Bayesian Optimization with Fisher Information Geometry: Gradient Bounds and Trust-Region Methods

Saksham Kiroriwal[1], Julius Pfrommer[1], and Jürgen Beyerer[1,2]

[1]Cognitive Industrial Systems, Fraunhofer IOSB, Karlsruhe, Germany
[2]Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany
{saksham.kiroriwal, julius.pfrommer, juergen.beyerer}@iosb.fraunhofer.de

Abstract

We study Bayesian optimization (BO) through the lens of information geometry. Pulling back the Fisher information metric through the surrogate posterior map yields a local sensitivity tensor on the input space, which leads to an upper bound on the gradient of reparameterizable acquisition functions (AFs). This view explains vanishing-gradient behavior in high-dimensional BO and provides a common interpretation of heuristics such as RAASP and dimension-scaled lengthscales. The upper bound can also serve as a generic recipe to understand the behavior of new acquisition functions or surrogates in high input dimensions as well. Using the bounds, we also identify necessary conditions for an input location to be an optimizer of the AF for a univariate Gaussian predictive posterior. Building on this analysis, we propose FITR, a trust-region-based BO method that replaces lengthscale-based scaling with local pullback-Fisher weights. FITR is not restricted to GP kernels with explicit lengthscales and easily generalizes to non-isotropic surrogates. FITR shows competitive performance on several benchmarks.

1 Introduction

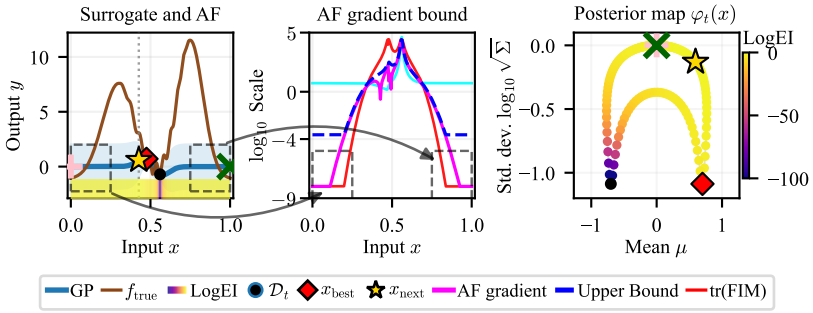

Figure 1: Left: GP surrogate (SE kernel) and LogEI [1] acquisition function. Middle: visualization of the upper bound for the acquisition function gradient from Proposition 4.2. Right: induced geometry on the input space by the posterior map in the statistical manifold defined by predictive mean and standard deviation. The color bar represents the same LogEI as in the left panel. Distant inputs can have similar predictions. Gray regions contain no observations and have a small pullback trace.

Bayesian Optimization (BO) is a popular framework for optimizing black-box functions with expensive evaluations [2]. It has proven to be effective in hyperparameter tuning [3], robotics policy

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

search [4], and engineering design [5]. As explained in Section 2, BO alternates between fitting a probabilistic surrogate $\mathcal{M}_t$ to the observed data $\mathcal{D}_t$ and optimizing an acquisition function (AF) $\alpha(\cdot)$ to select the next input candidate $\mathbf{x}_{t+1}$. In practice, the inner AF optimization is often non-trivial. The AF is usually non-convex, depends on the surrogate through a predictive distribution, and becomes increasingly difficult to optimize as the input dimension D_x grows.

This issue is now well documented in the high-dimensional BO (HDBO) literature. Recent work shows that both surrogate training and AF optimization can suffer from vanishing gradients in large dimensions [6–8]. Proposed fixes include dimensionally scaled lengthscale priors [6, 7], Random Axis Aligned Subspace Perturbations (RAASP) initializations [9], conformal mappings [10], and trust-region-based search [11]. A central question remains: what common mechanism makes AF optimization collapse, and how much of that mechanism is tied to the surrogate model?

