JANUS: ONLINE JACOBIAN-ALIGNED INFILL FOR BLACK-BOX OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Population optimizers collect evaluations that can also support local models of the objective. Such models can guide search, but their predictions may be unreliable under a limited budget. We introduce JANUS, a candidate-generation module that fits a local model to the host's existing evaluations and uses it to replace a few native candidates with a model-based descent proposal and metric-shaped exploration samples. For nonnegative cost components, a fitted residual Jacobian supplies both proposals through its normal matrix. With scalar feedback, an objective gradient and a positive semidefinite metric are obtained from the scalar fit. Clipping, covariance trace normalization, and limited candidate allocation control the proposal scale and budget, while the host retains evaluation-based selection and state updates. In the benchmark summaries, JANUS achieves the lowest mean error among external baselines on **12/15** covered 500D BBOB functions and reduces mean UAV planning cost by **12.8%** relative to DE, the strongest external baseline by mean cost, at 2,500 evaluations. Scalar-reference controls show function-dependent gains without a consistent advantage from sharing the fit, supporting local-model proposals as a task-dependent complement to host adaptation.

1 INTRODUCTION

Population optimizers evaluate candidates to guide selection, but the same observations also describe how the objective changes locally. Some evaluations expose richer information, such as errors for individual calibration scenes. CMA-ES and DE already use geometric information through covariance adaptation and population differences (Hansen, 2016; Storn & Price, 1997); a fitted local model offers an additional signal. The challenge is to use this signal while limiting the influence of an inaccurate model.

In high dimensions, a small evaluation window can constrain some directions while leaving others poorly identified. Following the fitted model may then give an unhelpful step or repeatedly sample uninformative directions. Local modeling and population search have complementary roles: the model suggests where to search, while the host evaluates proposals and retains alternative search directions. Local Bayesian optimization (Eriksson et al., 2019) and model-assisted evolution (Jin, 2011) provide established ways to combine local estimation with broader search.

JANUS fits a local model to the host's existing evaluations and uses it to replace a few native candidates with a model-based descent proposal and metric-shaped exploration samples. These replacement candidates, called *infill*, are evaluated alongside the remaining native candidates and passed to the host's usual update. The population size and total evaluation budget remain unchanged.

The two proposal roles use the same local metric. Exploitation takes a damped local-model descent direction; exploration allocates variance according to inverse fitted sensitivity. Damping regularizes the solves, the covariance blend weight w controls the shape of exploration, and the candidate fraction ρ controls its share of evaluations. Clipping and trace normalization keep proposals on the host's sampling scale before boundary handling.

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

For a concrete residual construction, let a calibration candidate x return one nonnegative error $r_j(x)$ per scene. The objective is their average,

$$f(x) = \frac{1}{m}\sum_{j=1}^{m} r_j(x), \qquad r_j(x) \geq 0. \tag{1}$$

Taking $s_j(x) = \sqrt{r_j(x)}$ expresses this objective exactly as $m^{-1}\|s(x)\|^2$. A local residual fit supplies a Jacobian J, a descent signal $J^\top s$, and a normal matrix $J^\top J$. JANUS reuses the normal matrix for a Gauss–Newton proposal and exploration sampling. Scalar feedback requires a distinct gradient calculation: a factorization of a fitted metric alone does not determine an objective descent direction.

Local modeling within evolutionary search is well established (Kern et al., 2006; Loshchilov et al., 2012). Guided ES shapes random search with supplied surrogate gradients (Maheswaranathan et al., 2019); EvoGrad2 combines CMA statistics with learned gradients and Hessians (Kfir et al., 2026). JANUS focuses on how a metric from the current evaluation history serves two candidate roles within an existing host. It needs no offline training, a property also shared by online operator-learning methods such as ABOM (Wang et al., 2026). Sharing the local fit is a computational and architectural choice whose optimization benefit is evaluated, not assumed.

We assess this design through cross-domain benchmark summaries and a separate scalar-reference study with recorded traces. The summaries show gains in several settings, especially at short UAV budgets; the controlled study finds objective-dependent effects without a consistent advantage from shared fitting. Our contributions are:

1. A candidate-generation interface that reuses a local metric for exploitation and exploration, with distinct constructions for residual and scalar observations.
2. Algebraic properties for fitted-model descent, proposal length, and pre-clipping covariance trace.
3. An empirical assessment combining benchmark summaries with controlled tests of proposal direction, fit sharing, and local-model quality.

2 Related Work

Geometry in population search. CMA-ES learns a search covariance from selected samples (Hansen, 2016); natural-gradient interpretations explain related distribution updates (Wierstra et al., 2014; Ollivier et al., 2017; Akimoto et al., 2010). LM-CMA and VkD-CMA reduce representation cost (Loshchilov, 2014; Akimoto & Hansen, 2016), while FOCAL modifies adaptation to recover local Hessian structure (Shir et al., 2010). JANUS supplies additional candidates from an explicit local fit, leaving the candidate-mode host's update rule intact.

Local models and guided search. Local interpolation and least-squares solvers include NEWUOA, BOBYQA, and DFO-LS (Powell, 2006; Powell et al., 2009; Cartis et al., 2019); TuRBO combines local probabilistic models with global sample allocation (Eriksson et al., 2019). Model-assisted evolution also reuses objective evaluations (Kern et al., 2006; Loshchilov et al., 2012; Jin, 2011). Guided ES uses supplied surrogate gradients (Maheswaranathan et al., 2019), and EvoGrad2 combines CMA statistics with gradient/Hessian learning (Kfir et al., 2026). JANUS exposes a fitted metric through both a point proposal and a temporary sampling covariance, with their acceptance and subsequent adaptation handled by the host.

Learned operators and online adaptation. Several MetaBBO methods learn across task distributions (Ma et al., 2023; Lange et al., 2023; Chen et al., 2024; Li et al., 2025). ABOM adapts parameterized operators using target-task data alone (Wang et al., 2026). Avoiding offline training therefore does not distinguish JANUS by itself; the local model and its candidate interface do.

Multi-objective infill. MOO hosts use nondominated sorting, hypervolume, decomposition, reference vectors, or geometric survival (Deb et al., 2002; Beume et al., 2007; Zhang & Li, 2007; Deb & Jain, 2014; Cheng et al., 2016; Panichella, 2022). Gaussian-process metamodels also guide single- and multi-objective search (Emmerich et al., 2006). JANUS uses a local objective-vector metric to generate offspring while retaining the host's survival rule.

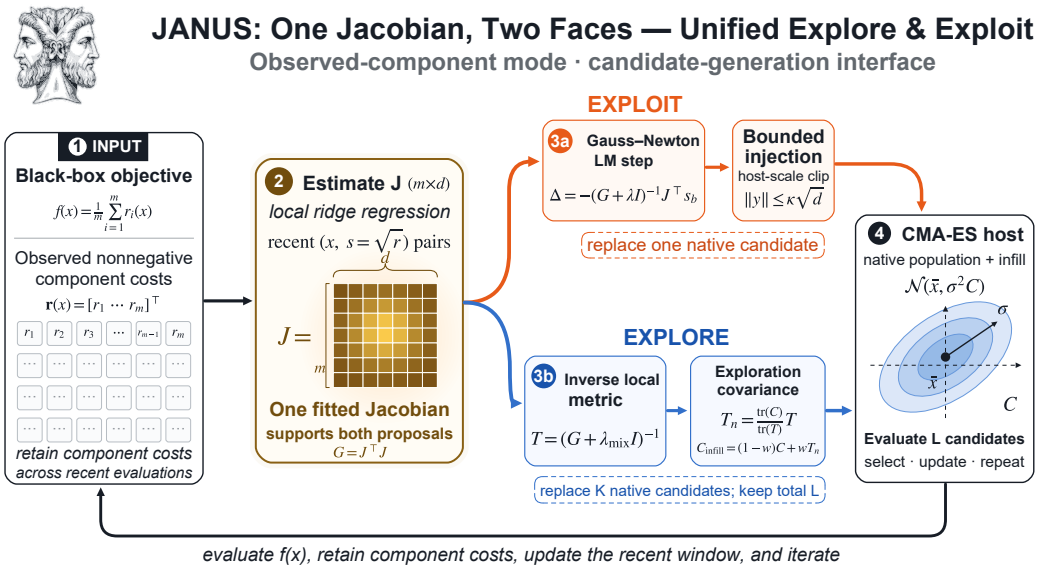


Figure 1: **JANUS with observed components and a CMA-style host.** A fitted Jacobian supplies the exploitation proposal and inverse local metric. The orange and blue branches replace existing candidate slots, after which the host evaluates and updates the population. With scalar-only feedback, the fitted objective gradient replaces $J^\top s_b$ in the exploitation step.

3 METHOD

JANUS maps recent evaluations to an exploitation point and metric-shaped exploration samples (Fig. 1). Here d denotes the search dimension.

3.1 CANDIDATE-GENERATION INTERFACE

The host $\mathcal{A}$ is the population optimizer. At a scheduled generation, it proposes L native candidates; JANUS fits the recent window $\mathcal{W}$ and replaces selected slots. All L points are evaluated and passed to the host's usual update. This preserves its update rules and evaluation budget, while external candidates can alter its state trajectory.

A CMA-style host samples $x \sim \mathcal{N}(\bar{x}, \sigma^2 C)$, with mean $\bar{x}$, step size $\sigma > 0$, and covariance shape $C \succ 0$; these set the injection scale. Other hosts need an adapter for scale, boundary handling, and external-candidate updates. Damping $\lambda, \lambda_{\text{mix}}$, blend w, and exploration fraction ρ regulate distinct aspects of the proposals.

Algorithm 1 fits and injects at scheduled generations after warm-up; intervening generations remain native. Exploration slots are budgeted from the nominal population size. An incomplete final batch is evaluated but passed to the host update only if supported.

3.2 LOCAL MODEL: A DESCENT SIGNAL AND A METRIC

For scalar minimization, the model returns a signal $q \in \mathbb{R}^d$ and a positive semidefinite (PSD) metric $G \in \mathbb{R}^{d\times d}$ in search coordinates. Let x_b be the lowest-value point in the fitting window. Standardize inputs as $z_i = D^{-1}(x_i - \mu)$, where μ is the window mean and D contains coordinate standard deviations floored away from zero. Let Z have rows $z_i^\top$. The window scales D and host covariance C serve different purposes:

Coordinates	Definition	Purpose
Search	x	Parameters evaluated by the black box
Standardized	$z = D^{-1}(x - \mu)$	Numerical scaling for fitting
Host-whitened	$y = C^{-1/2}(x - \bar{x})/\sigma$	Distance relative to the host distribution

Algorithm 1 JANUS at scheduled model-refresh generations

Input: Host $\mathcal{A}$, budget B, window n, warm-up n_0, interval h;
exploration fraction ρ, damping $\lambda, \lambda_{\text{mix}}$, blend w, clip κ

Initialize $\mathcal{W} = \varnothing$, evaluated count $t = 0$, generation $\ell = 0$
while $t < B$ **do**
  Ask for a native population of size P; retain $L = \min(P, B - t)$ candidates
  if $|\mathcal{W}| \geq n_0$, $\ell \bmod h = 0$, and $L \geq 2$ **then**
    Fit the local model; mark all L slots as available
    if scalar minimization **then**
      Select x_b; use Eq. 2 and fitted-score selection for one proposal
      Replace one slot and remove it from the available slots
    end if
    $K \leftarrow \min(\text{round}(\rho P), \text{number of available slots})$
    Replace K available slots by samples from Eq. 3
    Clip model proposals relative to the host sampling scale
  end if
  Apply boundary rules to all L points and evaluate them; update $\mathcal{W}$
  Retain the most recent n window points
  Call host update if the batch is supported; otherwise retain only its evaluations
  $t \leftarrow t + L$; $\ell \leftarrow \ell + 1$
end while

Observed nonnegative components. For Eq. 1, set $s_j = \sqrt{r_j}$ and fit b, W by minimizing $\sum_{i \in \mathcal{W}} \|s(x_i) - b - W^\top z_i\|^2 + \beta \|W\|_F^2$, where $\beta > 0$ and $W \in \mathbb{R}^{d \times m}$. Mapping slopes back gives

$$J = W^\top D^{-1}, \qquad q = J^\top s_b, \qquad G = J^\top J,$$

where $s_b = s(x_b)$. Thus one residual fit supplies both proposal roles.

