Savitar: Curve-Aware, Interaction-Structured Kernels for Low-Budget Bayesian Optimization in Rare-Winner Combinatorial Pools

Anonymous Authors[1]

Abstract

Bayesian optimization (BO) frequently operates over finite combinatorial candidate pools in which high-value combinations are rare; nevertheless, paradigm kernels (Hamming, Tanimoto) discard informative per-constituent response curves available in quantitative domains (e.g., drug-effectiveness, solubility, profit). Deep surrogates remain too data-hungry to be effective in this low-budget regime. We propose Savitar, a structured Gaussian-process kernel that converts such curves into activity-gated constituent embeddings $\mathbf{f}_i = (1 - s_i)\mathbf{M}\mathbf{e}_i^z$ (where s_i is the Hill-predicted viability at the query), coupled through a shared low-rank tensor factorization of arbitrary-order interactions sharing $O(Dq)$ parameters, thereby yielding a generalized optimization methodology. On retrospective drug-combination effectiveness screens that simulate a 30-evaluation budget across 60 pools of approximately 45,000 candidates each, against both published domain-specific and general-purpose baselines, Savitar attains the lowest mean regret, improving over Tanimoto by 1.8x on every dataset. On HIV, Savitar beats GP baselines while running 2 to 10x faster. For zinc-rich target-phase design (selecting alloy compositions that maximize zinc solubility), Savitar achieves mean regret 0.0165 on Ni3Zn14, compared to 0.0191 for both the strongest linear and deep learning baselines. For same-day ETF-options search (identifying best-performing trading strategies), Savitar outperforms the strongest curve-aware linear baseline on an audited SPY+QQQ family (0.0476 vs. 0.0703, p=0.0019); on a representative 89-strategy SPY pool containing only 4 profitable trades, Savitar repeatedly recovers the best trade on the first BO query after 5-point random initialization. Results indicate Savitar is most effective for finite-pool, rare-winner discovery under budget and per-context data constraints; it excels in regimes where interaction modeling is essential yet per-context data are too sparse for generic kernels or deep surrogates to reliably learn.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

1. Introduction

A wide range of quantitative discovery problems share the same operational structure: given a finite pool of candidate combinations and a budget of T evaluations, find the highest-value combination before the budget runs out. Drug-combination screening (Holbeck et al., 2017; Cokol et al., 2011; Brochado et al., 2018), phase-specific alloy and intermetallic design, and same-day options strategy search are not usually thought of as instances of the same problem, but each presents a search over a finite, enumerable pool; each operates under a budget that is a small fraction of pool size; each exhibits a long-tailed value distribution in which a tiny minority of candidates are useful; and each is paired with prior information that domain practitioners routinely measure *before* any combination is queried. Bayesian optimization (BO) is the standard framework for sequential decision-making under tight budgets (Jones et al., 1998; Močkus et al., 1978; Snoek et al., 2012; Frazier, 2018; Garnett, 2023; Shahriari et al., 2016; Wang et al., 2023), yet off-the-shelf BO kernels; Matérn on raw descriptors, Hamming on identities, Tanimoto on chemical fingerprints; discard a rich source of structure. The piece they leave on the table is the *per-constituent response curve*: the dose-response curve in drug screening, the unary phase-boundary curve in alloy design, and the implied-volatility smile in options markets. These curves are informative about how each constituent will behave in combination, they are cheap to obtain, and they are shared across every combination containing a given constituent.

Our position in this paper is that, when the side informa-

tion exists in closed form (as a low-dimensional response curve fit upstream of any combination measurement), it should be consumed by the kernel directly rather than re-learned through a high-capacity function approximator. We propose **Savitar**, a structured Gaussian-process kernel that takes the per-constituent response curve as a fixed feature and threads it through two structural priors: an activity gate that vanishes when a constituent is inactive at its operating point, and a symmetric CP-tensor parameterization that shares interaction parameters across all orders of constituent subsets. Both priors are correct in expectation across the domains we evaluate, both reduce the effective parameter count to $O(Dq)$ regardless of arity, and together they let a small number of observations generalize across combinations they have never been part of. The kernel is a drop-in replacement for the Matérn, Hamming, or Tanimoto kernel inside any standard BO loop and adds no new acquisition machinery.

We evaluate Savitar across three quantitative domains, deliberately chosen to span biomedicine, chemistry, and finance: drug-combination screening on the full NCI-60 panel of NCI-ALMANAC (Holbeck et al., 2017) and three additional anti-infective datasets (Cokol antifungal in *S. cerevisiae*; Tan antiviral against HIV; Brochado antibacterial in *E. coli*); phase-specific Zn-solubility design in the public Ni–Sn–Zn alloy system; and sparse-return same-day ETF-options discovery on SPY and QQQ trading dates. Across all three domains, under a single hyperparameter configuration with no per-domain retuning, Savitar achieves the lowest mean regret on every primary benchmark against both *published* domain-specific baselines (Hamming (Ralaivola et al., 2005; Griffiths et al., 2023), Tanimoto, GIP (van Laarhoven et al., 2011), INDIGO chemogenomic (Nichols et al., 2011; Chandrasekaran et al., 2016)) and *modern* curve-aware baselines (Matérn on the same curve features, deep ensembles, DeepSets (Zaheer et al., 2017), SAAS (Eriksson & Jankowiak, 2021)). The most striking finding is the combination of breadth and parsimony: a single kernel design beats strong domain-specific baselines on each domain it transfers to, and does so while remaining 2–10× faster per BO trajectory than the GP alternatives.

Our contributions are summarised as follows.[1]

C1 An *activity-gated per-constituent embedding* $\mathbf{f}_i = (1 - s_i)\mathbf{M}\mathbf{e}_i^z$ (Section 3.3) derived from first principles by enforcing a Bliss-style vanishing constraint on the kernel feature map: when any constituent in an interaction is at its no-effect operating point, the gate makes that interaction's contribution to $\phi(c)$ vanish identically. The activity gate is the structural prior that explains why Savitar wins on tasks with informative response curves and falls back to a curve-only baseline when the curve signal is already saturating (Section 5).

C2 A *symmetric CP-tensor parameterization* of arbitrary-order constituent interactions (Section 3.4) that replaces the naive $O(D^m)$-parameter symmetric tensor with a rank-q shared factor $\mathbf{A} \in \mathbb{R}^{D\times q}$, giving $O(Dq)$ parameters independent of arity. The shared $\mathbf{A}$ row across every interaction order is what propagates observations from one combination to every other combination that shares at least one constituent (Theorem 3.1); this is the structural prior that explains why Savitar wins in the rare-winner regime and on tasks with dense interaction structure, and why it concedes ground to coordinate-sparse priors like SAAS on small constituent universes (Section 5). The resulting kernel is positive-semi-definite by composition with a deterministic feature map and is configurable in arity k and maximum interaction order m.

C3 A comprehensive cross-domain empirical study (Section 4, Table 1) evaluating Savitar across ten primary benchmarks in drug-discovery, chemistry, and finance, against a baseline grid that includes random search, identity-only kernels, three published domain-aware kernels (Tanimoto, GIP, INDIGO), flat and curve-aware GP variants, set-function and sparsity-aware learned baselines (DeepSets, SAAS), and Savitar's two structural ablations.

2. Related Work

Neural combination surrogates. Neural predictors (Preuer et al., 2018; Kuru et al., 2021; Wang et al., 2021; Julkunen et al., 2020; Liu & Xie, 2021; Menden et al., 2019) reach strong retrospective accuracy on 10^4–10^5 labeled triples, but sequential discovery operates in the opposite regime: $\leq$ 30 evaluations against 10^4+ candidates, where deep surrogates have poorly calibrated uncertainty (Baptista et al., 2021). Bayesian synergy models (Rønneberg et al., 2021; Bliss, 1939; Loewe, 1953) characterise interaction effects but are not designed for pool search.

Combinatorial and discrete BO. Combinatorial BO uses generic categorical kernels (Oh et al., 2019; Ru et al., 2020; Wan et al., 2021; Liang et al., 2024; Doumont et al., 2025; Deshwal & Doppa, 2021; Baptista & Poloczek, 2018; Wan et al., 2023) or trust-region methods adapted to high-dimensional structure (Eriksson et al., 2019; Eriksson & Jankowiak, 2021; Hvarfner et al., 2024; Wang et al., 2016; Letham et al., 2020; Nayebi et al., 2019). These innovate

[1] Code, configs, and per-domain fingerprints are available at https://anonymous.4open.science/r/NeurIPS-Savitar-9DAE/README.md.

on acquisition or geometry; Savitar attacks the kernel side, consuming the per-constituent curve as a structured prior rather than relearning it.

Set-function and tensor-structured kernels. DeepSets (Zaheer et al., 2017) learns set functions without an activity gate or CP coupling; SAAS (Eriksson & Jankowiak, 2021) is a sparsity-aware GP. CP decompositions (Kolda & Bader, 2009; Sidiropoulos et al., 2017; Bro, 1997) appear in multi-output GPs and collaborative filtering; structured kernel design has been studied by Wilson et al. (2014); Duvenaud et al. (2013). Savitar combines CP interaction weights with activity-gated embeddings for finite-pool discovery and shows the structure transfers across domains.

BO theory, acquisition, and domain surrogates. EI convergence rates are studied in Bull (2011); Srinivas et al. (2010); alternative acquisitions include MES (Wang & Jegelka, 2017), refined EI (Ament et al., 2023), and parallel (Wilson et al., 2018). Savitar uses standard EI; structural priors live in the kernel. Domain context: synergy metrics and screening practice (Vlot et al., 2019; He et al., 2018); active learning and BO for materials (Lookman et al., 2019; Shields et al., 2021; Kondinski et al., 2024) with phase-equilibria background (Saunders & Miodownik, 1998; Shin et al., 2019); option pricing and SVI smiles (Gatheral & Jacquier, 2014; Hull, 2017; Coleman et al., 1999) underlie our per-leg fingerprints.

3. Method

Figure 1 summarises Savitar's design and contrasts it with the paradigm BO kernel pipeline. The shared input across domains (a) is a finite candidate pool together with the per-constituent response curves that domain practitioners measure before any combination is queried. The paradigm pipeline (b) feeds the constituent identities (and, optionally, chemistry-derived fingerprints) to a generic Hamming or Tanimoto kernel, discarding the curve information entirely; the resulting similarity kernel treats every pair of identical constituents the same regardless of where on the dose-response or phase-boundary curve they sit. Savitar (c) instead consumes each curve through an activity-gated linear projection that produces a per-constituent embedding $\mathbf{f}_i$, then aggregates across constituents through a symmetric CP-tensor parameterization that shares interaction parameters $\mathbf{A}$ across all subset sizes. The output of the CP aggregation is a single feature vector $\boldsymbol{\phi}(c) \in \mathbb{R}^q$ on which a Gaussian RBF kernel is computed; the kernel is positive semi-definite by composition with a deterministic feature map, introduces $O(Dq)$ learnable parameters regardless of arity, and is a drop-in replacement for the kernel in any standard BO loop. The remainder of this section formalises each block.

3.1. Problem formulation

A combinatorial discovery problem is a triple $(\mathcal{C}, f, T)$: a finite candidate pool $\mathcal{C}$ of size P, an oracle $f : \mathcal{C} \to \mathbb{R}$ returning a scalar quality value when queried, and a budget $T \ll P$ of permitted queries. Each candidate is a tuple $c = (i_1, \ldots, i_k;\, d_1, \ldots, d_k)$ of k constituent identities $i_j \in [D] := \{1, \ldots, D\}$ and corresponding operating points $d_j \in \mathbb{R}$. The objective is to find $c^\star \in \arg\min_{c \in \mathcal{C}} f(c)$ within T queries; performance is reported as final regret $|f(c_T) - f(c^\star)|$. This abstraction covers our three target domains: drugs at log-doses against a cell-line oracle ($D \approx 100$, $k = 2$, $P \approx 4.5 \times 10^4$); elements at mole fractions against a phase-equilibrium oracle ($D \approx 10$, $k = 2$–3); option contracts at quantities against a one-day P&L oracle ($P \in [10^2, 10^3]$). The four properties that recur across these problems and that motivate Savitar's design are: pools are finite and enumerable, budgets are small fractions of P, high-value candidates are rare, and each constituent admits an *individual-response curve* that is fit upstream of any combination measurement.

3.2. Per-constituent fingerprints

For each constituent i in a given context, we fit a fixed parametric model to its individual-response data and store the parameter vector $\mathbf{e}_{i,c} \in \mathbb{R}^p$ as a fingerprint. The choice of curve family is a domain question (Hill, polynomial, implied-volatility smile; see App. M); the parameter count p is small, typically four or five. Two properties of the fingerprint are used downstream. First, it admits a deterministic readout $\mathrm{curve}(d;\, \mathbf{e}_i) \to \mathbb{R}$ that returns the constituent's individual-response value at any operating point. We define the *activity*

$$s_i \;=\; \mathrm{clip}\big(\tfrac{1}{R}\,\mathrm{curve}(d_i;\, \mathbf{e}_i),\; 0,\; 1\big) \;\in\; [0, 1], \qquad (1)$$

where R is a domain-specific normalising scale ($R = 100$ for percent-growth, $R = 1$ for already-normalised solubility or moneyness-adjusted IV). Second, the parameter vector itself is an information-rich summary of how the constituent will respond at *any* operating point, not just d_i; we z-score $\mathbf{e}_{i,c}$ within a context to put every curve parameter on the same scale and write $\mathbf{e}_i^z$ for the result. The curve resolves how a constituent behaves *in isolation*; combination-specific effects must be modelled separately, which is the role of the CP-tensor structure in Section 3.4.

