LARGE LANGUAGE MODELS AS PREPROCESSING OPTIMISERS IN CLINICAL SPECTROSCOPY

Anonymous authors
Paper under double-blind review

ABSTRACT

Clinical spectroscopy has emerged as a powerful tool across biomedical research, with applications spanning from disease diagnosis and screening, to biomarker detection, therapeutic monitoring and intraoperative assessment. However, despite substantial progress, the analysis of spectroscopic data remains a major bottleneck, with current workflows still heavily reliant on trial and error, expert intuition and prior experience. Here, we introduce a multi-agent, large language model (LLM) framework for automated optimisation of preprocessing pipelines for spectral data. Our system combines a memory-derived information set, or *dossier*, with LLM-based agents that reason over previously evaluated preprocessing pipelines in order to propose, rank, and iteratively refine preprocessing strategies by following a multi-reward rule. Unlike exhaustive, brute-force search over a parameter space, the agent constructs candidates sequentially, enabling more targeted exploration. We benchmark our approach across 18 scenarios on four public clinical datasets. Our agent matches the accuracy of exhaustive grid search while evaluating fewer than 4% as many pipelines, and outperforms sequential stepwise optimisation in 17 of 18 scenarios. Ablation across LLM backbones and dossier configurations further shows that access to optimisation history is a key contributor to performance. These results demonstrate the potential of history-informed agents to enable more automated, adaptive and reproducible preprocessing pipelines in spectroscopy.

1 INTRODUCTION

Spectroscopic techniques, such as Raman spectroscopy (RS), Fourier transform infrared (FTIR) spectroscopy and diffuse optical/reflectance spectroscopy (DOS/DRS), enable rapid, non-destructive and label-free chemical analysis with minimal sample preparation (Butler et al., 2016; Baker et al., 2014; Bydlon et al., 2015). As such, they have emerged as powerful analytical tools for biomedical research and clinical medicine; with applications spanning from disease diagnosis and screening (Haka et al., 2005; Paraskevaidi et al., 2018; Giamougiannis et al., 2021b; Vališ et al., 2024), to biomarker detection (Qian et al., 2008; Plou et al., 2022), therapeutic monitoring (Cerussi et al., 2007; Cong et al., 2026) and intraoperative assessment (Jermyn et al., 2015; Hollon et al., 2020). Many of these applications involve (binary) classification of spectra into clinically relevant groups, such as disease states or tissue types. Despite substantial progress, however, robust analysis of spectroscopic data remains a major challenge, limiting the reproducibility, scalability and broader translation of spectroscopic methods into clinical practice (Baker et al., 2018; Coca-Lopez et al., 2025).

A critical component of spectroscopic data analysis is spectral preprocessing. Spectroscopic methods yield high-dimensional measurements that are often affected by various sources of non-specific variation, such as instrumental artefacts (e.g. cosmic-ray spikes), sample-derived backgrounds (e.g. fluorescence in RS, scattering-induced baselines in FTIR), experimental noise (e.g. detector and environmental noise), batch effects and sample preparation variability (Bocklitz et al., 2011; Isabelle et al., 2016; Butler et al., 2018; Vulchi et al., 2024). These can obscure biologically relevant spectral signatures and substantially degrade downstream analysis and predictive modelling. To mitigate these effects, spectroscopy workflows typically apply a series of preprocessing steps, such as despiking, smoothing, baseline correction and normalisation, to reduce non-specific variation and standardise collected measurements before analysis. While the impact of these steps is well documented (Bocklitz et al., 2011; Guo et al., 2021; Wahl et al., 2022), algorithm and hyperparameter selection remains

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

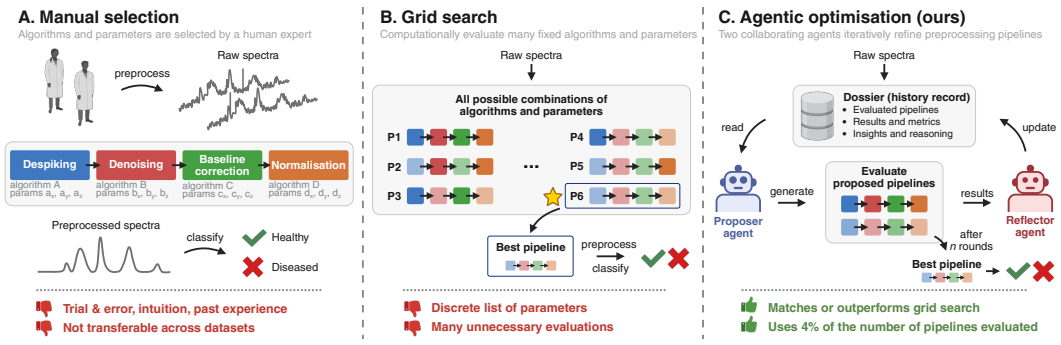


Figure 1: Comparative overview of preprocessing optimisation strategies in spectroscopy – manual selection (A), exhaustive grid search (B), and our proposed approach based on LLMs as optimisers (C).

largely reliant on manual trial and error, expert intuition and prior experience on comparable samples (Figure 1A).

A quantitative alternative is to search a predefined space of preprocessing pipelines and select configurations based on held-out performance. This is commonly achieved using exhaustive grid search over a selected set of algorithms and hyperparameters (Figure 1B) (Afseth et al., 2006; Lee et al., 2018; Storey & Helmy, 2019; Wahl et al., 2022; Kizilov et al., 2026; Han et al., 2026). However, exhaustive search is restricted to configurations specified in advance, can expend substantial computation on unpromising candidates, and may miss effective pipelines outside the search space. More targeted strategies based on genetic algorithms (Jarvis & Goodacre, 2005; Bocklitz et al., 2011; Brunel et al., 2021) and statistical experimental design (Gerretzen et al., 2015) can reduce the number of candidates explored, but still operate within manually specified sets of algorithms and parameters.

Recently, LLM-based agents have emerged as optimisers that enable adaptive, feedback-informed search. Agents can propose candidates conditioned on past performance (Yang et al., 2024), synthesise directional feedback from the optimisation trace (Nie et al., 2024), route observations into the weights (Lv et al., 2025), and distil the accumulated history into a compact form (Ran et al., 2025). These capabilities enable optimisation strategies that iteratively interpret previous outcomes, revise the search trajectory, and explore complex spaces without exhaustively enumerating candidates. Consequently, agentic optimisation is increasingly being applied across scientific and engineering domains, from computational design to experimental and laboratory workflows (Boiko et al., 2023; Ma et al., 2024; Chen et al., 2025; Murphy, 2026).

Agentic approaches have also recently emerged in spectroscopy for spectral interpretation (Noh et al., 2026), literature parsing and analysis (Fu et al., 2025; Xie et al., 2026), instrument control (Lim et al., 2026), and experimental planning and automation (Rittig et al., 2026). However, their use for performance-driven optimisation of spectral preprocessing remains largely unexplored.

Here, we develop an LLM-based agentic system for automated preprocessing optimisation that iteratively proposes, evaluates and refines preprocessing pipelines by reasoning over the accumulated search history and adaptively exploring the space of algorithms and hyperparameters (Figure 1C).

Contributions and findings. Our key contributions and findings are:

- We developed a multi-agent LLM-as-optimiser framework for automated optimisation of spectral preprocessing based on a *proposer* agent, a *reflector* agent, and a structured memory-derived information set over preprocessing pipelines, which we call *dossier*.
- We benchmarked two different LLM families and two parameter scales (8B and 70B parameters) across four clinical datasets (binary classification). We showed that our agentic framework matches the performance of exhaustive grid search while evaluating fewer than 4% as many pipelines, and outperforms sequential stepwise optimisation in 17 of 18 scenarios.
- We performed a range of ablation studies and empirically show that the history of the *dossier* is a key contributor to performance.

2 PROBLEM FORMULATION

Preprocessing search space. In this work, we consider preprocessing pipelines $\mathcal{S}$ composed of four stages, namely:

$$\mathcal{S} = \big(\text{despiking} \rightarrow \text{denoising} \rightarrow \text{baseline correction} \rightarrow \text{normalisation}\big), \tag{1}$$

which operate on spectral measurements $x \in \mathbb{R}^q$, where q is the number of spectral channels, and are applied sequentially in the order specified in Equation 1, in line with established guidelines (Guo et al., 2021). These are preprocessing steps commonly used in spectral data analysis to remove cosmic-ray spikes, reduce noise, correct baseline variation, and normalise spectral intensity, respectively. Each stage $s \in \mathcal{S}$ comes with a corresponding library of methods $\mathcal{M}_s$ to choose from, which also contains the identity $\varnothing$ (i.e. skip stage s). Each method $m \in \mathcal{M}_s$ has a parameter domain Θ_m, which is a product of integer ranges and real intervals subject to method-specific constraints. We write $\theta \in \Theta_m$ for a particular setting of those parameters. For instance, the parameter domain Θ_{SG} of a Savitzky-Golay denoising filter can be defined as $\Theta_{\mathsf{SG}} = \big\{(w, o, d) \in \mathbb{N}^3 \;\big|\; w \text{ is odd}\big\}$, where w specifies the window length, o specifies the polynomial order, and d specifies the derivative order (Savitzky & Golay, 1964).

The full action space $\mathcal{A}$ over $\mathcal{S}$ is thus given by the product:

$$\mathcal{A} \;=\; \prod_{s \in \mathcal{S}} \; \bigcup_{m \in \mathcal{M}_s} \{m\} \times \Theta_m, \tag{2}$$

where each action $a = \big[(m_s, \theta_m)\big]_{s \in \mathcal{S}}$ defines a preprocessing pipeline that induces a transform $T_a : \mathbb{R}^{n \times q} \rightarrow \mathbb{R}^{n \times q'}$, where q' is the number of spectral channels after preprocessing and $\theta_m \in \Theta_m$. The action space $\mathcal{A}$ is typically infinite since Θ_m contains real intervals. In our grid search experiments, we select a fixed set of discrete values for each real-valued Θ_m.

Pipeline evaluation. We focus on binary classification, a common framing in clinical spectroscopy and the one for which public cohorts are most widely available. We evaluate preprocessing strategies on clinical datasets $D = (X, y)$ containing patient measurements $X \in \mathbb{R}^{n \times q}$ and binary labels $y \in \{0, 1\}^n$ (e.g. *healthy* vs. *diseased*). For a given pipeline a, we define a reward vector $\mathbf{r}(a)$ combining ϕ evaluation metrics $\mu_1, \ldots, \mu_\phi$ as follows:

$$\mathbf{r}(a) \;=\; \Big(r_1(a), \ldots, r_\phi(a)\Big), \qquad r_i(a) \;=\; \frac{1}{RF} \sum_{\rho=1}^{R} \sum_{k=1}^{F} \mu_i\Big(y_{\mathcal{V}_{\rho k}},\; f_{\rho k}^{(a)}\big(T_a(X)_{\mathcal{V}_{\rho k}}\big)\Big), \tag{3}$$

where, for repeat ρ and fold k, $\mathcal{I}_{\rho k}$ and $\mathcal{V}_{\rho k}$ are the disjoint index sets of the training and validation patients produced by stratified, patient-grouped F-fold cross-validation (CV), $(\cdot)_{\mathcal{I}_{\rho k}}$ denotes restriction to those rows, and $f_{\rho k}^{(a)}$ is the classifier fitted on $\big(T_a(X)_{\mathcal{I}_{\rho k}},\, y_{\mathcal{I}_{\rho k}}\big)$. Here, we set $\phi = 2$ with $\mu_1 :=$ area under the receiver operating characteristic curve (AUC) and $\mu_2 :=$ sensitivity.

Three assumptions of Equation 3 carry through the paper. First, each repeat ρ draws a patient-level class-balanced subsample before train-test splitting, so every CV fold is balanced by construction and μ_ϕ is not inflated by class imbalance. Second, every method in $\bigcup_s \mathcal{M}_s$ is a per-spectrum operator, so applying T_a before splitting is equivalent to applying it within each fold and does not introduce information leakage. In contrast, feature scaling is not a per-spectrum operation and is therefore fitted separately within each fold. Third, fold assignments are seeded, making $\mathbf{r}(a)$ deterministic in a. Re-evaluating an action returns a bit-identical vector and yields no new information.

