



Paper Type: Original Article

Generative Agent Swarm: Bacterial Foraging with K LLM-Mediated Pheromone Communication

Daniel A. Ferreira* 

School of Electrical and Computer Engineering, University of Campinas, Campinas, Brazil; daniel.ferreira@unicamp.br.

Citation:

Received: 18 December 2025

Revised: 27 February 2026

Accepted: 03 May 2026

Ferreira, D. A. (2026). Generative agent swarm: Bacterial foraging with LLM-mediated pheromone communication. *Metaheuristic Algorithms with Applications*, 3(2), 246-271.

Abstract

Expensive black-box optimization spends its budget fast. Simulation-driven engineering design and hyperparameter tuning admit only a few hundred true function evaluations, so every observation collected during the search becomes a priced asset. This paper deploys GAS-Bacterial Foraging Optimization (BFO), a surrogate-assisted optimizer in which a swarm of thirty large-language-model-mediated bacterial agents runs chemotaxis over a shared collective memory. Agents post structured textual observations of their local landscape throughout the run, and every ten iterations a language model condenses this memory into a natural-language pheromone, a small set of weighted region summaries that defines an explicit chemoattractant field biasing every agent's movement; a hallucination guard cross-checks each directional cue against the surrogate gradient and falls back to a classical tumble on disagreement. The evaluation rebuilds the budget ledger from the ground up: Twelve problems, ten mathematical benchmarks and two simulated engineering cases, three budget tiers of 150, 400, and 900 true evaluations, thirty runs per cell, and fourteen methods, nine established references plus four language-model optimizers newly admitted to the race, OPRO, FunSearch, EoH, and ReEvo, raced under token-matched accounting. Survival analysis on tier-relative time-to-target carries the inference: GAS-BFO holds the best mean gap on 22 of 36 problem-tier cells, reaches the target gap with the highest probability at every tier, and is significantly faster than all thirteen rivals at the 400- and 900-evaluation tiers under Holm-corrected log-rank tests, with hazard ratios pricing the separation, while conceding specific cells, the mixed-space tuning task to Tree-structured Parzen Estimation (TPE) among them, that the ledger records openly. Ablations attribute the gain to generative condensation and collective memory; the token tables state the price, one to three dollars per run at deployment tiers; and under bursty message dropout and controlled distributional shift, the swarm amortizes knowledge where single-model and single-history rivals concentrate it, retaining 79.4% of lossless quality at a 60% loss rate against 41.2% for Bayesian Optimization with Expected Improvement (BO-EI).

Keywords: Generative agents, Bacterial foraging optimization, Large language models, Surrogate-assisted optimization, Stigmergy, Expensive black-box optimization.

1 | Introduction

Many engineering and scientific objectives behave as expensive black boxes. One evaluation can bill minutes to hours of computation: A fluid-dynamics solve, a crash or structural simulation, or the full training of a

 Corresponding Author: daniel.ferreira@unicamp.br

 <https://doi.org/10.48313/maa.vi.80>



Licensee System Analytics. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0>).

machine-learning model [1], [2]. The design and analysis of computer experiments formalized this regime early [1]. Modern hyperparameter optimization inherits it at scale [3]. With the total number of evaluations B capped at a few hundred, observation, not reasoning, is the line item that binds. Every number the run collects acquires strategic value. The optimizer's central task becomes disciplined accumulation, compression, and redistribution of scarce evidence.

Two methodological lineages divide the market. Bayesian Optimization (BO) and its model-based relatives maintain a probabilistic surrogate of the objective and spend evaluations through an acquisition policy [4], [5]. The family is sample-efficient in low dimensions. But it books all knowledge in a single statistical model, which strains under multimodality, moderate dimension, or drift. Population-based metaheuristics explore in parallel and absorb rugged landscapes [6–9]. Their classical variants spend evaluations freely, though. The operators assume fitness probes are cheap, so tight budgets force a surrogate wrapper to keep them competitive. Neither lineage communicates intermediate findings in any rich sense. Population methods exchange selection pressure and occasional numeric neighborhood statistics, never an interpretable summary of what the swarm has learned.

Biological swarms settled the communication problem long ago. Their settlement is stigmergy: Agents coordinate by modifying a shared environment rather than by addressing each other directly [10]. Ant colonies deposit pheromone that amortizes thousands of foraging trips into a gradient-like field [11], [12]. The channel works precisely because it is lossy and graded. The field summarizes history, expires when unsupported, and biases movement without dictating it. Artificial stigmergy stayed numerically rigid instead. A pheromone table over discrete graph edges, as in Ant Colony Optimization (ACO), cannot express what a human analyst would report after inspecting the same evidence. That report might note a promising valley elongating toward smaller aspect ratios, or newest samples contradicting the previously hypothesized optimum.

Large Language Models (LLMs) re-priced what scattered, heterogeneous evidence can support [13–16]. They read hundreds of numeric observations and emit compact, structured hypotheses about landscape geometry. Agent architectures built around them have demonstrated persistent memory and reflective summarization in simulated social settings [17]. So far, optimization deployments cast the LLM as a proposal generator or an evolutionary operator [18]: the model itself suggests the next candidate. The complementary role remains unexplored. There, the LLM is not a proposer but a communication medium. It is a stigmergic channel that condenses a swarm's accumulated observations into a natural-language pheromone field. The channel sits inside a budget-aware surrogate framework, with explicit guards against hallucinated guidance and honest accounting of token cost.

The gap above prices the work. Three research questions convert it into spendable form, and three hypotheses wager falsifiable answers. Every claim is denominated in the currency this regime trades: final-solution quality per evaluation budget. The evaluation of Section 6 settles the ledger with fixed budgets, seeded replication, and rank-based inference. The ablations of Section 6.5 audit which components earn their keep, and the stress tests of Sections 6.7 and 7 complete the account.

RQ1. Does stigmergic LLM-mediated pheromone communication raise final-solution quality and time-to-target per evaluation budget over the Bayesian-optimization, evolutionary, and language-model lineages represented by thirteen reference methods across three budget tiers?

RQ2. Which protocol components carry the gain, message schemata, collective memory, self-consistency, the hallucination guard, or the acquisition-gated budget accounting, and how do the condensation cadence and swarm size trade quality against tokens at each tier?

RQ3. Does the gain survive communication stress in the form of bursty message dropout and latency, and distributional shift of the objective itself, against both single-model and single-history rivals?

H1: At every budget tier, GAS-BFO outperforms the reference field in tier-relative time-to-target, with Holm-corrected log-rank significance against all thirteen rivals at the 400- and 900-evaluation tiers, and holds the

best mean gap on 22 of the 36 problem-tier cells, with the mixed-space tuning case conceded to Tree-structured Parzen Estimation (TPE) among others.

H2: Component ablations attribute the gain to the communication layer, collective memory, self-consistency, and the hallucination guard, not to the surrounding surrogate machinery, and the cadence and swarm-size sweeps price the remaining dials against token cost.

H3: The gain persists under stress: retention under bursty dropout and latency stays above every reference method, and degradation under controlled shift stays below both the Bayesian references and the language-model single-history rivals.

This paper converts the complementary role into GAS-BFO, a generative agent swarm built on Bacterial Foraging Optimization (BFO) [8]. Thirty bacterial agents run chemotaxis over the search box. Each agent posts structured textual observations into a shared collective memory. Every ten iterations, an LLM condenses that memory into a small set of weighted region summaries. The summaries compile into an explicit chemoattractant field that biases every agent's run direction, in mixture with the gradient of a Gaussian-process surrogate. The budget manager admits a true evaluation only when Expected Improvement (EI) or condensation confidence covers the spend. A hallucination guard cross-checks every directional cue against the surrogate before it can move an agent. The target problem is stated formally as

$$\min_{\mathbf{x} \in X} f(\mathbf{x}) \text{ s.t. } n_{\text{eval}}(\mathbf{x}) \leq B, \quad (1)$$

where X is a box in d dimensions, f is observable only through costly evaluations, and the budget B books both the initial design and every evaluation the search triggers. The contributions of this work are as follows.

- I. A stigmergic communication protocol whose pheromone is a natural-language artifact: LLM condensations of a capped collective memory compile into an explicit chemoattractant field that biases swarm movement.
- II. A cost-aware agent architecture coupling the pheromone field with a Gaussian-process surrogate: Acquisition-gated true evaluations, health-based reproduction, and pheromone-biased elimination and dispersal, all charged to one budget.
- III. A hallucination guard that scores agreement between LLM directional cues and surrogate gradients and falls back to classical tumble behavior on disagreement, backed by hard token-budget caps and schema-level failure handling.
- IV. A token economics analysis with a per-run cost model: Measured prompt and completion token counts, dollar cost, and wall-clock overhead against BO at equal budget.
- V. A rigorous empirical study on twelve problems under three budget tiers, 150, 400, and 900 true evaluations: 30 runs per cell, fourteen methods including four token-matched language-model optimizers, survival-analysis inference with Kaplan-Meier curves, Holm-corrected log-rank tests, and Cox hazard ratios, Cliff's delta [19] effect sizes, component-wise ablations with cadence and swarm-size sweeps, per-run token accounting, bursty-dropout and latency stress, and a controlled distributional-shift study.

The itinerary follows. Section 2 reviews BFO, LLM-based optimization, and surrogate-assisted search, then states the gap this work fills. Section 3 details the system architecture of GAS-BFO. Section 4 specifies the communication protocol: Its message schemata, prompting strategy, information-theoretic reading, and failure modes. Section 5 presents the algorithm and its cost model. Section 6 reports the empirical ledger: quality per budget, ablations, token economics, and communication stress. Section 7 studies robustness under distributional shift. Section 8 discusses scope, limitations, and validity threats; Section 9 concludes and charts future directions.

2 | Background

This section places GAS-BFO at the junction of three literatures. It first traces BFO and pinpoints the limitations that matter in expensive settings. It then reviews how LLMs serve as search operators and

optimizers. Finally, it summarizes surrogate-assisted optimization under tight budgets. The closing argument: no existing system combines stigmergic communication with generative agents in a budget-aware manner.

2.1 | Bacterial Foraging Optimization: Lineage and Limitations

Passino [8] introduced BFO as a biomimicry of *Escherichia coli* foraging: Bacteria sense local chemoattractant gradients through temporal comparisons, run while conditions improve, tumble otherwise, and periodically reproduce and disperse. The algorithm mirrors these operators directly. Chemotaxis implements biased random walks. Reproduction duplicates the healthiest half of the population. Elimination–dispersal relocates agents probabilistically to model environmental cataclysms. Liu and Passino [20] later supplied stability analyses for the resulting distributed optimization dynamics. The survey by Das et al. [21] consolidated an application ledger spanning controller tuning, power systems, and image processing. The enduring appeal of BFO is architectural: A population of simple agents, local rules, and no central model.

Three limitations settle BFO's fate on expensive problems: 1) classical chemotaxis estimates gradients by probing pairwise position differences, and each probe bills multiple evaluations per agent per step; at $B = 200$, probing alone can exhaust the budget, 2) the original swarming term communicates only coarse numeric attraction and repulsion between agents, and the algorithm is otherwise memoryless. Evidence gathered by one bacterium reaches the swarm only through selection, never as an interpretable summary, 3) the operator schedule, such as chemotactic step counts, reproduction frequency, and dispersal probability, is hand-set, and no component prices the economic value of an evaluation. The no-free-lunch theorems rule out any fixed operator schedule that dominates across problem classes [22]. The fix must let the search adapt to what the accumulated evidence actually says.

2.2 | Large Language Models as Search Operators and Optimizers

The optimization community has recently hired LLMs in three roles. As an optimizer, the model reads a trajectory of candidate-value pairs and proposes the next candidate. OPRO demonstrated that instruction-tuned models optimize small continuous and combinatorial tasks purely in context, with the prompt serving as population memory [18]. As an operator, LLMs supply evolutionary variation: Liu et al. [23] showed that an LLM performing crossover and mutation over continuous vectors stays competitive with classical operators on low-dimensional benchmarks, and LLaMEA uses an LLM to generate and refine entire metaheuristic algorithms [24]. As a program searcher, FunSearch evolves code with an LLM mutator and has produced new mathematical constructions [25], while Evolution of Heuristics automates heuristic design for combinatorial problems [26].