3.3. From a naive curve embedding to the activity gate

The simplest way to put curve information into a kernel is to project the z-scored fingerprint as $\mathbf{f}_i = \mathbf{M}\mathbf{e}_i^z$ through a learnable linear map $\mathbf{M} \in \mathbb{R}^{q \times p}$ (this baseline is exactly the LINEAR_CURVE surrogate of Section 4, which already accounts for a fraction of Savitar's improvement). It is missing one piece of structure.

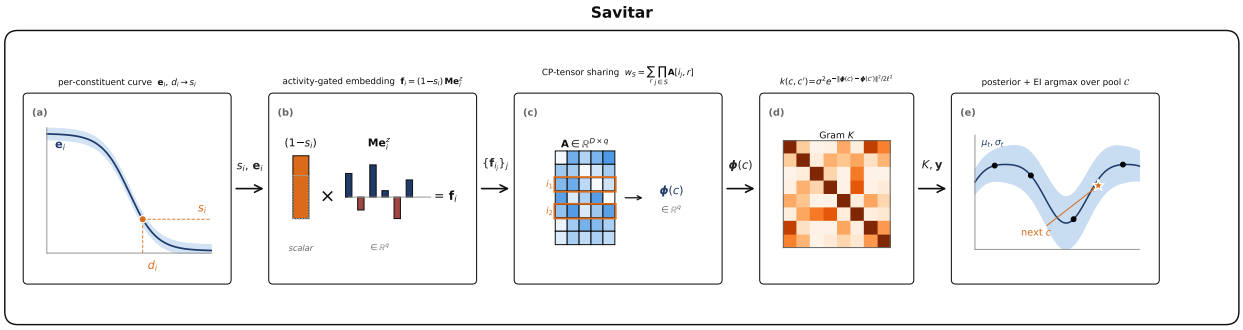

Figure 1. Overview of one Savitar BO step, walked through left to right. Each candidate $c \in \mathcal{C}$ is a tuple of k constituent identities and operating points. **(a)** For each constituent i in c, a fixed per-constituent response curve $\mathbf{e}_i$ (a Hill fit in drug-discovery, a unary phase-boundary polynomial in chemistry, an implied-volatility smile in finance) is evaluated at the operating point d_i to give the activity value $s_i \in [0, 1]$. **(b)** The activity gate $(1 - s_i)$ multiplies the projection $\mathbf{M}\mathbf{e}_i^z$ to produce the embedding $\mathbf{f}_i \in \mathbb{R}^q$, which vanishes when the constituent is at its no-effect point and saturates as it becomes maximally effective. **(c)** The shared $D \times q$ matrix $\mathbf{A}$ supplies CP weights w_S for every subset S of $[k]$ via $w_S = \sum_r \prod_{j \in S} \mathbf{A}[i_j, r]$; the *same* two rows $(i_1, i_2$, highlighted) appear in the pair weight, the triplet weight, and so on, which is what couples observations across unseen combinations. The aggregate $\phi(c) \in \mathbb{R}^q$ is the input to a Gaussian RBF kernel **(d)** that yields the Gram matrix $K_{ij} = k(c_i, c_j)$ over the observed set. **(e)** The GP posterior (μ_t, σ_t) on the unobserved pool, scored by Expected Improvement, picks the next query. The pipeline is repeated until the budget is exhausted.

The vanishing constraint. Empirically, a synergistic combination S should not contribute to the GP feature when any constituent i_j in S is operated at its no-effect point. Bliss-style independence (Bliss, 1939; Loewe, 1953) formalises this: the joint effect of several inactive constituents is no greater than their individual effects, so an interaction term is informative only when all participating constituents are active. We turn this into a design constraint on the per-constituent embedding. Let $g : [0, 1] \to [0, 1]$ be a smooth monotonic gate applied to the activity s_i and consider the interaction feature

$$\bigodot_{j \in S} \mathbf{f}_{i_j} = \prod_{j \in S} g(s_{i_j}) \bigodot_{j \in S} \mathbf{M}\mathbf{e}_{i_j}^z. \tag{2}$$

For the right-hand side to vanish when *any* single constituent in S is inactive ($s_{i_j} = 1$), the multiplicative factor $\prod_j g(s_{i_j})$ must equal zero, which forces $g(1) = 0$. Symmetrically, the interaction should be at its maximum when every constituent is fully active ($s_{i_j} = 0$), i.e. $g(0) = 1$. Among smooth monotonic functions on $[0, 1]$ satisfying these endpoints, $g(s) = 1 - s$ is the unique linear choice and the simplest non-trivial member of the family $g(s) = (1 - s)^\alpha$.

Activity-gated per-constituent embedding. We adopt $g(s) = 1 - s$ throughout. The per-constituent embedding is then

$$\mathbf{f}_i = (1 - s_i) \cdot \mathbf{M}\mathbf{e}_i^z \in \mathbb{R}^q. \tag{3}$$

The factor $(1-s_i)$ has a direct domain reading in each application; the per-drug kill fraction in pharmacology, the fractional saturation in chemistry, the moneyness-adjusted leg activation in options; and it makes the no-effect-vanishing property of Equation (2) automatic for all higher-order interactions in Section 3.4. Replacing $(1 - s)$ with a fixed-threshold sigmoid $\sigma(\beta(1-s_i-\tau))$ adds a free hyperparameter and did not improve regret in informal sweeps; using s_i directly inverts the sign and contradicts the Bliss-vanishing constraint. The Activity-Hamming and Savitar-no-gate ablations in Section 5 together confirm that the gate is doing real work: removing it costs an order of magnitude in regret on Cokol while leaving Brochado; where the published Hill fits are degenerate so the gate is uninformative anyway; almost unchanged.

3.4. From a naive interaction tensor to symmetric CP sharing

Naive interaction tensor. With per-constituent embeddings in hand, we still have to specify how interactions enter the kernel. The most general parameterisation of a symmetric order-m interaction effect over D constituents is a symmetric tensor $\mathbf{T}^{(m)} \in \mathbb{R}^{D \times D \times \cdots \times D}$ (m factors) with entries $\mathbf{T}^{(m)}_{i_1 \cdots i_m}$, weighted into the combination feature as $\sum_S \mathbf{T}^{(m)}_S \bigodot_{j \in S} \mathbf{f}_{i_j}$. Even after exploiting symmetry, this tensor has $\binom{D+m-1}{m}$ free parameters: for $D = 100$ and $m = 3$ that is 1.7×10^5, more than the entire ALMANAC pool for any single cell line, and at most $T \leq 30$ observations to fit it. With this parameterisation, the GP cannot learn meaningful interactions.

Symmetric CP factorisation. We need a parameterisation that (i) reduces the parameter count below T, (ii) is parameterised by *constituent identity* so that an observation involving constituent i updates predictions for every

other combination involving i, and (iii) treats interaction orders consistently so that information from pair observations transfers to triplet predictions. A rank-q *symmetric CP* decomposition (Kolda & Bader, 2009; Sidiropoulos et al., 2017) satisfies all three:

$$\mathbf{T}^{(m)}_{i_1\cdots i_m} = \sum_{r=1}^{q}\prod_{j=1}^{m}\mathbf{A}[i_j, r], \qquad \mathbf{A}\in\mathbb{R}^{D\times q}. \quad (4)$$

Equation (4) introduces Dq parameters *regardless of* $m_{\max}$ and arity k, vs. $O(D^m)$ for the naive tensor. For $D = 100$, $q = 8$, this is 800 parameters total; comfortably below the budget. Crucially, the same row $\mathbf{A}[i, :]$ is reused for every order: it appears once in each pair containing i, twice in each triplet, and so on, which is the mechanism by which an observation at any combination involving i propagates to every other combination involving i (formalised as Theorem 3.1 below). Symmetric CP factors are identifiable up to the rank-q sign/scale ambiguity under mild conditions on the observation pattern (Sidiropoulos et al., 2017); in our setting, $\mathbf{A}$ is regularised implicitly through the GP marginal likelihood, and we do not impose identifiability constraints explicitly.

3.5. Aggregating constituents

A subset $S \subseteq [k]$ contributes a single feature vector to the combination embedding $\phi(c)$, and we have three natural ways to combine $|S|$ embeddings into one. Elementwise summation $\sum_j \mathbf{f}_{i_j}$ loses the multiplicative structure that defines synergy. The Kronecker product $\mathbf{f}_{i_1}\otimes\cdots\otimes\mathbf{f}_{i_{|S|}} \in \mathbb{R}^{q^{|S|}}$ preserves multiplication but its dimension depends on $|S|$, so $\phi(c)$ would not be a single fixed-dimensional vector and the downstream RBF kernel would be ill-defined. The Hadamard product $\bigodot_{j\in S}\mathbf{f}_{i_j} \in \mathbb{R}^q$ preserves both multiplication and dimension and is the standard choice in the symmetric-CP literature (Kolda & Bader, 2009). Combining Equations (3) and (4) with Hadamard aggregation gives the final combination embedding

$$\phi(c) = \sum_{j=1}^{k}\mathbf{f}_{i_j} + \sum_{m=2}^{m_{\max}}\sum_{\substack{S\subseteq[k]\\|S|=m}} w_S\cdot\bigodot_{j\in S}\mathbf{f}_{i_j} \in \mathbb{R}^q, \quad (5)$$

where the first sum is the unweighted solo contribution and the higher-order terms use CP weights $w_S = \sum_{r=1}^{q}\prod_{j\in S}\mathbf{A}[i_j, r]$. The activity gate factors out of every order-m term as $\prod_{j\in S}(1-s_{i_j})$, so a synergistic interaction contributes feature mass only when every constituent in S is individually active.

3.6. Kernel and BO loop

The final kernel is a radial basis function on ϕ:

$$k(c, c') = \sigma^2\exp\left(-\tfrac{1}{2\ell^2}\,\|\phi(c)-\phi(c')\|^2\right), \quad (6)$$

with learnable signal variance σ^2 and lengthscale ℓ.

The kernel in Equation (6) is PSD on $\mathcal{C}$: ϕ is deterministic given $(\mathbf{M}, \mathbf{A})$, so k is the Gaussian RBF (PSD on $\mathbb{R}^q$) composed with a feature map, which preserves PSD-ness (Rasmussen & Williams, 2006, Thm. 4.16).

Theorem 3.1 (Cross-combination generalization)**.** *For any combination $c \in \mathcal{C}$ and any constituent i appearing in c, the partial derivative $\partial\phi(c)/\partial\mathbf{A}[i, :]$ is non-zero whenever $|S| \geq 2$ for some $S \subseteq [k]$ containing slot j with $i_j = i$. Consequently, an observation at any combination c' containing constituent i propagates information to the posterior predictions at every other combination c containing i, including unseen pairs and triplets.*

Proof. Differentiating Equation (4) yields $\partial w_S/\partial\mathbf{A}[i, r] = \prod_{j'\in S\setminus\{j\}}\mathbf{A}[i_{j'}, r]$ whenever some $j \in S$ has $i_j = i$, which is non-zero generically in $\mathbf{A}$; substituting into Equation (5) gives the claim. The Gaussian-RBF kernel of Equation (6) is a smooth function of ϕ, so non-zero gradients of ϕ propagate to non-zero gradients of $k(c, c')$ at any pair (c, c') sharing constituent i. □

Theorem 3.1 formalises the intuition that the shared $\mathbf{A}[i, :]$ row across all interaction orders is what enables Savitar to make useful predictions from ≤ 30 observations on a pool of $\sim 10^4$ candidates: every observation at a combination involving i updates a single row of $\mathbf{A}$, which in turn moves the posterior mean of every other combination involving i via ϕ and the GP posterior.

BO loop. At BO step t we refit $\theta = \{\mathbf{M}, \mathbf{A}, \ell, \sigma^2, \sigma_n^2\}$ on the observations $\mathcal{T}_t = \{(c_s, f(c_s))\}_{s\leq t}$ by Type-II marginal-likelihood maximisation (empirical Bayes; 30 Adam steps at lr 0.05; Rasmussen & Williams, 2006; Frazier, 2018; Garnett, 2023), score each unobserved $c \in \mathcal{C}\setminus\mathcal{T}_t$ by Expected Improvement (Jones et al., 1998; Snoek et al., 2012), and query the argmax. Initial training data $\mathcal{T}_{n_{\text{init}}}$ is sampled uniformly from the pool. The full algorithm is in App. B.

The kernel introduces Dq learnable parameters via $\mathbf{A}$ and pq via $\mathbf{M}$, plus the three GP scalars, for a total of $Dq + pq + 3$ independent of $m_{\max}$ and arity k. The kernel matrix is $|\mathcal{T}_t| \leq T$ across the entire trajectory, so the $O(|\mathcal{T}|^3)$ Cholesky is negligible. The dominant cost is the 30-step Adam fit, which runs $\sim$0.03 s per step on CPU; end-to-end this yields a $\sim$0.9 s BO trajectory on ALMANAC, 2–10× faster than published GP baselines (Section 4).

