Match the Geometry Before the Surrogate: Extreme Low-Budget Optimization in High Dimensions

Michal Průšek[1,2] **Adam Novozámský**[2] **Filip Šroubek**[2]
[1]Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University in Prague
[2]Institute of Information Theory and Automation, Czech Academy of Sciences, Prague
{prusek,novozamsky,sroubekf}@utia.cas.cz

Abstract

Low-budget molecular optimization is often framed as Bayesian optimization in a learned latent space, but standard benchmarks typically allow hundreds or thousands of evaluations, whereas a realistic wet-lab round may allow only a few dozen. We revisit this regime using a SELFIES molecular variational autoencoder (VAE) with a Gaussian prior and GuacaMol objectives at $N = 30$ evaluations. The main finding is not a new surrogate model, but an interaction in candidate generation. The encoder means of a reference set of molecules concentrate near a narrow spherical shell, while many high-dimensional optimizers propose candidates from boxes or box-shaped trust regions. We study a simple constraint, the *on-sphere scaffold*, that restricts candidate generation to this shell. We then introduce two minimal optimizers to probe what remains useful once the support is fixed. GeoWalk performs geodesic local search on the shell without a surrogate. After 30 evaluations, it shows no detected difference from a Gaussian-process (GP) surrogate on the shell, which is ahead earlier in the budget. GeoCMA adds covariance learning in a random subspace. It does not improve on GeoWalk in the molecular benchmark. On LassoBench it has a lower mean loss than GeoWalk, but full-dimensional CMA-ES remains stronger on most tasks. A factorial experiment with a single GP crosses the candidate support (shell or box) with the acquisition maximizer (a finite candidate set or a continuous optimizer). In this Gaussian-prior VAE benchmark, shell candidates outperformed box candidates when acquisition was maximized over a candidate set, but not with continuous maximization by L-BFGS-B. These results come from a controlled study of one Gaussian-prior VAE benchmark and do not recommend VAEs for molecular optimization. The practical lesson is to control the candidate support and the acquisition maximizer before attributing a gain to the surrogate.

1 Introduction

Molecular and protein-design loops are often constrained by assays rather than by computation. A single wet-lab round may evaluate only a few dozen variants, each requiring substantial cost and weeks of turnaround time. This is far from the usual evaluation setting for high-dimensional Bayesian optimization (HD-BO) [1–3], where benchmark budgets often range from hundreds to thousands of oracle calls. In principle, Bayesian optimization is well suited to such expensive black-box objectives: it fits a cheaper surrogate model, typically a Gaussian process (GP) or another learned predictor, and uses it to choose the next candidate to evaluate. However, the budget mismatch raises a simple yet consequential question: *In the extreme low-budget scenario (*30 *evaluations), what should the optimizer actually do?*

We study this question in a widely used molecular-design pipeline built around a SELFIES [4] Transformer VAE, released with local latent space Bayesian optimization (LOL-BO) [2] and pre-

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

trained by its authors on the GuacaMol training set. The VAE has latent dimension $D = 256$, and we evaluate it on the GuacaMol goal-directed tasks [5]. This benchmark underlies several recent latent-space BO comparisons, and keeping its generator fixed lets us isolate the optimization layer. When the molecules from the ZINC-250k reference set [6, 7] are encoded by the VAE, their encoder means concentrate near a narrow spherical shell rather than filling the surrounding box. Many HD-BO methods, however, generate candidates inside axis-aligned boxes or box-shaped trust regions. A Bayesian optimizer picks its next point by maximizing an acquisition function, a score of how promising a point is under the surrogate. It can score a finite set of randomly drawn candidates and take the best one, or it can start from a point and climb the acquisition function with a gradient-based optimizer, which we call continuous maximization. Here that optimizer is L-BFGS-B on both supports. In this Gaussian-prior VAE benchmark, shell candidates outperformed box candidates when acquisition was maximized over a candidate set, but not with continuous maximization by L-BFGS-B. The GP is fitted in the same way on both supports, but box candidates are rescaled per coordinate to $[0, 1]^D$ before the GP, and shell candidates are not. With candidate-set maximization, the shell is ahead on 9 of the 11 tasks of our protocol.

The main object of study is therefore not a new surrogate model, but candidate generation: where candidates come from, and how an acquisition function is maximized over them. Instead of proposing candidates from an axis-aligned box, the *on-sphere scaffold* places them on the empirical spherical shell occupied by the encoder means of the reference set. Formally, candidates have the form $x = \bar{r}u$, where $\bar{r}$ is the mean encoder norm of the run's cold-start molecules and $u \in S^{D-1}$ is a unit direction in the D-dimensional latent space. This constrains only the norm of a candidate, not its direction. We introduce two deliberately simple optimizers, **GeoWalk** and **GeoCMA**, as probes of what remains useful once candidate generation is restricted to the encoder-mean shell. The first, GeoWalk, samples the shell uniformly and then takes geodesic steps from the best point found so far. Despite having no surrogate, no learned covariance, and no acquisition function, GeoWalk shows no detected difference from a GP on the shell after thirty evaluations, but the GP is ahead earlier in the budget. The second optimizer, GeoCMA, is positioned not as a competitor to GeoWalk on the molecular VAE benchmark but as a complementary regime probe. It keeps the same on-sphere scaffold but replaces both phases of GeoWalk with covariance learning in a random low-dimensional subspace, started from the best cold-start point, to mark where adding such structure becomes useful. On the molecular VAE benchmark itself GeoCMA does not improve over GeoWalk. On two benchmarks outside chemistry, LassoBench [8] and NAS-Bench-101 [9], it has better means than GeoWalk. Full-dimensional CMA-ES, however, has the lower loss on most LassoBench tasks, and on NAS-Bench-101 GeoCMA is not detectably ahead of random search.

The paper makes three contributions. First, a factorial experiment shows that, in this benchmark, the advantage of shell candidates over box candidates depends on how the acquisition is maximized. Second, we introduce GeoWalk and GeoCMA as minimal diagnostic optimizers that show what remains of this advantage without a GP. Third, all baselines are compared on one matched set, with every known deviation from the published methods reported (Appendix H and the method cards in the repository), because comparisons between latent optimizers can be confounded by candidate generation.

Practical lesson. When a learned latent space is optimized with few evaluations, first check where the candidates lie, for example by comparing their norms with those of encoded data, and how the acquisition is maximized. Otherwise, a gain attributed to the surrogate may come from candidate generation instead. A search without a surrogate on the same support, such as GeoWalk in this paper, shows how much the surrogate adds.

2 Related Work

Latent-space molecular optimization often follows a VAE-based pipeline: a variational autoencoder maps discrete molecules to continuous latent vectors, the latent space is searched with a Gaussian-process model of the objective, and selected latent points are decoded into molecules for scoring [7]. Later methods refined different parts of this pipeline. LOL-BO [2] introduced local trust regions in latent space. CoBO [10] regularized the VAE so that nearby latents have similar objective values. InvBO [3] addressed the encoder–decoder misalignment by inverting the decoder so that latent codes reconstruct their inputs consistently. NF-BO [11] replaced the VAE altogether with an autoregressive normalizing flow that admits an exact left-inverse and removes the reconstruction gap.

EULBO [12] jointly optimizes the ELBO of its variational GP surrogate and a utility-based acquisition objective so that the surrogate is fitted with optimization in mind. COWBOYS [13] decouples the GP surrogate from the generative model entirely, fitting them separately and combining them via a Bayesian update. The authors of COWBOYS also observed that molecular VAE latents occupy a thin shell, and argued that box-shaped search spaces cannot target it. These methods form the natural background for our study, but most of them are evaluated with budgets far larger than those available in a wet-lab round. We focus on the opposite regime: *What remains reliable when the entire optimization loop consists of only a few dozen decoded and scored candidates?*

High-dimensional BO provides several mechanisms that could, in principle, help in this low-budget regime. TuRBO [1] uses local trust regions, BAxUS [14] adapts nested low-dimensional embeddings, SAASBO [15] introduces a sparse axis-aligned subspace prior on the GP lengthscales, and REMBO [16] and HeSBO [17] search through random embeddings. A related linear-embedding line [18] uses Mahalanobis-style kernels on random subspaces; like GEOCMA, it restricts search to a low-dimensional subspace, but remains box-supported in the ambient space. Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [19, 20] provides a complementary evolutionary baseline that does not rely on a probabilistic surrogate.

Linear BO [21] applies an inverse stereographic projection inside the kernel of a linear GP; this changes the geometry seen by the model, while the search domain stays a cube. The on-sphere scaffold instead restricts where candidates are proposed. A complementary line of work designs Gaussian processes that respect a Riemannian manifold structure. GaBO [22] originally proposed Bayesian optimization on Riemannian manifolds, including the sphere, using a geodesic squared-exponential kernel. The Riemannian Matérn variant [23] uses Matérn Gaussian processes on Riemannian manifolds [24].

Recent work has also shown that Gaussian processes can remain competitive in high dimensions when their priors, lengthscales, and kernel defaults are chosen carefully [25–27]. These methods provide a strong basis for comparison. Our goal is not to show that Gaussian-process optimization cannot work in high dimensions, but to test how its results depend on where candidates come from and on how the acquisition is maximized. This view is also consistent with recent work emphasizing strong non-BO baselines [28, 29]: in the VAE studied here, the appropriate random baseline is not arbitrary sampling from a box, but sampling from the encoder-mean shell. We note that direct discrete-space methods such as graph genetic algorithms [28] bypass the VAE entirely and are highly competitive at larger budgets. We restrict scope to the question of how to use a learned latent space when one is already available, since this is the setting in which a candidate support must be chosen. We make no claim about which molecular model to use.

3 Methods

3.1 Molecular optimization setup

The molecular objectives are the 12-task subset of the GuacaMol goal-directed benchmark [5], combining multi-property optimization (amlodipine, fexofenadine, osimertinib, perindopril, ranolazine, sitagliptin, zaleplon), deco hop, scaffold hop, valsartan SMARTS, and median-molecule (median molecule 1, median molecule 2) objectives. We adopt this 12-task subset directly from the LOL-BO code [30]. The valsartan SMARTS task requires generating a molecule that matches a strict SMARTS pattern. No method tested produced a match at $N = 30$, so the per-task score is 0 for every row. We exclude it from cross-task aggregation rather than letting an all-zero column dominate noise statistics, leaving 11 effective tasks. Because every row scores 0 on it, leaving it out multiplies every cross-task mean by $12/11$ and changes no comparison. Cross-task means are means of per-task means. On a matched set with the same number of seeds in every task this equals the pooled mean, and the two differ only where a task has missing cells. A cell is one run of a method on one task with one seed. Each GuacaMol table, or block of a table, uses one matched set of cells, those completed by every row, and prints its size. A row that covers only part of that set is marked, and it uses the part it covers. The LassoBench tables print the coverage of every cell instead. Cells in which a failed GP fit forced a fallback step are included, and the repository counts them per method.

Each run is limited to $N = 30$ evaluations. The first 5 evaluations are fixed cold-start molecules shared by all optimizers for the same seed, and the optimizer then makes exactly 25 proposals. A proposal that decodes to an already scored molecule, or does not decode, uses one of the 25.

The cold-start set is drawn without replacement from a fixed file of the first $10{,}000$ ZINC-250k molecules using the seed as random-number-generator state, so all methods see identical cold-start molecules within a seed but different molecules across seeds. They are the same for every task and method, are encoded by the encoder mean, and are scored directly. SELFIES tokens outside the VAE's vocabulary (stereo tokens) are removed before encoding; this affects $63.1\,\%$ of the ZINC-250k molecules and every method alike. We report the best objective value found within these 30 evaluations. For the main benchmark, each method is run on the 11 non-degenerate GuacaMol tasks and on 40 random seeds, numbered 42–81. The runs of NAS-Bench-101, of two further codecs and with $N = 1000$ (Section 4.4 and Appendices E and F) are stored runs from an earlier code version and were not rerun. The BAxUS row of Table 1 has 439 of the 440 cells, because the GP fit failed in one cell of the Zaleplon task.

The primary test of every comparison is the exact two-sided Wilcoxon signed-rank test [31] on the per-task mean differences (11 tasks on GuacaMol, six on LassoBench), reported with the number of tasks on which a method is ahead. NAS-Bench-101 has a single task, so it is tested over seeds. The task is the unit because the cold-start molecules of a seed are shared across tasks, so the cells of one seed are not independent. Cell-level Wilcoxon tests are secondary, and so is Cohen's d, which we report as a cell-level effect size, computed as the mean paired difference divided by the standard deviation of paired differences. Benjamini–Hochberg correction [32] is applied within families: the five contrasts of the factorial experiment, the three recipe rows of Table 2, the tests of the published recipes (Table 15), and the LassoBench tests of the published methods. The secondary cell-level tests of the appendix tables are corrected within each block. The per-task comparison of GEOWALK with the GP on the shell (Appendix C) uses Holm correction over the 11 tasks. The task-level p in Tables 1 and 4 is uncorrected, and we call a result significant only within a corrected family.

The GuacaMol tasks have very different absolute score ranges (per-task means range from below 0.1 to above 0.7, Table 8). The standard error of the unweighted cross-task mean is therefore dominated by between-task heterogeneity rather than by between-method differences and is not informative for method comparison. We therefore base statistical claims on *per-task* differences and do not attach standard-error bands to cross-task means.

3.2 Empirical latent geometry and the *on-sphere scaffold*

The motivation for the on-sphere scaffold comes from the empirical geometry of the VAE latent space. When the $249{,}455$ molecules of the ZINC-250k reference set, which is not the VAE's training set, are encoded by the VAE, their encoder mean vectors $\mu(x) \in \mathbb{R}^{256}$ do not fill the surrounding ambient space uniformly. Instead, their norms concentrate near a relatively thin spherical shell, with empirical mean and standard deviation 7.98 ± 0.71 and 5th–95th percentile range $[6.81, 9.14]$. This is well below the norm $\sqrt{D - 1/2} \approx 16$ of a typical draw from the $\mathcal{N}(0, I_{256})$ prior. The covariance Σ of the encoder means has effective rank (participation ratio) $R_0 = (\operatorname{tr}\Sigma)^2/\operatorname{tr}(\Sigma^2) = 64.7$, and 66 principal components account for $90\,\%$ of the total variance (Appendix I). We never sample posterior noise during optimization, so we measure the shell radius on encoder means. These statistics describe the shell. They do not guarantee that it is the best place to search for every objective. Relatedly, Arvanitidis et al. [33] show that the latent space of a deep generative model is in general not flat Euclidean and that ignoring this structure can substantially mislead distance-based queries. The empirical concentration of VAE latents on a sphere is consistent with a separate line of work on *hyperspherical* VAEs, which makes this geometry explicit by placing the latent distribution on the unit sphere [34–36]. Our observation can be read as the implicit version of the same phenomenon: rather than redesigning the codec, the *on-sphere scaffold* uses the implicit shell that this Gaussian-prior VAE already exhibits as a constraint on candidate generation. Every newly proposed candidate is placed on this shell, the *empirical sphere*,

$$x = \bar{r}u, \qquad u \in S^{D-1}, \tag{1}$$

where $\bar{r}$ is the empirical average norm of the encoded cold-start latents within each run and u is a unit direction in the D-dimensional latent space. This $\bar{r}$ is an estimate of the shell radius reported above, and Appendix I compares it with other estimators and with fixed radii. A uniform sample from this scaffold is obtained by drawing $z \sim \mathcal{N}(0, I_D)$ and returning $\bar{r}z/\|z\|$. The scaffold constrains the norm of a candidate; it does not assume that the latent distribution is isotropic. Restricting the directions to the principal subspace of the encoder means did not improve on uniform directions

(Appendix I). Random search, GEOWALK, GEOCMA, and the on-sphere GP variants use the scaffold in Eq. (1), whereas the external baselines use their own candidate generation (Appendix H).