3 PROPOSED METHOD

In our proposed setup, an LLM-based multi-agent system explores preprocessing algorithms and their parameter domains to iteratively design preprocessing pipelines and optimise classification performance. At each round, a *proposer* agent proposes candidate pipelines from a structured record of the evaluated history, denoted *dossier*, which includes previously top-performing pipelines, rejected pipelines, LLM predictions and data measures (Sections 3.1 and 3.2). Each proposal is checked against a registry of admissible parameter ranges before it enters the action space. Admitted candidates

Algorithm 1 Agentic-guided multi-reward racing

Input: libraries $\{\mathcal{M}_s\}$, bounds $\{\Theta_m\}$, evaluation budget T, proposal batch size B, admission cap C, tolerance ϵ, operator op
$\mathcal{A}_0 \leftarrow$ seed pool of randomly parameterised pipelines drawn from $\{\mathcal{M}_s\}$
$\mathcal{B} \leftarrow \emptyset$ ▷ $\mathcal{B}$: blocked method combinations
$\mathcal{M}em \leftarrow \emptyset$ ▷ memory state $\mathcal{M}em$: observations, predictions, invariants
EVALUATE(identity pipeline) ▷ bootstrap reference, no LLM call
while evaluations used $< T$ **do**
  $\mathcal{D}_t \leftarrow$ BUILDDOSSIER$\big($top-8 played, last-8 retired, $\{\Theta_m\}$, last-10 predictions, last-3 invariants$\big)$
  $P_t \leftarrow \text{LLM}_{\text{proposer}}(\mathcal{D}_t,\ B)$ ▷ first agent is called
  $P_t^v \leftarrow \{\, p \in P_t : \text{VALID}(p, \Theta) \wedge p \notin \mathcal{A}_{t-1} \wedge \text{FAMILY}(p) \notin \mathcal{B} \,\}$
  $A_t \leftarrow \text{ADMIT}(P_t^v,\ C)$
  $\mathcal{A}_t \leftarrow \mathcal{A}_{t-1} \cup A_t$
  if $A_t = \emptyset$ **then** ▷ no novel valid candidate this round
    $A_t \leftarrow \{\, a \,\}$, a drawn uniformly from $\{a \in \mathcal{A}_{t-1} : n_t(a) = 0\}$, or from $\mathcal{A}_{t-1}$ if that set is empty ▷ fallback, no prediction
  end if
  for all $a \in A_t$ **do**
    $\mathbf{r} \leftarrow \text{EVALUATE}(a)$ ▷ Equation 3, or $\perp$ on failure
    if $\mathbf{r} = \perp$ **then**
      $\mathcal{B} \leftarrow \mathcal{B} \cup \{\text{FAMILY}(a)\},\ \mathcal{A}_t \leftarrow \mathcal{A}_t \setminus \{a\}$ ▷ $\mathcal{B}$ blocks line 8, not the pool
    else
      update $\hat{\mathbf{r}}_t(a),\ n_t(a)$, append observation and prediction to $\mathcal{M}em$
      $\mathcal{A}_t \leftarrow \mathcal{A}_t \setminus \big\{a' : \text{op}\big(\hat{\Delta}_t(a')\big) > \epsilon + \hat{\sigma}_t(a')\big\}$ ▷ Equations 4–6
    end if
  end for
  append $\text{LLM}_{\text{reflector}}(A_t,\ \mathbf{r})$ to $\mathcal{M}em$ ▷ second agent is called, producing one invariant
end while
return $\arg\max_a \big(\hat{r}_{T,1}(a),\ \hat{r}_{T,2}(a)\big)$ ordered lexicographically ▷ AUC, then sensitivity as the tie-break

are evaluated by CV, and those whose aggregated gap to the top pipelines exceeds a fixed tolerance level are rejected based on a coherent risk operator racing rule over AUC and sensitivity (Section 3.3). A *reflector* agent, then collects performance and pipeline information from the round and summarises it into a short *invariant* carried into the next *dossier*. The *dossier* is then updated with the new evidence and used to guide subsequent proposals. This process continues until a fixed budget of evaluations is exhausted. The full process is summarised in Algorithm 1.

3.1 STRUCTURE OF THE DOSSIER

A key component of our approach is our *dossier*, which is inspired by the Optimization by PROmpting (OPRO)'s meta-prompt by Yang et al. (2024). The *dossier* is built from scratch each round and captures preprocessing pipelines have been proposed, past predictions, reasoning, and evaluation history. These are organised into eight sections as follows: the task and remaining budget (§1); the top-ranked evaluated pipelines with their scores (§2); the pipelines that have been rejected/retired together with the reason for rejection/retirement (§3, see following sections); the top-performing methods m_s for each stage s, together with the *reflector*'s *invariants* (§4, see Section 3.2); the parameter domains Θ_m (§5); model predictions paired with realised values (§6), spectral diagnostics for the current best (§7); and the output contract (§8), which declares the method libraries $\mathcal{M}_s$, the JSON schema each proposal must satisfy, and the rules constraining it. Figure 5 and Table 2 in Appendix A provide further information about the structure of the *dossier* and examples for each section. §7 is the only section containing data-derived measurements rather than search bookkeeping, covering class separability, shape preservation and peak structure (see Table 3 in Appendix A). Three *dossier* sections are capped to keep the *dossier* bounded in size: §2 contains the eight best-scoring pipelines, §3 contains the eight most recent retirements, and §6 contains the ten most recent predictions.

3.2 THE PROPOSER AND REFLECTOR AGENTS

At each round t, a dossier $\mathcal{D}_t$ is built and passed to the *proposer* LLM agent, which generates a batch of B pipeline candidates, P_t, constructed using RamanSPy (Georgiev et al., 2024). Unlike

in the original OPRO implementation by Yang et al. (2024), the generation is not trusted and each generated candidate pipeline p is parsed and checked against the parameter domains Θ. If a violation is detected, the candidate p is rejected with a logged reason, so that an invalid configuration can never enter the active set $\mathcal{A}_t \subseteq \mathcal{A}$ as a valid pipeline a. Up to C pipelines are admitted to $\mathcal{A}_t$ each round, which are individually evaluated using patient-level F-fold cross-validation, based on the reward defined in Equation 3. Note that the *proposer* is used before the pipelines are scored, so it never sees the outcome of its own batch directly.

After each pipeline in the active set is evaluated, our second, *reflector* LLM agent receives a one-line summary for each admitted pipeline, containing the predicted and observed AUC and sensitivity, the signed error, and a justification the *proposer* had offered. The *reflector* model is then asked to produce a single statement of at most twenty words, which we call an *invariant*, capturing what the *proposer* should not change on the next round. To limit error propagation through free-text memory, we constrain the generation as follows: outputs are capped at 256 tokens, reduced to the first non-empty line, and discarded if they begin with deliberative markers (e.g. *okay*, *let's*), or return the sentinel `NONE` for inconclusive batches.

A persistent memory store $\mathcal{M}em$ retains the reward vector measured for every evaluated pipeline, every predicted score paired with the actual score, and every accepted *invariant*. It reaches the model through exactly two reads, both ordered by recency rather than relevance: the ten most recent prediction and outcome pairs populate §6, and the three most recent *invariants* populate §4. The model never queries the store $\mathcal{M}em$, so an *invariant* is the only channel by which a conclusion from round t reaches round $t+1$ as language rather than as a number (see Figures 5 and 6 in Appendix A). At the end of each round, we clear each *dossier* section and re-run the search, so the contribution of the accumulated evidence is measured rather than assumed. The search continues until the predefined budget T of CV evaluations is reached. The number of rounds is an outcome, since a round contributes between zero and C evaluations.

3.3 Multi-reward racing

Each round, pipelines whose gap to the best-performing candidates is large enough are retired. We quantify this using the per-reward gap to the incumbent leaders, which we define as follows, inspired by Dann et al. (2023):

$$\hat{\Delta}_t(a) \;=\; \Big(\max_{b\in\mathcal{A}_t} \hat{r}_{t,1}(b) - \hat{r}_{t,1}(a),\; \ldots,\; \max_{b\in\mathcal{A}_t} \hat{r}_{t,m}(b) - \hat{r}_{t,m}(a)\Big). \tag{4}$$

We aggregate the calculated per-reward gaps $\hat{\Delta}_t(a)$ using an operator $\mathrm{op} : \mathbb{R}^m \to \mathbb{R}$. This must be a coherent risk measure, meaning monotone, positively homogeneous, sub-additive and translation invariant, and we use the conditional value-at-risk family,

$$\mathrm{op} := \mathrm{CVaR}_\alpha(x) \;=\; \min_{c\in\mathbb{R}} \Big\{ c + \frac{1}{1-\alpha}\cdot\frac{1}{m}\sum_{i=1}^{m}[x_i - c]_+ \Big\}, \tag{5}$$

which recovers the mean at $\alpha = 0$ and the maximum as $\alpha \to 1$. Here c is an auxiliary threshold of the Rockafellar-Uryasev variational form (Rockafellar & Uryasev, 2000), eliminated by the minimisation. The objective is convex and piecewise linear in c, with breakpoints at the entries of x, so minimising over c is exact. Choosing $\mathrm{op} = \max$ retires an action as soon as any single reward shows it to be suboptimal, while $\mathrm{op} = \mathrm{mean}$ requires the average gap to be large.

In our setup, an action is retired when its aggregated gap exceeds a fixed tolerance ϵ plus its own empirical dispersion $\hat{\sigma}_t(a)$, i.e.:

$$\mathrm{op}\big(\hat{\Delta}_t(a)\big) \;>\; \epsilon + \hat{\sigma}_t(a), \qquad \hat{\sigma}_t(a) = \max_i \; \frac{s_i(a)}{\sqrt{n_t(a)}}, \tag{6}$$

where $s_i(a) := sd(r_i(a))$ denotes the sample standard deviation of the observed rewards $r_i(a)$, computed over the $n_t(a)$ evaluations of pipeline a up to round t. An action with $n_t(a) < 2$ has no estimable spread and is given a prior $\sigma_0 = 0.05$, set on the scale of the spread between pipelines rather than that of the reward range. Since evaluation is deterministic across all our experiments, no action is evaluated more than once, so $n_t(a) \in \{0, 1\}$ and thus $\hat{\sigma}_t(a) = \sigma_0$ and $\mathrm{op}\big(\hat{\Delta}_t(a)\big) = \epsilon + \sigma_0$ for all a and t.

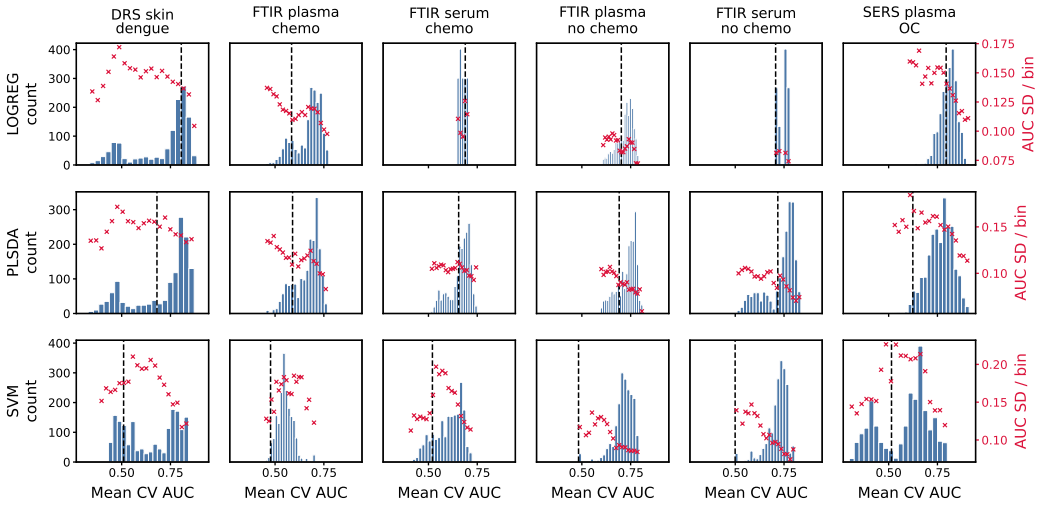


Figure 2: Distribution of classification performance across $\sim 2,500$ preprocessing pipelines evaluated during *grid brute force* search for each of the 18 scenarios considered. Dashed lines indicate performance on raw data. Red crosses show the standard deviation of AUC values for each bin.

3.4 Failed evaluations are missing data

A pipeline may crash, time out, or produce a non-finite metric (e.g. when using an unstable baseline fit). Such an outcome provides limited information about the quality of the pipeline and is thus recorded as a blank record $\perp$ rather than as a score, to avoid assigning a non-running pipeline a higher rank (e.g. 0.5) than a pipeline that ran but scored below chance. On $\perp$, we record no reward, admit no memory observation, and drop the action. Its method combination enters the list of blocked actions $\mathcal{B}$, and no further parameterisation of it may be proposed, since a combination that fails numerically usually fails across its parameter range. The block applies to admission and not to the pool.