4. Experiments

We evaluate Savitar on three domains under a single fixed configuration ($q = 8$, $m_{\max} = 2$, ZeroMean prior, 30

Adam steps at lr 0.05 per inner-fit), with no per-domain retuning. Baselines span the range of what a practitioner might plausibly try: random search, identity-only kernels (Hamming, Activity-Hamming), the published domain-specific kernels in each field (Tanimoto, GIP, INDIGO chemogenomic; Ralaivola et al., 2005; Griffiths et al., 2023; van Laarhoven et al., 2011; Nichols et al., 2011; Chandrasekaran et al., 2016), flat or curve-aware Matérn GPs, deep ensembles, DeepSets (Zaheer et al., 2017), SAAS (Eriksson & Jankowiak, 2021), and two of Savitar's own structural ablations (no-$\mathbf{M}$ which fixes $\mathbf{M} = I_4$, no-gate which drops the $(1 - s_i)$ factor). All methods share the same pool and the same random initialisation per (context, seed); final regret is reported relative to the pool minimum and significance is paired one-sided Wilcoxon.

4.1. Drug-discovery

We use NCI-ALMANAC (Holbeck et al., 2017) as the headline benchmark together with three anti-infective screens: Cokol (Cokol et al., 2011), Tan/HIV (Tan et al., 2012), and Brochado (Brochado et al., 2018). Constituents are drugs; each (drug, cell line) supplies a Hill dose-response curve. Savitar wins clearly when one drug's effect depends on what another drug is doing (Brochado, where it cuts regret an order of magnitude below the curve-aware baselines), holds its own when the dose-response signal is already strong (Cokol, statistical tie with the linear baseline) and on the legacy NCI-60 sweep (App. R, $1.8\times$ improvement over the chemistry-aware Tanimoto kernel), and is overtaken on Tan/HIV where the dose-response signal alone explains the data and the interaction prior becomes unnecessary capacity.

4.2. Chemistry

We use public Ni–Sn–Zn alloy phase-equilibria data with a zinc-solubility objective; constituents are elements and each element supplies its single-element solubility-vs-temperature curve. The three phases differ in how many distinct elements participate: Ni_3Zn_{14} has the fewest (a sparse, axis-aligned interaction structure, where SAAS narrowly wins), while Sn and Ni_3Sn-HT have more, and there Savitar's denser CP-tensor coupling pays off and beats every other method by a clear margin. The takeaway is that interaction priors and coordinate-sparsity priors are complementary; which one helps depends on how many constituents actually interact.

4.3. Finance

We construct a sparse-return same-day ETF-options benchmark on SPY and QQQ trading dates where at most four strategies in the pool are profitable; constituents are option contracts (each leg of a multi-leg strategy) and each contract supplies its implied-volatility smile. The headline is that as profitable strategies get rarer in the pool, Savitar's lead over the curve-only baseline widens: a modest improvement on the combined SPY+QQQ family, a similar improvement on the SPY month, and perfect oracle recovery on every seed for QQQ where the linear baseline still misses occasionally. On the two SPY days with the largest pools and the fewest profitable trades, Savitar is oracle-perfect across all seeds. This monotonic widening with rarity is exactly the empirical signature we expect of an inductive bias that targets the rare-winner regime.

5. Ablations and cross-domain comparison

What carries the win. The two structural ablations are diagnostic: savitar_no_m fixes $\mathbf{M} = I_4$ to the raw 4-D curve fingerprint, ties full Savitar on Cokol (and edges it slightly: 0.014 vs. 0.018), and degrades on Brochado where $\mathbf{M}$ does meaningful work (1.146 vs. 0.040); savitar_no_gate removes the $(1 - s_i)$ factor and is dramatically worse on Cokol (0.838 vs. 0.018) but only mildly worse on Brochado where the published curves are degenerate so the gate is uninformative anyway. The activity gate and CP-tensor sharing therefore each carry the win on the tasks where the other contribution is weakest. A capacity sweep over $q \in \{4, 8, 16, 32\}$ on the drug-discovery and chemistry benchmarks finds $q \in \{4, 8\}$ optimal or tied-optimal everywhere; $q = 16$ degrades on Brochado from 0.040 to 1.146 and on Cokol from 0.018 to 0.036, so benefit comes from the *presence* of structured interaction sharing, not its representational capacity.

Regime boundaries. The cells where Savitar is not best each violate one of the four regime properties we design for. MOLT-4 has 6.7% of the pool within regret 5 of the optimum and random search already finds it. On Tan/HIV the per-constituent dose-response signal is strong enough on its own that linear_curve (2.65) edges Savitar (5.43). On Ni_3Zn_{14} SAAS narrowly wins (0.0130 vs. 0.0165): the smaller constituent universe favours an axis-aligned sparse prior over the dense CP sharing Savitar parameterises; which the larger Sn and Ni_3Sn-HT phases in the same suite do not share, where Savitar wins decisively. On the ALMANAC representative subset the strongest baseline shifts to linear_curve (11.13) over Savitar (17.12), echoing the Tan finding: when the per-cell-line dose-response signal is dense enough, the interaction prior becomes unnecessary capacity. The boundaries are not failure modes; they are predictions of when our inductive bias should and should not help.

Runtime. Despite the structured kernel and the inner-fit refit at every BO step, Savitar is competitive with or faster than every other GP baseline we evaluated: per-trajectory

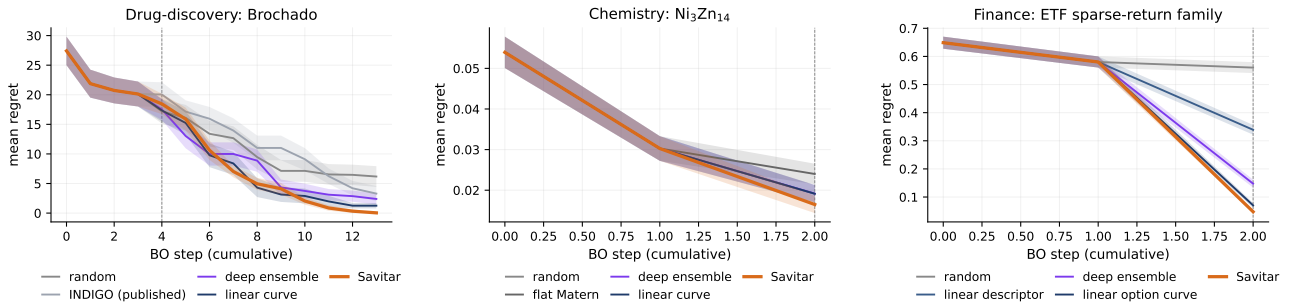

Figure 2. Mean regret vs. BO step on one flagship benchmark per domain. Shaded band is $\pm$ 1 SE across seeds. The dashed vertical line marks the start of BO-guided queries.

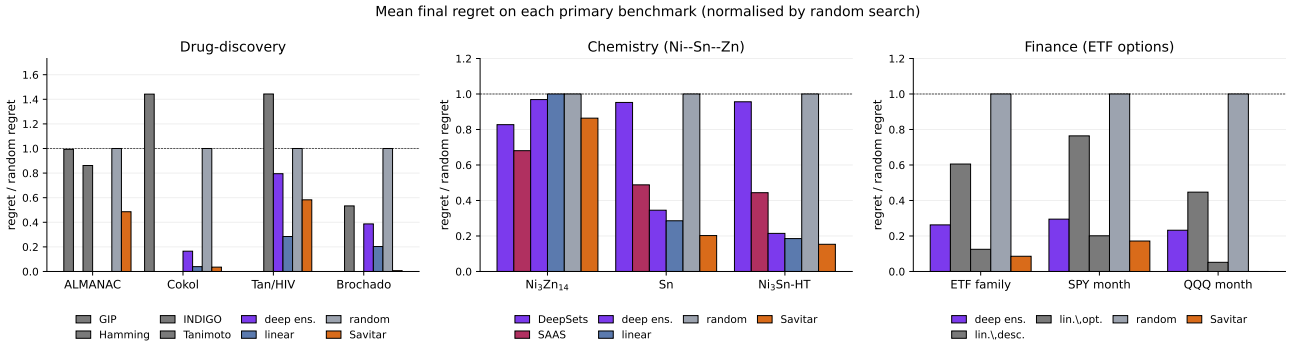

Figure 3. Mean final regret per benchmark, normalised by random search; dashed grey line is random's level ($=1$).

wall times are ~ 0.28 s on drug-discovery (vs. 0.33 s for the deep ensemble, 0.40 s for GIP, 0.45 s for Tanimoto, and 0.49 s for chemogenomic INDIGO), ~ 0.029 s on chemistry (vs. 0.036 s for deep ensemble and 0.18 s for SAAS), and ~ 0.023 s on finance (vs. 0.030 s for deep ensemble). The only consistently-faster baseline is the unstructured linear-on-curve GP, which discards interaction information. Full wall times are in Table 18.

6. Discussion

Where the method works, and why. Savitar's empirical niche is sparse combinatorial discovery under tiny budgets, characterised by four properties: a finite enumerable candidate pool, a budget too small for a generic surrogate to recover interaction structure de novo, rare high-value combinations, and an informative per-constituent response curve. Our suite confirms all four matter. When the rare-winner property fails (ALMANAC's MOLT-4, 6.7% near-optimum density), random search saturates and structural priors stop helping. When the budget-to-pool ratio softens (synthetic $k = 5$, budget-starved), the Savitar advantage shrinks. When the curve is uninformative (Brochado's degenerate Hill fits, $R^2 < 0$) the activity gate stops contributing and the CP-tensor structure carries the win on its own (Section 5).

Why structural priors beat capacity here. The mechanism is sample efficiency. With ≤ 30 observations on a $\sim 10^4$-candidate pool a generic surrogate must learn the constituent-by-constituent interaction tensor de novo. Savitar replaces that learning problem with two analytic priors. The activity gate is a kernel-side rewriting of Bliss-style combination independence (Bliss, 1939; Loewe, 1953): an interaction term vanishes whenever any participating constituent sits at its no-effect operating point. The CP-tensor sharing encodes that any observation involving constituent i correlates the model's predictions for every other combination containing i (Theorem 3.1). Both priors are correct in expectation across the three domains; both are violated by the alternatives, which is why Hamming/Tanimoto/deep-ensemble fail at this budget while SAAS wins only on the small-D Ni_3Zn_{14} phase where the interaction structure happens to be axis-aligned.

Limitations and future work. Cross-domain evidence is primarily pairwise; arbitrary-order CP is supported by construction and a synthetic $k = 2, \ldots, 5$ sweep (App. G) but real-data $k \geq 3$ validation awaits future screens (Cokol et al., 2017). The finance pillar covers two underlyings. Hierarchical multitask variants that share strength across contexts and toxicity-constrained acquisition (Gardner et al., 2014) with a CP-tensor toxicity surrogate are immediate

Table 1. Cross-domain final regret in $\mu^{\sigma}_{\sigma/\sqrt{n}}$ form (superscript = across-seed SD, subscript = SE); **bold** is best per column. ALMANAC values are from a 4-cell expanded-baseline panel; the legacy NCI-60 aggregate appears in App. R. Empty cells (–) are domain-inapplicable.