Scalar observations. The gradient comes from the fitted scalar objective. A linear fit $\hat{f}(z) = b + a^\top z$ gives $q = g = D^{-1}a$ and $G = gg^\top$. For $\hat{f}(z) = c + a^\top z + \frac{1}{2} z^\top H z$, let $z_b = D^{-1}(x_b - \mu)$ and use

$$q = g = D^{-1}(a + H z_b), \qquad G = D^{-1} H_+ D^{-1},$$

where H_+ retains the positive eigenvalues. Diagonal or low-rank fits reduce estimation cost. The gradient and PSD metric are distinct outputs of the scalar fit; G is not a residual decomposition of the objective.

Multiple objectives. Fit the objective vector directly by the same local regression and form $G = J^\top J$. The MOO interface uses this metric for exploration only; it assumes no direction that decreases every objective. Pareto selection stays with the host.

3.3 Exploitation: a bounded local-model step

We regularize the metric before using it to precondition the local descent signal:

$$\Delta = -(G + \lambda I)^{-1} q, \qquad \lambda > 0. \tag{2}$$

This is a damped Gauss–Newton step in residual mode, denoted Δ_{GN}, and a metric-preconditioned gradient step in scalar mode. Choose $\eta \in \{0.25, 0.5, 1\}$ to minimize the fitted score $\|s_b + \eta J \Delta\|^2$ or $\hat{f}(z_b + \eta D^{-1}\Delta)$, without black-box calls, then set $x_{\text{raw}} = x_b + \eta\Delta$.

Following Hansen (2011), transform x_{raw} to host-whitened coordinates and clip its radius to $\kappa\sqrt{d}$, obtaining y_e. Map back as $x_e = \bar{x} + \sigma C^{1/2} y_e$ and apply boundary handling. The clip is centered at $\bar{x}$, so it can change the direction from x_b; model-relative descent concerns the unclipped direction.

3.4 Exploration: inverse local metric sampling

Covariance mixing (CMIX) allocates exploration variance using the inverse local metric:

$$T = (G + \lambda_{\text{mix}} I)^{-1}, \quad T_n = \frac{\text{tr}(C)}{\text{tr}(T)} T, \quad C_{\text{infill}} = (1 - w)C + w T_n, \tag{3}$$

with $\lambda_{\text{mix}} > 0$ and $0 \leq w \leq 1$. Samples come from $\mathcal{N}(\bar{x}, \sigma^2 C_{\text{infill}})$ before clipping and boundary handling. If $Gv = \gamma v$, the inverse metric assigns variance proportional to $(\gamma + \lambda_{\text{mix}})^{-1}$ along v. Low fitted sensitivity can reflect a flat direction or insufficient observations. Damping keeps the inverse well-defined, while normalization gives $\text{tr}(C_{\text{infill}}) = \text{tr}(C)$. This temporary covariance generates candidates; it is not assigned to the host state.

3.5 Controlling the model's influence

ρ controls the fraction of slots assigned to exploration; w controls the covariance shape within those slots. Thus $\rho = 0$ gives the exploitation-only JANUS-GN variant, while $w = 1$ gives pure inverse-metric sampling for the allocated slots. With m observed component or objective coordinates, some blended configurations use

$$w = \text{clip}\left(w_{\text{base}} \frac{\min(m, |\mathcal{W}|)}{d}, w_{\min}, w_{\max} \right). \tag{4}$$

This count-based heuristic is distinct from a measure of fit reliability. The early AWB deployment reshapes the host covariance directly and is identified separately in Sec. 5.5.

4 Theoretical Properties

The first property verifies the exploitation direction on the fitted model; the second bounds displacement and preserves pre-clipping sampling variance. Neither guarantees improvement on the black-box objective.

Proposition 1 (Descent on the fitted residual model)**.** *For $\hat{f}(\Delta) = \|s_b + J\Delta\|^2$ and $\lambda > 0$, the residual step satisfies*

$$\nabla \hat{f}(0)^\top \Delta_{\text{GN}} = -2 s_b^\top J (J^\top J + \lambda I)^{-1} J^\top s_b \leq 0,$$

with strict inequality when $J^\top s_b \neq 0$.

For scalar fits, $g^\top \Delta = -g^\top (G + \lambda I)^{-1} g < 0$ when $g \neq 0$. Both statements follow from positive definiteness and concern the direction before clipping or boundary handling.

Proposition 2 (Bounded proposals and normalized exploration)**.** *Clipping to $\|y_e\| \leq \kappa\sqrt{d}$ gives $\|x_e - \bar{x}\| \leq \kappa\sigma\lambda_{\max}(C^{1/2})\sqrt{d}$. Moreover, Eq. 3 satisfies $\text{tr}(C_{\text{infill}}) = \text{tr}(C)$ before clipping and boundary handling.*

The length bound follows from the operator norm, and trace preservation from linearity. They control scale, not the probability of visiting useful regions.

What a finite window identifies. For centered linear residual regression, $\text{rank}(J) \leq \min(m, \text{rank}(Z)) \leq \min(m, n-1, d)$. Hence some directions may remain unidentified. If $Jv = 0$, then $(J^\top J + \lambda_{\text{mix}} I)^{-1} v = \lambda_{\text{mix}}^{-1} v$: this eigenvalue is set by damping rather than measured sensitivity. This rank argument does not apply to a general scalar quadratic proxy. Complete proofs and the separate rank-deficient covariance bound are in App. E.

5 Experiments

We ask whether local-model candidates improve fixed-budget outcomes, how the two proposal roles behave across tasks and hosts, and whether controlled comparisons support benefits from their directions or shared fit.

Benchmark summaries describe BBOB, UAV, and MOO outcomes; recorded scalar-reference controls test proposal directions and fit sharing; AWB illustrates an early reshaping deployment. BBOB supplies scalar values, MOO objective vectors, and AWB scene errors. The UAV summaries establish total costs but do not specify the component interface. Original BBOB/MOO adapters and run-level records are unavailable, so those summaries cannot verify execution of the current method. We distinguish these descriptive results from the controlled study below; implementation details are collected in App. A.1.

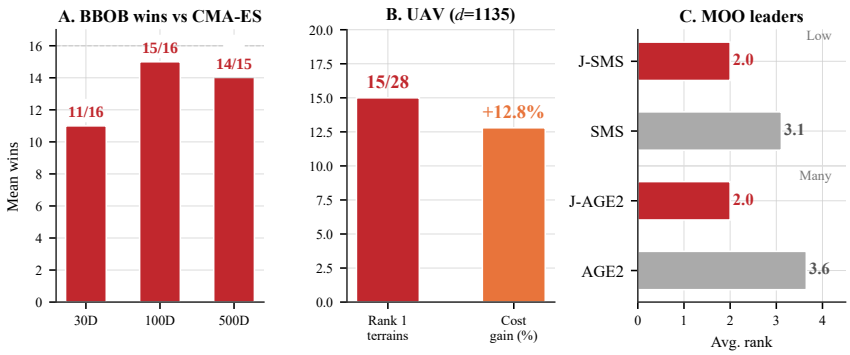


Figure 2: **JANUS at a glance.** BBOB mean wins over CMA-ES, UAV performance, and MOO average ranks. The BBOB counts use the tabulated means and available function coverage (Sec. 5.1); they are not significance counts.

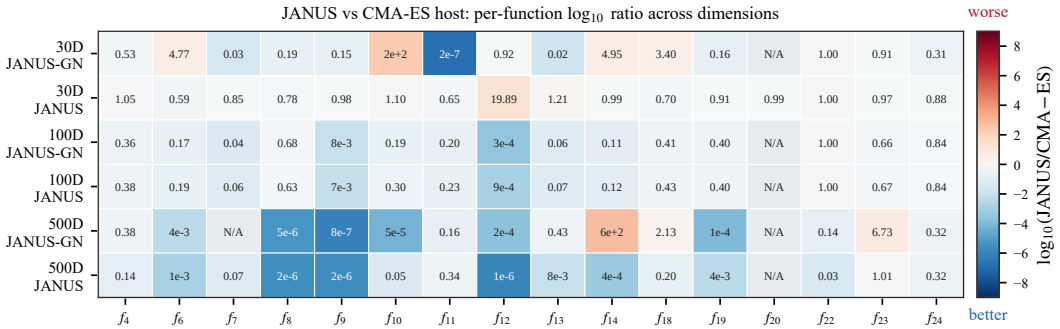


Figure 3: Per-function ratios of the tabulated BBOB means in Tables 11 and 1. Cells show the ratio; color encodes its $\log_{10}$ (blue=better, red=worse). N/A marks missing coverage or nonpositive values, for which a log ratio is undefined.

5.1 Scalar BBOB: Geometry from Scalar Evaluations ($d \in \{30, 100, 500\}$)

BBOB exposes scalar feedback. The reported MetaBox/COCO-compatible protocol (Ma et al., 2023) uses 16 functions, $[-5, 5]^d$, and 20,000 FEs. Tuned 30D JANUS uses 10 runs, other 30D columns and all 100D columns use 30; 500D JANUS/external baselines use 10, with GN-only a separate 30-run reference. JANUS-GN denotes exploitation only, and full JANUS also enables CMIX.

Comparison with CMA-ES. JANUS has lower means than CMA-ES on 11/16, 15/16, and 14/15 functions at 30D, 100D, and 500D (Figs. 2–3, Table 1). The 500D host lacks f_{20}. These are mean comparisons, not significance counts; losses occur on $f_4, f_{10}, f_{12}, f_{13}$ at 30D and f_{23} at 500D. Table 11 retains the lower-dimensional means and standard deviations.

External baselines at 500D. JANUS leads external baselines on 12/15 covered functions; RLEPSO leads f_{11}, f_{24} and SYMBOL f_{23}. Over this coverage, geometric means of rounded per-function means are 339 for JANUS, 6.75×10^4 for CMA-ES, and 1.02×10^4 for LDE, the strongest external aggregate: about a $30\times$ reduction relative to LDE. Including GN-only gives JANUS nine best cells. Table 1 shows the principal comparisons; Table 12 expands the baseline set.

Local solvers and a DE host. With a DE host (30 runs per function), JANUS-DE wins 86%/91%/96% of run comparisons at 30/100/500D, with geometric-mean ratios 0.86/0.58/0.59. This extends the reported candidate-interface results beyond CMA-ES. Table 13 retains these summaries and separate local-solver checks; they do not establish a general ordering between optimizer families.

5.2 High-Dimensional UAV Path Planning (d=1135)

The UAV task optimizes a 1135-dimensional spherical-coordinate flight path under terrain-dependent costs. The reported protocol covers 28 held-out terrains and 30 runs each (840 final values per method)

Table 1: **BBOB $d = 500$: mean final error on the 15 functions with external baseline coverage ($[-5, 5]^d$, 20,000 FEs; f_{20} omitted). External baselines and JANUS use 10 runs; JANUS-GN is a 30-run reference. Bold blue=best, green=second. Lower is better. Wins count best means within this displayed subset, including JANUS-GN; they are not significance counts. Full JANUS leads external baselines on 12/15 functions. The full table is in App. C.1.

	Host	NN-BBO / MetaBBO baselines				Ours	
Func	**CMA-ES**	**RS**	**SYMBOL**	**RLEPSO**	**LDE**	**JANUS-GN**	**JANUS**
f_4	4.2e4	3.7e5	1.2e5	2.5e4	1.6e4	1.6e4	**5992**
f_6	1.8e6	1.6e7	5.6e6	9.2e5	9.9e5	7103	**2113**
f_7	1.5e4	3.9e4	1.8e4	1.2e4	4824	–	**988**
f_8	5.9e8	5.9e8	4.7e7	1.5e7	1.3e7	3072	**1426**
f_9	6.1e8	4.1e8	2.4e6	9.5e5	1.8e5	**508.7**	1323
f_{10}	3.3e7	2.8e8	8.9e7	1.8e7	1.1e7	**1799**	1.5e6
f_{11}	1.2e4	5833	3099	2715	3141	**1934**	4037
f_{12}	8.2e9	3.0e10	4.8e9	1.9e9	1.5e9	1.8e6	**1.0e4**
f_{13}	1.1e4	1.5e4	8913	5772	5347	4728	**88.6**
f_{14}	267	655	176	82.5	73.0	1.5e5	**0.12**
f_{18}	73.9	140	70.3	54.7	40.6	157.6	**14.5**
f_{19}	3006	2096	24.2	22.4	12.3	**0.34**	11.7
f_{22}	84.1	86.4	81.2	63.6	62.2	11.6	**2.87**
f_{23}	1.65	1.64	**1.46**	1.65	1.64	11.1	1.66
f_{24}	2.3e4	2.1e4	9315	**7426**	7660	7448	7435
Wins	0	0	1	1	0	4	**9**

at 2,500 FEs. Baselines include DE, PSO, RS, JDE21 (Brest et al., 2021), CMA-ES, SAHLPSO (Tao et al., 2021), and MetaBBO methods including RLDEAFL (Guo et al., 2025).