4 Experimental setup

Datasets. We performed experiments on four publicly available clinical datasets, spanning three spectroscopic techniques, two disease areas and three cohort sizes: (i) surface-enhanced Raman spectroscopy (SERS) measurements from plasma samples from healthy and ovarian cancer patients (Paraskevaidi et al., 2018), (ii-iii) attenuated total reflection (ATR)-FTIR measurements from plasma (ii) and serum (iii) samples from healthy and ovarian cancer patients (treated and non-treated with chemotherapy) (Giamougiannis et al., 2021a;b), and (iv) DRS measurements from skin for dengue screening (Poh et al., 2020). Further information about cohort sizes and spectral acquisition is provided in Table 4 in Appendix B. Each dataset was re-derived from the deposited raw files, prior to any preprocessing. When multiple measurements were available for a patient, these were averaged before preprocessing and classification. We performed experiments on six dataset combinations of modality, specimen and task, using three classifier families – a radial basis function (RBF)-kernel, support vector machine (SVM), partial least squares discriminant analysis (PLS-DA) and ℓ_2-regularised logistic regression (LogReg).

Protocol. For each of the 18 scenarios, we ran three independent agentic campaigns and performed preprocessing optimisation using three different LLM backbones for the *proposal* and *reflector* models: `deepseek-r1:8b`, `llama3.1:8b` and `llama3.1:70b`. Each campaign was run with a budget of 30 iterations, proposal batch size $B = 10$, admission cap $C = 5$, $\mathrm{op} = \mathrm{mean}$, $\epsilon = 0.01$, $\sigma_0 = 0.05$, and $R = 5$ repeats of $F = 5$ CV-folds performed using different random keys. All LLMs were run locally using a temperature of 0 over 2,160 evaluations across the 18 scenarios. To allow meaningful comparisons, all six methods were evaluated using the same metrics and CV splits.

Table 1: Benchmark results. Top-1 classification performance achieved by each preprocessing optimisation strategy across the 18 scenarios. Values report mean cross-validated AUC over five repeats. Best results are shown in **bold**. Wall times were measured on a MacBook M4 Max laptop, except for llama3.1 70B (*), which was measured on a separate workstation (1× RTX Pro 6000).

Scenario	Raw	Stepwise	Grid	deepseek-r1:8b	llama3.1:8b	llama3.1:70b
DRS (skin, LogReg)	0.806	0.873	0.886	**0.893**	0.863	0.886
DRS (skin, PLS-DA)	0.681	0.868	0.872	**0.880**	0.873	0.877
DRS (skin, SVM)	0.510	0.819	0.843	0.863	0.858	**0.867**
ATR-FTIR (plasma, +chemo, LogReg)	0.585	0.741	0.775	**0.809**	0.775	0.771
ATR-FTIR (plasma, +chemo, PLS-DA)	0.590	**0.768**	**0.768**	0.749	0.749	0.761
ATR-FTIR (plasma, +chemo, SVM)	0.477	0.595	**0.706**	0.680	0.683	0.644
ATR-FTIR (plasma, no chemo, LogReg)	0.702	0.726	0.792	0.811	**0.839**	0.832
ATR-FTIR (plasma, no chemo, PLS-DA)	0.691	0.766	0.813	0.829	0.820	**0.839**
ATR-FTIR (plasma, no chemo, SVM)	0.483	0.774	0.792	0.820	**0.829**	0.828
ATR-FTIR (serum, +chemo, LogReg)	0.689	0.705	0.705	**0.769**	0.747	0.746
ATR-FTIR (serum, +chemo, PLS-DA)	0.655	0.742	0.750	**0.756**	0.743	0.740
ATR-FTIR (serum, +chemo, SVM)	0.520	0.659	**0.724**	0.700	0.653	0.663
ATR-FTIR (serum, no chemo, LogReg)	0.707	0.781	0.781	0.820	0.795	**0.836**
ATR-FTIR (serum, no chemo, PLS-DA)	0.718	0.801	0.836	**0.846**	0.826	0.819
ATR-FTIR (serum, no chemo, SVM)	0.499	0.779	0.805	**0.823**	0.822	0.798
SERS (plasma, LogReg)	0.795	0.893	0.915	**0.929**	0.889	0.880
SERS (plasma, PLS-DA)	0.624	0.875	**0.913**	0.881	0.879	0.890
SERS (plasma, SVM)	0.515	0.791	0.801	**0.837**	0.816	0.771
Mean	0.625	0.775	0.804	**0.816**	0.803	0.803
Wall times	-	20 min	4 days	38.2 hours	10.6 hours	11.4 hours*

Baselines. We benchmark our method against three baselines: (i) no preprocessing (**raw**); (ii) sequential stage-wise optimisation, where each preprocessing stage is optimised in turn and the best choice is fixed before proceeding to the next (**greedy stepwise**); and (iii) a full exhaustive evaluation of all pipelines over a predefined discrete parameter grid (**grid brute force**). Note that the number of pipelines considered by *grid brute force* grows multiplicatively with the number of preprocessing algorithms and parameter choices: we perform up to $\sim 2{,}500$ evaluations for each scenario, compared with only ~ 45 for *greedy stepwise*.

5 Results

5.1 Bad preprocessing is worse than no preprocessing

First, we examine the effect of preprocessing on downstream classification. Figure 2 shows the distribution of classification performance across all preprocessing pipelines evaluated in the exhaustive *grid brute force* search and all 18 scenarios. We make the following two observations. First, the raw-data baseline is not a lower bound on performance: depending on the scenario, between 0.7% and 65% of the tested pipelines perform worse than the identity pipeline (i.e. no preprocessing). Hence, preprocessing is not guaranteed to improve classification performance and can just as readily degrade it. Second, dispersion generally decreases as mean performance increases, with the noisiest evaluations concentrated in the lower-performing pipelines, which an adaptive search should discard.

5.2 Agentic optimisation outperforms grid search and stepwise optimisation

Building on these results, we benchmark our agentic system against the three baseline methods on all 18 scenarios. Table 1 summarises the results, showing the classification performance of the best preprocessing pipeline selected by each optimisation strategy for each scenario (extended results are provided in Figures 7 and 8 in Appendix C). Averaged across the 18 scenarios, the three agentic campaigns achieved AUCs of 0.816 (`deepseek-r1:8b`), 0.803 (`llama3.1:8b`) and 0.803 (`llama3.1:70b`), compared with 0.625 for raw data, 0.775 for the *greedy stepwise* optimisation, and 0.804 for *grid brute force*. The three agentic campaigns outperformed *greedy stepwise* optimisation in 17, 14 and 14 of the 18 scenarios, respectively, with mean improvements of $+0.041 \pm 0.006$, $+0.028 \pm 0.008$ and $+0.027 \pm 0.008$ AUC per scenario. The `deepseek-r1:8b` campaign also outperformed brute-force grid search in 14 of the 18 scenarios, with a mean improvement of $+0.012 \pm 0.006$ AUC. The other two agentic campaigns performed comparably to *grid brute*

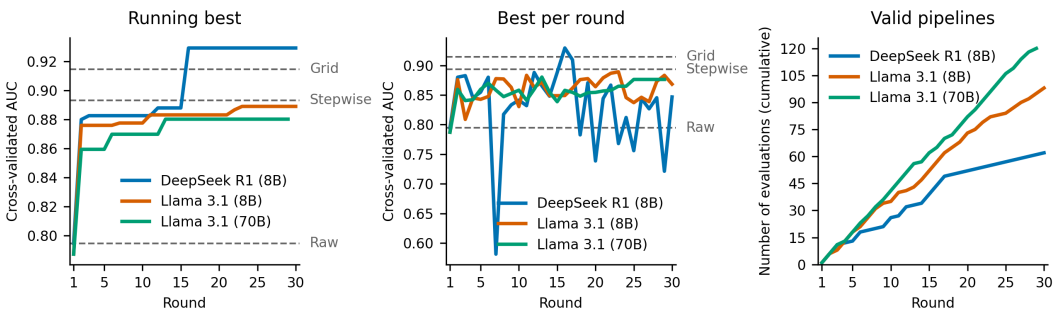


Figure 3: Classification performance across 30 rounds of preprocessing optimisation guided by agentic search on a selected scenario [SERS (plasma, LogReg)].

force, with mean differences of -0.001 ± 0.007 and -0.002 ± 0.008 AUC, respectively. Moreover, the agentic search required substantially fewer evaluations, with roughly 70 evaluations per campaign and scenario, compared with $\sim$ 2,500 pipelines for *grid brute force*. This resulted in runtimes of 10.6–38.2 hours versus four days for *grid brute force* (see Table 5 in Appendix C).

5.3 Preprocessing proposals improve as optimisation history accumulates

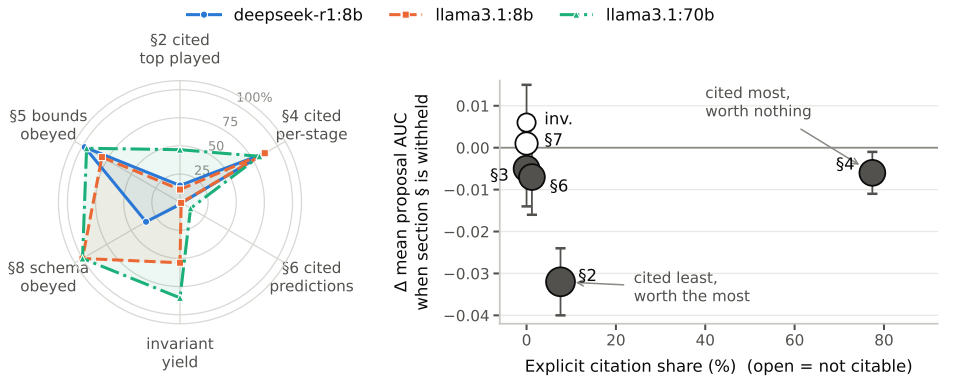
Next, we examine how proposal quality changes as optimisation history accumulates. Figure 3 shows the optimisation traces of our three agentic campaigns for a representative scenario [SERS (plasma, LogReg)], depicting the best proposals at each round (dotted lines) and the running best (solid lines). In this example, the agentic search with `deepseek-r1:8b` discovers the best preprocessing pipeline, reaching an AUC of 0.929 at round 16 and outperforming *greedy stepwise* optimisation (0.893). However, note that its performance varies considerably across rounds, with many proposals falling well below the best pipeline found so far. In contrast, the two `llama3.1` campaigns improve more steadily but plateau at AUCs of 0.889 (8B) and 0.880 (70B). The best preprocessing pipelines derived by each of the three agentic campaigns on this scenario are given in Table 6 in Appendix C.

Extended analysis across all 18 scenarios shows that proposal quality generally improves as optimisation progresses, with a mean AUC of LLM-proposed pipelines increasing from 0.686 in the first third of the campaigns to 0.720 in the final third. In fact, performance improves in 16 out of the 18 scenarios ($p = 0.001$, sign test), with a positive correlation with round index in 14 of 18 ($p = 0.031$, sign test) (see Table 7 in Appendix C). However, this improvement comes with reduced search diversity, with the proportion of distinct method families across proposals decreasing from 52.5% to 40.0% on average during the optimisation ($p = 0.001$, sign test). Additionally, prediction calibration does not significantly improve, with the mean absolute error between predicted and observed performance only decreasing from 0.076 to 0.068 ($p = 0.096$, sign test). The optimisation traces for all 18 scenarios are provided in Figure 9 in Appendix C.

5.4 Analysis of model reasoning, size and the importance of the dossier

To better understand the optimisation process and the components driving performance gains, we conduct a series of ablation studies examining model reasoning, the contribution of each *dossier* section, and the effect of LLM size. We summarise our main findings below.