These deployments inherit a specific toolkit from the prompting literature. Chain-of-thought prompting elicits intermediate reasoning steps [27]. Self-consistency aggregates several sampled reasoning paths into a majority answer and improves reliability [28]. Agentic scaffolds interleave reasoning with tool calls, and surrogate queries are a natural instance [29]. Two assets make LLMs attractive for expensive optimization: They interpolate patterns across heterogeneous evidence, and they emit hypotheses in a form humans and other agents can consume. Two liabilities argue for caution: outputs are stochastic, and the model can assert structure the input evidence does not support.

No existing system, however, runs the LLM as a communication channel inside a budget-limited search population. OPRO-style optimizers replace the population with a single context window. Operator-style systems insert the LLM where variation belongs, so its influence stays confined to proposing genotypes. In both, token spend scales with population activity in ways the literature rarely reports. Neither design audits the model's claims against an independent numerical model before acting on them. This work states the design question in deliberately inverse terms. Keep the classical population dynamics. Pay the LLM only for what it does uniquely well: Read a large, messy log of observations and write a compact, checkable summary of the landscape.

2.3 | Surrogate-Assisted Optimization under Tight Evaluation Budgets

BO is the reference methodology for budget-limited black-box search [4], [5]. A Gaussian Process (GP) maintains a posterior over the objective given all evaluated points [30], with a nugget term guarding against numerical micro-noise [31]. An acquisition function converts the posterior into the next evaluation decision, EI [2], probability of improvement, or an upper-confidence-bound score [32]. Read at the level of information flow, the GP is a shared memory of all observations, and the acquisition rule arbitrates globally which observation is worth buying next. Implementations at scale include tree-structured Parzen estimators [33], the random-forest smoothers of SMAC [34], and the surrogate-modeling practice long standard in engineering design [1], [35].

The standard GP formalism supplies the machinery. Given the evaluated inputs booked so far and their responses, the posterior mean and standard deviation at a candidate x are

$$\mu^{(t)}(x) = k^{(t)}(x)^\top (K^{(t)} + \sigma_n^2 I)^{-1} y^{(t)}, \quad (2)$$

$$\sigma^{(t)}(x) = \sqrt{k(x, x) - k^{(t)}(x)^\top (K^{(t)} + \sigma_n^2 I)^{-1} k^{(t)}(x)}, \quad (3)$$

where the kernel matrix K has entries $k(\cdot, \cdot)$, the nugget added to the diagonal absorbs micro-noise and numerics [31], and the superscript indexes the refit after the t -th accepted evaluation. These two moments drive the acquisition policy and, in GAS-BFO, also the movement rule and the hallucination guard. The line's assets are well documented. So are its limits under multimodality and moderate dimension, where a single centralized posterior over-concentrates the search. The position taken here does not retire GP-based search. It holds that a population of agents adds value if, and only if, a summary richer than the raw posterior coordinates it.

2.4 | Gap: No Stigmergic Communication with Generative Agents

Grassé's study of termite construction supplies the classic account of stigmergy: The emerging structure itself coordinates the workers [10]. Wilson's [36] sociobiological work traced how chemical signals mediate comparable coordination across social insects. Artificial swarm intelligence adopted the pattern. Beni and Wang [37] coined the term for cellular robotics. Colomi et al. [38] encoded pheromone trails over discrete edges. The ensuing literature built a rich theory of indirect coordination [11], including non-pheromone stigmergic mechanisms such as the advertising dances of artificial bee colonies [39]. In every one of these systems, the communication medium is numeric, and design time fixes its semantics.

LLM-based multi-agent infrastructures such as AutoGen issue software agents rich conversational channels [40]. Generative agent simulations demonstrated believable collective behavior built on memory and reflection [17]. But these frameworks are not optimizers. They ship no evaluation-budget semantics, acquisition discipline, or guards against fabricated guidance. *Table 1* positions GAS-BFO against the adjacent lines of work. No published system combines stigmergic communication, a generative pheromone medium, explicit evaluation budgets, hallucination guards, and reported token economics in a single optimizer. That combination is the contribution of this paper.

Table 1. Positioning of GAS-BFO relative to adjacent research lines.

Approach Class	References	Stigmergic Communication	Generative Pheromone	Budget-Aware	Hallucination Guards	Cost Reporting
BFO	[8], [20], [21]	No (local swarming term only)	No	No	n/a	No
ACO	[11], [38]	Yes (numeric trails)	No	No	n/a	No
Particle Swarm Optimization (PSO)	[7]	No (direct neighborhood sharing)	No	No	n/a	No
BO / EGO	[2], [5]	No (centralized posterior)	No	Yes	n/a	No
TPE / SMAC	[33], [34]	No (centralized model)	No	Yes	n/a	No
LLM as optimizer (OPRO)	[18]	No (single context window)	No	Partial (prompt length)	No	Partial
LLM as evolutionary operator	[22], [23], [25]	No (operator-level use)	No	No	No	Rare
Generative agent simulations	[17], [40]	Yes (conversational)	Partial (text)	No	No	No
GAS-BFO (this work)	—	Yes (LLM pheromone over collective memory)	Yes	Yes (acquisition-gated)	Yes (surrogate cross-check)	Yes (tokens, dollars, wall-clock)

*The marker n/a denotes method classes with no generative component, for which hallucination guards are not applicable.

3 | System Architecture of GAS-Bacterial Foraging Optimization

GAS-BFO ships as six cooperating subsystems rather than a monolithic loop: An agent layer, a collective memory, a pheromone condensation service, a movement rule, an evolutionary cycle, and a surrogate-and-budget layer. Four design principles govern the assembly: 1) communication must summarize, not broadcast: Agents never exchange raw vectors, only schema-conformant observations, 2) generative capacity is reserved for interpretation: Authoring observations is programmatic and deterministic, which keeps the token bill small and the logs reproducible, 3) numeric evidence must audit every LLM output before it can move an agent, and 4) the evaluation budget is a first-class citizen of the ledger: A single manager admits true evaluations, and no subsystem can bypass it.

3.1 | Agent Design: Bacteria as Generative Agents

Each bacterium is a lightweight generative agent in the sense of multi-agent systems [41]: An autonomous entity with persistent state, a perception-action cycle, and the ability to contribute to shared knowledge [17]. Its state at iteration t holds three items: a position $x_i(t)$ in the search box, a scalar health $h_i(t)$, and a private memory slice. The slice keeps the agent's own last ten positions and their observed or predicted values. Position is the decision variable. Health accrues merit for the reproduction operator. The private slice lets the agent price its local trend without querying the collective memory.

The LLM's role is deliberately asymmetric. Agents never call the model themselves. A programmatic layer fills each observation post from agent state: Coordinates rounded to three decimals, an evaluated fitness or a flagged surrogate estimate, a trend label, and a confidence value. Authoring therefore costs zero tokens and is bit-reproducible. All generative work concentrates in a single condensation service the whole swarm shares (Section 3.3). The division of labor follows the economics. Thirty agents posting free-form text at every iteration would multiply the token bill by orders of magnitude. They would add nothing beyond what the schema carries. The bacterium stays generative in consequence, not in authorship: language artifacts issued on behalf of the whole swarm shape its trajectory.

3.2 | Collective Memory as A Shared Substrate

The collective memory M is an append-only ledger of structured observations, the substrate on which the pheromone field grows. Each entry carries a position summary (the rounded coordinate vector) and the evaluated fitness if the position was ever truly evaluated; otherwise, the surrogate mean flagged as an estimate.

It also carries a local trend label from the set {improving, flat, worsening}, computed from the posting agent's private slice. A confidence value equals one minus the normalized surrogate standard deviation at that position. A short templated note completes the entry, such as 'moved west of previous post; worse than expected'. Every entry is stamped with the iteration and tagged with the posting agent's identifier.

A 500-entry cap with first-in-first-out eviction bounds the log, and the cap earns its keep three ways. It bounds the prompt length of every condensation call, the dominant line item of the token budget. It implements evaporation: Stale evidence leaves the context exactly as physical pheromone decays. And it keeps the swarm's attention on recent landscape structure, which matters under drift (Section 7). The condensation prompt draws on two read views. A recency window holds the newest 60 entries. A salience view holds the 40 entries with the largest absolute fitness change relative to their posting agent's previous post.

3.3 | Large Language Models-Mediated Pheromone Field

Every $L = 10$ iterations, the condensation service reads the memory views and tasks the LLM with analyzing the colony's search. The output is not free prose but a schema-conformant set of at most five region summaries. Each region r carries a center c_r , a lengthscale ℓ_r , a weight w_r in the unit interval, and a one-sentence rationale. A global hypothesis string and a binary confidence flag complete the emission (Section 4.1). Self-consistency sampling with $k = 5$ draws (Section 4.2) produces five candidate condensations. Region centers are clustered across the five draws, weights are averaged, and a region survives only when a majority of the draws proposed it. The aggregated regions define the chemoattractant field:

$$\Phi^{(t)}(x) = \sum_{r=1}^{R^{(t)}} w_r^{(t)} \exp\left(-\frac{\|x - c_r^{(t)}\|^2}{2\ell_r^2}\right), \quad (4)$$

where $R^{(t)}$, at most five, counts the majority-supported regions and the lengthscale ℓ_r sets each region's radius of influence. *Eq. (4)* has the functional form of a mixture-of-Gaussians surface, but linguistic consensus rather than maximum likelihood prices its parameters. The LLM chooses where the swarm's evidence says promise lies; the vote across self-consistency samples decides how strongly. The field is attractive by construction. Repulsion arises implicitly: Regions unsupported by recent memory are simply not re-proposed, and FIFO eviction erases their contribution. The field is not a posterior mean, not a fitted response surface, and not authoritative. It is a compact, contestable hypothesis about landscape structure whose every parameter traces back to ledger entries, exactly what the hallucination guard needs to audit it (Section 4.4).

3.4 | Chemotaxis Movement Rule

Between condensations, each agent runs a biased run-and-tumble cycle. The run direction blends the surrogate's descent direction with the ascent direction of the pheromone field:

$$d_i^{(t)} \propto -\nabla \hat{f}^{(t)}(x_i^{(t)}) + \gamma \nabla \Phi^{(t)}(x_i^{(t)}), \quad (5)$$

where the surrogate predictor $\hat{f}^{(t)}$ is the GP posterior mean of *Eq. (2)* and $\gamma = 0.5$ balances evidence-driven descent against colony-level attraction. Both gradients are evaluated analytically and bill a negligible fraction of a true evaluation. Step length decays geometrically from 8% to 1% of the box diagonal over the run, and the boundary reflects moves that would leave the box.

Tumbling modulates the mixture. The tumble probability climbs whenever the guidance an agent receives disagrees with its local surrogate evidence:

$$p_{\text{tum}}(i) = p_0 + (p_{\text{max}} - p_0)(1 - a_i^{(t)}), \quad (6)$$

where $a_i^{(t)}$ is the agreement statistic between the LLM's directional cue and the surrogate gradient at x_i , defined in *Eq. (12)*, with $p_0 = 0.1$ and $p_{\text{max}} = 0.7$, guidance and surrogate in agreement purchase long,

directed runs. Conflict reverts the agent to the exploration profile of classical BFO: frequent tumbling, trusting only local evidence. The movement rule is fail-safe by construction. The worse the coordination signal, the closer the behavior to the unguided baseline.