Acquisition and its maximization. The GP of the factorial experiment uses the log expected improvement (qLogEI); the acquisition of every other row is given in its label or, where the label has none, in Appendix H. We compare two ways of maximizing it. With *candidate-set argmax*, each step draws $2{,}048$ fresh uniform candidates from the support and proposes the one with the highest acquisition value. With *continuous maximization*, the acquisition is optimized by L-BFGS-B, the quasi-Newton optimizer of BoTorch [37], from 10 starts chosen among 512 raw samples, on both supports. In the box the raw samples are Sobol points of the box, and the optimizer is bounded by the box. On the shell the raw samples are random directions $v \in \mathbb{R}^D$, left unbounded; the acquisition is optimized over v and evaluated at $\bar{r}v/\|v\|$, so every iterate lies on the shell. The *box* spans, in each coordinate, the 1st to 99th percentile of $50{,}000$ encoded ZINC-250k molecules. The kernel and its prior are the same on both supports: a Matérn-5/2 kernel with the dimension-scaled lengthscale prior (DSP) of Hvarfner et al. [25]. Box candidates are rescaled per coordinate to $[0,1]^D$ before the GP, and shell candidates are not.

3.3 GEOWALK: geodesic local search on the empirical sphere

GEOWALK is a deliberately minimal optimizer for the *on-sphere scaffold*. It is designed to test how far one can get without fitting a surrogate model once **candidate generation is restricted to the encoder-mean shell**. Like every optimizer on the molecular benchmark, GEOWALK starts from the same $N_0 = 5$ cold-start points. The remaining budget is $T = N - N_0$ evaluations. GEOWALK first uses $N_1 = \lfloor \rho T \rfloor$ of them for uniform exploration of the empirical sphere, where ρ is the exploration ratio. The remaining $T - N_1$ evaluations are used for local geodesic refinement from the best latent point found so far. The two phases balance exploration and exploitation: the uniform samples explore the shell, and the geodesic steps exploit the best point found so far. At the default budget $N = 30$, we use $\rho = 0.64$, which gives $N_1 = 16$ exploration samples and 9 refinement steps.

The refinement phase is a spherical variant of the standard $(1+1)$ evolution strategy. In each refinement step, the algorithm keeps the best point found so far, proposes one new candidate near it on the sphere, and updates the reference point only if the new candidate improves the objective. Let $x_\star$ be the current best latent and let $u_\star = x_\star/\|x_\star\|$ be its unit direction. To propose a nearby candidate without leaving the sphere, we draw $v \sim \mathcal{N}(0, I_D)$, remove its radial component, normalize the resulting tangent direction, and take a geodesic step:

$$v \leftarrow v - \langle v, u_\star \rangle u_\star, \qquad v \leftarrow v/\|v\|, \tag{2}$$

$$u_t = \cos(\alpha_t) u_\star + \sin(\alpha_t) v, \qquad x_t = \bar{r} u_t. \tag{3}$$

The angle α_t decays linearly from $\alpha_{\max} = 0.60$ to $\alpha_{\min} = 0.15$ radians during the refinement phase. This creates a simple spherical analogue of a shrinking local search region, but without fitting a GP, learning a covariance matrix, or optimizing an acquisition function. We also tested an adaptive step-size variant related to the classical $1/5$ success rule [38, 39]. That rule raises the step size when more than about a fifth of recent steps improve the objective, and lowers it otherwise. Unlike that rule, this variant updates the angle after every step: it starts at 0.30 radians, is multiplied by 1.2 after an improving step and by 0.85 otherwise, and stays within $[0.05, 0.50]$. The adaptive variant underperforms the fixed decreasing schedule, and so does a constant angle of 0.15, while constant angles of 0.60 and 0.35 show no detected difference from the fixed schedule. We therefore use the fixed angular schedule as a deliberate low-budget design choice rather than as a replacement for full evolution-strategy step-size theory. A full sensitivity analysis of ρ and the angle schedule is reported in Appendix B.

3.4 GEOCMA: covariance learning in a lifted spherical subspace

GEOCMA is designed to test when the geodesic random walk of GEOWALK is no longer sufficient. It keeps the same *on-sphere scaffold* from Eq. (1), but replaces both phases of GEOWALK with covariance learning. The motivation is that **some search spaces may contain directional structure that can be learned even from a small budget**, so that the sampling distribution on the sphere can be refined toward promising directions.

To do this at low budget, GEOCMA runs Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [19, 20] in a random low-dimensional subspace rather than in the full D-dimensional latent space. We draw a Gaussian random matrix and compute an orthonormal basis $Q \in \mathbb{R}^{D \times \tilde{D}}$ by QR decomposition, with $\tilde{D} = 16$ in our experiments. CMA-ES works in the projected coordinates $Q^\top x$ of this subspace. There is no uniform exploration phase: its search distribution starts at the projection of the best cold-start point, and the projected cold-start points with their scores are its first update. Each proposal $s \in \mathbb{R}^{\tilde{D}}$ is normalized to the unit sphere in the subspace, lifted to the ambient latent space by Q, whose orthonormal columns preserve the norm, and scaled to the empirical radius:

$$\tilde{s} = \frac{s}{\|s\|}, \qquad x = \bar{r}\, Q\tilde{s}. \tag{4}$$

On the molecular benchmark, CMA-ES runs with population size $\lambda = 5$ and initial step size $\sigma_0 = 0.5$, and every GEOCMA proposal lies on the same empirical shell as the proposals of GEOWALK, restricted to the directions of the subspace. After this normalization only the direction of s matters: the covariance learned by CMA-ES shapes which directions are proposed, and its step size sets how widely they spread around the current mean of its search distribution. The two other benchmarks differ in the support. On LassoBench, the CMA-ES sample is lifted without either normalization and clipped to the box $[-1, 1]^D$, so there GEOCMA does not search a sphere (Section 4.4). On NAS-Bench-101, the arch2vec latent space [40] has 16 dimensions, so the subspace is the whole space and GEOCMA is CMA-ES on the sphere. Full pseudocode for both methods is provided in Appendix A (Algorithms 1 and 2), and Appendix B compares $\tilde{D} = 8$ and $\tilde{D} = 64$ with the default.

4 Experiments

The experiments are organized around four questions. (1) How do established optimizers compare with uniform random search on the shell (Section 4.1)? (2) On the shell, does a search without a surrogate differ from a GP (Section 4.2)? (3) Does the advantage of the shell come from the support, from the acquisition maximizer, or from both, when the two are varied separately (Section 4.3)? (4) When does the random-walk behavior of GEOWALK stop being sufficient, so that covariance learning in GEOCMA becomes useful (Section 4.4)?

4.1 Random search on the shell and the external baselines

The first experiment tests whether the empirical latent sphere is already a strong search support, even before adding any surrogate model or adaptive local structure. We compare uniform random search on the scaffold in Eq. (1) with classical high-dimensional BO and evolutionary baselines, as adapted to this benchmark (Appendix H), on one matched set (Table 1). The comparators are TuRBO [1], BAxUS [14] and full-dimensional CMA-ES, the random-embedding methods REMBO and HeSBO, and four published GP recipes (Vanilla BO, Papenmeier MSR, Xu Standard GP and Linear BO), which recent work reports as strong in high dimensions (Section 2). We keep the VAE fixed, so methods that update it during the search, such as LOL-BO [2] and InvBO [3], are not compared. Random search is intentionally minimal: at each step it draws a fresh Gaussian vector, normalizes it to the unit sphere, scales it by $\bar{r}$, decodes the latent point and evaluates the objective.

The bottom rows of Table 1 show that this simple on-sphere random search already has a higher mean at $N = 30$ than every comparator that proposes candidates off the shell. GaBO and the Riemannian Matérn GP also propose at radius $\bar{r}$, with kernels defined on the sphere, so they are not counted among the rows off the shell. Random search is ahead on most tasks against each of these comparators. With the task as the unit of the test (Section 3.1), none of these leads is significant. Table 1 does not print this test, because its p column tests each row against GEOWALK, not against random search. The higher mean of random search should not be read as a general criticism of these methods: most were designed for larger budgets or for direct parameter spaces. Appendix H lists the restrictions under which each method ran.

Against the published recipes, GEOWALK has the higher mean in every case, but the difference is significant only against one of the two recipes of Papenmeier et al., MLE (scaled) (Table 15 in Appendix H). The closest of them is Linear BO [21], which runs in its authors' box $[-8, 8]^D$ from the shared cold-start molecules. GEOWALK is higher on 10 of the 11 tasks, and the difference is not significant. Appendix D also has our port of Linear BO to the shell.

Table 1: **Cross-task means and paired statistics for GEOWALK versus all comparators at $N = 30$** on one matched set of $n = 440$ (task, seed) cells of the 11 GuacaMol tasks. REMBO and HeSBO propose on the empirical shell; GaBO and Riemannian Matérn GP propose on the sphere manifold; all other comparators propose off the shell. Δ is the improvement of GEOWALK over the comparator. W/L counts the matched cells where GEOWALK wins or loses, excluding ties. p_{task} is the exact two-sided Wilcoxon test on the 11 per-task mean differences, not corrected. d is Cohen's d of the paired cell differences. Bold marks the highest mean. Notes are in Appendix H.

Method	Cross-mean	Δ	W/L	p_{task}	d
CMA-ES [19, 20]	0.4026	+0.0200	253/148	0.12	0.25
TuRBO [1]	0.3790	+0.0437	270/123	0.032	0.43
BAxUS[‡] [14]	0.3907	+0.0320	283/101	0.024	0.34
REMBO [16]	0.4092	+0.0135	238/144	0.002	0.23
HeSBO [17]	0.4120	+0.0106	225/158	0.032	0.17
Vanilla BO [25]	0.4008	+0.0218	238/159	0.083	0.26
Papenmeier MSR [26]	0.3899	+0.0327	281/109	0.019	0.40
Xu Standard GP [27]	0.3951	+0.0275	270/120	0.042	0.37
Linear BO [21]	0.4101	+0.0125	226/172	0.032	0.17
GaBO [22]	0.4153	+0.0074	231/165	0.17	0.13
Riemannian Matérn GP [23]	0.4059	+0.0167	230/152	0.014	0.26
This paper, on the empirical shell (Eq. 1):					
Random Search (shell)	0.4118	+0.0108	194/64	0.00098	0.30
GEOCMA	0.4158	+0.0068	225/169	0.054	0.11
GEOWALK	**0.4226**	–	–	–	–

4.2 What a surrogate adds on the shell

We next ask whether a Gaussian-process surrogate still provides a substantial advantage once candidate generation is restricted to the same on-sphere scaffold. We compare GEOWALK with uniform random search on the shell and with the GP of Section 3.2 with candidate-set argmax, which we call *the GP on the shell*. It adapts the lengthscale prior of Vanilla BO [25] to the shell. It is not a baseline but the scaffold with a surrogate, so Table 2 prints it beside its box version, the GP in the box, next to the recipes adapted from Papenmeier MSR and Xu Standard GP. The cross-mean of the GP on the shell is above that of every method in Table 1, GEOWALK included, but its lead over GEOWALK is not significant, so it is the main comparator of GEOWALK here.

Earlier in the budget, the GP on the shell is ahead of GEOWALK, and after thirty evaluations we detect no difference between them. At that budget, GEOWALK is higher on 6 of the 11 tasks. The task-level test does not separate the two ($p = 0.83$), and no per-task difference is significant (Appendix C). After twenty evaluations of the same runs, the GP on the shell is ahead of GEOWALK on every task (Appendix D). At that point GEOWALK is still in its uniform exploration phase, which ends only after its 21st evaluation (Section 3.3), so until then its runs coincide with random search. The gain of GEOWALK over uniform random search on the shell at $N = 30$ is small but consistent: GEOWALK is higher than random search on every task (Table 8). The memory and the geodesic refinement steps of GEOWALK therefore add value beyond the scaffold alone.

4.3 Candidate support and acquisition maximization

Table 2 compares three GP recipes on the box and on the shell: a Matérn-5/2 GP with the DSP prior [25], one with the lengthscale initialized at $0.1\sqrt{D}$ as in MSR [26], and one initialized at $\sqrt{D}$ as in Xu et al. [27]. Each recipe uses L-BFGS-B in the box and candidate-set argmax on the shell, so these rows change the acquisition maximizer together with the support and cannot isolate the support. The $0.1\sqrt{D}$ box row also changes the acquisition function (Appendix H). A factorial experiment with the DSP-prior GP therefore separates the two factors (Table 3). It crosses the support, shell or box, with the maximizer: candidate-set argmax or continuous maximization by L-BFGS-B, the same on both supports (Section 3.2). Two of its four cells are the two DSP rows of Table 2. The five contrasts of the factorial, with their task counts and p values, are in Appendix J.

Table 2: **Box versus empirical shell within each GP recipe.** Each GP on the empirical shell (candidate-set argmax) and in the box $[p_{01}, p_{99}]$ (L-BFGS-B), on one matched set of $n = 440$ cells; cross-means as in Table 1. Tasks counts the tasks, of the 11, on which the shell version has the higher mean. $p^{\mathrm{BH}}_{\mathrm{task}}$ is the task-level test of the shell version against the box version, corrected within the three recipes. The two DSP rows are the GP on the shell and the GP in the box of Sections 4.2 and 4.3.

Variant	Candidate support	Cross-mean	Tasks shell higher	$p^{\mathrm{BH}}_{\mathrm{task}}$
Matérn-5/2 GP (DSP prior), shell	Empirical shell	0.4228	9/11	0.081
Matérn-5/2 GP (DSP prior), box	$[p_{01}, p_{99}]$	0.3989		
Matérn-5/2 GP ($\ell_0 = 0.1\sqrt{D}$), shell	Empirical shell	0.4176	9/11	0.081
Matérn-5/2 GP ($\ell_0 = 0.1\sqrt{D}$), box	$[p_{01}, p_{99}]$	0.3845		
Matérn-5/2 GP ($\ell_0 = \sqrt{D}$), shell	Empirical shell	0.4149	9/11	0.083
Matérn-5/2 GP ($\ell_0 = \sqrt{D}$), box	$[p_{01}, p_{99}]$	0.3957		

Table 3: **The support $\times$ acquisition-maximization factorial: cross-mean of each cell.** One Matérn-5/2 GP (DSP prior) on $n = 440$ matched cells; continuous maximization is L-BFGS-B on both supports. Cross-means as in Table 1. The five contrasts of these cells, with their task counts and corrected p values, are in Table 19 (Appendix J).