Citation frequency does not reflect section importance. First, we examine how the *proposer* agent uses the information in the *dossier*. Figure 4 shows the proportion of proposals that obeyed (i.e. hard constraints) or cited (i.e. used during candidate generation) each *dossier* section, alongside the effect on performance when sections are removed. We observe that the sections most frequently used by the agent are not necessarily those that contribute most to downstream performance. Further analysis over all 18 scenarios shows that the most frequently cited section of the *dossier* is the per-stage summary (§4), which appears in 81.5–87.0% of all proposals, compared with 11.2–46.6% for top-ranked pipelines (§2), 0–0.7% for rejected/retired pipelines (§3), and 1.5–10.9% for model predictions (§6) (see Table 8 in Appendix D). Yet removing §4 has little effect on performance (-0.006 ± 0.005 AUC), whereas removing §2 substantially hurts performance (-0.032 ± 0.008). In



(a) dossier profile by campaign

(b) Leave-One-Out section § impact on performance

what the model is shown, one real line per citable section

§2 top played, cap 8	1	WhitakerHayes(kernel_size=7,..)--SavGol(..)--Poly(..)	0.876	0.801	
§4 per-stage dominance	despike	WhitakerHayes	0.826	4	"..should not re-derive the dominance of White
§6 predictions vs actuals, cap 10	2	WhitakerHayes(..)--Poly(..)	0.73	0.88	+0.15	"exploits §4.."

Figure 4: Analysis of the reasoning of the *proposal* model and the importance of the *dossier*. (a) Proportion of proposals obeying/citing each *dossier* section. (b) Change in LLM AUC predictive performance when a *dossier* section is dropped.

fact, dropping §2 is the only ablation to reduce performance in all four paired runs (see Table 9 in Appendix D), highlighting that the record of top-performing candidates is key for guiding the search

The *dossier* is key for generating valid pipelines. We next examine which information in the *dossier* enables the generation of valid pipelines. We observe that when given only the task description, models return valid pipelines in only 0.8% of the time (see Table 9 in Appendix D). Adding the output contract yields well-formed outputs, but all $5{,}668$ proposed pipelines were still rejected as they contained unrecognised parameter names rather than out-of-range values. Adding section §5 resolved this issue, with only 192 rejections that were genuine constraint violations. This suggests that the registry supplies the API vocabulary required to generate valid pipelines, which, together with incorporating relevant parameter bounds during the optimisation process, substantially reduces the number of invalid parameters and thus pipelines explored.

Increasing model size improves calibration, but not optimisation performance. The 70B `llama3.1` model outperformed its smaller counterpart (8B) only in 6 of 18 scenarios, with no significant difference in achieved AUC (mean difference -0.006 AUC, Wilcoxon $p = 0.25$). However, mean absolute prediction error decreases from 0.071 to 0.041 between 70B and 8B ($p = 5.3 \times 10^{-5}$, signed test), and the rank correlation between predicted and measured AUC goes from $+0.507$ to $+0.791$ (see Table 10 and Figure 10 in Appendix D). Prediction errors were concentrated in the baseline correction stage (see Table 11 in Appendix D), suggesting a systematic pipeline-specific bias rather than a model-specific effect. Thus, in our experiments, model scale improves candidate assessment but does not yield better pipelines within the search budget.

6 Conclusion

In conclusion, we have developed here an LLM-based agentic system for automated preprocessing optimisation in spectroscopy. We benchmarked our framework on 18 scenarios across 3 spectroscopic modalities, and showed that it matches the performance of exhaustive grid search while evaluating fewer than 4% as many pipelines (four days reduced to 11 hours), and outperforms sequential stepwise optimisation in 17 of 18 scenarios. We empirically show that our *dossier* infrastructure is a key contributor to performance, guiding pipeline generation, optimisation and search. Looking ahead, our system provides a proof of concept for more autonomous, reasoning-based workflows in spectroscopy.

AI USE STATEMENT

In this work, we used generative AI tools to assist with code organization, formatting, and debugging. We have not used generative AI tools to generate or modify the scientific methodology, experimental design, data, results, analysis, or conclusions.

Additionally, generative AI tools were used to assist with code documentation and presentation. All AI-assisted code and content were reviewed, tested, and verified by the authors. We take full responsibility for the final content of this work, including all text, claims, code, results, and other artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

All datasets used in this study are publicly available and are listed below with their corresponding sources: (i) SERS dataset from Paraskevaidi et al. (2018) [https://figshare.com/articles/dataset/Raman_spectroscopic_techniques_to_detect_ovarian_cancer_biomarkers_in_blood_plasma/6744206]; (ii) ATR-FTIR (plasma) from Giamougiannis et al. (2021b) [https://figshare.com/articles/dataset/Raman_Ovarian_Cancer/14562312]; (iii) ATR-FTIR (serum) from Giamougiannis et al. (2021a) [https://figshare.com/articles/dataset/Raman_Ovarian_Cancer/14562312]; (iv) DRS dataset from Poh et al. (2020) [https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0228923].

All experiments were performed using fixed random seeds to ensure reproducibility. Each experiment was repeated five times, using the same set of random seeds across all methods to ensure a consistent basis for comparison. Unless otherwise specified, data splits and all other experimental settings were kept identical across methods.

For experiments involving LLMs, we used the following models: (i) DeepSeek-R1:8B, (ii) Llama 3.1:8B, and (iii) Llama 3.1:70B, as described in Section 4. All models were run using the same experimental pipeline and evaluation protocol. Relevant parameters and inference settings are reported in the corresponding experimental sections.

Experiments were performed on the following hardware configurations: (i) workstation: AMD Ryzen Threadripper PRO 7975WX (up to 5.3 GHz), 256 GB DDR5 RAM, 4 TB M.2 storage in RAID 1, 24 TB HDD storage in RAID 1, and NVIDIA RTX PRO 6000 GPU; (ii) laptop: Apple MacBook Pro with M4 Max chip and 48 GB unified memory.

REFERENCES

Nils Kristian Afseth, Vegard Herman Segtnan, and Jens Petter Wold. Raman spectra of biological samples: A study of preprocessing methods. *Applied Spectroscopy*, 60(12):1358–1367, 2006.

Matthew J Baker, Júlio Trevisan, Paul Bassan, Rohit Bhargava, Holly J Butler, Konrad M Dorling, Peter R Fielden, Simon W Fogarty, Nigel J Fullwood, Kelly A Heys, et al. Using Fourier transform ir spectroscopy to analyze biological materials. *Nature Protocols*, 9(8):1771–1791, 2014.

Matthew J Baker, Hugh J Byrne, John Chalmers, Peter Gardner, Royston Goodacre, Alex Henderson, Sergei G Kazarian, Francis L Martin, Julian Moger, Nick Stone, et al. Clinical applications of infrared and Raman spectroscopy: state of play and future challenges. *Analyst*, 143(8):1735–1757, 2018.

Thomas Bocklitz, Angela Walter, Katharina Hartmann, Petra Rösch, and Jürgen Popp. How to pre-process Raman spectra for reliable and stable models? *Analytica Chimica Acta*, 704(1-2): 47–56, 2011.

Daniil A. Boiko, R. MacKnight, Ben Kline, and Gabe Gomes. Autonomous chemical research with large language models. *Nature*, 624(7992):570–578, 2023.

Benjamin Brunel, Fatima Alsamad, and Olivier Piot. Toward automated machine learning in vibrational spectroscopy: Use and settings of genetic algorithms for pre-processing and regression optimization. *Chemometrics and Intelligent Laboratory Systems*, 219:104444, 2021.

Holly J Butler, Lorna Ashton, Benjamin Bird, Gianfelice Cinque, Kelly Curtis, Jennifer Dorney, Karen Esmonde-White, Nigel J Fullwood, Benjamin Gardner, Pierre L Martin-Hirsch, et al. Using Raman spectroscopy to characterize biological materials. *Nature Protocols*, 11(4):664–687, 2016.

Holly J Butler, Benjamin R Smith, Robby Fritzsch, Pretheepan Radhakrishnan, David S Palmer, and Matthew J Baker. Optimised spectral pre-processing for discrimination of biofluids via ATR-FTIR spectroscopy. *Analyst*, 143(24):6121–6134, 2018.

Torre M Bydlon, Rami Nachabé, Nimmi Ramanujam, Henricus JCM Sterenborg, and Benno HW Hendriks. Chromophore based analyses of steady-state diffuse reflectance spectroscopy: current status and perspectives for clinical adoption. *Journal of Biophotonics*, 8(1-2):9–24, 2015.

Albert Cerussi, David Hsiang, Natasha Shah, Rita Mehta, Amanda Durkin, John Butler, and Bruce J Tromberg. Predicting response to breast cancer neoadjuvant chemotherapy using diffuse optical spectroscopy. *Proceedings of the National Academy of Sciences*, 104(10):4014–4019, 2007.

Angelica Chen, Samuel Don Stanton, Frances Ding, Robert G Alberstein, Andrew Martin Watkins, Richard Bonneau, Vladimir Gligorijevic, Kyunghyun Cho, and Nathan C. Frey. Generalists vs. specialists: Evaluating LLMs on highly-constrained biophysical sequence optimization tasks. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 9029–9072. PMLR, 2025.

Nicolas Coca-Lopez, Victor Alcolea-Rodriguez, Miguel A Bañares, Sandor Brockhauser, Julien Gorenflot, Alex Henderson, Ron Hildebrandt, Nina Jeliazkova, Nikolay Kochev, Enrique Lozano Diz, et al. Artificial intelligence-powered Raman spectroscopy through open science and FAIR principles. *ACS Nano*, 19(44):38189–38218, 2025.

Lili Cong, Jiaqi Wang, Sijun Huang, Xiaxia Man, Yi Guo, Shuping Xu, and Songling Zhang. Machine Learning-Enhanced Analysis of Exosomal Surface Sialic Acid Using Surface-Enhanced Raman Spectroscopy for Ovarian Cancer Diagnosis and Therapeutic Monitoring. *Advanced Science*, 13 (18):e18190, 2026.

Christoph Dann, Yishay Mansour, and Mehryar Mohri. Reinforcement learning can be more efficient with multiple rewards. In *International Conference on Machine Learning*, pp. 6948–6967. PMLR, 2023.

Jiashun Fu, Xuyang Liu, Wensheng Cai, Haohao Fu, and Xueguang Shao. SpeLL: An Agent for Natural Language-Driven Intelligent Spectral Modeling. *Journal of Chemical Information and Modeling*, 65(15):7844–7850, August 2025. ISSN 1549-9596. doi: 10.1021/acs.jcim.5c01236.

Dimitar Georgiev, Simon Vilms Pedersen, Ruoxiao Xie, Álvaro Fernández-Galiana, Molly M. Stevens, and Mauricio Barahona. RamanSPy: An Open-Source Python Package for Integrative Raman Spectroscopy Data Analysis. *Analytical Chemistry*, 96(21):8492–8500, 2024.

Jan Gerretzen, Ewa Szymańska, Jeroen J Jansen, Jacob Bart, Henk-Jan van Manen, Edwin R Van Den Heuvel, and Lutgarde MC Buydens. Simple and effective way for data preprocessing selection based on design of experiments. *Analytical Chemistry*, 87(24):12096–12103, 2015.

Panagiotis Giamougiannis, Camilo LM Morais, Rita Grabowska, Katherine M Ashton, Nicholas J Wood, Pierre L Martin-Hirsch, and Francis L Martin. A comparative analysis of different biofluids towards ovarian cancer diagnosis using Raman microspectroscopy. *Analytical and Bioanalytical Chemistry*, 413(3):911–922, 2021a.

Panagiotis Giamougiannis, Camilo LM Morais, Brice Rodriguez, Nicholas J Wood, Pierre L Martin-Hirsch, and Francis L Martin. Detection of ovarian cancer ($\pm$neo-adjuvant chemotherapy effects) via ATR-FTIR spectroscopy: comparative analysis of blood and urine biofluids in a large patient cohort. *Analytical and Bioanalytical Chemistry*, 413(20):5095–5107, 2021b.

Shuxia Guo, Jürgen Popp, and Thomas Bocklitz. Chemometric analysis in Raman spectroscopy from experimental design to machine learning–based modeling. *Nature Protocols*, 16(12):5426–5459, 2021.

Abigail S Haka, Karen E Shafer-Peltier, Maryann Fitzmaurice, Joseph Crowe, Ramachandra R Dasari, and Michael S Feld. Diagnosing breast cancer by using Raman spectroscopy. *Proceedings of the National Academy of Sciences*, 102(35):12371–12376, 2005.

Sung-Hyuk Han, Jiyun Park, Kwang-Bae Lee, Cheol-Hwan Park, and Dong-Yup Lee. Variable-specific preprocessing enables cross-scale transferable Raman models from high-throughput bioreactor data. *Bioresources and Bioprocessing*, 13(1):122, 2026.