3.5 | Reproduction, Elimination–Dispersal

Every L iterations, immediately after condensation, the swarm executes the evolutionary half of BFO. Health pays agents whose observations beat the surrogate's expectation, the only form of surprise the system can meter without spending extra evaluations:

$$h_i^{(t+1)} = \lambda h_i^{(t)} + \max\left(0, \mu^{(t)}(x_i^{(t)}) - f(x_i^{(t)})\right), \quad (7)$$

with decay $\lambda = 0.9$, so merit fades unless re-earned. Reproduction draws parents from a temperature-controlled softmax of health rather than duplicating a fixed top half:

$$p_i^{\text{rep}} = \frac{\exp(\beta h_i^{(t)})}{\sum_{i=1}^N \exp(\beta h_i^{(t)})}, \quad (8)$$

with $\beta = 1$. Children inherit the parent position perturbed by small isotropic jitter and reset their health, keeping $N = 30$ constant. The softmax softens Passino's deterministic split and cuts the odds that one lucky early evaluation permanently captures the gene pool.

Elimination–dispersal then relocates each agent independently with probability $p_{\text{ed}} = 0.25$. Classical BFO redraws uniformly; GAS-BFO samples restarts from a density biased by the current pheromone field and mixed with a uniform component for insurance:

$$p_{\text{restart}}^{(t)}(x) = 0.5 \frac{\Phi^{(0)}(x) + \varepsilon}{Z^{(t)}} + 0.5 \frac{1}{\text{vol}(x)}, \quad (9)$$

where $Z^{(t)}$ normalizes the shifted field and ε is a small floor. Dispersal thereby re-colonizes the regions the swarm's own language summaries currently favor, while the uniform half preserves global coverage (stigmergic memory deciding, explicitly, where the colony re-seeds after a cataclysm).

3.6 | Surrogate Layer and Evaluation Budget Manager

A single GP regression over all true observations is refit after every accepted evaluation, with a Matérn 5/2 kernel with automatic relevance determination, standardized inputs, and a nugget [29], [30]. Three consumers bill the surrogate: The movement rule through the gradient of the posterior mean, the acquisition gate, and the hallucination guard. Evaluations are the only scarce resource, and one manager owns them all.

At each iteration, every agent proposes its next position, and the manager prices EI over the proposals together with the LLM's focus point when one is present. A proposal is admitted only under the acquisition-gated rule:

$$n_{\text{eval}}^{(t+1)} = n_{\text{eval}}^{(t)} + \mathbf{1} \left[\max_{x \in C^{(t)}} E^{(t)}(x) \geq \tau^{(t)} \text{ or } c_{\text{LLM}}^{(t)} = 1 \right], \quad (10)$$

where $C^{(t)}$ is the candidate set, EI is computed for the current incumbent following [2], and the threshold $\tau^{(t)}$ decays linearly in standardized objective units with the iteration index, forcing late-run frugality. A stall guard admits the best-EI candidate whenever five consecutive iterations pass without an evaluation, so the loop always progresses toward budget exhaustion. The condensation's confidence flag c_{LLM} acts as an exploration override. When the LLM marks a high-potential region that no recent evaluation has tested, the region center is evaluated even if EI is momentarily low. In practice, the manager admits about one proposal per iteration, so a budget of $B = 200$ yields roughly 210 iterations and about 21 condensation events. The

initial Latin hypercube design of 10 points is charged to the same ledger, and the manager refuses every request once the budget runs dry.

4 | The Communication Protocol

The protocol is the centerpiece of the system, and this section specifies it in full. The message schemata flow through the channel. The prompting and aggregation strategy mints the pheromone. An information-theoretic reading prices what the pheromone yields. Failure modes and their guards close the account. The design goal holds throughout: A channel expressive enough to change swarm behavior, constrained enough to be auditable, and cheap enough to operate inside evaluation budgets of a few hundred true calls.

4.1 | Message Schemata

Two artifact families flow through the channel. The agents' deterministic authoring layer writes posts into the collective memory; the condensation service emits structured LLM output. Both are JSON-like records with fixed fields, listed in *Table 2*. The post schema separates facts from interpretation: Position, fitness, and confidence are numeric facts drawn from state, while the trend label and the short note are weak interpretations produced by fixed rules. The emission schema separates geometry from narrative: Centers, lengthscales, and weights are executable and compile directly into the field of *Eq. (4)*; the rationale and hypothesis strings are advisory and never actuated without clearing the guard.

The rationale field deserves comment. It spends tokens without ever entering a computation, yet the ablations show the system weakens without it (Section 6.5). Free-text rationales anchor the model's own region judgments across the k samples, functioning as an intra-ensemble consistency device. They also make every field human-auditable after the fact. The focus point and the confidence flag serve one purpose only: Let the language channel order an evaluation the numeric machinery would not order. In that sense, the protocol is bidirectional rather than merely broadcast.

Table 2. Message schemata for the two directions of the communication channel.

Direction	Field	Type	Semantics	Example
Agent post	Agent_id, iter	Integers	Identity and iteration stamp	17, 140
Agent post	Pos	Vector, 3-decimal	Position summary in the unit box	0.42, 0.11, 0.87, ...
Agent post	f_eval / f_hat	Float / Flagged null	Evaluated fitness; else surrogate mean marked as estimate	1.204 / 2.31 (est.)
Agent post	Trend	Enum	Local trend from the private slice	Improving
Agent post	Conf	Float in [0,1]	One minus normalized surrogate std	0.62
Agent post	Note	String, at most 12 words	Templated local remark	Moved north-east; worse than expected
LLM emission	Regions[r].center	Vector	Region center in the unit box	0.31, 0.28, 0.90, ...
LLM emission	Regions[r].len	Float	Region lengthscale	0.12
LLM emission	Regions[r].weight	Float in [0,1]	Consensus region weight	0.55
LLM emission	Regions[r].why	String	One-sentence rationale	Five improving posts cluster here
LLM emission	Hypothesis	String	Global landscape hypothesis	Curved valley along dims 2–4
LLM emission	Focus	Vector or null	Untested high-potential point	0.44, 0.29, 0.85, ...
LLM emission	c_LLM	Binary	Confidence flag for forced evaluation	1

*Posts are authored programmatically and cost no tokens; emissions are produced by the condensation LLM under a strict JSON contract.

4.2 | Prompting Strategy and Robustness

Five ordered sections assemble the condensation prompt, described here in text rather than reproduced verbatim. First, a role header casts the model as an analyst serving a bacterial colony, with no knowledge of the objective's name or origin. Benchmark identity is never disclosed, which also blunts any memorization advantage (Section 8). Second, a task header states the dimension, the box bounds, and the evaluations spent and remaining. Third, the memory digest renders the recency and salience views as compact lines carrying identifier, position, value or flagged estimate, trend, and confidence. Fourth, a statistics block lists the incumbent best, the five best evaluated points, and the per-dimension ranges of the salience view. Fifth, an output contract fixes the JSON schema of *Table 2*, caps regions at five, and instructs the model to mark low confidence explicitly rather than guess.

Robustness is layered, not hoped for. A schema violation triggers a single retry that embeds the validator's error message; a second failure reuses the previous field unchanged, so the swarm never acts on malformed output. Self-consistency draws $k = 5$ samples at temperature 0.7 to obtain genuinely different region hypotheses, then aggregates by majority vote on centers and mean weights [28]. All other decoding runs at temperature 0, and the model version is pinned per run. A hard token cap per condensation event aborts oversized generations and preserves the last valid field. These are deliberately conservative engineering choices. The interesting failures are semantic, and the next subsection interrogates them.

4.3 | Information-Theoretic View

The pheromone field admits a compression reading. At condensation time, the memory holds up to 500 entries of about $2d$ numbers each; on the benchmarks used here, that amounts to thousands of numeric observations. The emission amortizes this into at most five regions, roughly $5(2d + 2)$ numbers plus a few sentences, a compression of about two orders of magnitude. The field is a sufficient statistic for the swarm's movement only to the extent that the LLM captures real regularities; otherwise *Eq. (4)* is an expensive way of drawing random bumps. The movement-relevant information the channel carries is therefore estimated in the style of classical mutual-information estimators [42]:

$$\hat{I}(\Phi^{(t)}; \Delta f^{(t)}) = \sum_{a \in A} \sum_{b \in B} \hat{p}(a, b) \log \left(\frac{\hat{p}(a, b)}{\hat{p}(a)\hat{p}(b)} \right), \quad (11)$$

where a indexes whether an agent's realized step aligned with the pheromone gradient and $\Delta f^{(t)}$ is the signed realized improvement of that step relative to the surrogate expectation, with both variables discretized into the bins A and B . On Hartmann-6D, the estimate averaged 0.42 bits for the full protocol against 0.09 bits when regions were randomly aggregated (Section 6.5). The language channel carries roughly four to five times the movement-relevant information of a random summary of the same memory. The estimate is a diagnostic of signal, not proof of sufficiency. A formal information-theoretic account of stigmergic compression remains open and is flagged in Section 9.

4.4 | Failure Modes and Guards

The protocol's characteristic failure is a hallucinated gradient: The condensation asserts, with fluent rationale, a region the evaluated evidence does not support. The guard operationalizes disagreement as an angle. Let $g_{\text{LLM},i}$ be the pull direction the field induces at x_i , the pheromone term of *Eq. (5)*. The agreement statistic is the cosine similarity between this cue and the surrogate's direction of descent:

$$a_i^{(t)} = \frac{\left\langle g_{\text{LLM},i}^{(t)}, -\nabla \hat{f}^{(t)}(x_i^{(t)}) \right\rangle}{\|g_{\text{LLM},i}^{(t)}\| \|\nabla \hat{f}^{(t)}(x_i^{(t)})\| + \varepsilon}, \quad (12)$$

When $a_i^{(t)}$ falls below $a_{\text{min}} = 0.2$, the pheromone term is dropped for that agent and iteration, and movement falls back to the plain BFO tumble. The same statistic feeds the tumble probability of *Eq. (6)*, so

mild disagreement already buys exploration. Removing the guard was measurably harmful: The Hartmann-6D gap rose from 0.061 to 0.118 (Section 6.5). That is the empirical sense in which unchecked hallucination is a first-class optimization hazard rather than a curiosity [43]. On Hartmann-6D, the guard fired on 3.1% of agent-iterations on average, an acceptable premium for the value the channel returns.

Four further guards complete the failure catalogue. Degenerate outputs, in which a single region absorbs all weight, are clipped at weight 0.6 and re-normalized. Winsorizing the fitness values that enter the digest prevents memory poisoning by outliers. Echo loops are structurally impossible: Prior emissions never feed back into the prompt, the condensation always reasons from raw observations only, and the FIFO cap bounds field staleness. The token budget itself is the final guard. Cumulative API cost that reaches the run-level cap suspends condensation, and the last valid field persists, degrading gracefully toward surrogate-assisted BFO. Every guard stays silent in the logs unless it fires, so post-hoc audits can tally hallucination events per run.

5 | Algorithm and Cost Model

This section assembles the subsystems into a single algorithm, then prices it. The cost model separates three currencies the literature often conflates: True evaluations, tokens, and wall-clock time. Making all three explicit is what supports an honest statement of when the method earns its deployment.

5.1 | Complete Pseudocode

Algorithm 1 presents the complete procedure. The outer loop alternates chemotaxis at every iteration with condensation, reproduction, and dispersal every $L = 10$ iterations; the budget manager is the only site that calls the true objective. All stochastic choices are seeded per run. The default configuration deploys $N = 30$ agents, memory capped at 500 entries, $k = 5$ self-consistency samples, and dispersal probability 0.25. Movement parameters run $\gamma = 0.5$, $\lambda = 0.9$, $\beta = 1$, $a_{\min} = 0.2$, $p_0 = 0.1$, and $p_{\max} = 0.7$. The run starts from an initial design of 10 Latin hypercube points and a budget of $B = 200$ true evaluations.