Support	Candidate-set argmax	Continuous (L-BFGS-B)
Empirical shell	0.4228	0.3999
Box $[p_{01}, p_{99}]$	0.3925	0.3989

The shell helps when the acquisition is maximized over a finite candidate set, and not when it is maximized continuously by L-BFGS-B, which on the shell optimizes over directions and stays on it. Table 3 gives the cross-mean of each cell, and Table 19 in Appendix J the five contrasts, each with the number of tasks on which it is positive and its p value. With candidate-set argmax, the cross-mean is 0.4228 on the shell and 0.3925 in the box. The shell is ahead on 9 of the 11 tasks, and this difference is significant. With L-BFGS-B, the shell is ahead on 6 of the 11 tasks, and the difference is not significant. The interaction, the difference between these two support effects, is also significant: on 9 of the 11 tasks, the lead of the shell over the box is larger under candidate-set argmax than under L-BFGS-B. The maximizer acts in opposite directions on the two supports. On the shell, candidate-set argmax is ahead of L-BFGS-B on most tasks. In the box, L-BFGS-B is ahead of candidate-set argmax on nearly every task. Both of these contrasts are significant.

The recipe rows of Table 2 point the same way. For each recipe, the shell version is ahead of the box version on 9 of the 11 tasks, but none of these differences is significant.

4.4 When covariance learning is useful

Once candidate generation is restricted to the empirical shell, a simple geodesic random walk shows no detected difference from the GP on the shell after thirty evaluations. On the molecular benchmark, GEOCMA does not improve on GEOWALK, which is higher on 9 of the 11 tasks (Table 4). To find where a random walk stops being sufficient, we compare GEOWALK and GEOCMA on NAS-Bench-101 and LassoBench.

On these two benchmarks we make no claim about a shell, and the sphere is only a fixed-radius search domain. The radius and the clipping of each benchmark are in Appendix I. NAS-Bench-101 [9] is searched through the 16-dimensional latent space of an arch2vec graph VAE [40] over about $423{,}000$ trained architectures, with CIFAR-10 validation accuracy as the objective. LassoBench [8] is a sparse-regression benchmark with no decoder. We use six of its tasks of increasing dimension: four synthetic tasks, DNA and leukemia. There, GEOCMA clips its points to the box $[-1, 1]^D$ without normalizing them, so it does not search a sphere. The column *Best other method* of Table 4 is the best of the comparators of Table 1 and the GP on the shell on GuacaMol, and the

Table 4: **GEOWALK and GEOCMA across benchmark regimes.** $N = 30$. Arrows give the optimization direction, and bold marks the better of the two means. GuacaMol: mean of per-task means on the matched cells. LassoBench: each method's loss divided by the random-search mean loss of the task, averaged over the six tasks. n is the number of paired cells, or of paired seeds on NAS-Bench-101. Best other method: the best mean among the other methods that cover the row's matched set. Δ is paired and favors the better method, $|d|$ is Cohen's d of the paired cells, and p_{task} is the task-level test of Section 3.1, uncorrected across the rows; ^{a} marks the paired test over seeds, because NAS-Bench-101 has a single task.

Benchmark	Dir.	GEOWALK	GEOCMA	Δ	$\lvert d \rvert$	p_{task}	n	Best other method
GuacaMol mol VAE (256D)	↑	**0.423**	0.416	0.007	0.11	0.054	440	GP on the shell, 0.423
NAS-Bench-101 graph VAE (16D)	↑	0.937	**0.942**	0.004	0.94	0.020^a	10	random search, 0.941
LassoBench, $D \in [60, 7129]$, normalized	↓	0.796	**0.495**	0.300	0.37	0.094	240	CMA-ES, 0.428

best row of Table 12 on LassoBench. On NAS-Bench-101 it is the best of random search, TuRBO, a GP and full-dimensional CMA-ES.

Table 4 summarizes the boundary. GEOWALK remains ahead of GEOCMA in mean on the molecular VAE benchmark, where local random movement on the empirical sphere is sufficient. Outside the molecular benchmark, GEOCMA has the better mean of the two, but neither method is detectably ahead of the best other method there. On LassoBench, GEOCMA has a lower normalized loss than GEOWALK on 5 of the six tasks, but the task-level test does not detect this difference. The best method there is full-dimensional CMA-ES started at the origin (Appendix H): it has a lower normalized loss than GEOCMA on five of the six tasks. It also has a lower loss than GEOWALK on all six tasks, but this sweep is not significant once corrected within the published methods (Table 13). On NAS-Bench-101, which has a single task and is therefore tested over seeds, GEOCMA improves over GEOWALK. Random search, however, is ahead of GEOWALK there, with a Wilcoxon p of 0.037 over the matched seeds, and GEOCMA does not differ detectably from random search.

5 Discussion

The experiments support a simple interpretation: in the studied molecular VAE benchmark, where the candidates came from mattered more than which surrogate scored them, as long as the acquisition was maximized over a candidate set. Candidates on the encoder-mean shell then outperformed candidates from the box. All three methods on the shell are the on-sphere scaffold with a different search inside it. The GP on the shell is the scaffold with a surrogate, GEOWALK is the scaffold without one, and GEOCMA is the scaffold with covariance learning. The GP on the shell has the highest cross-mean of any method on this benchmark (Tables 1 and 2), and GEOWALK is level with it after thirty evaluations with no surrogate at all. At this budget of thirty evaluations the surrogate is therefore superfluous once the candidates lie on the shell. Earlier in the budget it is ahead, while GEOWALK is still exploring. This does not make surrogates unnecessary in general: on LassoBench and NAS-Bench-101, GEOCMA has better means than GEOWALK, while on the molecular benchmark covariance learning does not improve on the walk at $N = 30$. Comparisons of latent optimizers should therefore control and report the support and the acquisition maximizer before a gain is attributed to the surrogate.

Limitations. The advantage of the shell is shown only for candidate sets drawn uniformly from the whole support; local candidate sets, as in trust-region methods, were not tested. The evidence comes from this Gaussian-prior VAE and the 11 GuacaMol tasks of the protocol. In a separate run of the same factorial experiment on nine further GuacaMol tasks, the shell was ahead under candidate-set argmax on only three of them (not shown in the tables). On a second VAE, Chembed [41], whose encoder means also form a shell, moving the GP to the shell gave no detected gain (Appendix E). Locating a shell is therefore no guarantee that searching on it helps.

Budget scaling stops at $N = 1000$, in stored runs on ten seeds (Appendix F), so results should not be read asymptotically. The cross-domain evidence for GEOCMA rests mainly on LassoBench, because the NAS-Bench-101 row has only 10 matched seeds. Other latent spaces were not tested: text VAEs, normalizing flows, diffusion models and the soft-prompt spaces of multimodal language models. Finally, better low-budget molecular optimization could lower experimental cost, but it has potential dual-use risks if applied to harmful objectives.

Conclusion. In this Gaussian-prior VAE benchmark, shell candidates outperformed box candidates when acquisition was maximized over a candidate set, but not with continuous maximization by L-BFGS-B. The GP on the shell, our scaffold with a surrogate, had the highest cross-mean of any method tested. A surrogate-free geodesic walk on the shell was level with it after thirty evaluations, so at this budget the surrogate is superfluous. Earlier in the budget the surrogate was ahead. This is a controlled study of a Gaussian-prior VAE benchmark, not a recommendation to use VAEs for molecular optimization. The lesson of this study is to control the candidate support and the acquisition maximizer before attributing a gain to the surrogate.

Code and data availability. The code, run configurations, result files and the generator of every table, figure and number are available at `https://github.com/michalprusek/geometry-before-surrogates-neurips2026`. The NAS-Bench-101 embedding is not available; its result files are included. The project page, with the paper, the poster and the results, is `https://staff.utia.cas.cz/novozada/geo`.

Acknowledgments and Disclosure of Funding

This project started in the Legal Data Analysis Lab, a master's practical course at the Technical University of Munich; we thank Matthias Grabmair and Mohamed Elganayni for their guidance during the course. This work was supported by the Czech Science Foundation, grant GA25-15933S, and by the Grant Agency of the Czech Technical University in Prague, grant SGS24/141/OHK4/3T/14. The experiments were run on the computing servers of the Institute of Information Theory and Automation, Czech Academy of Sciences. The authors declare no competing interests.

References

[1] David Eriksson, Michael Pearce, Jacob R. Gardner, Ryan D. Turner, and Matthias Poloczek. Scalable global optimization via local Bayesian optimization. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/hash/6c990b7aca7bc7058f5e98ea909e924b-Abstract.html.

[2] Natalie Maus, Haydn T. Jones, Juston S. Moore, Matt J. Kusner, John Bradshaw, and Jacob R. Gardner. Local latent space Bayesian optimization over structured inputs. In *Advances in Neural Information Processing Systems*, volume 35, pages 34505–34518. Curran Associates, Inc., 2022. doi: 10.52202/068431-2500.

[3] Jaewon Chu, Jinyoung Park, Seunghun Lee, and Hyunwoo J. Kim. Inversion-based latent Bayesian optimization. In *Advances in Neural Information Processing Systems*, volume 37, pages 68258–68286. Curran Associates, Inc., 2024. doi: 10.52202/079017-2180.

[4] Mario Krenn, Florian Häse, AkshatKumar Nigam, Pascal Friederich, and Alan Aspuru-Guzik. Self-referencing embedded strings (SELFIES): A 100% robust molecular string representation. *Machine Learning: Science and Technology*, 1(4):045024, 2020. doi: 10.1088/2632-2153/aba947.

[5] Nathan Brown, Marco Fiscato, Marwin H. S. Segler, and Alain C. Vaucher. GuacaMol: Benchmarking models for de novo molecular design. *Journal of Chemical Information and Modeling*, 59(3):1096–1108, 2019. doi: 10.1021/acs.jcim.8b00839.

[6] John J. Irwin, Teague Sterling, Michael M. Mysinger, Erin S. Bolstad, and Ryan G. Coleman. ZINC: A free tool to discover chemistry for biology. *Journal of Chemical Information and Modeling*, 52(7):1757–1768, 2012. doi: 10.1021/ci3001277.

[7] Rafael Gómez-Bombarelli, Jennifer N. Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D. Hirzel, Ryan P. Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, 4(2):268–276, 2018. doi: 10.1021/acscentsci.7b00572.

[8] Kenan Šehić, Alexandre Gramfort, Joseph Salmon, and Luigi Nardi. LassoBench: A high-dimensional hyperparameter optimization benchmark suite for Lasso. In *Proceedings of the First International Conference on Automated Machine Learning*, volume 188 of *Proceedings of Machine Learning Research*, pages 2/1–24. PMLR, 2022. URL https://proceedings.mlr.press/v188/sehic22a.html.

[9] Chris Ying, Aaron Klein, Eric Christiansen, Esteban Real, Kevin Murphy, and Frank Hutter. NAS-Bench-101: Towards reproducible neural architecture search. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7105–7114. PMLR, 2019. URL https://proceedings.mlr.press/v97/ying19a.html.

[10] Seunghun Lee, Jaewon Chu, Sihyeon Kim, Juyeon Ko, and Hyunwoo J. Kim. Advancing Bayesian optimization via learning correlated latent space. In *Advances in Neural Information Processing Systems*, volume 36, pages 48906–48917. Curran Associates, Inc., 2023. doi: 10.52202/075280-2124.

[11] Seunghun Lee, Jinyoung Park, Jaewon Chu, Minseo Yoon, and Hyunwoo J. Kim. Latent Bayesian optimization via autoregressive normalizing flows. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=ZCOwwRAaEl.

[12] Natalie Maus, Kyurae Kim, Geoff Pleiss, David Eriksson, John P. Cunningham, and Jacob R. Gardner. Approximation-aware Bayesian optimization. In *Advances in Neural Information Processing Systems*, volume 37, pages 21114–21140. Curran Associates, Inc., 2024. doi: 10.52202/079017-0665.

[13] Henry Moss, Sebastian W. Ober, and Tom Diethe. Return of the latent space COWBOYS: Rethinking the use of VAEs for Bayesian optimisation of structured spaces. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 44956–44970. PMLR, 2025. URL https://proceedings.mlr.press/v267/moss25a.html.

[14] Leonard Papenmeier, Luigi Nardi, and Matthias Poloczek. Increasing the scope as you learn: Adaptive Bayesian optimization in nested subspaces. In *Advances in Neural Information Processing Systems*, volume 35, pages 11586–11601. Curran Associates, Inc., 2022. doi: 10.52202/068431-0842.

[15] David Eriksson and Martin Jankowiak. High-dimensional Bayesian optimization with sparse axis-aligned subspaces. In *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pages 493–503. PMLR, 2021. URL https://proceedings.mlr.press/v161/eriksson21a.html.

[16] Ziyu Wang, Frank Hutter, Masrour Zoghi, David Matheson, and Nando de Freitas. Bayesian optimization in a billion dimensions via random embeddings. *Journal of Artificial Intelligence Research*, 55:361–387, 2016. doi: 10.1613/jair.4806.

[17] Amin Nayebi, Alexander Munteanu, and Matthias Poloczek. A framework for Bayesian optimization in embedded subspaces. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 4752–4761. PMLR, 2019. URL https://proceedings.mlr.press/v97/nayebi19a.html.

[18] Ben Letham, Roberto Calandra, Akshara Rai, and Eytan Bakshy. Re-examining linear embeddings for high-dimensional Bayesian optimization. In *Advances in Neural Information Processing Systems*, volume 33, pages 1546–1558. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/10fb6cfa4c990d2bad5ddef4f70e8ba2-Abstract.html.

[19] Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001. doi: 10.1162/106365601750190398.

[20] Nikolaus Hansen, Youhei Akimoto, and Petr Baudis. CMA-ES/pycma on Github. Zenodo, 2019. URL https://doi.org/10.5281/zenodo.2559634.

[21] Colin Doumont, Donney Fan, Natalie Maus, Jacob R. Gardner, Henry Moss, and Geoff Pleiss. We still don't understand high-dimensional Bayesian optimization. In *Proceedings of The 29th International Conference on Artificial Intelligence and Statistics*, volume 300 of *Proceedings of Machine Learning Research*, pages 2629–2637. PMLR, 2026. URL https://proceedings.mlr.press/v300/doumont26a.html.

[22] Noémie Jaquier, Leonel Rozo, Sylvain Calinon, and Mathias Bürger. Bayesian optimization meets Riemannian manifolds in robot learning. In *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pages 233–246. PMLR, 2020. URL https://proceedings.mlr.press/v100/jaquier20a.html.

[23] Noémie Jaquier, Viacheslav Borovitskiy, Andrei Smolensky, Alexander Terenin, Tamim Asfour, and Leonel Rozo. Geometry-aware Bayesian optimization in robotics using Riemannian Matérn kernels. In *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pages 794–805. PMLR, 2022. URL https://proceedings.mlr.press/v164/jaquier22a.html.