Table 2: **UAV main comparison at 2,500 FEs (d=1135).** Lower is better except rank-1 counts. Pale blue marks JANUS; blue/green mark best/second among the displayed methods. The full 12-method, 28-terrain table and path visualizations are in App. C.3.

Method	**Mean**	**Median**	**Rank-1**
JANUS	**9,600**	**6,644**	**15**
DE	11,006	9,261	1
JDE21	11,017	8,579	1
RLDEAFL	11,411	8,197	0
CMA-ES	14,486	10,908	0

JANUS achieves the lowest mean cost (9,600) and median (6,644), with mean cost 12.8% below DE (11,006), the strongest external baseline by mean cost (Table 2). Across terrains, JANUS ranks first on 15/28 when all 12 methods, including GN-only, are compared. The aggregate improvement therefore coexists with terrain-dependent outcomes.

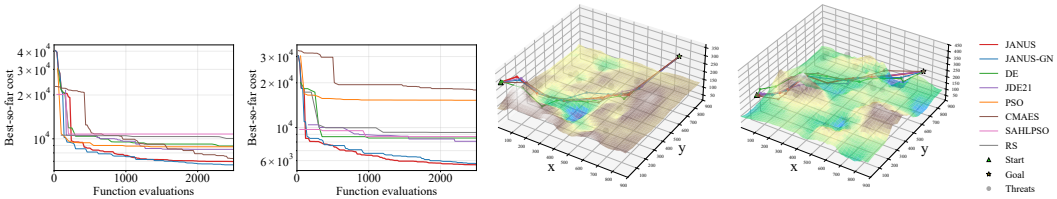
The reported Bonferroni-corrected Mann–Whitney tests yield 24 improvements, three comparisons without a detected difference, and one loss against CMA-ES. The per-terrain results and separate BO-like subset checks are in App. C.3 and Table 13; the latter cover different terrains and do not rank JANUS against BO-like methods.

Budget dependence. At 10,000 FEs, JANUS and CMA-ES mean costs are 8,488 and 8,587; at 20,000 they are 8,019 and 8,023 (Table 15). The advantage is concentrated at the short budget and approaches parity as the budget increases.

Figure 4 shows early improvements and representative paths on T1/T17. These examples illustrate the short-budget trajectories; the full terrain table also includes GN-only wins (App. C.3).

5.3 Multi-Objective Optimization (38 tasks)

For MOO, the objective vector supplies a local metric for exploration offspring ($\rho = 5\%$), while the host retains Pareto survival. The low-objective suite has 18 ZDT/DTLZ/WFG tasks with 2–3 objectives and an SMS-EMOA host; the many-objective suite has 20 tasks with 5/8/10 objectives and AGE-MOEA2. Each uses 20,000 evaluations and 10 runs. Hypervolume (HV) measures dominated



(a) T1 convergence (b) T17 convergence (c) T1 paths (d) T17 paths

Figure 4: **UAV convergence and path quality** on two representative terrains. JANUS separates early and produces smooth low-cost paths; JANUS-GN follows more conservatively, while other baselines often plateau higher under the 2,500-FE budget.

objective-space volume up to a reference point; larger is better. Table 3 retains the reported ranks and normalized HV; their ranking pool and normalization details are unavailable.

Average rank and hypervolume. The 5% variants have reported average rank 2.00, with normalized HV 0.994 versus 0.972 for SMS-EMOA and 0.972 versus 0.915 for AGE-MOEA2. Rank and HV do not always favor the same variant: JANUS-AGE2 2% has HV 0.973 and rank 2.30. The original tests report improvements on 3/18 and 9/20 tasks; these unadjusted counts establish neither familywise significance nor equivalence elsewhere.

Table 3: **MOO summaries:** 20,000 FEs, 10 runs/task. Ranks are reported, not recomputed over these rows. Blue/green mark best/second displayed values per regime. $+/\approx/-$: reported unadjusted Mann–Whitney wins/no detected difference/losses against the host ($p < 0.05$); per-run tests cannot be revalidated.

Regime	Method	Avg. Rank	Norm. HV	$+/\approx/-$
Low	**JANUS-SMS 5%**	**2.00**	**0.994**	3/15/0
(2–3 obj.)	JANUS-SMS 2%	2.50	0.993	–
	SMS-EMOA	3.11	0.972	–
	NSGA-II	3.61	0.967	–
	MOEA/D	5.00	0.926	–
	Random	6.50	0.756	–
	CMA-ES	8.11	0.711	–
Many	**JANUS-AGE2 5%**	**2.00**	0.972	9/11/0
(5/8/10 obj.)	JANUS-AGE2 2%	2.30	**0.973**	–
	AGE-MOEA2	3.65	0.915	–
	NSGA-III	5.00	0.888	–
	RVEA	6.40	0.865	–
	NSGA-II	6.50	0.917	–
	MOEA/D	7.45	0.820	–

With CUDA-accelerated metric estimation, the 5%-infill variants take $1.46\times$ and $1.22\times$ their SMS-EMOA and AGE-MOEA2 hosts' runtime. This overhead matters when objective evaluations are cheap.

5.4 Ablation and Controlled Diagnostics

Table 4(a) separates the proposal roles in the reported BBOB/UAV configurations. Full JANUS has the lowest BBOB ratio (0.83); CMIX-only has the lowest UAV ratio (0.88). The preferred combination depends on the task. The historical MOO GN entries, whose scalarization is unspecified, are retained separately in Table 19; they do not test the exploration-only MOO interface defined here.

Does sharing help? Panel (b) summarizes 288 scalar-reference runs: three functions, two dimensions, eight seeds, six variants, and 3,000 FEs. Shared uses one diagonal quadratic fit. Random and orthogonally rotated controls preserve proposed whitened radii; shared-B and separate-B use common or independent exponentially reweighted fits on the same window. Shared/host ratios are 0.814/0.716 on f_6, with smaller changes on f_{22}, f_{23}. Shared-B/separate-B reverses ordering across settings. None of 24 paired log-error sign-flip contrasts survives Holm correction. These results distinguish task-dependent proposal gains from an unestablished general advantage of sharing.

Table 4: **Proposal roles and fit-sharing controls.** (a) Reported ratios to the host. (b) Paired geometric-mean error ratios, eight seeds, 3,000 FEs. Shared-B/separate-B compares common/independent fit reweighting. Lower ratios favor the numerator.

(a) Component configurations: ratio to the host				
Variant	GN	CMIX	BBOB	UAV
Base host			1.00	1.00
GN only	✓		0.85	0.91
CMIX only		✓	0.92	**0.88**
JANUS	✓	✓	**0.83**	0.89

(b) Scalar reference: function-dependent effects				
	100D		500D	
Function	Shared/host	Shared-B/separate-B	Shared/host	Shared-B/separate-B
f_6	0.814	0.888	0.716	1.088
f_{22}	0.985	1.000	0.977	1.022
f_{23}	0.997	1.028	0.998	1.016

5.5 Industrial Case Study: Auto White Balance

We calibrate auto white balance (AWB) in a proprietary image-processing pipeline with 1135 parameters and 111 scenes. Each evaluation runs the pipeline across all scenes. Table 5 reports running-best mean reproduction angle error (MRAE) at indexed checkpoints through 5,000 in longer runs. The early GN+CMIX deployment reduces MRAE by 22.5% relative to CMA-ES. It reshapes the host covariance directly, unlike candidate-only CMIX. Because the baseline counter includes resumed trial IDs, the table is an indexed trajectory comparison rather than a certified match of physical evaluation counts.

Table 5: **AWB calibration.** Running-best MRAE at indexed trial checkpoints; lower is better. One run per method; GN+CMIX uses the early covariance-reshaping implementation. Blue=best, green=second.

Method	**@1000**	**@2000**	**@3000**	**@4000**	**@5000**
CMA-ES (baseline)	1.526	1.276	1.177	1.127	1.056
JANUS-GN (exploit only)	**1.154**	**0.968**	0.918	0.866	0.837
JANUS (GN + CMIX reshape)	1.200	0.993	**0.897**	**0.863**	**0.819**

5.6 Implementation Settings

The reported configurations use a 300-point window. Benchmark settings are ridge 10^{-3}, $\lambda \in [1, 10]$, $\kappa = 1$, and refresh every 4–6 generations (50 for 30D/100D BBOB); MOO uses $\rho = 0.05$. AWB uses ridge/damping 1, warm-up 40, and refresh 4. The scalar reference uses diagonal quadratic fits, ridge/damping 1, warm-up 80, refresh 4, $\kappa = 1$, $w = 1$, and $\rho = 0.2$.

6 Conclusion

JANUS exposes local geometry through a model-based point and metric-shaped exploration, with clipping, trace normalization, and candidate allocation controlling their influence on the host. Benchmark summaries indicate function-dependent and short-budget gains; controlled tests do not establish a consistent advantage from sharing the fit. The design confines local-model guidance to candidate generation, leaving acceptance and adaptation to the host.

AI USE STATEMENT

Generative AI was used during the present manuscript revision for language editing, condensing and reorganizing text, checking numerical summaries, clarifying statistical interpretation and theoretical scope, preparing the vector method diagram, and assisting with reference code and supplementary diagnostic analyses. This statement covers only the present revision. The authors are responsible for the final content of the submission.

REFERENCES

Youhei Akimoto and Nikolaus Hansen. Projection-based restricted covariance matrix adaptation for high dimension. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, pp. 197–204, 2016.

Youhei Akimoto, Yuichi Nagata, Isao Ono, and Shigenobu Kobayashi. Bidirectional relation between cma evolution strategies and natural evolution strategies. In *Parallel Problem Solving from Nature, PPSN XI*, pp. 154–163. Springer, 2010.

Nicola Beume, Boris Naujoks, and Michael Emmerich. Sms-emoa: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653–1669, 2007.

Janez Brest, Mirjam Sepesy Maučec, and Borko Bošković. Self-adaptive differential evolution algorithm with population size reduction for single objective bound-constrained optimization: Algorithm j21. In *2021 IEEE Congress on Evolutionary Computation (CEC)*, pp. 817–824. IEEE, 2021.

Coralia Cartis, Jan Fiala, Benjamin Marber, and Lindon Roberts. Improving the flexibility and robustness of model-based derivative-free optimization solvers. *ACM Transactions on Mathematical Software*, 45(3):1–41, 2019.

Jiacheng Chen, Zeyuan Ma, Hongshu Guo, Yining Ma, Jie Zhang, and Yue-Jiao Gong. SYMBOL: Generating flexible black-box optimizers through symbolic equation learning. In *The Twelfth International Conference on Learning Representations*, 2024.

Ran Cheng, Yaochu Jin, Markus Olhofer, and Bernhard Sendhoff. A reference vector guided evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 20(5):773–791, 2016.

Kalyanmoy Deb and Himanshu Jain. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, 18(4):577–601, 2014.

Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

Michael T. M. Emmerich, Kyriakos C. Giannakoglou, and Boris Naujoks. Single- and multiobjective evolutionary optimization assisted by gaussian random field metamodels. *IEEE Transactions on Evolutionary Computation*, 10(4):421–439, 2006.

David Eriksson, Michael Pearce, Jacob Gardner, Ryan D. Turner, and Matthias Poloczek. Scalable global optimization via local Bayesian optimization. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Hongshu Guo, Sijie Ma, Zechuan Huang, Yuzhi Hu, Zeyuan Ma, Xinglin Zhang, and Yue-Jiao Gong. Reinforcement learning-based self-adaptive differential evolution through automated landscape feature learning. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 1117–1126, 2025.

Nikolaus Hansen. Injecting external solutions into cma-es. Research Report RR-7748, INRIA, October 2011.

Nikolaus Hansen. The cma evolution strategy: A tutorial. *arXiv preprint arXiv:1604.00772*, 2016.

Yaochu Jin. Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation*, 1(2):61–70, 2011.