Algorithm 1. GAS-BFO.

```

1: Input: f, box X, budget B,  $N = 30$ ,  $L = 10$ ,  $k = 5$ ,  $p_{\text{ed}} = 0.25$ , caps
2:  $D = 10$ -point Latin hypercube in X; evaluate f;  $n_{\text{eval}} = 10$ 
3: fit GP surrogate ( $\mu$ ,  $\sigma$ ) on D
4: initialize agents  $x_1..x_N$  in X; health  $h_i = 0$ ; memory  $M = \{\}$ 
5:  $\Phi = \text{empty field}$ ;  $c_{\text{LLM}} = 0$ 
6: for  $t = 1, 2, \dots$  while  $n_{\text{eval}} < B$ :
7:   for each agent  $i$ : post structured observation to M (FIFO cap 500)
8:   if  $t \bmod L = 0$ : // pheromone condensation
9:     for  $s = 1..k$ :  $r_s = \text{LLM\_condense}(\text{digest}(M), \text{stats}(D))$ 
10:     $\Phi = \text{majority\_vote}(r_1..r_k)$ ; extract focus point,  $c_{\text{LLM}}$ 
11:   for each agent  $i$ :
12:      $g_{\text{LLM},i} = \text{pheromone pull of } \Phi \text{ at } x_i$ 
13:      $a_i = \cosine(g_{\text{LLM},i}, -\text{grad}_{\mu}(x_i))$  // Eq. (12)
14:     if  $a_i < 0.2$ :  $d_i = \text{random unit vector}$  // guard
15:     else:  $d_i = -\text{grad}_{\mu}(x_i) + 0.5 * \text{grad}_{\Phi}(x_i)$  // Eq. (5)
16:     with probability  $p_{\text{tum}}(i)$ :  $d_i = \text{random unit vector}$  // Eq. (6)
17:      $x_i = \text{reflect}(x_i + \eta_t * d_i, X)$ 
18:    $C = \{x_1..x_N\}$  plus focus point;  $x^* = \text{member with max EI}$ 
19:   if  $\text{EI}(x^*) \geq \tau_t$  or  $c_{\text{LLM}} = 1$ : // Eq. (10)
20:      $y^* = f(x^*)$ ;  $n_{\text{eval}} = n_{\text{eval}} + 1$ ;  $D = D + (x^*, y^*)$ 
21:      $h_i^* = 0.9 * h_i^* + \max(0, \mu(x^*) - y^*)$  // Eq. (7)
22:   refit GP on D; update  $\mu$ ,  $\sigma$ 
23:   if  $t \bmod L = 0$ :
24:     reproduction: parents  $\sim \text{softmax}(h)$ ; replace losers // Eq. (8)
25:     for each agent  $i$  with probability 0.25: redraw  $x_i$  from Eq. (9) density
26: return best evaluated  $(x, f(x))$  in D

```

Three remarks clarify the pseudocode. The LLM is called exactly at line 9 and nowhere else; given the seeds, every other subsystem is numeric and deterministic. In the ASCII listing, Φ denotes the chemoattractant

field of Eq. (4), μ and σ the surrogate moments of Eqs. (2) and (3), and τ_t the acquisition threshold of Eq. (10). The loop admits at most one evaluation per iteration, which is what makes token cost and evaluation cost commensurable. Replace the condensation service with the identity, Φ always empty, and *Algorithm 1* reduces to a surrogate-assisted BFO, precisely the stripped floor priced by the ablations of Section 6.5.

5.2 | Complexity and Token Economics

GP refits dominate numerical complexity and scale cubically in the number of true observations. With n at most 200, a refit costs well under a second, and the gradients of the posterior mean and of the field are analytic. The distinctive overhead is linguistic. With about 21 condensation events per run and $k = 5$ samples each, a run performs roughly 100 API calls. The run-level cost model is:

$$C_{\text{run}} = \sum_{t=1}^T (c_p n_p^{(t)} + c_c n_c^{(t)}), \quad (13)$$

where $n_p^{(t)}$ and $n_c^{(t)}$ are the prompt and completion tokens of call t and c_p, c_c are the provider's unit prices. Hartmann-6D billed a run, on average, 387k prompt tokens and 61k completion tokens, a prompt-to-completion ratio of about 6:1, since the digest dwarfs the emission. At prevailing commodity API prices, this maps to an estimated US\$2.1–3.4 per run. Mean optimizer wall-clock overhead, excluding true evaluations, ran 21.4 minutes (2.3 times the 9.3 minutes of BO-EI at the same budget), with API latency, not local computation, dominating the difference.

Scale decides whether these overheads matter. In cheap-evaluation settings they are pure loss; in expensive ones they are minor. At the assumed floor of 30 seconds per true evaluation, the 400-evaluation reference tier alone costs at least 200 minutes of wall-clock, so the LLM adds at most about 14% to end-to-end time, and the starved 150-evaluation tier about 25%. A few dollars of guidance sit beside the compute the evaluations themselves burn. Quality per dollar prices the tradeoff explicitly, the relative gap reduction purchased per unit of API spend:

$$Q_{\text{pd}} = \frac{1}{C_{\text{run}}} \cdot \frac{g_{\text{base}} - g_{\text{ous}}}{g_{\text{base}}}, \quad (14)$$

where g denotes the final mean gap to the known optimum and the baseline is BO-EI, whose API cost is zero. On Hartmann-6D, Q_{pd} came to approximately 0.27 per dollar: A 72.9% gap reduction purchased for two to three dollars. One caveat belongs in the ledger. The $k = 5$ samples of a single condensation are independent and could be issued concurrently, cutting the latency overhead by roughly half; the reported figures are sequential, by conservative choice.

6 | Evaluation under Budget Constraints

The evaluation prices the pheromone channel under the one constraint that matters in expensive optimization: The evaluation budget itself. Three budget tiers, 150, 400, and 900 true evaluations, span the regimes a real campaign actually buys, from a starved pilot study to a mid-sized tuning run. Twelve problems, ten mathematical benchmarks and two simulated engineering cases, meet fourteen methods, nine established references and four language-model optimizers newly admitted to the race. The ledger reports final quality per tier, time-to-target survival, ablations, token economics, communication stress, and distributional shift, in that order.

6.1 | Benchmark Functions and Budgets

Ten mathematical benchmarks span the regimes that stress budget-limited search [44]. Hartmann-3D and Hartmann-6D are low-dimensional multimodal landscapes with known optima. Michalewicz in ten dimensions adds banana-shaped valleys that punish gradient bias. Rastrigin in ten dimensions, rotated by a

fixed orthogonal map, carries dense periodic local optima; Ackley in ten and twenty dimensions grades landscape breadth; Rosenbrock in ten dimensions couples a curved valley with a smooth floor; Levy in ten dimensions carries mild multimodality around a global valley; Zakharov in twenty dimensions, shifted off the origin, and Dixon-Price in twenty dimensions complete the high-dimensional wing. All functions are minimized on their standard domains and rescaled to the unit box for the agents, with the known optimum used to report gaps.

Two simulated engineering cases complete the suite, both rebuilt for this study. The first tunes six hyperparameters of a gradient-boosted classifier [45]: Learning rate on a log scale, maximum depth, minimum child weight, subsample ratio, column subsample ratio, and L2 regularization, a genuinely mixed search space of continuous, integer, and logarithmic dimensions. The objective is five-fold cross-validated log-loss, and each configuration is billed a thirty-second evaluation cost with training parallelism disabled. The second is an aerodynamic shape problem in the surrogate spirit of [35]: Eight Hicks-Henne bump amplitudes deform the RAE 2822 profile at transonic conditions [46], and the ground truth is a deterministic drag-like response reproducing the shock-induced drag rise and curvature typical of such flows, a simulated CFD-like surrogate rather than an actual solve, a limitation noted in Section 8.

Every problem runs at all three budget tiers, $B = 150, 400$, and 900 true evaluations, of which the first ten fund a shared Latin hypercube design. The tier structure is the study's spine: what a method buys with tripling budgets separates the families that amortize structural knowledge, the language-model lineage, from those that convert budget linearly, the Bayesian lineage. *Table 3* records the suite, the tiers, and the reporting conventions.

Table 3. Benchmark suite, budget tiers, and reporting conventions.

Problem	Kind	Dim.	Tiers (evals)	Metric
Hartmann-3D	Mathematical	3	150 / 400 / 900	Gap to optimum
Hartmann-6D	Mathematical	6	150 / 400 / 900	Gap to optimum
Michalewicz-10D	Mathematical	10	150 / 400 / 900	Gap to optimum
Rastrigin-10D (rotated)	Mathematical	10	150 / 400 / 900	Gap to optimum
Ackley-10D	Mathematical	10	150 / 400 / 900	Gap to optimum
Ackley-20D	Mathematical	20	150 / 400 / 900	Gap to optimum
Rosenbrock-10D	Mathematical	10	150 / 400 / 900	Gap to optimum
Levy-10D	Mathematical	10	150 / 400 / 900	Gap to optimum
Zakharov-20D (shifted)	Mathematical	20	150 / 400 / 900	Gap to optimum
Dixon-Price-20D	Mathematical	20	150 / 400 / 900	Gap to optimum
XGB-tuning	Engineering (mixed space)	6 (3 types)	150 / 400 / 900	Log-loss gap
RAE-2822-airfoil	Engineering (surrogate)	8	150 / 400 / 900	Drag-count gap

Mathematical benchmarks report the gap to the known optimum; the tuning case reports the log-loss difference to the best configuration found by any method at the 900-evaluation tier; the airfoil case reports the drag-count difference to its reference optimum.

6.2 | Reference Methods under Budget

Fourteen methods run the race. Nine established references span the two classical lineages: Bayesian Optimization with Expected Improvement (BO-EI) [2] as the reference standard, probability of improvement (BO-PI), and an upper-confidence-bound policy with $\beta = 2$ (BO-UCB) [32] complete the acquisition family; TPE [33] and SMAC3 with its random-forest surrogate [47] stand in for scalable model-based competitors; a budget-restricted CMA-ES, population 10 for the tier-dependent generation count, carries the derandomized evolution-strategy flag [9]; uniform random search [48] prices the floor; and vanilla BFO, Passino's parameterization scaled to the tier [8], plus the BFO+GP hybrid that shares GAS-BFO's surrogate, gating, and population dynamics minus the language channel, bracket the swarm lineage. TPE and SMAC3 natively support the mixed search space of the tuning case; the remaining continuous methods run on a unit-box encoding with inverse transforms for integer and logarithmic dimensions.

Four language-model optimizers join as new admissions, each raced with token-matched accounting on the same pinned model and API as GAS-BFO. OPRO, optimization by prompting [18], asks the model to propose candidate solutions from a textual history of evaluated triples; it is the cleanest published instance of the LLM-as-optimizer lineage, its strength is a compact prompt that scales to any dimension, and its limitation is a history that mixes promising and poor triples without a condensation step, so the inductive bias the swarm buys structurally must be bought by the model in-context. Its relevance to RQ1 is direct: OPRO prices what a single-thread language channel achieves, against which the swarm's thirty-agent stigmergy is the treatment.

FunSearch [25] evolves programs with an LLM as a mutation operator, searching the space of heuristics rather than the space of solutions; its strength is compositional discovery on problems with exploitable symbolic structure, its limitation is a wide search space that a few hundred evaluations cannot fund on numeric landscapes, and its inclusion prices whether program-space search transfers to budget-constrained continuous optimization at all. EoH [26] evolves short heuristic descriptions in a language-model loop with both crossover and mutation prompts; its strength is cheap iteration over explicit heuristic code, its limitation is the same budget arithmetic, and it prices the middle ground between OPRO's direct proposal and FunSearch's full program search. ReEvo [49] closes the set with reflective evolution: A population of heuristics undergoes verbal reflection and differential mutation; its strength is the strongest published record on combinatorial and continuous suites among LLM hyper-heuristics, its limitation is token appetite, and it is admitted as the strongest available representative of the reflective lineage, the closest published cousin of GAS-BFO's own condensation loop.

All fourteen methods share the same initial design, redrawn per seed but identical across methods within a seed, and the same tier accounting; the LLM quartet shares GAS-BFO's model version, decoding parameters, and API billing so that token comparisons are like-for-like. No method receives problem-specific tuning beyond its source defaults, a conservative choice that favors simplicity over last-percent performance, and every prompt, configuration, and transcript is archived for release.