Inspired by the work of Herrmann et al. [12], we analyze BO from a different perspective. As shown in Figure 1, instead of looking at the AF only as a function over $\mathcal{X}$, we utilize the chain rule and information geometry. The surrogate $\mathcal{M}_t$ is a map from input space $\mathcal{X}$ to the statistical manifold $\mathcal{S}$ of predictive distributions. These predictive distribution parameters, like mean and variance, are the parameters that the AF really depends on. Information geometry [13, 14] becomes a natural way to study how the surrogate induces geometry on the input space through the pullback Fisher tensor. This induced geometry tells us how strongly local perturbations in $\mathbf{x}$ change the predictive distribution.

This view separates two effects that are usually entangled in AF optimization. The acquisition design determines how sensitive the utility is to changes in the predictive distribution. The surrogate-induced geometry determines whether moving in input space produces such changes at all. Section 4 formalizes this separation through an acquisition-gradient bound. This perspective then becomes a common analytical tool for understanding HDBO:

- In Section 4, we analyze the pullback geometry induced by the surrogate posterior map and derive a bound for the AF gradient norm $|\nabla_{\mathbf{x}}\alpha|_2$ that separates acquisition-side sensitivity from surrogate-induced geometry.
- We use this factorization to analyze vanishing-gradient behavior in HDBO under a common framework. This yields a unified interpretation of dimensionally scaled priors [6, 7] and RAASP [8, 9].
- In Section 5, we propose the Fisher-Information Trust Region (FITR), which replaces kernel-lengthscale scaling by local pullback-Fisher weights. FITR applies to differentiable probabilistic surrogates that satisfy the stated regularity conditions and is not restricted to GP kernels with explicit lengthscales. Section 6 shows that FITR improves on TuRBO-REI on several benchmarks and remains competitive with strong baselines on the remainder.

2 Preliminaries

The aim of BO is to maximize a black-box function f_{true} over a feasible set $\mathcal{X} \subset \mathbb{R}^{D_x}$. At each BO iteration t, we assume that we have observed data $\mathcal{D}_t = \{(\mathbf{x}_n, y_n)\}_{n=1}^N$, where input $\mathbf{x}_n \in \mathcal{X}$ and output $y_n = f_{\text{true}}(\mathbf{x}_n) + \varepsilon_n$. We allow $y_n \in \mathbb{R}^{D_y}$ to be multi-dimensional and assume $\varepsilon_n \sim \mathcal{N}(0, \sigma_\varepsilon^2 I_{D_y})$, which reduces to $\mathcal{N}(0, \sigma_\varepsilon^2)$ in the scalar-output case. BO alternates between fitting a probabilistic surrogate $\mathcal{M}_t$ and maximizing an AF via $\arg\max_{\mathbf{x}\in\mathcal{X}} \alpha(\mathbf{x}; \mathcal{D}_t)$ to obtain the next input candidate $\mathbf{x}_{t+1}$ [15, 16]. We then evaluate $y_{t+1} = f_{\text{true}}(\mathbf{x}_{t+1})$ and update the observed data $\mathcal{D}_{t+1} = \mathcal{D}_t \cup \{(\mathbf{x}_{t+1}, y_{t+1})\}$. We repeat until a stopping criterion is met.

Surrogate Models At each iteration t, the surrogate model $\mathcal{M}_t$ models the predictive distribution $p(y|\mathbf{x}, \mathcal{D}_t) = p(y; \boldsymbol{\theta}_{t,\mathbf{x}})$, where $\boldsymbol{\theta}_{t,\mathbf{x}} \in \mathcal{S}$ are the parameters of the predictive distribution [17]. The hyperparameters $\boldsymbol{\lambda}_t$ of $\mathcal{M}_t$ specify the distribution parameters $\boldsymbol{\theta}_{t,\mathbf{x}} \leftarrow \mathcal{M}_t(\mathbf{x}; \mathcal{D}_t)$ and can be trained by maximizing the marginal log-likelihood [18] or another suitable objective [19, 20].