Stefan Kern, Nikolaus Hansen, and Petros Koumoutsakos. Local meta-models for optimization using evolution strategies. In *Parallel Problem Solving from Nature IX*, pp. 939–948, 2006. doi: 10.1007/11844297_95.

Yedidya Kfir, Elad Sarafian, Yoram Louzoun, and Sarit Kraus. Evograd: Evolutionary-weighted gradient and hessian learning for black-box optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 36991–36999, 2026. doi: 10.1609/aaai.v40i43.41027.

Robert Tjarko Lange, Tom Schaul, Yutian Chen, Tom Zahavy, Valentin Dalibard, Chris Lu, Satinder Singh, and Sebastian Flennerhag. Discovering evolution strategies via meta-black-box optimization. In *International Conference on Learning Representations*, 2023.

Xiaobin Li, Kai Wu, Yujian Betterest Li, Xiaoyu Zhang, Handing Wang, and Jing Liu. Pretrained optimization model for zero-shot black box optimization. *Advances in Neural Information Processing Systems*, 37:14283–14324, 2024.

Xiaobin Li, Kai Wu, Xiaoyu Zhang, and Handing Wang. B2opt: Learning to optimize black-box optimization with little budget. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39 (17):18502–18510, Apr. 2025.

Ilya Loshchilov. A computationally efficient limited memory cma-es for large scale optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 397–404, 2014.

Ilya Loshchilov, Marc Schoenauer, and Michele Sebag. Self-adaptive surrogate-assisted covariance matrix adaptation evolution strategy. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 321–328, 2012. doi: 10.1145/2330163.2330210.

Zeyuan Ma, Hongshu Guo, Jiacheng Chen, Zhenrui Li, Guojun Peng, Yue-Jiao Gong, Yining Ma, and Zhiguang Cao. Metabox: A benchmark platform for meta-black-box optimization with reinforcement learning. *Advances in Neural Information Processing Systems*, 36:10775–10795, 2023.

Zeyuan Ma, Jiacheng Chen, Hongshu Guo, Yining Ma, and Yue-Jiao Gong. Auto-configuring exploration-exploitation tradeoff in evolutionary computation via deep reinforcement learning. In *Proceedings of the Genetic and Evolutionary Computation Conference*, 2024.

Niru Maheswaranathan, Luke Metz, George Tucker, Dami Choi, and Jascha Sohl-Dickstein. Guided evolutionary strategies: Augmenting random search with surrogate gradients. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*. PMLR, 2019. URL https://proceedings.mlr.press/v97/maheswaranathan19a.html.

Yann Ollivier, Ludovic Arnold, Anne Auger, and Nikolaus Hansen. Information-geometric optimization algorithms: A unifying picture via invariance principles. *Journal of Machine Learning Research*, 18(18):1–65, 2017.

Annibale Panichella. An improved pareto front modeling algorithm for large-scale many-objective optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 565–573, 2022.

Michael J. D. Powell. The newuoa software for unconstrained optimization without derivatives. *Large-Scale Nonlinear Optimization*, pp. 255–297, 2006.

Michael JD Powell et al. The bobyqa algorithm for bound constrained optimization without derivatives. *Cambridge NA Report NA2009/06, University of Cambridge, Cambridge*, 26(26-46):1, 2009.

Ofer M. Shir, Jonathan Roslund, and Herschel Rabitz. Forced optimal covariance adaptive learning: modified cma-es for efficient hessian determination. In *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*, GECCO '10, pp. 421–422, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781450300728. doi: 10.1145/1830483.1830563. URL https://doi.org/10.1145/1830483.1830563.

Rainer Storn and Kenneth Price. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359, 1997. doi: 10.1023/A:1008202821328.

Jianyong Sun, Xin Liu, Thomas Bäck, and Zongben Xu. Learning adaptive differential evolution algorithm from optimization experiences by policy gradient. *IEEE Transactions on Evolutionary Computation*, 25(4):666–680, 2021.

Xinmin Tao, Xiangke Li, Wei Chen, Tian Liang, Yetong Li, Jie Guo, and Lin Qi. Self-adaptive two roles hybrid learning strategies-based particle swarm optimization. *Information Sciences*, 578: 457–481, 2021.

Chao Wang, Licheng Jiao, Lingling Li, Jiaxuan Zhao, Guanchun Wang, Fang Liu, and Shuyuan Yang. Task-free adaptive meta black-box optimization. In *International Conference on Learning Representations*, 2026.

Daan Wierstra, Tom Schaul, Jan Peters, and Jürgen Schmidhuber. Natural evolution strategies. *Journal of Machine Learning Research*, 15:949–980, 2014.

Shiyuan Yin, Yi Liu, GuoLiang Gong, Huaxiang Lu, and Wenchang Li. Rlepso: Reinforcement learning based ensemble particle swarm optimizer. In *Proceedings of the 2021 4th International Conference on Algorithms, Computing and Artificial Intelligence*, pp. 1–6, 2021.

Taeyoung Yun, Kiyoung Om, Jaewoo Lee, Sujin Yun, and Jinkyoo Park. Posterior inference with diffusion models for high-dimensional black-box optimization. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=EXds2NBOoq.

Qingfu Zhang and Hui Li. Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, 2007.

Zhengfei Zhang, Yunyue Wei, and Yanan Sui. An invariant information geometric method for high-dimensional online optimization. In *6th Annual Learning for Dynamics & Control Conference*, pp. 641–653. PMLR, 2024.

JANUS: Online Jacobian-Aligned Infill

Supplementary Material

A Implementation Scope and Reproducibility

A.1 Implementation Provenance

The evidence comes from four sources, summarized in Table 6. This distinction separates the candidate interface defined in the main paper from the implementation used for each result.

Table 6: **Implementation and evidence sources.**

Source	Available material	Scope
Benchmark summaries	Aggregate BBOB/UAV/MOO tables and figures	Describe outcomes; original BBOB/MOO adapters and per-run records are unavailable
Released sampler	Anonymous snapshot: component collection, candidate and reshaping modes	Inspectable implementation with its own defaults and cadence
AWB deployment	Indexed trajectories from one longer run per method	Early covariance reshaping; resumed baseline IDs prevent certified physical-FE matching
Scalar reference	Fixed code, settings and 288 recorded runs	Short-budget controls and diagnostics; not a reproduction of the original benchmark tables

Released sampler. Inspection of the supplied anonymous snapshot verifies that residual regression standardizes inputs and maps slopes back to search coordinates. The collector requires component losses or scene errors and skips pure scalar trials without those fields; its linear setting follows the observed-residual branch. Its nonlinear scalar branch uses a separately fitted linear gradient. Single-objective quadratic metrics retain positive Hessian eigenvalues, whereas one MOO diagonal variant uses absolute diagonal curvature. Its MOO component path square-roots nonnegative component losses, unlike the raw-objective-vector fit specified in Sec. 3.

Sampling and cadence. The released candidate sampler uses the trace-normalized metric and may reuse a cached metric between Jacobian refreshes; a separate mode assigns the mixture to the host covariance. The AWB deployment uses the reshaping mode. Algorithm 1 and the scalar reference instead fit and inject only at scheduled generations. The release's MOO example uses NSGA-II and does not reproduce the missing SMS-EMOA/AGE-MOEA2 adapters.

Defaults and benchmark mapping. Released defaults include ridge and both damping values equal to 1, automatic window/warm-up, a four-generation refresh interval, and candidate fraction 0.4. These differ from some archived benchmark settings. Original BBOB/MOO adapters and run-level records are unavailable, so the summaries do not verify the exact scalar extension or implementation in Sec. 3. In particular, the original MOO component-ablation labels do not specify the scalarization behind their GN entries; the MOO interface defined here uses metric-shaped exploration only.

Statistical scope. The original MOO summaries report 12/38 taskwise improvements without a reported regression. Their per-run data do not accompany the summaries, so pairing, test sidedness, reference-point construction, effect-size intervals, and corrected p-values cannot be reconstructed. These counts remain reported taskwise annotations. The separate scalar-reference tests have recorded traces, explicit pairing, and Holm correction; they do not validate the original MOO tests.

A.2 Notation and Variants

Table 7 defines the common interface while keeping the residual, scalar, and objective-vector constructions distinct.

Table 7: **Notation and variants.** Residual and scalar observations share the candidate interface but supply different descent signals.

Symbol	Meaning
$x, f(x)$	Search parameters and the scalar black-box value
$\mathcal{W}$	Recent evaluated points used for fitting
μ, D	Window mean and diagonal coordinate scales for standardized fitting
$\bar{x}, \sigma, C$	CMA host mean, step size and covariance shape for injection scale
$s_j(x) = \sqrt{r_j(x)}$	Residual coordinate for an observed nonnegative component
J	Fitted residual or objective-vector Jacobian in search coordinates
g	Fitted scalar-objective gradient at the window incumbent
q	Exploitation signal: $J^\top s_b$ for residuals, g for scalar feedback
G	Positive semidefinite local metric; $J^\top J$ for residual/vector fits
$\lambda, \lambda_{\mathrm{mix}}$	Damping for exploitation and exploration, respectively
ρ, w	Exploration slot fraction and covariance blend weight, respectively
κ	Clipping-radius multiplier in host-whitened coordinates
Variant	**Mechanism**
JANUS-GN	Exploitation-only label; scalar mode uses the fitted objective gradient
JANUS	Exploitation and metric-shaped exploration in the candidate interface
JANUS-SMS / AGE2	Metric-shaped offspring with SMS-EMOA / AGE-MOEA2 survival
JANUS-DE	Candidates passed through a DE adapter
Random / rotated	Direction controls with matched proposed whitened radii
AWB reshaping	Early deployment that directly mixes the host covariance

A.3 Host Adapters and Protocol Controls

The host adapter provides a search scale, boundary rules, and a valid way to include external candidates in its update. A CMA adapter uses $\sigma^2 C$ for scale and host-whitened clipping. DE and MOO adapters need their own corresponding definitions; an ask/tell interface alone does not fix them. The candidate-only interface leaves direct covariance updates to the host, while a reshaping implementation must be named explicitly.

Fitting coordinates and host-whitened coordinates serve different purposes. The former standardize the regression window; the latter measure a candidate's distance relative to the current population distribution. Slopes must be mapped back before an exploitation displacement is added to a search-space point. Diagnostic fitting uses its own random stream and held-out split and does not generate optimization candidates.

The scalar reference's final partial batch consumes the remaining evaluation budget but does not update a full-population host. Its exploration quota is computed from the nominal population size and capped by the available retained slots. This convention preserves the logged evaluation count without inventing a partial-population update.

A.4 Related Model-Assisted Designs

Table 8 compares the information and model roles described in the cited papers. It is a mechanism comparison, not a matched numerical benchmark. The ABOM strip uses a different search domain, and no matched EvoGrad2 experiment is reported.

Table 8: **Related model-assisted designs.** Information sources and uses described in the cited work; this table makes no numerical superiority claim.

Method	Information source	Model and role in search
Guided ES (Maheswaranathan et al., 2019)	Supplied surrogate gradients plus objective queries	A guiding subspace shapes finite-difference sampling for a descent estimate
TuRBO (Eriksson et al., 2019)	Objective evaluations in local regions	Local Gaussian-process models propose points; a bandit strategy allocates samples across regions
EvoGrad2 (Kfir et al., 2026)	Objective queries and CMA statistics	Gradient/Hessian learning combines explicit local estimates with evolutionary information
ABOM (Wang et al., 2026)	Populations and fitness on the target task	Parameterized evolutionary operators adapt online using an elite archive
JANUS	Recent objective values or observed components	A local signal and metric generate descent/exploration proposals in selected host slots

B Experimental Protocols and Hyperparameters

This section records search spaces, budgets, run counts, and the hyperparameters accompanying the benchmark summaries. These are not universal JANUS defaults; AWB and the scalar reference use their separately stated configurations.

B.1 Search Spaces and Evaluation Budgets

BBOB. The tabulated comparisons use $[-5, 5]^d$ and 20,000 FEs. Table 11 uses 10 runs for tuned 30D JANUS, 30 for other 30D columns, and 30 for all 100D columns. At 500D, JANUS and external baselines use 10 runs; JANUS-GN is a separate 30-run reference. External baselines have no f_{20} result, leaving 15 functions with common coverage. Mean comparisons and geometric means use this coverage explicitly; the figures use the same tabulated means. ABOM is a separate reported reference with a different search box.