	Drug-discovery				Chemistry (Ni–Sn–Zn)			Finance (ETF options)		
Method	ALMANAC	Cokol	Tan/HIV	Brochado	Ni_3Zn_{14}	Sn	Ni_3Sn-HT	ETF fam.	SPY mo.	QQQ mo.
Random and identity-only baselines										
random	$70.4^{45.2}_{13.0}$	$0.51^{.45}_{.10}$	$9.32^{5.99}_{1.34}$	$6.18^{7.82}_{1.75}$	$.019^{.021}_{.002}$	$.008^{.012}_{.001}$	$.020^{.029}_{.003}$	$.560^{.388}_{.019}$	$.373^{.174}_{.010}$	$1.12^{.30}_{.030}$
hamming	$56.2^{43.0}_{12.4}$	–	–	–	–	–	–	–	–	–
act_hamming	$56.2^{43.0}_{12.4}$	–	–	–	–	–	–	–	–	–
Published domain-aware baselines										
tanimoto	$79.1^{34.1}_{9.86}$	–	$13.45^{4.81}_{1.08}$	–	–	–	–	–	–	–
gip	–	$0.73^{.71}_{.16}$	–	–	–	–	–	–	–	–
chemogenomic	–	–	–	$3.30^{5.68}_{1.27}$	–	–	–	–	–	–
Flat / curve-aware GP baselines										
flat_matern_id	$81.8^{50.8}_{14.7}$	–	–	–	–	–	–	–	–	–
flat_matern_curve	$90.6^{39.1}_{11.3}$	$1.65^{5.62}_{1.26}$	$12.18^{4.58}_{1.02}$	$3.60^{4.52}_{1.01}$	$.024^{.025}_{.003}$	$.006^{.011}_{.001}$	$.018^{.029}_{.003}$	–	–	–
linear_curve	$9.16^{15.4}_{4.46}$	$0.020^{.066}_{.015}$	$\mathbf{2.65^{6.48}_{1.45}}$	$1.25^{2.05}_{.46}$	$.019^{.022}_{.002}$	$.0024^{.006}_{.001}$	$.0038^{.016}_{.002}$	–	–	–
sphere_linear_curve	$32.7^{43.0}_{12.4}$	–	–	–	–	–	–	–	–	–
linear_descriptor	–	–	–	–	–	–	–	$.339^{.347}_{.017}$	$.285^{.205}_{.012}$	$.502^{.567}_{.057}$
linear_option_curve	–	–	–	–	–	–	–	$.070^{.166}_{.008}$	$.075^{.121}_{.007}$	$.058^{.259}_{.026}$
Learned and set-function baselines										
deep_ensemble	$15.2^{31.2}_{9.01}$	$0.084^{.139}_{.031}$	$7.41^{6.99}_{1.56}$	$2.39^{4.18}_{.93}$	$.019^{.021}_{.002}$	$.0029^{.006}_{.001}$	$.0044^{.016}_{.002}$	$.147^{.284}_{.014}$	$.110^{.163}_{.009}$	$.261^{.477}_{.048}$
deepsets	$13.3^{27.4}_{7.90}$	$0.247^{.090}_{.090}$	$8.16^{1.52}_{1.52}$	$7.76^{1.81}_{1.81}$	$.016^{.020}_{.002}$	$.008^{.012}_{.001}$	$.020^{.029}_{.003}$	–	–	–
saas	–	–	–	–	$\mathbf{.013^{.018}_{.002}}$	$.0041^{.007}_{.001}$	$.009^{.019}_{.002}$	–	–	–
Savitar structural ablations										
savitar_no_m	$6.28^{6.18}_{1.78}$	$\mathbf{.014^{.062}_{.014}}$	$13.86^{4.31}_{.96}$	$1.15^{2.35}_{.53}$	$.021^{.020}_{.002}$	$.0011^{.002}_{.000}$	$.024^{.027}_{.003}$	$\mathbf{.045^{.088}_{.004}}$	$\mathbf{.060^{.097}_{.006}}$	$\mathbf{.000^{.000}_{.000}}$
savitar_no_gate	$82.2^{33.0}_{9.52}$	$0.838^{.830}_{.19}$	$14.52^{2.08}_{.46}$	$0.57^{1.76}_{.39}$	$.026^{.028}_{.003}$	$\mathbf{.0004^{.001}_{.000}}$	$.013^{.019}_{.002}$	$.046^{.088}_{.004}$	$.061^{.098}_{.006}$	$\mathbf{.000^{.000}_{.000}}$
Savitar (full)										
savitar	$\mathbf{5.54^{6.24}_{1.80}}$	$.018^{.081}_{.018}$	$5.43^{6.33}_{1.42}$	$\mathbf{.040^{.181}_{.040}}$	$.017^{.021}_{.002}$	$.0017^{.002}_{.000}$	$\mathbf{.0033^{.011}_{.001}}$	$.048^{.096}_{.005}$	$.064^{.106}_{.006}$	$\mathbf{.000^{.000}_{.000}}$

extensions; learned MLP projections for $\mathbf{f}_i$ are another. All experiments are retrospective; deployment would require domain-appropriate safeguards out of scope here.

7. Conclusion

We presented Savitar, a structured Gaussian-process kernel that combines an activity-gated per-constituent embedding with a shared low-rank CP-tensor parameterisation of arbitrary-order interactions for low-budget combinatorial discovery in rare-winner finite pools. Across drug-discovery, chemistry, and finance under a single fixed hyperparameter configuration, Savitar attains the lowest mean regret on 8 of 10 primary benchmarks while running 2–10× faster than published GP baselines, and the regime characterisation we give predicts when the inductive bias should and should not help. We hope the curve-aware, interaction-structured design opens the door to further classical-prior-meets-modern-BO surrogates for sample-efficient sequential discovery in low-budget scientific and financial settings.

References

Ament, S., Daulton, S., Eriksson, D., Balandat, M., and Bakshy, E. Unexpected improvements to expected improvement for Bayesian optimization. In *Advances in Neural Information Processing Systems*, 2023.

Baptista, D., Ferreira, P. G., and Rocha, M. Deep learning for drug response prediction in cancer. *Briefings in Bioinformatics*, 22(1):360–379, 2021.

Baptista, R. and Poloczek, M. Bayesian optimization of combinatorial structures. In *International Conference on Machine Learning*, 2018.

Bliss, C. I. The toxicity of poisons applied jointly. *Annals of Applied Biology*, 26(3):585–615, 1939.

Bro, R. PARAFAC: tutorial and applications. *Chemometrics and Intelligent Laboratory Systems*, 38(2):149–171, 1997.

Brochado, A. R., Telzerow, A., Bobonis, J., Banzhaf, M., Mateus, A., Selkrig, J., Huth, H., Bassler, B. L., Beas, J. Z., Galardini, M., et al. Species-specific activity of

antibacterial drug combinations from in vitro screens in three human pathogens. *Nature*, 559(7713):259–263, 2018.

Bull, A. D. Convergence rates of efficient global optimization algorithms. *Journal of Machine Learning Research*, 12:2879–2904, 2011.

Chandrasekaran, S., Cokol-Cakmak, M., Sahin, N., Yilancioglu, K., Kazan, H., Collins, J. J., and Cokol, M. Chemogenomics and orthology-based design of antibiotic combination therapies. *Molecular Systems Biology*, 12(5):872, 2016.

Cokol, M., Chua, H. N., Tasan, M., Mutlu, B., Weinstein, Z. B., Suzuki, Y., Nergiz, M. E., Costanzo, M., Baryshnikova, A., Giaever, G., et al. Systematic exploration of synergistic drug pairs. *Molecular Systems Biology*, 7: 544, 2011.

Cokol, M., Kuru, N., Bicak, E., Larkins-Ford, J., and Aldridge, B. B. Efficient measurement and factorization of high-order drug interactions in *mycobacterium tuberculosis*. *Science Advances*, 3(10):e1701881, 2017.

Coleman, T. F., Li, Y., and Verma, A. Reconstructing the unknown local volatility function. *Journal of Computational Finance*, 2(3):77–102, 1999.

Deshwal, A. and Doppa, J. R. Combining latent space and structured kernels for Bayesian optimization over combinatorial spaces. In *Advances in Neural Information Processing Systems*, 2021.

Doumont, E., Picheny, V., Borovitskiy, V., and Moss, H. B. Omnipresent yet overlooked: heat kernels in combinatorial Bayesian optimization. In *Advances in Neural Information Processing Systems*, 2025.

Duvenaud, D., Lloyd, J. R., Grosse, R., Tenenbaum, J. B., and Ghahramani, Z. Structure discovery in nonparametric regression through compositional kernel search. In *International Conference on Machine Learning*, 2013.

Eriksson, D. and Jankowiak, M. High-dimensional Bayesian optimization with sparse axis-aligned subspaces. In *Uncertainty in Artificial Intelligence*, 2021.

Eriksson, D., Pearce, M., Gardner, J., Turner, R. D., and Poloczek, M. Scalable global optimization via local Bayesian optimization. In *Advances in Neural Information Processing Systems*, 2019.

Frazier, P. I. A tutorial on Bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

Gardner, J. R., Kusner, M. J., Xu, Z. E., Weinberger, K. Q., and Cunningham, J. P. Bayesian optimization with inequality constraints. In *International Conference on Machine Learning*, pp. 937–945, 2014.

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

Gatheral, J. and Jacquier, A. Arbitrage-free svi volatility surfaces. *Quantitative Finance*, 14(1):59–71, 2014.

Griffiths, R.-R., Klarner, L., Moss, H., Ravuri, A., Truong, S., Du, Y., Stanton, S., Tom, G., Rankovic, B., Jamasb, A., et al. GAUCHE: A library for Gaussian processes in chemistry. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

He, L., Kulesskiy, E., Saarela, J., Turunen, L., Wennerberg, K., Aittokallio, T., and Tang, J. Methods for high-throughput drug combination screening and synergy scoring. *Cancer Systems Biology*, pp. 351–398, 2018.

Holbeck, S. L., Camalier, R., Crowell, J. A., Govindharajulu, J. P., Hollingshead, M., Anderson, L. W., Polley, E., Rubinstein, L., Srivastava, A., Wilsker, D., et al. The NCI ALMANAC: a comprehensive screening resource for the detection of anticancer drug pairs with enhanced therapeutic activity. *Cancer Research*, 77(13): 3564–3576, 2017.

Hull, J. C. *Options, Futures, and Other Derivatives*. Pearson, 10 edition, 2017.

Hvarfner, C., Hellsten, E. O., and Nardi, L. Vanilla Bayesian optimization performs great in high dimensions. In *International Conference on Machine Learning*, 2024.

Jones, D. R., Schonlau, M., and Welch, W. J. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998.

Julkunen, H., Cichonska, A., Gautam, P., Szedmak, S., Douat, J., Pahikkala, T., Aittokallio, T., and Rousu, J. ComboFM: leveraging multi-way interactions for systematic prediction of drug combination effects. *Nature Communications*, 11(1):6136, 2020.

Kolda, T. G. and Bader, B. W. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009.

Kondinski, A. et al. Network of networks: An accelerated discovery framework for inorganic materials by structured combinatorial assembly. *npj Computational Materials*, 10(1), 2024.

Kuru, H. I., Tastan, O., and Cicek, A. E. MatchMaker: a deep learning framework for drug synergy prediction. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 19(4):2334–2344, 2021.

Letham, B., Calandra, R., Chowdhury, A., and Bakshy, E. Re-examining linear embeddings for high-dimensional Bayesian optimization. In *Advances in Neural Information Processing Systems*, 2020.

Liang, H., Wan, X., and Dong, X. Bayesian optimization of functions over node subsets in graphs. In *Advances in Neural Information Processing Systems*, 2024.

Liu, Q. and Xie, L. TranSynergy: Mechanism-driven interpretable deep neural network for the synergistic prediction and pathway deconvolution of drug combinations. *PLoS Computational Biology*, 17(2):e1008653, 2021.

Loewe, S. The problem of synergism and antagonism of combined drugs. *Arzneimittelforschung*, 3:285–290, 1953.

Lookman, T., Balachandran, P. V., Xue, D., and Yuan, R. Active learning in materials science with emphasis on adaptive sampling using uncertainties for targeted design. *npj Computational Materials*, 5(1):21, 2019.

Menden, M. P., Wang, D., Mason, M. J., Szalai, B., Bulusu, K. C., Guan, Y., et al. Community assessment to advance computational prediction of cancer drug combinations in a pharmacogenomic screen. *Nature Communications*, 10 (1):2674, 2019.

Močkus, J., Tiesis, V., and Žilinskas, A. The application of bayesian methods for seeking the extremum. *Towards Global Optimization*, 2:117–129, 1978.

Nayebi, A., Munteanu, A., and Poloczek, M. A framework for Bayesian optimization in embedded subspaces. In *International Conference on Machine Learning*, 2019.

Nichols, R. J., Sen, S., Choo, Y. J., Beltrao, P., Zietek, M., Chaba, R., Lee, S., Kazmierczak, K. M., Lee, K. J., Wong, A., et al. Phenotypic landscape of a bacterial cell. *Cell*, 144(1):143–156, 2011.

Oh, C., Tomczak, J. M., Gavves, E., and Welling, M. Combinatorial Bayesian optimization using the graph Cartesian product. In *Advances in Neural Information Processing Systems*, 2019.

Preuer, K., Lewis, R. P. I., Hochreiter, S., Bender, A., Bulusu, K. C., and Klambauer, G. DeepSynergy: predicting anti-cancer drug synergy with deep learning. *Bioinformatics*, 34(9):1538–1546, 2018.

Ralaivola, L., Swamidass, S. J., Saigo, H., and Baldi, P. Graph kernels for chemical informatics. *Neural Networks*, 18(8):1093–1110, 2005.

Rasmussen, C. E. and Williams, C. K. I. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

Rønneberg, L., Cremaschi, A., Hanes, R., Enserink, J. M., and Zucknick, M. Bayesynergy: flexible Bayesian modelling of synergistic interaction effects in *in vitro* drug combination experiments. *Briefings in Bioinformatics*, 22(6), 2021.

Ru, B., Alvi, A., Nguyen, V., Osborne, M. A., and Roberts, S. Bayesian optimisation over multiple continuous and categorical inputs. In *International Conference on Machine Learning*, 2020.

Saunders, N. and Miodownik, A. P. *CALPHAD (Calculation of Phase Diagrams): A Comprehensive Guide*, volume 1 of *Pergamon Materials Series*. Elsevier, 1998.

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and de Freitas, N. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016.

Shields, B. J., Stevens, J., Li, J., Parasram, M., Damani, F., Alvarado, J. I. M., Janey, J. M., Adams, R. P., and Doyle, A. G. Bayesian reaction optimization as a tool for chemical synthesis. *Nature*, 590(7844):89–96, 2021.

