Large Discovery Models (LDM v0.1):
Empirically-grounded Model-Based Open-Ended Search

Zhongwei Yu [*16] **Yan Song** [*26] **Xue Yan** [*6] **Anjie Liu** [‡16] **Xingyu Lu** [‡16]
Yihang Chen [‡26] **Huichi Zhou** [26] **Siyuan Guo** [4] **Luoyang Sun** [36]
Sihan Chen [6] **Xiangning Yu** [56] **Jun Wang** [2†]

[1] The Hong Kong University of Science and Technology (Guangzhou)
[2] University College London [3] Institute of Automation, Chinese Academy of Sciences
[4] Jilin University [5] Tianjin University [6] AI Lab, The Yangtze River Delta
*Equal contribution as co-first authors. ‡Equal contribution as co-second authors.
† Corresponding author. jun.wang@ucl.ac.uk.

Abstract

Scientific discovery often requires optimising costly-to-evaluate objectives over vast, structured, and open-ended hypothesis spaces, such as molecules, protein sequences, and computer programs. Large language models (LLMs), or domain-specific generative models, provide flexible generative priors over these spaces. However, their likelihoods and self-assessments do not provide reliable estimates of empirically measured results or calibrated epistemic uncertainty, particularly for novel candidates outside the observed data distribution. We propose the Large Discovery Model (LDM), an empirically grounded recurrent architecture in which an LLM generates and refines candidate designs, while a continually updated Bayesian non-parametric reward model, instantiated here as a Gaussian process (GP), predicts their quality and quantifies uncertainty. A GP-derived acquisition function guides candidate generation, refinement, and selection, while new observations update the surrogate and discovery memory. We formulate this process as a KL-regularised, acquisition-guided search objective and implement it through candidate reweighting, iterative refinement, and decomposed feedback on predicted quality, uncertainty, and constraint satisfaction. We evaluate LDM on neural-network training-program search, antibody CDRH3 sequence design, and multi-objective molecular optimisation. These studies demonstrate how LDM integrates structured generative capabilities of LLMs with uncertainty-aware, experiment-grounded search to support open-ended discovery across diverse scientific design spaces.
This initial release (LDM v0.1) accompanies the public framework, code, and benchmarks available at https://github.com/yzailab/Large-Discovery-Models.

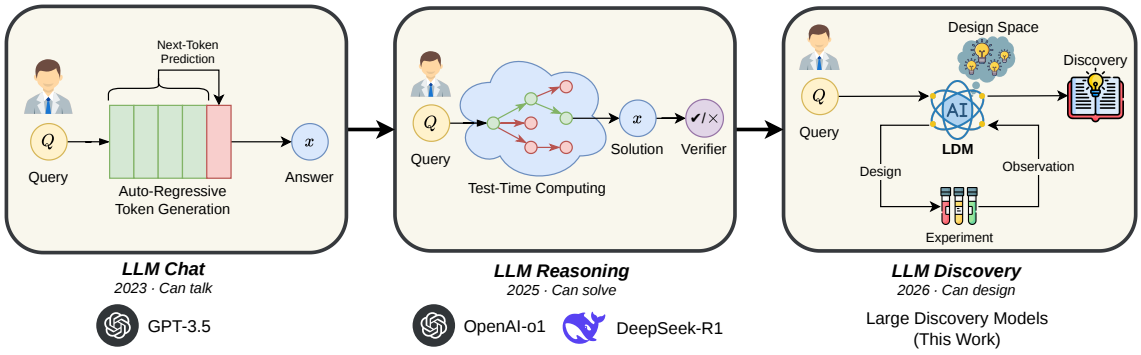


Figure 1: **From generation and reasoning to open-ended discovery.** *LLM generation* produces a response to a user query through autoregressive prediction (Brown et al., 2020). *LLM reasoning* uses additional inference-time computation, such as chain-of-thought, tree search, or best-of-N sampling, to generate, evaluate, and refine candidate solutions, typically with access to a cheap verifier (OpenAI, 2024; Wang et al., 2024; DeepSeek-AI, 2025). *Large Discovery Models* extend this loop to open-ended scientific design spaces, where evaluation requires costly interaction with the external world. Experimental observations update the model and guide the subsequent generation and selection of hypotheses or designs.

1 Introduction

The development of large language models (LLMs) has progressed from open-ended generation towards increasingly capable forms of reasoning. Pretraining provides broad linguistic and conceptual knowledge, while inference-time scaling allows a model to generate, compare, and refine multiple candidate solutions. This approach has produced substantial gains in mathematics, coding, theorem proving, and game playing, where solutions may be difficult to find but comparatively easy to evaluate. Formal derivations can be assessed by process or outcome verifiers, programs can be executed against unit tests, and game strategies can be evaluated using exact rules or simulators (Yao et al., 2023a; Madaan et al., 2023; Snell et al., 2025). AlphaZero-like methods make this feedback loop explicit by combining generation, evaluation, and search (Silver et al., 2017; Feng et al., 2024). OpenR (Wang et al., 2024) provides an open framework for studying such reasoning systems, while AlphaEvolve (Novikov et al., 2025) couples language-model proposals with automated evaluators at scale. Together, these developments demonstrate that inference-time computation is particularly effective when reliable and repeatable feedback can turn generation into directed search.

Scientific discovery poses a more challenging setting. The candidate space may be vast, structured, and only partially specified, while feedback often depends on expensive simulation, accelerator experiments, physical assays, or real-world experiment. Evaluations cannot be repeated freely, and the resulting feedback may be delayed or noisy. A discovery system must therefore decide not only which candidates to evaluate, but also which hypotheses and designs should enter the search process in the first place. Figure 1 illustrates this progression from generation and reasoning to discovery.

Definition 1 (Scientific discovery)**.** Scientific discovery, in the setting considered here, is a sequential process in which an agent generates hypotheses or designs, selects a limited number for evaluation through interaction with an external evaluator, and uses the resulting empirical observations to update its beliefs and guide subsequent search. A discovery occurs when this process identifies a previously unknown and non-trivial relationship, mechanism, or design that is supported by empirical evidence.[1]

Definition 1 highlights two differences between scientific discovery and reasoning with cheap verifiers. First, scientific search is often *open-ended*: relevant candidates may lie outside the region represented by the current model or reachable through its existing search operators. Second, empirical feedback is *budgeted*: only a small fraction of the generated candidates can be evaluated. The agent must therefore learn from limited observations, model an unknown objective, and allocate its experimental budget to candidates that are expected to be useful or informative (Shahriari et al., 2016; Frazier, 2018).

Current LLM-based research systems address only part of this problem. Autonomous research agents can generate hypotheses, write code, execute experiments, analyse results, and produce scientific manuscripts (Lu et al., 2024). LLM-generated research proposals have also been judged more novel than proposals written by human experts (Si et al., 2024). However, these proposals exhibit limited diversity and weaker feasibility, and LLMs do not reliably evaluate their own ideas. End-to-end assessments similarly identify persistent limitations in experimental

[1]This is an operational definition of empirically grounded, search-based scientific discovery: the class of scientific problems that can be formulated as sequential search over hypotheses or designs, with selected candidates evaluated through physical experiments, simulations, computational tests, or other external sources of evidence. It does not encompass all forms of scientific discovery, particularly those for which the relevant search space, evaluation procedure, or objective cannot yet be specified. Moreover, *unknown* and *non-trivial* are relational notions, defined relative to the agent's prior knowledge and, where appropriate, the knowledge of the relevant scientific community. A result may therefore be a discovery for an agent while constituting a rediscovery from the community's perspective.

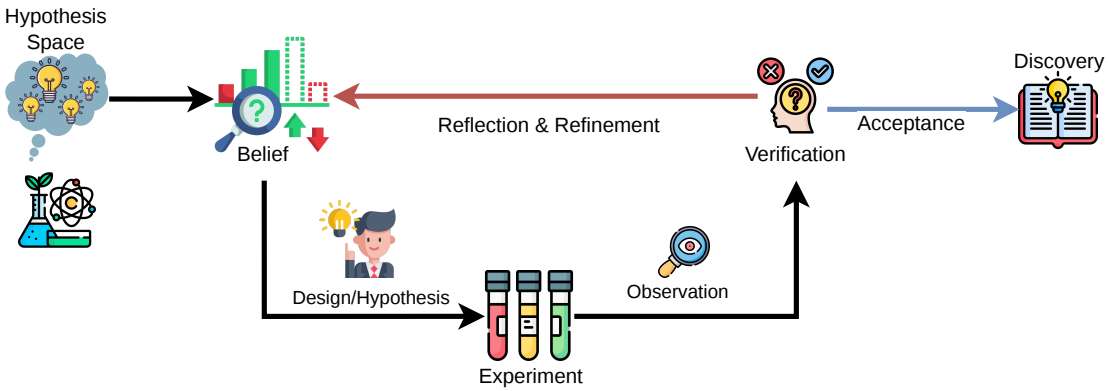


Figure 2: **The Experimental Scientific Discovery (ESD) loop.** An agent generates candidate hypotheses or designs, selects a subset for costly evaluation, observes the outcomes, and uses those observations to update its beliefs and guide the next round of search, as described in Definition 1.

execution, novelty assessment, and methodological reasoning (Beel et al., 2025; Xu et al., 2026; Starace et al., 2025). Thus, fluent generation and procedural automation do not by themselves provide the empirically grounded value signal needed for scientific discovery.

This limitation also appears in scientific design. An LLM assigns high probability to candidates that are plausible under its learned distribution, but linguistic or sequence likelihood need not correlate with an externally measured scientific property (Gupta et al., 2025; Chang et al., 2025). LLM-only search can therefore produce valid candidates without reliably determining which ones merit expensive evaluation. Bayesian optimisation (BO) offers a complementary capability. Given a limited set of observations, BO learns a probabilistic surrogate of the unknown objective and uses an acquisition function to balance predicted performance against epistemic uncertainty (Jones et al., 1998; Srinivas et al., 2010; Shahriari et al., 2016; Frazier, 2018). It can therefore prioritise promising or informative experiments under a limited evaluation budget. However, BO still requires a representation of the design space and a mechanism for proposing or reaching candidates. In structured and combinatorial spaces, such as molecules, proteins, programs, and experimental protocols, optimising the acquisition function may itself be difficult (Balandat et al., 2020; Griffiths and Hernández-Lobato, 2020). BO can efficiently rank the candidates exposed by its current representation and search operators, but it may fail to reveal valuable candidates outside that search support.

These observations expose complementary limitations. LLMs can generate and modify structured candidates, but lack a reliable estimate of their external scientific value. BO grounds candidate selection in empirical observations, but its effectiveness depends on which candidates its representation and search procedure can expose. Scientific discovery requires both capabilities: the search frontier must expand, and experimental resources must be allocated using an uncertainty-aware value signal that balances expected value with information gain.

We formalise this problem as *sequential inverse design*. A forward model predicts the properties of a given design, whereas inverse design asks which candidate should be evaluated next to achieve a desired property or maximise an unknown reward. Evaluations may involve physical measurements, simulations, computational tests, or other external sources of evidence. Because these evaluations can be costly, only a limited number of candidates can be examined. At the same time, the structured design space may be combinatorial, effectively unbounded, or impossible to enumerate. Accurately modelling the reward across the entire design space is therefore infeasible.

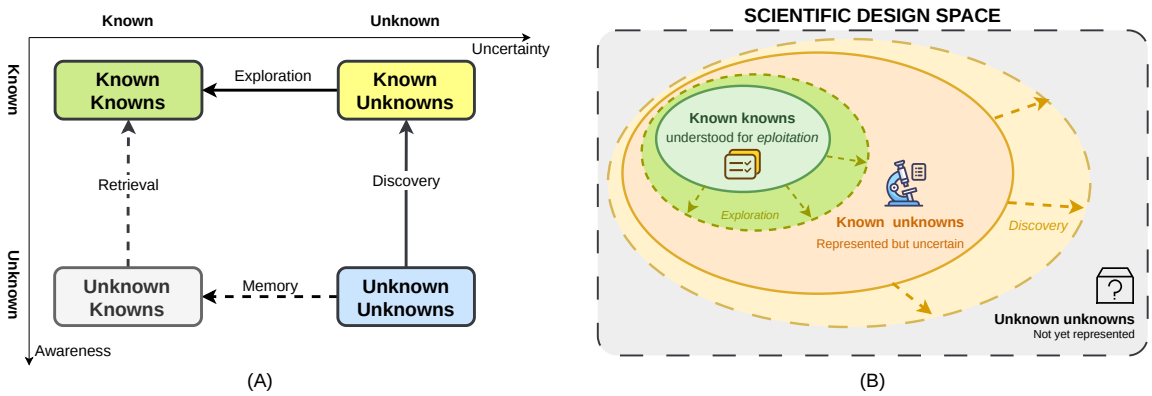


Figure 3: **Epistemic regimes of experimental scientific search.** **(A)** An adaptation of the Rumsfeld Matrix (Krogerus and Tschäppeler, 2018), organised by search awareness and objective uncertainty. **(B)** A scientific agent moves among three regimes in the design space.

Instead, the agent maintains a *probabilistic surrogate*: a data-efficient model of the unknown reward learned from the candidates evaluated so far. For each candidate within its modelling support, the surrogate provides both a predicted reward and an estimate of *epistemic uncertainty*. Epistemic uncertainty reflects limitations in the surrogate's knowledge arising from insufficient or uninformative observations. It can therefore be reduced by collecting relevant new evidence. This differs from irreducible randomness or measurement noise in the evaluation itself. After each evaluation, the new observation is incorporated into the surrogate, updating both its predictions and its uncertainty.

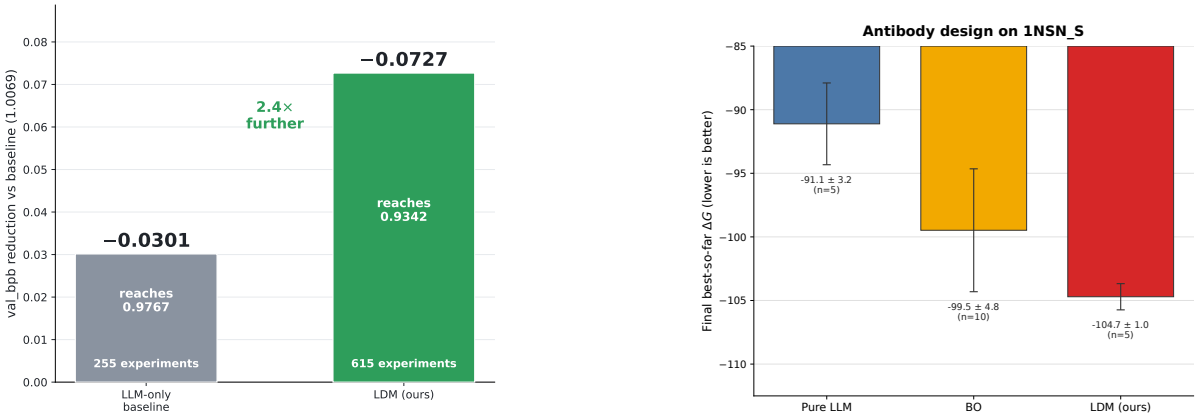
An *acquisition function* converts the surrogate's predictions and uncertainties into a measure of how useful it would be to evaluate or further develop each candidate. It balances candidates predicted to perform well against candidates that remain uncertain but potentially valuable. An acquisition value is therefore not simply a predicted reward; it represents the candidate's value to the sequential search process. The agent uses this signal to allocate limited inference-time computation and to select candidates for costly empirical evaluation. It consequently alternates among candidate generation, evaluation selection, and belief updating.

This formulation separates two epistemic dimensions. The first is *predictive uncertainty*: how uncertain the surrogate is about the reward of a candidate represented within its current modelling support. The second is *search awareness*: whether a candidate is represented by, or reachable through, the current proposal and search process. Together, these dimensions distinguish three modes of scientific search:

- *Exploitation* selects reachable candidates that the surrogate predicts will perform well.
- *Exploration* evaluates reachable candidates about which the surrogate has high epistemic uncertainty, thereby improving knowledge within the current search support.
- *Discovery* expands or reshapes the search support to reveal candidates, design families, or relationships that were not previously represented or reachable.

Figure 3 illustrates this perspective by adapting the four epistemic regimes to scientific design. It distinguishes uncertainty about candidates within the current search support from a lack of awareness of candidates outside that support. Exploration reduces uncertainty about what is already represented, whereas discovery changes what can be represented and reached.

Motivated by this distinction, we introduce the *Large Discovery Model* (LDM), an empirically grounded recurrent architecture that couples an LLM proposal model with a continually updated



(a) Performance on AutoResearch.

(b) Performance on antibody design.

Figure 4: **Complementary limitations of LLM-only and BO-based search. (a)** LDM improves upon the LLM-only AutoResearch baseline under the same H100 hardware setting. **(b)** Final best-so-far binding energy on the antibody-design task, where LDM outperforms the LLM-only and BO-only baselines. Lower binding energy is better. For details, we refer to § 6.

probabilistic surrogate. The LLM generates, edits, and refines structured candidates, thereby constructing a dynamic search frontier. The surrogate estimates the candidates' rewards and quantifies its epistemic uncertainty about them. The acquisition function uses these estimates to determine which candidates merit further refinement, additional inference-time computation, or costly empirical evaluation. Each new observation updates the surrogate and, through the acquisition signal, redirects subsequent generation and search.

The acquisition function is therefore more than a post-generation filter. It provides an empirically grounded value signal for allocating both computational and evaluation resources. The LLM expands and reshapes the set of reachable candidates, while the surrogate and acquisition function guide search over this evolving set. In this way, LDM extends inference-time scaling from tasks with cheap and repeatable verifiers to scientific-design problems characterised by costly, noisy, and limited empirical feedback.

We evaluate LDM in three structured scientific-design domains: neural-network training-program search on AutoResearch (Karpathy, 2026), antibody CDRH3 sequence design, and multi-objective molecular optimisation. LDM exceeds the best previously reported AutoResearch result. With a weak language-model prior, it remains competitive with AntBO (Khan et al., 2022), a specialised BO method for antibody design. It also achieves substantial improvements over both LLM-only and BO-only baselines in multi-objective molecular design. Figure 4 presents selected results, with the full experimental evaluation deferred to § 6.

Our contributions are fourfold. First, we formulate scientific discovery as inference-time search over a dynamic, structured design space, guided by an uncertainty-aware value model grounded in empirical observations. Second, we introduce an acquisition-tilted search policy that combines the structured prior of an LLM with inference-time computation and sequential experimental feedback. Third, we provide a regret analysis that separates surrogate-model error from the suboptimality of the LLM-based search process. Finally, we demonstrate the framework across three classes of scientific objects: computer programs, biological sequences, and molecules.

The remainder of the paper is organised as follows. § 2 introduces the idea of discovery and its epistemic regimes. § 3 derives the LDM formulation and its acquisition-tilted search method. § 4 presents the practical algorithm. § 5 provides the regret analysis. Finally, § 6 reports the empirical results.

2 Search and Discovery in Open-Ended Design Spaces

In this work, we aim to discover a design or hypothesis $x \in \mathcal{X}$ that achieves the highest reward $R(x)$, i.e. to find or closely approximate $x^\star \in \arg\max_{x \in \mathcal{X}} R(x)$. This setting presents two intrinsic difficulties. First, R is a black-box function that is expensive to evaluate and potentially noisy. Second, the hypothesis space $\mathcal{X}$ is typically open-ended: vast, combinatorial, structured, or not even fully specified in advance, as in the design of molecules, biological sequences, computer programs, and experimental workflows.

This differs from classical numerical optimisation or combinatorial search, which generally assumes a fixed and explicitly specified domain together with a gradient, numerical oracle, or known set of valid search operations. In scientific discovery, the full hypothesis space may not be enumerable, and the search may need to formulate hypotheses that lie beyond its current representation or reach.

To reason about what is currently known, we introduce *a probabilistic surrogate*, i.e. a posterior $p(R \mid \mathcal{D}_t)$ over the reward surface R conditioned on the dataset $\mathcal{D}_t = \{(x_i, r_i)\}_{i=1}^{t-1}$ of previously evaluated designs and their noisy feedback $r_i \approx R(x_i)$. Its predictive belief is usually summarised by posterior moments $(\mu_t(x), \sigma_t(x))$, representing the expected reward and predictive uncertainty on the surrogate's modelling support. The surrogate therefore describes the search's current empirical knowledge, but it does not by itself determine or enumerate the full hypothesis space.

Using this surrogate-relative notion of knowledge, we partition $\mathcal{X}$ into four epistemic regimes, as illustrated in Figure 3.

- *Known knowns* are evaluated designs whose rewards are sufficiently well understood. The surrogate predicts their values with low posterior uncertainty $\sigma_t(x)$.
- *Known unknowns* are designs that the search can formulate and the surrogate can model, but whose rewards remain unresolved. They are represented by high posterior uncertainty $\sigma_t(x)$ and can be resolved through further evaluation.
- *Unknown knowns* are designs or evaluation results that exist in an external source, such as a database, a previous experiment, or a parallel discovery process, but are absent from the current search context. They can become available only through an explicit retrieval operation.
- *Unknown unknowns* are designs that lie beyond the search's current effective reach. This can occur for two reasons: (i) $\mathcal{X}$ is too large, combinatorial, or effectively unbounded for the current search procedure to reach the design; or (ii) the design lies outside the surrogate's modelling support, so that $(\mu_t(x), \sigma_t(x))$ does not constitute a meaningful calibrated prediction.

These regimes distinguish uncertainty, reachability, modelling, and knowledge. A design does not become known merely because it belongs to a mathematically defined space; a reachable design is not necessarily supported by the surrogate; and information stored externally is not available unless it is retrieved into the current context. All four regimes may therefore persist throughout the search.

The *search frontier* is the conceptual boundary of what the current procedure can effectively formulate, reach, and model. Designs inside this frontier are accessible to the search, although their rewards may remain uncertain. Designs outside it have not yet entered the modelled search process. This distinction gives rise to three operations closely related to Wang et al. (2008):

- *Exploitation* acts on known knowns by selecting designs for which the surrogate predicts both high reward $\mu_t(x)$ and low uncertainty $\sigma_t(x)$.

- *Exploration* resolves known unknowns by evaluating reachable but uncertain designs. The resulting observations reduce posterior uncertainty and may turn known unknowns into known knowns.
- *Discovery* expands the search frontier by formulating hypotheses that were previously outside the search's effective reach. This brings unknown unknowns into the modelled hypothesis space, where they become known unknowns that can subsequently be evaluated.

Discovery is therefore not simply the evaluation of an untested design. An untested design that already lies within the surrogate's modelling support is a known unknown and belongs to exploration. Discovery instead changes what the search can express, reach, or meaningfully model. Experimental evaluation then determines the value of the newly surfaced hypothesis.

The three operations correspond to three transitions in Figure 3: discovery moves hypotheses from unknown unknowns to known unknowns, exploration moves them from known unknowns to known knowns, and exploitation operates within the known-known regime. The unknown-known regime requires separate memory read and write operations. We leave memory retrieval and federated discovery to future work and focus here on discovery, exploration, and exploitation within a single sequential search process.

This formulation is conceptual and does not prescribe a particular discovery algorithm. The next section introduces the Large Discovery Model, which instantiates these operations by combining a generative foundation model, the probabilistic surrogate introduced above, and an acquisition-guided inference-time search policy.

3 The Large Discovery Model

We are now ready to define the *Large Discovery Model* (LDM). LDM is a policy for sequential inverse design built from three interacting components: a generative foundation model, a probabilistic surrogate, and a model-based acquisition function.

Same as before at iteration t, let $\mathcal{D}_t = \{(x_i, r_i)\}_{i=1}^{t-1}$ denotes the empirical evaluation history, where x_i is a previously evaluated design and r_i is its observed reward. We use $\mathcal{C}_t$ to denote the broader *search context* available to the generative model. This context may include $\mathcal{D}_t$, previously generated candidates, surrogate predictions, acquisition values, constraint feedback, refinement trajectories, and other forms of search memory. Thus, $\mathcal{D}_t$ contains the empirical evidence used to fit the surrogate, whereas $\mathcal{C}_t$ contains the information used to condition candidate generation.

(i) A generative foundation model. We denote by

$$p_{\theta,\alpha}(x \mid \mathcal{C}_t) \tag{1}$$

the proposal distribution over plausible designs conditioned on the current search context. Here, θ denotes the parameters of the underlying generative model, while α denotes its inference configuration, such as the prompting strategy, sampling temperature, reasoning procedure, refinement scheme, or inference-time compute budget. We use $p_{\theta,\alpha}$ throughout the paper to make explicit that the effective proposal distribution depends not only on the model parameters but also on how the model is used at inference time.

The generative model supplies a structured prior over the design space $\mathcal{X}$. It enables the search to generate candidates that reflect domain knowledge, semantic plausibility, and structural constraints without requiring $\mathcal{X}$ to be explicitly enumerated. By changing its context or inference procedure, the model can also expand and reshape the set of candidates reachable by the current search.

(ii) A probabilistic surrogate. As introduced in Section 2, the posterior $p(R \mid \mathcal{D}_t)$ represents the current belief about the unknown reward function, conditioned on the empirical observations collected so far. For a candidate x, the predictive mean $\mu_t(x)$ estimates its expected reward, while the predictive uncertainty $\sigma_t(x)$ quantifies the surrogate's epistemic uncertainty about that estimate. High epistemic uncertainty indicates that the prediction is weakly supported by the available evidence and may be reduced through additional informative evaluations. It is therefore distinct from irreducible randomness or measurement noise in the evaluation process.

The surrogate provides the empirical grounding that the generative model alone generally lacks. Although a foundation model may encode substantial domain knowledge, its likelihoods and self-assessments do not necessarily provide calibrated estimates of a task-specific quantitative reward, particularly for novel candidates outside the observed data distribution.

(iii) A model-based acquisition function. From the surrogate's predictive distribution, we construct an acquisition function $a_t(x)$ that assigns each candidate a scalar value to the sequential search process. Depending on its form, the acquisition function may favour candidates with high predicted reward, candidates with high epistemic uncertainty, or a principled combination of the two. It therefore mediates the trade-off between exploitation and exploration.

The acquisition value should not be interpreted simply as the predicted reward of a candidate. Rather, it measures the potential value of allocating further computation or empirical evaluation to that candidate. It can be used both to decide which candidates warrant costly external evaluation and to direct inference-time operations such as sampling, ranking, refinement, and branch expansion.