UAV path planning. UAV uses a 1135-dimensional spherical-coordinate path parameterization decoded to Cartesian trajectories before evaluation. We use 28 held-out terrains, 30 runs, and a 2,500-FE budget; the long-budget check repeats the protocol at 10,000 and 20,000 FEs. Table 17 keeps per-terrain results because the terrain distribution is heterogeneous.

Multi-objective optimization. MOO experiments use pymoo tasks and host survival operators: SMS-EMOA-style selection for low-objective tasks and AGE-MOEA2 (Panichella, 2022)-style survival for many-objective tasks. Each task uses 20,000 evaluations and 10 runs. The objective vector supplies the local metric. The original summaries describe unified post-hoc reference points, but the complete task list, reference-point construction, HV-normalization formula, and ranking pool are not recoverable. In particular, ranks above seven in Table 3 cannot be ranks computed solely among its seven displayed methods per regime; no such recomputation is implied.

Table 9: **Settings reported with the original benchmark summaries.** These ranges do not specify the verified AWB deployment or new reference diagnostics; see Sec. 5.6 and App. A.3.

Symbol	**Value**	**Meaning**
window	300	Recent inputs and scalar values or residuals for local fitting
β	10^{-3}	Ridge regularization for local regression
λ	$[1, 10]$	Damping in the exploitation system
κ	1.0	Bounded-injection clip factor
w_{base}	0.5	Base mixing weight before dimension scaling
$[w_{\min}, w_{\max}]$	$[0.02, 0.6]$	Clip range for the adaptive mixing weight
assembly period	4–6 gen†	Generations between Jacobian re-estimation (†50 for the scalar BBOB setting)
ρ (MOO)	0.05	Fraction of offspring allocated to JANUS infill

Reported sensitivity sweep. The 100D BBOB study varies window size, exploitation damping, exploration damping, candidate fraction, and mixing-weight bounds across 18 one-at-a-time configurations. Most configurations remain within $1.5\times$ the default geometric-mean final error, with the largest reported variation in window size. A short window supplies fewer observations, while a long window may include less representative regions. These considerations motivate the tradeoff, but the sweep measures final objective values rather than Jacobian accuracy.

C Additional Benchmark Results

C.1 BBOB Per-Function Results

Scalar BBOB provides objective values rather than an observed residual vector, making the choice of local scalar model important. The per-function tables preserve differences that aggregate summaries can hide. Table 10 is a reported ABOM reference under a different search domain; it is separate from the host comparisons that follow.

Table 10: **Dense BBOB function strip.** Reported $\log_{10}(\mathrm{JANUS}/\mathrm{ABOM})$ final-value ratios under different search domains. Negative values denote lower reported JANUS values, not matched-protocol wins. Green/red text marks the sign; the pale-blue column averages available ratios.

Function	30D	100D	500D	Below ref.	Mean log-ratio	Pattern
F4	-0.75	-1.08	-0.23	3/3	-0.69	all lower
F6	-9.39	-4.54	-0.30	3/3	-4.74	all lower
F7	-1.84	-2.11	-0.67	3/3	-1.54	all lower
F8	-1.53	-1.57	-1.66	3/3	-1.59	all lower
F9	-3.25	-3.05	-2.04	3/3	-2.78	all lower
F10	-3.30	-3.48	-0.52	3/3	-2.43	all lower
F11	-3.59	-3.11	-0.18	3/3	-2.29	all lower
F12	-7.41	-9.37	-2.06	3/3	-6.28	all lower
F13	-1.40	-2.37	-0.82	3/3	-1.53	all lower
F14	-2.39	-5.17	-1.60	3/3	-3.05	all lower
F18	-3.28	-1.65	-0.34	3/3	-1.75	all lower
F19	1.17	0.80	-0.20	1/3	0.59	mixed
F20	ref<0	ref<0	-1.84	1/1	-1.84	one valid
F22	-0.93	-1.12	0.34	2/3	-0.57	mixed
F23	1.02	0.02	-0.00	1/3	0.35	mixed
F24	-0.15	-0.59	-0.03	3/3	-0.26	all lower

Counts use the signs of the reported ratios, including the negative value rounded to -0.00; undefined ratios are excluded. The underlying run-level ratios are unavailable.

Negative strip entries denote lower reported JANUS values, without establishing matched-protocol superiority over ABOM. The following mean$\pm$std table gives the host comparisons at $d = 30$ and $d = 100$. Full JANUS has lower means than CMA-ES on 11/16 and 15/16 functions, respectively; the GN-only column shows that adding CMIX changes individual outcomes even when win counts coincide.

Table 11: **BBOB $d = 30$ and $d = 100$:** mean±std; lower is better. 30D JANUS uses tuned configuration (fuse_every=50, 10 seeds); JANUS-GN 30D and all 100D from the host-controlled run (30 seeds). Bold blue=best, green=second-best. **W/T/L**: wins/ties/losses vs CMA-ES per function (mean comparison).

	$d = 30$			$d = 100$		
Func	**JANUS-GN**	**JANUS**	**CMA-ES**	**JANUS-GN**	**JANUS**	**CMA-ES**
f_4	**35.12 ±7.39**	70.01 ±14.79	66.59 ±13.86	**979 ±78.20**	1014 ±74.55	2685 ±689
f_6	2.44e-07 ±3.8e-08	**2.99e-08 ±3.7e-08**	5.11e-08 ±7.4e-08	**3140 ±659**	3416 ±1031	1.80e+04 ±2845
f_7	**0.276 ±0.53**	8.36 ±2.84	9.82 ±8.33	**229 ±84.01**	301 ±95.92	5101 ±1841
f_8	**3.15 ±2.26**	13.13 ±1.60	16.75 ±12.93	2101 ±3846	**1922 ±3465**	3075 ±4851
f_9	**1.71 ±1.31**	11.50 ±1.70	11.69 ±1.27	792 ±1751	**670 ±1817**	9.69e+04 ±2.8e+05
f_{10}	2.62e+04 ±8.9e+04	134 ±107	**122 ±123**	**2.06e+07 ±4.3e+06**	3.28e+07 ±7.0e+06	1.08e+08 ±2.3e+07
f_{11}	**9.18e-06 ±1.1e-05**	38.79 ±52.69	60.02 ±87.47	**2.09e+05 ±2.2e+04**	2.41e+05 ±2.8e+04	1.04e+06 ±8.6e+05
f_{12}	**0.025 ±0.044**	0.543 ±1.08	0.0273 ±0.075	**1.35e+05 ±1.2e+05**	4.35e+05 ±8.1e+05	4.59e+08 ±3.0e+08
f_{13}	**7.82e-03 ±2.6e-03**	0.602 ±0.85	0.498 ±0.48	**6.66 ±2.41**	7.09 ±2.23	105 ±29.86
f_{14}	2.35e-06 ±4.7e-07	**4.70e-07 ±2.5e-07**	4.75e-07 ±2.0e-07	**150 ±39.76**	165 ±42.95	1376 ±443
f_{18}	1.33 ±2.36	**0.272 ±0.35**	0.391 ±0.24	**87.35 ±32.66**	91.36 ±28.72	213 ±45.14
f_{19}	**0.559 ±0.85**	3.09 ±1.74	3.41 ±2.01	**7.86 ±0.36**	7.93 ±0.27	19.63 ±21.75
f_{20}	**-37.97 ±4.61**	1.86 ±0.2	1.87 ±0.19	-32.33 ±2.58	**-33.10 ±2.37**	-2.83 ±1.20
f_{22}	86.57	86.57	86.57	86.57	86.57	86.57
f_{23}	**2.93 ±0.4**	3.13 ±0.33	3.23 ±0.5	**4.84 ±0.34**	4.88 ±0.34	7.30 ±0.57
f_{24}	**41.77 ±8.40**	119 ±79.99	135 ±65.53	888 ±77.19	**882 ±85.20**	1055 ±27.55
W/T/L vs CMA-ES	11/1/4	11/1/4	–	15/1/0	15/1/0	–

The host tables and ABOM strip use different comparison sets. In the 500D host table, JANUS improves the mean on f_{22} and is slightly worse on f_{23}. Figures 2 and 3 use the same tabulated means and common-coverage rules.

The tuned low-dimensional configuration and the GN-only variant both achieve 11 mean wins at 30D, with different per-function values. ABOM (Wang et al., 2026) has public source code, but the strip retains reported values from its wider search domain. These cross-protocol ratios are descriptive references; they do not isolate the cause of a gain or loss.

The GN-only and full columns preserve the observed exploitation/exploration tradeoff. Neither the tabulated means nor the separate controls select one variant as best for every scalar objective.

Full d=500 comparison. Table 12 expands the main paper's four-baseline slice to include GLHF, LES, GLEET, SynCMA, and DiBO, alongside the CMA-ES host and Random Search. The "NN-BBO/MetaBBO" heading groups comparison methods, not a common learning protocol: they differ in model class, training requirements, and online adaptation. The reported search box, evaluation budget, and run counts define the numerical comparison. The methods are RS, GLHF (Li et al., 2024), SYMBOL (Chen et al., 2024), LES (Lange et al., 2023), RLEPSO (Yin et al., 2021), GLEET (Ma et al., 2024), LDE (Sun et al., 2021), SynCMA (Zhang et al., 2024), and DiBO (Yun et al., 2025), together with the CMA-ES host. The B2Opt implementation used in that comparison fixes its policy input at 10D and is omitted at 500D.

Table 12: **BBOB $d = 500$, full baseline comparison:** mean final error on 16 functions ($[-5, 5]^d$, 20,000 FEs). External baselines and JANUS use 10 runs; JANUS-GN is a 30-run reference. Bold blue=best, green=second. Lower is better. B2Opt is unavailable at 500D. Wins count best means among the displayed columns; the f_{20} result counts only in this within-table comparison, since it has no external baseline. Full JANUS leads external baselines on 12/15 covered functions. No taskwise significance is inferred from these means.

	Host	NN-BBO / MetaBBO baselines									Ours	
Func	**CMA-ES**	**RS**	**GLHF**	**SYMBOL**	**LES**	**RLEPSO**	**GLEET**	**LDE**	**SynCMA**	**DiBO**	**JANUS-GN**	**JANUS**
f_4	4.2e4	3.7e5	4.7e5	1.2e5	2.4e5	2.5e4	3.1e4	1.6e4	3.4e4	2.1e5	1.6e4	**5992**
f_6	1.8e6	1.6e7	1.8e7	5.6e6	9.3e6	9.2e5	1.5e6	9.9e5	5.8e5	9.8e6	7103	**2113**
f_7	1.5e4	3.9e4	4.7e4	1.8e4	2.3e4	1.2e4	1.2e4	4824	1.5e4	3.5e4	–	**988**
f_8	5.9e8	5.9e8	7.1e8	4.7e7	5.1e7	1.5e7	3.1e7	1.3e7	9.5e7	5.1e8	3072	**1426**
f_9	6.1e8	4.1e8	4.7e8	2.4e6	4543	9.5e5	4.4e6	1.8e5	7.0e6	4.6e8	**508.7**	1323
f_{10}	3.3e7	2.8e8	3.7e8	8.9e7	2.1e8	1.8e7	2.5e7	1.1e7	4.1e7	2.2e8	**1799**	1.5e6
f_{11}	1.2e4	5833	9115	3099	5136	2715	2746	3141	3437	6575	**1934**	4037
f_{12}	8.2e9	3.0e10	4.8e10	4.8e9	1.1e10	1.9e9	2.5e9	1.5e9	5.5e9	1.9e10	1.8e6	**1.0e4**
f_{13}	1.1e4	1.5e4	1.5e4	8913	1.3e4	5772	6709	5347	9840	1.3e4	4728	**88.6**
f_{14}	267	655	838	176	1458	82.5	112	73.0	252	633	1.5e5	**0.12**
f_{18}	73.9	140	171	70.3	371	54.7	57.9	40.6	69.5	115	157.6	**14.5**
f_{19}	3006	2096	2488	24.2	2500	22.4	39.9	12.3	45.3	2392	**0.34**	11.7
f_{20}	–	–	–	–	–	–	–	–	–	–	**16.9**	17.3
f_{22}	84.1	86.4	86.5	81.2	1184	63.6	69.1	62.2	75.5	86.0	11.6	**2.87**
f_{23}	1.65	1.64	1.85	**1.46**	1202	1.65	1.50	1.64	1.64	1.65	11.1	1.66
f_{24}	2.3e4	2.1e4	2.2e4	9315	8878	**7426**	8514	7660	9434	2.1e4	7448	7435
Wins	0	0	0	1	0	1	0	0	0	0	5	**9**

d=1000 extension. The separately reported five-function extension uses $[-5, 5]^d$, 20,000 FEs, and 30 runs, with geometric-mean final errors 1.19×10^9 for CMA-ES and 1.27×10^6 for JANUS-GN (about $936\times$). The available aggregate does not identify the five-function list or provide per-function results. It therefore remains a separate summary and does not enter the 16-function counts or plots.