Todd C Hollon, Balaji Pandian, Arjun R Adapa, Esteban Urias, Akshay V Save, Siri Sahib S Khalsa, Daniel G Eichberg, Randy S D'Amico, Zia U Farooq, Spencer Lewis, et al. Near real-time intraoperative brain tumor diagnosis using stimulated Raman histology and deep neural networks. *Nature Medicine*, 26(1):52–58, 2020.

M Isabelle, J Dorney, A Lewis, GR Lloyd, O Old, N Shepherd, M Rodriguez-Justo, H Barr, K Lau, I Bell, et al. Multi-centre Raman spectral mapping of oesophageal cancer tissues: a study to assess system transferability. *Faraday Discussions*, 187:87–103, 2016.

Roger M Jarvis and Royston Goodacre. Genetic algorithm optimization for pre-processing and variable selection of spectroscopic data. *Bioinformatics*, 21(7):860–868, 2005.

Michael Jermyn, Kelvin Mok, Jeanne Mercier, Joannie Desroches, Julien Pichette, Karl Saint-Arnaud, Liane Bernstein, Marie-Christine Guiot, Kevin Petrecca, and Frederic Leblond. Intraoperative brain cancer detection with Raman spectroscopy in humans. *Science Translational Medicine*, 7 (274):274ra19, 2015.

Mykyta Kizilov, Vsevolod Cheburkanov, Joseph Harrington, and Vladislav V Yakovlev. Modular and automated workflow for streamlined Raman signal analysis. *Journal of Raman Spectroscopy*, 57 (2):360–377, 2026.

Loong Chuen Lee, Choong-Yeun Liong, and Abdul Aziz Jemain. Effects of data pre-processing methods on classification of ATR-FTIR spectra of pen inks using partial least squares-discriminant analysis (PLS-DA). *Chemometrics and Intelligent Laboratory Systems*, 182:90–100, 2018.

Jeonghyun Lim, Hyeonju Lee, and Dongha Shin. Artificial Raman Expert: Autonomous Spectroscopic Reasoning Driven by Localized Large Language Models. *ACS Sensors*, 11(6):4971–4979, 2026.

Qitan Lv, Tianyu Liu, and Hong Wang. Exploiting edited large language models as general scientific optimizers. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5212–5237, 2025.

Pingchuan Ma, Tsun-Hsuan Wang, Minghao Guo, Zhiqing Sun, Joshua B. Tenenbaum, Daniela Rus, Chuang Gan, and Wojciech Matusik. LLM and simulation as bilevel optimizers: A new paradigm to advance physical scientific discovery. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 33940–33962. PMLR, 2024.

Kevin Murphy. Model Discovery Agent: LLM-assisted Bayesian experiment design for data-efficient discovery of mechanistic world models, 2026. arXiv:2608.09696.

Allen Nie, Ching-An Cheng, Andrey Kolobov, and Adith Swaminathan. The Importance of Directional Feedback for LLM-based Optimizers, 2024. arXiv:2405.16434.

Heewoong Noh, Namkyeong Lee, Gyoung S. Na, Kibum Kim, and Chanyoung Park. IR-Agent: Expert-Inspired LLM Agents for Structure Elucidation from Infrared Spectra. In *The Fourteenth International Conference on Learning Representations*, 2026.

Maria Paraskevaidi, Katherine M Ashton, Helen F Stringfellow, Nicholas J Wood, Patrick J Keating, Anthony W Rowbottom, Pierre L Martin-Hirsch, and Francis L Martin. Raman spectroscopic techniques to detect ovarian cancer biomarkers in blood plasma. *Talanta*, 189:281–288, 2018.

Javier Plou, Pablo S Valera, Isabel Garcia, Carlos DL de Albuquerque, Arkaitz Carracedo, and Luis M Liz-Marzan. Prospects of surface-enhanced Raman spectroscopy for biomarker monitoring toward precision medicine. *ACS Photonics*, 9(2):333, 2022.

Abdul Halim Poh, Faisal Rafiq Mahamd Adikan, Mahmoud Moghavvemi, Sharifah Faridah Syed Omar, Khadijah Poh, Mohamad Badrol Hisyam Mahyuddin, Grace Yan, Mohammad Aizuddin Azizah Ariffin, and Sulaiman Wadi Harun. Precursors to non-invasive clinical dengue screening: Multivariate signature analysis of in-vivo diffuse skin reflectance spectroscopy on febrile patients in Malaysia. *PLoS One*, 15(4):e0228923, 2020.

Ximei Qian, Xiang-Hong Peng, Dominic O Ansari, Qiqin Yin-Goen, Georgia Z Chen, Dong M Shin, Lily Yang, Andrew N Young, May D Wang, and Shuming Nie. In vivo tumor targeting and spectroscopic detection with surface-enhanced Raman nanoparticle tags. *Nature Biotechnology*, 26(1):83–90, 2008.

Nian Ran, Yue Wang, Xiaoyuan Zhang, Zhongzheng Li, Qingsong Ran, Wenhao Li, and Richard Allmendinger. ExLLM: Experience-Enhanced LLM Optimization for Molecular Design and Beyond, 2025. arXiv:2502.12845.

Jan G. Rittig, Luise F. Kaven, and Philippe Schwaller. Agentic AI in Process Analytical Technology: LLM Assistants for Chemometric Workflows, 2026. chemrxiv.15007499.

R. Tyrrell Rockafellar and Stanislav Uryasev. Optimization of conditional value-at-risk. *The Journal of Risk*, 2(3):21–41, 2000.

Abraham Savitzky and Marcel JE Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, 36(8):1627–1639, 1964.

Emily E. Storey and Amr S. Helmy. Optimized preprocessing and machine learning for quantitative Raman spectroscopy in biology. *Journal of Raman Spectroscopy*, 50(7):958–968, 2019.

Jan Vališ, Markéta Fousková, Daniela Janstová, Lucie Habartová, Jaromír Petrtýl, Luboš Petruželka, Alla Synytsya, and Vladimír Setnička. Automated classification pipeline for real-time in vivo examination of colorectal tissue using Raman spectroscopy. *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy*, 313:124152, 2024.

Ravi Teja Vulchi, Volodymyr Morgunov, Rajendhar Junjuri, and Thomas Bocklitz. Artifacts and anomalies in Raman spectroscopy: a review on origins and correction procedures. *Molecules*, 29 (19):4748, 2024.

Joel Wahl, Elisabeth Klint, Martin Hallbeck, Jan Hillman, Karin Wårdell, and Kerstin Ramser. Impact of preprocessing methods on the Raman spectra of brain tissue. *Biomedical Optics Express*, 13 (12):6763–6777, 2022.

Zujie Xie, Zixuan Chen, Jiheng Liang, Xiangyang Yu, and Ziru Yu. LUMIR: An LLM-driven unified agent framework for multi-task infrared spectroscopy reasoning. *Analytica Chimica Acta*, 1382: 344857, 2026.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. Large Language Models as Optimizers. In *International Conference on Learning Representations*, volume 2024, pp. 12028–12068, 2024.

A SYSTEM ARCHITECTURE

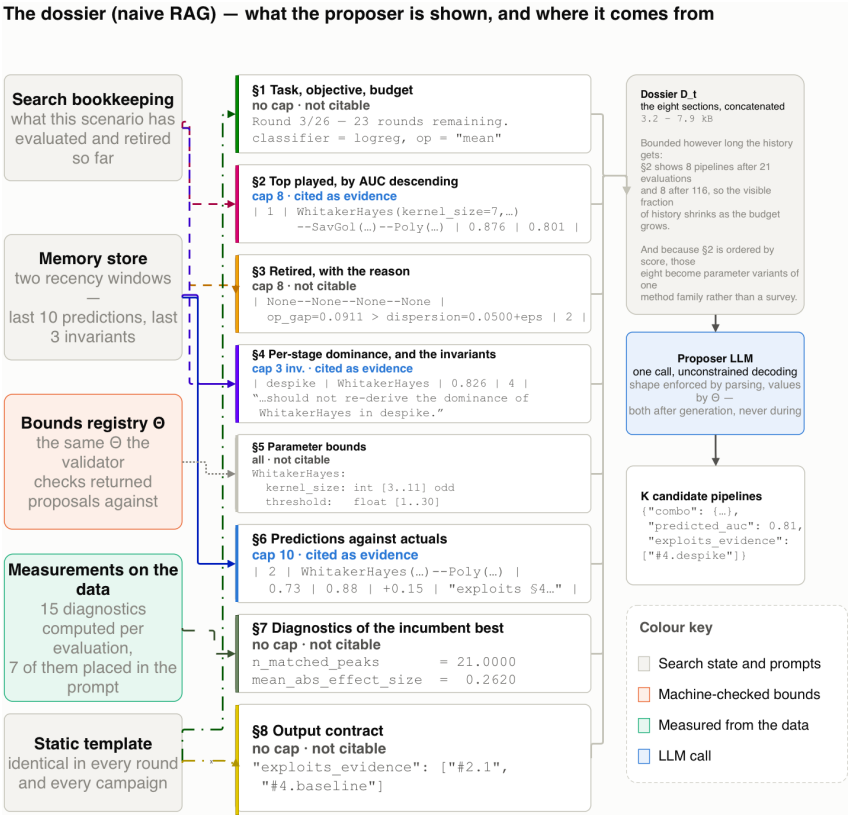


Figure 5: Anatomy of the dossier $\mathcal{D}_t$: the eight sections and the memory reads that populate them.

Table 2: Anatomy of the dossier. Cap is the number of entries retained. Cited is whether the content of the section has been explicitly cited by the Agent/LLM.

§	Contents	Cap	Cited?	Example line
1	Task, objective, budget	—	no	`Round 3/26 -- 23 rounds remaining. classifier = logreg, op="mean"`
2	Top played, by AUC desc	8	yes	`| 1 | WhitakerHayes(kernel_size=7,...) -SavGol(...)-Poly(...) | 0.876 | 0.801 |`
3	Retired, with reason	8	no*	`| None-None-None-None | op_gap=0.0911 > dispersion=0.0500+eps | 2 |`
4	Per-stage dominance, plus reflector invariants	3 inv.	yes	`| despike | WhitakerHayes | 0.826 | 4 |` "…should not re-derive the dominance of WhitakerHayes in despike."
5	Parameter bounds	all	no	`WhitakerHayes: kernel_size: int [3..11] odd threshold: float [1..30]`
6	Predictions vs. actuals	10	yes	`| 2 | WhitakerHayes(...)-Poly(...) | 0.73 | 0.88 | +0.15 | "exploits §4..." |`
7	Diagnostics of current best	—	no	`n_matched_peaks = 21.0000 mean_abs_effect_size = 0.2620`
8	Output contract	—	no	`"exploits_evidence": ["#2.1", "#4.baseline"]`

* §3 appears in the rules only as a prohibition ("do not repeat a combo from §2 or §3"), never as citable evidence.

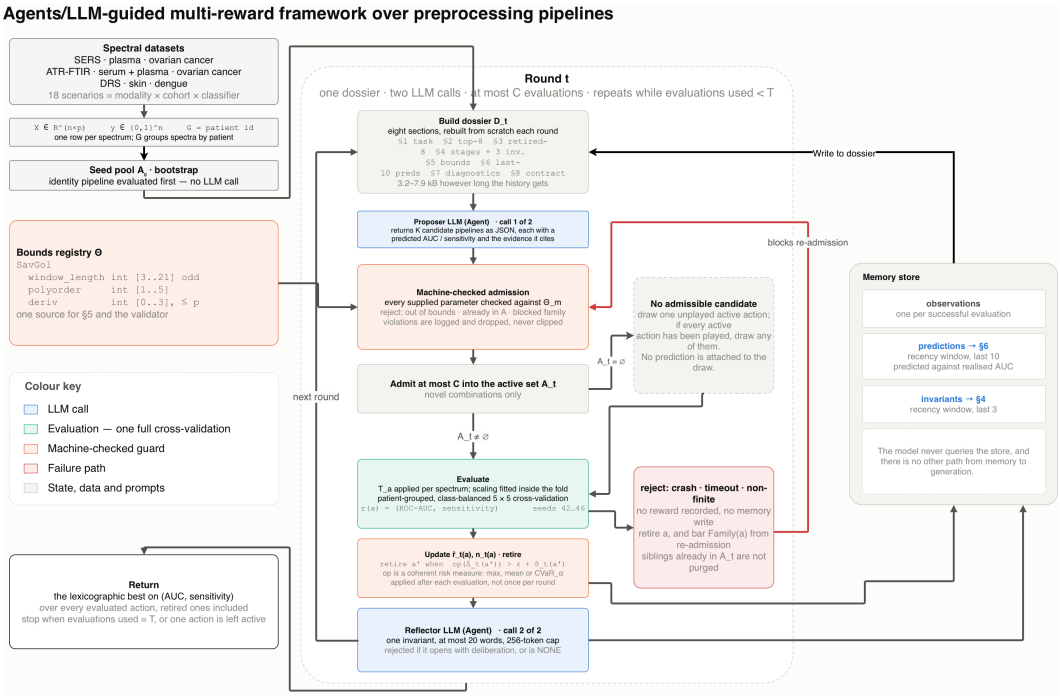


Figure 6: Architecture of the LLM-guided optimiser. The proposer, reflector, and memory store interact through the dossier rebuilt each round.