These three components play complementary roles. The generative model proposes structured candidates and can move the search frontier towards designs not previously considered. The surrogate grounds the search in empirical observations and quantifies epistemic uncertainty. The acquisition function uses this predictive belief to determine which reachable candidates are most valuable to develop or evaluate. LDM thereby uses empirically calibrated feedback to direct the inference-time search of the generative model.

We seek a search distribution that attains high expected acquisition value while remaining sufficiently close to the structured prior supplied by the generative model. Assume that $\mathcal{X}$ is a measurable space, with its σ-algebra omitted for simplicity. At iteration t, we define

$$\pi_t = \operatorname*{arg\,max}_{q \in \Delta(\mathcal{X})} \left\{ \mathbb{E}_{x \sim q}[a_t(x)] - \frac{1}{\eta} \mathrm{KL}(q \,\|\, p_{\theta,\alpha}(\cdot \mid \mathcal{C}_t)) \right\}, \tag{2}$$

where $\Delta(\mathcal{X})$ denotes the set of probability measures over $\mathcal{X}$, KL is the Kullback–Leibler divergence, and $\eta > 0$ controls the strength of acquisition-based tilting. The expected-acquisition term moves probability mass towards candidates that are valuable under the empirically grounded surrogate. The KL term prevents the search policy from departing arbitrarily far from the generative model's prior over plausible and well-formed designs. The parameter η controls this trade-off: larger values place greater emphasis on acquisition value, whereas smaller values keep the search closer to the generative prior.

The inference configuration α affects π_t through $p_{\theta,\alpha}(\cdot \mid \mathcal{C}_t)$ and therefore changes the reference distribution against which the KL divergence is measured. Increasing inference-time computation may reshape this proposal distribution through additional sampling, reasoning, refinement, or structured search. The acquisition function then determines how that computation is allocated according to predicted empirical value and epistemic uncertainty.

Equation (2) is deliberately model-agnostic: it does not depend on a particular generative architecture, probabilistic surrogate, or acquisition function. Its variational form follows the

Gibbs variational principle and is closely related to Gibbs posteriors (Donsker and Varadhan, 1975; Bissiri et al., 2016), KL-regularised policy search (Peters et al., 2010), maximum-entropy reinforcement learning (Haarnoja et al., 2018), and control as probabilistic inference (Levine, 2018). Accordingly, our contribution is not the generic KL-regularised objective in isolation. The distinctive role of Equation (2) in LDM is to connect a structured generative prior to an acquisition value derived from a continually updated empirical surrogate. This connection allows external observations to direct both inference-time computation and sequential evaluation over structured, potentially open-ended design spaces.

The resulting policy is recurrent (see Fig 5). Candidates generated under the current search context are scored and refined using the surrogate-derived acquisition signal; selected candidates are then evaluated by an external empirical source; and the resulting observations update $\mathcal{D}_t$, the surrogate, and the next search context $\mathcal{C}_{t+1}$. LDM therefore does not merely reweight a fixed set of generated samples. It repeatedly couples generation, empirical evaluation, belief updating, and search-frontier expansion.

In the following sections, we first show that Equation (2) admits a closed-form acquisition-tilted policy. We then describe how this ideal policy can be approximated through inference-time sampling, selection, and iterative refinement, with particular attention to test-time compute scaling. We subsequently instantiate the probabilistic surrogate as a Gaussian process and construct a_t from its posterior predictive distribution. Finally, we describe how the same acquisition signal can support parameter adaptation: in addition to directing inference through α, it may provide supervision for updating the generative parameters θ through fine-tuning.

3.1 An Acquisition-Tilted Search Distribution

With the LLM prior $p_{\theta,\alpha}$ and the acquisition function a_t defined, our desired search policy π_t follows directly from the variational objective in Eq. (2). This variational problem is a canonical instance of the Gibbs variational principle (equivalently, the Donsker–Varadhan representation of the KL divergence) (Donsker and Varadhan, 1975; Bissiri et al., 2016), whose unique maximiser is the exponentially tilted distribution. This form of KL-regularised reward maximisation is the same mathematical foundation underlying maximum-entropy reinforcement learning, control-as-inference (Levine, 2018), and the closed-form policy objective in KL-regularised RLHF (Rafailov et al., 2023).

Proposition 1 (Optimal search distribution)**.** *The unique maximiser of* (2) *is*

$$\pi_t(x) = \frac{1}{Z_t}\, p_{\theta,\alpha}(x \mid \mathcal{C}_t)\, \exp\{\eta\, a_t(x)\}, \qquad Z_t = \int_{x\in\mathcal{X}} p_{\theta,\alpha}(x \mid \mathcal{C}_t)\, \exp\{\eta\, a_t(x)\}\, \mathrm{d}x. \tag{3}$$

Proof. Please refer to Appendix A.1. □

Eq. (3) shows that the optimal policy π_t takes an acquisition-tilted form, where the base measure is reweighted by the exponent of the acquisition function. Interestingly, this policy draws an analogy to the infinite-armed bandit framework (Berry et al., 1997; Wang et al., 2008): as scientific designs (arms) cannot be listed in advance, they are revealed from a reservoir distribution, and the policy tilts that reservoir toward arms it judges valuable. In this reading, $p_{\theta,\alpha}(\cdot \mid \mathcal{C}_t)$ plays the role of the reservoir, $\eta\, a_t(x)$ acts as the strength-of-reward tilt, and Z_t is the usual log-sum-exp normaliser.

A clear interpretation of the inference configuration α and the KL-regularisation coefficient η can be drawn through standard analysis of infinite-armed bandits. We restrict attention to the common case in which the reservoir admits the temperature form

$$p_{\theta,\alpha}(x \mid \mathcal{C}_t) \propto p_\theta(x \mid \mathcal{C}_t)^\alpha, \tag{4}$$

where p_θ denotes the unconditional base proposal. This is the exponential family of reservoirs commonly used in the bandit literature, and it makes α transparent as the strength of the prior: large values of α concentrate probability mass on modes of p_θ, while $\alpha \to 0$ spreads mass toward a uniform measure on $\mathcal{X}$. Substituting this form into (3) and taking the mode gives the bandit-style argmax

$$x_{t+1} = \arg\max_{x \in \mathcal{X}} \Big[\alpha \log p_\theta(x \mid \mathcal{C}_t) + \eta\, a_t(x) \Big], \tag{5}$$

which picks the arm that maximises a weighted combination of the prior log-mass $\log p_\theta$ and the reward tilt a_t, i.e., the bandit analogue of selecting the most promising revealed arm.

Examining the two limiting cases of the temperature form $p_{\theta,\alpha}(x \mid \mathcal{C}_t) \propto p_\theta(x \mid \mathcal{C}_t)^\alpha$ situates LDM relative to related paradigms. As $\eta \to 0$ the tilt vanishes and π_t collapses to the raw reservoir $p_{\theta,\alpha}(\cdot \mid \mathcal{C}_t)$, so search is driven entirely by the generative model with no data feedback. As $\alpha \to 0$ the reservoir spreads toward a uniform measure on $\mathcal{X}$ and the policy becomes $\pi_t \propto \exp\{\eta\, a_t(\cdot)\}$, recovering classical Bayesian optimisation once η is also taken large. LDM is the regime between these two limit points, where both the structured prior and the uncertainty-aware acquisition contribute to the policy. We note, however, that some LLM inference backends do not admit the temperature form $p_\theta(x \mid \mathcal{C}_t)^\alpha$, and the preceding analysis should be read as the canonical case under which the role of α is most cleanly interpreted.

With this construction, LDM defines a discovery policy π_t from which new designs may be drawn using either a stochastic or a deterministic approach, as illustrated by Eq. (5). In practice, the normalising constant Z_t is never computed explicitly, and we instead rely on inference-time search to approximate draws from the tilted distribution.

3.2 Surrogate Modelling of the Reward Function

To efficiently navigate the design space, we require a probabilistic model over function space that treats the reward function R as a random function and models the posterior belief $R \mid \mathcal{D}_t$ conditioned on our cumulative evaluations. Our LDM formulation is largely inspired by the uncertainty-aware decision-making paradigm of Bayesian optimisation (Garnett, 2023).

We model this objective using a Gaussian Process (GP) (Rasmussen and Williams, 2006). GPs serve as highly effective surrogates in this setting because they are exceptionally sample-efficient, admit exact Bayesian updates, and provide principled non-parametric uncertainty estimates across unexplored regions of the design space. Formally, we define the prior distribution over the reward function as:

$$R(x) \sim \mathcal{GP}\big(m(x),\, k(x, x')\big), \tag{6}$$

where $m : \mathcal{X} \to \mathbb{R}$ is the prior mean function and $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a positive-definite kernel function capturing spatial correlation.

By conditioning this prior on the historical data $\mathcal{D}_t = \{(x_i, r_i)\}_{i=1}^t$, we obtain a Gaussian posterior distribution characterised by a predictive mean $\mu_t(x)$ and a predictive variance $\sigma_t^2(x)$ for any candidate point $x \in \mathcal{X}$. These two statistical moments allow us to construct acquisition functions that elegantly balance exploitation (seeking high predicted rewards) and exploration (targeting regions of high uncertainty). For instance, the popular Upper Confidence Bound (UCB) acquisition function is formulated as:

$$\alpha_t^{\mathrm{UCB}}(x) = \mu_t(x) + \sqrt{\beta_t}\, \sigma_t(x), \tag{7}$$

where $\beta_t > 0$ is a parameter scaling the exploration incentive. The complete technical details of the posterior inference, along with the formulation of other common acquisition functions, are provided in Appendix B.1.

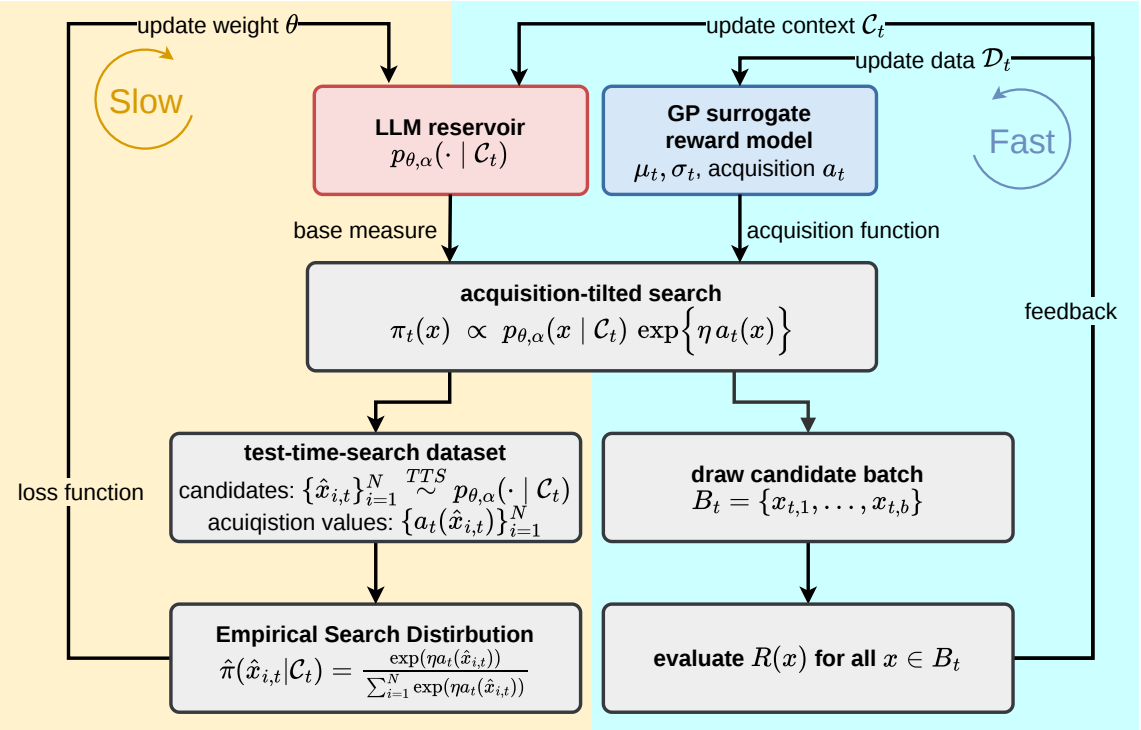


Figure 5: The Large Discovery Model loop. Th right (cyan) region denotes the *fast learning loop* of the LDM's recurrent discovery: a candidate batch B_t is drawn and evaluated, and the observations update the surrogate data $\mathcal{D}_t$ and the LLM context $\mathcal{C}_t$. The left (yellow) side corresponds to the *slow learning loop*, which amortises the acquisition-tilted search policy into the LLM's weights θ via LDM-TTS fine-tuning (see §3.4). All candidates $\{\hat{x}_{i,t}\}_{i=1}^N$ generated by the LLM during test-time search, alongside their respective acquisition values, are recorded to a TTS dataset. Using this dataset, we distil the empirical search distribution into the network, endowing the foundation model with generalisable discovery expertise to enable stronger discovery performance on subsequent tasks. The fast loop iterates with each batch of reward evaluation within a specific discovery task, while the slow loop is executed only after sufficient new samples spanning diverse tasks have been collected for the TTS dataset.

Because UCB decomposes the decision value into a high-mean and a high-uncertainty term, the tilted policy $\pi_t(x) \propto p_{\theta,\alpha}(x \mid \mathcal{C}_t) \exp\{\eta\, a_t(x)\}$ of Eq. (3) instantiates the discovery–exploration–exploitation triangle of Figure 3 directly: the LLM reservoir term realises discovery, the high-σ_t tail of a_t realises exploration, and the high-μ_t mode realises exploitation. We use UCB as the running example precisely because the trade-off is explicit in its closed form; in richer acquisitions the same triangle is realised only implicitly through the surrogate.

3.3 Acquisition-Tilted Inference-Time Search

Exact evaluation or sampling of π_t is computationally intractable over the high-dimensional, combinatorial design spaces typical of scientific discovery. We therefore approximate the tilted distribution via inference-time search: the LLM proposes a finite pool of candidates, which are then reweighted by the acquisition function to align with the target distribution. This framework naturally supports test-time compute scaling, mirroring the broader paradigm in general LLM reasoning but guided by a calibrated, task-grounded value signal rather than model-internal confidence (Wang et al., 2024).

We consider two canonical approximation schemes. The deterministic best-of-N rule targets the mode of the tilted distribution: we draw N independent candidates from the LLM prior,

$$\{\hat{x}_{t,i}\}_{i=1}^{N} \sim p_{\theta,\alpha}(\cdot \mid \mathcal{C}_t),$$

and select the candidate with the highest acquisition score:

$$x_{t+1} = \underset{i\in\{1,\dots,N\}}{\arg\max}\, a_t(\hat{x}_{t,i}).$$

The proposal $p_{\theta,\alpha}$ may comprise a collection of hyperparameters, e.g., temperature and top-p for an autoregressive decoder, or a noise schedule for a diffusion sampler, depending on the backend. The single explicit tunable below is η; the rest of $p_{\theta,\alpha}$ is held fixed.

The stochastic scheme approximates sampling from the full tilted distribution via softmax reweighting, where sampling probability is proportional to the exponentiated acquisition value:

$$\Pr(x_t = \hat{x}_{t,i}) = \frac{\exp\{\eta\, a_t(\hat{x}_{t,i})\}}{\sum_{j=1}^{N} \exp\{\eta\, a_t(\hat{x}_{t,j})\}}. \tag{8}$$

For batch selection with $b > 1$, these generalise naturally to top-b selection for the deterministic case and Gumbel-top-b sampling for the stochastic case.

Candidate pool size N plays the critical role of *test-time compute scaling*, a powerful technique for LLM reasoning (Wang, 2025), and LDM adapts this core advantage to discovery tasks. Larger N improves the fidelity of the approximation to the optimal tilted distribution and lifts empirical performance, at the cost of additional LLM inference and surrogate evaluation. This aligns with the established finding that test-time compute can substitute for model scale in LLM reasoning, with the key distinction that LDM uses an externally calibrated acquisition function as guidance, rather than the model's own self-evaluation or linguistic plausibility.

The full-candidate reweighting scheme is simple, modality-agnostic, and forms the basis of our experiments. For sequentially generated design spaces such as sequences and programs, the principle can be extended to intermediate generation steps via a prefix value function $V(x_{t,\le i}) = \mathbb{E}_{x_{t,>i}}[a_t(x_{t,\le i}, x_{t,>i})]$, enabling more advanced methods such as beam search and Monte Carlo Tree Search (Feng et al., 2024). These multi-step variants offer further search coverage at higher computational cost, and we leave their full empirical evaluation as future work.

3.4 Fine-Tuning to Reduce Test-Time Search

The LDM policy in Equation (3) combines the structured generative prior (reservoir) $p_{\theta,\alpha}(\cdot \mid \mathcal{C}_t)$ with the acquisition function a_t. A high-budget implementation of LDM test-time search, denoted by LDM-TTS, approximates this policy by generating a large pool of candidates and using the acquisition function to rank, select, and refine the most valuable ones. Although effective, repeating this procedure at every search round can require substantial inference-time computation. We therefore propose to distil the behaviour of high-budget LDM-TTS into a more efficient student proposer through fine-tuning, as illustrated in Figure 5.

At iteration t, the acquisition value of a candidate x may be written abstractly as

$$a_t(x) = \mathrm{Acq}(\mu_t(x), \sigma_t(x), g_t(x)), \tag{9}$$

where $\mu_t(x)$ and $\sigma_t(x)$ are the surrogate's posterior predictive mean and epistemic uncertainty, respectively. The term $g_t(x)$ collects additional task-dependent considerations, such as constraint satisfaction, novelty, evaluation cost, diversity, or expected improvement of the current Pareto frontier.

Suppose that high-budget LDM-TTS generates a candidate pool

$$\widehat{\mathcal{X}}_t = \{\hat{x}_{t,i}\}_{i=1}^{N}$$

under the search context $\mathcal{C}_t$. We define an acquisition-weighted empirical distribution over this pool by

$$\widehat{\pi}_t(\hat{x}_{t,i} \mid \mathcal{C}_t) = \frac{\exp\{\eta\, a_t(\hat{x}_{t,i})\}}{\sum_{j=1}^{N} \exp\{\eta\, a_t(\hat{x}_{t,j})\}}, \tag{10}$$

where $\eta > 0$ controls how strongly the distribution concentrates on candidates with high acquisition values. This finite-pool distribution approximates the ideal acquisition-tilted policy in Equation (3).

Let $p_\phi(x \mid \mathcal{C}_t)$ denote the student proposer, initialised from the underlying generative model. We fine-tune it by minimising the acquisition-weighted negative log-likelihood

$$\mathcal{L}_{\text{distill}}(\phi) = -\sum_{t} \sum_{i=1}^{N} \widehat{\pi}_t(\hat{x}_{t,i} \mid \mathcal{C}_t) \log p_\phi(\hat{x}_{t,i} \mid \mathcal{C}_t). \tag{11}$$

Equivalently, this objective projects the finite-pool teacher policy $\widehat{\pi}_t$ onto the family of distributions represented by the student model. The resulting proposer assigns greater probability to candidates that high-budget LDM-TTS would regard as valuable, allowing some of the benefits of test-time search to be transferred into the model parameters.

Importantly, the training signal is richer than the single candidate eventually selected for costly empirical evaluation. The unselected candidates record which alternatives the generative model considered and how the surrogate ranked their exploitation, exploration, and discovery value under the same search context. Fine-tuning on this information amortises part of the high-budget search process: the student learns to propose stronger candidates directly and can therefore achieve comparable search quality with a smaller candidate pool or fewer refinement steps at test time.

Because the surrogate and acquisition function change as new empirical observations are collected, the distilled policy is associated with the search states on which it was trained. It may therefore be updated periodically as the empirical dataset and search frontier evolve, rather than being treated as a permanently fixed replacement for online LDM search.

Reasoning augmentation. Acquisition values rank candidates but do not explicitly state why a candidate is useful for research progress. We therefore optionally augment selected teacher traces with a short rationale

$$z_{t,i} = \text{Explain}\Big(\mathcal{C}_t, \hat{x}_{t,i}, \mu_t(\hat{x}_{t,i}), \sigma_t(\hat{x}_{t,i}), g_t(\hat{x}_{t,i}), a_t(\hat{x}_{t,i})\Big). \tag{12}$$

The rationale is grounded in the surrogate value decomposition. It can explain whether a candidate exploits high predicted utility, probes an uncertain region, satisfies a constraint, preserves diversity, or tests a new candidate family after search stagnation. Thus, unlike reasoning traces for closed-world problems, discovery rationales explain why a scarce experiment is worth running.

We construct an SFT dataset of tuples $(\mathcal{C}_t, z_{t,i}, \hat{x}_{t,i})$, retaining or sampling candidate traces according to $\widehat{\pi}_t^{(N)}$. Let $\bar{w}_{t,i}$ be normalized weights derived from Eq. (10). A student model p_ϕ is trained with the weighted autoregressive objective

$$\mathcal{L}_{\text{SFT}}(\phi) = -\sum_{t} \sum_{i=1}^{N} \bar{w}_{t,i} \left[\log p_\phi(z_{t,i} \mid \mathcal{C}_t) + \log p_\phi(\hat{x}_{t,i} \mid \mathcal{C}_t, z_{t,i})\right]. \tag{13}$$

Algorithm 1 The Large Discovery Model Loop

Require: Reward function $R : \mathcal{X} \to \mathbb{R}$; prior mean $m(\cdot)$; kernel $k(\cdot, \cdot)$; initial dataset $\mathcal{D}_1$; initial context $\mathcal{C}_1$; evaluation budget T; batch size b; candidate pool size N.

for $t = 1, \ldots, T$ **do**
  Update the active decision domain: $A_t \leftarrow \operatorname{supp} p_{\theta,\alpha}(\cdot \mid \mathcal{C}_t)$
  Fit Gaussian-process surrogate over A_t using data $\mathcal{D}_t$ to obtain the posterior $R \mid \mathcal{D}_t \sim \mathcal{GP}(\mu_t, k_t)$, given by Eqs. (32), (33) ▷ Full derivation in Appendix B.1.
  Construct acquisition function a_t by combining $\{\mu_t, \sigma_t\}$ ▷ e.g. Eq. (35) or (34).
  Draw $B_t = \{x_{t,1}, \ldots, x_{t,b}\}$ from $\pi_t \propto p_{\theta,\alpha}(\cdot \mid \mathcal{C}_t) \exp\{\eta\, a_t\}$ using Algorithm 2
  Observe black-box noisy rewards $r_{t,i} = R(x_{t,i}) + \epsilon_{t,i}$ for all $x_{t,i} \in B_t$
  Update dataset: $\mathcal{D}_{t+1} \leftarrow \mathcal{D}_t \cup \{(x_{t,i}, r_{t,i})\}_{i=1}^{b}$
  Update context: $\mathcal{C}_{t+1} \leftarrow \mathcal{C}_t \cup \{(x_{t,i}, r_{t,i})\}_{i=1}^{b} \cup \{\text{Reflection feedback (if applicable)}\}$
return $\arg\max_{(x,r)\in\mathcal{D}_{T+1}} r$

Algorithm 2 Acquisition-Guided Inference-time Sampling

Require: LLM prior $p_{\theta,\alpha}(\cdot \mid \mathcal{C}_t)$; acquisition function a_t; tilt parameter η; pool size N; batch size b; selection mode $\in \{\text{deterministic}, \text{stochastic}\}$.

Sample N candidate designs from the LLM prior: $\{\hat{x}_{t,i}\}_{i=1}^{N} \sim p_{\theta,\alpha}(\cdot \mid \mathcal{C}_t)$
Compute acquisition scores $s_i \leftarrow a_t(\hat{x}_{t,i})$ for $i = 1, \ldots, N$
if selection mode = deterministic **then**
  Let $i_1, \ldots, i_b$ be the indices of the b largest scores s_i, and set $B_t \leftarrow \{\hat{x}_{t,i_k}\}_{k=1}^{b}$ ▷ Top-b by acquisition.
else if selection mode = stochastic **then**
  Compute exponentiated-acquisition weights $w_i \leftarrow \exp\{\eta\, s_i\}$ for $i = 1, \ldots, N$
  Sample $B_t = \{\hat{x}_{t,i_k}\}_{k=1}^{b}$ from the weights $w_1, \ldots, w_N$ via Gumbel-top-b ▷ See §B.3.
return B_t

Equivalently, trajectories may be sampled from $\widehat{\pi}_t^{(N)}$ and optimised with an unweighted likelihood objective. The rationale is an auxiliary training target and can be omitted at deployment.

Fine-tuning shifts the student reservoir toward candidate families that high-budget LDM-TTS repeatedly identifies as acquisition-relevant. In the regret decomposition of Section 5, this can reduce the discovery gap by improving reservoir coverage and reduce the LDM sampling shortfall by increasing the near-acquisition reservoir mass $\kappa_t(\zeta_t)$. Fine-tuning does not replace the online surrogate: new observations must still update μ_t, σ_t, and a_t. Instead, it reduces the candidate-pool size and inner-search budget needed before the surrogate can identify valuable experimental actions.

4 The Algorithm

In this section, we present the algorithm to apply an LDM in sequential experimental designs for scientific discovery. As illustrated in Figure 5, the LDM drives an iterative loop that tightly coordinates a generative foundation model with a data-driven verification surrogate. Each round constructs the tilted search policy, draws a batch of promising candidates, evaluates their with real experiments, and updates both the numeric results and the generative context with new observations.If the surrogate identifies a stale regime (cf. §6), the LLM is invoked to append a *reflection feedback* summary to $\mathcal{C}_t$, thereby ensuring that subsequent proposals remain aligned with the most recent mechanistic understanding. This operation is orthogonal to the tilted-search core. The complete iterative procedure is formalised in Algorithm 1, while the inner

inference-time sampling loop is specified in Algorithm 2.

4.1 Policy Construction and Realisation

We construct the approximate tilted policy by reweighting an LLM-derived base measure with the acquisition function. Depending on the design space, the base measure $p_{\theta,\alpha}(\cdot \mid \mathcal{C}_t)$ can be instantiated in two ways.

Direct Generation. The LLM directly generates complete candidate designs from its autoregressive decoding distribution. Invalid candidates can be removed by rejection sampling with a validity checker.

Indirect Parameterisation. When direct generation of valid designs is inefficient, the LLM instead specifies a parameterised search region, such as its centre, radius, or local bounds. An external sampler then generates candidates within this region. This decouples LLM inference from candidate generation and allows the pool size N to be scaled more efficiently.

Both paradigms integrate seamlessly with the inference-time sampling pipeline. The choice between deterministic top-b selection and stochastic Gumbel sampling depends on the use case: deterministic selection maximises expected acquisition value for a given pool size, while stochastic sampling promotes structural diversity and reduces redundant evaluation in batch settings. The candidate pool size N serves as the primary dial for test-time compute scaling, enabling a continuous trade-off between inference cost and search quality.