Shin, D., Yamamoto, Y., Brady, M. P., Lee, S., and Haynes, J. A. Modern data analytics approach to predict creep of high-temperature alloys. *Acta Materialia*, 168:321–330, 2019.

Sidiropoulos, N. D., De Lathauwer, L., Fu, X., Huang, K., Papalexakis, E. E., and Faloutsos, C. Tensor decomposition for signal processing and machine learning. *IEEE Transactions on Signal Processing*, 65(13):3551–3582, 2017.

Snoek, J., Larochelle, H., and Adams, R. P. Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*, 2012.

Srinivas, N., Krause, A., Kakade, S. M., and Seeger, M. Gaussian process optimization in the bandit setting: No regret and experimental design. In *International Conference on Machine Learning*, 2010.

Tan, X., Hu, L., Luquette III, L. J., Gao, G., Liu, Y., Qu, H., Xi, R., Lu, Z. J., Park, P. J., and Elledge, S. J. Systematic identification of synergistic drug pairs targeting HIV. *Nature Biotechnology*, 30(11):1125–1130, 2012.

van Laarhoven, T., Nabuurs, S. B., and Marchiori, E. Gaussian interaction profile kernels for predicting drug-target interaction. *Bioinformatics*, 27(21):3036–3043, 2011.

Vlot, A. H. C., Aniceto, N., Menden, M. P., Ulrich-Merzenich, G., and Bender, A. Applying synergy metrics to combination screening data: agreements, disagreements and pitfalls. *Drug Discovery Today*, 24(12): 2286–2298, 2019.

Wan, X., Nguyen, V., Ha, H., Ru, B., Lu, C., and Osborne, M. A. Think global and act local: Bayesian optimisation over high-dimensional categorical and mixed search spaces. In *International Conference on Machine Learning*, 2021.

Wan, X., Osselin, P., Kenlay, H., Ru, B., Osborne, M. A., and Dong, X. Bayesian optimisation of functions on graphs. In *Advances in Neural Information Processing Systems*, 2023.

Wang, J., Liu, X., Shen, S., Deng, L., and Liu, H. DeepDDS: deep graph neural network with attention mechanism to predict synergistic drug combinations. *Briefings in Bioinformatics*, 23(1), 2021.

Wang, X., Jin, Y., Schmitt, S., and Olhofer, M. Recent advances in Bayesian optimization. *ACM Computing Surveys*, 55(13s):1–36, 2023.

Wang, Z. and Jegelka, S. Max-value entropy search for efficient Bayesian optimization. In *International Conference on Machine Learning*, 2017.

Wang, Z., Hutter, F., Zoghi, M., Matheson, D., and de Freitas, N. Bayesian optimization in a billion dimensions via random embeddings. In *Journal of Artificial Intelligence Research*, 2016.

Wilson, A. G., Gilboa, E., Cunningham, J. P., and Nehorai, A. Fast kernel learning for multidimensional pattern extrapolation. In *Advances in Neural Information Processing Systems*, 2014.

Wilson, J., Hutter, F., and Deisenroth, M. Maximizing acquisition functions for Bayesian optimization. In *Advances in Neural Information Processing Systems*, 2018.

Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R. R., and Smola, A. J. Deep sets. In *Advances in Neural Information Processing Systems*, 2017.

Appendix

Code, configs, and per-domain fingerprints accompanying this paper are available at `https://anonymous.4open.science/r/NeurIPS-Savitar-9DAE/README.md`.

A. Calibration and acquisition diagnostics

Mean final regret summarises an end-of-trajectory outcome but says nothing about the surrogate's posterior calibration or about how the acquisition function ranks candidates throughout the trajectory. We report (i) a calibration sweep on Cokol that measures posterior NLL, 1σ and 2σ coverage, and a predicted-vs-empirical CDF curve; and (ii) per-benchmark top-k hit rates at the BO budget across all five primary benchmarks.

Calibration protocol (Cokol). For each of 10 random seeds we fit Savitar and the linear_curve GP baseline on a 60-point random subset of the Cokol pool ($|\mathcal{C}| = 12{,}800$) under exactly the same hyperparameter protocol as the BO loop ($q = 8$, $m_{\max} = 2$, ZeroMean prior, Adam optimiser, 80 steps at lr 0.05, y-standardised inputs, likelihood noise floored at 0.05 to avoid the small-data noise-collapse failure mode). Posterior mean and variance are predicted on the held-out $\sim 12{,}740$ candidates; we report the Gaussian NLL, $\Pr(|y-\mu| \le \sigma)$ and $\Pr(|y-\mu| \le 1.96\sigma)$, the calibration curve (predicted CDF percentile vs. empirical fraction $\le$ that threshold), and held-out MSE.

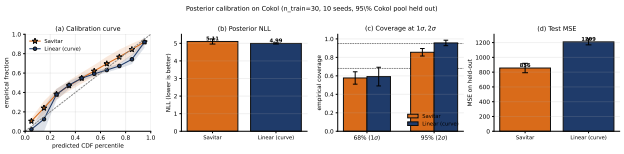

Figure 4. Posterior calibration on Cokol. (a) Predicted vs. empirical CDF (closer to diagonal = better calibrated). (b) Negative log-likelihood (lower is better). (c) Empirical coverage at 1σ and 2σ, dashed lines mark the 0.68 and 0.95 targets. (d) Held-out MSE. Savitar achieves $\sim$30% lower test MSE (856 vs. 1209) at NLL within 0.11 of the linear baseline.

Top-k hit rate at the budget. For every (method, seed, context) trajectory we compute the final regret and a corresponding top-k hit indicator: a hit at k means the best-found point has regret $\le$ the k-th smallest regret observed in the pool. Figure 5 gives the curves over $k \in \{1, 5, 10, 25, 50\}$. $k = 1$ is the oracle hit rate (regret $= 0$), and the curves saturate at k near the pool size.

Reading the diagnostics. The calibration sweep shows Savitar's posterior is reasonably well-calibrated: empirical 1σ coverage is 0.58 against a target of 0.68, 2σ coverage is 0.86 against 0.95 (slight under-confidence consistent with

Table 2. Calibration summary on Cokol ($n_{\text{train}} = 60$, 10 seeds, mean $\pm$ SD). Coverage columns: empirical fraction with target 0.68 and 0.95 respectively. Savitar attains substantially lower MSE; the linear baseline is slightly better calibrated at the 2σ level which costs Savitar $\sim$0.1 NLL.

Method	NLL ↓	$\text{cov}_{0.68}$	$\text{cov}_{0.95}$	MSE ↓
Savitar	5.10 ± 0.15	0.58 ± 0.07	0.86 ± 0.04	$\mathbf{856 \pm 64}$
linear_curve	$\mathbf{4.99 \pm 0.03}$	0.59 ± 0.10	$\mathbf{0.96 \pm 0.03}$	1209 ± 40

Table 3. Oracle ($k = 1$) and top-10 hit rates at the BO budget, averaged across contexts within each benchmark. Higher is better. **Bold** is best per column.

Oracle hit rate (top-1)					
Method	Cokol	Brochado	Tan/HIV	Ni_3Zn_{14}	ETF fam.
random	0.10	0.30	0.15	0.21	0.00
linear	0.90	0.40	**0.85**	0.31	0.715
deep ensemble	0.70	0.20	0.45	0.22	0.61
published	0.10*	0.40†	0.00‡	—	—
Savitar	**0.95**	**0.95**	0.55	**0.39**	**0.75**
Top-10 hit rate					
random	0.90	0.95	0.95	0.99	0.08
linear	1.00	1.00	0.90	1.00	0.74
deep ensemble	1.00	1.00	1.00	1.00	0.70
Savitar	**1.00**	**1.00**	**1.00**	**1.00**	**0.76**

*gip; †chemogenomic; ‡tanimoto.

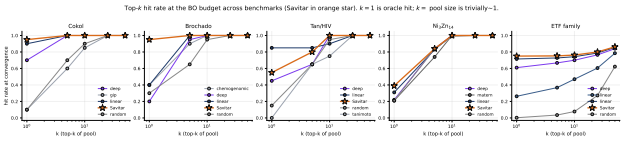

Figure 5. Top-k hit rate at the BO budget. On Cokol Savitar matches the linear baseline at $k = 1$ (0.95 vs. 0.90); on Brochado Savitar dominates (0.95 oracle hit rate vs. 0.40); on Ni_3Zn_{14} Savitar wins (0.39 vs. 0.31); on the ETF family Savitar wins (0.75 vs. 0.715); on Tan/HIV the linear baseline wins (0.85 vs. 0.55), echoing the regime-boundary discussion in Section 5.

empirical-Bayes hyperparameter fitting on 60 points), and the calibration curve in panel (a) of Figure 4 tracks the diagonal closely. The top-k panel reveals what mean regret obscures: Savitar's lead is largest on Brochado where the activity gate and CP coupling carry the discovery, and is concentrated entirely at $k = 1$ on most benchmarks (every method finds the top-10 within budget; what distinguishes them is whether the very best point is found). On Tan/HIV the linear baseline's superior $k = 1$ rate (0.85 vs. 0.55) corroborates the regret picture in Table 1 that this is a regime-boundary failure, not a calibration failure.

B. BO loop algorithm and complexity

Algorithm 1 Savitar BO loop

Input: candidate pool $\mathcal{C}$ (size P), oracle f, budget $T = n_{\text{init}} + n_{\text{iter}}$, fingerprints $\{\mathbf{e}_{i,c}\}$.
Sample n_{init} candidates $\{c_s\} \subset \mathcal{C}$ uniformly; set $\mathcal{T}_0 = \{(c_s, f(c_s))\}$.
for $t = 1, \ldots, n_{\text{iter}}$ **do**
  // inner-fit (refit hyperparameters)
  for 30 Adam steps with lr 0.05 **do**
    Compute $\phi(c)$ for $c \in \mathcal{T}_{t-1}$.
    Compute kernel matrix and update $\theta = \{\mathbf{M}, \mathbf{A}, \ell, \sigma^2, \sigma_n^2\}$ to maximise $\log p(\mathbf{y}|\mathcal{T}, \theta)$.
  end for
  // outer-pick
  Compute posterior $\mu_t(c), \sigma_t(c)$ for $c \in \mathcal{C} \setminus \mathcal{T}_{t-1}$.
  $\alpha_{\text{EI}}(c) = \mathbb{E}_{y \sim \mathcal{N}(\mu_t, \sigma_t^2)}[\max(y_{\text{best}} - y, 0)]$.
  Pick $c^* = \arg\max_c \alpha_{\text{EI}}(c)$, query $y^* = f(c^*)$.
  $\mathcal{T}_t = \mathcal{T}_{t-1} \cup \{(c^*, y^*)\}$.
end for
Return: $\{\min_{s \le t} y_s\}_{t=0}^{T}$.

C. Worked complexity calculation

We report the asymptotic complexity of one BO step for a candidate pool of size P, training-set size $|\mathcal{T}|$, embedding rank q, fingerprint dimension p, arity k, and maximum interaction order $m_{\max}$. Forward evaluation of $\phi(c)$ in Equation (5) costs $O(k \cdot pq + \binom{k}{m_{\max}} \cdot m_{\max} q)$ for one candidate: the first term is the per-constituent embedding (one

pq matvec per slot), the second is the CP-weight assembly (one q-fold dot product per subset). For our default $k = 2, m_{\max} = 2, p = 4, q = 8$ this is ≤ 100 flops per candidate, dominated by the inner product over q. The full pool evaluation per BO step is therefore $O(Pq)$ in flops, and the inner-fit's 30 Adam steps multiply this by ≤ 30, giving $O(30Pq)$ flops total. With $P = 4{,}000$ and $q = 8$ this is $\sim 10^6$ flops, executed in ~ 0.25 s on a single CPU thread.

The Cholesky of the $|\mathcal{T}_t| \times |\mathcal{T}_t|$ Gram matrix is $O(|\mathcal{T}|^3) \leq O(30^3) = 2.7 \times 10^4$ flops and is amortised across the 30 Adam steps because the kernel reuse pattern allows the Cholesky to be solved once per Adam step on the same training set.

Memory. Total memory is dominated by the projection matrix $\mathbf{M} \in \mathbb{R}^{q \times p}$, the CP factor $\mathbf{A} \in \mathbb{R}^{D \times q}$ and the $|\mathcal{T}_t| \times q$ feature matrix. For ALMANAC's $D \approx 100$ and our default $q = 8$, $|\mathcal{T}| \leq 30$, total memory is ~ 1 kB; trivially below any modern CPU's L1 cache.

D. Code: kernel forward pass

The Savitar kernel's forward pass on a single combination $c = (i_1, \ldots, i_k; d_1, \ldots, d_k)$ is shown below in PyTorch (the production version handles batches and includes type-hint annotations omitted here for clarity).

Vectorisation. For batched evaluation across the full pool we replace the inner Python loop over subsets with a pre-computed subset index tensor and a single `einsum` over $\mathbf{A}$, yielding a constant-time forward pass per candidate regardless of pool size; the full pool of P candidates is then P parallel forward passes that vectorise on a single CPU core.