Table 3: The spectral diagnostics reported to the model in dossier §7, computed on the transformed matrix for the current best pipeline. Example values are for `WhitakerHayes-SavGol-Poly` on the SERS ovarian-cancer task.

Metric	What it measures	Example
Class separability: did the transform help?		
`mean_abs_effect_size`	mean |Cohen's d| over wavenumbers	0.2620
`max_abs_effect_size`	the single most discriminative channel	1.3076
`mean_abs_auc_deviation`	mean |per-channel AUC $-$ 0.5|	0.0768
Shape preservation: how much did it distort?		
`pearson_mean_spectra`	correlation of the two class-mean spectra	0.9883
`cosine_mean_spectra`	cosine similarity of the same pair	0.9883
Peak structure: did bands survive, and move?		
`n_matched_peaks`	peaks present in both class means	21
`mean_abs_peak_shift`	mean |Δposition|, axis units	0.5217

B DATASETS

Table 4: Extended information about the used datasets.

Modality	Specimen	rows	groups (task)	bands	loaded range
SERS (Paraskevaidi et al., 2018)	plasma	55	28 / 27	1187	500.0–1799.5 cm^{-1}
ATR-FTIR (Giamougiannis et al., 2021b)	plasma	423	307 / 71	852	900.3–1799.3 cm^{-1}
ATR-FTIR (Giamougiannis et al., 2021a)	serum	423	307 / 71	852	900.3–1799.3 cm^{-1}
DRS (Poh et al., 2020)	skin	222	194 / 28	2677	173.9–2494.9 nm

C EXTENDED RESULTS

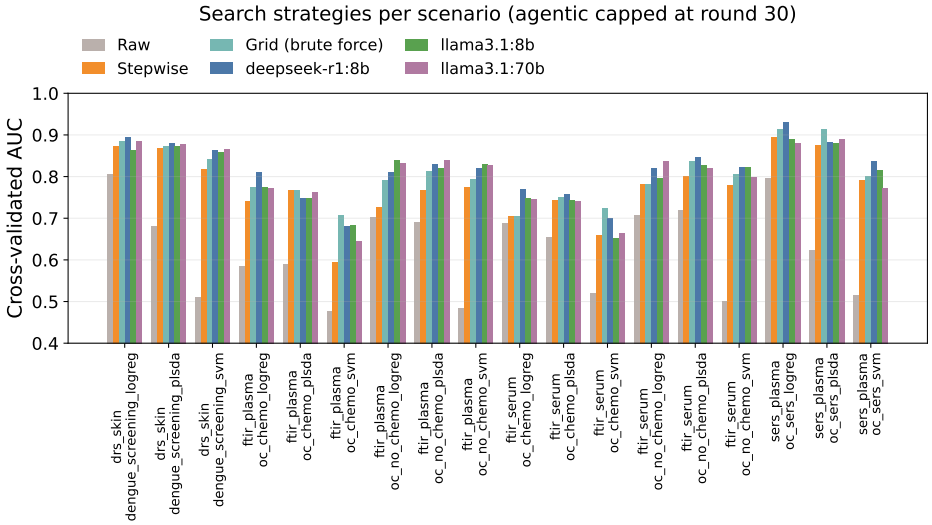


Figure 7: Benchmark results (30 rounds). Top-1 classification performance achieved by each preprocessing optimisation strategy across the 18 scenarios.

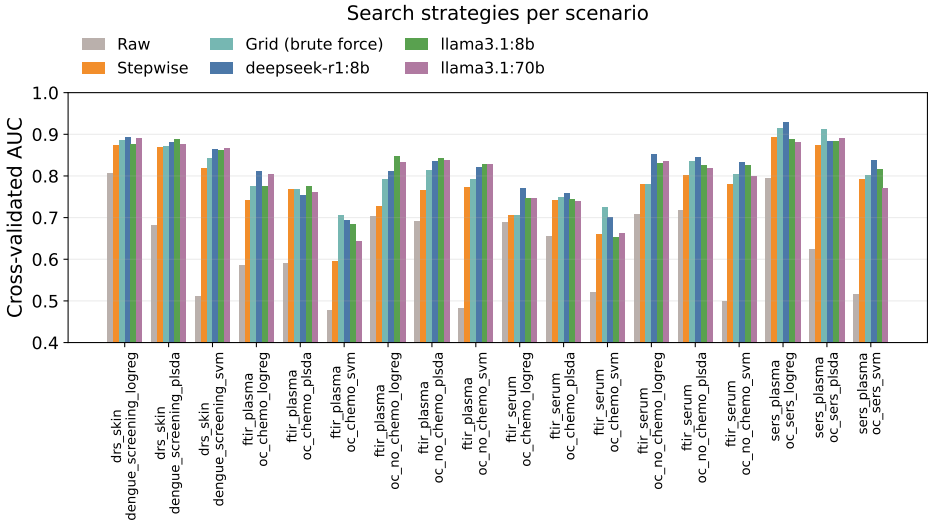


Figure 8: Extended benchmark results (120 rounds). Top-1 classification performance achieved by each preprocessing optimisation strategy across the 18 scenarios.

Table 5: Wall-clock cost of the three agentic campaigns.

			wall clock (h)			per unit (s)	
Campaign	rounds	evals	total	cross-val.	proposal	per round	per eval
deepseek-r1:8b	936	2160	40.8	2.63	**38.2**	157	68
llama3.1:8b	977	2160	11.2	0.65	**10.6**	41	19
llama3.1:70b	601	2160	12.1	0.61	**11.4**	72	20

Table 6: The highest-AUC pipeline each campaign found on `sers_plasma_oc_sers_logreg`, by stage, with the parameters as proposed. All three settle on a polynomial baseline and differ on everything else, including whether to despike or normalise at all.

Stage	`deepseek-r1:8b`	`llama3.1:8b`	`llama3.1:70b`
despike	*identity*	`WhitakerHayes` kernel_size=7,threshold=15	`WhitakerHayes` kernel_size=9,threshold=20
denoise	`Whittaker` lam=500	`IASLS`‡ diff_order=3,lam=5000	`SavGol` deriv=0,polyorder=3,window_length=11
baseline	`Poly` poly_order=8	`Poly` poly_order=6	`Poly` poly_order=8
normalise	`SNV`	*identity*	`Vector`
AUC	0.929	0.889	0.880
found in round	16	23	13

Table 7: Proposal quality and calibration across rounds for the `llama3.1:8b` campaign, comparing the first and last third of each scenario's rounds over LLM-proposed pipelines only. *Scenarios improving* counts increases, except for prediction error where a decrease is the improvement. p-values are sign tests over the 18 scenarios.

Measure	first third	last third	scenarios improving
mean AUC of proposals	0.686	**0.720**	16/18 ($p = 0.001$)
hit rate above early baseline	49.0%	**61.8%**	14/18 ($p = 0.031$)
$\rho(\text{round}, \text{AUC}) > 0$	—	—	14/18 ($p = 0.031$)
mean |prediction error|	0.076	**0.068**	13/18 ($p = 0.096$, n.s.)
distinct family combinations	**52.5%**	40.0%	2/18 ($p = 0.001$)

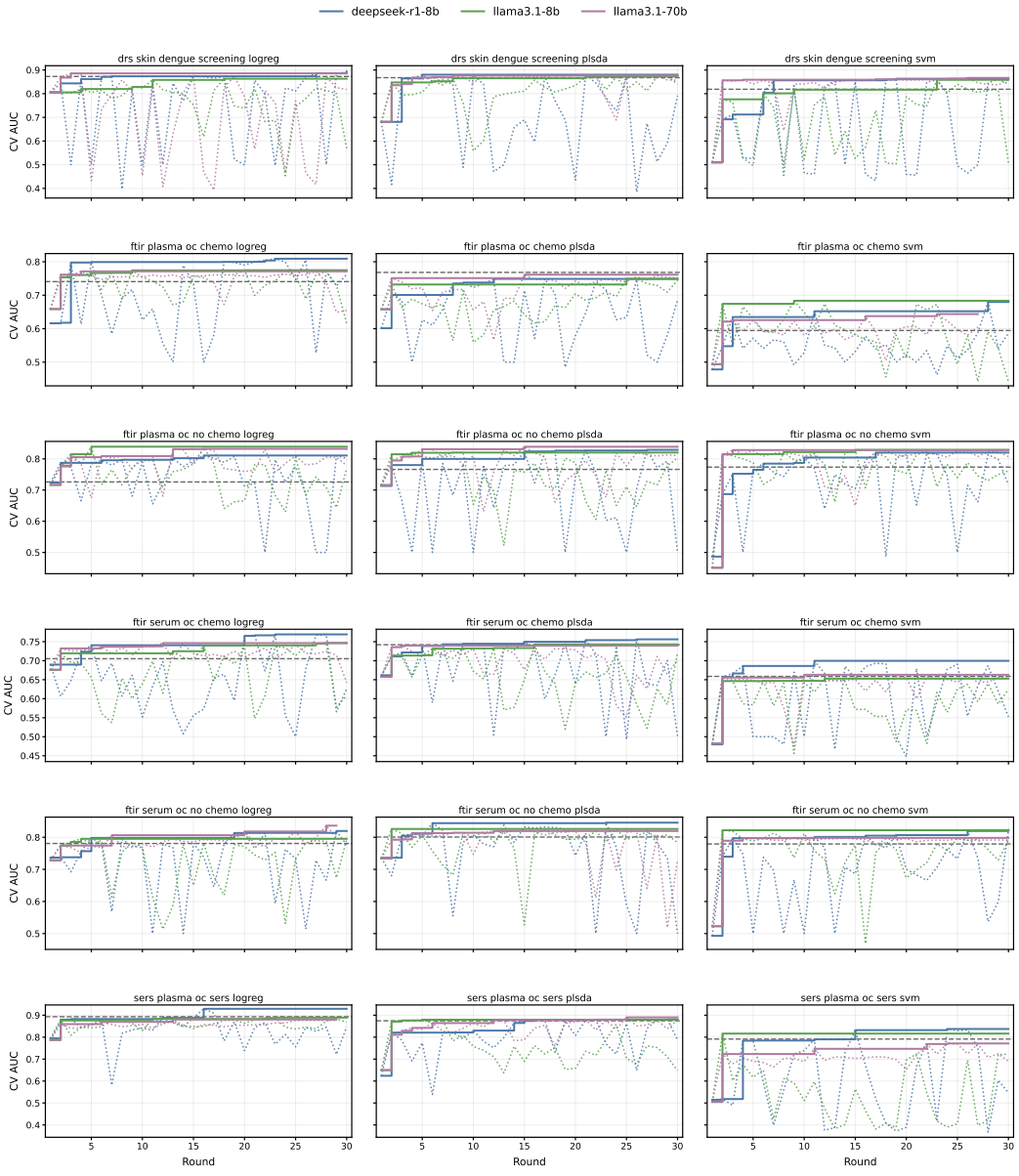


Figure 9: Figure 3 for every scenario. Rows are the (modality, specimen, task) group, columns the classifier, so the three panels of a row share a task and a y-scale. Dotted is the best of each round, the solid step the running best, the grey dashed line greedy stepwise. The pattern Section 5.2 draws from one scenario is general. The step lines flatten early almost everywhere, and the campaign with the highest step is often not the one whose dotted line sits closest to it.