4.2 Surrogate on the Dynamic Support

Under indirect parameterisation, the LLM defines not only candidate proposals but also the active search domain. We write this domain as

$$A_t := \operatorname{supp} p_{\theta,\alpha}(\cdot \mid \mathcal{C}_t) \subseteq \mathcal{X}.$$

Here, $\mathcal{X}$ may denote the full space of experimental designs, while A_t is a lower-dimensional parameterised subspace selected by the LLM at round t, such as a set of schedule variables, architectural choices, or local search bounds.

This restricted support simplifies both search and surrogate modelling. Rather than modelling the full space $\mathcal{X}$, the GP is defined over A_t and can be fit using historical observations whose designs lie in the current support: $\mathcal{D}_t|_{A_t} = \{(x, r) \in \mathcal{D}_t : x \in A_t\}$, where r denotes the observed reward associated with design x. Because A_t is explicitly parameterised, the GP kernel and mean function can be defined directly in this reduced space. As the LLM changes or expands the active search domain across rounds, the surrogate is updated accordingly on the new A_t.

4.3 Practical Implementation

The framework supports several extensions commonly needed in scientific discovery: (1) *constrained and cost-aware acquisition*, where feasibility constraints and heterogeneous evaluation costs are incorporated through constrained expected improvement or cost-normalised acquisition functions; (2) *decomposed acquisition feedback*, where surrogate outputs such as predictive mean, epistemic uncertainty, constraint slack, and estimated cost are provided separately to the LLM to support more targeted proposal updates; (3) *multi-objective optimisation*, where the scalar acquisition function is replaced by Expected Hypervolume Improvement (EHVI) for vector-valued objectives; and (4) *batch selection*, where, for parallel evaluations with $b > 1$, candidates

are sampled without replacement from the finite-pool tilted distribution using Gumbel-top-b sampling (see §B.3). Technical details are provided in Appendix B.

5 Theoretical Analyses

We now present theoretical analyses of the LDM. This section is self-contained; readers primarily interested in the empirical results may safely skip it. Let $\mathcal{X}$ denote the design space, and let $x^\star \in \arg\max_{x\in\mathcal{X}} R(x)$ be a global maximiser of the unknown objective function. Note that performance is measured with respect to the true objective R, rather than the uncertainty-aware acquisition function. After T trials, the simple regret (no separate symbol beyond the per-round regret) is defined as

$$\mathrm{Reg}_T = R(x^\star) - \max_{1\le t\le T} R(x_t). \tag{14}$$

At round t, let $p_\theta(\cdot \mid \mathcal{C}_t)$ denote the LLM reservoir distribution after applying the actual decoding restrictions of §method (top-k or top-p truncation, etc.). The corresponding temperature-adjusted proposal is

$$p_{\theta,\alpha}(x \mid \mathcal{C}_t) \;\propto\; p_\theta(x \mid \mathcal{C}_t)^\alpha,$$

and its support (the LLM-controlled dynamic search domain of §4) is

$$A_t \;:=\; \mathrm{supp}\; p_{\theta,\alpha}(\cdot \mid \mathcal{C}_t) \;=\; \{x \in \mathcal{X} : p_{\theta,\alpha}(x \mid \mathcal{C}_t) > 0\}.$$

The LDM then samples the next candidate $x_t \sim \pi_t$, where

$$\pi_t(x) = \frac{\exp(\eta\, a_t(x))\, p_{\theta,\alpha}(x \mid \mathcal{C}_t)}{\sum_{u\in A_t} \exp(\eta\, a_t(u))\, p_{\theta,\alpha}(u \mid \mathcal{C}_t)}. \tag{15}$$

Here, $a_t(x) = \mu_t(x) + \sqrt{\beta_t}\sigma_t(x)$ is the UCB acquisition function (matching the LDM convention of §method).

To relate this performance criterion to the per-round behaviour of the algorithm, define the instantaneous regret at round t as $\mathrm{Reg}_t = R(x^\star) - R(x_t)$. Here, t indexes evaluations under the true objective R, rather than internal LLM interaction steps. Equivalently, the simple regret after T trials is $\mathrm{Reg}_T = \min_{1\le t\le T} \mathrm{Reg}_t$. In the analysis below, we first study the cumulative behaviour of $\{\mathrm{Reg}_t\}_{t=1}^T$, from which a simple regret guarantee follows via the standard relation $\mathrm{Reg}_T \le \frac{1}{T}\sum_{t=1}^T \mathrm{Reg}_t$.

Instantaneous Regret Decomposition as Discovery Gap and Optimisation Regret. For each round t, define the best objective value attainable within the current reservoir support as $R^\star_{t,A} = \max_{x\in A_t} R(x)$. Then, the instantaneous regret admits the exact decomposition

$$\mathrm{Reg}_t = R(x^\star) - R(x_t) \tag{16}$$

$$= \underbrace{R(x^\star) - R^\star_{t,A}}_{\mathrm{Reg}_t^{\mathrm{disc}}} + \underbrace{R^\star_{t,A} - R(x_t)}_{\mathrm{Reg}_t^{\mathrm{opt}}}. \tag{17}$$

The first term, $\mathrm{Reg}_t^{\mathrm{disc}}$, is the *discovery gap*: it measures the loss incurred because the current reservoir support A_t may not contain designs sufficiently close to the global optimum. This term is zero whenever $x^\star \in A_t$. The second term, $\mathrm{Reg}_t^{\mathrm{opt}}$, is the *optimisation regret* within the currently discovered region: it measures how far the sampled point x_t is from the best candidate available in A_t.

This decomposition separates the two roles of the LDM. The reservoir distribution controls whether high-quality regions are discovered at all, while the acquisition tilt controls how effectively the algorithm selects promising candidates among those currently reachable.

Below, we will give the necessary assumptions and definitions to support the regret bound analysis.

Assumption 1 (UCB calibration)**.** For a non-decreasing sequence $(\beta_t)_{t\geq 1}$, with probability at least $1-\delta_{\mathrm{gp}}$, simultaneously for all $t \leq T$ and $x \in \mathcal{X}$,

$$|R(x) - \mu_t(x)| \leq \sqrt{\beta_t}\,\sigma_t(x).$$

This event is implied by the standard kernel bandit conditions: $R(x) = m(x) + g(\phi(x))$ with $g \in \mathcal{H}_k$, bounded RKHS norm, bounded kernel diagonal, and conditionally sub-Gaussian observation noise (Srinivas et al., 2010; Chowdhury and Gopalan, 2017).

Assumption 2 (Reservoir coverage and local regularity)**.** Let $\mathrm{d}_{\mathcal{X}}$ be a distance on the design space and define the reservoir coverage radius $r_t^{\mathrm{cov}} = \inf_{x\in A_t} \mathrm{d}_{\mathcal{X}}(x, x^\star)$. The objective is one-sided Hölder around the optimum: there exist $L > 0$ and $\chi \in (0, 1]$ such that, for all $x \in \mathcal{X}$,

$$R(x^\star) - R(x) \leq L\,\mathrm{d}_{\mathcal{X}}(x, x^\star)^\chi.$$

Let $a^\star_{t,A} = \sup_{x\in A_t} a_t(x)$ be the maximum value of the acquisition function at t-th round. Then, for any tolerance $\zeta_t \geq 0$, let $U_t(\zeta_t) = \{x \in A_t : a_t(x) \geq a^\star_{t,A} - \zeta_t\}$ be the set of candidates in the current reservoir support A_t whose acquisition value is within ζ_t of the best acquisition value. We denote the reservoir probability mass assigned to this near-best set by

$$\kappa_t(\zeta_t) = p_{\theta,\alpha}\big(U_t(\zeta_t) \mid \mathcal{C}_t\big). \tag{18}$$

The bound below depends on the size of $\kappa_t(\zeta_t)$: a small near-UCB mass means that the LLM reservoir makes high-acquisition designs hard to sample even after exponential tilting.

$$\kappa_t(\zeta) \;\geq\; \kappa_0(\zeta) \;+\; \rho_\zeta \cdot N_t^{\mathrm{good}}, \tag{19}$$

where $N_t^{\mathrm{good}} := |\{(x_i, r_i) \in \mathcal{C}_t : r_i \geq r^{\mathrm{thresh}}\}|$ counts high-reward exemplars in the context. Equation (19) formalises the adaptive condition: the near-acquisition reservoir mass grows monotonically as good exemplars accumulate.

Lemma 1 (Gibbs near-UCB sampling)**.** *Conditioned on the history $\mathcal{C}_t$, fix any $\zeta_t \geq 0$ such that the near-UCB set has reservoir mass $\kappa_t(\zeta_t)$. Then, for any $\rho_t \in (0, 1)$, we have*

$$\mathbb{P}\left(a^\star_{t,A} - a_t(x_t) \leq \zeta_t + \frac{1}{\eta}\log\frac{1}{\rho_t\kappa_t(\zeta_t)} \,\middle|\, \mathcal{C}_t\right) \geq 1 - \rho_t. \tag{20}$$

Let

$$\gamma_T = \sup_{B\subset\mathcal{X},\ |B|\leq T} \frac{1}{2}\log\det(I + \lambda^{-1}K_B), \qquad C_\lambda = \frac{2}{\log(1+\lambda^{-1})},$$

where K_B is the kernel matrix on B.

Theorem 1 (Simple regret of large discovery model)**.** *Suppose Assumptions 1 and 2 hold. Fix $\delta_{\mathrm{gp}}, \delta_{\mathrm{samp}} \in (0, 1)$ and choose $\rho_t > 0$ such that $\sum_{t=1}^T \rho_t \leq \delta_{\mathrm{samp}}$. Then, for any $\zeta_t \geq 0$ with $\kappa_t(\zeta_t) > 0$, with probability at least $1 - \delta_{\mathrm{gp}} - \delta_{\mathrm{samp}}$,*

$$\frac{1}{T}\sum_{t=1}^T \mathrm{Reg}_t \leq \underbrace{\frac{L}{T}\sum_{t=1}^T (r_t^{\mathrm{cov}})^\chi}_{\textit{discovery gap}} + \underbrace{2\sqrt{\frac{C_\lambda\beta_T\gamma_T}{T}}}_{\textit{GP-UCB optimisation}} + \underbrace{\frac{1}{T}\sum_{t=1}^T\left[\zeta_t + \frac{1}{\eta}\log\frac{1}{\rho_t\kappa_t(\zeta_t)}\right]}_{\textit{LDM sampling shortfall}}.$$

The first term is the gap of not yet having generated candidates near the true optimum. The second is the usual GP-UCB information-gain term within the covered region. The third is the cost of sampling from the tilted LDM distribution rather than exactly maximising UCB over A_t; it decreases when the acquisition tilt η is sharper or when the near-UCB reservoir mass $\kappa_t(\zeta_t)$ is larger.

The LLM affects the regret bound through the reservoir support A_t and its distribution $p_{\theta,\alpha}(\cdot \mid \mathcal{C}_t)$. First, an informative LLM can propose candidates closer to the global optimum, thereby reducing the coverage radius r_t^{cov} and hence the discovery-gap term. Second, it can assign substantial probability mass to candidates with high acquisition values, increasing $\kappa_t(\zeta_t)$ and reducing the LDM sampling-shortfall term. Thus, the LLM provides a structured, history-dependent proposal distribution that can focus evaluations on semantically plausible and promising regions of the design space, rather than searching the entire space uniformly.

6 Experiments and Case Studies

We evaluate the LDM through three case studies that cover distinct structured search domains. §6.1 presents an autonomous-research loop over neural-network training programs (Karpathy, 2026); §6.2 demonstrates the competitive performance of LDM in antibody CDRH3 sequence design (Khan et al., 2022); and §6.3 attests to the effectiveness of LDM in multi-objective small-molecule drug discovery against the KRAS G12D target (Kumar et al., 2026). §6.4 then distils the LDM test-time search policy into a reusable proposer across all three domains, and §6.5 isolates which components of LDM actually drive the gains.

6.1 Autonomous research loop

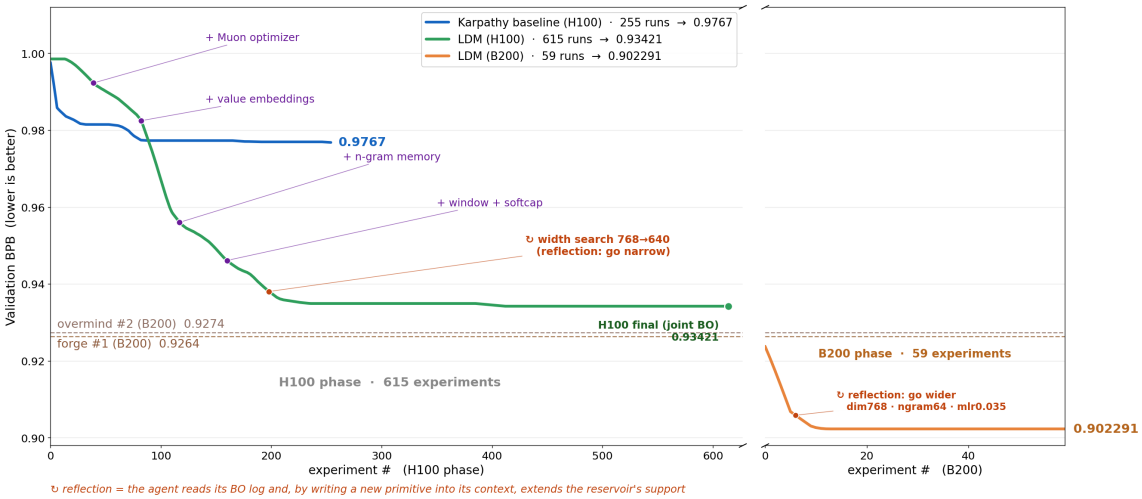


Figure 6: **The discover loop on `autoresearch`.** Validation `val_bpb` (lower is better) versus search progress. The H100 panel compares the no-discover Karpathy baseline with the LDM; the B200 panel reports a matched-hardware leaderboard run of the LDM. B200 width scaled for legibility.

We use Karpathy's public `autoresearch` project (Karpathy, 2026) as a testbed for acquisition-guided LLM search. In `autoresearch`, a coding agent repeatedly modifies a single file, `train.py`; each candidate program is evaluated by training a small language model for a fixed five-minute budget and measuring validation bits per byte (`val_bpb`, lower is better). The original system provides only the LLM proposal-and-evaluation loop, to which we add the surrogate and

Change	Why it helps under a fixed time budget	Δval_bpb
n-gram hash memory	memorises local n-gram patterns at zero FLOP	−0.0275
QK-norm	steadier steps, more learning per step	−0.0137
Narrower model (dim 768→640)	cheaper steps ⇒ more steps in 300 s	−0.0109
Windowed attention + softcap	cheaper attention ⇒ more tokens/sec	−0.0108
Squared-ReLU	richer per-token MLP representation	−0.0087
Value embeddings	extra per-token signal at near-zero cost	−0.0079
Muon optimizer	orthogonalised gradients, more learning per token	−0.0048

Table 1: **The `autoresearch` run as physics-grounded discovery.** Each row lists a change the LDM retained under a fixed 300 s budget, the physical mechanism by which it helps, and the resulting change in `val_bpb`.

acquisition-guided search. Under the LDM notation, a program is a design x with reward

$$R(x) = -\texttt{val\_bpb}(x),$$

where the agent conditioned on the current context defines the proposal distribution $p_\theta(\cdot \mid \mathcal{C}_t)$, and the accumulated program–performance pairs form the dataset $\mathcal{D}_t$ used to fit the surrogate. This benchmark provides a natural setting for LDM: candidates are executable programs, each evaluation returns a quantitative non-differentiable objective, and the fixed training schedule enables controlled comparison across search methods.

6.1.1 Experiment setting

We instantiate LDM using indirect parameterisation. Rather than generating raw program text, the LLM identifies tunable degrees of freedom in `train.py`, such as learning rate, width, depth, and related hyperparameters, thereby defining a continuous search space. We fit a Gaussian process with an RBF kernel over this space, keeping its hyperparameters fixed for stability in the small-data regime. Expected Improvement (Eq. (34)) is used as the acquisition function.

Each round the LLM proposes a pool of candidate configurations conditioned on $\mathcal{C}_t$, and the surrogate selects the next configuration to evaluate by maximising EI, with a diversity term for batched rounds. When progress stalls, reflection allows the LLM to revise the parameterisation by adding or reshaping search dimensions. The primary comparison holds the coding model, environment, and five-minute-per-run budget fixed, contrasting the original Karpathy LLM-only loop against the full LDM.

6.1.2 Results

Figure 6 shows an initial period of improvement followed by a plateau and a second improvement phase after the search space is revised. Under the same H100 five-minute-per-run schedule, the Karpathy LLM-only baseline plateaus at `val_bpb`≈ 0.9767 after roughly 255 runs, while LDM continues to 0.93421. This comparison measures the end-to-end effect of the full system; component-level results are reported in Section 6.5 and Appendix C.

Separately, Figure 6 reports a matched-hardware leaderboard run on B200. LDM reaches `val_bpb` = 0.902291 in 59 runs, compared with FORGE (0.9264) and OVERMIND (0.9274) on the `autoresearch@home` leaderboard.[2] Under the fixed training-time budget, improvements must come from processing more tokens or learning more effectively from each token. Table 1 summarises the retained changes and their measured contributions, reducing `val_bpb` from 0.9961 to 0.902291 overall.

[2] https://www.ensue-network.ai/lab/autoresearch

6.2 Antibody CDRH3 design

We select antibody CDRH3 loop design as our second case study because it represents a canonical high-dimensional combinatorial optimisation problem with a well-defined design space. The CDRH3 loop is the primary determinant of antibody binding specificity, with sequences of length $L \approx 11$ drawn from 20 amino acids, yielding a discrete space of size 20^{11}. This setting falls into the *known unknown* regime: the syntax and boundaries of the design space are fully specified, but the sequence-to-affinity mapping is unknown, highly rugged, and expensive to evaluate. Traditional combinatorial Bayesian optimisation methods struggle with the intractable size of this discrete space, while pure LLM generation lacks calibrated value guidance. The task therefore tests whether LDM can combine structured proposal generation with acquisition-guided selection in a large discrete search space.

We use the Absolut! structure-based simulator as a black-box oracle that returns the lattice binding energy E_{bind} (Khan et al., 2022). We maximise $R(x) = -E_{\text{bind}}(x)$ because lower binding energy implies stronger affinity, and each evaluation is a single expensive oracle call.

6.2.1 Experimental settings

Experiments run sequentially, with one antibody candidate selected per iteration. We evaluate across five PDB targets, including 1ADQ_A, 1FBI_X, 1H0D_C, 1NSN_S, and 1OB1_C, with a total budget of 200 evaluations per target, reporting mean performance and one standard deviation over multiple random seeds.

The baseline methods considered in this study can be grouped into four categories: **Pure LLM** approaches, which perform direct autoregressive sequence generation; combinatorial Bayesian optimisation (BO) frameworks (**TURBO**, **COMBO**) that have been adapted to operate over discrete sequence spaces; the domain-specialised optimiser **AntBO**; and a naive random search strategy. Our goal is to position LDM as a competitive, general-purpose optimisation methodology, rather than to claim superiority over the domain-specialised AntBO approach.

We implement four LDM variants formed by two orthogonal design choices: two LLM base generation strategies, paired with two acquisition weighting rules. Here **Policy** denotes the indirect parameterisation of §4: the LLM parameterises a sampling policy (with internal complexity), and the policy implicitly induces the base measure $p_{\theta,\alpha}(\cdot \mid \mathcal{C}_t)$. The **Direct** base measure generates sequences token-by-token with a small sample pool of $m = 5$. The **Policy** base measure lets the LLM output structured search parameters to spawn a much larger candidate pool. For weighting, **Max** picks the highest-acquisition candidate (mode approximation), while **Softmax** samples proportionally to exponentiated acquisition scores to approximate the full tilted distribution. Table 2 summarises all combinations.

LDM Variant	Base Measure $p_\theta(x \mid \mathcal{C}_t)$	Reduction Rule
Direct_Max	Direct token-level sequence generation	Max
Direct_Softmax	Direct token-level sequence generation	Softmax
Policy_Max	Structured search policy from LLM output	Max
Policy_Softmax	Structured search policy from LLM output	Softmax

Table 2: Implemented LDM algorithm variants for computational antibody design.

6.2.2 Results

Figure 7 compares all methods across the five protein targets. Policy-based LDM variants consistently outperform direct generation variants, achieving affinity levels comparable to AntBO,

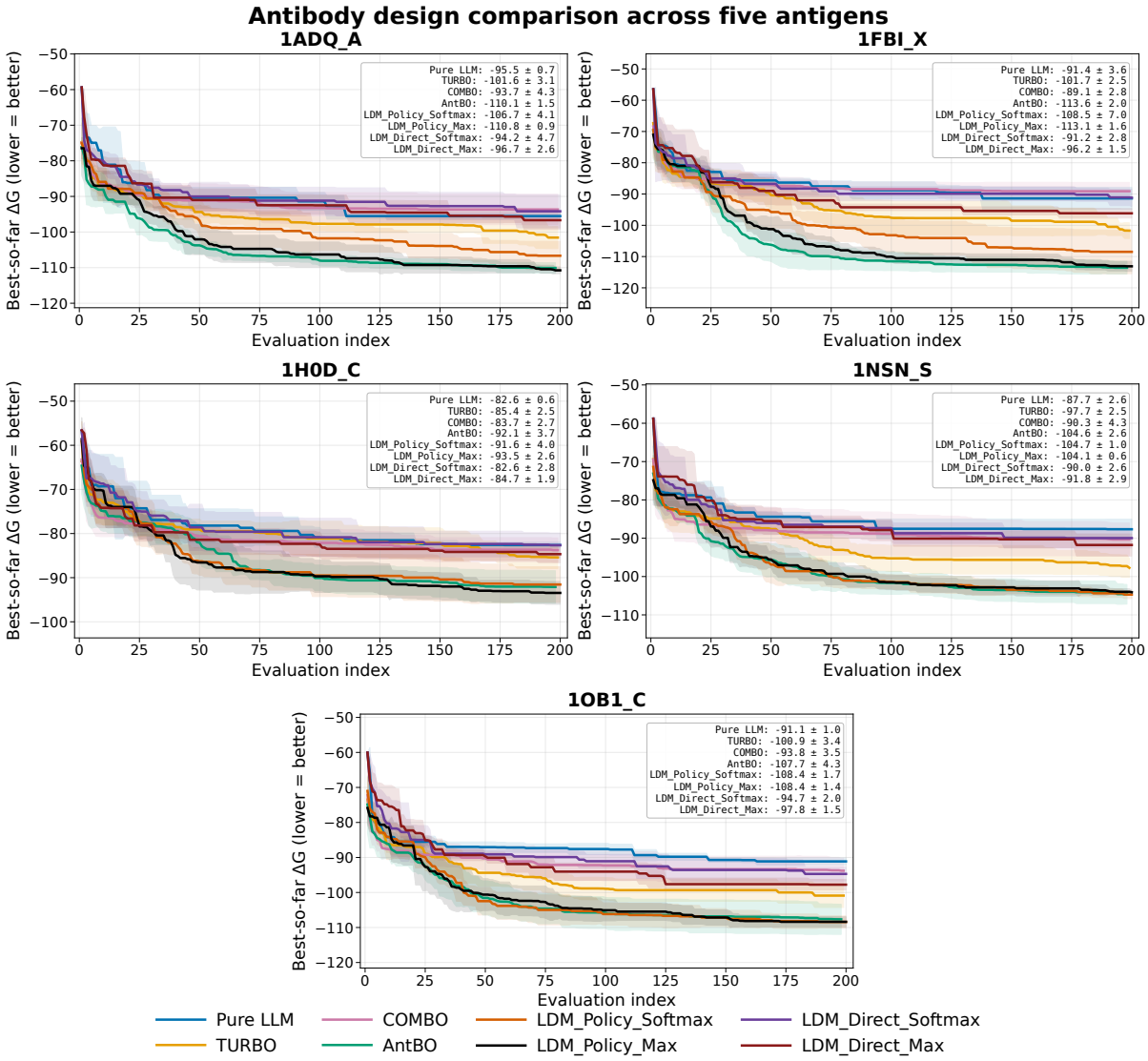


Figure 7: Binding affinity performance across five antibody antigens. Axes: horizontal = number of sequence evaluations; vertical = best-so-far binding energy score (lower values indicate stronger binding). Curves correspond to all tested baselines and LDM variants.

while Direct variants perform only marginally better than standalone LLMs.

Protein fitness landscapes are sparse and rugged, restricting the utility of small batches of directly generated sequences; even post-hoc acquisition reweighting cannot recover promising candidates the LLM fails to propose. In contrast, policy-mode LDM uses the LLM to define where to search rather than to emit final sequences, while the surrogate selects which candidates to evaluate; this expands the candidate pool without additional oracle calls.

Figure 8 visualises a representative LDM optimisation trajectory on the 1FBI_X antigen, illustrating the core exploit–explore–discover cycle described in §4. Initial LLM sampling quickly hits a local performance ceiling; the LDM detects stagnation and triggers discovery steps to expand the sequence families under consideration. Each numbered marker indicates a search-space update followed by further improvement and local refinement.

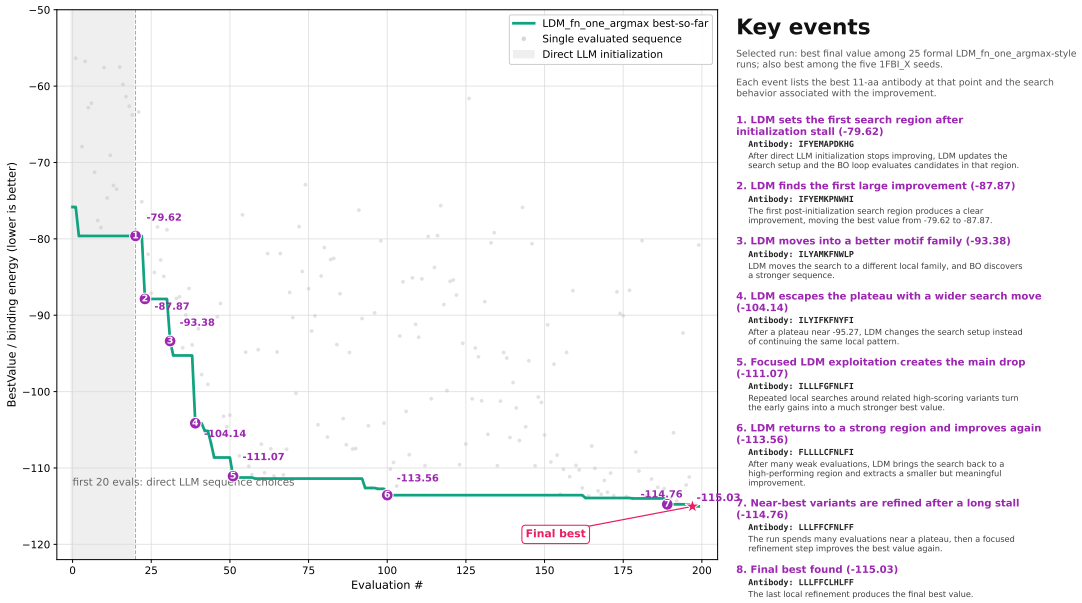


Figure 8: Optimisation trajectory of the top-performing LDM run on antigen 1FBI_X. Horizontal axis counts evaluation rounds; vertical axis tracks the best binding energy found to date (lower = better). Step curve records global improvement milestones, with numbered markers highlighting major discovery events where the system introduces structurally novel sequence families after local plateaus.

