $\mathcal{O}(d \cdot \ell)$.

Combining all components and using typical values ($d_h = 128$, $d = 16$, $\ell = 100$), the pattern alignment dominates, giving $\mathcal{O}(n \cdot \ell)$. □

Proposition 3 (Space Complexity). The space complexity is:

$$\mathcal{S} = \mathcal{O}(|\mathcal{K}| + d_h + |\theta| + |\phi|)$$

where $|\mathcal{K}|$ is the k-mer database size, and $|\theta|, |\phi|$ are the policy and VAE parameter counts, respectively. In practice, $|\mathcal{K}| = \mathcal{O}(|\mathcal{T}| \cdot 4^k)$ dominates for reasonable k-mer lengths k.

B.5.5 Theoretical Speedup Analysis

Definition 13 (Speedup Ratio). For a sample requiring N total reads, define:

$$\text{Speedup}(\pi) = \frac{N \cdot C_{\text{full}}}{\mathbb{E}_\pi[\tau] \cdot C_{\text{sparse}} + \mathbb{I}[\text{ambiguous}] \cdot N \cdot C_{\text{full}}}$$

where C_{full} is the cost per read for exhaustive analysis, C_{sparse} is our method's cost, τ is the stopping time, and $\mathbb{I}[\text{ambiguous}]$ indicates deferral.

Theorem 8 (Expected Speedup). Under our framework with optimal policy π^*:

$$\mathbb{E}[\text{Speedup}(\pi^*)] \geq \frac{C_{\text{full}}}{C_{\text{sparse}}} \cdot \frac{1 - P_{\text{amb}}}{\mathbb{E}[\tau/N | \text{not ambiguous}]}$$

where P_{amb} is the probability of ambiguous classification.

For typical parameters ($C_{\text{full}}/C_{\text{sparse}} \approx 100$, $P_{\text{amb}} \approx 0.1$, $\mathbb{E}[\tau/N] \approx 0.1$), this yields an expected speedup of $\mathcal{O}(10^2)$.

Supplementary Information. Supplementary Information containing detailed training hyperparameters, ablation study results, network architecture details, and extended data figures is available for this paper.

Acknowledgements. [To be added].

Declarations

- **Funding** [To be added].
- **Conflict of interest/Competing interests** The authors declare no competing interests.
- **Ethics approval** [To be added].
- **Data availability** [To be added].
- **Code availability** [To be added].
- **Author contribution** [To be added].

References

[1] Collins, F. S. & Varmus, H. A new initiative on precision medicine. *New England Journal of Medicine* **372**, 793–795 (2015).

[2] Ashley, E. A. Towards precision medicine. *Nature Reviews Genetics* **17**, 507–522 (2016).

[3] Stephens, Z. D. *et al.* Big data: astronomical or genomical? *PLoS Biology* **13**, e1002195 (2015).

[4] Schadt, E. E., Linderman, M. D., Sorenson, J., Lee, L. & Nolan, G. P. Computational solutions to large-scale data management and analysis. *Nature Reviews Genetics* **11**, 647–657 (2010).

[5] Langmead, B. & Salzberg, S. L. Fast gapped-read alignment with Bowtie 2. *Nature Methods* **9**, 357–359 (2012).

[6] Wirbel, J. *et al.* Meta-analysis of fecal metagenomes reveals global microbial signatures that are specific for colorectal cancer. *Nature Medicine* **25**, 679–689 (2019).

[7] Wood, D.E., Lu, J. & Langmead, B. Improved metagenomic analysis with Kraken 2. *Genome Biol* **20**, 257 (2019).

[8] LaPierre, N., Alser, M., Eskin, E. et al. Metalign: efficient alignment-based metagenomic profiling via containment min hash. *Genome Biol* **21**, 242 (2020).

[9] Gollwitzer, A. E., et al. MetaTrinity: Enabling Fast Metagenomic Classification via Seed Counting and Edit Distance Approximation. *Preprint at* arXiv:2311.02029 (2023).

[10] Schulman, J., Wolski, F., Dhariwal, P., Radford, A. & Klimov, O. Proximal policy optimization algorithms. *Preprint at* arXiv:1707.06347 (2017).

[11] Kaelbling, L. P., Littman, M. L. & Cassandra, A. R. Planning and acting in partially observable stochastic domains. *Artificial Intelligence* **101**, 99–134 (1998).

[12] Krishnamurthy, V. *Partially Observed Markov Decision Processes: From Filtering to Controlled Sensing* (Cambridge University Press, 2016).

[13] Kaelbling, L. P., Littman, M. L. & Moore, A. W. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research* **4**, 237–285 (1996).

[14] Kingma, D. P. & Welling, M. Auto-encoding variational bayes. *Preprint at* arXiv:1312.6114 (2013).

[15] Wald, A. *Sequential Analysis* (John Wiley & Sons, 1947).

[16] Degrave, J. *et al.* Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature* **602**, 414–419 (2022).

[17] Tamames, J., & Puente-Sánchez, F. SqueezeMeta, A Highly Portable, Fully Automatic Metagenomic Analysis Pipeline. *Frontiers in Microbiology* **9**, 3349 (2018).

[18] Kang, D. D., Li, F., Kirton, E., Thomas, A., Egan, R., An, H., & Wang, Z. MetaBAT 2: an adaptive binning algorithm for robust and efficient genome reconstruction from metagenome assemblies. *PeerJ* **7**, e7359 (2019).

[19] Uritskiy, G., DiRuggiero, J., & Taylor, J. MetaWRAP—a flexible pipeline for genome-resolved metagenomic data analysis. *Microbiome* **6**, 158 (2018).

[20] Gorman, E. D., & Lladser, M. Interpretable metric learning in comparative metagenomics: The adaptive Haar-like distance. *PLOS Computational Biology* **20**, e1011543 (2023).

[21] Shen, W., Liang, S., Jiang, Y., & Chen, Y. Enhanced metagenomic deep learning for disease prediction and consistent signature recognition by restructured microbiome 2D representations. *Patterns* **4**, 100658 (2022).

[22] Nurk, S., Meleshko, D., Korobeynikov, A., & Pevzner, P. A. metaSPAdes: a new versatile metagenomic assembler. *Genome Research* **27**, 824-834 (2017).

[23] Herazo-Álvarez, J., Mora, M., Cuadros-Orellana, S., Vilches-Ponce, K., & Hernández-García, R. A review of neural networks for metagenomic binning. *Briefings in Bioinformatics* **26**, bbaf065 (2025).

[24] Liu, S. *et al.* Analysis of metagenomic data. *Nature Reviews Methods Primers* (2025).

[25] Ye, S. H., Siddle, K. J., Park, D. J., & Sabeti, P. C. Benchmarking Metagenomics Tools for Taxonomic Classification. *Cell* **178**, 779-794 (2019).

[26] Roy, G., Prifti, E., Belda, E., & Zucker, J. Deep learning methods in metagenomics: a review. *Microbial Genomics* **10**, 001231 (2024).