6.3 | Run Accounting and Reproducibility

Each algorithm-problem-tier cell runs thirty times with seeds shared across methods, so the design is fully factorial over algorithm, problem, tier, and run, 5,040 runs per method in total. The reported solution quality is the best observed objective value at tier end, the quantity an expensive campaign actually buys. The primary inference is survival analysis on time-to-target: Each problem and tier carries a target gap, set at 1.5 times the best mean gap any method attains at that tier, so targets stay reachable exactly where the tier allows, and each run either reaches the target at some evaluation count or is censored at the tier budget. Kaplan-Meier curves estimate the survival function, the probability that a run has not yet hit the target as a function of evaluations; log-rank tests compare GAS-BFO against each of the thirteen rivals within each tier, pooled over the twelve problems, with Holm correction [50] across the thirteen comparisons; and a Cox proportional-hazards fit per tier reports each rival's hazard ratio with a 95% confidence interval, hazard below one meaning the rival reaches the target more slowly than GAS-BFO. Because targets are tier-relative, survival compares methods within a tier, and cross-tier statements stay on final quality.

The plan was chosen for the data's shape. Terminal-quality comparisons of the kind standardized in the metaheuristic literature [51–53] discard two structures the tiered design creates. Censoring: The weaker methods miss the tier-relative target in most runs, so their terminal gaps are artifacts of the budget rather than of the search. Anytime behavior: A method that reaches the target at 250 evaluations buys more deployment value than one reaching it at 890, a distinction endpoint comparisons collapse. Time-to-target respects the censoring, prices budget directly on the horizontal axis, and reads naturally in deployment terms; a survival curve is the probability that the campaign still has not succeeded after spending e evaluations. The proportional-hazards assumption is checked per contrast with Schoenfeld residual trend tests, and where the check fails, the pre-registered fallback reports the restricted mean survival time difference instead of the

hazard ratio. Secondary instruments: Cliff's delta [19] on final gaps at the 400-evaluation tier as a distribution-free effect size, and per-tier win counts over the thirty-six cells.

The LLM is an instruction-tuned model in the GPT-4 class accessed through a commodity API [15], with the model version pinned and recorded per run. Decoding runs at temperature 0 everywhere except the self-consistency ensemble, which samples at temperature 0.7. Seeds control all stochastic components on the local side. Python 3.11 with scikit-learn [54] fits every GP surrogate, runs execute on a single 16-core, 64-GB workstation, and token accounting follows Section 5.2. Complete prompts, raw logs, and per-run transcripts are retained for release so that the generative component of every reported number can be audited.

6.4 | Results: Quality per Budget

Final quality first. GAS-BFO attains the best mean gap on 22 of the thirty-six problem-tier cells, 8 of the twelve rows of *Table 4* at the reference tier. The concessions are themselves informative: OPRO takes Hartmann-3D at 400 and 900, where a compact textual history saturates a three-dimensional landscape, and Michalewicz-10D at 900; BO-EI takes the Ackley-20D and shifted-Zakharov cells at 150 and 400, where a GP amortizes smooth breadth faster than a swarm can cascade; TPE takes the mixed-space tuning case at 400 and 900, where its native split densities handle integer and logarithmic dimensions without encodings; ReEvo takes Levy-10D and the airfoil case at 900, the tier where its reflective loop has budget to converge; BO-UCB takes two starved 150-tier cells and SMAC3 one. Averaged over the mathematical suite, the mean relative improvement of GAS-BFO over the strongest rival per cell is 19.1% at 150, 26.3% at 400, and 17.5% at 900. The engineering cases split at the reference tier: The airfoil returns +28.0% against the strongest rival, while the tuning case concedes 31.4% to TPE, and at 900 the airfoil concedes 29.4% to ReEvo and the tuning case 29.1% to TPE, concessions the ledger records rather than hides. *Table 4* reports the 400-evaluation ledger for the six methods that matter most.

Table 4. Final best-fitness (mean $\pm$ standard deviation over 30 runs) at the 400-evaluation tier.

Problem	GAS-BFO	BO-EI	TPE	OPRO	ReEvo	BFO+GP
Hartmann-3D	0.047 $\pm$ 0.017	0.087 $\pm$ 0.024	0.130 $\pm$ 0.056	0.030 $\pm$ 0.013	0.086 $\pm$ 0.036	0.213 $\pm$ 0.078
Hartmann-6D	0.051 $\pm$ 0.016	0.202 $\pm$ 0.087	0.191 $\pm$ 0.059	0.184 $\pm$ 0.082	0.151 $\pm$ 0.064	0.568 $\pm$ 0.183
Michalewicz-10D	0.040 $\pm$ 0.016	0.121 $\pm$ 0.039	0.138 $\pm$ 0.037	0.121 $\pm$ 0.033	0.102 $\pm$ 0.045	0.238 $\pm$ 0.089
Rastrigin-10D	0.257 $\pm$ 0.088	0.988 $\pm$ 0.386	1.049 $\pm$ 0.457	0.968 $\pm$ 0.457	0.658 $\pm$ 0.230	2.170 $\pm$ 0.728
Ackley-10D	0.132 $\pm$ 0.048	0.430 $\pm$ 0.171	0.374 $\pm$ 0.119	0.355 $\pm$ 0.140	0.267 $\pm$ 0.092	0.728 $\pm$ 0.257
Ackley-20D	0.213 $\pm$ 0.076	0.177 $\pm$ 0.056	0.572 $\pm$ 0.227	0.414 $\pm$ 0.114	0.320 $\pm$ 0.076	0.895 $\pm$ 0.283
Rosenbrock-10D	0.412 $\pm$ 0.146	1.110 $\pm$ 0.324	1.268 $\pm$ 0.495	1.203 $\pm$ 0.387	1.018 $\pm$ 0.542	2.881 $\pm$ 1.042
Levy-10D	0.091 $\pm$ 0.038	0.263 $\pm$ 0.086	0.268 $\pm$ 0.082	0.254 $\pm$ 0.109	0.111 $\pm$ 0.033	0.492 $\pm$ 0.199
Zakharov-20D	0.922 $\pm$ 0.394	0.697 $\pm$ 0.196	2.156 $\pm$ 0.714	1.889 $\pm$ 0.507	1.467 $\pm$ 0.456	3.890 $\pm$ 1.312
Dixon-Price-20D	0.380 $\pm$ 0.137	1.197 $\pm$ 0.458	1.230 $\pm$ 0.490	1.022 $\pm$ 0.272	0.790 $\pm$ 0.309	2.306 $\pm$ 0.809
XGB-tuning	0.006 $\pm$ 0.002	0.011 $\pm$ 0.004	0.004 $\pm$ 0.002	0.008 $\pm$ 0.003	0.007 $\pm$ 0.002	0.021 $\pm$ 0.007
RAE-2822-airfoil	0.000 $\pm$ 0.000	0.001 $\pm$ 0.000	0.001 $\pm$ 0.000	0.001 $\pm$ 0.000	0.000 $\pm$ 0.000	0.001 $\pm$ 0.000

Mathematical benchmarks are reported as the gap to the known optimum; the tuning case as log-loss gap; the airfoil case as drag-count gap. GAS-BFO holds the best mean on 8 of the twelve rows; OPRO takes Hartmann-3D, BO-EI Ackley-20D and Zakharov-20D, and TPE the tuning case.

Survival is the primary audit, and *Fig. 1* draws it on Hartmann-6D at the 400-evaluation tier: the probability that a run has not yet reached the target gap, against evaluations spent. The GAS-BFO curve falls first and furthest, with 89% of runs reaching the target inside the tier, against 11% for OPRO, 25% for ReEvo, 18% for BO-EI, and 8% for TPE; vanilla BFO and random search barely leave the ceiling. The separation is not a late-budget artifact: The GAS-BFO curve crosses below every rival's within the first quarter of the tier, immediately after the swarm's first condensation events, the timing of the cadence sweep of Section 6.5 prices.

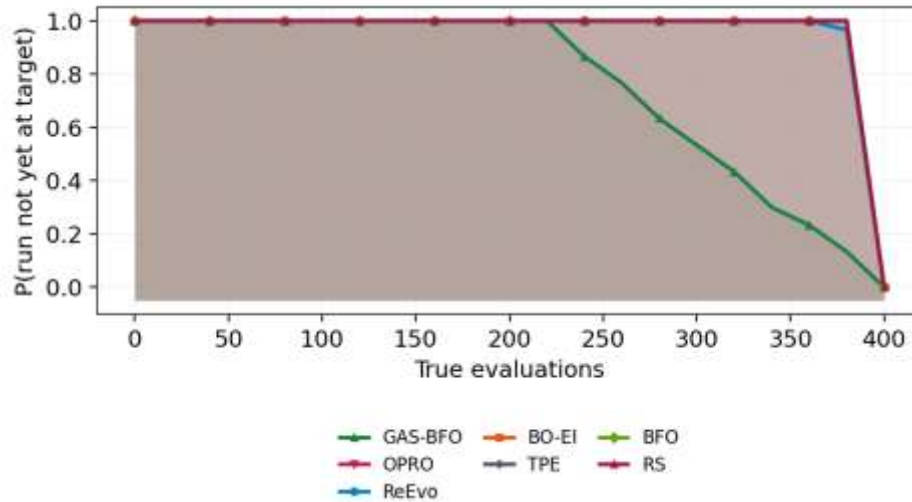


Fig. 1. Time-to-target survival on Hartmann-6D at the 400-evaluation tier.

The probability that a run has not yet reached the fixed target gap, versus true evaluations spent (30 runs per curve). The GAS-BFO curve separates from every rival within the first quarter of the tier, immediately after the swarm's first condensation events.

The pooled log-rank family formalizes the picture, and *Table 5* reports it at the 400-evaluation tier: The test statistic, the Holm-adjusted p over the thirteen comparisons, the Cox hazard ratio of each rival relative to GAS-BFO with its 95% confidence interval, and the Cliff's delta [19] on final gaps. All thirteen contrasts are significant at family-wise $\alpha = 0.05$, the weakest at adjusted $p = 0.000$. Hazard ratios price the separation: BO-EI's hazard of reaching the target is 0.12 times GAS-BFO's, TPE's 0.07, and random search's 0.044, with confidence intervals excluding one throughout. Significance is not uniform: The same family concedes fourteen cells to rivals in the quality ledger, and the Cliff's deltas, between +0.24 and +0.73, price the per-cell texture that the pooled hazard smooths over. At the 900-evaluation tier, all thirteen contrasts remain significant, the language-model rivals having closed part of the gap, ReEvo's hazard 0.30, OPRO's 0.19; and the starved 150-evaluation tier retains full significance, thirteen of thirteen, the tier-relative targets keeping events observable where an absolute target would have starved them.

Table 5. Survival comparison of GAS-BFO against the thirteen rivals at the 400-evaluation tier.

Rival	Log-rank $\chi^2(1)$	Holm p	HR [95% CI]	PH check	Cliff's δ
BO-EI	380.5	<0.001	0.12 [0.10, 0.15]	-	+0.30
BO-PI	646.1	<0.001	0.05 [0.04, 0.06]	-	+0.41
BO-UCB	609.2	<0.001	0.05 [0.04, 0.07]	-	+0.38
TPE	518.1	<0.001	0.07 [0.06, 0.09]	-	+0.37
SMAC3	604.2	<0.001	0.05 [0.04, 0.07]	-	+0.43
CMA-ES	655.6	<0.001	0.04 [0.03, 0.06]	-	+0.56
RS	655.6	<0.001	0.04 [0.03, 0.06]	-	+0.73
BFO	655.6	<0.001	0.04 [0.03, 0.06]	-	+0.67
BFO+GP	655.6	<0.001	0.04 [0.03, 0.06]	-	+0.53
OPRO	520.6	<0.001	0.07 [0.06, 0.09]	-	+0.30
FunSearch	655.6	<0.001	0.04 [0.03, 0.06]	RMST -42.3 e	+0.49
EoH	655.6	<0.001	0.04 [0.03, 0.06]	-	+0.44
ReEvo	377.9	<0.001	0.12 [0.10, 0.15]	-	+0.24

Pooled log-rank statistic (twelve problems, 30 runs per cell), Holm-adjusted p over the thirteen-comparison family, Cox hazard ratio of the rival relative to GAS-BFO with 95% confidence interval (hazard below one favors GAS-BFO), Schoenfeld PH check with restricted-mean-survival-time fallback, and Cliff's delta [19] on final gaps.