6.3 Small-molecule drug discovery

We select multi-objective small-molecule lead discovery as the third case study to test LDM's handling of the *unknown unknown* regime. Unlike the antibody setting, the molecular design space is not given as a fixed finite set of candidates. Valid SMILES span a vast and open-ended chemical space, making exhaustive enumeration infeasible. Standard multi-objective Bayesian optimisation (MOBO) with string kernels lacks strong structural priors and saturates rapidly. Auxiliary expansion tools such as ReaSyn only generate analogues around fixed seed molecules, limiting exploration to narrow local chemical neighbourhoods. This setting tests whether LDM can discover new molecular scaffolds beyond the initial seed neighbourhoods.

Our multi-objective task optimises two conflicting rewards for the KRAS G12D target (Kumar et al., 2026): the negated AutoDock Vina (Trott and Olson, 2010) docking score (R_{Vina}) and neural-network binding activity (R_{act}), both maximised. We quantify performance via the hypervolume of the Pareto front, with a fixed reference point across all experiments and a total evaluation budget of 80 iterations.

6.3.1 Experimental settings

Independent Gaussian Process surrogates are fitted for each reward dimension, using a subsequence string kernel to measure similarity between SMILES strings. Expected Hypervolume Improvement (EHVI) is used as the multi-objective acquisition function. Baselines include vanilla MOBO and random search. We evaluate two LDM variants that share the same tilted sampling pipeline but differ in candidate pool construction: (1) **Direct-Softmax**, where the LLM generates 128 SMILES candidates for acquisition-weighted resampling; and (2) **ReaSyn-Softmax**, where the LLM first proposes a small set of seed SMILES and ReaSyn generates local analogues to form the candidate pool before softmax selection.

6.3.2 Results

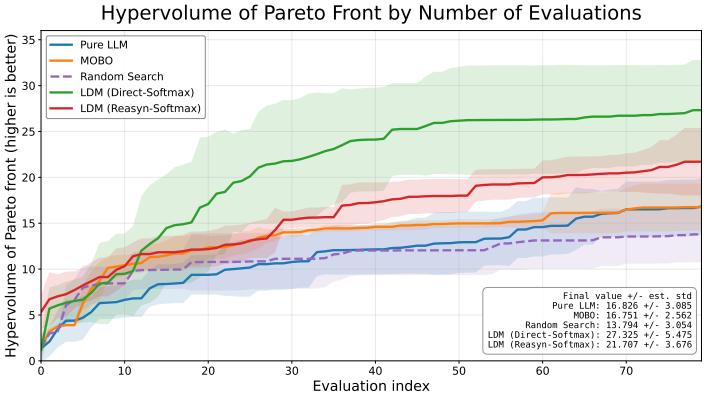


Figure 9: Pareto hypervolume growth for molecular design. Horizontal axis = evaluation count; vertical axis = dominated hypervolume of the Pareto front (higher values denote better multi-objective trade-offs). Separate curves track performance of random search, standard MOBO, and the two LDM variants.

Figure 9 shows that both LDM variants outperform all baselines by a substantial margin, with Direct-Softmax delivering the strongest hypervolume gains. Conventional MOBO plateaus after roughly 40 evaluations, trapped within limited chemical regions, while direct LLM generation sustains steady front expansion across the full budget.

The performance gap between the two LDM variants arises from a breadth–synthesizability trade-off. General-purpose LLMs carry broad pre-training knowledge of diverse SMILES structures; large-batch direct generation unlocks wider scaffold exploration. ReaSyn's local analogue generation improves synthetic tractability but restricts structural diversity, slowing Pareto expansion for early-stage lead discovery where novel scaffolds are prioritised.

Figures 10 and 11 together illustrate a complete LDM discovery workflow for small molecules. The hypervolume curve partitions optimisation into distinct stages: initial scaffold identification, substituent screening, objective bifurcation into docking- and activity-focused chemotypes, cross-family feature recombination, and final fine-tuning. The structural plot tracks how the model iteratively uncovers unrelated molecular backbones, a capability absent from standard BO or seed-local expansion tools.

6.4 Fine-Tuning Experiments

We evaluate discovery fine-tuning across three structured design domains: multi-turn program search in AutoResearch, combinatorial antibody CDRH3 design, and multi-objective molecular discovery. In every experiment, the fine-tuned proposer is placed inside the same acquisition-guided LDM loop as the base proposer. Fine-tuning therefore changes the proposal reservoir, whereas the online surrogate continues to provide value and uncertainty estimates from the current experimental history.

We compare action-only discovery fine-tuning with reasoning-augmented fine-tuning. The latter trains the proposer to generate a surrogate-grounded research-progress rationale before emitting the candidate action.

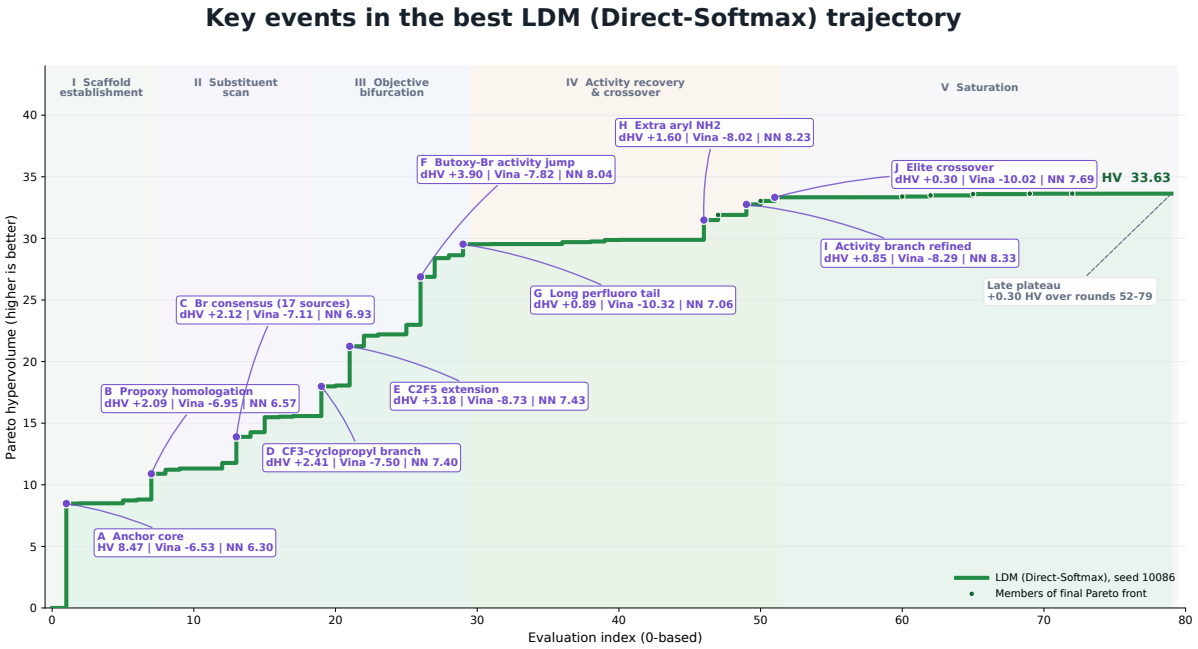


Figure 10: Key milestone events along a Direct-Softmax molecular optimisation trajectory. X-axis counts evaluation steps; Y-axis plots cumulative Pareto hypervolume (higher = better). Step curve tracks global front improvements, with labelled markers denoting structural discovery events that open new families of active molecules. Shaded bands partition the optimisation into five sequential search phases. In each text box, `HV` represents Hyper-Volumn, `Vina` represents the score of Autodock Vina, and `NN` represents the neural network as the oracle activity prediction.

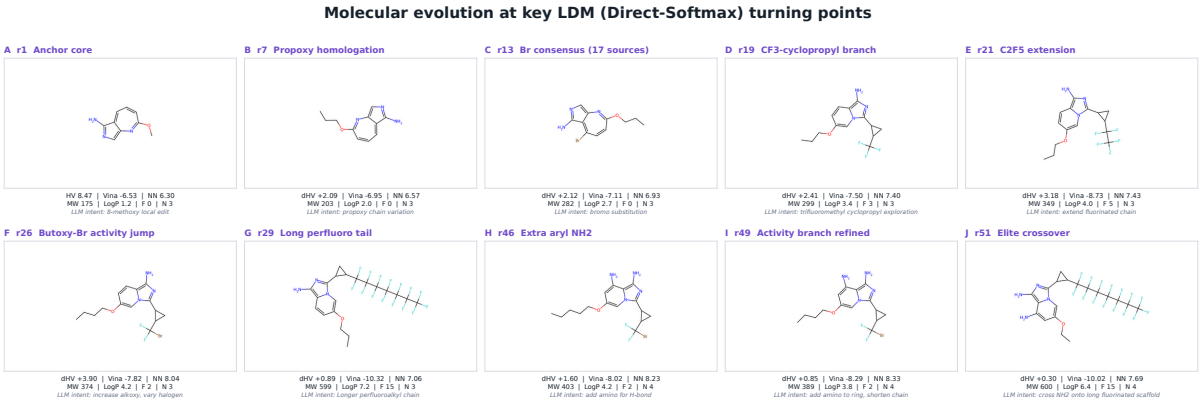


Figure 11: Two-dimensional molecular structures corresponding to the ten core discovery milestones labelled in Figure 10. Each structure is generated from canonical trajectory SMILES strings, arranged in chronological order of discovery to visualise the evolution of core chemical scaffolds and functional group substitutions.

6.4.1 AutoResearch

We first study fine-tuning in AutoResearch, where an agent repeatedly edits a persistent training program and evaluates each proposal through a fixed-budget nanoGPT training run. This setting requires the model to interpret an evolving experiment ledger, recognize performance plateaus, and propose changes that test new program mechanisms.

Figure 12 compares fine-tuned and base Qwen3.5-9B proposers over 80 LDM-TTS iterations. The reasoning-augmented fine-tuned model achieves the best validation performance, reaching

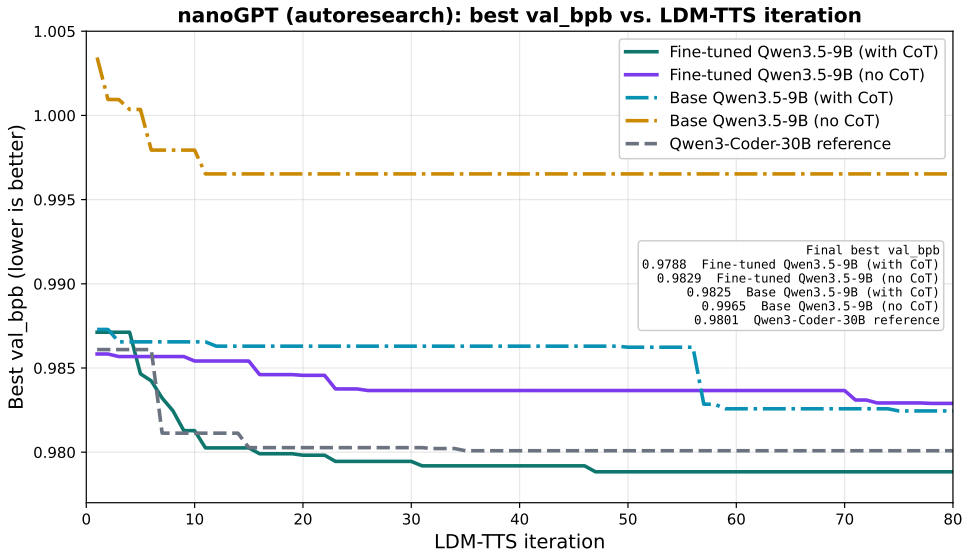


Figure 12: **Reasoning-augmentation ablation on AutoResearch.** Best validation bits-per-byte (lower is better) versus LDM-TTS iteration inside an identical acquisition-guided loop, comparing fine-tuned and base proposers with and without reasoning augmentation and the Qwen3-Coder-30B reference.

val_bpb = 0.9788, lower than the Qwen3-Coder-30B reference at 0.9801. Fine-tuning without reasoning reaches 0.9829, the base model with reasoning reaches 0.9825, and the base model without reasoning stalls at 0.9965. These results indicate that fine-tuning enhances the quality and diversity of the program-search reservoir, and that the explicit trace of research progress further facilitates the model's ability to escape local program families and explore a broader region of the program space.

6.4.2 Antibody CDRH3 Design

We next evaluate antibody CDRH3 design, where the proposer searches a sparse and rugged sequence space under a limited binding-energy evaluation budget. We compare fine-tuned and base Qwen3.5-9B models with and without reasoning augmentation across five antigens. All variants use the same acquisition-guided LDM loop with expected-improvement acquisition and the same parallel evaluation budget.

Figure 13 shows that fine-tuning improves over the corresponding base proposer across the five antigens. Both fine-tuned variants achieve lower binding energies than the base models, while the reasoning-augmented fine-tuned model obtains the lowest average binding energy among the four configurations. This result indicates that the distilled policy improves sequence-family proposals even in a large discrete design space.

6.4.3 Multi-objective Small-Molecule Discovery

Finally, we evaluate cross-task transfer on the KRAS G12D molecular-design task. The task jointly optimizes docking and neural activity, and performance is measured by best-so-far Pareto-front hypervolume over 80 expensive evaluations.

Figure 14 compares five inference-time policies inside the same LDM loop. The reasoning-augmented fine-tuned Qwen3.5-9B model achieves the highest final hypervolume, 26.660 ± 2.608. It exceeds the same fine-tuned student without reasoning, 22.279 ± 3.828, and both DeepSeek

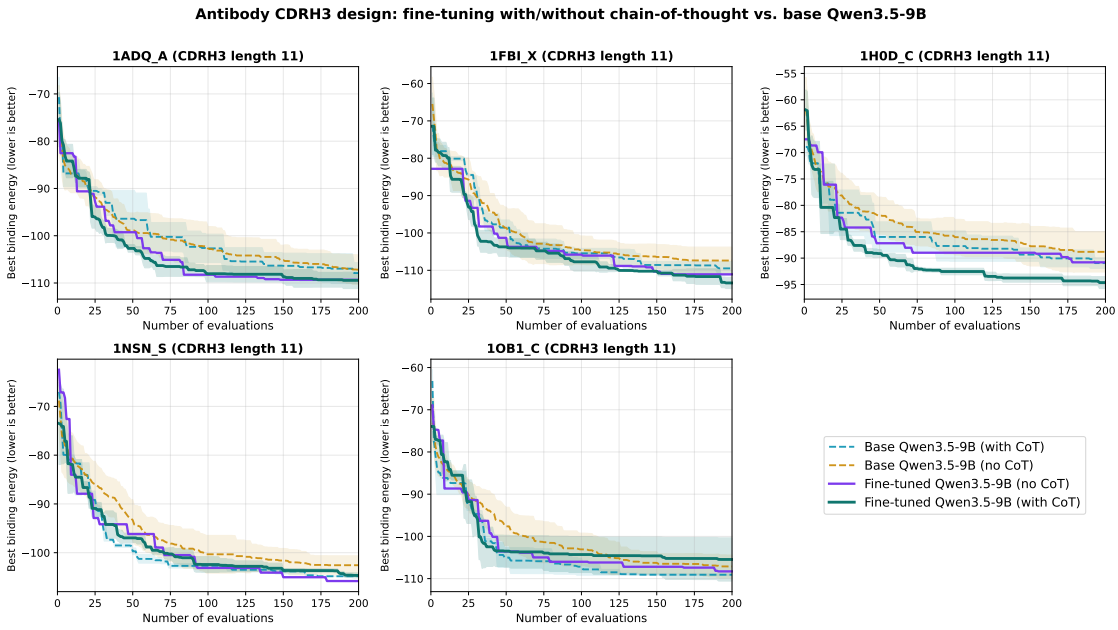


Figure 13: **Discovery fine-tuning for antibody CDRH3 design.** Best Absolut! binding energy, lower is better, versus the number of evaluations across five antigens. Fine-tuned Qwen3.5-9B variants with and without reasoning augmentation are compared with their corresponding base proposers in the same acquisition-guided LDM loop. Shaded bands denote mean $\pm$ standard deviation across available seeds.

V4 Flash teacher variants, which reach 24.548 ± 3.589 with reasoning and 23.582 ± 2.832 without reasoning. The untuned Qwen3.5-9B base proposer reaches 16.489 ± 5.867. The result shows that the distilled policy transfers beyond the source task and that reasoning augmentation improves this transfer, although the intervals overlap.

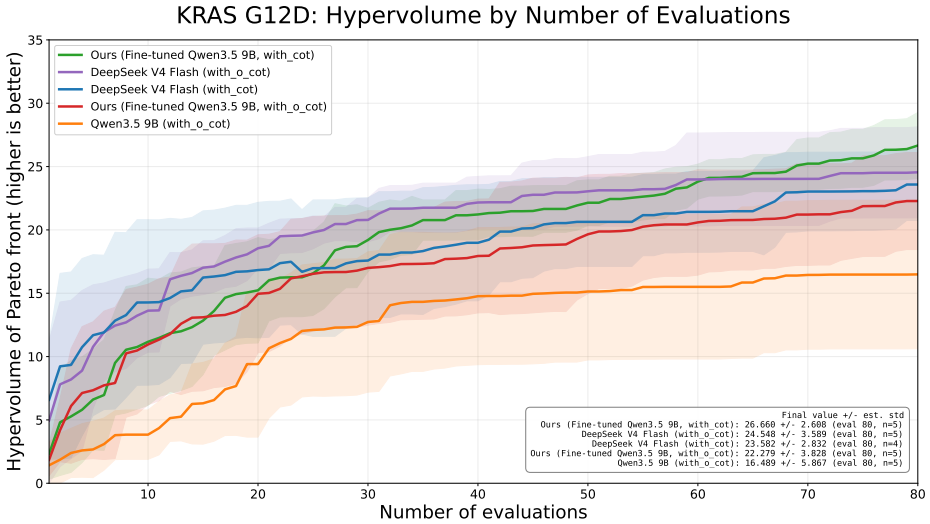


Figure 14: **Fine-tuning with reasoning-augmented discovery data on KRAS G12D.** Best-so-far Pareto-front hypervolume (higher is better) versus the number of expensive evaluations, comparing fine-tuned and base proposers inside the same LDM loop. Shaded bands denote one standard deviation.

Across programmes, biological sequences, and molecules, fine-tuning improves the proposal reservoir within the online LDM loop. The gains are strongest when the model is trained to represent not only the next action but also the surrogate-grounded rationale for why that action advances scientific search.

6.5 Ablation Studies

We next study how LDM depends on the test-time search budget and on its two temperature parameters, α (LLM sampling temperature) and η (acquisition tilt). Additional ablations, including the pure-LLM `autoresearch` loop, antibody hyperparameter sweeps, and single-step budget sweeps, are reported in Appendix C.

6.5.1 Test-time search scaling and acquisition functions

We first ask whether LDM exhibits the same test-time scaling behaviour that motivates inference-time search in language-model reasoning (Snell et al., 2025), but in the harder setting where the value is an expensive scientific objective approximated by a surrogate. We keep the outer five-minute training budget fixed and vary only the cheap inner-loop budget of test-time search. The inner budget has two knobs: N is the number of candidate `train.py` programmes the LLM proposes at each outer iteration, and H is the number of hypothesis/refinement branches each candidate is expanded into before the surrogate scores it; together they control the total acquisition-scored nodes per outer round, which we label NmHn so that N4H4 scores $4 \times 4 = 16$ nodes per round and N8H8 scores $8 \times 8 = 64$. We additionally vary whether the surrogate uses a fixed or dynamically expanded programme-feature representation and which acquisition is used as the value (posterior mean only, EI, or an uncertainty-aware value).

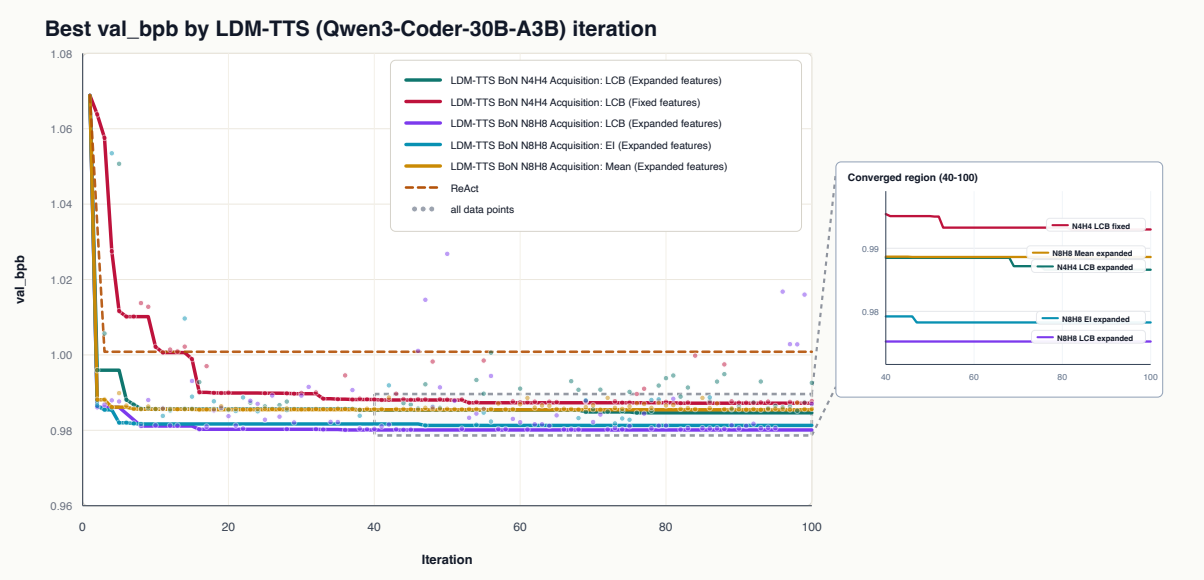


Figure 15: **Test-time search ablation on `autoresearch`.** Best validation `val_bpb` found up to iteration t (lower is better). Solid curves are LDM-TTS variants; the dashed curve is a ReAct-style no-acquisition reference (Yao et al., 2023b); gray points are all evaluated candidate programs. Each LDM-TTS variant is named NmHn for its inner budget, where N is the number of candidate programs the LLM proposes and H is the number of hypothesis/refinement branches each candidate is expanded into.

Figure 15 gives three conclusions. First, *increasing the test-time search budget helps.* The N8H8 expanded-feature curve reaches the best plateau, while the lower-budget N4H4 variants

converge higher. Since the outer training budget is unchanged, this improvement comes from spending more cheap computation before each expensive evaluation. Second, *dynamic feature expansion helps*, as `autoresearch` is not a fixed-dimensional hyperparameter problem. When the agent discovers a new mechanism, such as sparse memory or a new schedule family, the surrogate needs new coordinates on which to model it; fixed features compress those mechanisms into residual noise and produce weaker acquisition guidance. Third, *the acquisition itself matters*. Posterior-mean search is competitive early because it exploits the most promising known direction, but it plateaus above EI and the uncertainty-aware acquisition. EI improves on mean-only selection by rewarding improvement over the incumbent, while the uncertainty-aware value performs best because it deliberately allocates part of the test-time budget to under-modelled program families. These results validate the central claim that the LLM is the semantic search engine, yet the calibrated acquisition is the value that makes additional test-time compute useful.

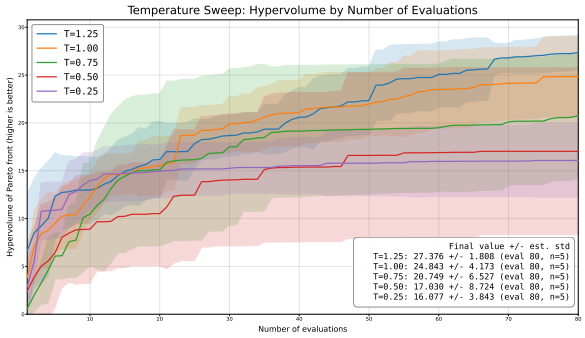
The same suite of test-time budget ablation sweeps was also conducted for the scientific design tasks detailed in §6.3 and §6.2, with full results presented in Appendix C.2.1 and C.2.2, respectively. For molecular design, a clear trend emerges: larger search budgets yield more effective discovery. This pattern is weaker for antibody design, where the scaling effect is small in magnitude even though the optimum remains target-dependent across antigens; see Appendix C.2.2 for the per-target curves. A likely explanation is that the underlying large language model provides only weak sequence-level priors in this domain. We further perform comparisons across different acquisition functions. Our results demonstrate that, given a sufficiently large search budget, well-chosen acquisition functions deliver consistent performance gains for both molecular and antibody design. Notably, the mean acquisition (a 0.5/0.5 scalarisation of the Vina and activity objectives) outperforms EHVI under low search budgets for molecular design. EHVI scores candidates against the current Pareto front, so it needs a large screening pool to be informative. Scalarised acquisitions do not, and therefore remain effective when the pool is small.