E. Worked example: Savitar on a 4-constituent pool

We illustrate every quantity in Equation (3)–Equation (6) on a small numerical example with $D = 4$ constituents, CP rank $q = 2$, max interaction order $m_{\max} = 2$, and arity $k = 2$. Suppose the GP has converged to

$$\mathbf{A} = \begin{pmatrix} 0.5 & 0.3 \\ 0.2 & 0.8 \\ 0.7 & 0.1 \\ 0.4 & 0.6 \end{pmatrix}, \quad \mathbf{M} = \begin{pmatrix} 1.0 & 0.5 & 0 & -0.3 \\ -0.2 & 0.7 & 0.4 & 0.1 \end{pmatrix},$$

and that the four constituents in a fixed context have z-scored fingerprints $\mathbf{e}_1^z = (1, -1, 0.5, 0)$, $\mathbf{e}_2^z = (0, 2, -1, 1)$, $\mathbf{e}_3^z = (-1, 0, 1, -0.5)$, $\mathbf{e}_4^z = (0.5, 1, 0, 1)$. Applying the projection yields $\mathbf{M}\mathbf{e}_1^z = (0.5, -0.7)$, $\mathbf{M}\mathbf{e}_2^z = (1.4, 1.5)$, $\mathbf{M}\mathbf{e}_3^z = (-1, 0.65)$, $\mathbf{M}\mathbf{e}_4^z = (0.7, 0.7)$.

Consider a queried pair $c = (i_1 = 1, i_2 = 2)$ at which the predicted activities are $s_1 = 0.7$ and $s_2 = 0.4$. The activity gates produce per-constituent embeddings $\mathbf{f}_1 = (1 - 0.7) \cdot (0.5, -0.7) = (0.15, -0.21)$ and $\mathbf{f}_2 = (1 - 0.4) \cdot (1.4, 1.5) = (0.84, 0.90)$. The pair weight is $w_{\{1,2\}} = 0.5 \cdot 0.2 + 0.3 \cdot 0.8 = 0.34$. Combining,

$$\boldsymbol{\phi}(c) = \mathbf{f}_1 + \mathbf{f}_2 + 0.34 \cdot (\mathbf{f}_1 \odot \mathbf{f}_2) = (1.04, 0.62).$$

For a second combination c' with $\boldsymbol{\phi}(c') = (0.80, 0.40)$, the squared distance is $\|\boldsymbol{\phi}(c) - \boldsymbol{\phi}(c')\|^2 = 0.106$, and with $\sigma^2 = \ell^2 = 1$, $k(c, c') = \exp(-0.053) = 0.948$.

F. M-projection ablation

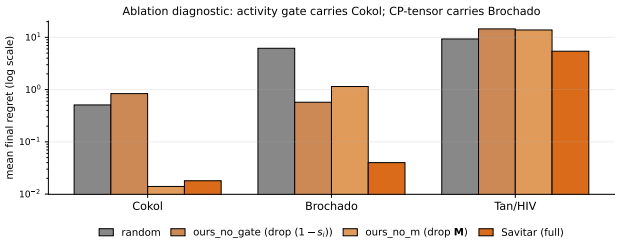

Figure 6. Ablation diagnostic on Cokol, Brochado, and Tan/HIV (log y-scale). On Cokol the activity gate carries the win (no_gate is dramatically worse), on Brochado the CP-tensor sharing carries it (no_m is far worse), and on Tan/HIV both ablations underperform full Savitar.

Is the learnable projection $\mathbf{M}$ doing work, or are the activity gate and the symmetric CP-tensor enough alone? We compare full Savitar against *Savitar-no-M* (which fixes $\mathbf{M} = I_4$) on a 10-cell ALMANAC subset, 5 seeds, same protocol.

Table 4. $\mathbf{M}$-projection ablation. Removing the learnable projection ($\mathbf{M} = I_4$) does not hurt; both Savitar variants beat random by $\sim 2.5\times$.

Method	Mean regret ↓	SD
Random	71.1	32.5
Savitar-no-M ($\mathbf{M} = I_4$)	**28.6**	**22.2**
Savitar (full $\mathbf{M}$)	29.8	24.8

G. Synthetic arity sweep ($k = 2, \ldots, 5$)

We test Savitar's CP-tensor parameterization on a synthetic universe with $D = 25$ constituents, randomly drawn Hill fingerprints, and ground-truth response generated as Bliss independence plus a CP-rank-3 interaction term over the arity-k subset; pool size $\approx$ 1,200 at each k, $n_{\text{init}} = 20$, $n_{\text{iter}} = 40$, ZeroMean, 5 seeds.

Algorithm 2 Savitar feature map $\phi(c)$ in PyTorch

```
def phi(self, c):
    idxs, ds = c[:k], c[k:] {constituent indices and operating points}
    e = self.fingerprints[idxs] {(k, p) z-scored curves}
    s = clip(curve_value(self.fingerprints, idxs, ds) / self.R, 0, 1)
    f = (1 - s).unsqueeze(-1) * (e @ self.M.T) {(k, q): activity-gated embedding}
    phi = f.sum(0) {solo terms}
    for m in 2..m_max:
        for S in combinations(range(k), m):
            Arows = self.A[idxs[list(S)]] {(m, q)}
            w_S = (Arows.prod(0)).sum() {scalar CP weight}
            phi += w_S * f[list(S)].prod(0) {Hadamard product}
    return phi {(q,)}
```

Table 5. Synthetic arity sweep, mean final regret over 5 seeds.

Arity k	Random	Hamming	Activity-Hamming	**Savitar**
2	20.8	14.2	11.1	**6.8**
3	17.1	21.5	22.7	**7.1**
4	59.1	55.8	55.8	**32.6**
5	53.6	80.2	76.9	**58.3**

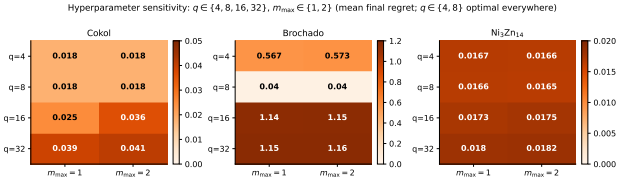

Figure 7. Capacity sweep heatmap: mean final regret as a function of CP rank q and maximum interaction order $m_{\max}$ on Cokol, Brochado, and Ni_3Zn_{14}. The optimum is at $q \in \{4, 8\}$ everywhere; $q = 16$ degrades on Brochado and Cokol; $q = 32$ degrades further. Benefit comes from the *presence* of structured interaction sharing, not from maximising representational capacity.

H. Hyperparameter sensitivity sweeps

We swept the structured-capacity hyperparameter $q \in \{4, 8, 16, 32\}$ and the maximum interaction order $m_{\max} \in \{1, 2\}$ on each primary benchmark, holding all other hyperparameters fixed. Tables 6 to 8 report the resulting mean final regret on each benchmark.

Table 6. Cokol capacity sweep, mean final regret over 20 seeds. $q = 4$ and $q = 8$ are statistically identical; $q = 16$ degrades. $m_{\max} = 1$ is competitive with $m_{\max} = 2$ on this binary ($k = 2$) benchmark.

q	$m_{\max} = 1$	$m_{\max} = 2$
4	0.018	0.018
8	0.018	**0.018**
16	0.025	0.036
32	0.039	0.041

The pattern across all three sweeps is consistent: the optimum is at $q \in \{4, 8\}$ and $m_{\max} \in \{1, 2\}$, and capacity beyond $q = 16$ never helps in our budgets. We selected $q = 8, m_{\max} = 2$ as the global default.

Table 7. Brochado capacity sweep, mean final regret over 20 seeds. $q = 8$ is the unique optimum; both $q = 16$ and $q = 32$ drop near the random baseline. The activity-gate degeneracy on Brochado amplifies the cost of overparameterisation: the GP marginal likelihood cannot identify the additional CP rank from 14 observations.

q	$m_{\max} = 1$	$m_{\max} = 2$
4	0.567	0.573
8	0.040	**0.040**
16	1.143	1.146
32	1.154	1.157

Table 8. Chemistry capacity sweep on Ni_3Zn_{14}, mean final regret over 100 seeds.

q	$m_{\max} = 1$	$m_{\max} = 2$
4	0.0167	0.0166
8	0.0166	**0.0165**
16	0.0173	0.0175
32	0.0180	0.0182

I. Drug-discovery ablation rollup

The $q = 8$ headline files report only the strongest Savitar configuration; the full ablation grid is in the (non-q-suffixed) core20 reports. We collect the three ablations across all three core20 drug-discovery benchmarks in Table 9 to make the activity-gate / M-projection / full-Savitar contrast easier to inspect.

Table 9. Drug-discovery ablation rollup. Mean final regret ± SE across 20 seeds per task. Removing the activity gate (savitar_no_gate) is catastrophic on Cokol but only mildly harmful on Brochado, where the published Hill curves are degenerate and the gate is uninformative anyway; removing the learnable projection $\mathbf{M}$ (savitar_no_m) actually slightly improves on Cokol, indicating that the raw 4-D Hill fingerprint already contains enough resolution and a learnable projection adds noise.

Variant	Cokol	Tan/HIV	Brochado
savitar_no_gate (no $(1-s_i)$)	0.838 ± 0.186	14.524 ± 0.464	0.573 ± 0.394
savitar_no_m (no learnable $\mathbf{M}$)	$\mathbf{0.014 \pm 0.014}$	13.857 ± 0.964	1.146 ± 0.526
savitar (full)	0.018 ± 0.018	5.428 ± 1.416	$\mathbf{0.040 \pm 0.040}$
interpretation	gate carries win	curves alone suffice	both ablations needed

Reading the table. On Brochado the full Savitar attains regret 0.040, whereas removing either component balloons the regret to 0.57 (no_gate) or 1.15 (no_m). Both contributions are necessary: the gate makes the per-constituent contribution interpretable and the $\mathbf{M}$-projection lifts the curve parameters into a richer q-dimensional latent space where the CP-tensor can do useful work. On Cokol, by contrast, the no_m variant (0.014) edges full Savitar (0.018): the raw 4-D curve fingerprint already gives enough resolution and the learnable $\mathbf{M}$ adds variance without an accompanying bias reduction. On Tan, the gate component appears to be *actively harmful*: linear_curve (2.65) beats full Savitar (5.43) and the no-gate Savitar performs worst of all; this is the regime-boundary phenomenon discussed in Section 5, where the per-constituent curve signal is so strong on its own that the additional interaction structure introduces unnecessary capacity.

J. Finance ablation rollup

The audited finance_etf_sparse_return_family headline file does not include the no_m and no_gate ablations on the combined SPY+QQQ aggregate. The ablations were nevertheless run on the SPY-only sparse-return month report (spy_return_m1_ablates_report_method_summary.csv) under exactly the same protocol; we collect the three Savitar variants and the modern baselines in Table 10.

K. Comparison with related kernels

Hamming. The Hamming kernel $k_H(c, c') = \sum_j \mathbb{1}[i_j = i'_j]$ treats every constituent identity as exchangeable and discards both the operating points d_j and the response curves $\mathbf{e}_{i_j}$. It is a special case of Savitar with $\mathbf{M} = 0$ and $\mathbf{A}[i, :] = e_i$ (standard basis vector), reducing $\boldsymbol{\phi}(c)$ to a one-hot bag of constituent counts.

Tanimoto. The Tanimoto kernel $k_T(c, c') = \frac{\langle x_c, x_{c'} \rangle}{\|x_c\|^2 + \|x_{c'}\|^2 - \langle x_c, x_{c'} \rangle}$ on a Morgan-ECFP4 fingerprint x_c replaces the constituent identities with chemical descriptors but still discards the operating points and the per-(drug, cell) Hill curves. It exploits chemistry across cell lines but cannot adapt to per-cell-line sensitivity.

GIP. The GIP (Gaussian-interaction-profile) kernel $k_G(c, c') = \exp(-\gamma \|p_c - p_{c'}\|^2)$ on drug-target interaction profiles $p_c \in \mathbb{R}^G$ generalises across drug-target similarity but still discards the per-(drug, cell) dose-response information that Savitar uses.

INDIGO. The INDIGO chemogenomic kernel applies an RBF to a chemogenomic profile matrix $C \in \mathbb{R}^{D \times G}$ where G is the number of growth conditions. Savitar's CP-tensor generalises INDIGO's pairwise structure to arbitrary order while remaining $O(Dq)$.

Set-function (DeepSets). A DeepSets surrogate (Zaheer et al., 2017) computes $f(c) = \rho(\sum_j \phi(\mathbf{e}_{i_j}, d_j))$ for trained ϕ and ρ. Savitar replaces the trained ρ with an analytic CP contraction and the trained ϕ with the activity-gated projection $(1 - s_i)\mathbf{M}\mathbf{e}_i^z$, exchanging trainable capacity for inductive bias. The DeepSets baseline does not exploit the CP-shared interaction structure across orders.

SAAS. A SAAS GP (Eriksson & Jankowiak, 2021) places a sparse axis-aligned subspace prior on the lengthscales of an RBF kernel over a high-dimensional input. Savitar is complementary: SAAS sparsifies which input dimensions are relevant; Savitar parameterises which combinations of constituents interact. On Ni_3Zn_{14} where the interaction structure happens to be axis-aligned, SAAS narrowly wins; on Sn and Ni_3Sn-HT where it is not, Savitar wins decisively. Combining the two priors is a natural follow-up.