D ABLATION STUDIES

Table 8: Share of proposals citing each dossier section, pooled over all 18 scenarios. A section counts at most once per proposal. Only §2, §4 and §6 are presented as citable by the output contract, so §1, §5, §7 and §8 cannot be cited and are omitted. §3 is shown because it appears in the dossier's rules, though only in a prohibition. The last three columns are observed rather than self-reported: bounds and schema compliance, and the share of rounds in which the reflector's invariant survived the filters of Section 3.2. These are the numbers behind Figure 4(a).

		cited (self-reported)				obeyed (observed)		
Campaign	proposals	§2	§3	§4	§6	§5 bounds	§8 schema	invariant
deepseek-r1:8b	3216	14.9%	0.0%	86.9%	1.5%	97.8%	34.8%	2.0%
llama3.1:8b	7384	11.2%	0.0%	87.0%	1.2%	80.1%	100.0%	53.8%
llama3.1:70b	5758	46.6%	0.7%	81.5%	10.9%	95.8%	100.0%	85.0%

Table 9: Dossier ablation, each arm against the full dossier, paired within (seed, scenario), $n = 4$ pairs. *Worse in* counts the pairs the arm lost, which at $n = 4$ carries information the mean does not. Upper block: leave-one-out over the evidence sections. Lower block: grouped removals and the control ladder §1 ⊂ §1§8 ⊂ §1§5§8 ⊂ full. The two bottom rungs admitted no proposals, so their AUC columns describe the fallback draw rather than the model. † marks the arm whose responses do not parse.

Dossier	mean prop. AUC	Δ vs full	worse in	best AUC	§5 bounds	§8 contract
full dossier	0.645	—	—	0.750	77.6%	100.0%
– §2 top played	0.613	-0.032 ± 0.008	4/4	0.733	81.3%	100.0%
– §3 eliminated	0.641	-0.005 ± 0.009	3/4	0.756	80.2%	100.0%
– §4 dominance	0.639	-0.006 ± 0.005	3/4	0.746	83.6%	100.0%
– §6 predictions	0.639	-0.007 ± 0.009	3/4	0.742	85.3%	100.0%
– §7 diagnostics	0.646	$+0.001 \pm 0.002$	2/4	0.754	80.3%	100.0%
– invariants	0.651	$+0.006 \pm 0.009$	1/4	0.736	84.0%	100.0%
– all history (§2§4§7)	0.595	-0.050 ± 0.008	4/4	0.736	75.0%	100.0%
– self-model (§6 + inv.)	0.668	$+0.023 \pm 0.010$	1/4	0.751	77.5%	100.0%
controls only (§1§5§8)	0.649	$+0.003 \pm 0.005$	2/4	0.758	75.8%	100.0%
task + contract (§1§8)	—	—	—	—	0.0%	100.0%
task only (§1) †	—	—	—	—	0.0%	0.8%

Table 10: Predicted against measured AUC, pooled over every proposal a campaign made. *Signed error* is measured minus predicted, so a negative value is over-prediction. Spearman is the rank correlation between predicted and measured AUC, with *** denoting $p < 10^{-3}$.

Campaign	n	mean pred.	mean actual	signed err.	|err|	Spearman
deepseek-r1:8b	1528	0.772	0.757	-0.015	0.038	+0.798***
llama3.1:8b	1691	0.736	0.702	-0.034	0.071	+0.507***
llama3.1:70b	2052	0.757	0.740	-0.017	0.041	+0.791***

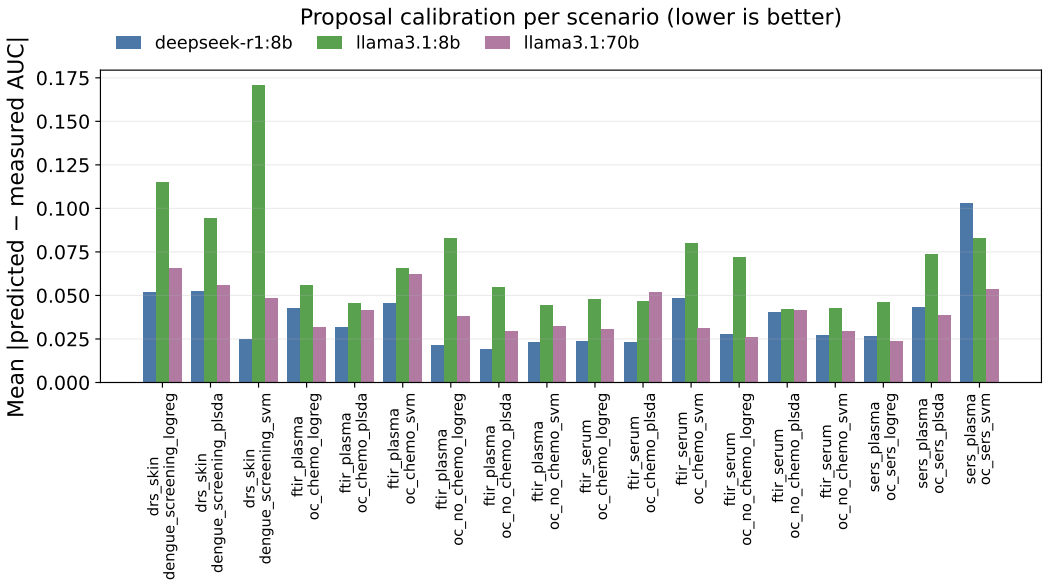


Figure 10: Mean absolute error between the AUC a campaign predicted for a pipeline and the AUC it measured, per scenario. Lower is better. The `llama3.1:70b` campaign improves on `llama3.1:8b` in all 18 scenarios.

Table 11: Signed prediction error by method family, pooled over all 18 scenarios. Entries are measured minus predicted AUC, so a negative value is over-prediction, and $\pm$ is the standard error over the n proposals of that family. A dash marks a family a campaign never proposed. Averaged over campaigns, a polynomial baseline carries a signed error of $+0.002$ while every penalised least-squares alternative is over-predicted, at -0.019 for `PenalisedPoly`, -0.021 for `iModPoly`, -0.029 for `IRSQR`, -0.044 for `IASLS` and -0.050 for `ASLS`.

Method family	deepseek-r1:8b	llama3.1:8b	llama3.1:70b
WhitakerHayes → SavGol → Poly → Vector	$+0.009 \pm 0.003$ (n=62)	-0.019 ± 0.006 (n=156)	$+0.000 \pm 0.002$ (n=291)
WhitakerHayes → SavGol → Poly → SNV	-0.003 ± 0.010 (n=6)	-0.018 ± 0.005 (n=155)	$+0.004 \pm 0.002$ (n=160)
WhitakerHayes → SavGol → IASLS → Vector	$+0.002 \pm 0.012$ (n=7)	-0.092 ± 0.007 (n=228)	-0.061 ± 0.006 (n=68)
WhitakerHayes → SavGol → IASLS → SNV	-0.058 ± 0.030 (n=7)	-0.085 ± 0.009 (n=142)	-0.054 ± 0.006 (n=106)
WhitakerHayes → SavGol → PenalisedPoly → Vector	-0.028 ± 0.014 (n=2)	-0.051 ± 0.015 (n=63)	-0.020 ± 0.005 (n=90)
None → Whittaker → Poly → Vector	-0.004 ± 0.004 (n=81)	$+0.066 \pm 0.060$ (n=2)	-0.010 ± 0.005 (n=66)
WhitakerHayes → Whittaker → Poly → SNV	-0.007 ± 0.003 (n=115)	$+0.015 \pm 0.013$ (n=25)	$+0.000 \pm 0.009$ (n=8)
WhitakerHayes → None → Poly → Vector	$+0.017 \pm 0.007$ (n=12)	$+0.031 \pm 0.016$ (n=4)	$+0.017 \pm 0.004$ (n=113)
WhitakerHayes → Whittaker → iModPoly → Vector	-0.027 ± 0.004 (n=111)	-0.020 ± 0.014 (n=4)	-0.015 (n=1)
WhitakerHayes → SavGol → PenalisedPoly → SNV	—	-0.068 ± 0.015 (n=57)	-0.009 ± 0.005 (n=57)
WhitakerHayes → SavGol → ASLS → SNV	-0.041 ± 0.037 (n=3)	-0.078 ± 0.016 (n=46)	-0.030 ± 0.007 (n=57)
None → Whittaker → IASLS → Vector	-0.032 ± 0.006 (n=51)	$+0.006 \pm 0.033$ (n=5)	-0.022 ± 0.009 (n=47)
None → SavGol → Poly → Vector	-0.017 ± 0.024 (n=19)	$+0.054 \pm 0.054$ (n=3)	-0.015 ± 0.004 (n=75)
WhitakerHayes → None → Poly → None	-0.019 ± 0.005 (n=95)	—	-0.026 ± 0.007 (n=2)
WhitakerHayes → SavGol → IRSQR → Vector	-0.071 ± 0.031 (n=15)	-0.026 ± 0.011 (n=65)	$+0.010 \pm 0.008$ (n=16)
WhitakerHayes → None → PenalisedPoly → Vector	-0.017 ± 0.008 (n=6)	$+0.069 \pm 0.006$ (n=2)	-0.008 ± 0.006 (n=85)

E ADDITIONAL RESULTS

Table 12: Evaluations by origin, pooled over all 18 scenarios. *Identity* is the all-`None` pipeline evaluated once per scenario as a reference, and *parametric sampler* is the fallback draw of Algorithm 1 line 11 from the randomly parameterised seed pool.

Campaign	Origin	n	mean AUC	median	max
`deepseek-r1:8b`	LLM proposal	1528	**0.757**	0.781	0.929
	Parametric sampler	614	0.648	0.658	0.904
`llama3.1:8b`	LLM proposal	1691	**0.702**	0.725	0.889
	Parametric sampler	451	0.665	0.672	0.887
`llama3.1:70b`	LLM proposal	2052	**0.740**	0.760	0.891
	Parametric sampler	90	0.688	0.704	0.878
—	Identity (raw)	54	0.634	0.658	0.806

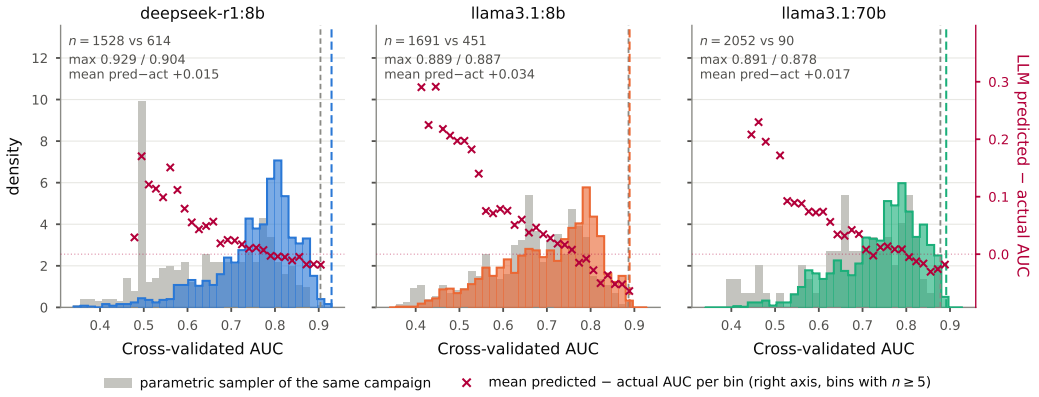


Figure 11: Evaluated pipelines by origin, per campaign (18 scenarios). LLM proposals are shown against the fallback draws of the same campaign (*parametric sampler*), with the mean predicted−actual AUC per bin on the right axis.

Table 13: Admitted pipelines running a method in a stage it is not registered for. The AUC comparison is paired within scenario, because misplacements concentrate in particular scenarios and between-scenario AUC differences dwarf the effect. *Scenarios improved* counts the scenarios in which the the averaged AUC of the misplaced pipelines outscored the conformant ones.