As shown in the manifold view in Figure 1, the surrogate model can also be interpreted as a smooth map from inputs to a family of probability distributions. Each input $\mathbf{x} \in \mathcal{X}$ is mapped to a point $\boldsymbol{\theta}_{t,\mathbf{x}}$ on the statistical manifold $\mathcal{S}$, the set of all distributions the surrogate can produce [13]. Input locations that are far apart in Euclidean space can map to nearby predictive parameters because the surrogate gives similar predictions there. This map is at least C^1 differentiable, a mild condition satisfied by many popular surrogate models such as Gaussian processes (GPs) with differentiable

kernels [18] or Bayesian neural networks (BNNs) with smooth activations [21]. The *posterior map* φ_t induced by $\mathcal{M}_t$ can be defined as

$$\varphi_t : \mathcal{X} \to \mathcal{S}, \quad \mathbf{x} \mapsto \boldsymbol{\theta}_{t,\mathbf{x}}, \text{ such that } \boldsymbol{\theta}_{t,\mathbf{x}} \in \mathcal{S}, \tag{1}$$

A common choice for the surrogate model is a Gaussian process (GP). For a scalar output $y \in \mathbb{R}$ at an input location $\mathbf{x} \in \mathcal{X}$, $\mathcal{M}_t$ models $\boldsymbol{\theta}_{t,\mathbf{x}} = (\mu_{t,\mathbf{x}}, \Sigma_{t,\mathbf{x}})$ and gives $p(y|\mathbf{x}, \mathcal{D}_t) = \mathcal{N}(y; \mu_{t,\mathbf{x}}, \Sigma_{t,\mathbf{x}})$ with $\mathcal{S} = \mathbb{R} \times \mathbb{R}_+$ [18]. GPs are discussed in further detail in Appendix A.

Acquisition Function (AF) The AF α is the expected utility over the predictive posterior evaluated at a candidate location $\mathbf{x} \in \mathcal{X}$ and can be defined as $\alpha(\mathbf{x}) = \mathbb{E}_y\left[a(y; \boldsymbol{\theta}_{t,\mathbf{x}}, \boldsymbol{\xi})\right]$, where a is the associated utility function for the AF and $\boldsymbol{\xi}$ are respective acquisition parameters. The next query is $\mathbf{x}_{t+1} = \arg\max_{\mathbf{x}} \alpha(\mathbf{x}; \mathcal{D}_t)$ [16, 17, 22]. Since $p(y|\mathbf{x}, \mathcal{D}_t)$ can be parameterized by $\boldsymbol{\theta}_{t,\mathbf{x}}$, α depends on $\mathbf{x}$ only through $\boldsymbol{\theta}_{t,\mathbf{x}}$. Using the reparameterization trick [23], we introduce a common notation to represent both closed-form and Monte Carlo AFs as

$$\alpha(\mathbf{x}) = \mathbb{E}_{\boldsymbol{\zeta}}[\bar{\alpha}(\boldsymbol{\theta}_{t,\mathbf{x}}, \boldsymbol{\zeta}; \boldsymbol{\xi})]\,, \tag{2}$$

where $\boldsymbol{\zeta}$ is base noise drawn independently of $\mathbf{x}$ [17, 23]. However for closed-form AFs, $\bar{\alpha}$ is independent of $\boldsymbol{\zeta}$ and (2) can be trivially written as $\alpha(\mathbf{x}) = \bar{\alpha}(\boldsymbol{\theta}_{t,\mathbf{x}}; \boldsymbol{\xi})$. Due to this, the closed-form AF and the $\boldsymbol{\zeta}$-dependent integrand used in the MC form can, in general, be different formulations of $\boldsymbol{\theta}_{t,\mathbf{x}}$. Appendix B discusses this in detail and lists both forms for different AFs.