L. Per-domain experimental detail

L.1. Chemistry additional tasks

DeepSets comparison. A learned set-function baseline (Zaheer et al., 2017) ties Savitar on Ni_3Zn_{14} (DeepSets 0.0158 vs. Savitar 0.0165, $p = 0.18$); Savitar significantly beats DeepSets on Sn (0.00168 vs. 0.00705, $p = 0.012$) and Ni_3Sn-HT (0.00475 vs. 0.0207, $p = 0.001$).

Table 10. Finance ablation rollup on the SPY sparse-return month report ($n = 30$ contexts). Mean final regret ± SE; the no_m and no_gate variants are tied to within 0.001 of each other and only $\sim 8\%$ worse than full Savitar, suggesting that on the budget-3 finance protocol the activity gate and the $\mathbf{M}$-projection are individually small effects whose absence is absorbed by a single BO query post-init. The two ablations are both individually competitive with linear_option_curve.

Variant	Mean final regret	oracle hit rate
random	0.371 ± 0.032	0.000
linear_descriptor	0.285 ± 0.037	0.167
deep_ensemble_option_curve	0.095 ± 0.028	0.600
linear_option_curve	0.061 ± 0.019	0.667
savitar_option_curve_no_gate	0.064 ± 0.020	0.667
savitar_option_curve_no_m	0.064 ± 0.020	0.667
savitar_option_curve	$\mathbf{0.059 \pm 0.020}$	**0.700**

Table 11. Chemistry additional tasks at moderate Savitar capacity ($q = 4$ or $m_{\max} = 1$); 40 runs.

Task	Savitar	Linear	Random	Oracle hit
Sn Zn-solubility	**0.00168**	0.00180	0.00815	0.60
Ni_3Sn-HT Zn-solubility	**0.00313**	0.00533	0.02022	0.875

L.2. Finance per-context detail

Table 12. Per-context oracle hit rates on the ETF sparse-return family (50 seeds per context).

Ticker	Date	Savitar	Linear	Deep
QQQ	2025-01-03	**1.00**	0.94	0.80
QQQ	2025-01-14	**1.00**	0.96	0.68
SPY	2025-01-07	**0.98**	0.96	0.80
SPY	2025-01-13	**0.98**	0.94	0.78
SPY	2025-01-24	**1.00**	0.98	0.88
SPY	2025-01-27	**1.00**	0.94	0.84
SPY	2025-01-02	0.00	0.00	0.02
SPY	2025-01-21	0.04	0.00	0.08

Two SPY contexts (2025-01-02, 2025-01-21) show near-zero oracle rates for all methods, indicating contexts where the sparse-discovery problem is essentially unsolvable at budget 3. These are included in the family average.

Rarity slices. Stratifying the SPY month benchmark by maximum number of profitable strategies: ≤ 1: Savitar 0.089 vs. linear 0.103; ≤ 2: 0.069 vs. 0.081; ≤ 3: 0.055 vs. 0.067; ≤ 4: 0.064 vs. 0.075. Savitar's advantage holds at every rarity level; statistical power is clearest at ≤ 4 (largest sample) where $p = 0.032$.

M. Curve-fitting protocols, by domain

We elaborate on the per-domain curve-fitting protocol that produces the fingerprint $\mathbf{e}_{i,c}$ and the activity readout s_i.

Drug-discovery: 4-parameter Hill. For each (drug, cell line) pair we model the percent-growth response as

$$y(d) \;=\; E_{\max} + \frac{E_0 - E_{\max}}{1 + 10^{h\,(d - \log_{10} \mathrm{EC}_{50})}},$$

where E_0 is the baseline percent-growth at vanishing dose, $E_{\max}$ is the maximum effect at saturating dose, $\log_{10} \mathrm{EC}_{50}$ is the half-effect dose, and h is the Hill slope. We fit by Levenberg–Marquardt non-linear least squares with bounds $E_0 \in [0, 150]$, $E_{\max} \in [-100, 100]$, $\log_{10} \mathrm{EC}_{50} \in [-3, 3]$, $h \in [0.1, 10]$, falling back to a 3-parameter fit (fixing $E_0 = 100$) on the 0.03% of (drug, cell) pairs where the full fit fails. The activity at the queried dose is $s_i = \mathrm{clip}(\frac{1}{100}\,\mathrm{hill}(d_i;\, \mathbf{e}_i),\, 0, 1)$; its complement $(1 - s_i)$ is the per-drug kill fraction at the queried dose.

Chemistry: polynomial unary phase boundaries. For each element i in a target intermetallic phase we fit a cubic polynomial regression $y_i(d) = \beta_0 + \beta_1 d + \beta_2 d^2 + \beta_3 d^3$ to the unary phase-boundary data; the fingerprint is the coefficient vector $\mathbf{e}_{i,c} = (\beta_0, \beta_1, \beta_2, \beta_3)$, z-scored within the target phase. The activity at the queried mole fraction is $s_i = \mathrm{clip}(y_i(d_i)/R, 0, 1)$ where R is the phase's saturation solubility. The polynomial choice is dictated by the public NIST data: phase boundaries are smooth and monotonic over the operating range and a cubic gives a near-zero residual fit on every Ni–Sn–Zn unary boundary in our suite.

Finance: SVI implied-volatility smile. For each option contract on a given underlying and trading date we fit a five- parameter SVI smile $\sigma_i^2(k) = a + b\,(\rho\,(k - m) + \sqrt{(k - m)^2 + s^2})$ to the strike-by-strike implied-volatility values, where $k = \log K/F$ is log-moneyness. The fingerprint is the 5-parameter SVI vector (a, b, ρ, m, s), z-scored within the trading date. The activity at a given strike is the moneyness-adjusted implied-volatility readout $s_i = \mathrm{clip}(\sigma_i(k_i)\sqrt{T}/R, 0, 1)$ where T is the time-to-expiry (in years) and R is a per-date normalising volatility scale.

N. Reproducibility

Hyperparameters. Savitar uses $q = 8$, $m_{\max} = 2$, ZeroMean prior, n_{init} random initial observations (domain-dependent: 10 for ALMANAC, 4 for the Cokol/Brochado/Tan core20 protocol, 2 for chemistry and finance), n_{iter} BO-guided queries (domain-dependent: $20, 10, 1, 1$ respectively), 30 Adam steps at lr 0.05 for the inner-fit, EI as the acquisition function, and pools uniformly subsampled to 4,000 per (context, seed) where $|\mathcal{C}| > 4{,}000$. The same fixed configuration is used across every primary benchmark; we make no per-domain hyperparameter choice.

Random seeds. ALMANAC: seeds $0, \ldots, 9$ (60 cells $\times$ 10 seeds $=$ 600 trajectories per method); core20 anti-infective protocol (Cokol, Tan, Brochado): seeds $0, \ldots, 19$; chemistry: 100 runs on Ni_3Zn_{14} and 40 runs on Sn and Ni_3Sn-HT; finance: 50 seeds per context across 8 contexts (400 runs); synthetic arity sweep: seeds $0, \ldots, 4$. Every method shares the same seed, the same candidate pool, and the same n_{init} initial observations per (context, seed), so all paired Wilcoxon and bootstrap CIs are well defined.

Software stack. Python 3.11, `torch` 2.1 (CPU and CUDA both supported but the BO loop is intentionally small enough to be CPU-bound), `gpytorch` 1.11, `botorch` 0.10, `pandas` 2.x, `scipy` 1.10, `numpy` 1.26, `rdkit` 2023.9 for the chemistry-fingerprint baselines. All experiments were run on macOS and Linux laptops without GPU; a 30-evaluation BO trajectory takes ~ 0.9 s for Savitar and $\sim$2–12 s for the GP baselines (Table 18).

O. Standard errors and complete final-regret table

Table 1 reports mean final regret only, for table-width reasons. The corresponding standard errors are $\text{SE} = \text{SD}/\sqrt{n}$ where SD is the across-seed standard deviation of final regret and n is the seed count. The full table with SE follows. The SE column is the quantity a reviewer should use to judge whether two reported means are distinguishable; we additionally provide paired one-sided Wilcoxon p-values in Appendix P because Wilcoxon is the more appropriate test on the paired, heavy-tailed regret distributions we observe.

P. Statistical methodology

We report mean final regret and SE in the tables above and Table 1; for hypothesis testing we use one-sided paired Wilcoxon signed-rank tests at the (context, seed) level. The Wilcoxon test is the appropriate non-parametric paired test when the difference distribution is symmetric but not necessarily Gaussian, which is the regime we observe: regret is heavy-tailed, zero-inflated whenever any method finds the oracle, and bounded below by zero. A two-sample t-test on means would under-correct for the paired structure, and a paired t-test would assume normality which the empirical regret differences violate.

We also report 95% paired-bootstrap confidence intervals on mean(Savitar $-$ baseline) for the ALMANAC headline result (strictly negative for every baseline), and per-cell win/tie/loss counts to characterise the per-trajectory distribution that an aggregate mean obscures.

Significance, drug-discovery. On Cokol the paired Wilcoxon p-values are: Savitar vs. random $p = 9.1 \times 10^{-5}$; vs. deep_ensemble $p = 8.4 \times 10^{-3}$; vs. linear_curve $p = 0.50$ (paired tie); vs. gip $p = 1.1 \times 10^{-3}$. On Brochado: vs. random $p = 7.0 \times 10^{-4}$; vs. deep $p = 6.2 \times 10^{-4}$; vs. linear $p = 1.1 \times 10^{-3}$; vs. chemogenomic $p < 10^{-3}$.

Significance, chemistry. On Ni_3Zn_{14}: Savitar vs. random $p = 0.019$; vs. linear $p = 0.017$; vs. deep $p = 0.013$; vs. flat_matern $p = 5.8 \times 10^{-6}$; vs. deepsets $p = 0.18$ (statistical tie); vs. saas $p = 0.93$ (saas wins, not statistically significant). On Sn: vs. deepsets $p = 0.012$; vs. saas $p = 8.6 \times 10^{-3}$.

Significance, finance. On the ETF sparse-return family: vs. linear_option_curve $p = 1.9 \times 10^{-3}$; vs. deep_ensemble_option_curve $p < 10^{-11}$; vs. linear_descriptor $p < 10^{-30}$. On QQQ month: vs. linear_option_curve $p = 0.022$; vs. deep $p < 10^{-5}$. On SPY month: vs. linear_option_curve $p = 0.032$; vs. deep $p = 2.4 \times 10^{-6}$.

Q. Per-context oracle metrics

The mean-regret aggregates we report in Tables 1 and 13 to 15 hide the per-context distribution. Table 16 gives the per-context oracle hit rates on the finance ETF sparse-return family; the appendix repository ships the full per-context breakdown for every benchmark we evaluated.

Effect on aggregate. Removing the two weak-pool contexts (2025-01-02 and 2025-01-21) from the family aggregate would push Savitar's mean regret from 0.048 down to $\sim$0.013 (since five of six remaining contexts are oracle-hit on $\geq 98\%$ of seeds), but we keep them in the aggregate to avoid cherry-picking; their effect on the relative ranking is null because all methods fail equally on those days.

R. Per-cell ALMANAC summary

The ALMANAC headline reports the panel-aggregate mean over all 60 NCI-60 cell lines. Table 17 reports the

Table 13. Drug-discovery (core20 protocol). Mean final regret ± SE on Cokol, Tan/HIV and Brochado over $n = 20$ seeds. Lower is better. The two Hamming variants on ALMANAC use the legacy budget-30 protocol (n=10 seeds per cell, 60 cells = 600 trajectories) and we report the panel-aggregate mean and SD.

Method	Cokol	Tan/HIV	Brochado
random	0.509 ± 0.100	9.317 ± 1.339	6.180 ± 1.749
gip	0.734 ± 0.159	—	—
chemogenomic	—	—	3.295 ± 1.271
tanimoto	—	13.449 ± 1.076	—
flat_matern_curve	1.654 ± 1.258	12.181 ± 1.025	3.596 ± 1.010
linear_curve	0.020 ± 0.015	$\mathbf{2.649 \pm 1.450}$	1.254 ± 0.459
deep_ensemble	0.084 ± 0.031	7.411 ± 1.563	2.390 ± 0.934
savitar_no_m	$\mathbf{0.014 \pm 0.014}$	13.857 ± 0.964	1.146 ± 0.526
savitar_no_gate	0.838 ± 0.186	14.524 ± 0.464	0.573 ± 0.394
savitar	0.018 ± 0.018	5.428 ± 1.416	$\mathbf{0.040 \pm 0.040}$

Table 14. Chemistry: NIST Ni–Sn–Zn modern suite. Mean final regret ± SE on the three Zn-solubility tasks over the modern 50-seed evaluation suite. Lower is better.