[24] Viacheslav Borovitskiy, Alexander Terenin, Peter Mostowsky, and Marc Peter Deisenroth. Matérn Gaussian processes on Riemannian manifolds. In *Advances in Neural Information Processing Systems*, volume 33, pages 12426–12437. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/92bf5e6240737e0326ea59846a83e076-Abstract.html.

[25] Carl Hvarfner, Erik Orm Hellsten, and Luigi Nardi. Vanilla Bayesian optimization performs great in high dimensions. In *Proceedings of the 41st International Conference on Machine*

Learning, volume 235 of *Proceedings of Machine Learning Research*, pages 20793–20817. PMLR, 2024. URL https://proceedings.mlr.press/v235/hvarfner24a.html.

[26] Leonard Papenmeier, Matthias Poloczek, and Luigi Nardi. Understanding high-dimensional Bayesian optimization. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 47902–47923. PMLR, 2025. URL https://proceedings.mlr.press/v267/papenmeier25a.html.

[27] Zhitong Xu, Haitao Wang, Jeff M. Phillips, and Shandian Zhe. Standard Gaussian process is all you need for high-dimensional Bayesian optimization. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=kX8h23UG6v.

[28] Austin Tripp and José Miguel Hernández-Lobato. Genetic algorithms are strong baselines for molecule generation. arXiv preprint, 2023. URL https://arxiv.org/abs/2310.09267.

[29] Miguel González-Duque, Richard Michael, Simon Bartels, Yevgen Zainchkovskyy, Søren Hauberg, and Wouter Boomsma. A survey and benchmark of high-dimensional Bayesian optimization of discrete sequences. In *Advances in Neural Information Processing Systems*, volume 37, pages 140478–140508. Curran Associates, Inc., 2024. doi: 10.52202/079017-4459.

[30] Natalie Maus. Local latent Bayesian optimization (LOLBO). GitHub repository, official implementation of Local Latent Space Bayesian Optimization over Structured Inputs, 2022. URL https://github.com/nataliemaus/lolbo. Commit 0876147, 31 October 2022.

[31] Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945. doi: 10.2307/3001968.

[32] Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(1):289–300, 1995. doi: 10.1111/j.2517-6161.1995.tb02031.x.

[33] Georgios Arvanitidis, Lars Kai Hansen, and Søren Hauberg. Latent space oddity: On the curvature of deep generative models. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=SJzRZ-WCZ.

[34] Tim R. Davidson, Luca Falorsi, Nicola De Cao, Thomas Kipf, and Jakub M. Tomczak. Hyperspherical variational auto-encoders. In *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence*, pages 856–865. AUAI Press, 2018. URL https://www.auai.org/uai2018/proceedings/papers/309.pdf.

[35] Jiacheng Xu and Greg Durrett. Spherical latent spaces for stable variational autoencoders. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4503–4513. Association for Computational Linguistics, 2018. doi: 10.18653/v1/D18-1480.

[36] Nicola De Cao and Wilker Aziz. The Power Spherical distribution. In *ICML 2020 Workshop on Invertible Neural Networks, Normalizing Flows, and Explicit Likelihood Models (INNF+)*, 2020. URL https://arxiv.org/abs/2006.04437.

[37] Maximilian Balandat, Brian Karrer, Daniel R. Jiang, Samuel Daulton, Ben Letham, Andrew Gordon Wilson, and Eytan Bakshy. BoTorch: A framework for efficient Monte-Carlo Bayesian optimization. In *Advances in Neural Information Processing Systems*, volume 33, pages 21524–21538. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/f5b1b89d98b7286673128a5fb112cb9a-Abstract.html.

[38] Ingo Rechenberg. *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Number 15 in Problemata. Frommann-Holzboog, Stuttgart, 1973. ISBN 3772803733.

[39] Jens Jägersküpper. Algorithmic analysis of a basic evolutionary algorithm for continuous optimization. *Theoretical Computer Science*, 379(3):329–347, 2007. doi: 10.1016/j.tcs.2007.02.042.

[40] Shen Yan, Yu Zheng, Wei Ao, Xiao Zeng, and Mi Zhang. Does unsupervised architecture representation learning help neural architecture search? In *Advances in Neural Information Processing Systems*, volume 33, pages 12486–12498. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/937936029af671cf479fa893db91cbdd-Abstract.html.

[41] Hugo Talibart and Dimitri Gilis. Learning a chemistry-aware latent space for molecular encoding and generation with a large-scale transformer variational autoencoder. *Journal of Cheminformatics*, 18(1):123, 2026. doi: 10.1186/s13321-026-01243-0.

[42] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html.

[43] RDKit: Open-source cheminformatics. Release 2026.03.1, https://doi.org/10.5281/zenodo.19250388, 2026. URL https://www.rdkit.org.

[44] Jacob R. Gardner, Geoff Pleiss, Kilian Q. Weinberger, David Bindel, and Andrew Gordon Wilson. GPyTorch: Blackbox matrix-matrix Gaussian process inference with GPU acceleration. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/hash/27e8e17134dd7083b050476733207ea1-Abstract.html.

A Detailed pseudocode and settings for GEOWALK and GEOCMA

Algorithms 1 and 2 restate the GEOWALK and GEOCMA procedures described in Sections 3.3 and 3.4 as explicit pseudocode. The cold-start, on-sphere scaffold (Eq. 1), and GEOCMA lift step (Eq. 4) match the main-text description; the inner CMA-ES loop (sample population → evaluate → `tell`) is the standard $(\mu/\mu_w, \lambda)$ generation scheme of Hansen and Ostermeier [19], written here in population-per-generation form. The deferred-tell variant used in our implementation evaluates one candidate per step and updates CMA-ES once a full population of size λ has accumulated, which is mathematically equivalent. Table 5 lists the settings of GEOWALK and GEOCMA in the GuacaMol experiments.

Algorithm 1 GEOWALK: geodesic local search on the empirical sphere

Require: Encoder mean μ, decoder D_{dec}, oracle f, latent dimension D, budget N, cold-start molecules $\{m_i\}_{i=1}^{N_0}$, exploration ratio ρ, angles $\alpha_{\max}, \alpha_{\min}$

Encode the cold-start molecules to encoder means and score them directly: $x_i \leftarrow \mu(m_i)$ and $y_i \leftarrow f(m_i)$ for $i = 1, \ldots, N_0$
Set $\bar{r} \leftarrow \frac{1}{N_0} \sum_{i=1}^{N_0} \|x_i\|$ and $T \leftarrow N - N_0$ and $N_1 \leftarrow \lfloor \rho T \rfloor$
for $t = 1$ to N_1 **do**
  Draw $z \sim \mathcal{N}(0, I_D)$ and set $x \leftarrow \bar{r} z / \|z\|$
  Evaluate $y \leftarrow f(D_{\text{dec}}(x))$ and add (x, y) to the evaluated set
end for
for $t = N_1 + 1$ to T **do**
  Let $(x_\star, y_\star)$ be the best evaluated point so far and set $u_\star \leftarrow x_\star / \|x_\star\|$
  Draw $v \sim \mathcal{N}(0, I_D)$ and project it onto the tangent space: $v \leftarrow v - \langle v, u_\star \rangle u_\star$
  Normalize the tangent direction: $v \leftarrow v / \|v\|$
  Set $\alpha_t \leftarrow \alpha_{\max} - (\alpha_{\max} - \alpha_{\min})(t - N_1 - 1) / \max(T - N_1 - 1, 1)$
  Propose $x \leftarrow \bar{r}(\cos \alpha_t \, u_\star + \sin \alpha_t \, v)$
  Evaluate $y \leftarrow f(D_{\text{dec}}(x))$ and add (x, y) to the evaluated set
end for
return The best evaluated candidate

Algorithm 2 GEOCMA: covariance learning in a lifted spherical subspace

Require: Encoder mean μ, decoder D_{dec}, oracle f, latent dimension D, budget N, cold-start molecules $\{m_i\}_{i=1}^{N_0}$, subspace dimension $\tilde{D}$, CMA-ES population size λ, initial step size σ_0

Encode the cold-start molecules to encoder means and score them directly: $x_i \leftarrow \mu(m_i)$ and $y_i \leftarrow f(m_i)$ for $i = 1, \ldots, N_0$
Set $\bar{r} \leftarrow \frac{1}{N_0} \sum_{i=1}^{N_0} \|x_i\|$
Draw a Gaussian random matrix $G \in \mathbb{R}^{D \times \tilde{D}}$ and form the orthonormal basis $Q \in \mathbb{R}^{D \times \tilde{D}}$ via the QR decomposition $G = QR$
Project the cold-start latents into the subspace: $s_i \leftarrow Q^\top x_i$ for $i = 1, \ldots, N_0$
Let $s_\star$ be the projected cold-start point with the highest oracle score
Initialize CMA-ES with mean $s_\star$, step size σ_0, and population size λ, then pre-condition it on the cold-start data $\{(s_i, y_i)\}_{i=1}^{N_0}$
Set $T \leftarrow N - N_0$ and $t \leftarrow 0$
while $t < T$ **do**
  Sample a new population $\{s^{(1)}, \ldots, s^{(\lambda)}\}$ from CMA-ES and set $K \leftarrow \min(\lambda, T - t)$
  for $k = 1$ to K **do**
    Normalize within the subspace: $\tilde{s} \leftarrow s^{(k)} / \|s^{(k)}\|$
    Lift to the ambient space: $x \leftarrow \bar{r} \, Q \tilde{s}$ (the columns of Q are orthonormal, so $\|Q\tilde{s}\| = 1$)
    Evaluate $y_k \leftarrow f(D_{\text{dec}}(x))$ and add (x, y_k) to the evaluated set
  end for
  if $K = \lambda$ **then**
    Update CMA-ES with the evaluated population: `tell`$\big(\{(s^{(k)}, y_k)\}_{k=1}^{\lambda}\big)$
  end if
  $t \leftarrow t + K$
end while
return The best evaluated candidate

Table 5: **Settings of GEOWALK and GEOCMA on GuacaMol.** These settings hold for every task and seed. The settings that differ on LassoBench and NAS-Bench-101 are given in Section 4.4 and Appendix I.

Setting	Value
GEOWALK	
Cold-start points N_0	5
Evaluations after the cold start T	25
Budget $N = N_0 + T$	30
Exploration ratio ρ	0.64
Exploration steps $N_1 = \lfloor \rho T \rfloor$	16 uniform proposals on the shell
Refinement steps $T - N_1$	9 geodesic steps
Angle $\alpha_{\max} \to \alpha_{\min}$	$0.60 \to 0.15$ rad
Angle schedule	linear in the refinement step, over $\max(T - N_1 - 1, 1) = 8$ intervals
Anchor of a refinement step	best evaluated point so far, cold starts included
Radius $\bar{r}$	mean norm of the N_0 cold-start latents
GEOCMA	
Subspace dimension $\tilde{D}$	16
Basis Q	QR decomposition of a Gaussian $D \times \tilde{D}$ matrix
Population size λ	5
Initial step size σ_0	0.5
Initial mean	$Q^\top x$ of the best cold start
Pre-conditioning	the $N_0 = 5$ cold starts, projected by $Q^\top$, told as the first generation
Update	deferred: one evaluation per step; a generation is told once λ evaluations have accumulated
Lift	normalize in the subspace, map by Q, scale to $\bar{r}$
CMA-ES library and options	pycma 4.4.4; population λ, verbosity -9, seed drawn from NumPy's global stream (seeded by the run seed)
Shared protocol (GuacaMol)	
Tasks	11 GuacaMol tasks
Seeds	42–81 (40 seeds)
Cold start	N_0 molecules drawn without replacement from the first 10,000 ZINC molecules (NumPy RandomState of the seed), encoded by the encoder mean, scored directly
Decoding	greedy, at most 80 tokens
Time cap per proposal	decoding capped at 30 s, scoring not time-limited
Time limit per run	3,600 s for both methods, a guard against hangs, not a setting of the methods

B Hyperparameter sensitivity of GEOWALK and GEOCMA

GEOWALK has two design choices. The first is the exploration ratio ρ (default 0.64, giving $N_1 = 16$ uniform-sphere samples and $T - N_1 = 9$ geodesic refinement steps out of the $T = 25$ post-cold-start budget). The second is the angle schedule, which decays linearly from $\alpha_{\max} = 0.60$ to $\alpha_{\min} = 0.15$ radians during the refinement phase. Table 6 reports a paired comparison between the default schedule and six variants: three constant-angle variants (covering the start, middle, and end of the default schedule), two exploration-ratio variants ($\rho \approx 0.32$ and $\rho \approx 0.96$), and the success-based rule of Section 3.3. All variants are evaluated on the same 11 non-degenerate GuacaMol tasks, on one matched set of 440 cells.

Of the three constant-angle variants, only $\alpha = 0.15$ (the end of the schedule) underperforms the default linear schedule: it is ahead of the default on only 2 of the 11 tasks. The constants $\alpha = 0.60$ (the beginning of the schedule) and $\alpha = 0.35$ (the middle) show no detected difference from the default. The exploration ratio shows an asymmetric pattern: raising ρ from 0.64 to ≈ 0.96 (almost all uniform exploration) puts the variant behind the default on all 11 tasks, while reducing ρ to ≈ 0.32 (less exploration, more refinement) shows no detected difference. The success-based rule is ahead of the default on only 1 of the 11 tasks. Overall, no variant is detectably better than the default, and three are worse.

Subspace dimension of GEOCMA. The subspace dimension $\tilde{D} = 16$ is a low-budget design choice. With population size $\lambda = 5$, the 25 evaluations give five CMA-ES generations whatever $\tilde{D}$ is, so $\tilde{D}$ sets how many directions CMA-ES adapts, not how often it updates. The bottom block of Table 6 compares $\tilde{D} = 8$ and $\tilde{D} = 64$ with the default, and neither difference is detected (task-level $p = 0.21$ and 0.70).

Table 6: **Sensitivity of GEOWALK and GEOCMA to their settings.** Top: six variants of the step schedule and the exploration of GEOWALK, each against the default (angle $0.60 \to 0.15$, 16 exploration steps). Bottom: the subspace dimension $\tilde{D}$ of GEOCMA, 8 and 64 against the default 16. Both blocks use the same $n = 440$ completed cells, as in Table 1. Means are means of per-task means. Bold marks the highest mean of the top block. The difference Δ is the variant minus the default, computed from unrounded means. The effect size d is Cohen's d of the cell differences. The column p_{BH} is the two-sided Wilcoxon signed-rank test over the cells. It is Benjamini–Hochberg corrected within each block, over 6 and 2 rows. Tasks counts the tasks on which the variant has a higher mean than the default. The column p_{task} is the exact two-sided Wilcoxon test on the 11 per-task mean differences, not corrected. The task-level test of the bottom block is given in the text.