Fisher Information Matrix For a parametric distribution family $p(y; \boldsymbol{\theta})$ with $\boldsymbol{\theta} \in \mathcal{S}$, the *Fisher information matrix* (FIM) $\mathcal{I}_{\mathcal{S}}(\boldsymbol{\theta}) \in \mathbb{R}^{D_\theta \times D_\theta}$ is the expected outer product of $\nabla_{\boldsymbol{\theta}} \log p(y; \boldsymbol{\theta})$ as

$$\left[\mathcal{I}_{\mathcal{S}}(\boldsymbol{\theta})\right]_{ij} = \mathbb{E}_{p(y;\,\boldsymbol{\theta})}\left[\frac{\partial \log p(y; \boldsymbol{\theta})}{\partial \theta_i}\,\frac{\partial \log p(y; \boldsymbol{\theta})}{\partial \theta_j}\right]. \tag{3}$$

Amari and Nagaoka [24] showed that the FIM induces a Riemannian metric, known as the *Fisher–Rao metric*, on the statistical manifold $\mathcal{S}$. The Fisher–Rao metric explains the local geometry of the manifold [13, 25]. It quantifies the sensitivity of the predictive distribution $p(y; \boldsymbol{\theta})$ to changes in $\boldsymbol{\theta}$.

Riemannian Geometry A *Riemannian manifold* $(\mathcal{M}, G)$ is a smooth manifold $\mathcal{M}$ equipped with a Riemannian metric G, whose pointwise evaluation G_p defines a positive definite inner product on each tangent space $T_p\mathcal{M} \simeq \mathbb{R}^{d_p}$ [14]. The geodesic distance between $p_1, p_2 \in \mathcal{M}$ is the minimum path length on the manifold that connects the two points and is given by

$$d_G(p_1, p_2) = \inf_{\gamma} \int_0^1 \sqrt{G_{\gamma(t)}(\dot{\gamma}(t),\, \dot{\gamma}(t))}\,\mathrm{d}t, \tag{4}$$

where the infimum is over smooth curves $\gamma : [0, 1] \to \mathcal{M}$ with $\gamma(0) = p_1$ and $\gamma(1) = p_2$ [14]. The Fisher–Rao metric $\mathcal{I}_{\mathcal{S}}(\boldsymbol{\theta})$ induces a Riemannian structure on the statistical manifold $\mathcal{S}$ [13]. We define the pullback Fisher tensor through the posterior map φ_t and discuss its properties in Section 4.

3 Related Work

Information geometry and applications. Natural gradient descent (NGD) rescales the ordinary gradient by the inverse Fisher metric to use the local sensitivity [25, 26]. Work by Khan and Rue [27] shows how algorithms such as stochastic gradient descent (SGD) and Kalman filters can be derived from an algorithm called *Bayesian Learning Rule*, and how natural gradients can improve the resulting updates. As another application of information geometry, Kim et al. [28] presented Fisher-Sharpness-Aware Minimization (FSAM), which replaced the Euclidean ball in sharpness-aware minimization with a Fisher ellipsoid, improving generalization and robustness. Beik-Mohammadi et al. [29] learn Riemannian manifolds for geodesic motion generation in robotics.

Geometry-aware Bayesian optimization. Jaquier et al. [4], Jaquier and Rozo [30] introduced geometry-aware kernels that used geodesic distances in Bayesian optimization when the search space is non-Euclidean, such as robot orientation. However, in our work, we explore the geometry that is induced in the search space by the posterior map of the surrogate model $\mathcal{M}_t$ as mentioned in (1). We do not assume that the search space has any intrinsic geometric structure. Recent work by Yu et al.

[31] proposes GIT-BO with Tabular Foundation Model (TabPFNv2) as the surrogate model. They use the gradient of the predictive mean to identify a global active subspace for high-dimensional BO. Our use of Fisher geometry is local and trust-region based: we shape candidate regions around the current center using the predictive distribution, not only the mean. A direct comparison with GIT-BO would therefore mix surrogate choice with trust-region geometry.