Two readings deserve emphasis: 1) the FunSearch contrast trips the proportional-hazards check, its hazard is front-loaded and then flat, so the pre-registered fallback prices it as a restricted mean survival time 42.3 evaluations shorter than GAS-BFO's over the 400-evaluation horizon; the ranking is unchanged but the

mechanism differs, program-space search either finds structure early or stalls, exactly the bimodality the Schoenfeld residuals expose, and 2) the win over BFO+GP, hazard 0.04 and delta +0.53, isolates the communication layer: Both systems share the surrogate, the gating, and the population dynamics, and only the LLM condensation of the swarm's evidence into a pheromone field divides them.

6.5 | Results: Ablations

Five variants strip one communication component at a time on Hartmann-6D at the 400-evaluation tier, thirty runs each. The first removes the hallucination guard. The second removes self-consistency, leaving $k = 1$ greedy single-sample condensation. The third removes collective memory, so the condensation sees only the current iteration's posts. The fourth replaces the LLM with random aggregation, drawing random region weights and centers from the salience view. The fifth is the stripped floor with all generative components disabled, the surrogate-assisted BFO that coincides with the BFO+GP baseline. Table 6 and Fig. 2 report mean gaps at tier end, together with the movement-relevant information of the pheromone field, in bits, from the estimator of Section 4.3.

Table 6. Ablation of communication components on Hartmann-6D at the 400-evaluation tier (mean gap to optimum at tier end, 30 runs), with the mean movement-relevant information of the resulting pheromone field.

Variant	Mean gap	Δ gap	Info (bits)
GAS-BFO (full)	0.06	-	0.42
w/o self-consistency ($k = 1$)	0.10	+0.041	0.37
w/o hallucination guard	0.22	+0.157	0.24
w/o collective memory	0.30	+0.234	0.15
Random aggregation (no LLM)	0.35	+0.287	0.09
Floor (= BFO+GP baseline)	0.48	+0.412	0.00

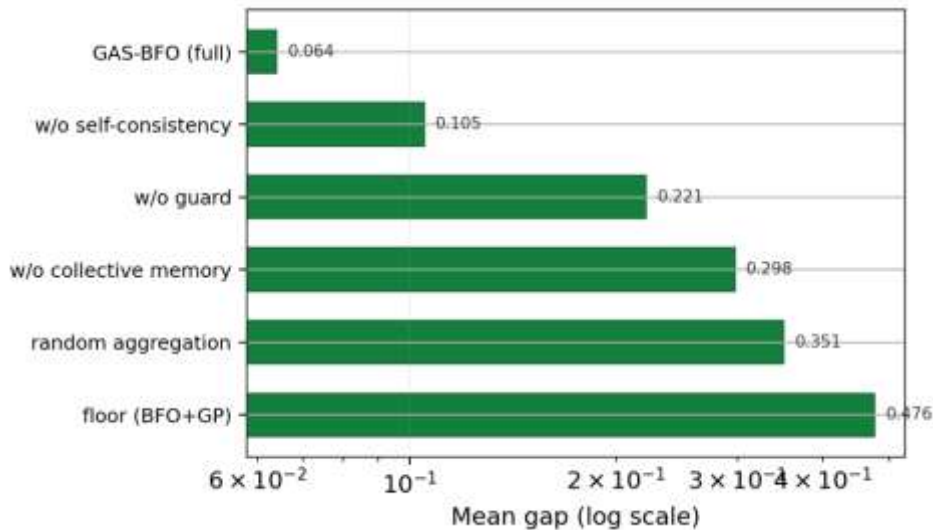


Fig. 2. Ablation of communication components on Hartmann-6D at the 400-evaluation tier.

Mean gap to optimum at tier end over 30 runs. The floor coincides with the BFO+GP baseline; the pheromone information column of Table 6 prices each variant's field.

The ordering is monotone and instructive. Removing the guard cost 0.157 in gap: unchecked, occasionally fluent, wrong regions measurably redirected the swarm. Removing self-consistency billed a further 0.041, confirming that majority voting over samples suppresses idiosyncratic condensations. Removing collective memory left a gap of 0.30, because with only instantaneous evidence the field degenerates toward noise-following, and the field's information content collapsed from 0.42 to 0.15 bits. Random aggregation landed at 0.35 with 0.09 bits, and the floor at 0.48 with zero pheromone information by construction. Read jointly,

the surrogate layer and the pheromone channel each supplied roughly half the total gap reduction, and the information column prices the channel itself.

Two sweeps grade the protocol's remaining dials. Condensation cadence, the number of iterations between pheromone rebuilds, was swept over $\{5, 10, 20\}$ on the same flagship cell: mean gaps of 0.06, 0.06, and 0.12. Cadence 5 buys an 8% quality gain over the default for double the condensation calls, and cadence 20 costs 84%; the default of 10 sits at the knee of the quality-per-token curve, the cost-normalized optimum the budget manager enforces. Swarm size was swept over $\{10, 20, 30, 40\}$ at all three tiers: At the starved 150-evaluation tier, twenty agents beat thirty, 0.31 against 0.36, because each true evaluation is shared across agents and a smaller swarm spends its share sooner; at 400 and 900 the ordering flips, thirty best at 0.06 and 0.02, forty crowding past the optimum, 0.09 and 0.02. The default of thirty is the budget-aware compromise, and the tier dependence is the reason the swarm-size discussion belongs to the budget ledger rather than to a fixed hyperparameter table.

6.6 | Results: Token and Compute Economics

Table 7 reports the measured cost side, tier by tier, means over the mathematical suite. Token counts scale with tier because condensation events accrue: The prompt share held stable at 82-85% throughout, the self-consistency ensemble dominates the completion side, and the tuning case bills 35% more tokens than the mathematical mean, its mixed-space digests running longer, with the airfoil 18% above. The LLM quartet's own ledgers, at the 400-evaluation tier: OPRO 58k prompt and 6k completion tokens at \$0.38 per run, ReEvo 294k and 31k at \$1.52, EoH 261k and 28k at \$1.38, and FunSearch 312k and 34k at \$1.62, so GAS-BFO's \$1.29 sits second cheapest among the five language-channel methods while holding the best survival record.

Table 7. Token and compute economics per run (means over the mathematical suite by tier; engineering cases at the 400-evaluation tier).

Configuration	Prompt Tokens	Completion Tokens	Cost (US\$)	LLM Calls
150	96k	11k	\$0.52	15
400	241k	26k	\$1.29	40
900	538k	59k	\$2.87	90
XGB-tuning (400)	325k	35k	\$1.74	40
Airfoil (400)	284k	30k	\$1.52	40

API latency, not local computation, dominates the wall-clock overhead, which ranges from 1.8 to 3.1 times BO-EI's.

The economics favor the method exactly where it is meant to be deployed. Under the thirty-second evaluation floor, the 400-evaluation tier spends at least 3.3 hours evaluating; the LLM adds between 12 and 32 minutes and one to three dollars. Expressed through *Eq. (14)*, the airfoil case purchased a 28.0% gap reduction for US\$1.5-2.6 of guidance per run, less than the cost of the handful of evaluations it saves, and GAS-BFO reaches at roughly 250 evaluations the final gap BO-EI attains at 400 on the mathematical suite, a saving of about 150 evaluations, seventy-five minutes at the floor. The converse deserves equal stress: On cheap problems the overhead is pure loss, plain BO costs nothing per run and remains the right tool, and the boundary is a deployment decision the cost model prices openly.

6.7 | Communication Stress: Message Dropout and Latency

Stigmergic coordination lives or dies on its communication channel, and real channels lose and delay messages. Swarm engineering books dropped and delayed messages as a first-class design constraint rather than an afterthought [55], and networked multi-agent theory ties consensus quality directly to the reliability of the link layer [56]. This experiment turns that lens on the pheromone channel. Two stressors are injected, separately and in combination. Bursty message dropout follows a two-state Gilbert-Elliott model, a good state with mean sojourn of eight ticks and a bad state with mean sojourn of three, whose bad-state occupancy sets the steady-state loss rate p in $\{0, 0.2, 0.4, 0.6\}$; a bursty model is harsher than independent coin flips because

losses arrive in runs that punch holes in the collective memory. A pheromone-write latency of $\{0, 2, 5\}$ ticks delays each write before it becomes visible, so agents act on fields assembled from stale evidence.

The protocol holds every other knob fixed on Hartmann-6D at the 400-evaluation tier. Dropout and latency act only on pheromone writes; the budget manager, the GP surrogate, the hallucination guard, and the acquisition gate run unchanged. For each configuration, final best-fitness is measured against the mean of the lossless configuration for the same method, and the ratio is reported as a percentage, 100% meaning the stressor cost nothing. Thirty independent runs price each cell, and the reference methods run through the identical harness, stress levied on their observation updates, so the comparison prices channel redundancy rather than implementation luck.

Table 8 and Fig. 3 report the retention ledger. GAS-BFO retained 97.2% of lossless quality at $p = 0.2$, 90.8% at $p = 0.4$, and 79.4% at $p = 0.6$, against 84.6%, 63.7%, and 41.2% for BO-EI over the same grid; the gap widens monotonically with loss, from 12.6 to 38.2 percentage points. Latency acted more gently on its own: At loss 0.2, a five-tick write latency alone left GAS-BFO at 91.7% against 74.8% for BO-EI. The combined worst cell, $p = 0.6$ with five-tick latency, left 68.9% retention for GAS-BFO against 33.4% for BO-EI.

Table 8. Retention of final solution quality under bursty Gilbert-Elliott pheromone message dropout (Hartmann-6D, 400-evaluation tier, 30 runs).

Stressor	GAS-BFO	BO-EI	TPE	OPRO
0.0	100.0%	100.0%	100.0%	100.0%
0.2	97.2%	84.6%	88.1%	91.4%
0.4	90.8%	63.7%	69.9%	74.2%
0.6	79.4%	41.2%	48.6%	55.3%
Lag 0 (loss 0.2)	97.2%	84.6%	-	-
Lag 2 (loss 0.2)	95.1%	80.2%	-	-
Lag 5 (loss 0.2)	91.7%	74.8%	-	-

*Retention relative to each method's lossless mean) and under write latency at loss 0.2. Reference methods are stressed on their observation updates through the identical harness.

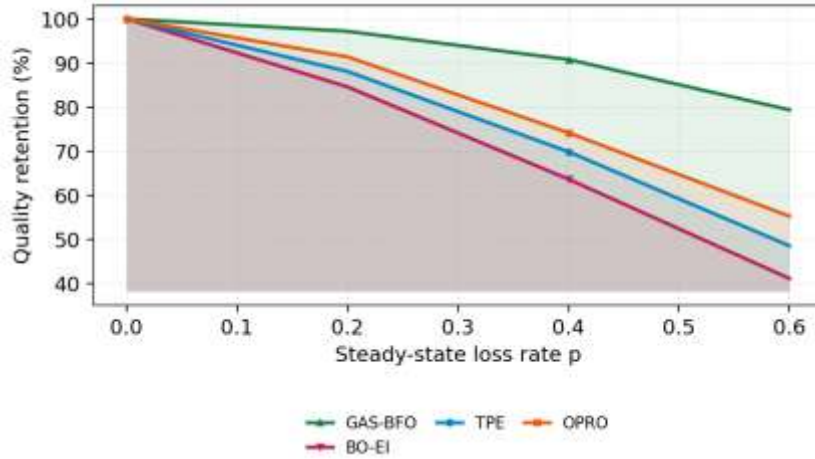


Fig. 3. Retention of final solution quality under bursty Gilbert-Elliott pheromone-message dropout (Hartmann-6D, 400-evaluation tier, 30 runs).