Variant	Mean	Δ	d	p_{BH}	Tasks row higher	p_{task}
constant angle 0.60	**0.4228**	+0.0001	0.00	0.11	4/11	0.46
constant angle 0.35	**0.4228**	+0.0001	0.00	0.76	6/11	0.97
constant angle 0.15	0.4189	−0.0038	−0.11	0.006	2/11	0.024
8 exploration steps ($\rho \approx 0.32$)	0.4202	−0.0024	−0.04	0.76	7/11	0.97
24 exploration steps ($\rho \approx 0.96$)	0.4133	−0.0093	−0.25	< 0.001	0/11	0.00098
success-based angle rule (start 0.30, $\times 1.2$ / $\times 0.85$, clamp $[0.05, 0.50]$)	0.4176	−0.0050	−0.14	0.005	1/11	0.0098
default ($0.60 \to 0.15$, 16 exploration steps) (reference)	0.4226	–	–	–	–	–

GEOCMA *subspace dimension, matched set* $n = 440$

$\tilde{D}$	Mean	Δ	d	p_{BH}
$\tilde{D} = 16$ (default)	0.4158	–	–	–
$\tilde{D} = 8$	0.4086	−0.0072	−0.10	0.22
$\tilde{D} = 64$	0.4184	+0.0026	0.04	0.22

C Per-task breakdown of GEOWALK vs the GP on the shell

Section 4.2 detects no difference between GEOWALK and the GP on the shell after thirty evaluations. Their paired difference is $\Delta = -0.0001$ at $n = 440$ matched task-seed cells (last row of Table 7). Table 7 shows the per-task breakdown across the 11 non-degenerate GuacaMol tasks, using the same 40 seeds as Table 1.

Per-task absolute differences are small: $|\Delta| \leq 0.0135$ across all 11 tasks. The largest differences favor the GP on the shell, on Sitagliptin and Osimertinib; the largest in favor of GEOWALK is on Ranolazine. The signs are balanced across tasks ($6/11$ favor GEOWALK). Only one per-task paired Wilcoxon test reaches $p < 0.05$ (Median 1, $p = 0.024$), and it does not survive Holm correction over the 11 tasks ($p_{\mathrm{Holm}} = 0.27$). Cohen's d values are small to moderate in either direction (range $[-0.21, +0.40]$). The cross-task mean -0.0001 is therefore a representative summary of the per-task picture rather than an artifact of cancellation.

Per-task means across the main baselines. Table 8 reports per-task means at $N = 30$ across the seven most directly comparable methods (GEOWALK, GEOCMA and five shell/box baselines). On most tasks, the five methods on the shell (GEOWALK, GEOCMA, random search on the shell, the GP on the shell, the Linear BO port to the shell) are clustered closely together within each row, while the GP in the box trails (median per-task gap to GEOWALK is -0.013). On Median Molecules 1 and Sitagliptin, however, the two box columns have the highest means.

Table 7: **GEOWALK versus the GP on the shell, per task.** Δ = GEOWALK minus the GP on the shell (Table 2). Per task, Δ, Cohen's d and the two-sided Wilcoxon p are paired over the seeds of the task's completed cells, as in Table 1. p_{Holm} is Holm-corrected over the 11 tasks. Last row: all matched cells, as a mean of per-task means. The p of the last row is the cell-level test over all cells. That test is not one of the per-task tests, so it has no p_{Holm} (–). Task codes: adip = Amlodipine MPO, dhop = Deco Hop, fexo = Fexofenadine MPO, med1 = Median molecules 1, med2 = Median molecules 2, osmb = Osimertinib MPO. Further codes: pdop = Perindopril MPO, rano = Ranolazine MPO, shop = Scaffold Hop, siga = Sitagliptin MPO, zale = Zaleplon MPO. MPO = multi-property objective.

Task	n	GEOWALK	GP on the shell	Δ	d	p	p_{Holm}
adip	40	0.5013	0.5037	−0.0024	−0.08	0.92	1.00
dhop	40	0.5706	0.5713	−0.0007	−0.05	0.82	1.00
fexo	40	0.6760	0.6753	+0.0006	+0.01	0.83	1.00
med1	40	0.1287	0.1221	+0.0066	+0.40	0.024	0.27
med2	40	0.1525	0.1512	+0.0013	+0.13	0.33	1.00
osmb	40	0.7142	0.7254	−0.0112	−0.21	0.44	1.00
pdop	40	0.4838	0.4769	+0.0069	+0.13	0.47	1.00
rano	40	0.5393	0.5322	+0.0071	+0.08	0.72	1.00
shop	40	0.4400	0.4406	−0.0005	−0.03	0.82	1.00
siga	40	0.0988	0.1123	−0.0135	−0.14	0.78	1.00
zale	40	0.3438	0.3393	+0.0044	+0.04	0.97	1.00
all tasks	440	0.4226	0.4228	−0.0001	0.00	0.50	–

Table 8: **Per-task means at $N = 30$.** All columns use one matched set of $n = 440$ completed cells, as in Table 1. Task codes as in Table 7. Last row: mean of per-task means. Bold: best per task. Box = $[p_{01}, p_{99}]$. The GP on the shell and the GP in the box are the two DSP rows of Table 2; the Linear BO ports are those of Table 9.

Task	n	GEOWALK	GEOCMA	Random Search (shell)	GP on the shell	GP in the box	Linear BO port (shell)	Linear BO port (box)
adip	40	0.501	0.499	0.498	**0.504**	0.459	0.500	0.474
dhop	40	0.571	0.569	0.568	**0.571**	0.561	0.570	0.560
fexo	40	**0.676**	0.669	0.662	0.675	0.628	0.666	0.628
med1	40	0.129	0.121	0.120	0.122	0.154	0.121	**0.156**
med2	40	0.153	0.152	0.151	0.151	0.150	**0.153**	0.150
osmb	40	0.714	0.719	0.708	**0.725**	0.702	0.719	0.714
pdop	40	**0.484**	0.463	0.469	0.477	0.378	0.475	0.405
rano	40	**0.539**	0.516	0.507	0.532	0.477	0.531	0.502
shop	40	0.440	0.440	0.436	**0.441**	0.427	0.438	0.424
siga	40	0.099	0.076	0.089	0.112	**0.128**	0.098	0.125
zale	40	0.344	**0.349**	0.323	0.339	0.323	0.340	0.349
mean	440	0.423	0.416	0.412	**0.423**	0.399	0.419	0.408

D Per-task convergence on the GuacaMol benchmark

Figure 1 shows the full 30-evaluation convergence traces for each of the 11 non-degenerate GuacaMol tasks, complementing the aggregate cross-task numbers in Table 1. Table 9 aggregates the same runs over tasks. It names its rows as Table 1 and Figure 1 do. The Linear BO port to the shell, which Table 1 does not print (Section 4.1), changes the support, the acquisition (qLogEI instead of LogEI) and its maximization, so we do not read the difference between the port and the published recipe as an effect of the support alone. The kernel bounds of the port were calibrated after a first run (Appendix H). The first 16 proposals of GEOWALK are uniform draws under the same seed as random search, so the two rows of Table 9 agree at $N = 10$ and $N = 20$. The GP on the shell is ahead of GEOWALK at $N = 20$ on all 11 tasks, and GEOWALK catches up only in its last evaluations (Table 9, bottom). In the aggregate, the GP rows on the shell are above the GP rows in the box at each budget of Table 9. These differences were not tested.

Table 9: **Best-so-far after $N = 10$, 20 and 30 evaluations on one matched set.** All rows except the one marked ‡ use the same $n = 440$ completed cells, as in Table 1. The row marked ‡ covers $n = 439$ of them (one run did not complete). Values are means of per-task means of the best score after N evaluations: 5 shared cold-start molecules and $N-5$ proposals. Bold marks the highest mean in the column. Linear BO is the published recipe of Table 15; the two ports are ours (Appendix H). Bottom block: GEOWALK minus the GP on the shell, paired on the same cells. The effect size d is Cohen's d of the cell differences. The column p_{BH} is the two-sided Wilcoxon signed-rank test over the cells. It is Benjamini–Hochberg corrected over the 3 budgets $N = 10$, 20 and 30.

Method	Support	$N = 10$	$N = 20$	$N = 30$
GEOWALK	shell	0.3612	0.3978	0.4226
GP on the shell	shell	0.3673	**0.4067**	**0.4228**
Linear BO port to the shell (calibrated bounds)	shell	**0.3681**	0.4066	0.4192
Random Search (shell)	shell	0.3612	0.3978	0.4118
Linear BO port, box	$[p_{01}, p_{99}]$	0.3499	0.3887	0.4079
Linear BO	box $[-8, 8]^D$	0.3479	0.3907	0.4101
GP in the box	$[p_{01}, p_{99}]$	0.3435	0.3784	0.3989
TuRBO	trust region in $[p_{01}, p_{99}]$	0.3350	0.3607	0.3790
BAxUS‡	nested subspaces of $[p_{01}, p_{99}]$	0.3336	0.3662	0.3907

GEOWALK − *the GP on the shell, paired on the same cells*

	$N = 10$	$N = 20$	$N = 30$
Δ	−0.0062	−0.0090	−0.0001
d	−0.07	−0.13	0.00
p_{BH}	0.27	0.002	0.50
Tasks, GEOWALK higher	2/11	0/11	6/11

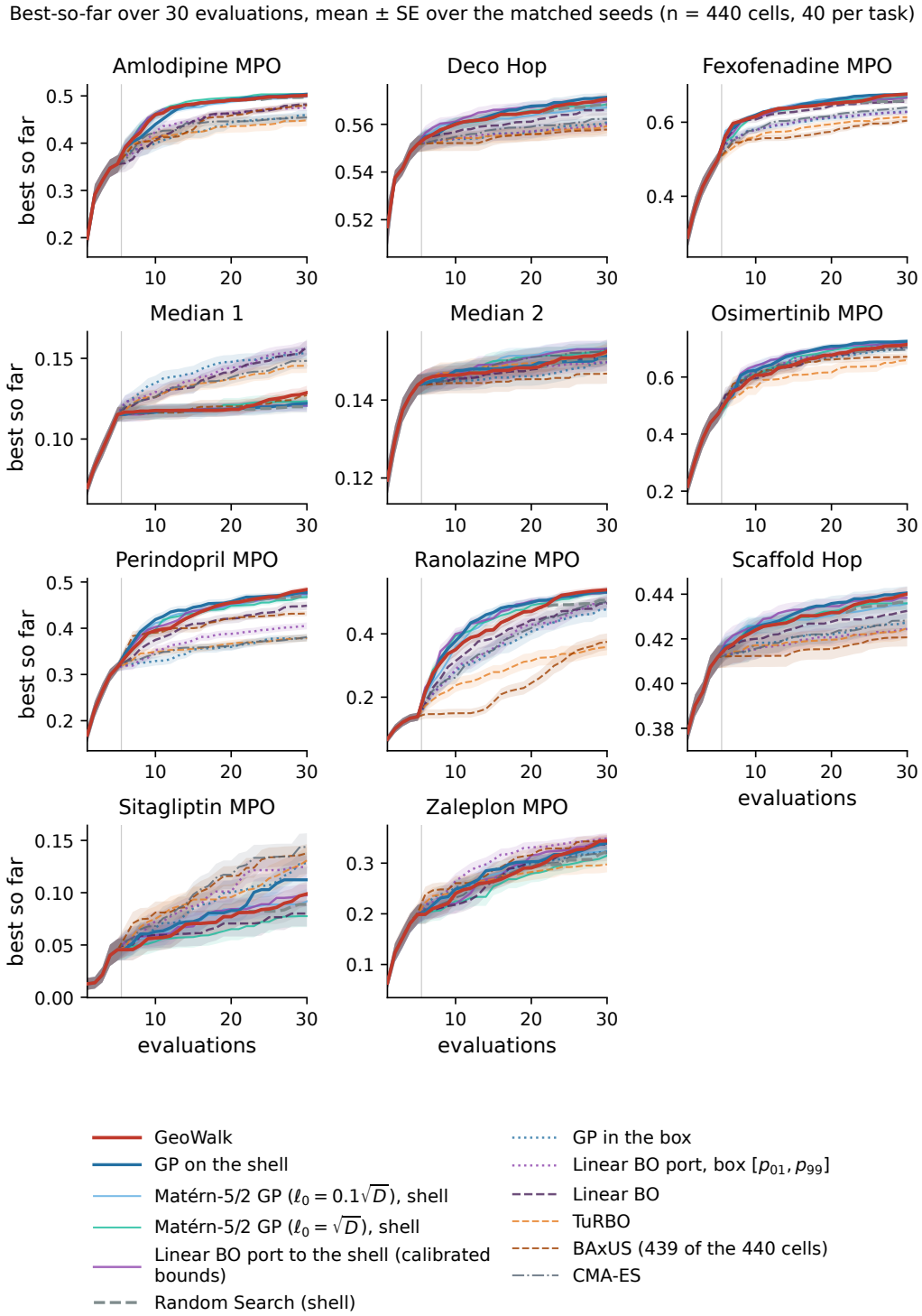


Figure 1: **Thirty-evaluation convergence on the 11 non-degenerate GuacaMol tasks.** Curves show the best score found so far over 30 evaluations, averaged over the seeds of one matched set of 440 cells; bands indicate standard error. The grey vertical line marks the end of the five shared cold-start molecules. The three curves drawn heavier are GeoWalk, the GP on the shell and random search on the shell, which Section 4.2 compares. The legend uses the names of Section 4.2 and Tables 1 and 2. The two ports of Linear BO appear in the appendix only (Tables 8, 9, 12 and 14).

E Scope of the codec comparison

The main experiments use the SELFIES Transformer VAE released with LOL-BO [2]. Stored runs exist for two further molecular codecs: Chembed [41], which is a second pre-trained Transformer VAE, and a Power-Spherical VAE [36]. For each codec we evaluated four methods at the same $N = 30$ budget. Table 10 reports the cross-task means.

Table 10: **Scope of the codec comparison (Appendix E).** The GP on the shell uses candidate-set argmax and the GP in the box L-BFGS-B, so shell versus box also changes the maximizer. Each codec has its own box $[p_{01}, p_{99}]$ in its own latent space. Each codec row uses the matched cells of its four methods (completed cells, as in Table 1). Means are means of per-task means. $\Delta_{\mathrm{shell-box}}$ = the GP on the shell minus the GP in the box. d = Cohen's d of the cell differences. p = two-sided Wilcoxon signed-rank test over the cells, not corrected. The task-level test is given in the text.

Codec	z normalized in decoder; decoding	n	GEOWALK	GEOCMA	GP on the shell	GP in the box	$\Delta_{\mathrm{shell-box}}$	d	p
SELFIES VAE	no, deterministic	440	0.4226	0.4158	0.4228	0.3989	+0.0239	0.30	< 0.001
Chembed (Transformer VAE)	no, deterministic	440	0.3765	0.3734	0.3860	0.3810	+0.0050	0.06	0.26
Power-Spherical VAE	yes, stochastic	440	0.3739	0.3706	0.3689	0.3594	+0.0094	0.11	0.052

The comparison does not extend the main result beyond the SELFIES VAE. The Power-Spherical VAE normalizes its latent inside the decoder and decodes stochastically, so it cannot test where candidates are placed radially. Chembed's encoder means also form a shell (the norm statistics on ZINC-250k are in the results bundle). Even so, on Chembed the GP on the shell is not detectably ahead of the GP in the box: it is ahead on 6 of the 11 tasks (task-level $p = 0.52$). On Chembed the GP on the shell also has a higher mean than GEOWALK. GEOWALK is higher on only 4 of the 11 tasks, and the difference is not significant. A systematic study across more diverse codecs (e.g., diffusion-based or normalizing-flow latent spaces) is left to future work.