Pullback metrics for probabilistic models. Tosi et al. [32] proposed the idea of pulling back a Riemannian metric for Gaussian Process Latent Variable Models (GPLVMs). They derived the expected metric tensor and used the resulting geodesics for improving data interpolation. Arvanitidis et al. [33] showed how the latent space can be characterized by a stochastic Riemannian metric, resulting in improved sampling and interpolation in latent space. Arvanitidis et al. [34] used the Fisher-Rao metric associated with the space of decoder distributions, enabling meaningful latent geometries for a broader class of decoders. Rozo et al. [35] generalized the construction to wrapped GPLVMs defined on Riemannian data manifolds, where they pull back a Riemannian structure from the data manifold to the latent space.

High-dimensional Bayesian optimization. Recent work by Hvarfner et al. [6], Xu et al. [7] has shown that BO can be effective in high dimensions (HDBO). They analyze the vanishing-gradient problem in surrogate-model and acquisition-function (AF) optimization in high dimensions. They propose a dimensionally scaled lengthscale prior to avoid the vanishing gradient problem. Papenmeier et al. [8] confirm the finding and propose $\sqrt{D_x}/10$ initialization of lengthscales for Maximum Likelihood Estimation (MLE) for GP surrogates. They also show that RAASP for AF optimization helps in preventing vanishing gradients.

Local search for global optimization and trust-region methods offer complementary approaches. TuRBO is a popular trust-region-based method proposed by Eriksson et al. [11] to address high dimensionality. They use local axis-aligned hyperrectangles to define the trust region around the current best point. Work by Papenmeier et al. [9, 36] extends trust-region-based methods to combinatorial and mixed spaces as well. However, existing trust-region constructions in this setting rely on GP surrogates with explicit lengthscale parameters like the squared exponential kernel or Matern kernel.

4 Manifold View of Bayesian Optimization

As mentioned in Section 2, the acquisition function depends on $\mathbf{x}$ through the surrogate's predictive parameters $\boldsymbol{\theta}_{t,\mathbf{x}}$. Viewing the surrogate as a smooth map $\varphi_t : \mathcal{X} \to \mathcal{S}$ into the statistical manifold of predictive distributions shifts the analysis of acquisition-gradient behavior to the geometry of this map. Pulling back the Fisher–Rao metric from $\mathcal{S}$ to $\mathcal{X}$ via φ_t yields the pullback Fisher tensor $\mathbf{G}_t(\mathbf{x})$, which quantifies how sensitively $p(y; \boldsymbol{\theta}_{t,\mathbf{x}})$ changes under local perturbations of the input.

We define the pullback Fisher tensor and establish an acquisition gradient bound (Section 4.1). Then in Section 4.2 we show how the critical λ scaling of $\sqrt{D_x}$ from [6–8] and other HDBO findings like RAASP can be viewed from the geometric perspective of the pullback Fisher tensor.

Assumption 4.1 (Surrogate Differentiability and Positive Fisher Metric)**.** *φ_t is C^1 on $\mathcal{X}$ (satisfied by GPs with differentiable kernels and by BNNs with smooth activations) and $\mathcal{I}_{\mathcal{S}}(\varphi_t(\mathbf{x})) \succ 0$ for all $\mathbf{x} \in \mathcal{X}$ (a minimum noise $\sigma_\varepsilon^2 > 0$ ensures this for GP regression).*

As discussed in Appendix D.1, the first condition makes φ_t differentiable so that the Jacobian J_t exists. The second ensures $\mathcal{I}_{\mathcal{S}}$ is a valid inner product on $\mathcal{S}$, so its inverse (required in the acquisition sensitivity (7)) is well-defined. The pullback Fisher tensor quantifies the sensitivity of the predictive distribution to perturbations in the input space at each input location.

Definition 4.1 (Pullback Fisher Tensor)**.** *Under Assumption 4.1, the pullback Fisher tensor induced on the input space $\mathcal{X}$ by the predictive distribution manifold $\mathcal{S}$ through the posterior map φ_t is*

$$\mathbf{G}_t(\mathbf{x}) = J_t(\mathbf{x})^\top \, \mathcal{I}_{\mathcal{S}}\big(\varphi_t(\mathbf{x})\big) \, J_t(\mathbf{x}) \;\in\; \mathbb{R}^{D_x \times D_x}, \tag{5}$$

where $\mathbf{J}_t(\mathbf{x}) = \partial \boldsymbol{\theta}_{t,\mathbf{x}} / \partial \mathbf{x} \in \mathbb{R}^{D_\theta \times D_x}$ is the Jacobian of the posterior map.