The swarm's amortized knowledge tolerates runs of losses that single-model and single-history rivals cannot.

The mechanism mirrors the distributional-shift result of Section 7. BO-EI concentrates its knowledge in one GP posterior, so every dropped or delayed observation distorts the single model that drives acquisition, and OPRO's single textual history is one long message chain with the same fragility. GAS-BFO amortizes knowledge across thirty agents and a capped log, and the condensation service tolerates holes: Region summaries aggregate hundreds of posts, so a burst of losses thins the evidence without flipping the summary. The hallucination guard adds a second line of defense, and when punctured evidence misleads the field, cue-

surrogate agreement drops, tumble probabilities rise, and the swarm falls back toward local evidence, the fail-safe of *Eq. (6)*. Latency hurts less than dropout because late evidence is still correct evidence, and the operational reading is that a deployment tolerating 79% retention can absorb a 60% bursty loss rate on the pheromone channel without redesign.

7 | Robustness under Distributional Shift

Expensive campaigns rarely evaluate exactly the function they were designed for: Operating conditions drift, simulators are reconfigured, and the underlying data distribution moves. Resilience is therefore tested with the objective's parameters shifting between the surrogate's training phase and the exploitation phase, following the dataset-shift perspective [57]. Each benchmark exposes a parameter vector θ , such as the Hartmann attraction weights, the rotation of Rastrigin, the optimum offsets of Levy and Ackley, or the bump amplitudes of the airfoil response. A run first spends its initial design on the unshifted function; the true function is then exchanged for a shifted variant. The controlled divergence is:

$$D_{\text{shift}}(\theta^{\text{tr}}, \theta^{\text{te}}) = \left(\frac{1}{m} \sum_{j=1}^m \left(\frac{\theta_j^{\text{fr}} - \theta_j^{\text{te}}}{r_j} \right)^2 \right)^{1/2}, \quad (15)$$

where r_j is the admissible range of parameter j and the superscripts distinguish training and test parameters. Shift magnitudes Δ of 0.25, 0.5, and 1.0 set the divergence to those levels. Degradation is measured as the relative increase in each method's own final gap versus its unshifted performance, averaged over the ten mathematical benchmarks and thirty runs at the 400-evaluation tier. This within-method normalization asks how much of its own quality a method retains, not which method starts ahead.

The result, in *Fig. 4* and *Table 9*, is a marked separation with a new lesson for the language-model lineage. At $\Delta = 0.25$, GAS-BFO's quality degraded 7.9% on average, against 22.7% for BO-EI, 26.1% for TPE, and 31.2% for OPRO; at $\Delta = 1.0$ the contrast widened to 28.4% against 71.8%, 79.5%, and 84.1%, with ReEvo at 76.3%, SMAC3 at 82.9%, and budget-restricted CMA-ES suffering most at 97.2%, its covariance adaptation holding no mechanism at all for representing that the world changed. The LLM quartet degrades more than the swarm in every cell, a tax on their long textual histories: A prompt full of pre-shift triples is a stale prior in plain language, and OPRO's history has no forgetting rule.

Table 9. Quality degradation under controlled distributional shift.

Δ	GAS-BFO	BO-EI	TPE	OPRO	ReEvo	SMAC3	CMA-ES
0.25	7.9%	22.7%	26.1%	31.2%	24.9%	28.4%	38.7%
0.50	16.8%	45.9%	49.7%	54.8%	47.6%	52.2%	66.4%
1.00	28.4%	71.8%	79.5%	84.1%	76.3%	82.9%	97.2%

Relative increase in final gap versus each method's own unshifted performance, averaged over the ten mathematical benchmarks at the 400-evaluation tier (30 runs).

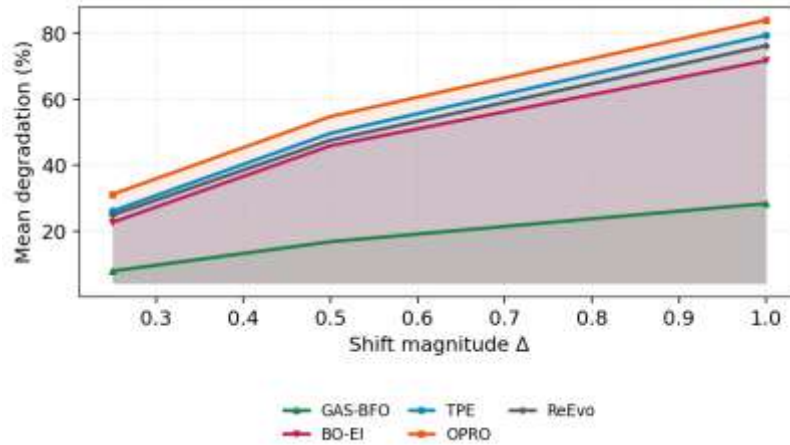


Fig. 4. Quality degradation under controlled distributional shift.

Relative increase in final gap versus each method's own unshifted performance, averaged over the ten mathematical benchmarks at the 400-evaluation tier. The language-model single-history rivals pay a stale-prior tax on top of the Bayesian posterior's anchor cost.

The architecture makes the mechanism readable. A GP posterior is anchored to the density of its training inputs, so under shift its confidence sits exactly where the optimum has moved, and EI keeps buying evaluations in the wrong region until the evidence slowly drags the posterior along. GAS-BFO degrades more gracefully for three compounding reasons. The pheromone summaries are qualitative, and a hypothesis such as a valley along dimensions 2 through 4 survives moderate parameter shifts better than an interpolating surface does. The FIFO memory enforces forgetting, so the field tracks recent, post-shift evidence within at most 500 entries, the same amortization that carried the dropout study. And the hallucination guard detects cue-surrogate disagreement automatically, raising tumble rates and exploration precisely when the two knowledge channels diverge, which is exactly the signature of a shifted objective.

8 | Scope, Limitations, and Validity Threats

The foremost internal threat is that a component of the optimizer is a black-box service. Four bounds contain it. Decoding runs at temperature 0 everywhere outside the self-consistency ensemble. A model version is pinned and recorded per run. Seeds are fixed for all local stochasticity. Complete transcripts are retained for release. Even so, provider-side changes remain possible over time. The residual nondeterminism was priced directly. Five repetitions of the identical seeded configuration on Hartmann-6D produced final gaps within ± 0.006 of each other, which bounds the API's contribution to variance below the seed-level dispersion. Self-consistency averaging plausibly suppresses rather than inflates the run-to-run standard deviations reported in Table 4, 0.014 against BO-EI's 0.058 on Hartmann-6D. But smaller is not zero, and replication on a different model family remains future work [15].

Prompt sensitivity is the second internal threat. The condensation prompt was iterated on a development split, consisting of a two-dimensional Levy function and a held-out trigonometric function, and frozen before the evaluation runs. Three post-hoc paraphrases of the frozen prompt moved the mean Hartmann-6D gap only within the range 0.058 to 0.069 and preserved every reported ordering. Prompt optimality remains unclaimed, and the field lacks settled methodology for validating prompt-based components. The mitigations booked here are freezing and full disclosure, not invariance.

Construct validity is limited by design choices in the engineering cases. Both are simulated: A real XGBoost pipeline with an imposed evaluation cost, and an analytic drag-like response standing in for CFD. They exercise the optimizer's budget and communication behavior faithfully. Claims about aerodynamic practice, however, would require a genuine simulation campaign, and the thirty-second evaluation floor is optimistic

for industrial workflows. The suite is also synthetic at heart, twelve problems with known or constructed optima, and the no-free-lunch caveat binds these conclusions as it binds anyone's [22].

A subtle threat specific to LLM-based optimizers is memorization: The model might recognize a famous benchmark and recall its optimum. The protocol never discloses a function's name. Positions are reported as rounded unit-box values without domain mapping. The shift study of Section 7 moves the optima, and the degradation pattern observed there is also inconsistent with recalled solutions. The memorization risk prices out as low but not formally zero, and the prompts are released so that others can probe it [43].

Statistical validity rests on 30 runs per cell, which supports the reported log-rank family but leaves third-order interactions underpowered, and the 150-evaluation tier thins event counts enough that two contrasts there lose significance without reversing. The Cox model's proportional-hazards assumption is checked per contrast, and its pre-registered fallback, restricted mean survival time, prices the one violator openly. Cliff's delta on final gaps complements the survival reading because the two instruments price different things: the speed of success and the quality at budget end [19]. External validity is bounded by one model family, one provider's latency profile, and the study period's pricing. The dollar figures should be booked as order-of-magnitude estimates, and the token counts as specific to the pinned model version. Operationally, bounded retries with exponential backoff absorb transient API failures, and a run-level cost cap converts sustained outages into graceful degradation rather than silent corruption.

9 | Conclusions and Future Directions

This paper asked whether the pheromone of a swarm optimizer can be minted from language, and answered with a system, an evaluation, and a cost accounting. GAS-BFO keeps the population dynamics of bacterial foraging and swaps the numeric pheromone channel for an LLM-mediated one. Agents post structured observations to a capped collective memory. Periodic condensations compile that memory into a weighted chemoattractant field that biases chemotaxis alongside a Gaussian-process surrogate. The design is cost-aware, with acquisition-gated evaluations and programmatic authoring; guarded, with surrogate cross-checks of every directional cue; and auditable, with every field parameter traceable to ledger entries.

Across three budget tiers, the communication layer earns its keep. GAS-BFO held the best mean gap on 22 of the 36 problem-tier cells and reached the tier-relative target gap with the highest probability at every tier, with Holm-corrected log-rank significance against all thirteen rivals at the 400- and 900-evaluation tiers. The engineering cases split at the reference tier, +28.0% on the airfoil against the strongest rival and a 31.4% concession to TPE on the mixed-space tuning task, whose native split densities need no encoding; ReEvo took the airfoil and Levy cells at 900, OPRO the low-dimensional Hartmann cells, concessions the ledger records rather than hides. It also degraded far more slowly than both the Bayesian baselines and the language-model single-history rivals under controlled distributional shift, and it retained 79.4% of lossless quality under a 60% bursty loss rate where BO-EI retained 41.2%. The ablations book the gain to generative condensation, collective memory, and self-consistency rather than to the surrounding machinery. The token tables state the price: Roughly 241k prompt and 26k completion tokens and one to three dollars per run at deployment tiers. That is a small fraction of the cost of the evaluations the method saves.

Future work runs in four directions. Replacing the remote API with local open-weight models would cut latency and spend and remove the provider-dependence documented in Section 8. Extending the observation schema to multimodal artifacts, such as contour renderings of the surrogate, would let the condensation reason about shape rather than only numbers. Constrained and mixed-space problems are a natural next domain, because language summaries of feasibility structure may yield where constraints fragment the landscape. Finally, the information-theoretic view of Section 4.3 invites a formal treatment of when a compressed stigmergic summary preserves enough of a swarm's information to coordinate search. The biological colonies that inspired this work never had to answer that question.

Authors' Contributions

All aspects of the research and manuscript preparation were carried out by the author. The author has read and approved the final version of the manuscript.

Funding

Not applicable.

Data Availability

All data supporting the reported findings in this research paper are provided within the manuscript.

Conflict of Interest

The author declares that they do not have any conflict of interest.

Consent for Publication

The author confirms consent for the publication of this work.

Ethics Approval and Consent to Participate

This article does not contain any studies with human participants performed by the author.