F Budget scaling beyond $N = 30$

The main experiments focus on the low-budget regime $N = 30$ motivated by wet-lab cost structures, but it is useful to ask whether the same pattern persists when more evaluations are available. Table 11 reports stored runs with $N = 1000$, read at budgets $N \in \{30, 50, 100, 200, 500, 1000\}$ on the same 11 non-degenerate GuacaMol tasks, using seeds 42–51. The table is descriptive; the goal is not to claim asymptotic dominance. GEOWALK and the GP on the shell remain close across all budgets, with matched differences of at most $+0.008$ for $N \geq 50$ and -0.002 at $N = 30$. GEOCMA and random search on the shell also improve steadily with budget, but remain below GEOWALK and, except for GEOCMA at $N = 100$ and $N = 200$, below the GP on the shell. The box row, a Thompson-sampling GP in the bounding box of the cold-start latents, is below every method on the shell at every budget. It differs from them in more than the support, so these runs carry no claim about the support.

Table 11: **Best-so-far of the stored $N = 1000$ runs at intermediate budgets (seeds 42–51).** One matched set of $n = 109$ of the 110 (task, seed) cells. Each column is the best value after N evaluations of the same runs. The $N = 30$ column is therefore a prefix of these runs, not the $N = 30$ results of Table 1. GP in the cold-start box: a Thompson-sampling GP in the bounding box of the five cold-start latents, not the box version of Table 2. Bold: the highest mean in the column. The Δ rows are paired differences on the matched cells.

Method	$N = 30$	$N = 50$	$N = 100$	$N = 200$	$N = 500$	$N = 1000$
GeoWalk	0.424	**0.445**	**0.463**	**0.479**	**0.497**	**0.509**
GP on the shell	**0.426**	0.443	0.459	0.471	0.489	0.501
GeoCMA	0.414	0.439	0.461	0.474	0.487	0.495
Random Search (shell)	0.409	0.424	0.444	0.460	0.478	0.491
GP in the cold-start box	0.380	0.395	0.420	0.430	0.446	0.458
Δ GeoWalk $-$ GP on the shell	-0.002	$+0.002$	$+0.004$	$+0.008$	$+0.008$	$+0.008$
Δ GeoWalk $-$ GP in the cold-start box	$+0.044$	$+0.050$	$+0.043$	$+0.049$	$+0.051$	$+0.051$

G LassoBench per dimension

Table 12 accompanies the LassoBench discussion in Section 4.4 by showing the per-dimension performance behind the aggregate row in Table 4. The GP rows other than TuRBO and BAxUS maximize the acquisition continuously with L-BFGS-B over the box $[-1, 1]^D$. TuRBO and BAxUS draw a Thompson sample over a finite candidate set inside a trust region. For BAxUS, the trust region lies in its embedded subspace. GeoCMA has a lower normalized loss than GeoWalk at 5 of the six dimensions. It has the lowest loss of all methods only at $D = 7129$. At the other dimensions, the CMA-ES recipe of LassoBench (all D dimensions, started at the origin) has a lower mean loss than GeoCMA. No difference between GeoWalk and a published method is significant after correction within the published methods (Table 13). The runs also reveal a practical limitation of full-dimensional GPs at very low budget. Several GP methods lack seeds at some dimensions, because a model fit failed or, for MSR and MLE (scaled) at $D = 1000$, because runs exceeded the limit of one hour. Both GeoWalk and GeoCMA avoid the full-dimensional GP fitting problem altogether.

Table 12: **LassoBench: final loss divided by the task's random-search mean loss, per dimension.** Each value is the mean over seeds of this ratio after 30 evaluations. A value of 1 equals random search, and a value above 1 is worse than random search. Lower is better. The tasks are $D = 60$ synthetic simple, $D = 100$ synthetic medium, $D = 180$ DNA, $D = 300$ synthetic high, $D = 1000$ synthetic hard, $D = 7129$ leukemia. In parentheses: the seeds with a finite loss ($n/40$ where fewer than the 40 seeds 42–81 have one). n.c.: no seed finished with a finite loss. n.a.: not applicable at this dimension, so not run. The coverage and time-outs of each method, and each method minus GeoCMA, are in the full table in the repository.

Method	$D = 60$	$D = 100$	$D = 180$	$D = 300$	$D = 1000$	$D = 7129$
GeoCMA	0.805 (40)	0.351 (40)	0.891 (40)	0.126 (40)	0.127 (40)	0.672 (40)
GeoWalk	0.566 (40)	0.687 (40)	0.974 (40)	0.748 (40)	0.868 (40)	0.931 (40)
Random search	1.000 (40)	1.000 (40)	1.000 (40)	1.000 (40)	1.000 (40)	1.000 (40)
RBF GP (DSP prior)	2.160 (40)	1.246 (40)	0.968 (40)	0.687 (40)	0.676 (40)	0.869 (40)
Matérn-5/2 GP ($\ell_0 = 0.1\sqrt{D}$)	1.141 (40)	0.793 (40)	0.994 (40)	0.890 (38/40)	0.865 (40)	0.916 (36/40)
Matérn-5/2 GP ($\ell_0 = \sqrt{D}$)	0.758 (40)	0.492 (40)	1.004 (39/40)	0.730 (25/40)	0.711 (38/40)	n.c.
Linear BO port	0.879 (40)	0.243 (40)	0.896 (40)	0.138 (40)	0.133 (40)	n.c.
Published methods						
Vanilla BO	1.710 (40)	0.856 (40)	0.954 (40)	0.754 (40)	0.685 (40)	0.899 (40)
Papenmeier MSR	0.692 (40)	1.011 (40)	0.994 (40)	0.819 (40)	0.717 (17/40)	n.c.
Papenmeier MLE (scaled)	0.616 (40)	0.519 (40)	0.998 (40)	0.834 (40)	0.946 (29/40)	n.c.
Xu Standard GP	0.400 (40)	0.356 (40)	1.016 (40)	0.806 (40)	n.a.	n.a.
Linear BO	0.759 (40)	0.271 (40)	0.898 (40)	0.125 (40)	0.131 (40)	n.c.
TuRBO	1.298 (40)	0.789 (40)	0.962 (40)	0.759 (40)	0.963 (40)	1.148 (40)
BAxUS	0.997 (40)	0.399 (40)	0.873 (40)	0.115 (40)	0.124 (40)	0.709 (40)
CMA-ES (LassoBench recipe)	0.478 (40)	0.287 (40)	0.874 (40)	0.109 (40)	0.125 (40)	0.692 (40)

Table 13: **LassoBench, continued from Table 12: GEOWALK against each published method.** Same runs and normalization as Table 12. These methods form the baseline family of LassoBench. Here a task is a dimension. The difference Δ is GEOWALK minus the method, a mean over tasks of the per-task mean normalized losses. A positive Δ means that GEOWALK has the higher, worse loss. Tasks counts the tasks on which GEOWALK has the lower mean loss, of those where both have runs. The column n gives the (task, seed) pairs that both completed. The column p_{task} is the exact two-sided Wilcoxon test on the per-task mean differences. The column $p_{\mathrm{task}}^{\mathrm{BH}}$ is p_{task} corrected by Benjamini–Hochberg over the 8 methods. Bold marks values below 0.05. With k tasks the exact two-sided test cannot go below $2/2^k$, so a method with fewer than 6 tasks cannot reach 0.05. The mark $^{\|}$ denotes the two arms that ran only on LassoBench; Table 1 prints the same method names under other settings. This family has 8 methods, and the family of Table 15 has 6 recipes.

GEOWALK *against each published method*

Method	Δ (GEOWALK $-$ method)	Tasks GEOWALK lower	n	p_{task}	$p_{\mathrm{task}}^{\mathrm{BH}}$
Vanilla BO	-0.181	3/6	240	1.00	1.00
Papenmeier MSR	-0.077	4/5	177	0.44	0.70
Papenmeier MLE (scaled)	-0.012	4/5	189	0.62	0.71
Xu Standard GP	$+0.099$	2/4	160	0.62	0.71
Linear BO	$+0.332$	1/5	200	0.19	0.44
TuRBO	-0.191	5/6	240	0.094	0.38
BAxUS$^{\|}$	$+0.259$	1/6	240	0.22	0.44
CMA-ES (LassoBench recipe)$^{\|}$	$+0.368$	0/6	240	0.031	0.25

H Baseline implementation details

This appendix summarizes every method of Tables 1 and 12 as it ran. Table 14 gives one row per method: its implementation, surrogate, acquisition and maximizer, and candidate support. The box version of the $0.1\sqrt{D}$ recipe of Table 2 uses the acquisition of MSR, analytic LogEI with sample-around-best restarts (RAASP in Table 14), so that row differs from its shell version in the acquisition as well as in the support and the maximizer. We print that row because it is the published call. A box run of this recipe with qLogEI, which has a higher mean than the printed box row, is in the full table in the repository. Neither row enters the factorial of Table 3, which uses the DSP recipe with the same maximizer on both supports.

The full card of every method is in the repository (`docs/baselines/BASELINE_CARDS.md`). Each card has ten fields: the software version, kernel, acquisition and maximizer, lengthscale initialization, lengthscale prior, candidate support, trust region or RAASP settings, seeds, repository of the published method, and every deviation from it.

Notes to Table 1. All rows are evaluated on one matched set of $n = 440$ (task, seed) cells. It is the intersection of the completed cells of the rows of the full version of the table, except the rows that cover only part of it. A completed cell is a run that made all 25 proposals after the 5 cold starts. In every row printed here, such a run reached all 30 evaluations. Runs in which a surrogate fit failed and a step fell back to a uniform proposal are included. The row marked ‡ covers only part of that set and uses the part it covers (BAxUS: $n = 439$, one run did not complete). Cross-mean = mean of per-task means, and bold marks the highest. The difference Δ is GEOWALK minus the row on the same cells, as a mean of per-task means of the paired differences. W/L counts the cells won and lost by GEOWALK. A cell with $|\mathrm{diff}| \leq 10^{-6}$ is a tie and is excluded from W/L. The effect size d is the mean of the cell differences divided by their standard deviation. The column p_{task} is the exact two-sided Wilcoxon test on the per-task mean differences. It is the primary test of the paper and is printed here without correction. The full version of this table is released with the code (`docs/baselines/TABLE1_FULL.md` and `docs/baselines/tab_main_full.tex`). It has all 31 comparator rows under the names of their implementation, and the cell-level Wilcoxon test, Benjamini–Hochberg corrected over the 31 rows. The rows not printed here are in the full version. Table 2 compares the GP recipes on the box and on the shell. TuRBO, Vanilla BO, Papenmeier MSR, Xu Standard GP and Linear BO are the published recipes, rebuilt from the authors' released code; Table 15 compares them with GEOWALK and the GP on the shell. The other rows adapt the cited method to this benchmark. Their deviations from the published methods are summarized in Table 14 and listed in full in the method cards released with the code (`docs/baselines/`).

The full version names each row by its implementation: CMA-ES is "CMA-ES in $\mathbb{R}^D$, $\sigma_0 = 0.5$, population 5"; BAxUS is "GP in nested count-sketch subspaces, Thompson sampling in a trust region"; REMBO is "GP in a random linear embedding, Thompson sampling"; HeSBO is "GP in a hashed embedding, Thompson sampling"; GaBO is "sphere Gaussian-kernel GP, EI, Riemannian trust-region maximizer"; Riemannian Matérn GP is "Riemannian Matérn GP, EI, Riemannian trust-region maximizer". The GP on the shell of Section 4.2, which Table 2 prints with its box version, is "Matérn-5/2 GP (DSP prior), qLogEI, candidate-set argmax". Box rows use the $[p_{01}, p_{99}]$ box, except Linear BO, which runs in $[-8, 8]^D$. Table 14 gives the support of every row. Abbreviations: DSP prior is the dimension-scaled lengthscale prior $\mathrm{LogNormal}(\sqrt{2} + 0.5 \ln D, \sqrt{3})$ of Hvarfner et al. The value ℓ_0 is the initial lengthscale before the maximum-likelihood fit. EI, LogEI and qLogEI are expected improvement, its logarithm (analytic) and its Monte Carlo batch version. Candidate-set argmax: the acquisition is maximized over a finite set of random candidates on the support. Sample-around-best: L-BFGS-B restarts include points sampled around the best observations. MSR and MLE (scaled) are the two recipes of Papenmeier et al., with maximum-likelihood lengthscales initialized at $0.1\sqrt{D}$. MSR adds RAASP perturbations (random axis-aligned subspace perturbations) in the acquisition maximizer, and MLE (scaled) does not. The $[p_{01}, p_{99}]$ box spans the per-coordinate 1st to 99th percentile of encoded ZINC latents. The radius $\bar{r}$ of the shell is the mean norm of the run's five cold-start latents. The set S^{D-1} is the unit sphere.

Table 14: **Overview of every printed method, one row per method and its variants.** Implementation: where the code comes from. Surrogate: model and fit. Acquisition, maximizer: how a proposal is chosen. Support, trust region: where the candidates are drawn ($\bar{r}$: mean norm of the run's five cold-start latents). Abbreviations: ARD = one lengthscale per input dimension. MAP = maximum a posteriori fit of the hyperparameters. RAASP = random axis-aligned subspace perturbations. qLogNEI = Monte Carlo log noisy expected improvement. UCB = upper confidence bound. PCA sphere = the sphere of radius $\bar{r}$ in the span of the top 64 principal components of 3,000 encoded ZINC latents. The two BAxUS rows run the BoTorch tutorial, not the authors' code, so Table 1 counts that method as an adapted one. Our adapted Matérn-5/2 GP uses the DSP prior with location $\sqrt{2} + 0.5 \ln D$, as in the paper of Hvarfner et al. Their released configuration, which the published recipe follows, uses 1.4 in place of $\sqrt{2}$. GaBO fell back to a uniform draw in $4{,}420$ proposals, in 254 of its 440 runs. The full card of every method and its run status are released with the code (`docs/baselines/`).