Figure 1 shows the induced geometry on a 1D Euclidean input space by the GP surrogate with SE kernel and LogEI AF using the pullback metric tensor. Figure 2 similarly shows the trace

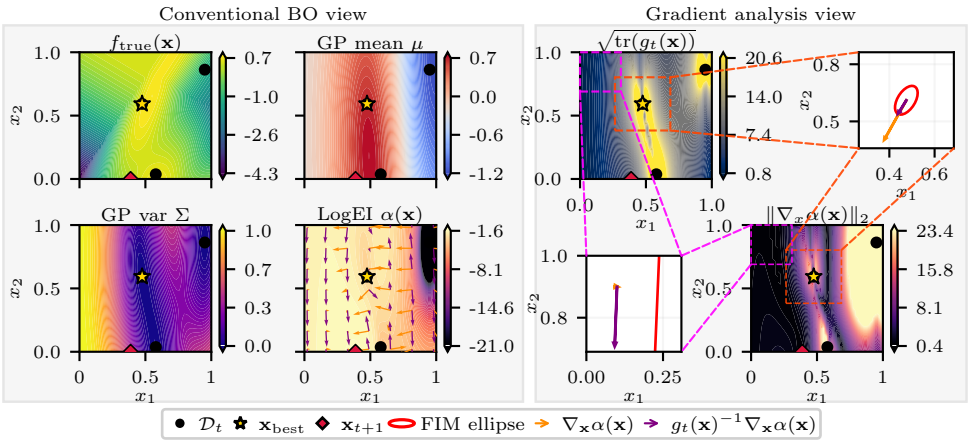

Figure 2: Conventional view of BO with a GP surrogate (SE kernel) and LogEI AF. The LogEI plot also shows the gradient (orange) and natural gradient (purple) vector field in the input space. The "gradient analysis" shows pullback metric trace $\sqrt{\text{tr}(\mathbf{G}_t(\mathbf{x}))}$ and AF gradient norm $\|\nabla_{\mathbf{x}}\alpha(\mathbf{x})\|_2$. Trace is non-degenerate near the observed data and collapses in regions where the GP posterior has reverted to the prior (no observations). The red ellipses are calculated using the pullback metric tensor at respective input locations.

of the pullback metric tensor for a 2D input space. Since for $D_x > 1$, the pullback metric is a tensor, the eigenvalues of $\mathbf{G}_t(\mathbf{x})$ reflect how strongly the predictive distribution changes with respect to perturbations in input space at $\mathbf{x}$ in respective directions. Large eigenvalues correspond to directions (not necessarily axes-aligned with $\mathcal{X}$) along which the distribution changes rapidly, small or zero eigenvalues to directions that are locally uninformative. This effect can be seen in the "Conventional BO view" panel of Figure 2, where we show the gradient $\nabla_{\mathbf{x}}\alpha(\mathbf{x})$ and the natural gradient $\mathbf{G}_t(\mathbf{x})^{-1}\nabla_{\mathbf{x}}\alpha(\mathbf{x})$ (where the inverse exists) vector field and also zoomed in view for two input locations denoted by the magenta and red boxes.

The pullback tensor $\mathbf{G}_t(\mathbf{x})$ is a Riemannian metric only when it is positive definite. Its rank is at most D_θ (at most 2 for a scalar Gaussian predictive), so the induced geometry is degenerate when $D_x > D_\theta$. Locally, perturbations at $\mathbf{x}$ in null directions leave the predictive parameters $\boldsymbol{\theta}_{\mathbf{x}}$ unchanged.