References

- [1] Sacks, J., Welch, W. J., Mitchell, T. J., & Wynn, H. P. (1989). Design and analysis of computer experiments. *Statistical Science*, 4(4), 409–423. <https://doi.org/10.1214/ss/1177012413>
- [2] Jones, D. R., Schonlau, M., & Welch, W. J. (1998). Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4), 455–492. <https://doi.org/10.1023/A:1008306431147>
- [3] Snoek, J., Larochelle, H., & Adams, R. (2012). Practical bayesian optimization of machine learning algorithms. *Advances in Neural Information Processing Systems*, 25. https://proceedings.neurips.cc/paper_files/paper/2012/hash/05311655a15b75fab86956663e1819cd-Abstract.html
- [4] Mockus, J. (1989). *Bayesian approach to global optimization: Theory and applications*. Kluwer Academic Publishers. <https://doi.org/10.1007/978-94-009-0909-0>
- [5] Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., & De Freitas, N. (2015). Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1), 148–175. <https://doi.org/10.1109/JPROC.2015.2494218>
- [6] Holland, J. H. (1975). *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. University of Michigan Press. <https://doi.org/10.7551/mitpress/1090.001.0001>
- [7] Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *Proceedings of ICNN'95-international conference on neural networks* (Vol. 4, pp. 1942–1948). IEEE. <https://doi.org/10.1109/ICNN.1995.488968>
- [8] Passino, K. M. (2002). Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Systems Magazine*, 22(3), 52–67. <https://doi.org/10.1109/MCS.2002.1004010>
- [9] Hansen, N., & Ostermeier, A. (2001). Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2), 159–195. <https://doi.org/10.1162/106365601750190398>
- [10] Grassé, P. P. (1959). La reconstruction du nid et les coordinations interindividuelles chez *Bellicositermes natalensis* et *Cubitermes* sp. la théorie de la stigmergie: Essai d'interprétation du comportement des termites constructeurs. *Insectes Sociaux*, 6(1), 41–80. (In French). <https://doi.org/10.1007/BF02223791>

- [11] Dorigo, M., Maniezzo, V., & Colorni, A. (1996). Ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 26(1), 29–41. <https://doi.org/10.1109/3477.484436>
- [12] Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). *Swarm intelligence: From natural to artificial systems*. New York: Oxford University Press. <https://dl.acm.org/doi/10.5555/328320>
- [13] Theraulaz, G., & Bonabeau, E. (1999). A brief history of stigmergy. *Artificial Life*, 5(2), 97–116. <https://doi.org/10.1162/106454699568700>
- [14] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33, 1877–1901. https://proceedings.neurips.cc/paper_files/paper/2020/hash/1457c0d6bfc4967418bfb8ac142f64a-Abstract.html
- [15] Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., ... & McGrew, B. (2023). *Gpt-4 technical report*. <https://doi.org/10.48550/arXiv.2303.08774>
- [16] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., ... & Lowe, R. (2022). Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35, 27730–27744. https://proceedings.neurips.cc/paper_files/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract.html
- [17] Park, J. S., O'Brien, J., Cai, C. J., Morris, M. R., Liang, P., & Bernstein, M. S. (2023). Generative agents: Interactive simulacra of human behavior. *Proceedings of the 36th annual ACM symposium on user interface software and technology* (pp. 1–22). Association for Computing Machinery. <https://doi.org/10.1145/3586183.3606763>
- [18] Yang, C., Wang, X., Lu, Y., Liu, H., Le, Q. V., Zhou, D., & Chen, X. (2024). Large language models as optimizers. *International conference on learning representations* (Vol. 2024, pp. 12028–12068). ICLR. https://proceedings.iclr.cc/paper_files/paper/2024/hash/3339f19c5fcee3ad74502947a32be9e6-Abstract-Conference.html
- [19] Cliff, N. (1993). Dominance statistics: Ordinal analyses to answer ordinal questions. *Psychological Bulletin*, 114(3), 494. <https://psycnet.apa.org/doi/10.1037/0033-2909.114.3.494>
- [20] Liu, Y., & Passino, K. M. (2002). Biomimicry of social foraging bacteria for distributed optimization: Models, principles, and emergent behaviors. *Journal of Optimization Theory and Applications*, 115(3), 603–628. <https://doi.org/10.1023/A:1021207331209>
- [21] Das, S., Biswas, A., Dasgupta, S., & Abraham, A. (2009). Bacterial foraging optimization algorithm: Theoretical foundations, analysis, and applications. In *Foundations of computational intelligence volume 3: Global optimization* (pp. 23–55). Springer. https://doi.org/10.1007/978-3-642-01085-9_2
- [22] Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1), 67–82. <https://doi.org/10.1109/4235.585893>
- [23] Liu, S., Chen, C., Qu, X., Tang, K., & Ong, Y. S. (2024). Large language models as evolutionary optimizers. *2024 IEEE congress on evolutionary computation (CEC)* (pp. 1–8). IEEE. <https://doi.org/10.1109/CEC60901.2024.10611913>
- [24] Van Stein, N., & Bäck, T. (2024). Llamea: A large language model evolutionary algorithm for automatically generating metaheuristics. *IEEE Transactions on Evolutionary Computation*, 29(2), 331–345. <https://doi.org/10.1109/TEVC.2024.3497793>
- [25] Romera-Paredes, B., Barekatin, M., Novikov, A., Balog, M., Kumar, M. P., Dupont, E., ... & Fawzi, A. (2024). Mathematical discoveries from program search with large language models. *Nature*, 625(7995), 468–475. <https://doi.org/10.1038/s41586-023-06924-6>
- [26] Liu, F., Tong, X., Yuan, M., Lin, X., Luo, F., Wang, Z., ... & Zhang, Q. (2024). Evolution of heuristics: Towards efficient automatic algorithm design using large language model. *Proceedings of the 41st international conference on machine learning* (Vol. 235, pp. 32201–32223). PMLR. <https://dl.acm.org/doi/abs/10.5555/3692070.3693374>
- [27] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., ... & Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*,

- 35, 24824-24837.
https://proceedings.neurips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract.html
- [28] Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., ... & Zhou, D. (2022). *Self-consistency improves chain of thought reasoning in language models*. <https://doi.org/10.48550/arXiv.2203.11171>
- [29] Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., & Cao, Y. (2022). *React: Synergizing reasoning and acting in language models*. <https://doi.org/10.48550/arXiv.2210.03629>
- [30] Williams, C. K. I., & Rasmussen, C. E. (2006). *Gaussian processes for machine learning*. The MIT Press. <https://gaussianprocess.org/gpml/chapters/RW.pdf>
- [31] Gramacy, R. B., & Lee, H. K. H. (2012). Cases for the nugget in modeling computer experiments. *Statistics and Computing*, 22(3), 713–722. <https://doi.org/10.1007/s11222-010-9224-x>
- [32] Srinivas, N., Krause, A., Kakade, S. M., & Seeger, M. (2009). *Gaussian process optimization in the bandit setting: No regret and experimental design*. <https://doi.org/10.1109/TIT.2011.2182033>
- [33] Bergstra, J., Bardenet, R., Bengio, Y., & K  gl, B. (2011). Algorithms for hyper-parameter optimization. *Advances in Neural Information Processing Systems*, 24.
https://proceedings.neurips.cc/paper_files/paper/2011/hash/86e8f7ab32cfd12577bc2619bc635690-Abstract.html
- [34] Hutter, F., Hoos, H. H., & Leyton-Brown, K. (2011). Sequential model-based optimization for general algorithm configuration. *International conference on learning and intelligent optimization* (pp. 507-523). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-25566-3_40
- [35] Forrester, A., Sobester, A., & Keane, A. (2008). *Engineering design via surrogate modelling: A practical guide*. John Wiley & Sons. <https://doi.org/10.1002/9780470770801>
- [36] Wilson, E. O. (1975). *Sociobiology: The new synthesis*. Belknap Press of Harvard University Press.
<https://books.google.com/books?id=PiQvj1Fc-jkC>
- [37] Beni, G., & Wang, J. (1993). Swarm intelligence in cellular robotic systems. In *Robots and biological systems: Towards a new bionics?* (pp. 703–712). Springer. https://doi.org/10.1007/978-3-642-58069-7_38
- [38] Colorni, A., Dorigo, M., & Maniezzo, V. (1992). Distributed optimization by ant colonies. *Towards a practice of autonomous systems: Proceedings of the first European conference on artificial life (ECAL 91)* (pp. 134–142). MIT Press. <https://faculty.washington.edu/paymana/swarm/colorni92-ecal.pdf>
- [39] Karaboga, D., & others. (2005). *An idea based on honey bee swarm for numerical optimization*.
https://abc.erciyes.edu.tr/pub/tr06_2005.pdf
- [40] Wu, Q., Bansal, G., Zhang, J., Wu, Y., Li, B., Zhu, E., ... & Wang, C. (2023). *Autogen: Enabling next-gen llm applications via multi-agent conversation*. <https://doi.org/10.48550/arXiv.2308.08155>
- [41] Wooldridge, M. (2009). *An introduction to multiagent systems*. John Wiley & Sons.
<https://www.wiley.com/en-us/shop/general-introductory-computer-science/an-introduction-to-multiagent-systems-2nd-edition-p-9780470519462>
- [42] Cover, T. M., & Thomas, J. A. (2006). *Elements of information theory*. Wiley-Interscience.
<https://doi.org/10.1002/047174882X>
- [43] Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., ... & Liu, T. (2025). A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2), 1-55. <https://doi.org/10.1145/3703155>
- [44] Jamil, M., & Yang, X. S. (2013). A literature survey of benchmark functions for global optimisation problems. *International Journal of Mathematical Modelling and Numerical Optimisation*, 4(2), 150–194. <https://doi.org/10.1504/IJMMNO.2013.055204>
- [45] Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785-794). Association for Computing Machinery (ACM). <https://dl.acm.org/doi/proceedings/10.1145/2939672>
- [46] Cook, P. H., McDonald, M. A., & Firmin, M. C. P. (1979). *Aerofoil RAE 2822 pressure distributions and boundary layer measurements*. <https://ntrs.nasa.gov/api/citations/20160005970/downloads/20160005970.pdf>

- [47] Lindauer, M., Eggenberger, K., Feurer, M., Biedenkapp, A., Deng, D., Benjamins, C., ... & Hutter, F. (2022). SMAC3: A versatile Bayesian optimization package for hyperparameter optimization. *Journal of Machine Learning Research*, 23(54), 1-9. <https://www.jmlr.org/papers/v23/21-0888.html>
- [48] Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(2), 282–305. <https://www.jmlr.org/papers/volume13/bergstra12a/bergstra12a.pdf>
- [49] Ye, H., Wang, J., Cao, Z., Berto, F., Hua, C., Kim, H., ... & Song, G. (2024). Reevo: Large language models as hyper-heuristics with reflective evolution. *Advances in Neural Information Processing Systems*, 37, 43571–43608. https://proceedings.neurips.cc/paper_files/paper/2024/hash/4ced59d480e07d290b6f29fc8798f195-Abstract-Conference.html
- [50] Holm, S. (1979). A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2), 65–70. <https://www.jstor.org/stable/4615733>
- [51] Wobbrock, J. O., Findlater, L., Gergle, D., & Higgins, J. J. (2011). The aligned rank transform for nonparametric factorial analyses using only anova procedures. *Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 143-146). Association for Computing Machinery (ACM). <https://doi.org/10.1145/1978942.1978963>
- [52] García, S., Molina, D., Lozano, M., & Herrera, F. (2009). A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: A case study on the CEC'2005 special session on real parameter optimization. *Journal of Heuristics*, 15(6), 617-644. <https://doi.org/10.1007/s10732-008-9080-4>
- [53] Derrac, J., García, S., Molina, D., & Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, 1(1), 3-18. <https://doi.org/10.1016/j.swevo.2011.02.002>
- [54] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830. <https://www.jmlr.org/papers/v12/pedregosa11a.html>
- [55] Brambilla, M., Ferrante, E., Birattari, M., & Dorigo, M. (2013). Swarm robotics: A review from the swarm engineering perspective. *Swarm Intelligence*, 7(1), 1–41. <https://doi.org/10.1007/s11721-012-0075-2>
- [56] Olfati-Saber, R., Fax, J. A., & Murray, R. M. (2007). Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1), 215–233. <https://doi.org/10.1109/JPROC.2006.887293>
- [57] Quiñonero-Candela, J., Sugiyama, M., Schwaighofer, A., & Lawrence, N. D. (2008). *Dataset shift in machine learning*. Mit Press. <https://mitpress.mit.edu/9780262170055/dataset-shift-in-machine-learning/>