	`deepseek-r1:8b`	`llama3.1:8b`	`llama3.1:70b`
admitted pipelines	1528	1691	2052
with a foreign family	36	278	0
share	2.4%	16.4%	0.0%
scenarios affected	1/18	11/18	0/18
occurrences, by family and the stage it occupied			
`IASLS` in `denoise` (registered: `baseline`)	—	210	—
`Whittaker` in `despike` (registered: `denoise`)	—	68	—
`WhitakerHayes` in `denoise` (registered: `despike`)	36	—	—
`ASLS` in `denoise` (registered: `baseline`)	—	14	—
`Vector` in `baseline` (registered: `normalise`)	—	5	—
`IRSQR` in `denoise` (registered: `baseline`)	—	1	—
effect on AUC			
mean AUC, conformant	0.754	0.692	0.740
mean AUC, foreign family	0.876	0.757	—
paired Δ within scenario	+0.106	+0.029	—
scenarios improved	1/1	9/11	—

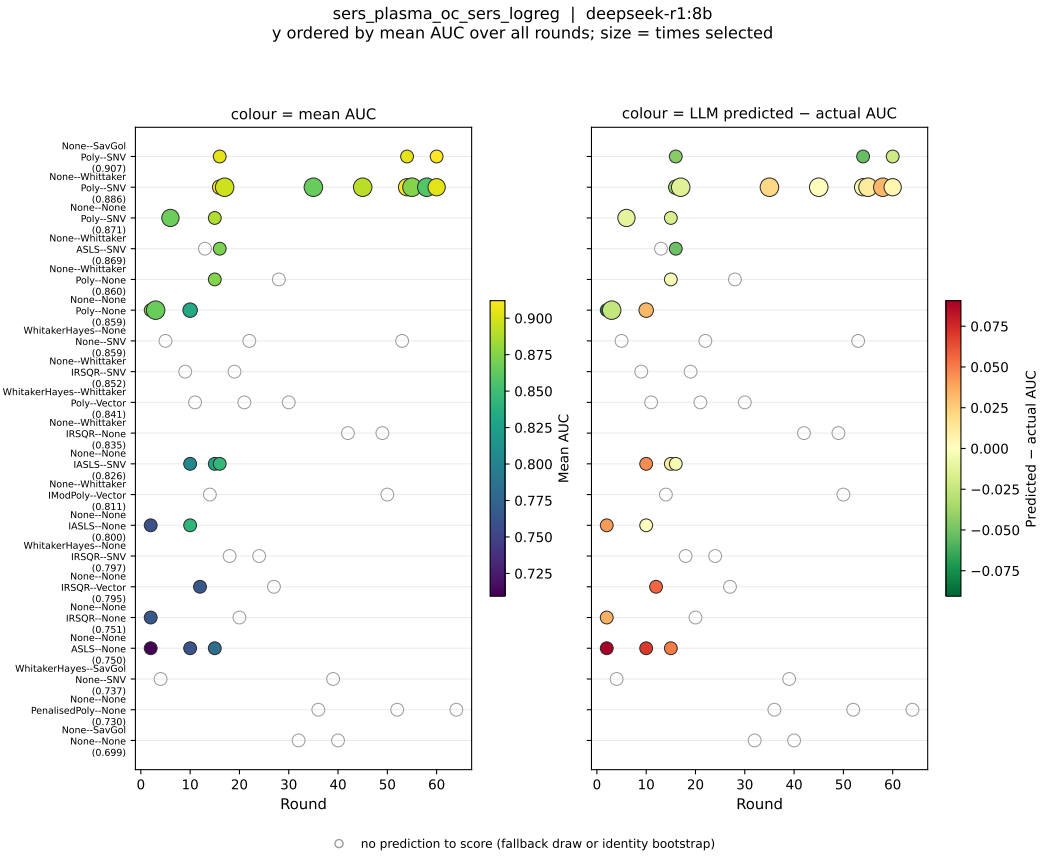


Figure 12: Method families evaluated per round on `sers_plasma_oc_sers_logreg`, `deepseek-r1:8b` campaign (120 evaluations over 64 rounds, 52 distinct families, of which the 20 most-evaluated are shown, covering 73% of evaluations). Round runs against family, bubble area is proportional to the evaluations that family received in that round, and colour and row order are both given by its mean AUC over all rounds. The square at the left margin marks each family's mean on the same colour scale, so the ordering is legible for families evaluated in a single round. In the RHS figure, colour is the predicted LLM AUC - actual AUC.

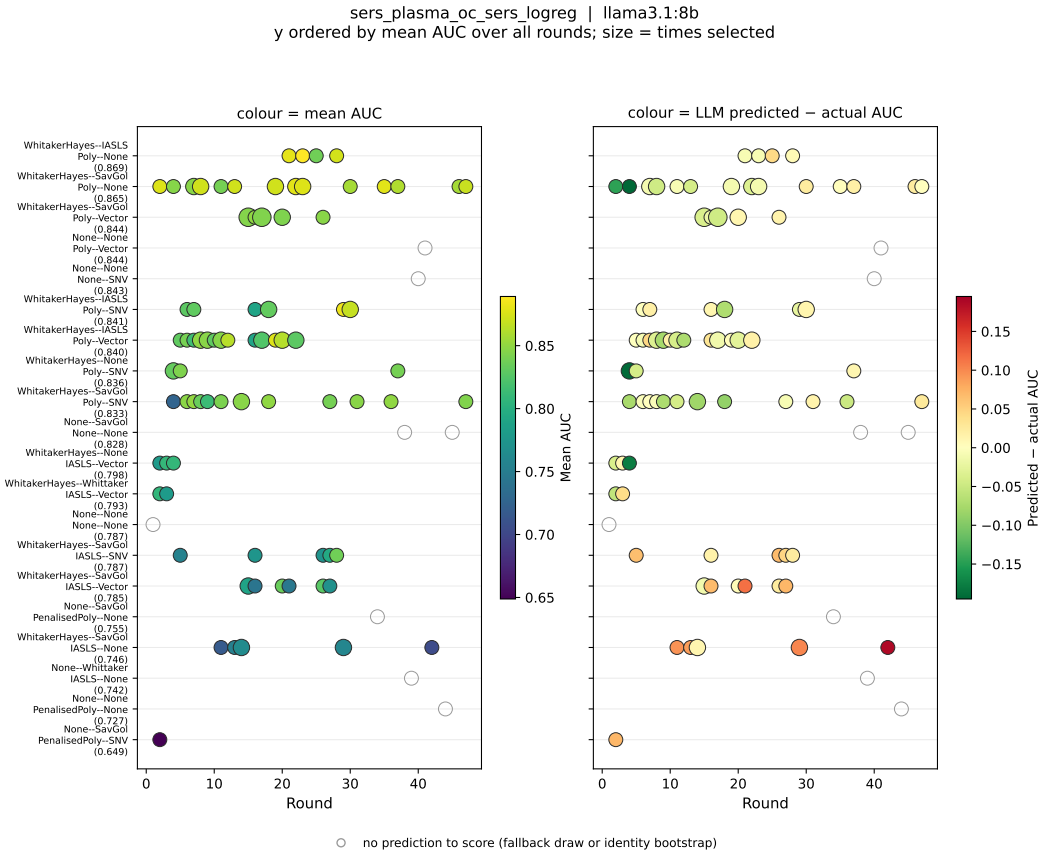


Figure 13: The same scenario under the `llama3.1:8b` campaign (120 evaluations over 49 rounds, 30 families, of which the 20 most-evaluated are shown, covering 92% of evaluations).

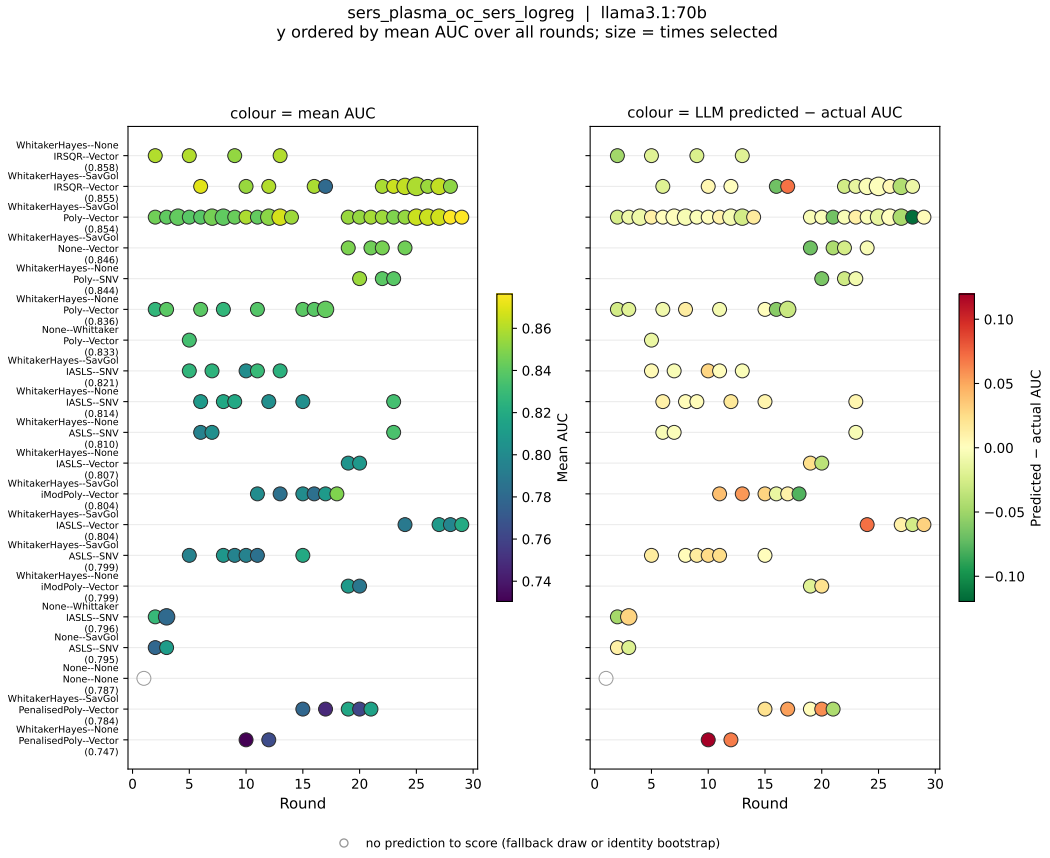


Figure 14: The same scenario under the `llama3.1:70b` campaign (120 evaluations over 29 rounds, 24 families, covering 97% of evaluations).

Table 14: Proposal yield per round, pooled over all 18 scenarios. The denominator is what the search asked for, $K = 10$ proposals per scenario-round, not what came back parseable, so a round whose response could not be extracted contributes zeros rather than being dropped. *Valid* is the share passing the §5 bounds and §8 schema checks. *Admitted* the share surviving those checks and de-duplication against the existing action space. Admission is capped at $C_{\text{grow}} = 5$ new combinations per round, so no cell can exceed 50%. The early rounds of both `llama3.1` campaigns sit exactly at that ceiling and are censored, not merely high. Only scenario-rounds in which a request was issued are counted, so a campaign that exhausts its budget early is not charged for rounds it never ran.

	`deepseek-r1:8b`		`llama3.1:8b`		`llama3.1:70b`	
Round	valid	admitted	valid	admitted	valid	admitted
2	77.8%	38.9%	96.7%	50.0%	92.8%	50.0%
3	54.4%	27.8%	77.2%	40.0%	92.2%	50.0%
4	27.8%	13.9%	74.4%	34.4%	93.9%	48.3%
5	33.3%	16.7%	71.7%	33.9%	99.4%	44.4%
6	27.2%	13.9%	70.0%	26.1%	98.9%	40.6%
7	32.8%	16.7%	76.1%	23.9%	96.7%	35.6%
8	11.1%	5.6%	68.9%	20.6%	98.9%	36.7%
9	21.1%	11.1%	73.3%	19.4%	98.9%	33.3%
10	44.4%	20.0%	68.3%	12.8%	95.6%	31.7%
11	44.4%	22.2%	72.2%	19.4%	95.6%	35.0%
12	27.8%	13.9%	67.2%	16.1%	96.7%	28.9%
13	16.7%	8.3%	70.6%	12.8%	95.0%	35.0%
14	27.2%	13.9%	76.1%	18.9%	95.6%	31.1%
15	33.3%	16.7%	70.6%	18.9%	95.0%	35.0%
16	48.9%	25.0%	73.3%	20.6%	96.7%	31.1%
17	49.4%	25.0%	76.1%	19.4%	95.0%	37.8%
18	33.3%	16.7%	77.2%	21.1%	96.1%	32.2%
19	43.9%	22.2%	81.1%	21.7%	97.2%	37.8%
20	22.2%	9.4%	79.4%	18.9%	96.7%	31.1%
21	50.0%	25.0%	81.1%	16.1%	95.6%	30.6%
22	38.9%	19.4%	80.6%	21.1%	93.9%	30.0%
23	32.8%	16.7%	77.2%	15.0%	95.6%	34.4%
24	21.7%	11.1%	76.1%	20.6%	93.9%	29.4%
25	22.2%	11.1%	71.7%	15.6%	96.1%	40.0%
26	27.8%	13.9%	74.4%	20.0%	96.7%	28.9%
27	22.2%	11.1%	71.7%	16.7%	97.2%	37.8%
28	38.9%	19.4%	76.1%	21.7%	95.9%	28.2%
29	40.6%	21.1%	79.4%	13.3%	96.5%	31.2%
30	27.8%	13.9%	80.6%	20.0%	91.3%	29.3%
mean	34.5%	17.3%	75.5%	21.7%	95.8%	35.4%