6.5.2 Hyperparameter sensitivity

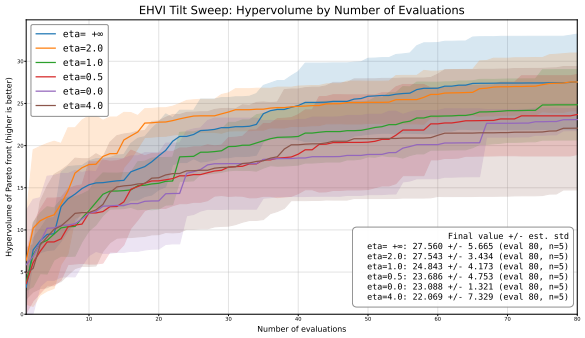
We next study the sensitivity of LDM to its two temperature parameters on the KRAS G12D molecular design task of §6.3. Here, the decoding parameter α, which is the LLM sampling temperature T_{LLM}, controls the diversity of LLM-generated SMILES candidates, while the acquisition temperature η controls the strength of acquisition-guided selection. As $\eta \to 0$ the policy collapses to the raw LLM reservoir and as $\eta \to \infty$ it reduces to acquisition-function argmax ($\eta = \infty$ in the experiments below).

For the LLM sampling temperature ablation, we fix $\eta = 1$ and vary the LLM-decoding temperature T_{LLM} in $\{0.25, 0.5, 0.75, 1, 1.25\}$. As shown in Figure 16a, increasing T_{LLM} generally improves the final optimization performance: the setting $T_{\text{LLM}} = 1.25$ achieves the highest final hypervolume, while lower temperatures degrade performance. For the acquisition temperature ablation, we fix $T_{\text{LLM}} = 1$ and vary $\eta \in \{0, 0.5, 1, \infty\}$. Figure 16b shows that increasing η generally improves the final hypervolume. Overall, on molecular design both sufficient LLM exploration and effective acquisition guidance are necessary for LDM, with the acquisition tilt η acting as the dominant driver of final Pareto-front expansion.

Antibody CDRH3 design (§6.2) exhibits a different sensitivity pattern. The open-source LLM provides relatively weak sequence-level priors for this domain, so directly generated candidates are often uninformative in the large combinatorial search space. Policy-based LDM mitigates this limitation by using the LLM to parameterise a search region and then generating a larger candidate pool within that region. As a result, performance is less sensitive to additional test-time reweighting through the sampling and acquisition temperatures. This is consistent with



(a) Effect of LLM sampling temperature T_{LLM} with fixed acquisition temperature $\eta = 1$.



(b) Effect of acquisition temperature η with fixed LLM sampling temperature $T_{\text{LLM}} = 1$.

Figure 16: **Hyperparameter ablations of LDM-TTS on molecular design.** Each curve reports the hypervolume of the Pareto front over the evaluation trajectory. Higher values indicate better multi-objective optimization performance.

the results in §6.2, where policy-mode LDM outperforms direct generation primarily through structured search-space parameterisation rather than strong sequence-level priors or aggressive acquisition weighting. Full temperature ablations for antibody design are provided in Appendix C.3.

6.6 Overall Discussion

Across the three domains, the experiments highlight several recurring properties of LDM.

First, combining calibrated Bayesian surrogate acquisition with LLM-driven inference-time search consistently outperforms standalone LLMs and vanilla Bayesian optimisation. LLMs supply native semantic generation over complex discrete spaces where BO cannot easily propose valid candidates, while Gaussian process surrogates deliver reliable uncertainty-aware value signals missing from uncalibrated LLM self-scoring. Neither component alone reproduces the full performance gains observed in the integrated LDM pipeline.

Second, the optimal LLM operating mode is domain-dependent, governed by the strength of the model's pre-trained domain priors and the structure of the design space. For sparsely covered sequence spaces such as antibody CDRH3, policy-mode LLM search expands exploration coverage by defining local search zones. For domains in which the LLM already produces useful candidates, such as code and SMILES, large-batch direct generation achieves superior Pareto/affinity improvements. The unified tilted distribution formulation accommodates both paradigms without modification to the underlying acquisition-guided optimisation logic.

Third, LDM can revise the proposal space when local search stops improving. This differs from standard BO settings in which the optimisation domain is fixed in advance. In our experiments, such search-space updates are often followed by further improvements after local plateaus.

Fourth, the ablation and fine-tuning studies in §6.5 and §6.4 clarify why the framework works. Test-time search helps only when the additional inference-time compute is filtered by a calibrated, uncertainty-aware acquisition rather than by language plausibility, and the surrogate must track the search space as it evolves. When the LLM reservoir is itself weakly informative, as in antibody CDRH3 design, acquisition reweighting adds little beyond the policy's structural search. The fine-tuning results further show that the tilted search policy can be distilled into a single lightweight proposer, and that grounding distilled actions in surrogate value improves both same-domain and cross-task transfer. Together, these findings indicate that the search policy itself is learnable and reusable.

Finally, these experiments support the view of scientific design as a sequential decision problem rather than only a prediction problem. LDM uses the LLM to construct and adapt candidate proposals, while the surrogate guides which candidates should be evaluated under a limited experimental budget. Important limitations remain, including the cubic scaling of exact Gaussian Processes with the number of observations and the possibility of surrogate misspecification or reward exploitation. Extending the framework to larger experimental budgets and more reliable surrogate models remains an important direction for future work.

7 Related Work

This section situates the LDMs within the broader literature on generative models, statistical optimisation, and their intersection for scientific discovery. The aim is not to provide an exhaustive survey, but a focused positioning of LDM within the key strands of related work that directly motivate its design and delineate its contributions. We structure the discussion in three parts. We first review the remarkable progress in LLMs, including their scaling laws, emergent reasoning capabilities, and post-training methods; we then highlight the limitations of pure generative models for discovery, particularly the lack of calibrated uncertainty and expensive, sparse reward signals. Next, we survey the statistical learning paradigm for scientific discovery, from multi-armed bandits to Bayesian optimisation (BO), and discuss its strengths and challenges in high-dimensional or open-ended, unrepresented scientific spaces. Finally, we examine the growing body of work on hybrid approaches that combine LLMs with BO, clarify the differences between LDM and the most closely related methods, and articulate the open problems our framework addresses.

7.1 Large Language Models and Their Limitations in Scientific Discovery

The early development of language models was marked by several landmark architectures, including the Transformer (Vaswani et al., 2017), GPT-1 (Radford et al., 2018), and BERT (Devlin et al., 2019). The past few years have witnessed transformative progress in large language models. Scaling laws (Kaplan et al., 2020) have established a predictable correlation in LLMs' performance with the scale of model weights, data, and test-time compute. The remarkable success of GPT-3 (Brown et al., 2020) further propelled research along the scaling direction. As models scale, they exhibit emergent abilities (Wei et al., 2022) not present in smaller counterparts, including complex reasoning via chain-of-thought prompting (Wei et al., 2023; Kojima et al., 2023), in-context learning (Han et al., 2025), and instruction following (Ouyang et al., 2022). In addition, post-training methods, such as supervised fine-tuning (SFT) and reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022), can better adapt LLMs to downstream tasks and the users' preferences.

Since 2023, *Test-time compute* has been a crucial development that pushes LLMs from language understanding to reasoners for complex problems, where the key insight is that the use of additional computation at inference time greatly improves output quality. Early research utilises verifiers (Cobbe et al., 2021; Lightman et al., 2023) to guide reasoning to solve complex mathematical problems. Further, *tree of thoughts* (Yao et al., 2023a) performs deliberate search over intermediate reasoning states with self-evaluation. Self-Refine (Madaan et al., 2023) iterates generate–critique–revise cycles. More generally, Snell et al. (2025) show that optimal allocation of test-time compute allows small models to outperform much larger models. The underlying search machinery, including UCT (Kocsis and Szepesvári, 2006) and the PUCT rule popularised by AlphaZero (Silver et al., 2017), has been applied to LLM decoding and training (Feng et al., 2024). The combination of inference-time search with verifier-guided refinement underpins renowned systems such as OpenAI's o1 (OpenAI, 2024) and DeepSeek-R1 (DeepSeek-

AI, 2025). When cheap supervision is repeatedly available, *reinforcement learning with verifiable rewards* (RLVR) (DeepSeek-AI, 2025; Shao et al., 2024) has led to impressive results in mathematics, coding, and reasoning tasks. A unified tutorial perspective on these methods is provided by Wang (2025), and OpenR (Wang et al., 2024) further establishes a comprehensive technical framework.

Despite the significant progress of LLM reasoning, how LLMs engage in open-ended scientific discovery remains questionable. Studies show that LLMs are also poorly calibrated for scientific decisions—their internal confidence does not reliably reflect uncertainty about an external property (Gupta et al., 2025; Kristiadi et al., 2024). Moreover, end-to-end execution benchmarks consistently show that frontier LLM agents cannot yet reliably discover novel, high-quality solutions: on ResearchClawBench the strongest autonomous agent scores only 21.5/50 across 40 real-paper-derived tasks (Xu et al., 2026), and on PaperBench agents replicate ICML papers at 21.0% against 41.4% by PhD researchers (Starace et al., 2025). Even at the ideation stage, novelty rankings that initially favour LLMs (Si et al., 2024) reverse once ideas are executed (Si et al., 2025). LLM-generated ideas also exhibit narrow exploration, converging on the centroid of existing literature (Tang and Yang, 2026).

Therefore, scientific discovery essentially requires a novel paradigm of foundation models. Crucially, the key challenge is not intrinsic LLM failures: when proposals are paired with automated evaluators and search, genuinely novel results are recovered (Novikov et al., 2025). The bottleneck is the absence of a calibrated external value signal. Test-time compute and reinforcement learning typically rely on cheap verifiers or reward models, such as ground-truth answers or trained networks from large data, to guide the search of reasoning steps. This is built upon access to rich feedback data or ground-truth answers, which, however, is usually scarce in scientific domains.

7.2 Statistical Learning for Scientific Discovery

A mature body of statistical learning approaches addresses the problem of optimising a black-box reward function with noisy evaluations. *Multi-arm bandit* constitutes a representative framework conceptualising the exploration–exploitation trade-off (Lattimore and Szepesvári, 2020), where the upper confidence bound (UCB) (Auer, 2002) provides a simple way to balance these competing objectives with sublinear regret guarantees. However, classical multi-arm bandit settings assume a finite set of arms, yet scientific design spaces are often effectively unbounded.

This challenge of unbounded space is then captured by the *infinitely-armed bandit* (Berry et al., 1997; Wang et al., 2008) frameworks, where new arms must be discovered from a *reservoir*. The performance in such settings is governed by the reservoir's tail of near-optimal arms, a quantity known as the *discovery gap*. The LDM framework explicitly adopts this view, treating the LLM as an adaptive reservoir and analysing the discovery gap through a missing-mass argument (McAllester and Schapire, 2000; Berend and Kontorovich, 2013) in § 5. In comparison, LDM utilises an LLM as its reservoir, endowing the search process with broad scientific priors and rich semantic knowledge to inform candidate generation. In addition, the LLM's context $\mathcal{C}_t$ encapsulates the full evaluation history, enabling in-context learning to synthesise insights from prior trials and propose progressively more effective candidate designs. The regret analysis in § 5 decomposes the cost of search into a surrogate-estimation term and a reservoir-quality term, with the latter shrinking as more inference-time compute is spent.

Scientific discovery tasks are further constrained by costly evaluation procedures and limited computational budget. This demand motivates *model-based* bandit variants that explicitly extract knowledge from evaluation history and make the most of such knowledge in decision making. To this end, *Bayesian optimisation* (BO) (Mockus, 1975; Jones et al., 1998; Garnett, 2023) provides a powerful framework for such a variant that uses a probabilistic surrogate

following the rigour of Bayesian inference, where typically a Gaussian process (GP) (Rasmussen and Williams, 2006) serves as the surrogate. The GP provides predictive mean and variance, which are combined into an *acquisition function* that guides the selection of the next evaluation. For example, canonical implementations such as Efficient Global Optimisation (EGO) (Jones et al., 1998) and GP-UCB (Srinivas et al., 2010) use expected improvement (EI) and UCB (Auer, 2002) as their acquisition function, respectively. Theoretical analysis shows that GP-UCB enjoys sublinear cumulative regret in terms of the maximum information gain, a measure of how informative the observations are about the objective (Srinivas et al., 2010). These theoretical guarantees make BO a principled tool for experimental design for scientific discovery (Yu et al., 2026).

BO has been successfully applied across chemistry, materials science, and biology, often requiring domain-specific adaptations. Examples include the PHOENICS (Häse et al., 2018) and Gryffin (Häse et al., 2021) frameworks for chemical discovery. Combinatorial BO frameworks such as AntBO (Khan et al., 2022) have been developed to handle the discrete sequence space of CDRH3 loops for antibody design. Benchmarking studies across multiple materials domains (Liang et al., 2021) have compared different surrogate models and acquisition functions, while applications in chemical synthesis (Shields et al., 2021) and material discovery (Shoyeb Raihan et al., 2024) further demonstrate the versatility of BO. For a comprehensive overview of BO for chemical problems, see (Wu et al., 2024). To search among discrete sequences (e.g., SMILES and proteins), methods such as BOSS (Bayesian Optimisation over String Spaces) (Moss et al., 2020) use string kernels to directly construct GPs over the string space. To handle molecule design, latent-space approaches combine variational autoencoders with BO, though they struggle with validity and out-of-distribution generation (Griffiths and Hernández-Lobato, 2020). More recent frameworks like COMBO (Dreczkowski et al., 2023) address combinatorial and mixed-variable optimisation. Multi-objective extensions (Expected Hypervolume Improvement) (Ponweiser et al., 2008; Daulton et al., 2021) handle the competing objectives common in drug discovery and materials science.

Despite these advances, pure statistical methods face fundamental challenges. First, they struggle to propose valid candidates in vast, unbounded spaces of practical scientific problems. Next, they cannot easily incorporate high-level semantic domain knowledge. Additionally, they are often unresponsive to user-specified design constraints or preferences. These limitations create a natural opportunity for complementarity with generative models, which can propose semantically meaningful candidates and be steered by domain knowledge. This brings us to the growing body of hybrid approaches that combine the strengths of LLMs and BO.

7.3 Hybrid Approaches: LLM-Guided Discovery

The recognition that LLMs and statistical learning offer complementary strengths has led to a growing body of hybrid methods. A high-level approach is OPRO (Yang et al., 2024), which uses an LLM as an optimizer by describing the task in natural language and iteratively generating solutions from a prompt containing a history of solution–score pairs. While powerful in prompt optimisation, OPRO lacks the calibrated uncertainty and principled exploration of a Bayesian surrogate.

Several works have integrated LLMs directly into the BO pipeline. LLAMBO (Liu et al., 2024) uses LLMs for zero-shot warm-starting, surrogate modelling, and candidate sampling, showing gains in low-data hyperparameter optimisation regimes. BoChemian (Rankovic and Schwaller, 2023) and its extension (Ranković et al., 2025) use LLM embeddings as features for BO over chemical reactions, demonstrating that fine-tuning LLMs with uncertainty-aware objectives can improve discovery rates. LABO (Chen et al., 2026) proposes a gating criterion to dynamically balance LLM predictions against experimental observations. Other works use

LLMs to generate or refine kernels for GP surrogates (Suwandi et al., 2025), handle natural language feedback (Kobalczyk et al., 2025), or perform analog circuit design (Yin et al., 2024; Chen et al., 2024). The LLM-as-warm-start paradigm, where the LLM suggests the initial candidates or a search region, is also explored in (Ramos et al., 2025; Yin et al., 2024). In catalysis, Ramos et al. (2025) use frozen LLMs for in-context learning to enable BO without feature engineering, demonstrating the potential of LLMs to operate directly in language space.

Despite various attempts to combine the advantages of LLMs and BO, some recent studies have challenged their solidity. A sobering perspective is provided by Gupta et al. (2025), who find that current LLMs show no sensitivity to experimental feedback in BO tasks, and classical methods such as linear bandits and GP optimisation consistently outperform LLM agents. Similarly, Kristiadi et al. (2024) take a dispassionate stance, concluding that uncertainty taken directly from point-estimated LLMs is unreliable, and that LLMs are useful for BO over molecules primarily only when they serve as fixed feature extractors for principled GP surrogates and are pretrained or finetuned on domain-specific data. These critiques directly motivate LDM's design: we retain an explicit calibrated posterior (the GP surrogate) and use the LLM only as a candidate reservoir.

Another closely related work to LDM is dLLM (Yuan et al., 2026), which for *offline* black-box optimisation runs a masked-diffusion tree search where leaf candidates are scored by expected improvement under a GP fitted to an offline dataset. In contrast, LDM targets the standard *online* recurrent loop with sequential, expensive evaluations. The core is that LDM must raise designs that are valuable not only for the current trial but also for gaining knowledge for the future.

In summary, LLMs provide powerful generative priors but lack calibration and robust optimisation; meanwhile, statistical methods provide principled, uncertainty-aware search but struggle to propose candidates in semantic, combinatorial spaces. LDM bridges this gap by making the acquisition function the value signal of an LLM-driven inference-time search, creating a principled, model-based framework for scientific discovery.

8 Conclusion

The core challenge of scientific discovery lies in searching for optimal solutions within vast, structured and open-ended design spaces under a limited budget of costly evaluations. Existing paradigms exhibit complementary limitations: large language models (LLMs) possess strong structured generative capabilities and domain priors, but cannot reliably estimate the true value of external objectives or their associated uncertainty. Bayesian optimisation, by contrast, delivers uncertainty-aware experimental decisions from sparse observations, yet struggles to autonomously propose valid candidates in complex discrete spaces. This work presents the Large Discovery Model (LDM), an experiment-grounded recurrent architecture that deeply couples the generative prior of LLMs with the value signal from a Gaussian process surrogate. Framing scientific design as a sequential inverse decision problem under an evaluation budget, LDM operates a closed loop of generation, evaluation and update that simultaneously expands the search frontier and allocates experimental resources precisely.

At the theoretical and algorithmic level, LDM establishes a KL-regularised, acquisition-guided optimisation framework and derives a closed-form acquisition-tilted optimal search distribution. This formulation extends the classical dichotomy between exploitation and exploration into a tripartite scientific search regime of exploitation, exploration and discovery, unifying the generative prior and empirical value signal within a single mathematical framework. We further provide a complete regret decomposition that explicitly delineates the performance boundaries of generative coverage gap, surrogate fitting error and sampling strategy suboptimality. Al-

gorithmically, the framework supports two candidate generation paradigms, direct generation and indirect parameterisation, and is complemented by a decomposed feedback mechanism to guide targeted iterative refinement, making it adaptable to design spaces of varying structure. Building on this core, we introduce a post-training amortisation scheme: through reasoning-augmented supervised fine-tuning, the high-budget test-time search policy is amortised into the model weights, substantially reducing inference-time computational cost while preserving search quality.

Empirical evaluation across three heterogeneous domains, including the auto-research scenario of neural-network training programme, antibody CDRH3 sequence design, and multi-objective molecular optimisation, systematically validates the generality of the LDM framework and its core findings. First, the structured generative capacity of LLMs and the uncertainty calibration of Bayesian surrogates exhibit strong complementarity, with their integrated performance consistently outperforming both LLM-only and BO-only baselines. Second, the discovery mechanism is central to breaking performance plateaus: purely local optimisation converges rapidly to local optima, whereas expansion of the search frontier continuously lifts the attainable performance ceiling. Third, both test-time compute scaling and post-training fine-tuning reliably improve search efficiency, and the optimal generation paradigm correlates with the strength of domain priors. Fine-tuning experiments further demonstrate stable gains across cross-target and cross-task transfer, indicating that the model learns generalisable discovery decision logic rather than task-specific memorisation.

Nevertheless, the current framework is subject to several constraints and limitations. The Gaussian process surrogate incurs cubic computational complexity in the number of observations, limiting its scalability under large experimental budgets. Test-time search relies on batch candidate sampling, which entails substantial LLM inference overhead. The kernel functions and feature representations of the surrogate still require manual customisation per domain, resulting in limited automation for cross-domain transfer. Additionally, framework performance is bounded by the coverage of the LLM's pretrained domain knowledge; in domains with weak priors, it is difficult to achieve substantial superiority over specialised methods. Finally, the current closed loop is updated solely based on in-process experimental observations, and has not yet systematically incorporated external existing knowledge and data, leaving a large body of unknown knowns unutilised in the search process.

For future work, a core direction is to address the utilisation of unknown knowns. We will introduce memory retrieval mechanisms and federated discovery frameworks to incorporate external knowledge bases, published experimental results and data from parallel exploration processes into the search closed loop. Combined with techniques such as sparse Gaussian processes (Snelson and Ghahramani, 2005; Titsias, 2009), this will simultaneously resolve the dual challenges of limited context capacity and high computational complexity as observational data scales. Building on this, we will explore end-to-end surrogate specification, in which the LLM autonomously learns kernel functions and prior settings for the surrogate, eliminating manual adaptation across domains. We will also advance discovery-oriented post-training paradigms, exploring more sophisticated training paths such as preference learning and reinforcement learning to further internalise the acquisition-tilted decision logic into the model's generative distribution. This would endow the model with native capability to balance exploitation, exploration and discovery, ultimately enabling more efficient open-ended scientific discovery.

References

Peter Auer. Using Confidence Bounds for Exploitation-Exploration Trade-offs. *Journal of Machine Learning Research*, 3(Nov):397–422, 2002. ISSN 1533-7928.

Maximilian Balandat, Brian Karrer, Daniel R. Jiang, Samuel Daulton, Benjamin Letham, Andrew Gordon Wilson, and Eytan Bakshy. BoTorch: A framework for efficient Monte-Carlo Bayesian optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020.

Joeran Beel, Min-Yen Kan, and Moritz Baumgart. Evaluating sakana's AI scientist: Bold claims, mixed results, and a promising future? *ACM SIGIR Forum*, 59(1):1–20, 2025. doi: 10.1145/3769733.3769747. URL https://doi.org/10.1145/3769733.3769747.

Daniel Berend and Aryeh Kontorovich. On the concentration of the missing mass. *Electronic Communications in Probability*, 18:1–7, 2013.

Donald A. Berry, Robert W. Chen, Alan Zame, David C. Heath, and Larry A. Shepp. Bandit problems with infinitely many arms. *The Annals of Statistics*, 25(5):2103–2116, 1997.

Pier Giovanni Bissiri, Chris C. Holmes, and Stephen G. Walker. A general framework for updating belief distributions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(5):1103–1130, 2016. doi: 10.1111/rssb.12158.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners, July 2020.

Chih-Yu Chang, Milad Azvar, Chinedum Okwudire, and Raed Al Kontar. Llinbo: Trustworthy llm-in-the-loop bayesian optimization. *arXiv preprint arXiv:2505.14756*, 2025.

Guojin Chen, Keren Zhu, Seunggeun Kim, Hanqing Zhu, Yao Lai, Bei Yu, and David Z. Pan. LLM-Enhanced Bayesian Optimization for Efficient Analog Layout Constraint Generation, December 2024.

Zhuo Chen, Xinzhe Yuan, Jianshu Zhang, Jinzong Dong, Ruichen Zhou, Yingchun Niu, Tianhang Zhou, Yu Yang Fredrik Liu, Yuqiang Li, Nanyang Ye, and Qinying Gu. LABO: LLM-Accelerated Bayesian Optimization through Broad Exploration and Selective Experimentation, May 2026.

Sayak Ray Chowdhury and Aditya Gopalan. On kernelized multi-armed bandits. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, volume 70 of *PMLR*, pages 844–853, 2017.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training Verifiers to Solve Math Word Problems, November 2021.

Samuel Daulton, Maximilian Balandat, and Eytan Bakshy. Parallel Bayesian optimization of multiple noisy objectives with expected hypervolume improvement. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, 2021.

DeepSeek-AI. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning, January 2025.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, May 2019.

Monroe D. Donsker and S. R. Srinivasa Varadhan. On a variational formula for the principal eigenvalue for operators with maximum principle. *Proceedings of the National Academy of Sciences*, 72(3):780–783, 1975. doi: 10.1073/pnas.72.3.780.

Kamil Dreczkowski, Antoine Grosnit, and Haitham Bou-Ammar. Framework and Benchmarks for Combinatorial and Mixed-variable Bayesian Optimization. In *37th Conference on Neural Information Processing Systems (NeurIPS 2023) Track on Datasets and Benchmarks*, 2023.

Xidong Feng, Ziyu Wan, Muning Wen, Stephen Marcus McAleer, Ying Wen, Weinan Zhang, and Jun Wang. Alphazero-like Tree-Search can Guide Large Language Model Decoding and Training, February 2024.

Peter I. Frazier. A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

Roman Garnett. *Bayesian Optimization*. Cambridge University Press, 2023.

Ryan-Rhys Griffiths and José Miguel Hernández-Lobato. Constrained Bayesian optimization for automatic chemical design using variational autoencoders. *Chemical Science*, 11(2):577–586, 2020. ISSN 2041-6520. doi: 10.1039/c9sc04026a.

Rushil Gupta, Jason Hartford, and Bang Liu. LLMs for Bayesian Optimization in Scientific Domains: Are We There Yet?, September 2025.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1861–1870. PMLR, 2018. URL https://proceedings.mlr.press/v80/haarnoja18b.html.

Chi Han, Ziqi Wang, Han Zhao, and Heng Ji. Understanding Emergent In-Context Learning from a Kernel Regression Perspective, September 2025.

Florian Häse, Loïc M. Roch, Christoph Kreisbeck, and Alán Aspuru-Guzik. Phoenics: A Bayesian Optimizer for Chemistry. *ACS Central Science*, 4(9):1134–1145, September 2018. ISSN 2374-7943, 2374-7951. doi: 10.1021/acscentsci.8b00307.

Florian Häse, Matteo Aldeghi, Riley J. Hickman, Loïc M. Roch, and Alán Aspuru-Guzik. Gryffin: An algorithm for Bayesian optimization of categorical variables informed by expert knowledge. *Applied Physics Reviews*, 8(3):031406, September 2021. ISSN 1931-9401. doi: 10.1063/5.0048164.

Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient Global Optimization of Expensive Black-Box Functions. *Journal of Global Optimization*, 13(4):455–492, December 1998. ISSN 1573-2916. doi: 10.1023/A:1008306431147.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling Laws for Neural Language Models, January 2020.

Andrej Karpathy. autoresearch: AI agents running research on single-GPU nanochat training automatically. https://github.com/karpathy/autoresearch, 2026. GitHub repository.

Asif Khan, Alexander I. Cowen-Rivers, Antoine Grosnit, Derrick-Goh-Xin Deik, Philippe A. Robert, Victor Greiff, Eva Smorodina, Puneet Rawat, Kamil Dreczkowski, Rahmad Akbar, Rasul Tutunov, Dany Bou-Ammar, Jun Wang, Amos Storkey, and Haitham Bou-Ammar. AntBO: Towards real-world automated antibody design with combinatorial Bayesian optimisation. *arXiv preprint arXiv:2201.12570*, 2022.