C.2 Additional Baseline Comparisons

Table 13 (referenced from the main paper, Sec. 5.1 and Sec. 5.2) adds three checks: classical local modeling on BBOB, JANUS-DE as a non-CMA host check, and BO-style surrogate modeling on 1135-dimensional UAV.

Restricted-covariance baselines (LM-CMA, VkD-CMA). Table 14 empirically compares two restricted-covariance hosts—LM-CMA (LPCMA) (Loshchilov, 2014) and VkD-CMA (VLPCMA) (Akimoto & Hansen, 2016)—against a full-rank CMA-ES host on the 16 held-out BBOB functions at $d \in \{30, 100\}$, 10 seeds, 5,000 FEs, all under this single shared protocol. It reports geometric-mean final *error* (value minus the known optimum) at 5,000 FEs over 10 seeds, and is therefore separate from the DFO panel of Table 13, which reports geometric-mean final *value* at 20,000 FEs over 30 seeds. These practical implementations need not have the purely rank-k covariance assumed in Prop. 4; the proposition therefore does not predict their performance. Empirically, LPCMA achieves geometric-mean 67.4 (d=30) and 4.98×10^3 (d=100) and VLPCMA achieves 2.33×10^3 and 1.96×10^4; both are worse than the full-rank CMA-ES host (14.8 and 332) at every dimension, and the gap widens from $4.6\times$ at 30D to $15.0\times$ at 100D for LPCMA. The observed ordering and its change with dimension are empirical results for these implementations and budgets, not consequences of Prop. 4.

Table 13: **Extended baseline checks (Sec. 5.1).** Lower is better except win rate. Bold blue=best, green=second.

(a) BBOB DFO comparison, geometric-mean final value.		
Method	d=30	d=100
JANUS-GN	**0.1**	**14.7**
CMA-ES host	0.8	20.1
BOBYQA	1351	2112
COBYLA	1447	2524
DFO-LS	1864	2641
NEWUOA	2419	5477

(b) JANUS-DE vs. vanilla DE, host-agnostic check.			
Metric	d=30	d=100	d=500
JANUS-DE win rate	86%	91%	96%
JANUS-DE geo ratio	0.86	0.58	0.59

(c) UAV BO-like subset, d=1135, 2,500 FEs.	
Method	**Mean cost**
JANUS (28-terrain reference)	9,600
ET-BO (3-terrain subset)	**13,923**
RF-SMAC (3-terrain subset)	14,022
TuRBO (3-terrain subset)	14,634
Random (3-terrain subset)	17,701

Panel (a): reported final-value aggregates; the exact aggregation set and treatment of nonpositive values are unavailable, so no error-ratio inference is made. *Panel (c):* the JANUS reference and BO-like subset cover different terrains; best/second colors compare only the four subset methods.

Table 14: **BBOB restricted- vs. full-rank host comparison:** 16 held-out functions, 10 seeds, 5,000 FEs, $[-5, 5]^d$. Geometric-mean final error across 16 functions, and per-function wins (count of functions on which each method attains the lowest error). Both LPCMA (LM-CMA) and VLPCMA (VkD-CMA) underperform CMA-ES under these budgets. This is an empirical comparison; Prop. 4 applies only to purely rank-k covariances and does not imply this ordering. The full-rank CMA-ES host is $4.6\times$ better than LM-CMA at 30D and $15.0\times$ better at 100D. Bold blue=best, green=second-best.

	d=30		d=100	
Method	**Geo-Mean**	**Wins**	**Geo-Mean**	**Wins**
CMA-ES (host, full-rank)	**14.8**	**16**	**332**	**16**
LM-CMA (LPCMA)	67.4	0	4.98e3	0
VkD-CMA (VLPCMA)	2.33e3	0	1.96e4	0
Random Search	3.63e3	0	1.90e4	0

Readout. The DFO panel reports smaller aggregate final values for host-assisted local proposals under its stated budgets. It does not establish that local interpolation or surrogate-assisted search is generally ineffective. The DE panel provides a separate non-CMA host comparison. In the UAV panel, ET-BO, RF-SMAC-like, TuRBO-like, and Random share a three-terrain subset; JANUS's 9,600 is the 28-terrain aggregate from Table 16. The latter is retained as context and is excluded from the subset's best/second annotations.

C.3 UAV Per-Terrain Results

The UAV summaries combine 28 terrains and 30 runs per terrain at $d = 1135$. JANUS has the lowest aggregate mean and median at 2,500 FEs, while the terrain table includes settings led by GN-only or external methods.

Table 15: **UAV budget sweep over 28 terrains and 30 runs.** Lower is better.

Budget	**JANUS**	**CMA-ES**	**Gain**
2,500	**9,600**	14,486	−33.7%
10,000	**8,488**	8,587	−1.2%
20,000	**8,019**	8,023	−0.05%

Table 16: **UAV path planning:** 28 held-out terrains, 30 runs, 2,500 FEs (840 values/method). Blue/green mark best/second values in the displayed table; lower is better except rank-1 counts. Ratio is mean/JANUS. The final column retains reported unpaired Mann–Whitney counts for JANUS against the row method: wins/no detected difference/losses, using the Bonferroni threshold $p < 0.05/28$.

#	Method	Family	Mean ±Std	Median	IQR	Ratio vs JANUS	Terrains rank 1	JANUS vs row method
1	**JANUS**	Ours	**9,600 ±5,966**	**6,644**	9,579	**1.00**	**15**	–
2	DE	Traditional	11,006 ±3,865	9,261	7,412	1.15	1	24/3/1
3	JDE21	Adaptive	11,017 ±5,604	8,579	**2,795**	1.15	1	–
4	**JANUS-GN**	Ablation	11,317 ±5,373	9,508	9,815	1.18	8	0/28/0
5	RLDEAFL	MetaBBO	11,411 ±5,818	8,197	9,227	1.19	0	–
6	GLEET	MetaBBO	11,603 ±7,085	7,455	9,808	1.21	1	–
7	PSO	Traditional	11,792 ±4,454	9,156	8,223	1.23	0	–
8	RS	Traditional	13,891 ±6,859	10,394	9,169	1.45	0	23/5/0
9	CMA-ES	Adaptive	14,486 ±6,559	10,908	9,956	1.51	0	24/3/1
10	SAHLPSO	Adaptive	14,865 ±6,885	11,265	9,716	1.55	0	–
11	GLHF	MetaBBO	20,354 ±5,918	20,921	7,137	2.12	0	–
12	LES	MetaBBO	26,968 ±14,006	26,220	20,365	2.81	2	–

The mean advantage over CMA-ES is concentrated at the short budget. At 10,000 FEs the means are 8,488 and 8,587; at 20,000 FEs they are 8,019 and 8,023. These values approach parity rather than establishing a convergence advantage.

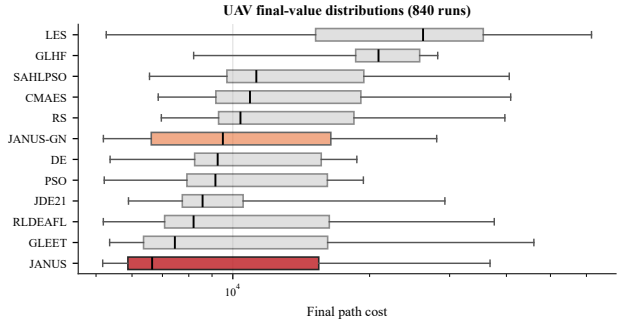
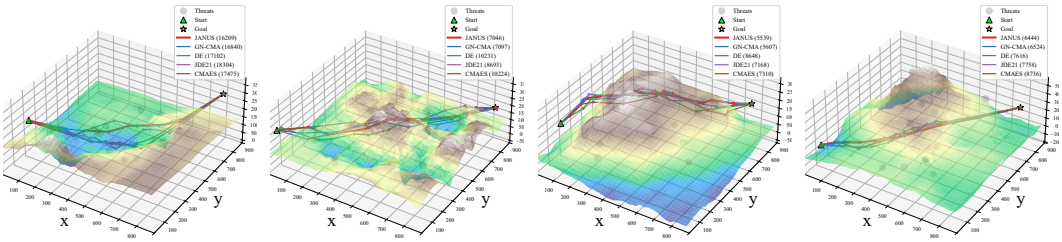


Figure 5: UAV final-value distributions over 840 runs.

The distribution contains 840 final values per method. The lower JANUS mean and median do not imply the best result on every terrain; Table 17 retains the individual comparisons.

3D path visualizations. The main paper includes the representative T1/T17 convergence and path panels. Figure 6 therefore keeps only additional terrain-level path examples: T9/T41 are hard cases where non-JANUS baselines can find safer early paths, while T25/T55 are representative JANUS wins. The examples complement the aggregate distribution and per-terrain table.



(a) T9: hard (b) T25: JANUS win (c) T41: hard (d) T55: JANUS win

Figure 6: **Additional UAV 3D path visualizations** on four terrains (T9/T25/T41/T55). JANUS often finds smooth, low-cost trajectories, but T9/T41 show that this behavior is not universal under the short 2,500-FE budget.

Table 17 retains all terrains and the original test annotations. In the reported comparison, T29 is a loss against CMA-ES and T9/T11/T19 have no statistically detected difference; these are not equivalence claims.

Table 17: **UAV terrain audit.** Every held-out terrain is kept; gain compares JANUS with the CMA-ES host under the stated 2,500-FE protocol. Colors retain the reported Bonferroni-corrected Mann–Whitney U results across 28 terrains, consistent with Table 16: green=JANUS win, amber=no statistically detected difference, red=JANUS loss.

Terrain	JANUS	CMA-ES	Gain	Terrain	JANUS	CMA-ES	Gain
T1	7035	10062	+30.1%	T29	15998	10607	-50.8%
T3	9716	17753	+45.3%	T31	6044	18533	+67.4%
T5	5974	19695	+69.7%	T33	7696	10066	+23.5%
T7	5834	18681	+68.8%	T35	5304	7547	+29.7%
T9	23155	19402	-19.3%	T37	10520	18362	+42.7%
T11	21077	18970	-11.1%	T39	10395	17266	+39.8%
T13	7527	10261	+26.6%	T41	6621	9257	+28.5%
T15	6528	8369	+22.0%	T43	6119	16738	+63.4%
T17	6584	20170	+67.4%	T45	5975	8912	+32.9%
T19	22186	19407	-14.3%	T47	8285	9291	+10.8%
T21	7515	10439	+28.0%	T49	5690	8906	+36.1%
T23	5807	8322	+30.2%	T51	9329	11420	+18.3%
T25	11623	18375	+36.7%	T53	7572	10028	+24.5%
T27	15966	27321	+41.6%	T55	6728	21444	+68.6%

The original comparison reports Bonferroni-corrected unpaired Mann–Whitney tests over 28 terrains, yielding 24 improvements, 3 no-difference outcomes (T9/T11/T19), and 1 loss (T29) against CMA-ES. This matches the reported 24/3/1 summary. On those four non-win terrains, GN-only ranks first among the 12 displayed methods. T29 has reported means of 6,063 for GN-only, 15,998 for full JANUS, and 10,607 for CMA-ES. These values demonstrate variation in the component ordering; they do not directly measure Jacobian accuracy or establish why the original full variant deteriorated.

C.4 MOO Supplementary Results

The grouped MOO results show that JANUS's effect depends on objective count and host. The objective vector supplies the local metric, while the named host performs Pareto survival.

Table 19: **Component ablation across representative domains.** Ratios are relative to the base host; lower is better for BBOB/UAV and higher is better for MOO.