Method	Implementation	Surrogate	Acquisition, maximizer	Support, trust region
Methods of this paper (GuacaMol, LassoBench and NAS-Bench-101)				
GeoWalk and its variants	this paper	none	16 uniform draws, then geodesic steps, angle 0.60 to 0.15 rad	shell of radius $\bar{r}$ (NAS: mean embedding norm; LassoBench: clipped)
GeoCMA and its variants	this paper; CMA-ES from pycma	none (CMA-ES)	16-dim. subspace, population 5, step 0.5 (NAS: sd of cold starts)	shell of radius $\bar{r}$ (NAS: whole space, mean norm; LassoBench: box)
Random search, on the shell and on other supports	this paper	none	uniform draws; Gaussian fit: $N(\hat{\mu}, \hat{\Sigma})$ draws; NAS: 30 table entries	shell, PCA sphere, convex combinations of latents, Gaussian; LassoBench: $[-1, 1]^D$
GP recipes adapted in this repository				
Matérn-5/2 GP (DSP prior)	own BoTorch GP; adapts Hvarfner et al. 2024	Matérn-5/2, ARD, DSP lengthscale prior, MAP fit	qLogEI; argmax of 2,048 draws or L-BFGS-B	shell, box $[p_{01}, p_{99}]$, or cube directions at $\bar{r}$
Matérn-5/2 GP ($\ell_0 = 0.1\sqrt{D}$)	own BoTorch GP; adapts Papenmeier et al. 2025	Matérn-5/2, ARD, start $0.1\sqrt{D}$, no lengthscale prior	qLogEI, argmax or L-BFGS-B; one row LogEI with RAASP	shell or box $[p_{01}, p_{99}]$; LassoBench: $[-1, 1]^D$
Matérn-5/2 GP ($\ell_0 = \sqrt{D}$)	own BoTorch GP; adapts Xu et al. 2025	Matérn-5/2, ARD, start $\sqrt{D}$; no lengthscale/ outputscale prior	qLogEI; argmax over 2,048 candidates or L-BFGS-B	shell or box $[p_{01}, p_{99}]$; LassoBench: $[-1, 1]^D$
Linear BO port, box	kernel copied from Doumont et al.	spherical linear kernel, $\mathrm{LogNormal}(\sqrt{2}, \sqrt{3})$ prior	LogEI; L-BFGS-B, 4 restarts, sample-around-best	box $[p_{01}, p_{99}]$ (rescaled); LassoBench: $[-1, 1]^D$
Linear BO port to the shell (calibrated bounds)	kernel copied from Doumont et al.	as the box port; kernel bounds calibrated to the shell	qLogEI; argmax over 2,048 candidates on the shell	shell; the kernel-bound rule was set after a first run on the shell
RBF GP (DSP prior)	own BoTorch GP; adapts Hvarfner et al. 2024	RBF, ARD, LogNormal(1.4 + 0.5 $\ln D$, $\sqrt{3}$), MAP	qLogNEI; L-BFGS-B, 4 restarts, sample-around-best	$[-1, 1]^D$ (LassoBench only)
Published recipes				

Method	Implementation	Surrogate	Acquisition, maximizer	Support, trust region
Vanilla BO	authors' code (Ax 0.3.7) rebuilt on BoTorch	RBF, ARD, LogNormal(1.4 + 0.5 $\ln D$, $\sqrt{3}$), MAP	qLogNEI (in-box baseline); BoTorch, 4 restarts	box $[p_{01}, p_{99}]$; LassoBench: $[-1, 1]^D$
Vanilla BO, all observations as baseline (the other run of the same recipe)	as the row above	as the row above	qLogNEI, all observations as baseline; BoTorch	box $[p_{01}, p_{99}]$
Papenmeier MSR	authors' released code, rebuilt	Matérn-5/2, ARD, start $\sqrt{D}/10$, learned noise	LogEI; 5 restarts, RAASP (sigma 0.001)	box $[p_{01}, p_{99}]$; LassoBench: $[-1, 1]^D$; no trust region
Papenmeier MLE (scaled)	authors' released code, rebuilt	Matérn-5/2, ARD, start $\sqrt{D}/10$, learned noise	LogEI; 5 restarts, no sampling around the best	box $[p_{01}, p_{99}]$; LassoBench: $[-1, 1]^D$
Xu Standard GP	authors' released code, rebuilt	Matérn-5/2, ARD, Uniform(1e-10, 30) prior, start $\sqrt{D}$	UCB (beta 1.5); 10 restarts, 1,000 raw samples	box $[p_{01}, p_{99}]$; LassoBench: $[-1, 1]^D$, not applicable (n.a.) on the two largest tasks
Linear BO	authors' released code; kernel copied	spherical linear kernel, LogNormal($\sqrt{2}$, $\sqrt{3}$) prior	LogEI; L-BFGS-B, 4 restarts, sample-around-best	authors' box $[-8, 8]^{256}$; LassoBench: $[-1, 1]^D$
TuRBO	authors' turbo_1.py and gp.py, ported	Matérn-5/2, ARD, ExactGP trained by Adam (50 steps)	Thompson sampling of the predictive over Sobol candidates	TuRBO-1 trust region in $[p_{01}, p_{99}]$; LassoBench: $[-1, 1]^D$
BAxUS (LassoBench)	BAxUS tutorial of BoTorch (tutorials/baxus)	BoTorch default RBF GP on the embedding	Thompson sampling; argmax of one posterior sample	nested count-sketch embeddings, BAxUS trust region
BAxUS (GuacaMol)	BAxUS tutorial of BoTorch (tutorials/baxus)	BoTorch default RBF GP on the embedding	Thompson sampling; argmax of one posterior sample	GuacaMol: nested count-sketch embeddings of the box $[p_{01}, p_{99}]$, BAxUS trust region
CMA-ES (LassoBench recipe)	LassoBench's CMA-ES example	none (pycma 4.4.4)	CMA-ES, population 20, step size 0.1, $x_0 = 0$	$[-1, 1]^D$ (LassoBench only)
Other baselines implemented in this repository				
REMBO and HeSBO	re-implements REMBO (Wang et al.), HeSBO (Nayebi et al.)	BoTorch default RBF GP on the 16-dim. embedding	Thompson sampling; argmax over 100 points of the box	16-dim. embedding lifted to the shell
GaBO	vendored GaBOtorch (Jaquier et al.) at 884f64a	sphere Gaussian kernel, Gamma(2, 0.15) outputscale prior	EI; Riemannian trust regions; uniform draw on failure	unit sphere S^{255}, proposals scaled to $\bar{r}$
Riemannian Matérn GP	vendored MaternGaBO (Jaquier et al.) at 84bffda	Riemannian Matérn kernel on the sphere, nu = 2.5	EI; robust Riemannian trust regions, 5 restarts	unit sphere S^{255}, proposals scaled to $\bar{r}$
CMA-ES (population 5 or 20)	this repository; CMA-ES from pycma	none (CMA-ES)	CMA-ES in $\mathbb{R}^{256}$, step size 0.5, population 5 or 20	$\mathbb{R}^{256}$, no bounds
GuacaMol stored runs				
GP in the cold-start box ($N = 1000$ runs)	this repository; not the recipe of Hvarfner et al.	Matérn-5/2, ARD, gpytorch defaults, no explicit priors	Thompson sampling; argmax over 2,560 Sobol points	bounding box of the five cold-start encodings

Table 15: **Published recipes against GEOWALK and the GP on the shell at $N = 30$ (baseline family).** All rows use one matched set of $n = 440$ cells. Mean and Δ are means of per-task means of best@30, and Δ = the reference minus the recipe. The reference is GEOWALK in the left group of columns and the GP on the shell in the right group. Tasks = tasks, of the 11, on which the reference has the higher mean. p_{task} = exact two-sided Wilcoxon test on the 11 per-task differences, the primary test. It is corrected by Benjamini–Hochberg within the 12 tests of this table. Bold marks $p^{\text{BH}}_{\text{task}} < 0.05$. The last row is the other run of the Hvarfner recipe, outside the corrected family; Table 14 describes both runs.

		Reference: GEOWALK				Reference: GP on the shell			
Recipe	Mean	Δ	Tasks	p_{task}	$p^{\text{BH}}_{\text{task}}$	Δ	Tasks	p_{task}	$p^{\text{BH}}_{\text{task}}$
Vanilla BO	0.4008	+0.0218	9/11	0.083	0.091	+0.0220	7/11	0.15	0.15
Papenmeier MSR	0.3899	+0.0327	9/11	0.019	0.056	+0.0329	10/11	0.0098	**0.039**
Papenmeier MLE (scaled)	0.3892	+0.0334	9/11	0.0098	**0.039**	+0.0335	10/11	0.0098	**0.039**
Xu Standard GP	0.3951	+0.0275	9/11	0.042	0.063	+0.0277	9/11	0.054	0.064
Linear BO	0.4101	+0.0125	10/11	0.032	0.063	+0.0127	9/11	0.054	0.064
TuRBO	0.3790	+0.0437	9/11	0.032	0.063	+0.0438	8/11	0.042	0.063
Vanilla BO, all observations as baseline (the other run of the same recipe)	0.3993	+0.0233	8/11	0.12	–	+0.0234	8/11	0.10	–

I Shell radius and alternative candidate supports

Table 16 and Figure 2 describe the shell of the encoder means on ZINC-250k. Let Σ and $\bar{\mu}$ be the covariance and the mean of the encoder means, and $m = \operatorname{tr}\Sigma + \|\bar{\mu}\|^2$. The second-order approximation of the mean norm, $\sqrt{m} - \operatorname{Var}(\|\mu\|^2)/(8m^{3/2})$, differs from the measured mean norm by $+0.03\,\%$. It is computed on the same encodings, so it describes the shell rather than predicting it. By default, stereo tokens outside the vocabulary of the VAE are dropped before encoding, and the affected atoms are lost. Every run of the paper uses this default pipeline (Section 3.1). Stripping stereochemistry with RDKit before encoding keeps these atoms, and the mean norm then changes by $-0.10\,\%$.

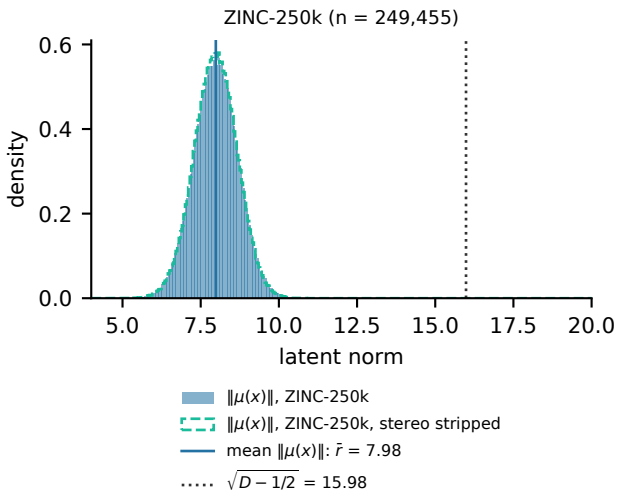


Figure 2: **Norms of the encoder means on ZINC-250k.** Histogram of $\|\mu(x)\|$ for the 249,455 molecules, encoded with the default pipeline (out-of-vocabulary stereo tokens dropped) and with stereochemistry stripped before encoding. The marks show the mean norm over ZINC-250k and $\sqrt{D-1/2} \approx 16$, the typical norm of a sample from the $\mathcal{N}(0, I_D)$ prior with $D = 256$.

Table 16: **Shell geometry of the encoder means $\mu(x)$ of the VAE ($D = 256$), on ZINC-250k.** Σ is the covariance of $\mu(x)$ and $m = \operatorname{tr}\Sigma + \|\bar{\mu}\|^2$, both with the $n-1$ normalization. SD is the standard deviation of $\|\mu\|$, also with $n-1$. The model radius and width are second-order (Gaussian) delta-method values, with $\operatorname{Var}(\|\mu\|^2) = 2\operatorname{tr}(\Sigma^2) + 4\bar{\mu}^\top\Sigma\bar{\mu}$. They are computed on the same encodings, so they describe the shell and do not predict it. Dropped tokens: SELFIES tokens outside the vocabulary of the VAE (stereo tokens) are removed before encoding.

Quantity	ZINC-250k (full set)	ZINC-250k, stereo stripped
Molecules encoded	249,455	249,455
Molecules with dropped tokens (count, share)	157,337 (63.1 %)	5 (0.0 %)
$\bar{r}$ = mean $\|\mu(x)\| \pm$ SD	7.9834 ± 0.709	7.9751 ± 0.692
5th and 95th percentile of $\|\mu\|$	6.81; 9.14	6.83; 9.11
$\operatorname{tr}\Sigma$ and $\|\bar{\mu}\|^2$ (two values)	59.10; 5.13	59.37; 4.71
Participation ratio $R_0 = \operatorname{tr}(\Sigma)^2/\operatorname{tr}(\Sigma^2)$	64.7	65.5
Principal components for 90 % and 99 % of the variance	66; 92	67; 93
Model radius $\sqrt{m} - \operatorname{Var}(\|\mu\|^2)/(8m^{3/2})$ (error vs $\bar{r}$, %)	7.9861 (+0.03)	7.9766 (+0.02)
Model width $\sqrt{\operatorname{Var}(\|\mu\|^2)}/(2\sqrt{m})$	0.678	0.676
Relative width SD/$\bar{r}$ vs $1/\sqrt{2R_0}$	0.0888 vs 0.0879	0.0868 vs 0.0874
$\bar{r}/\sqrt{D-1/2}$	0.499	0.499

Radius, estimator and thickness. The three blocks of Table 17 vary how GEOWALK sets the radius of its proposals. Fixed radii from 6 to 12 show no detected difference from $r = 8$ at the task level. A radius of 4 is ahead on only 3 of the 11 tasks, and a radius of 16 on only 3. Both have lower means, but neither difference is significant at the task level. Setting the radius to the median of the five cold-start norms, instead of their mean, gives a lower mean score. This variant is ahead on only 2 of the 11 tasks (task-level $p = 0.083$). A fixed radius equal to the mean norm over ZINC-250k and the three thick shells show no detected difference from the default.

Supports inside the latent cloud and directions. Every convex sampler and the Gaussian fit of Table 18 has a lower mean than uniform sampling on the shell (the convex samplers draw Dirichlet

Table 17: **Radius of GEOWALK proposals.** Three blocks, each against its own reference row. Fixed radius: every proposal is placed at radius r. The default estimates $\bar{r}$ per run from the five cold-start latents. Radius estimator: the statistic of the five cold-start norms that sets $\bar{r}$. Thick shells: the radius of each proposal is drawn independently from the stated range. The bounds p_{05} and p_{95} are the 5th and 95th percentiles of the encoder-mean norms of ZINC-250k. All blocks use the same $n = 440$ completed cells, as in Table 1. Means are means of per-task means. Bold marks the highest mean of each block. The difference Δ is the row minus the block's reference on the same cells. The effect size d is Cohen's d of the cell differences. The column p_{BH} is the two-sided Wilcoxon signed-rank test over the cells. It is Benjamini–Hochberg corrected within each block: 6 rows for the fixed radius, 2 for the estimator, 3 for the thick shells. Tasks counts the tasks on which the row has the higher mean. The column p_{task} is the exact two-sided Wilcoxon test on the 11 per-task mean differences, not corrected.