Katarzyna Kobalczyk, Zhiyuan Jerry Lin, Benjamin Letham, Zhuokai Zhao, Maximilian Balandat, and Eytan Bakshy. LILO: Bayesian Optimization with Interactive Natural Language Feedback, October 2025.

Levente Kocsis and Csaba Szepesvári. Bandit based Monte-Carlo planning. In *European Conference on Machine Learning (ECML)*, pages 282–293, 2006.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large Language Models are Zero-Shot Reasoners, January 2023.

Agustinus Kristiadi, Felix Strieth-Kalthoff, Marta Skreta, Pascal Poupart, Alán Aspuru-Guzik, and Geoff Pleiss. A Sober Look at LLMs for Material Discovery: Are They Actually Good for Bayesian Optimization Over Molecules?, 2024.

Mikael Krogerus and Roman Tschäppeler. *The Decision Book: Fifty Models for Strategic Thinking*. W. W. Norton & Company, New York, revised edition, 2018. ISBN 9780393652376.

Varun Kumar, Abhinandan K. Danodia, Pradeep S. Jadhavar, Subhendu K. Mohanty, and Swapan K. Samanta. An overview of KRAS G12D inhibitors: Expanding the therapeutic frontier of KRAS in targeting KRAS G12D using diverse therapeutic modalities. *European Journal of Medicinal Chemistry*, 305:118555, March 2026. ISSN 0223-5234. doi: 10.1016/j.ejmech.2025.118555.

Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press, 1 edition, July 2020. ISBN 978-1-108-57140-1 978-1-108-48682-8. doi: 10.1017/9781108571401.

Sergey Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*, 2018.

Qiaohao Liang, Aldair E. Gongora, Zekun Ren, Armi Tiihonen, Zhe Liu, Shijing Sun, James R. Deneault, Daniil Bash, Flore Mekki-Berrada, Saif A. Khan, Kedar Hippalgaonkar, Benji Maruyama, Keith A. Brown, John Fisher Iii, and Tonio Buonassisi. Benchmarking the performance of Bayesian optimization across multiple experimental materials science domains. *npj Computational Materials*, 7(1):188, November 2021. ISSN 2057-3960. doi: 10.1038/s41524-021-00656-9.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's Verify Step by Step, May 2023.

Tennison Liu, Nicolás Astorga, Nabeel Seedat, and Mihaela van der Schaar. Large Language Models to Enhance Bayesian Optimization, March 2024.

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

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023.

David A. McAllester and Robert E. Schapire. On the convergence rate of Good–Turing estimators. In *Proceedings of the 13th Annual Conference on Computational Learning Theory (COLT)*, pages 1–6, 2000.

J. Mockus. On the Bayes Methods for Seeking the Extremal Point. *IFAC Proceedings Volumes*, 8(1):428–431, August 1975. ISSN 14746670. doi: 10.1016/S1474-6670(17)67769-3.

Henry Moss, David Leslie, Daniel Beck, Javier González, and Paul Rayson. BOSS: Bayesian Optimization over String Spaces. In *Advances in Neural Information Processing Systems*, volume 33, pages 15476–15486. Curran Associates, Inc., 2020.

Alexander Novikov, Ngân Vũ, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt Wagner, Sergey Shirobokov, Borislav Kozlovskii, Francisco J. R. Ruiz, Abbas Mehrabian, M. Pawan Kumar, Abigail See, Swarat Chaudhuri, George Holland, Alex Davies, Sebastian Nowozin, Pushmeet Kohli, and Matej Balog. Alphaevolve: A coding agent for scientific and algorithmic discovery. *arXiv preprint arXiv:2506.13131*, 2025.

OpenAI. OpenAI o1 System Card, December 2024.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*, March 2022.

Jan Peters, Katharina Mülling, and Yasemin Altun. Relative entropy policy search. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, pages 1607–1612. AAAI Press, 2010. doi: 10.1609/aaai.v24i1.7727.

Wolfgang Ponweiser, Tobias Wagner, Dirk Biermann, and Markus Vincze. Multiobjective Optimization on a Limited Budget of Evaluations Using Model-Assisted $\mathcal{S}$-Metric Selection. In Günter Rudolph, Thomas Jansen, Nicola Beume, Simon Lucas, and Carlo Poloni, editors, *Parallel Problem Solving from Nature – PPSN X*, pages 784–794, Berlin, Heidelberg, 2008. Springer. ISBN 978-3-540-87700-4. doi: 10.1007/978-3-540-87700-4_78.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training, 2018.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

Mayk Caldas Ramos, Shane S. Michtavy, Marc D. Porosoff, and Andrew D. White. Bayesian Optimization of Catalysis With In-Context Learning, May 2025.

Bojana Rankovic and Philippe Schwaller. BoChemian: Large Language Model Embeddings for Bayesian Optimization of Chemical Reactions. In *NeurIPS 2023 Workshop on Adaptive Experimental Design and Active Learning in the Real World*, 2023.

Bojana Ranković, Ryan-Rhys Griffiths, and Philippe Schwaller. Large language models as uncertainty-calibrated optimizers for experimental discovery, 2025.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams, and Nando de Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104 (1):148–175, 2016.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models, April 2024.

Benjamin J. Shields, Jason Stevens, Jun Li, Marvin Parasram, Farhan Damani, Jesus I. Martinez Alvarado, Jacob M. Janey, Ryan P. Adams, and Abigail G. Doyle. Bayesian reaction optimization as a tool for chemical synthesis. *Nature*, 590(7844):89–96, February 2021. ISSN 0028-0836, 1476-4687. doi: 10.1038/s41586-021-03213-y.

Ahmed Shoyeb Raihan, Hamed Khosravi, Srinjoy Das, and Imtiaz Ahmed. Accelerating material discovery with a threshold-driven hybrid acquisition policy-based Bayesian optimization. *Manufacturing Letters*, 41:1300–1311, October 2024. ISSN 2213-8463. doi: 10.1016/j.mfglet.2024.09.157.

Chenglei Si, Diyi Yang, and Tatsunori Hashimoto. Can LLMs generate novel research ideas? a large-scale human study with 100+ NLP researchers, 2024. URL https://arxiv.org/abs/2409.04109.

Chenglei Si, Tatsunori Hashimoto, and Diyi Yang. The ideation–execution gap: Execution outcomes of LLM-generated versus human research ideas, 2025. URL https://arxiv.org/abs/2506.20803.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*, 2017.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2408.03314.

Edward Snelson and Zoubin Ghahramani. Sparse Gaussian Processes using Pseudo-inputs. In *Advances in Neural Information Processing Systems*, volume 18. MIT Press, 2005.

Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, pages 1015–1022, 2010.

Giulio Starace, Oliver Jaffe, Dane Sherburn, James Aung, Jun Shern Chan, Leon Maksin, Rachel Dias, Evan Mays, Benjamin Kinsella, Wyatt Thompson, Johannes Heidecke, Amelia Glaese, and Tejal Patwardhan. PaperBench: Evaluating AI's ability to replicate AI research, 2025. URL https://arxiv.org/abs/2504.01848.

Richard Cornelius Suwandi, Feng Yin, Juntao Wang, Renjie Li, Tsung-Hui Chang, and Sergios Theodoridis. Adaptive Kernel Design for Bayesian Optimization Is a Piece of CAKE with LLMs, September 2025.

Yixuan Tang and Yi Yang. AI research agents narrow scientific exploration, 2026. URL https://arxiv.org/abs/2605.27905.

Michalis Titsias. Variational Learning of Inducing Variables in Sparse Gaussian Processes. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, pages 567–574. PMLR, April 2009.

Oleg Trott and Arthur J. Olson. Autodock vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of Computational Chemistry*, 31(2):455–461, 2010. doi: https://doi.org/10.1002/jcc.21334. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/jcc.21334.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, Long Beach, CA, USA, 2017.

Jun Wang. A Tutorial on LLM Reasoning: Relevant Methods behind ChatGPT o1, February 2025.

Jun Wang, Meng Fang, Ziyu Wan, Muning Wen, Jiachen Zhu, Anjie Liu, Ziqin Gong, Yan Song, Lei Chen, Lionel M Ni, et al. Openr: An open source framework for advanced reasoning with large language models. *arXiv preprint arXiv:2410.09671*, 2024.

Yizao Wang, Jean-Yves Audibert, and Rémi Munos. Algorithms for infinitely many-armed bandits. In *Advances in Neural Information Processing Systems (NeurIPS) 21*, pages 1729–1736, 2008.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent Abilities of Large Language Models, October 2022.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models, January 2023.

Yifan Wu, Aron Walsh, and Alex M. Ganose. Race to the bottom: Bayesian optimisation for chemical problems. *Digital Discovery*, 3(6):1086–1100, 2024. ISSN 2635-098X. doi: 10.1039/D3DD00234A.

Wanghan Xu, Shuo Li, Tianlin Ye, Qinglong Cao, Yixin Chen, Hengjian Gao, Yiheng Wang, Qi Li, Kun Li, Sheng Xu, Shengdu Chai, Fangchen Yu, Xiangyu Zhao, Zhangrui Zhao, Weijie Ma, Zijie Guo, Koutian Wu, Haoyu Zhou, Haoxiang Yin, Lixue Cheng, Chaofan Hu, Haoxuan Li, Lu Mi, Xuxuan Xie, Yifan Zhou, Ruizhe Chen, Zhiwang Zhou, Xingjian Guo, Yuhao Zhou, Xuming He, Shengyuan Xu, Xinyu Gu, Jiamin Wu, Mianxin Liu, Chunfeng Song, Fenghua Ling, Dongzhan Zhou, Shixiang Tang, Yuqiang Li, Mao Su, Peng Ye, Siqi Sun, Bin Wang, Xue Yang, Zhenfei Yin, Tianfan Fu, Guangtao Zhai, Wanli Ouyang, Bo Zhang, Lei Bai, and Wenlong Zhang. ResearchClawBench: A benchmark for end-to-end autonomous scientific research, 2026. URL https://arxiv.org/abs/2606.07591.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. Large Language Models as Optimizers, April 2024.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023a.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023b. URL https://arxiv.org/abs/2210.03629.

Yuxuan Yin, Yu Wang, Boxun Xu, and Peng Li. ADO-LLM: Analog Design Bayesian Optimization with In-Context Learning of Large Language Models. In *Proceedings of the 43rd IEEE/ACM International Conference on Computer-Aided Design*, pages 1–9, Newark Liberty International Airport Marriott New York NY USA, October 2024. ACM. ISBN 979-8-4007-1077-3. doi: 10.1145/3676536.3676816.

Zhongwei Yu, Rasul Tutunov, Alexandre Max Maraval, Zikai Xie, Zhenzhi Tan, Jiankang Wang, Bin Cao, Zijing Li, Liangliang Xu, Qi Yang, Jun Jiang, Sanzhong Luo, Zhenxiao Guo, Tongyi Zhang, Haitham Bou-Ammar, and Jun Wang. Efficient and Principled Scientific Discovery through Bayesian Optimization: A Tutorial, April 2026.

Ye Yuan, Can Chen, Zipeng Sun, Dinghuai Zhang, Christopher Pal, and Xue Liu. Diffusion large language models for black-box optimization. *arXiv preprint arXiv:2601.14446*, 2026.

A Detailed Proofs

We give the full proof of Proposition 1, Lemma 1 and Theorem 1.

A.1 Proof of Proposition 1

Proof. Write $J(q) = \mathbb{E}_{x\sim q}[a_t] - \frac{1}{\eta}\mathrm{KL}(q\|p_{\theta,\alpha})$. For any q,

$$J(q) = \sum_x q(x)a_t(x) - \tfrac{1}{\eta}\sum_x q(x)\log\frac{q(x)}{p_{\theta,\alpha}(x)} = -\tfrac{1}{\eta}\sum_x q(x)\log\frac{q(x)}{p_{\theta,\alpha}(x)e^{\eta a_t(x)}}.$$

Define $\pi_t(x) \propto p_{\theta,\alpha}(x)e^{\eta a_t(x)}$ with normaliser Z_t. Substituting,

$$J(q) = -\tfrac{1}{\eta}\,\mathrm{KL}(q\|\pi_t) + \tfrac{1}{\eta}\log Z_t \;\le\; \tfrac{1}{\eta}\log Z_t,$$

with equality iff $q = \pi_t$, since $\mathrm{KL}(q\|\pi_t) \ge 0$ with equality iff $q = \pi_t$. As $p_{\theta,\alpha} \propto p_\theta^\alpha$, the normalising constants combine into Z_t, giving (3). □

A.2 Proof of Lemma 1

Fix a round t and condition on the history $\mathcal{C}_t$. Under this conditioning, the effective reservoir support A_t, the inference-configured proposal $p_{\theta,\alpha}(\cdot \mid \mathcal{C}_t)$, and the acquisition function $a_t(x) = \mu_t(x) + \sqrt{\beta_t}\sigma_t(x)$ are fixed. Let

$$Z_t = \sum_{u\in A_t} \exp\{\eta\, a_t(u)\}\, p_{\theta,\alpha}(u \mid \mathcal{C}_t)$$

be the normalising constant of (15). For a tolerance $\zeta_t \ge 0$, write

$$U = U_t(\zeta_t) = \{x \in A_t : a_t(x) \ge a^\star_{t,A} - \zeta_t\}.$$

Every $u \in U$ has acquisition at least $a^\star_{t,A} - \zeta_t$. Hence the contribution of U alone gives the lower bound

$$\begin{aligned} Z_t &= \sum_{u\in A_t} \exp\{\eta\, a_t(u)\}\, p_{\theta,\alpha}(u \mid \mathcal{C}_t) \\ &\ge \sum_{u\in U} \exp\{\eta\, a_t(u)\}\, p_{\theta,\alpha}(u \mid \mathcal{C}_t) \\ &\ge \exp\{\eta(a^\star_{t,A} - \zeta_t)\}\, p_{\theta,\alpha}(U \mid \mathcal{C}_t) \\ &= \kappa_t(\zeta_t)\exp\{\eta(a^\star_{t,A} - \zeta_t)\}. \end{aligned} \tag{21}$$

Now fix $z \ge 0$ and define the bad event

$$B_z = \{x \in A_t : a^\star_{t,A} - a_t(x) > \zeta_t + z\}.$$

For every $x \in B_z$,

$$a_t(x) < a^\star_{t,A} - \zeta_t - z.$$

Therefore,

$$\begin{aligned} \mathbb{P}(x_t \in B_z \mid \mathcal{C}_t) &= \pi_t(B_z) \\ &= \frac{\sum_{x\in B_z}\exp\{\eta\, a_t(x)\}\, p_{\theta,\alpha}(x \mid \mathcal{C}_t)}{Z_t} \\ &\le \frac{\exp\{\eta(a^\star_{t,A} - \zeta_t - z)\}\, p_{\theta,\alpha}(B_z \mid \mathcal{C}_t)}{\kappa_t(\zeta_t)\exp\{\eta(a^\star_{t,A} - \zeta_t)\}} \\ &\le \frac{\exp\{-\eta z\}}{\kappa_t(\zeta_t)}, \end{aligned} \tag{22}$$

where the first inequality uses (21), and the second uses $p_{\theta,\alpha}(B_z \mid \mathcal{C}_t) \leq 1$. This proves the tail form

$$\mathbb{P}\left(a^\star_{t,A} - a_t(x_t) > \zeta_t + z \mid \mathcal{C}_t\right) \leq \frac{e^{-\eta z}}{\kappa_t(\zeta_t)}.$$

To obtain the high-probability statement, take

$$z = \frac{1}{\eta} \log \frac{1}{\rho_t \kappa_t(\zeta_t)}.$$

Since $\rho_t \in (0,1)$ and $\kappa_t(\zeta_t) \leq 1$, this z is non-negative. Substitution into (22) gives an upper bound of ρ_t on the bad event, so with conditional probability at least $1 - \rho_t$,

$$a^\star_{t,A} - a_t(x_t) \leq \zeta_t + \frac{1}{\eta} \log \frac{1}{\rho_t \kappa_t(\zeta_t)}.$$

This proves Lemma 1.

A.3 Proof of Theorem 1

The proof decomposes regret into a discovery term and an optimisation term, then controls the optimisation term using UCB calibration and Lemma 1.

As stated earlier, we can control the simple regret by analysing the upper bound of the cumulative regret, which can be decomposed by the discovery gap and optimisation regret as below.

$$\mathrm{Reg}_t = \underbrace{R(x^\star) - R^\star_{t,A}}_{\mathrm{Reg}_t^{\mathrm{disc}}} + \underbrace{R^\star_{t,A} - R(x_t)}_{\mathrm{Reg}_t^{\mathrm{opt}}}. \tag{23}$$

We begin with bounding the discovery gap. By Assumption 2, for every $x \in A_t$,

$$R(x^\star) - R(x) \leq L\, \mathrm{d}_{\mathcal{X}}(x, x^\star)^\chi.$$

Taking the infimum over $x \in A_t$ gives

$$\begin{aligned}
\mathrm{Reg}_t^{\mathrm{disc}} &= R(x^\star) - \sup_{x \in A_t} R(x) \\
&= \inf_{x \in A_t} \{R(x^\star) - R(x)\} \\
&\leq L \inf_{x \in A_t} \mathrm{d}_{\mathcal{X}}(x, x^\star)^\chi \\
&= L \left(\inf_{x \in A_t} \mathrm{d}_{\mathcal{X}}(x, x^\star) \right)^\chi \\
&= L (r_t^{\mathrm{cov}})^\chi.
\end{aligned} \tag{24}$$

Thus, the discovery price is exactly controlled by the current reservoir coverage radius.

Next, we form the simultaneous sampling event. For each t, apply Lemma 1 with failure probability ρ_t. Define

$$\xi_t = \zeta_t + \frac{1}{\eta} \log \frac{1}{\rho_t \kappa_t(\zeta_t)}.$$

The lemma gives, conditionally on $\mathcal{C}_t$,

$$\mathbb{P}(a^\star_{t,A} - a_t(x_t) \leq \xi_t \mid \mathcal{C}_t) \geq 1 - \rho_t.$$

Equivalently, the conditional failure probability is at most ρ_t. Iterating this bound over $t = 1, \dots, T$ and applying a union bound yields an event $\mathcal{E}_{\text{samp}}$ such that

$$\mathbb{P}(\mathcal{E}_{\text{samp}}) \geq 1 - \sum_{t=1}^{T} \rho_t \geq 1 - \delta_{\text{samp}},$$

and on $\mathcal{E}_{\text{samp}}$,

$$a^\star_{t,A} - a_t(x_t) \leq \xi_t \qquad \text{for all } t \leq T. \tag{25}$$

As such, we can bound the optimisation term on the good event. Let $\mathcal{E}_{\text{gp}}$ be the event in Assumption 1. On this event, for every $x \in \mathcal{X}$ and $t \leq T$,

$$R(x) \leq a_t(x), \qquad R(x) \geq \mu_t(x) - \sqrt{\beta_t}\sigma_t(x). \tag{26}$$

On $\mathcal{E}_{\text{gp}} \cap \mathcal{E}_{\text{samp}}$,

$$\begin{aligned} R^\star_{t,A} &= \sup_{x \in A_t} R(x) \\ &\leq \sup_{x \in A_t} a_t(x) \\ &= a^\star_{t,A} \\ &\leq a_t(x_t) + \xi_t, \end{aligned} \tag{27}$$

where the first inequality uses UCB optimism and the last inequality uses (25). Therefore

$$\text{Reg}_t^{\text{opt}} = R^\star_{t,A} - R(x_t) \leq a_t(x_t) - R(x_t) + \xi_t.$$

Using the lower side of (26) at the sampled point x_t,

$$\begin{aligned} a_t(x_t) - R(x_t) &= \mu_t(x_t) + \sqrt{\beta_t}\sigma_t(x_t) - R(x_t) \\ &\leq 2\sqrt{\beta_t}\sigma_t(x_t). \end{aligned} \tag{28}$$

Hence

$$\text{Reg}_t^{\text{opt}} \leq 2\sqrt{\beta_t}\sigma_t(x_t) + \xi_t. \tag{29}$$

Combining (23), (24), and (29), on $\mathcal{E}_{\text{gp}} \cap \mathcal{E}_{\text{samp}}$ we have, for every $t \leq T$,

$$\text{Reg}_t \leq L(r_t^{\text{cov}})^\chi + 2\sqrt{\beta_t}\sigma_t(x_t) + \xi_t.$$

Summing over t and using the monotonicity $\beta_t \leq \beta_T$,

$$\sum_{t=1}^{T} \text{Reg}_t \leq L\sum_{t=1}^{T}(r_t^{\text{cov}})^\chi + 2\sqrt{\beta_T}\sum_{t=1}^{T}\sigma_t(x_t) + \sum_{t=1}^{T}\xi_t. \tag{30}$$

By the standard information-gain variance bound for GP-UCB,

$$\sum_{t=1}^{T}\sigma_t^2(x_t) \leq C_\lambda \gamma_T, \qquad C_\lambda = \frac{2}{\log(1+\lambda^{-1})}.$$

Therefore, by Cauchy–Schwarz,

$$\sum_{t=1}^{T}\sigma_t(x_t) \leq \sqrt{T\sum_{t=1}^{T}\sigma_t^2(x_t)} \leq \sqrt{TC_\lambda\gamma_T}. \tag{31}$$

Substituting (31) into (30) gives

$$\sum_{t=1}^{T} \mathrm{Reg}_t \leq L \sum_{t=1}^{T} (r_t^{\mathrm{cov}})^{\chi} + 2\sqrt{T C_\lambda \beta_T \gamma_T} + \sum_{t=1}^{T} \xi_t.$$

Finally, substituting the definition of ξ_t,

$$\frac{1}{T}\sum_{t=1}^{T} \mathrm{Reg}_t \leq \frac{L}{T}\sum_{t=1}^{T} (r_t^{\mathrm{cov}})^{\chi} + 2\sqrt{\frac{C_\lambda \beta_T \gamma_T}{T}} + \frac{1}{T}\sum_{t=1}^{T}\left[\zeta_t + \frac{1}{\eta}\log\frac{1}{\rho_t \kappa_t(\zeta_t)}\right].$$

Assumption 1 gives $\mathbb{P}(\mathcal{E}_{\mathrm{gp}}) \geq 1 - \delta_{\mathrm{gp}}$, and the sampling argument gives $\mathbb{P}(\mathcal{E}_{\mathrm{samp}}) \geq 1 - \delta_{\mathrm{samp}}$. A final union bound gives

$$\mathbb{P}(\mathcal{E}_{\mathrm{gp}} \cap \mathcal{E}_{\mathrm{samp}}) \geq 1 - \delta_{\mathrm{gp}} - \delta_{\mathrm{samp}}.$$

This completes the proof of Theorem 1.

B Technical Implementation Details

This appendix elaborates on the core algorithmic and implementation details of the LDM framework, covering Gaussian process surrogate modelling, acquisition function design and batch sampling schemes. All implementations strictly follow the acquisition-tilted search framework presented in the main text, and only expand on the specific execution logic for surrogate model training and sampling.

B.1 Gaussian Process Surrogate

LDM employs a Gaussian Process (GP) as the probabilistic surrogate model. It fits the posterior distribution of the black-box reward function from historical experimental observations, and outputs calibrated predictive means and epistemic uncertainty to provide quantitative inputs for acquisition function computation.

B.1.1 Prior Specification

Let $\mathcal{X}$ denote the design space, $R(x)$ the true black-box reward function. Each experimental observation contains independent Gaussian noise:

$$r_i = R(x_i) + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, \sigma_n^2)$$

where σ_n^2 is the observation noise variance.

The GP prior is defined as:

$$R(x) \sim \mathcal{GP}\left(m(x),\, k(x, x')\right)$$

where:

- The prior mean function $m(x)$ adopts a constant prior, i.e. $m(x) = m_0$, with m_0 a scalar parameter that can be optimised from data;
- $k(x, x')$ is a positive-definite kernel function that characterises the correlation between samples in the design space. Its specific form is adapted to the design space, for example an RBF kernel for continuous parameter spaces or a string kernel for sequence spaces;
- The kernel hyperparameters, observation noise variance σ_n^2 and prior constant m_0 together form the set of parameters to be estimated for the GP.

B.1.2 Posterior Inference

Let $\mathcal{D}_t = \{(x_i, r_i)\}_{i=1}^t$ denote the cumulative historical observation dataset at iteration t. We define the $t \times t$ kernel matrix $\mathbf{K}_t$ and the t-dimensional cross-covariance vector $\mathbf{k}_t(x)$:

$$\mathbf{K}_t = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \dots & k(x_1, x_t) \\ k(x_2, x_1) & k(x_2, x_2) & \dots & k(x_2, x_t) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_t, x_1) & k(x_t, x_2) & \dots & k(x_t, x_t) \end{bmatrix}, \quad \mathbf{k}_t(x) = \begin{bmatrix} k(x_1, x) \\ k(x_2, x) \\ \vdots \\ k(x_t, x) \end{bmatrix} \in \mathbb{R}^t$$

By Bayesian conditioning, the posterior reward distribution for any candidate $x \in \mathcal{X}$ is Gaussian, with closed-form expressions for the posterior predictive mean and predictive variance:

$$\mu_t(x) = m_0 + \mathbf{k}_t(x)^\top \left(\mathbf{K}_t + \sigma_n^2 \mathbf{I}\right)^{-1} (\mathbf{r}_t - m_0 \cdot \mathbf{1}) \tag{32}$$

$$\sigma_t^2(x) = k(x, x) - \mathbf{k}_t(x)^\top \left(\mathbf{K}_t + \sigma_n^2 \mathbf{I}\right)^{-1} \mathbf{k}_t(x). \tag{33}$$

where $\mathbf{r}_t = (r_1, \dots, r_t)^\top$ is the vector of observed rewards, and $\mathbf{1}$ is an all-ones vector of length t. The posterior mean estimates the expected reward of a candidate, while the posterior variance quantifies the epistemic uncertainty of the prediction.

B.1.3 Hyperparameter Estimation

The GP hyperparameters are fitted via *Type-II maximum likelihood estimation*, whereby optimal hyperparameter values are determined by maximising the marginal log-likelihood of the observed data. The marginal log-likelihood of the observed data $\mathcal{D}_t$ is given by:

$$\log p(\mathbf{r}_t \mid \mathcal{D}_t) = -\frac{1}{2}(\mathbf{r}_t - m_0 \cdot \mathbf{1})^\top \left(\mathbf{K}_t + \sigma_n^2 \mathbf{I}\right)^{-1} (\mathbf{r}_t - m_0 \cdot \mathbf{1}) - \frac{1}{2} \log \left|\mathbf{K}_t + \sigma_n^2 \mathbf{I}\right| - \frac{t}{2} \log 2\pi$$

Optimisation is performed using gradient-based algorithms such as L-BFGS. The parameters to be optimised include kernel hyperparameters such as lengthscales and output variance, observation noise variance σ_n^2, and the prior constant m_0. This method automatically adapts to the smoothness and noise level of the design space, improving the fitting accuracy and uncertainty calibration of the surrogate model. Hyperparameter optimisation is re-run after each iteration with new observations to update the surrogate model.