Variant	GN	CMIX	BBOB ↓	UAV ↓	MOO ↑
Base host			1.00×	1.00×	1.00×
GN only	✓		0.85×	0.91×	**1.02**×
CMIX only		✓	0.92×	**0.88**×	0.97×
JANUS	✓	✓	**0.83**×	0.89×	1.01×

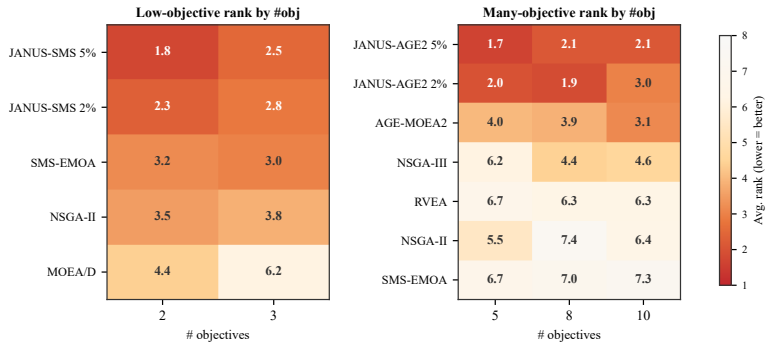


Figure 7: **MOO ranks by objective count.** JANUS-SMS leads low-objective tasks; JANUS-AGE2 leads many-objective tasks.

The reported grouped ranks favor JANUS-SMS at two objectives and JANUS-AGE2 at five and eight objectives; the difference is smaller at ten objectives (Fig. 7, Table 18). These outcomes locate the gains but do not determine whether they arise from model quality, survival, or their interaction.

Table 18: **MOO performance by objective count.** Three leading methods by reported average rank are shown for each group. Blue/green mark best/second distinct displayed values separately for each metric; ties receive the same mark. Lower rank and higher wins/HV are better.

Group	Method	Tasks	Avg. rank	Wins	Norm. HV
Low-objective tasks					
2	**JANUS-SMS 5%**	12	**1.75**	**8**	**1.000**
	JANUS-SMS 2%	12	2.33	1	0.996
	SMS-EMOA	12	3.17	0	0.974
3	**JANUS-SMS 5%**	6	**2.50**	2	0.982
	JANUS-SMS 2%	6	2.83	**3**	**0.986**
	SMS-EMOA	6	3.00	0	0.967
Many-objective tasks					
5	**JANUS-AGE2 5%**	6	**1.67**	**2**	**0.997**
	JANUS-AGE2 2%	6	2.00	**2**	0.983
	AGE-MOEA2	6	4.00	1	0.954
8	**JANUS-AGE2 2%**	7	**1.86**	**3**	**0.975**
	JANUS-AGE2 5%	7	2.14	2	0.959
	AGE-MOEA2	7	3.86	0	0.928
10	**JANUS-AGE2 5%**	7	**2.14**	**2**	**0.965**
	JANUS-AGE2 2%	7	3.00	1	0.963
	AGE-MOEA2	7	3.14	**2**	0.869

Archived component summary. Table 19 retains the complete original component summary; its BBOB/UAV columns also appear in the main paper. The MOO GN-only and combined labels do not specify their scalarization, so these cells do not test the exploration-only MOO interface in Sec. 3. They are retained as historical results rather than evidence for a defined MOO descent proposal. The following figure likewise preserves the original ratio, taskwise-test, and runtime summaries with the scope stated in App. A.1.

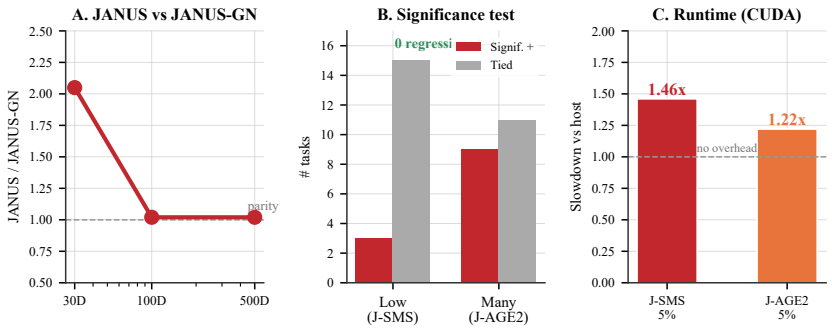


Figure 8: **Ablation, taskwise tests, and cost.** (A) JANUS/JANUS-GN ratio. (B) Reported unadjusted tests: 0 detected regressions, without an equivalence claim. (C) CUDA runtime: $1.46\times$/$1.22\times$.

CMIX fraction and runtime. Table 20 sweeps the fraction of offspring allocated to CMIX infill from 0.05 to 0.60. The sweep shows that the best fixed fraction is often larger than the conservative default, but the curve is not monotone on every problem. We therefore use a conservative fraction in the main experiments rather than tuning it per benchmark. Table 21 reports wall-clock overhead: with CUDA-accelerated metric estimation, JANUS-SMS 5% takes $1.46\times$ the runtime of SMS-EMOA and JANUS-AGE2 5% takes $1.22\times$ the runtime of AGE-MOEA2. This overhead is the price of online Jacobian estimation; it is acceptable when evaluations dominate runtime, but may matter for benchmarks with sub-millisecond evaluation cost.

Table 20: **CMIX candidate fraction ablation (mean HV; higher is better).** Bold blue=best per row, green=second-best.

Problem	0.05	0.10	0.20	0.30	0.40	0.50	0.60
DTLZ4_d12_n3	72.7	72.9	72.9	72.8	72.9	**72.9**	72.9
WFG1_d12_n2	8.10	9.23	9.64	9.50	9.63	9.51	**9.86**
WFG2_d12_n3	117	119	121	124	**131**	127	128
WFG4_d12_n2	7.02	7.51	7.47	7.99	7.71	8.07	**8.74**
ZDT1_d30_n2	6.70	6.75	6.81	6.94	7.09	7.07	**7.10**
ZDT2_d30_n2	5.85	5.90	5.88	5.91	6.12	**6.29**	6.27
ALL	36.2	36.9	37.3	37.8	**39.1**	38.4	38.9

Table 21: **Runtime overhead with CUDA-accelerated metric estimation.** Slowdown is the ratio of mean wall-clock time per independent run for the JANUS variant to its host baseline (lower is better; $1.00\times$ = no overhead). Values match Fig. 8C.

JANUS variant	Host baseline	Tasks	Runs	Slowdown
JANUS-SMS 5%	SMS-EMOA	18	180	$1.46\times$
JANUS-AGE2 5%	AGE-MOEA2	20	200	$1.22\times$

Per-task winner summary. The reported 5% variants have 16/38 wins: JANUS-SMS 5% has 10/18 low-objective wins and JANUS-AGE2 5% has 6/20 many-objective wins. This is not the union of wins across all 2%/5% variants. It also differs from the original 12/38 unadjusted Mann–Whitney count. The original per-run MOO records are unavailable; neither summary supplies a new multiplicity-adjusted inference. Grouped ranks and normalized HV show where the reported improvements occur. The fraction sweep and grouped ranks show variation across settings; they do not define a universal rule for increasing the exploration fraction.

C.5 AWB Deployment (d=1135)

To complement the synthetic benchmarks, we report a single deployment on a proprietary image signal processing (ISP) calibration task. The objective is the mean reproduction angle error (MRAE) of an auto white balance (AWB) module across m=111 test scenes; the search space is d=1135 ISP parameters. Each evaluation runs the full ISP pipeline on all scenes, making repeated large-scale experimentation infeasible. We report indexed checkpoints through 5,000 from one longer run per

method (seed 2024). The baseline was configured for 10,000 evaluations and the GN/CMIX runs for 20,000; only the common prefix is compared.

Table 5 uses the baseline-10k, mosaic-GN-20k, and mosaic-CMIX-20k logs. Trials are indexed from zero: checkpoint B includes indices $0, \ldots, B-1$. Every indexed prefix is complete. The baseline log contains conflicting repeated IDs between 1,000 and 1,189 after resumption; its last records agree with the persisted evaluation outputs. All five checkpoint minima are identical whether first or last occurrences are retained. Thus the table reports indexed checkpoints, not a verified count of distinct physical calls. The GN/CMIX prefixes have no conflicting repeated IDs. At checkpoint 5,000 the unrounded MRAEs are 1.055987, 0.837379, and 0.818649, respectively; $(1 - 0.818649/1.055987) \times 100 = 22.48\%$. The baseline 1,000-trial entry is 1.525794 (1.526 rounded), replacing an entry from a different run. GN/CMIX use a 300-point window, ridge and damping 1, warm-up 40, refresh every four generations, and $\kappa = 1$. The CMIX deployment directly changes C. Its available configuration sets `cmix_w=auto` and a 300-point window; the sampler default is $w_{\text{base}} = 0.5$, with bounds $[0.01, 0.20]$. The implementation resolves the weight using the configured window cap, giving $0.5 \min(111, 300)/1135 \approx 0.049$, inside those bounds. Here 111 is the AWB scene count; no corresponding UAV residual count is established by these records.

Readout. Both JANUS variants reduce MRAE by $\sim$21–22% relative to CMA-ES at checkpoint 5,000. The covariance-reshaping variant trails GN at checkpoint 1,000 and overtakes it by 3,000, reaching 0.819 versus 0.837 at 5,000. A single trajectory does not establish the cause of this ordering or predict its behavior beyond the reported prefix. Because this is a single run on a proprietary pipeline, we treat it as deployment evidence rather than a statistically controlled comparison.

D Controlled Scalar-Reference Diagnostics

The anonymous code supplement includes the reference scripts, protocol, per-evaluation JSON records, and a results verifier in `reference_diagnostics/`. The scalar-reference study comprises 288 optimization trials: IOH 0.3.22 BBOB functions f_6, f_{22}, f_{23}, dimensions 100 and 500, instance 1, eight optimizer seeds (101–108), and six variants. Each run uses 3,000 actual objective evaluations in $[-5, 5]^d$. The host is pyCMA with initial mean sampled uniformly in $[-4, 4]^d$, initial step size 2, its default population size, and active covariance updates disabled for every variant. All points are box-clipped before evaluation. Full batches are passed to the host update. If the remaining budget is smaller than the nominal population, the sampled batch is truncated, evaluated, and included in final-error and trajectory statistics, but is not passed to `tell`. These are short-budget controls on one instance per function, not the 20,000-FE benchmark protocol.

Local fit and proposals. The reference fits $c + a^\top z + \frac{1}{2}\sum_j h_j z_j^2$ on the most recent 300 points. Inputs are standardized; targets are centered and divided by the window standard deviation. Ridge is 1 with an unpenalized intercept. The scalar gradient uses the linear and quadratic terms, while $G = D^{-1}\,\mathrm{diag}(\max(h, 0))D^{-1}$. Both damping values equal 1. After 80 native evaluations, every fourth generation replaces one slot by a scalar exploitation point and $\mathrm{round}(0.2L)$ other slots by pure-metric CMIX, bounded by the remaining population size. Here L denotes the nominal host population size (called P in Algorithm 1); the remaining slots cap the quota in a truncated final batch. At least two retained slots are required for a scheduled fit/injection event. Intervening generations use native candidates. The exploitation scale is chosen from $\{0.25, 0.5, 1\}$ by fitted scalar value, and injected points are clipped to whitened radius $\sqrt{d}$ before box handling. Distinct random streams separate host sampling, proposals, bootstrap weights, and diagnostics.

Controls. Shared uses one deterministic fit for both roles. Random first constructs the same reference proposals on its own trace, then replaces their whitened directions with isotropic random directions at exactly the same radii. Rotated applies a common random signed permutation to the whitened proposal vectors: an orthogonal control preserving radii and covariance spectrum in that coordinate system. Box clipping may subsequently change their lengths; boundary effects are recorded. Thus these controls test direction information conditional on the model-derived radii, and still incur the fit cost. Shared-B uses one exponentially reweighted fit for both roles; separate-B draws independent exponential weights on the same window for the two fits. They have matched marginal fitting procedures but different estimation noise, so this comparison cannot isolate a purely

computational effect of sharing. With identical deterministic data and fitting, separate fits would be numerically identical and merely duplicate computation.

Diagnostics and inference. At each refresh, a separate 80/20 random split fits only the training points and predicts the held-out values; its normalized MSE divides by the training-mean predictor's MSE. These diagnostic fits do not affect proposals or consume extra objective evaluations. Candidate success means improving the incumbent before that generation, and candidate counts are retained by origin. Effective ranks and regularized condition numbers are recorded. Final-error comparisons pair optimizer seeds and use all 2^8 sign flips of paired log-error differences. Holm adjustment covers all 24 tests: shared versus host/random/rotated and shared-B versus separate-B at six function/dimension settings. The accompanying 95% intervals use 20,000 paired seed-bootstrap replicates and are descriptive, without simultaneous coverage.