Variant	Mean	Δ	d	p_{BH}	Tasks row higher	p_{task}
Fixed radius r (reference $r = 8$)						
$r = 4$	0.4124	−0.0103	−0.19	0.009	3/11	0.24
$r = 6$	0.4187	−0.0040	−0.08	0.20	5/11	0.64
$r = 10$	0.4204	−0.0022	−0.04	0.12	5/11	0.64
$r = 12$	0.4199	−0.0028	−0.05	0.064	5/11	0.70
$r = 16$	0.4117	−0.0110	−0.15	< 0.001	3/11	0.37
per-run $\bar{r}$ (default)	**0.4226**	0.0000	0.00	0.99	5/11	1.00
$r = 8$ (reference)	**0.4227**	–	–	–	–	–
Radius estimator (reference: mean of the five norms)						
median of the five cold-start norms	0.4207	−0.0019	−0.07	0.074	2/11	0.083
fixed 7.98 (ZINC mean norm)	0.4223	−0.0004	−0.01	0.82	5/11	0.52
mean of the five cold-start norms (reference)	**0.4226**	–	–	–	–	–
Thick shells (reference: single shell $\bar{r}$)						
$r \sim \bar{r}\,U[0.9, 1.1]$	0.4183	−0.0043	−0.08	0.099	3/11	0.054
$r \sim \bar{r}\,U[0.8, 1.2]$	0.4195	−0.0032	−0.05	0.13	3/11	0.17
$r \sim U[p_{05}, p_{95}]$	0.4204	−0.0022	−0.04	0.13	4/11	0.21
single shell $\bar{r}$ (default) (reference)	**0.4226**	–	–	–	–	–

combinations of k random latents, not uniform samples of the convex hull), with differences from -0.0209 to -0.0295. With the task as the unit, none of these differences is significant. Averaging, which moves candidates inside the shell, does not explain this gap: the sampler with $k = 1$ draws encoded molecules themselves, whose mean norm equals the shell radius, and it is about as weak as the sampler with $k = 64$. Two controls change the directions at the shell radius (Table 18). Random search restricted to the top-64 principal subspace has a lower mean than random search on the full shell, ahead on only 3 of the 11 tasks (task-level $p = 0.17$). The GP with candidate-set argmax over directions drawn from a cube centered at the origin and rescaled to $\bar{r}$ shows no detected difference from the GP on the shell (task-level $p = 0.90$). These directions come from that cube, not from the per-coordinate box of Section 4.3. These controls suggest that, among the tested alternatives, matching the radial scale is more important than explicitly imposing the observed directional structure.

Radius and clipping on NAS-Bench-101 and LassoBench. On NAS-Bench-101, both methods search a sphere whose radius is the mean norm of the embeddings, and GEOCMA starts with a step size equal to the standard deviation of its projected cold-start points. On LassoBench, GEOWALK uses the mean norm of its cold-start points as the radius and clips its proposals to the box $[-1, 1]^D$. GEOCMA clips its lifted points to the same box without normalizing them, so there it does not search a sphere.

Table 18: **Samplers inside the latent cloud, and direction controls.** First block: surrogate-free samplers against random search, which samples uniformly on the shell. The samplers draw convex combinations of k latents with Dirichlet(1) weights, or draw from a Gaussian fitted to the latents, a sample of 3,000 encoded ZINC-250k molecules. $\overline{\|z\|}$ is the mean norm of the candidates. For random search it is $\bar{r}$, the radius of each run. Second block: two direction controls. Random search on the sphere in the span of the top 64 principal components is compared with random search on the full shell. The GP (DSP prior) with qLogEI argmax over cube directions rescaled to $\bar{r}$ is compared with the same GP on the uniform shell. All rows use the same $n = 440$ completed cells, as in Table 1. Means are means of per-task means. Bold marks the highest mean of the first block. The difference Δ is the row minus its reference on the same cells. The effect size d is Cohen's d of the cell differences. The column p_{BH} is the two-sided Wilcoxon signed-rank test over the cells. It is Benjamini–Hochberg corrected within each block, over 8 rows and 2 comparisons. Tasks counts the tasks on which the row has the higher mean, of the 11. The column p_{task} is the exact two-sided Wilcoxon test on the 11 per-task mean differences, not corrected.

Variant	Mean	$\overline{\|z\|}$	Δ	d	p_{BH}	Tasks row higher	p_{task}
convex, $k = 1$	0.3838	7.98	-0.0280	-0.31	< 0.001	3/11	0.12
convex, $k = 2$	0.3888	6.62	-0.0230	-0.27	< 0.001	4/11	0.24
convex, $k = 3$	0.3909	5.83	-0.0209	-0.24	< 0.001	3/11	0.32
convex, $k = 4$	0.3829	5.32	-0.0289	-0.31	< 0.001	3/11	0.17
convex, $k = 8$	0.3893	4.24	-0.0225	-0.23	< 0.001	4/11	0.37
convex, $k = 16$	0.3855	3.46	-0.0263	-0.26	< 0.001	3/11	0.32
convex, $k = 64$	0.3823	2.62	-0.0295	-0.27	< 0.001	3/11	0.32
Gaussian fit $N(\hat{\mu}, \hat{\Sigma})$	0.3840	7.99	-0.0278	-0.33	< 0.001	3/11	0.12
uniform on the shell (random search) (reference)	**0.4118**	$\bar{r}$	–	–	–	–	–

Direction controls at the shell radius (reference: the same method with uniform directions on the shell)

Comparison	n	Variant	Shell	Δ	d	p_{BH}	Tasks variant higher	p_{task}
Random search: top-64 PCA sphere vs full sphere	440	0.4069	0.4118	-0.0049	-0.08	0.046	3/11	0.17
GP on the shell: cube directions rescaled to $\bar{r}$ vs uniform directions	440	0.4220	0.4228	-0.0008	-0.01	0.91	6/11	0.90

J The five contrasts of the factorial experiment

Table 19 lists the five contrasts of the factorial experiment of Section 4.3: the support effect under each maximizer, the maximizer effect on each support, and their interaction, the difference between the two support effects. Each contrast is the difference of two cross-means of Table 3, tested at the task level and corrected within the five contrasts (Section 3.1). Section 4.3 quotes the task counts of the factorial from this table.

K Compute and assets

Machines. Servers A and B each have two NVIDIA L40S GPUs (48 GB each) and two Intel Xeon Gold 6542Y CPUs. Server A ran the GuacaMol runs other than the stored runs and the two sphere-manifold GPs. Server B ran these two GPs, all LassoBench runs and the checks of the published recipes. All of these runs used only the CPUs of server B. The stored runs were recorded on server B and on server C. Server C has two Intel Xeon Gold 5315Y CPUs, an NVIDIA A100 GPU (80 GB) and an NVIDIA RTX A5000 GPU.

Compute. A table in the repository lists, for each block of runs, its machine, its process-hours and the median time of one run. A process-hour is one hour of wall-clock time as recorded in a result

Table 19: **The five contrasts of the factorial experiment of Section 4.3.** Differences of the cross-means of Table 3 on the same $n = 440$ matched cells. Δ is the mean of per-task means, Tasks counts the tasks with a positive contrast, and $p^{\mathrm{BH}}_{\mathrm{task}}$ is the task-level test, corrected within the five contrasts; bold marks $p^{\mathrm{BH}}_{\mathrm{task}} < 0.05$. The cell-level statistics and the bootstrap intervals are in the full table in the repository.

Contrast	Δ	Tasks $\Delta > 0$	$p^{\mathrm{BH}}_{\mathrm{task}}$
shell minus box, with candidate-set argmax	$+0.0303$	9/11	$\mathbf{0.017}$
shell minus box, with L-BFGS-B	$+0.0010$	6/11	0.90
candidate-set argmax minus L-BFGS-B, on the shell	$+0.0229$	9/11	$\mathbf{0.017}$
candidate-set argmax minus L-BFGS-B, in the box	-0.0064	1/11	$\mathbf{0.012}$
interaction: the first row minus the second	$+0.0294$	9/11	$\mathbf{0.012}$

file, and the totals sum these hours over the runs of a machine. The runs other than the stored runs took about 647 process-hours on server A and about $4{,}372$ on server B. Of the hours on server B, about $4{,}185$ were runs that checked the published recipes; the outcome of each check is in the results bundle, with the method cards. The stored runs that the printed tables read took about $1{,}289$ process-hours. This total counts only the runs whose result files record a time. The project also used compute for preliminary and failed runs that the paper does not report.

Assets and licenses. Every external asset is cited where it is used. The list below gives the license of each asset and its version where it has one. The third-party notices of the repository give the commit of every asset and list every third-party file with its license.

- *Molecules and models.* The SELFIES [4] VAE and code released with LOL-BO [2, 30] are MIT. GuacaMol [5] 0.5.2 is MIT; its data are CC BY 4.0, derived from ChEMBL under CC BY-SA 3.0. ZINC-250k [6], in the subset of Gómez-Bombarelli et al. [7], is free to use without a formal license; the file comes from the Apache-2.0 `chemical_vae` repository. The code of Chembed [41] is MIT. The weights of Chembed declare no license, and our repository does not redistribute them. We trained the Power-Spherical VAE of Appendix E with the `power_spherical` package of De Cao and Aziz [36], which is MIT.
- *Benchmarks.* LassoBench [8] is MIT for the code and the synthetic tasks; its real-world DNA and leukemia data fall under the terms of the LIBSVM website. NAS-Bench-101 [9] and arch2vec [40] are Apache-2.0.
- *Software.* PyTorch [42] 2.11.0, pycma [20] 4.4.4 and RDKit [43] 2026.03.1 are BSD-3. BoTorch [37] 0.17.2 and GPyTorch [44] 1.15.2 are MIT, and selfies 2.2.0 is Apache-2.0. The released code of Linear BO [21], GaBO [22] and the Riemannian Matérn GP [23] is MIT.
- *Ported code.* Our port of the official TuRBO code [1] (`https://github.com/uber-research/TuRBO`) is used for every TuRBO row of the paper and for the TuRBO curve of Figure 1. That code is under the TuRBO license, which is non-commercial, so the port is distributed under the same license, with the change notice it requires. The BAxUS row of Table 12 ports the BoTorch BAxUS tutorial, which is MIT; this port shares its file with the TuRBO port and is distributed with it under the TuRBO license.
- *Our code* is released under the MIT license, except the file with the two ports above.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims are supported in Section 4: Table 1 compares the methods on one matched set (its full version, with every row and the cell-level statistics, is in the repository), Table 2 gives the three recipe rows, Table 3 the factorial experiment whose contrasts (Table 19, Appendix J) carry the central claim, and Table 4 compares GEOWALK and GEOCMA across benchmarks; Appendices A–J give the details. The primary test is the exact task-level Wilcoxon test over the 11 tasks, with Benjamini–Hochberg correction within families. The abstract and conclusion state that this is a controlled study of a Gaussian-prior VAE benchmark, not a recommendation to use VAEs for molecular optimization; Section 3.1 and the Limitations restrict it to $N = 30$ and the 11 GuacaMol tasks of the protocol.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The Limitations paragraph of Section 5 states that the advantage of the shell is shown only for candidate sets drawn uniformly from the whole support; that the evidence comes only from the Gaussian-prior VAE benchmark studied and the 11 tasks of the protocol; that on nine further tasks the shell was ahead on three; that a second VAE (Chembed) gave no detected gain; that budgets above $N = 30$ come from stored runs; and that flows, diffusion models and language-model spaces were not tested.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.

- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper is empirical and presents no theorems. The shell statistics of Section 3.2 and Appendix I (mean norm, effective rank, a second-order description of the radius by two moments) are measured on the encoder means of the ZINC-250k reference set and describe the shell on the same encodings; they are not derived as a guarantee.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Algorithms 1 and 2 give complete pseudocode for GEOWALK and GEOCMA, with their constants in Sections 3.3 and 3.4. Section 3.1 specifies the tasks, seeds, cold-start protocol, budget accounting and statistical protocol, and Section 3.2 defines the acquisition, both maximizers, the box and the GP constructor. Appendix H summarizes every baseline as it ran; the method cards in the repository give every deviation from the published method. The code, the run configurations, the result files and the generator of every table, figure and number are released ("Code and data availability", Section 5). The SELFIES VAE weights are the public release accompanying LOL-BO [2].

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data

is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The code, the run configurations and the result files are released in the public repository given in the "Code and data availability" paragraph of Section 5, with instructions and the generator that writes every table, figure and number from the result files. GuacaMol, LassoBench, NAS-Bench-101 and the SELFIES VAE released with LOL-BO are public. ZINC-250k is the public subset distributed by Gómez-Bombarelli et al. [7]; it serves here as a reference set, not as the VAE's training set.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: Section 3.1 specifies the tasks, seeds, cold-start protocol and statistical procedure, and Section 3.2 the acquisition, the maximizers, the box and the GP. Sections 3.3 and 3.4 and Table 5 give the settings of GEOWALK and GEOCMA, and Appendix B their sensitivity. Baselines are adaptations of the published methods run in our harness, summarized in Appendix H, with every known deviation listed in the method cards of the repository; the factorial experiment is described in Section 4.3.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: The primary test of every comparison is the exact two-sided Wilcoxon signed-rank test on the 11 per-task mean differences, reported with the number of tasks on which a method is ahead (Section 3.1). Benjamini–Hochberg correction is applied within families: the five factorial contrasts, the three recipe rows of Table 2, the published-recipe tests of Table 15, and the LassoBench tests of the published methods; the task-level p of Tables 1 and 4 is uncorrected. The full version of Table 19 in the repository also reports a percentile bootstrap interval over tasks for each contrast. Cell-level Wilcoxon tests and Cohen's d are secondary, and every table prints the size of its matched set. Bands in Figure 1 are standard errors over seeds.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Appendix K names the machines and gives the total process-hours; a table in the repository gives the process-hours and the median time of one run for every block of runs, recomputed from the times recorded in the result files by a script of the repository. The runs other than the stored runs took about 647 process-hours on a server with two NVIDIA L40S GPUs and about 4,372 on the CPUs of a second server, most of them for checks of the published recipes. The stored runs that printed tables read took about 1,289 process-hours. The project also used compute for preliminary and failed runs that the paper does not report.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: The research uses publicly available benchmarks (GuacaMol, LassoBench, NAS-Bench-101) and does not involve human subjects, private data, or personally identifiable information. Dual-use risks of molecular optimization are acknowledged in Section 5.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: The Limitations paragraph of Section 5 notes that better low-budget molecular optimization could lower experimental cost (positive) and that it has dual-use risks if applied to harmful objectives (negative).

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper releases optimization code, run configurations and result files only; the underlying SELFIES VAE and oracle implementations are pre-existing public assets released by their original authors. No new generative model, pre-trained language model, or large-scale scraped dataset is released.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Every external asset is cited, and Appendix K gives its license and version. Our port of the official TuRBO code [1] (`https://github.com/uber-research/TuRBO`) is distributed under the non-commercial TuRBO license, with the change notice it requires. Third-party files keep their own licenses, listed in the repository's third-party notices.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: GeoWalk, GeoCMA, the experiment scripts, the run configurations and the result files are released in the public repository given in the "Code and data availability" paragraph of Section 5. It contains installation instructions, the generator that reproduces every table, figure and number of the paper, the full card and the run status of every method, a license file (MIT for the authors' code), and the notices of the third-party and ported files under their own licenses.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This research does not involve crowdsourcing or human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This research does not involve human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The proposed methods (GEOWALK and GEOCMA) contain no LLM component. As declared on OpenReview, LLMs were used for data processing and filtering, implementing standard methods, editing, and drafting sections of the paper; this does not affect the core methodology.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.