B.2 Acquisition Functions

Acquisition functions quantify the experimental value of each candidate based on the GP posterior mean and uncertainty, and act as the core value signal guiding search direction and balancing exploitation and exploration. LDM supports three acquisition functions for single- and multi-objective settings, all of which can be directly embedded into the acquisition-tilted search framework.

B.2.1 Expected Improvement

Expected Improvement (EI) (Jones et al., 1998) measures the expected gain of a candidate relative to the current best observed value. It is one of the most widely used acquisition functions in single-objective optimisation, and takes the form:

$$\alpha^{\mathrm{EI}}(x) = (\mu_t(x) - r_t^* - \xi)\, \Phi(z) + \sigma_t(x) \varphi(z), \quad z = \frac{\mu_t(x) - r_t^* - \xi}{\sigma_t(x)} \tag{34}$$

where $r_t^* = \max_{i \le t} r_i$ is the best reward observed so far, $\xi \ge 0$ is an exploration-adjustment hyperparameter, and $\Phi(\cdot)$ and $\varphi(\cdot)$ denote the cumulative distribution function and probability density function of the standard normal distribution, respectively. EI naturally balances exploitation and exploration, prioritising candidates with higher expected returns.

B.2.2 Upper Confidence Bound

The Upper Confidence Bound (UCB) (Srinivas et al., 2010) computes a weighted sum of the predictive mean and uncertainty. It has a concise form and enjoys rigorous theoretical guarantees on cumulative regret, and takes the form:

$$\alpha^{\mathrm{UCB}}(x) = \mu_t(x) + \sqrt{\beta_t}\,\sigma_t(x) \tag{35}$$

where $\beta_t > 0$ is the exploration weight coefficient, which may be set via theoretical formulae or tuned as a hyperparameter to control the exploration intensity of the algorithm. By adjusting β_t, UCB can flexibly switch between exploitation-biased and exploration-biased search strategies.

B.2.3 Expected Hypervolume Improvement

For multi-objective optimisation tasks, **Expected Hypervolume Improvement (EHVI)** (Ponweiser et al., 2008) is adopted as the acquisition function. EHVI computes the expected hypervolume gain of the current empirical Pareto front after adding a candidate, and automatically balances exploitation of favourable objective trade-offs and exploration of under-sampled design regions, without requiring pre-specified objective weights.

In the multi-objective setting, the core acquisition-tilted search framework remains unchanged; only the scalar acquisition value $a_t(x)$ is replaced with the EHVI value, enabling seamless extension of LDM to multi-objective discovery.

B.3 Batch Acquisition Sampling

In parallel experimental settings, multiple candidates must be selected for simultaneous evaluation at each iteration. LDM adopts Gumbel-top-k sampling to draw batches of candidates directly from the acquisition-tilted target distribution $\pi_t(x)$, without requiring additional diversity regularisation terms.

The implementation procedure is as follows:

1. For each candidate x_i in the pool generated by the LLM, compute its unnormalised tilted weight:
$$w_i = p_{\theta,\alpha}(x_i \mid \mathcal{C}_t) \cdot \exp\{\eta \cdot a_t(x_i)\}$$
where $p_{\theta,\alpha}(x_i \mid \mathcal{C}_t)$ is the conditional generation probability of the LLM, $a_t(x_i)$ is the acquisition function value, and η is the acquisition tilt strength coefficient.

2. Sample independent Gumbel noise $g_i \sim \mathrm{Gumbel}(0, 1)$ for each weight, rank candidates by $\log w_i + g_i$ in descending order, and select the top k candidates to form the evaluation batch. Each selected candidate is removed from the pool after selection.

Batch samples produced by this method strictly follow the target tilted distribution, naturally preserving structural diversity while maintaining acquisition value, and are compatible with batch evaluation pipelines in parallel experiments.

C Ablation Studies

These regimes stress different parts of the framework. `autoresearch` is a *multi-turn code-editing* task: the agent maintains a research state, repeatedly edits a persistent artifact `train.py`, and can change the representation of the search space as new mechanisms are discovered. Small-molecule and CDRH3 design are closer to *single-step proposal* tasks: at each BO round, the LLM proposes or parameterises a candidate pool, the surrogate acquisition selects from that pool, and the expensive oracle evaluates the selected designs.

The ablations therefore have two complementary purposes. First, in §C.1, we push the boundary of a pure current-agent research loop on nanoGPT by removing the calibrated BO value entirely. This tests how far an LLM agent can go when it can read its own logs and reflect, but cannot search against a surrogate posterior. Second, in §C.2, we keep the value model and study LDM test-time search itself. On nanoGPT this means varying the internal LDM-TTS settings, such as inner search budget, dynamic features, and acquisition choice. On molecules and CDRH3, it means asking whether open-source LLM proposers benefit from larger proposal and acquisition-selection budgets in single-step design rounds.

C.1 Pure LLM-based research loop

This ablation asks whether the LLM research loop alone is sufficient. Operationally, it is the $\eta = 0$ limit of Eq. (3): the agent still reads the history, edits `train.py`, launches experiments, and keeps improvements, but it does not receive a BO posterior, an acquisition score, or an uncertainty-directed suggestion for what to try next. We deliberately make this a strong LLM-only baseline rather than a straw man: it is meant to push the boundary of what a current advanced agent system can do on this task. The only extra scaffolding is a short reflection instruction that asks the Codex agent to monitor the result log as a whole, ask whether the current iteration verifies the previous hypothesis, extract any new mechanistic insight, and plan the next iteration. This is not human-in-the-loop steering of the scientific direction; it is an end-to-end LLM workflow whose context explicitly asks the agent to behave like a careful experimentalist. Thus the ablation tests the best version of the claim that an LLM, supplied with its own experimental transcript and enough task knowledge to edit the architecture and optimizer, can bootstrap an autonomous research process without an explicit value model.

Figure 17 shows that the pure LLM loop is not inert. It discovers several coherent research phases: first dense scaling, then sparse memory, then a rebalance of dense compute around the sparse memory idea, followed by confirmation and grid search around the 512-sparse frontier. The best-so-far curve falls from the initial `val_bpb` ≈ 1.069 to ≈ 1.01 after the first phase, to ≈ 0.991 after the second reflection, and to ≈ 0.959 after the sparse-frontier breakthrough. In other words, a modern code agent with a task-specific prompt and access to its own run log can already produce human-readable, scientifically reasonable progress: it can form hypotheses, translate them into code edits, test them, preserve improvements, and revise its research state.

The archived loop logs show what this "research work" consisted of. The agent did not merely enumerate nearby hyperparameters. After each phase it wrote a reflection report: it compressed the ledger into mechanistic claims, labelled memories by regime and mechanism, retrieved positive and negative examples, and used the labelled memories to choose the next experiment family. Table 3 summarises the resulting trajectory. The important point is the *statefulness*: the loop changes what it believes the problem is. It begins with ordinary dense scaling, decides that dense capacity has saturated under the five-minute budget, invents sparse associative value memory as a cheaper capacity axis, then shrinks the dense core so that the sparse-memory model can train in time, and finally switches to repeat-first, VRAM-aware frontier tuning once improvements fall into the noise band.

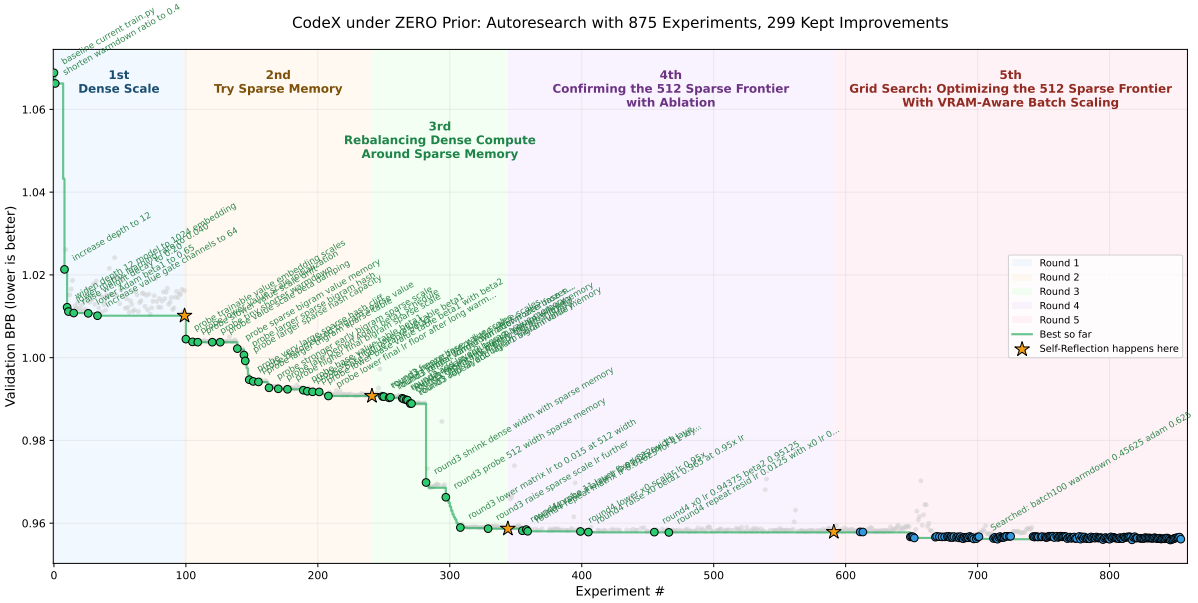


Figure 17: **Pure LLM research loop on `autoresearch`.** Validation `val_bpb` (lower is better) across 875 no-BO, no-acquisition experiments. Gray points are individual trials and the green step curve is the best value found so far; yellow stars mark self-reflection checkpoints where the agent summarises the ledger and enters a new research regime. The coloured bands show the loop's staged progress: dense scaling, sparse-memory invention, dense-compute rebalancing around sparse memory, ablation-based confirmation of the 512-sparse frontier, and final VRAM-aware batch/schedule search. The curve demonstrates that a pure LLM loop can do real sequential research, but also its ceiling: after the large regime shifts it plateaus near `val_bpb` ≈ 0.956, above the LDM result of 0.93421 in Figure 6.

This trace is the strongest evidence that the baseline is a genuine research loop. It performs causal ablations (for example, removing trigram memory, early bigram memory, or sparse scales), estimates the noise floor by repeating restored anchors, and eventually augments the ledger with trajectory diagnostics such as steps, tokens, MFU, train/eval gap, and loss slopes. It also learns what not to do: the contexts explicitly retrieve failed routing rewires, failed dense-width probes, crashes, and repeated optimizer mismatches as negative memories. In short, the loop accumulates procedural knowledge, not just a list of winning commits.

The limitation is equally visible. After the third breakthrough, the remaining ≈ 500 experiments are mostly local confirmation and grid search around the same frontier. The loop continues to keep small improvements, but the envelope is nearly flat, ending around `val_bpb` ≈ 0.956. This is better than the no-discover Karpathy baseline in Figure 6, which stalls at 0.9767, but it is still far from the LDM curve, which reaches 0.93421 under the same H100 five-minute-run setting. The pure LLM loop can reason about its own transcript and name plausible new regimes, but between those reflections it has no calibrated estimate of where improvement is likely, where uncertainty remains high, or when a plateau is evidence that the current boundary is exhausted.

This is the ablation's main conclusion. The bottleneck is not proposal expressivity: the LLM can write useful training code, exploit prior knowledge about how the architecture can be modified, and remember enough of the transcript to pursue multi-stage ideas. The bottleneck is value. Without μ_t, σ_t, and the acquisition a_t, **the run log remains a narrative memory rather than a searchable value landscape**. LDM supplies exactly that missing object: the surrogate turns the same history into a posterior over the program space, and the acquisition turns that posterior into a decision value for both local refinement and boundary-moving discover steps. The pure LLM loop therefore validates the reservoir side of the framework and shows how far

Phase	Working hypothesis	Evidence generated by the loop	Best `val_bpb`
Dense scaling	Capacity is the early bottleneck, but only while the model can still be trained inside the fixed budget.	Depth 10 and 12 give large drops; widening the depth-12 model helps; depth 14 worsens near the memory ceiling. The reflection turns this into the rule: dense capacity helps until the update budget becomes the bottleneck.	1.010136
Sparse value memory	Dense capacity is saturated, but cheap associative capacity may still help if routed through a bounded value path.	Trainable value scales, sparse hashed bigram memory, larger hash tables, decorrelated trigram memory, and phase-specific damping improve the frontier; broader routing and optimizer rewires are recorded as negative memories.	0.990738
Dense-compute rebalance	Sparse memory works best when the dense core is smaller and no longer steals the fixed training budget.	Causal ablations show trigram memory, early bigram memory, and trainable sparse scales matter. Shrinking dense width to 640 and then 512 creates the large breakthrough, after which matrix LR, final LR floor, and sparse-scale LR are retuned.	0.958666
Frontier confirmation	The 512 sparse-memory regime is a narrow compute/update-budget island, and remaining gains require repeats.	Widths below and above 512 lose; removing early ngram memory increases throughput but hurts validation, showing the memory path is causal. Restored-anchor repeats estimate a noise floor of roughly 3–5×10^{-4} BPB.	0.957766
VRAM-aware batch tuning	Spare VRAM should buy batch/throughput and coupled schedule tuning, not a return to dense scaling.	The context retrieves the dense-shrink successes and negative memories against larger dense models, then grid-searches batch size, warmdown, final LR, Adam damping, and weight decay while logging steps, tokens, MFU, and loss slopes.	0.955923

Table 3: **Mechanistic trace of the pure LLM research loop.** Each row is a phase extracted from the loop logs. The LLM acts like a lightweight experimental scientist: it writes a hypothesis, tests code-level interventions, records both successes and failures, updates a regime label, and carries those memories into the next phase.

current agent systems can already go, while also showing why the BO-value side is necessary for the performance gains in the main experiment.

C.2 Test-time Search for LDM

This ablation asks whether LDM exhibits the same test-time scaling behaviour that motivates inference-time search in language-model reasoning (Snell et al., 2025), but in the harder setting where the value is an expensive scientific objective approximated by a surrogate. We keep the outer evaluation budget fixed and vary only the cheap inner-loop budget of LDM-TTS. Concretely, the inner budget has four knobs: (i) how many candidate programmes the LLM proposes at each outer iteration, (ii) how many hypothesis or refinement branches the BO inner loop expands for each candidate before scoring, (iii) how many of those scored candidates the acquisition retains for the next round, and (iv) which acquisition function scores them. The detailed `autoresearch` test-time-search sweep, with its N4H4/N8H8 budgets and EI vs. posterior-mean comparison, already appears in Section 6.5; here we keep only the molecule and CDRH3 single-step budget sweeps.

The theory in §5 predicts exactly this dependence: increasing the near-acquisition mass available

to the tilted sampler reduces the LDM sampling shortfall term, but only if the added candidates are also evaluated by a calibrated value model rather than by language plausibility alone.

C.2.1 Small-molecule drug discovery

The molecular ablation repeats the same question in a chemically open-ended space. Here the expensive evaluation is a two-objective KRAS G12D score, and performance is the dominated Pareto hypervolume under the Vina/activity objectives of §6.3. We use the Direct-Softmax LDM pipeline with a Qwen3.5-9B proposer and vary two inner-loop budgets. A label `proposer{K}_bo{M}` means that the LLM proposes a pool of K candidate SMILES strings and the BO/EHVI inner loop is allowed to score up to M candidates and retain the best-scoring ones before the next expensive molecular evaluation is chosen. Thus K controls the breadth of the LLM reservoir draw, while M controls how much of that breadth is exposed to the calibrated multi-objective value.

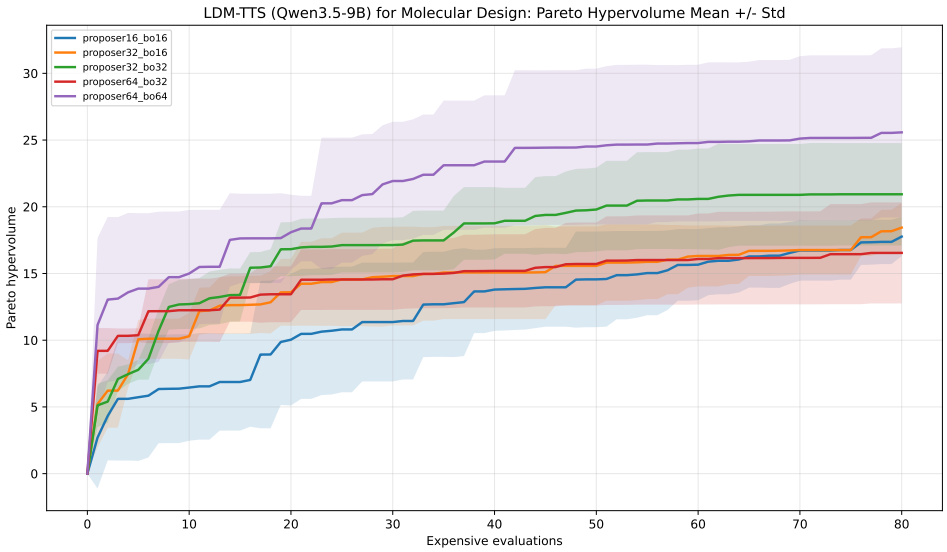


Figure 18: **Test-time search scaling for molecular design.** Pareto hypervolume (higher is better) over expensive molecular evaluations for Qwen3.5-9B Direct-Softmax LDM-TTS. Curves show the mean over independent runs, with shaded bands denoting one standard deviation; legend entries report the LLM proposal budget and BO/EHVI acquisition-scoring budget. Balanced scaling of both budgets improves hypervolume most: `proposer64_bo64` reaches the strongest final Pareto front, while increasing proposals without matching acquisition bandwidth gives smaller or unstable gains.

Figure 18 shows that test-time compute also scales in the molecular setting, but only when both inner budgets grow together. Increasing the LLM proposal budget from 16 to 32 improves the curve, and increasing the BO/EHVI scoring budget from 16 to 32 improves it further: `proposer32_bo32` dominates the smaller-budget settings through most of the run. `proposer64_bo64` makes the largest early jump and keeps the highest final hypervolume; the asymmetric `proposer64_bo32` does worse, which shows that broad SMILES generation is useful only when the EHVI side has enough bandwidth to filter the resulting candidates.

Figure 19 adds the acquisition-function dimension. At small budgets, mean- or UCB-style acquisitions (a 0.5/0.5 weighted scalarisation of the Vina and activity objectives) outperform the strict EHVI; as n grows, EHVI overtakes them and continues to improve, while mean-style acquisitions degrade or stagnate. The mechanism is bandwidth: EHVI is built on the strict Pareto frontier and therefore needs a large screening budget to be effective, because multi-objective

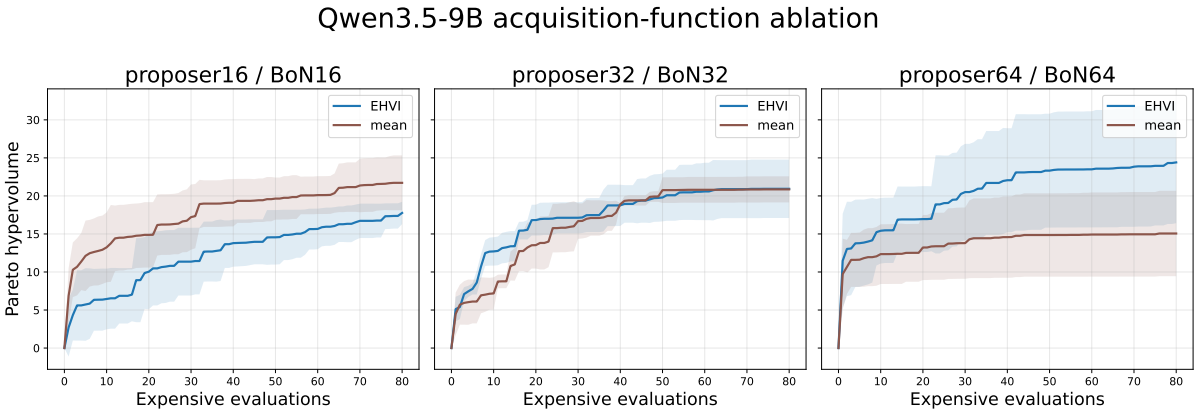


Figure 19: Abaltion on acquisition function over different search budgets for molecular design.

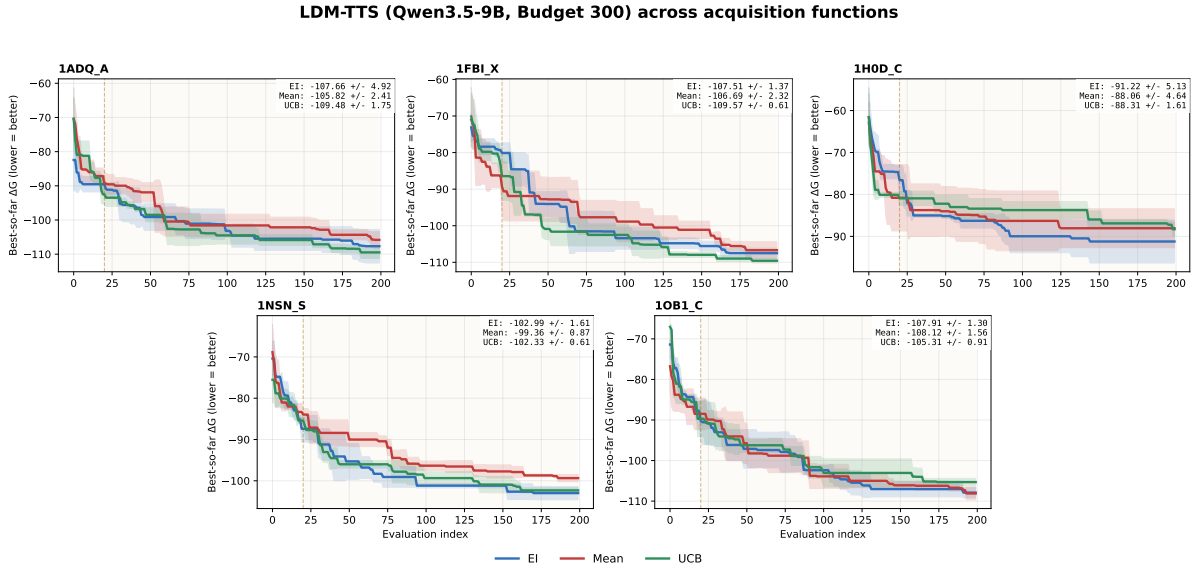


Figure 20: Comparison of different acquisition functions for LDM-TTS with a fixed proposer budget of 300 on Antibody CDRH3 design.

improvements correspond to sparse non-dominated points. Mean-style acquisitions collapse the two objectives into one scalar, so at small budgets the per-step randomness from a thin screening budget partially substitutes for the missing Pareto signal, smoothing out the optimisation and keeping the curve flat; once the budget grows, EHVI's strict-screening advantage takes over.

C.2.2 Antibody CDRH3 design

Figure 20 compares different acquisition functions on the five antigens at a fixed proposer budget of 300. LDM's calibrated acquisitions (EI, UCB and variants) are broadly better than mean-style and random baselines, mirroring the molecular trend above. The result shows that a calibrated acquisition still extracts value from a broader candidate pool even when the open-source LLM has only weak biological priors, and qualifies the §6.5.2 antibody observation: hyperparameter sweeps have only weak sensitivity in this regime, but the choice of acquisition function still matters once the BO side has enough candidates to screen.

The CDRH3 ablation is the sequence-design counterpart of the molecular budget-scaling experiment. The outer oracle budget is again fixed, but here the design space is a known, sparse,

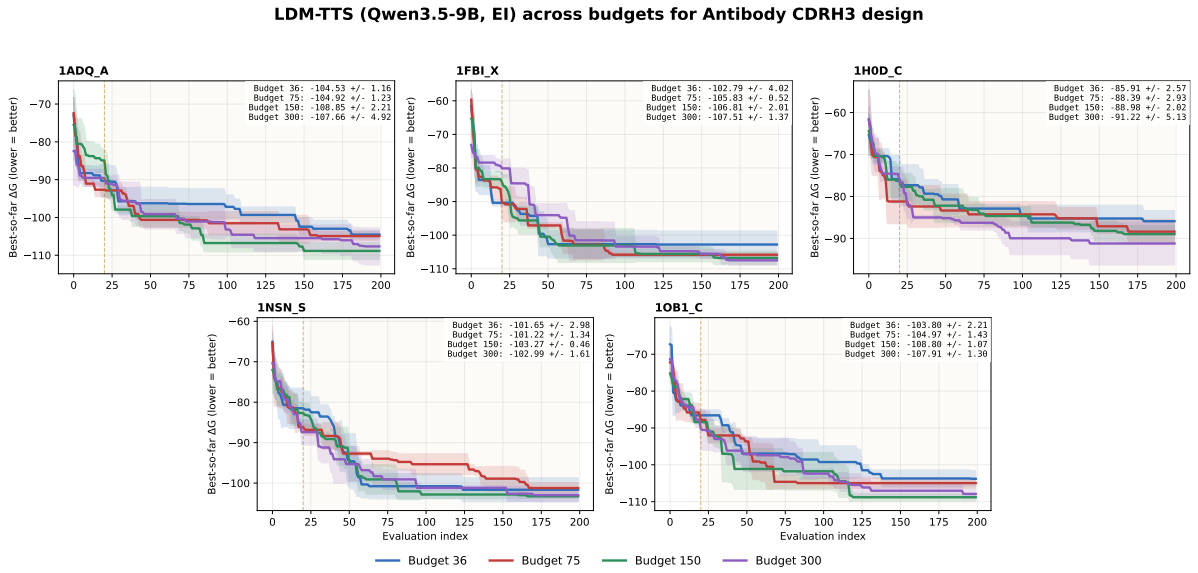


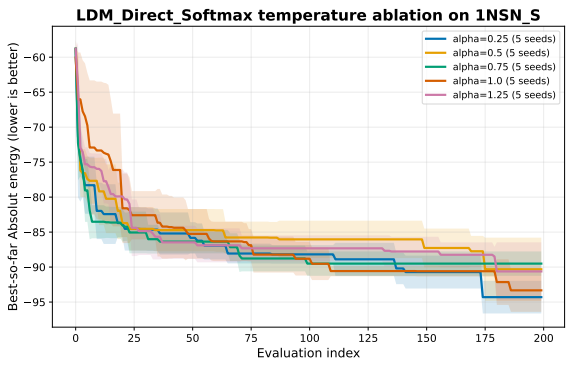
Figure 21: Effect of LDM-TTS proposer budget scaling on Antibody CDRH3 design. Each budget denotes the number of candidate sequences or sequence-neighbourhood samples generated before an expensive oracle evaluation. Larger inference-time proposal budgets improve the attainable binding-energy plateau across antigen targets.

discrete sequence space rather than an open-ended SMILES space. We therefore ask a narrower question: does the LDM-TTS proposer budget on its own improve the best sequence found? A label `Budget` N denotes an inner proposer budget of N candidate CDRH3 sequences or neighbourhood samples generated before the next expensive oracle query. The CDRH3 ablation fixes the BO inner-loop side at its maximum feasible setting: after validity checks, deduplication, and history filtering, the acquisition is allowed to score all remaining candidates. So the experiment changes only the cheap LLM-side proposal breadth, not the number of oracle evaluations.