Table 22: **Separate scalar-reference controls.** Mean final error ± standard deviation, eight optimizer seeds, IOH instance 1, 3,000 FEs. Shared uses one diagonal quadratic fit. Random matches the proposed whitened radii; rotated applies an orthogonal direction control. Shared-B and separate-B use matched bootstrap reweighting. These are new diagnostic runs, not reproductions of the 20,000-FE tables.

d	Func.	Host	Shared	Random	Rotated	Shared-B	Separate-B
100	f_6	686.60 ±164.36	553.38 ±103.55	692.53 ±143.87	721.57 ±119.10	540.16 ±103.23	604.54 ±82.05
100	f_{22}	17.32 ±9.43	17.07 ±9.33	17.32 ±9.40	17.28 ±9.40	17.07 ±9.33	17.07 ±9.33
100	f_{23}	5.45 ±0.37	5.44 ±0.41	5.28 ±0.38	5.45 ±0.27	5.49 ±0.24	5.34 ±0.25
500	f_6	9.74e+05 ±1.44e+05	7.00e+05 ±1.30e+05	8.15e+05 ±1.48e+05	8.65e+05 ±1.39e+05	8.41e+05 ±1.84e+05	7.70e+05 ±1.51e+05
500	f_{22}	53.83 ±5.99	52.55 ±5.17	52.08 ±6.50	51.71 ±5.69	52.68 ±4.34	51.66 ±5.41
500	f_{23}	1.73 ±0.03	1.73 ±0.02	1.72 ±0.02	1.73 ±0.04	1.73 ±0.05	1.71 ±0.03

Table 23: **Model diagnostics for the shared scalar reference.** Median held-out NMSE relative to a training-mean predictor, metric rank and stable rank, and candidate success percentages within the same refresh generations. Success means improving the pre-generation incumbent. NMSE below one favors the fitted model.

d	Func.	NMSE	Rank(G)	Stable rank	Native (%)	GN (%)	CMIX (%)
100	f_6	1.57	69.00	19.51	3.45	52.91	3.78
100	f_{22}	0.03	100.00	59.39	2.68	91.86	4.36
100	f_{23}	5.28	59.00	16.70	0.11	0.00	0.10
500	f_6	0.49	376.50	92.35	3.70	100.00	3.77
500	f_{22}	0.31	400.00	107.15	4.50	99.26	6.16
500	f_{23}	2.29	263.50	52.88	0.16	1.10	0.09

Table 24: **Target attainment and measured cost.** ECDF percentages aggregate three functions, eight seeds, and eight relative error targets per run (192 targets per dimension). CPU fit/solve timings below are medians of 21 isolated repetitions with a 300-point window; they exclude objective evaluation and host updates.

d	Variant	500 FEs	1,000 FEs	2,000 FEs	3,000 FEs
100	Host	25.0	38.5	49.0	49.0
100	Shared	25.0	40.1	49.0	49.0
100	Random	24.0	37.5	49.0	49.0
100	Rotated	24.5	37.5	49.0	49.0
500	Host	16.7	16.7	20.8	26.0
500	Shared	16.7	17.2	22.4	28.6
500	Random	16.7	16.7	21.9	28.1
500	Rotated	16.7	16.7	21.4	27.6

Dimension	Fit (ms)	Solve/sample (ms)
30	0.24	0.017
100	1.02	0.029
500	6.45	0.069
1000	12.38	0.120

Readout and uncertainty. On f_6, the paired geometric-mean error ratios for shared versus host are 0.814 (100D) and 0.716 (500D). Their descriptive 95% bootstrap intervals are $[0.669, 0.989]$ and $[0.600, 0.868]$; the Holm-adjusted p-values are 1.000 and 0.445. None of the 24 contrasts is significant at adjusted $p < 0.05$. Shared is not uniformly better than the direction controls, and the shared-B/separate-B ordering varies across settings. These results support a function-dependent interpretation rather than a general performance benefit from sharing.

The held-out NMSE is 5.28/2.29 on f_{23} at 100D/500D, worse than predicting the training mean, and the exploitation candidate rarely improves the incumbent there. Conversely, an imperfect global fit over the window can still produce useful local proposals, as on 100D f_6 (NMSE 1.57). Candidate success rates in Table 23 compare origins within the same refresh generations and remain descriptive: candidates within a run are dependent. A diagonal quadratic metric can have rank larger than the number of observations, so its algebraic rank is not evidence that every direction is identified by the data.

Direct Jacobian identifiability check. To separate estimation error from unknown benchmark derivatives, we also fit affine residual maps $s(x) = J_* x$ with $m = 111$, $n = 300$ independent standard-normal training inputs, and observation noise of standard deviation 0.01. Entries of J_* are independent $\mathcal{N}(0, 1/d)$. Ten seeds (301–310), ridge 1, and 100 fresh test inputs per seed give mean relative Frobenius Jacobian error 0.0094 at $d = 100$ and 0.6358 at $d = 500$; held-out NMSE is 8.81×10^{-5} and 0.4006. This controlled linear example shows the information loss when the window is smaller than the ambient dimension. It does not measure Jacobian error on UAV, AWB, or the scalar BBOB runs, and does not justify assuming that 300 points recover an arbitrary high-dimensional Jacobian.

Target attainment and computation. For Table 24, each run's target errors are the common native initial-population best error multiplied by $\{1, 0.5, 0.2, 0.1, 0.05, 0.02, 0.01, 0.001\}$. We count targets reached by the stated FE checkpoint, averaging over three functions and eight seeds at each dimension. This is a relative-target ECDF for this subset, not a standard full-suite COCO ERT analysis. At 3,000 FEs, host and shared both reach 49.0% of targets in 100D; in 500D they reach 26.0% and 28.6%. Kernel timings use one CPU thread and 21 repetitions per dimension. Per-run records separately retain host, fit, solve/sample, objective, and diagnostic costs. The diagonal reference avoids dense metric inversion; its timings cannot be assigned to a full-matrix JANUS implementation or to the original CUDA MOO runs.

E Mathematical Derivations and Proofs

Residual construction and coordinate mapping. For nonnegative components, $s_j = \sqrt{r_j}$ gives $f = m^{-1}\|s\|^2$. A local linear residual model supplies gradient $2J^\top s_b/m$ and normal-matrix approx-

imation $2J^\top J/m$. Absorbing the common factor into damping gives Eq. 2. Scalar fits instead supply their objective gradient; a PSD factorization alone does not reproduce the scalar objective.

For centered residual targets $S_c = S - \mathbf{1}\bar{s}^\top$ and design rows $z_i^\top = (x_i - \mu)^\top D^{-1}$, ridge regression gives

$$\widehat{W} = (Z^\top Z + \beta I)^{-1} Z^\top S_c, \qquad J = \widehat{W}^\top D^{-1}, \qquad \beta > 0.$$

This maps the standardized slopes to search coordinates. With an optional orthonormal basis U_r, fitting on ZU_r instead gives $J = \widehat{W}_r^\top U_r^\top D^{-1}$.

What the window can identify. Centering gives $\operatorname{rank}(Z) \le \min(n-1, d)$. Since the fitted coefficient matrix has rank at most both $\operatorname{rank}(Z)$ and m,

$$\operatorname{rank}(J) \le \min\{m, \operatorname{rank}(Z)\} \le \min\{m, n-1, d\}. \tag{5}$$

The residual dimension bounds J, not the input design matrix Z. For $n = 300, d = 1135$, at least 836 search directions lie in the fitted Jacobian's nullspace. With the AWB residual count $m = 111$, the lower bound is 1,024. These are properties of the fitted map, not statements that the true objective is flat in those directions. For any $v \in \ker J$,

$$(J^\top J + \lambda_{\text{mix}} I)^{-1} v = \lambda_{\text{mix}}^{-1} v.$$

Thus regularization determines the metric on unidentified directions. This argument concerns linear residual regression and does not bound the rank of an arbitrary diagonal quadratic scalar proxy.

Proof of Proposition 1. The fitted residual objective has $\nabla \hat{f}(0) = 2J^\top s_b$. Substituting the step gives

$$\nabla \hat{f}(0)^\top \Delta_{\text{GN}} = -2(J^\top s_b)^\top (J^\top J + \lambda I)^{-1} (J^\top s_b) \le 0.$$

The inverse is positive definite, so the inequality is strict for a nonzero fitted gradient. The scalar case similarly has $g^\top \Delta = -g^\top (G + \lambda I)^{-1} g < 0$ for $g \neq 0$. These are directional statements before clipping. For example, treating a scalar fit as $s = a^\top z$ with $a = 1, z_b = -1, \lambda = 1$ would give the residual step $+1/2$, which increases that scalar objective; the gradient-driven step is $-1/2$.

Proof of Proposition 2. The coordinate identity $x_e - \bar{x} = \sigma C^{1/2} y_e$ and the operator norm imply

$$\|x_e - \bar{x}\| \le \sigma \|C^{1/2}\|_2 \|y_e\| \le \kappa \sigma \lambda_{\max}(C^{1/2}) \sqrt{d}.$$

For $T_n = \operatorname{tr}(C) T / \operatorname{tr}(T)$,

$$\operatorname{tr}((1-w)C + wT_n) = (1-w)\operatorname{tr}(C) + w \operatorname{tr}(C) = \operatorname{tr}(C).$$

The first bound is before boundary repair and the trace identity is before clipping and boundary handling. Neither asserts model accuracy or improvement on the true objective.

Proposition 3 (Block-separable local least squares)**.** *If $J^\top J$ is block diagonal, the damped residual system decouples into independent continuous block subproblems.*

Proof. The inverse of $J^\top J + \lambda I$ is block diagonal, so each block of Δ depends only on its corresponding block of $J^\top s_b$. This does not identify the continuous solution with a historical fragment or a minimizer of a nonlinear component cost. Discrete historical assembly requires additional assumptions.

Proposition 4 (Auxiliary bound for purely rank-deficient covariances)**.** *Let $B \in \mathbb{R}^{d \times k}$ have orthonormal columns, $k < d$, and let $C' = B\widetilde{M}B^\top$. For any $J \in \mathbb{R}^{m \times d}$ and $\lambda > 0$, set $T = (J^\top J + \lambda I)^{-1}$ and $P_\perp = I - BB^\top$. Then*

$$\|C' - T\|_F \ge \|P_\perp T P_\perp\|_F \ge \frac{\sqrt{d-k}}{\sigma_{\max}(J)^2 + \lambda} > 0.$$

Proof. Write $a = (\sigma_{\max}(J)^2 + \lambda)^{-1}$. The spectral bound on $J^\top J$ gives

$$T \succeq aI. \tag{6}$$

Every admissible covariance satisfies

$$P_\perp C' P_\perp = 0. \tag{7}$$

Projection contracts the Frobenius norm, so

$$\|C' - T\|_F \ge \|P_\perp (C' - T) P_\perp\|_F = \|P_\perp T P_\perp\|_F. \tag{8}$$

On the $(d-k)$-dimensional range of $P_\perp$, the compressed matrix has all eigenvalues at least a. Hence

$$\|P_\perp T P_\perp\|_F \geq a\sqrt{d-k} = \frac{\sqrt{d-k}}{\sigma_{\max}(J)^2 + \lambda}. \tag{9}$$

Scope. Independence between B and J is unnecessary. With a full-rank isotropic or diagonal base, the zero-projection step fails; the proposition does not order the performance of practical LM-CMA, VkD-CMA, or CMA-ES. For $d = 2, k = 1, B = e_1, J = [0, 1], \lambda = 1$, both the gap for $C' = \mathrm{diag}(1, 0)$ and the lower bound equal $1/2$.

F Discussion and Operating Regimes

JANUS exposes estimated local geometry through a bounded proposal interface. Damping, clipping, and trace normalization regulate proposal scale; the host evaluates the resulting candidates and maintains alternative search directions. Low fitted sensitivity may reflect either a flat direction or limited observations, so these controls do not guarantee model accuracy or objective improvement.

The complete domain comparisons and scalar-reference controls in Apps. C and D show where the proposals help and where they do not. GN-only isolates the contribution of metric-shaped exploration, while the fit-sharing controls assess whether one model improves on independent fits. Together they support choosing and evaluating the proposal roles for each host, task, and budget; they do not establish a universal switching rule or a consistent performance advantage from sharing.