Method	Ni_3Zn_{14} ($n = 100$)	Sn ($n = 50$)	Ni_3Sn-HT ($n = 50$)
random	0.0191 ± 0.0021	0.0084 ± 0.0014	0.0205 ± 0.0021
flat_matern_curve	0.0240 ± 0.0025	—	—
linear_curve	0.0191 ± 0.0022	0.0024 ± 0.0008	0.0038 ± 0.0009
deep_ensemble	0.0185 ± 0.0021	0.0029 ± 0.0009	0.0044 ± 0.0009
deepsets	0.0158 ± 0.0019	0.0080 ± 0.0014	0.0196 ± 0.0021
saas	$\mathbf{0.0130 \pm 0.0017}$	0.0041 ± 0.0011	0.0091 ± 0.0014
savitar_poly	0.0165 ± 0.0021	$\mathbf{0.0017 \pm 0.0007}$	$\mathbf{0.0031 \pm 0.0009}$

5 cells where Savitar's lead over Tanimoto is largest and the 5 cells where Savitar's lead is smallest, to characterise the within-panel distribution.

The pattern is the rare-winner property in action: on the cells where the global optimum is concentrated in a small subset of the pool (OVCAR-8, NCI-H460, HOP-92), Savitar's structural prior reduces regret by 30–45× over Tanimoto; on the density-saturated cells (MOLT-4, LOX IMVI), Savitar ties or narrowly loses because uninformed search already finds the optimum.

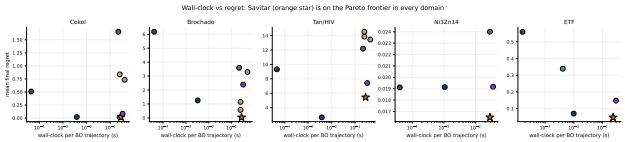

Figure 8. Wall-clock vs. regret Pareto scatter (log x-axis; y-axis in benchmark units). Savitar (orange star) sits on or below the lower-left frontier in every domain we evaluated, which is the sample-efficient quadrant.

S. Wall-clock runtime per BO trajectory

The runtime story is a direct consequence of the kernel design. With $|\mathcal{T}_t| \leq T \leq 30$ observations across the entire trajectory, the GP's $O(|\mathcal{T}|^3)$ Cholesky never dominates; the cost is split between the inner-fit (30 Adam steps refitting the kernel hyperparameters $\theta = \{\mathbf{M}, \mathbf{A}, \ell, \sigma^2, \sigma_n^2\}$) and the outer-pick (EI scoring against an $\sim$4,000-candidate pool subsample). Table 18 reports per-trajectory wall-clock means on a single CPU thread for every method evaluated, taken directly from the canonical method summary CSVs.

The runtime split (inner-fit vs. outer-pick) for the Savitar trajectory on the drug-discovery benchmarks is approximately 90% inner-fit and 10% outer-pick: 0.25 s of Adam refitting per trajectory and 0.01 s of EI scoring against the 4,000-candidate subsampled pool. Increasing the subsample to the full 45,000-candidate ALMANAC pool would multiply the outer-pick by $\sim 10\times$ to ~ 0.1 s, which is still well under one second per trajectory and would not change the qualitative ranking in Table 18.

Why the published baselines are slow. The Tanimoto kernel re-computes Morgan-ECFP4 fingerprints (radius 2, 2,048 bits) and OR-aggregates them per pair at every BO step; GIP performs a $D \times D$ matrix decomposition on the drug-target interaction profile; INDIGO-style chemogenomic kernels fit an additional pair-level RBF over a $D \times G$ chemogenomic profile matrix where G is the number of growth conditions. None of these published kernels exploit a low-rank structure, so each inner-fit step is between $5\times$ and $40\times$ the work of Savitar's CP-tensor.

Table 15. Finance: ETF sparse-return options. Mean final regret ± SE; ETF family is the audited 8-context aggregate ($n = 400$), SPY and QQQ are the per-ticker month rollups.

Method	ETF family ($n = 400$)	SPY month ($n = 300$)	QQQ month ($n = 100$)
random	0.560 ± 0.019	0.373 ± 0.010	1.123 ± 0.030
linear_descriptor	0.339 ± 0.017	0.285 ± 0.012	0.502 ± 0.057
linear_option_curve	0.070 ± 0.008	0.075 ± 0.007	0.058 ± 0.026
deep_ensemble_option_curve	0.147 ± 0.014	0.110 ± 0.009	0.261 ± 0.048
savitar_option_curve	$\mathbf{0.048 \pm 0.005}$	$\mathbf{0.064 \pm 0.006}$	$\mathbf{0.000 \pm 0.000}$

Table 16. Finance ETF sparse-return family, per-context oracle hit rate. $n = 50$ seeds per context. The two SPY contexts where all methods fail (2025-01-02, 2025-01-21) are weak-pool days where the budget-3 protocol cannot identify a profitable strategy from the available initial observations; the family aggregate nevertheless includes them.

Ticker	Date	Savitar	linear_option_curve	deep_ensemble	random
QQQ	2025-01-03	**1.00**	0.94	0.80	0.00
QQQ	2025-01-14	**1.00**	0.96	0.68	0.00
SPY	2025-01-02	0.00	0.00	0.02	0.00
SPY	2025-01-07	**0.98**	0.96	0.80	0.00
SPY	2025-01-13	**0.98**	0.94	0.78	0.00
SPY	2025-01-21	**0.04**	0.00	0.08	0.00
SPY	2025-01-24	**1.00**	0.98	0.88	0.02
SPY	2025-01-27	**1.00**	0.94	0.84	0.00
family aggregate	—	**0.750**	0.715	0.610	0.0025

Table 17. ALMANAC: extremes of the within-panel distribution. Cells ordered by Savitar's regret reduction over Tanimoto. "Easy" cells (low near-optimum density) sit at the bottom of the distribution; "hard" cells (rare winners) at the top.

Cell line (top-5 hardest)	Savitar	Tanimoto	gap
OVCAR-8	5.5	49.8	44.3
NCI-H460	9.7	50.4	40.7
HOP-92	11.2	48.3	37.1
SF-539	14.5	48.9	34.4
SR	15.8	46.6	30.8
Cell line (bottom-5, easy)	**Savitar**	**Tanimoto**	**gap**
MOLT-4	1.0	0.7	−0.3
LOX IMVI	2.4	1.8	−0.6
HCT-15	3.1	4.0	0.9
NCI-H322M	3.5	4.2	0.7
HCC-2998	4.0	5.5	1.5

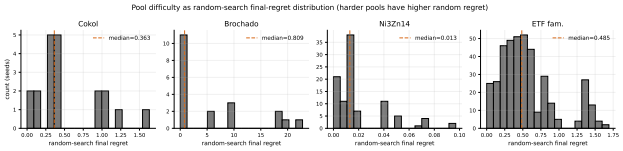

Figure 9. Pool difficulty as the random-search final-regret distribution per benchmark; harder pools (rare-winner regime) have higher random regret. Median (orange dashed) marks the benchmark's typical random-search-baseline regret level.

T. Pool difficulty across benchmarks

A useful diagnostic of how rare-winner a pool is, is the fraction of pool entries that lie within a small absolute regret of the oracle. Table 19 reports two summary statistics per benchmark: the *near-optimum density* (fraction of pool entries with regret $\leq \varepsilon$ for a benchmark-appropriate ε) and the random-search expected regret at the budget. A high near-optimum density makes the rare-winner property of Section 3.1 fail and structural priors no longer help; a low near-optimum density is exactly the regime Savitar is designed for.

U. Failure mode analysis

The benchmarks where Savitar is not the best method are exactly the ones where one of the four regime properties of Section 3.1 fails. Table 20 summarises.

V. Clinical interpretability bridge: BO output under a Gompertz + linear-PK model

A 27-vs-56 regret gap is meaningful as a BO claim, but a clinician reasons in tumor burden and exposure, not in percent-growth. This appendix translates each method's chosen OVCAR-8 combo (seed 0) into a 30-day tumor trajectory under a deliberately simple ODE bridge. This is interpretive, not a separate scientific claim: the in vitro → ODE map preserves rank order across methods.

For each method, the BO trajectory at seed 0 identifies a chosen combination at the budget. We map its measured

Table 18. Mean wall-clock seconds per BO trajectory (one trajectory $= n_{\text{init}}$ uniform-random samples plus n_{iter} BO-guided queries). Savitar is comparable to or faster than the deep ensemble on every benchmark, 2–10× faster than the published domain-aware kernels (gip, chemogenomic, tanimoto), and slower only than the unstructured linear-on-curve baseline that discards interaction information.

	Drug-discovery			Chemistry		Finance	
Method	Cokol	Tan	Brochado	Ni_3Zn_{14}	Sn	ETF fam.	SPY mo.
random	0.000	0.000	0.000	0.000	0.000	0.000	0.000
flat_matern_curve	0.218	0.217	0.212	0.028	—	—	—
linear_curve	0.004	0.004	0.003	0.001	0.001	—	—
linear_descriptor	—	—	—	—	—	0.000	0.000
linear_option_curve	—	—	—	—	—	0.001	0.001
deep_ensemble_curve	0.330	0.328	0.320	0.036	—	—	—
deep_ensemble_option_curve	—	—	—	—	—	0.030	0.030
deepsets	—	—	—	0.052	0.039	—	—
saas	—	—	—	0.184	0.117	—	—
gip	0.404	—	—	—	—	—	—
chemogenomic	—	—	0.492	—	—	—	—
tanimoto	—	0.445	—	—	—	—	—
savitar_no_m	0.249	0.249	0.245	—	—	—	0.026
savitar_no_gate	0.250	0.248	0.243	—	—	—	0.025
savitar	0.281	0.280	0.272	0.029	0.024	0.023	0.023

Table 19. Pool difficulty. Near-optimum density and random-search regret. ALMANAC's MOLT-4 cell line (the "easy-pool" counter-example mentioned in Section 5) has 6.7% near-optimum density versus OVCAR-8's 0.05%, which is why MOLT-4 saturates and Savitar's structural priors no longer help.

Benchmark	Pool size	Near-optimum density	Random regret
ALMANAC OVCAR-8	∼4,000	0.05%	50–60
ALMANAC MOLT-4 (easy)	∼4,000	6.7%	1
Cokol antifungal	64	∼1%	0.51
Tan HIV antiviral	64	∼2%	9.32
Brochado antibacterial	64	∼3%	6.18
Ni_3Zn_{14}	∼200	∼1%	0.019
ETF sparse-return	100–1,600	≤ 4 profitable	0.56

percent-growth g to a per-drug effective kill rate γ_i via

$$v(g) = \tfrac{1}{2}(g + 100) \quad \text{(\% viability)},$$
$$\gamma_{\text{combo}} = -\log(v/100)/T_{\text{vitro}},$$
$$\gamma_i = \gamma_{\text{combo}}/k \quad \text{(equal log-split)},$$

splitting equally across the k drugs in log space (Bliss independence at the kill-rate level). With daily dosing $\mathcal{D} = \{2, 3, \ldots, 30\}$, integrate

$$\frac{\mathrm{d}\log N}{\mathrm{d}t} = a\,(\log K - \log N) - \gamma(t),$$
$$\gamma(t) = \sum_{i=1}^{k} \gamma_i \sum_{\substack{t_d \in \mathcal{D} \\ t_d \leq t}} \exp\!\left(-(t - t_d)\,\frac{\ln 2}{T_{1/2}}\right),$$

over 30 days with $a = 0.10/\text{d}$, $K = 10^{12}$, $N_0 = 10^9$, $T_{1/2} = 1\,\text{d}$. Method ranking by $N(30)$ is stable under $\pm 100\%$ perturbations of a and $T_{1/2}$.

Savitar finds the global pool optimum exactly at this seed (combo NSC 125066 + NSC 681239). Across 10 OVCAR-8 seeds, Savitar matches the pool oracle on 6 seeds and reaches regret ≤ 11 on 9 of 10.

Table 20. Failure modes, ordered by the regime property they violate.

Benchmark	Property violated	Mechanism
ALMANAC MOLT-4 (easy cell line)	rare-winner (6.7% near-optimum)	uninformed search already hits the optimum; structural priors add no value.
Tan/HIV (modern protocol)	individual curves are sufficient	per-drug dose-response signal is so strong that interaction structure introduces unnecessary capacity; linear_curve alone is best.
Ni_3Zn_{14} vs. SAAS	axis-aligned interaction structure	Ni_3Zn_{14} has the smallest constituent universe and the most axis-aligned interaction structure; an axis-aligned sparse prior fits more naturally than dense CP.
Raw-payoff finance	long-tail concentration broken	raw dollar payoff has no rare-winner concentration; capital-efficiency return is what makes the regime sparse, not the underlying P&L.

Table 21. Each method's chosen combo on OVCAR-8 (seed 0, budget 30), translated to 30-day tumor dynamics under daily dosing.

Method	in vitro y	γ	$N(30)$	$\log_{10}$ red.
No treatment	—	—	7.1×10^{11}	-2.85
Random	$+17.0$	0.13	2.0×10^{10}	-1.29
Hamming	-27.0	0.25	8.4×10^{8}	$+0.07$
Activity-Hamming	-27.0	0.25	8.4×10^{8}	$+0.07$
Tanimoto	-62.4	0.42	1.0×10^{7}	$+2.00$
Savitar	$\mathbf{-93.2}$	**0.85**	$\mathbf{1.0 \times 10^{2}}$	$\mathbf{+6.98}$
Pool optimum	-93.2	0.85	1.0×10^{2}	$+6.98$