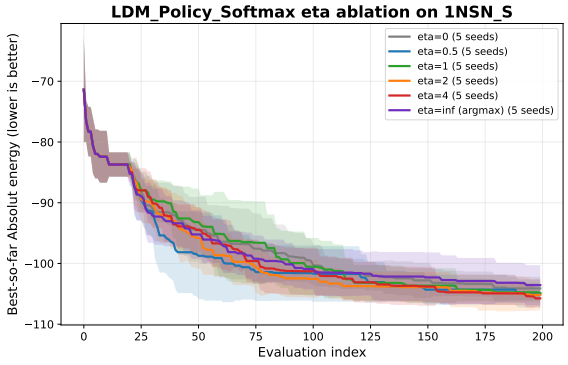
Figure 21 shows the budget effect across antigen landscapes, which is weaker than that observed in molecular design. The smallest budget, `Budget` 36, improves quickly at the beginning but usually reaches a higher plateau. Increasing the proposer budget to 75 already improves the final binding energy on most targets, and the larger 150 and 300 budgets give the best final result on all five antigens. The strongest setting is target-dependent: budget 150 is best on 1ADQ_A, 1NSN_S, and 1OB1_C, while budget 300 is best on 1FBI_X and 1H0D_C. This non-monotonicity is expected in a rugged sequence landscape with finite seeds and a noisy surrogate: after the acquisition has enough candidates to cover the useful local neighbourhoods, further expansion can add redundant or lower-quality regions as well as genuinely novel ones. The main conclusion is therefore not that the largest budget always wins, but that the low-budget regime is systematically acquisition-starved and that moderate-to-large test-time pools substantially lower the attainable binding-energy plateau.

This result complements the main CDRH3 comparison in Figure 7. There, policy-mode LDM outperforms direct token-level generation because the open-source LLM has a weaker biological sequence prior than it has for code or SMILES; emitting a few complete CDRH3 loops does not expose enough useful reservoir mass for BO to exploit. The ablation here isolates what happens once the LDM pipeline can expand the candidate pool: additional inference-time candidates become useful precisely because they are not selected by language plausibility alone. They are filtered by the surrogate acquisition, which converts a larger discrete reservoir into stronger best-so-far binding energy without spending extra oracle calls.

Together, the molecule and CDRH3 ablations show that test-time scaling for single-step design



(a) Effect of the LLM sampling temperature α in $\mathrm{LDM}_{\mathrm{DirectSoftmax}}$. The acquisition temperature is fixed to $\eta = 1$.



(b) Effect of the acquisition temperature η in $\mathrm{LDM}_{\mathrm{PolicySoftmax}}$. The LLM sampling temperature is fixed to $\alpha = 1$.

Figure 22: Hyperparameter ablations on the antibody-design task 1NSN_S. Each curve reports the mean best-so-far Absolut energy over five random seeds, with the shaded region showing one standard deviation. Lower values are better.

is useful only when the added samples are exposed to a calibrated value model. Richly pretrained SMILES models can absorb large direct proposal batches; sparse biological sequence spaces need enough sequence or neighbourhood coverage before acquisition filtering becomes effective. This complements the nanoGPT ablation: in multi-turn code search, LDM-TTS improves by searching a changing program landscape; in single-step design, it improves by scaling the candidate pool a calibrated acquisition can choose from.

C.3 Hyperparameter ablations for LDM.

We here supplement the sensitivity analysis of the two LDM variants on antigen 1NSN_S using five random seeds. For $\mathrm{LDM}_{\mathrm{DirectSoftmax}}$, we vary the LLM sampling temperature $\alpha \in \{0.25, 0.5, 0.75, 1, 1.25\}$ while fixing the acquisition temperature to $\eta = 1$. For $\mathrm{LDM}_{\mathrm{PolicySoftmax}}$, we fix $\alpha = 1$ and vary the acquisition temperature $\eta \in \{0, 0.5, 1, 2, 4, \infty\}$. Here, $\eta = 0$ corresponds to uniform selection among the policy representatives, while $\eta = \infty$ is equivalent to acquisition-function argmax. The per-temperature curves are plotted in Figure 22; we discuss their interpretation below.

This overall behaviour is consistent with the discussion in Section 6.5. Figure 22 shows the per-temperature curves. The open-source LLM lacks biological sequence domain priors, so its autoregressive proposals over the 20^{11} CDRH3 space are effectively near-random; in the policy variants the LLM already implements an implicit form of acquisition-aware importance sampling by expanding the candidate neighbourhood around promising incumbents rather than committing to a single sequence, which dampens the marginal effect of further reweighting at test time. Consequently, global sweeps over the sampling temperature α and the acquisition temperature η on antigen 1NSN_S produce only weak sensitivity: Direct-Softmax slightly favours low α and Policy-Softmax shows a marginal edge near $\eta = 4$, but all error bars overlap. Adjusting the two temperature knobs does not significantly change the final binding energy, which is consistent with the observation in §6.2 that the policy-mode LDM gain over direct generation comes from the LLM's structural priors rather than from acquisition reweighting alone.

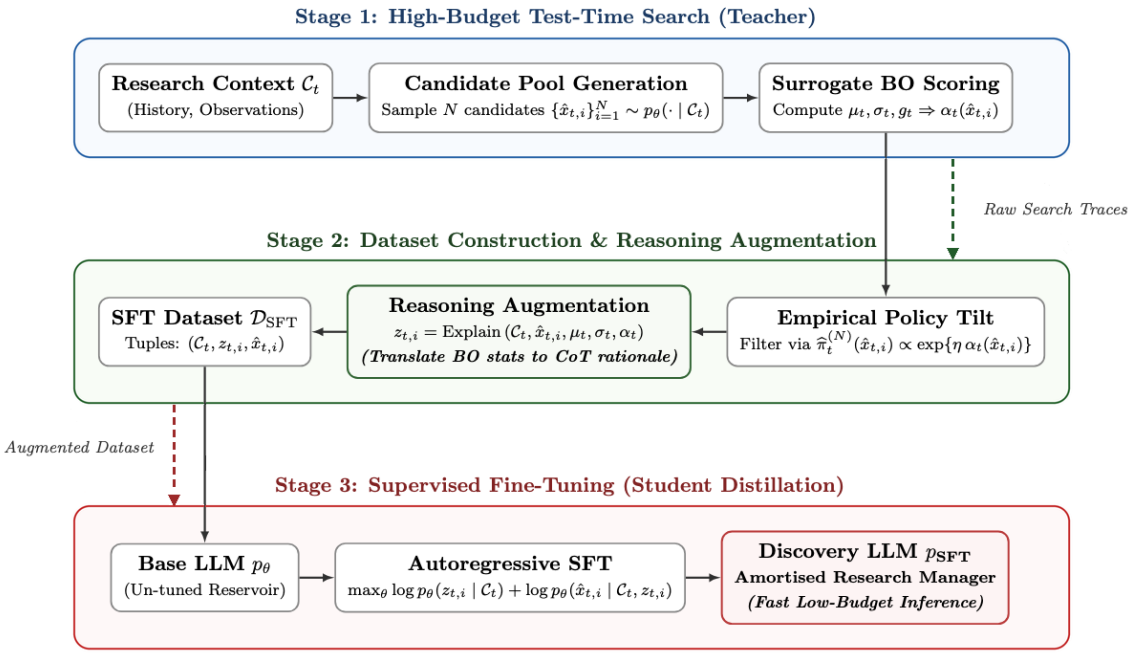


Figure 23: **Overview of the LDM training pipeline workflow. Stage 1:** High-budget test-time search generates candidate pools scored by surrogate acquisition values (α_t). **Stage 2:** High-value candidates are filtered via an empirical policy tilt $\widehat{\pi}_t^{(N)}$, and numerical BO statistics are translated into semantic reasoning rationales ($z_{t,i}$) via reasoning augmentation. **Stage 3:** The base model is fine-tuned on the CoT target $(z_{t,i}, \hat{x}_{t,i})$ using standard next-token prediction, compiling the high-budget search policy into an amortised student model (p_{SFT}) capable of fast, low-budget inference.

D Additional Fine-Tuning Analyses

Section 4.4 describes how high-budget LDM test-time search is distilled into a student proposer, and Section 6.4 reports the main fine-tuning results on AutoResearch, antibody design, and small-molecule discovery. This appendix provides complementary evidence that is not repeated in the main text.

D.1 Fine-Tuning as Value Distillation

Modern LLM fine-tuning paradigms can be distinguished by the decision-theoretic value they distill into the model weights. Chat LLMs distill human preference value, reasoning LLMs distill verification value, and Discovery LLMs distill epistemic acquisition value. The object distilled by LDM is therefore not the measured reward $R(x)$ alone, nor the static memorisation of a high-reward molecule, protein, or program. Instead, the goal is to distill the acquisition strategy that decides which scarce experiment is worth running under uncertainty. Figure 23 summarises this three-stage training pipeline: high-budget test-time search, reasoning-augmented dataset construction, and supervised distillation into the student proposer.

Table 4 illustrates how numerical acquisition evidence is translated into semantic supervision across the three discovery domains.

Domain	Acquisition evidence (Numerical)	Example augmented rationale (Semantic)
`autoresearch` code search	Best run `state_0019` reaches `val_bpb` 0.9811 by lowering `TOTAL_BATCH_SIZE` and tuning `ASPECT_RATIO`; several near-best runs also lower the batch size. The current parent already sits at the minimum batch size with only 5 active parameters and no improvement over the last 4 rounds.	Lowering the total batch size has been a robust positive move—more steps fit inside the fixed 30 s budget—but that lever is now exhausted and the active parameters have **plateaued over the last four rounds**, so the set is near its ceiling. Rather than re-tuning saturated knobs, I should **activate a currently-inactive parameter** (e.g. `MATRIX_LR`) to open a new optimisation dimension.
Antibody CDRH3 design	Best sequence `ADGHTKQNPRW` at −64.28; the last four candidates share the fixed prefix `ADGHTKQNPR` and scan only the final position (W −64.28, Y −62.69, A −59.38, S −57.96).	The history is a single-point scan of the last residue on an already-refined scaffold, with the bulky aromatic W best so far. The informative move is to keep the refined prefix and test an **untested residue of similar hydrophobic character**—leucine (L)—which may reproduce W's favourable hydrophobic contact; propose `ADGHTKQNPRL`.
Small-molecule discovery	29 molecules evaluated and the loop is flagged `stalled`; the top actives ($\sim$ 8.1) share one isoxazolone/thiazolone fused core with isopropoxy and cyano groups, docking ~ -8.	Because the loop is stalled and near-duplicates are disallowed, adding a methyl to a known winner carries no information. The higher-value move is a **recombination or scaffold hop**—e.g. combining the pyridine-ring variant with the morpholine side chain into a molecule not yet seen—so the candidate probes a genuinely new region of the Pareto frontier rather than exploiting a saturated one.

Table 4: **Reasoning augmentation turns numerical acquisition labels into semantic training signals.** The rationales are faithful (translated and condensed) excerpts of reasoning traces generated by the self-hosted DeepSeek V4 Flash teacher inside the high-budget LDM-TTS loop, one per domain. They describe the epistemic value of managing research progress by reading the evaluated history, diagnosing stagnation, and choosing an information-adding move, rather than focusing only on the domain-specific final solution.

D.2 Cross-Molecule Transfer without Reasoning Augmentation

We test the minimal form of discovery fine-tuning without an explicit reasoning target. Training data are collected from high-budget LDM-TTS traces on a source molecular optimisation task. The Qwen3.5-9B proposer is fine-tuned on acquisition-weighted examples and then evaluated on a different molecular optimisation task within the same acquisition-guided LDM loop.

This ablation retains the same LDM state and action schema as the reasoning-augmented setting, but omits the explicit research-progress rationale $z_{t,i}$.

This setting shows positive transfer across molecular tasks. The base Qwen3.5-9B LDM reaches a final hypervolume of 16.489 ± 5.867, whereas the fine-tuned model reaches 22.279 ± 3.828 under the same evaluation protocol. Because the discovery data are collected from one molecule and evaluated on another, the improvement cannot be explained by memorising the final molecules from the source task. It instead shows that acquisition-weighted fine-tuning can transfer part

of the research-management policy even without an explicit reasoning channel.

D.3 Supplementary Quantitative Tables

Table 5 retains the G12C results not shown in the main-text KRAS G12D learning curve. Table 6 collects the numerical cross-domain comparison underlying the three main-text figures.

Inference policy	**KRAS G12C** HV ↑	**KRAS G12D** HV ↑
Qwen3.5-9B base (without_cot)	11.64 ± 2.63	16.49 ± 5.87
Fine-tuned Qwen3.5-9B (without_cot)	14.73 ± 3.09	22.28 ± 3.83
DeepSeek V4 Flash (without_cot)	18.97 ± 1.76	24.55 ± 3.59
DeepSeek V4 Flash (with_cot)	19.98 ± 2.00	24.07 ± 3.12
Fine-tuned Qwen3.5-9B (with_cot)	$\mathbf{20.98 \pm 2.92}$	$\mathbf{26.66 \pm 2.61}$

Table 5: **Reasoning-augmentation ablation on small-molecule discovery.** Final Pareto-front hypervolume at an evaluation budget of 80, averaged over five seeds.

Model	**CoT**	**AutoResearch** val_bpb ↓	**Antibody** $-E_{\text{bind}}$ ↑	**SmallMol** HV ↑
Base Qwen3.5-9B	off	0.9965	102.6	16.49
Base Qwen3.5-9B	on	0.9825	104.4	—
LDM-SFT	off	0.9829	105.1	22.28
LDM-SFT	**on**	**0.9788**	**105.5**	**26.66**
Qwen3-Coder-30B ref.		0.9801	—	—

Table 6: **Cross-domain reasoning-augmentation ablation.** Terminal metrics for the three fine-tuning studies.

D.4 In-Distribution Single-Task Fit

Before evaluating the distilled policies in the acquisition loop, we verify that each single-task model fits its reasoning-augmented training data. We track training and held-out losses for `autoresearch` program search, small-molecule KRAS design, and antibody CDRH3 design. All three losses converge within a few dozen steps, while the held-out curves closely track the training curves. The absolute held-out losses differ by domain: 0.06 for `autoresearch`, 0.38 for small-molecule design, and 0.70 for antibody CDRH3 design. These differences reflect the different action spaces and target lengths, rather than a failure to fit the data.

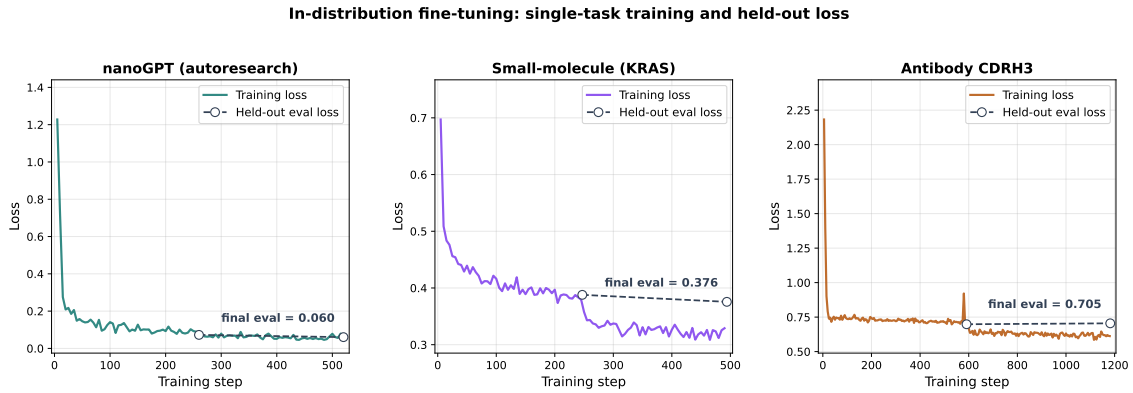


Figure 24: **In-distribution single-task fine-tuning.** Training loss (solid) and held-out evaluation loss (markers) versus training step for the `autoresearch`, small-molecule, and antibody CDRH3 policies. All three converge within a few dozen steps, and the held-out loss tracks the training loss, confirming a faithful in-distribution fit before acquisition-loop evaluation.

D.5 Case-level Out-of-distribution Generalisation

To test whether it learns a transferable acquisition policy rather than memorising antigen-specific patterns, we hold two of the five antibody targets (`1FBI_X`, `1H0D_C`) out of training entirely, fine-tune the mixed-task model on the other three, and evaluate on the held-out pair inside the same acquisition-guided loop, antigens whose sequences and binding landscapes were never seen during training.

Figure 25 shows the out-of-distribution model matches or exceeds the *in-distribution* model (which *was* trained on these antigens) on both held-out targets, and clearly beats base Qwen3.5-9B with and without chain-of-thought: it reaches −110.57 on `1FBI_X` (base −109.5, in-distribution −113.4) and −95.4 on `1H0D_C`, where it even surpasses the in-distribution model (−94.6) and the base (−90.9). Since the model never observed these antigens, the gain cannot be memorisation—the distilled acquisition policy transfers to unseen targets.

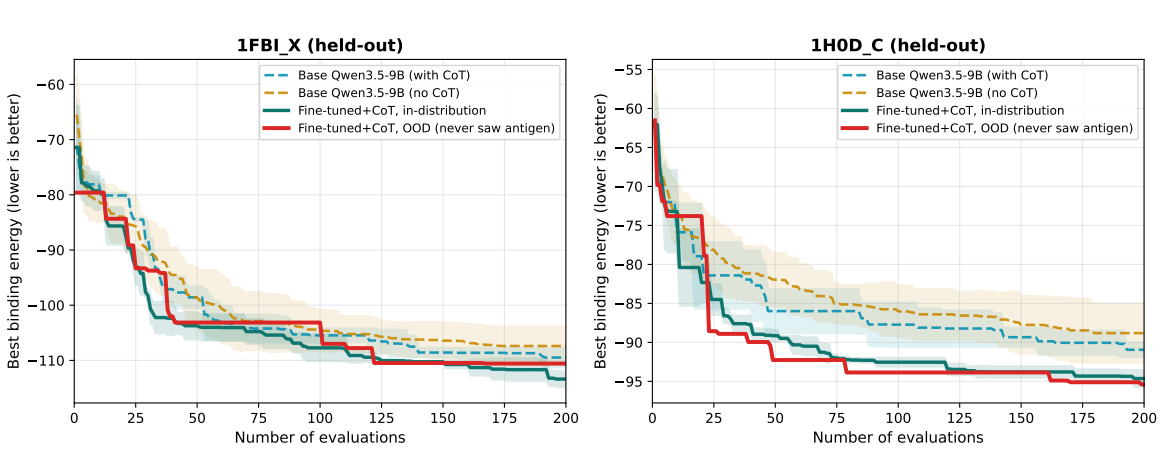


Figure 25: **Out-of-distribution generalisation to held-out antigens (3-train/2-test).** Best Absolut binding energy versus number of evaluations on the two held-out antigens. The mixed-task fine-tuned model evaluated on antigens it never saw during training (red) matches or exceeds the in-distribution fine-tuned model (teal) and clearly beats base Qwen3.5-9B with and without chain-of-thought (dashed).

D.6 Task-level Out-of-distribution Generalisation

The held-out-antigen study of Section D.5 still keeps antibody design inside the training mixture; it only withholds particular antigens. A more demanding question is whether the distilled behaviour survives when the entire protein task is removed. To answer it we fine-tune a chain-of-thought model on the `nanoGPT` and small-molecule trajectories, so that it never encounters an antibody, an Absolut binding landscape, or a single CDRH3 sequence, and then drop it, unchanged, into the antibody acquisition loop across all five antigens. Whatever competence the model shows here cannot be protein knowledge it does not possess; it can only be a task-agnostic acquisition policy carried over from the other two domains.

The outcome, in Figure 26, is striking. On four of the five targets this model performs on par with the *in-distribution* model that was trained directly on these antigens, reaching -113.1 on `1FBI_X` (against -113.4), -110.3 on `1ADQ_A` (-109.4), -106.6 on `1OB1_C` (-105.5), and -104.2 on `1NSN_S` (-104.7). A model that has never seen an antibody thus matches one trained on the very antigens it is tested on. The single place it falls short is `1HOD_C`, where it reaches only -89.4, no better than the base proposer (-90.9) and well behind the in-distribution model (-94.6), and with noticeably higher variance across seeds.

Reading the chain-of-thought traces makes the reason clear. Inside the loop, the model plays two roles at once: it *selects* among candidates according to the acquisition signal, and it *authors* new candidates *de novo*. These two roles come apart precisely on `1HOD_C`, whose landscape is unusually shallow (its best attainable energy is around -95, against -104 to -113 for the other targets) and whose optimum lies in a narrow basin that can be reached only by deliberately designing a specific aromatic/hydrophobic motif. Faced with the same situation, the two models reason quite differently:

> *In-distribution model:* "the best sequences mix a hydrophobic core (L, V, I, F) with polar and aromatic residues... let me construct a sequence: start with Y for π-stacking, keep a hydrophobic core in the middle, and place an aromatic residue at the terminus...", yielding `YRMQDQPWSLQ`, a sequence that is absent from the candidate pool.
>
> *Task-level model:* "I need to pick one sequence from the candidate pool of more than a thousand... I notice that `HLFQSCVSGPL` is already in the pool (id 169); it carries suitable hydrophobic and polar residues and violates none of the constraints, so I will choose it."

Where the in-distribution model composes the required motif from scratch, the task-level model has never learned the sequence priors of the protein domain, and so retreats to picking the most promising candidate the pool already offers; on `1HOD_C` this ceiling coincides with the base model's. On the remaining four antigens the optima can be found by selection alone, so carrying over the acquisition policy is sufficient and no gap appears.

Taken together, the task-level experiment separates two abilities that ordinary fine-tuning leaves entangled. What our distillation transfers is the one we set out to capture, namely knowing how to *use* the acquisition signal to explore and exploit, and it does so across tasks that share nothing but that structure. What it does not, and should not be expected to, transfer is *de novo* domain design, which depends on in-domain data and returns the moment such data is available, as both the in-distribution and case-level results confirm. This clean separation, in which acquisition generalises while domain-specific design does not, accounts for exactly where the task-level model succeeds and where it does not, and shows that the recipe distils precisely what ought to be transferable.

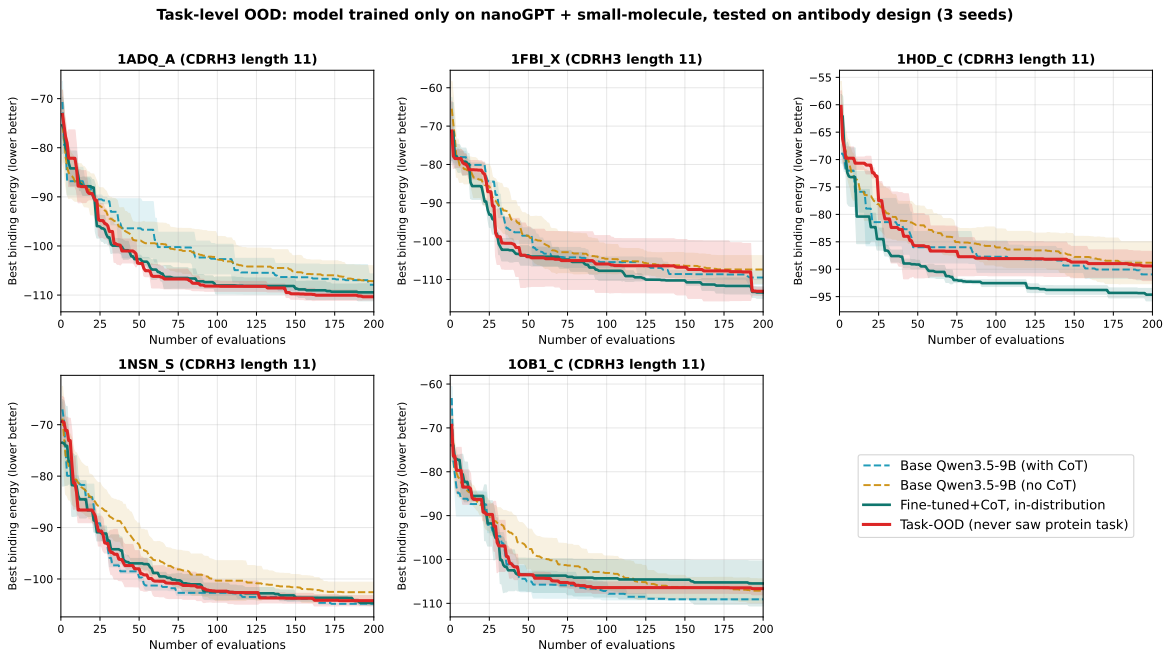


Figure 26: **Task-level out-of-distribution generalisation.** Best Absolut binding energy versus number of evaluations on all five antibody targets. The task-level model (red) was fine-tuned on *only* the `nanoGPT` and small-molecule trajectories and never saw the protein task, yet inside the antibody acquisition loop it matches or exceeds the in-distribution fine-tuned model (teal) on four targets (`1FBI_X`, `1ADQ_A`, `1OB1_C`, `1NSN_S`) and clearly beats base Qwen3.5-9B with and without chain-of-thought (dashed). The exception is `1H0D_C`, whose narrow optimum requires *de novo* motif design rather than candidate selection, the one capability that needs in-domain knowledge.