4.1 Acquisition Gradient Bound via Pullback Fisher Tensor

Proposition 4.2 (Acquisition Gradient Bound)**.** *Let Assumption 4.1 hold. For any $\mathbf{x} \in \mathcal{X}$, let $\boldsymbol{\theta}_{t,\mathbf{x}} = \varphi_t(\mathbf{x})$ and suppose the acquisition admits the representation $\alpha(\mathbf{x}) = \mathbb{E}_{\boldsymbol{\zeta}}[\bar{\alpha}(\boldsymbol{\theta}_{t,\mathbf{x}}, \boldsymbol{\zeta})]$, where the interchange of gradient and expectation is valid, for example under the scalar Gaussian reparameterization conditions of Appendix D.1, and $\mathbb{E}_{\boldsymbol{\zeta}}[\|\nabla_{\boldsymbol{\theta}_{t,\mathbf{x}}}\bar{\alpha}\|^2] < \infty$. Then*

$$\left\|\nabla_{\mathbf{x}}\,\alpha(\mathbf{x})\right\|_2^2 \;\le\; C_\alpha\big(\boldsymbol{\theta}_{t,\mathbf{x}}\big)\cdot \text{tr}\big(\mathbf{G}_t(\mathbf{x})\big)\,, \tag{6}$$

where the acquisition sensitivity *is*

$$C_\alpha(\boldsymbol{\theta}_{t,\mathbf{x}}) = \mathbb{E}_{\boldsymbol{\zeta}}\Big[\nabla_{\boldsymbol{\theta}_{t,\mathbf{x}}}\bar{\alpha}(\boldsymbol{\theta}_{t,\mathbf{x}},\boldsymbol{\zeta})^\top\, \mathcal{I}_{\mathcal{S}}(\boldsymbol{\theta}_{t,\mathbf{x}})^{-1}\;\nabla_{\boldsymbol{\theta}_{t,\mathbf{x}}}\bar{\alpha}(\boldsymbol{\theta}_{t,\mathbf{x}},\boldsymbol{\zeta})\Big]\,. \tag{7}$$

Proof is given in Appendix D.2. Equation 6 separates the roles of the surrogate and the acquisition in the inner BO optimization problem. The trace $\text{tr}(\mathbf{G}_t(\mathbf{x}))$ depends only on the surrogate map and measures how strongly the predictive distribution changes under local perturbations at $\mathbf{x}$. Equation 7 defines the *acquisition-side sensitivity* of the AF value to perturbations on the predictive manifold $\mathcal{S}$. For a fixed AF and predictive distribution family, the formula for acquisition-side sensitivity depends on the current predictive state $\boldsymbol{\theta}$ and the manifold metric $\mathcal{I}_{\mathcal{S}}(\boldsymbol{\theta})$, but not on which surrogate produced that state. Surrogates represented through the same Gaussian predictive family, such as GP posteriors and Gaussian predictive approximations to Bayesian neural networks (BNNs), therefore use the same acquisition-side formula when evaluated at the same predictive parameters. The surrogate-specific effect enters through $\text{tr}(\mathbf{G}_t(\mathbf{x}))$. Appendix D.4 discusses C_α for common AFs under a Gaussian predictive distribution.

Table 1: AF constants $C_\alpha(\boldsymbol{\theta})$ for scalar Gaussian predictive distribution $\theta \in \mathcal{S}$. Derivations are in Appendix D.4, including Lemma D.3.

AF	Analytic representation	Monte Carlo representation
EI	$\leq \Sigma(1 + 1/(4\pi))$	$\leq \frac{3}{2}\Sigma$
UCB	$= \Sigma(1 + \kappa_{\mathrm{UCB}}/2)$	$= \Sigma(1 + \kappa_{\mathrm{UCB}}\pi/4)$
PI	$\leq 1/(2\pi)$	$\leq 3\Sigma/(32\tau^2)$
LogEI	$= \dfrac{C_\alpha^{\mathrm{EI}}(\boldsymbol{\theta})}{\bar{\alpha}^{\mathrm{EI}}(\boldsymbol{\theta})^2}$	–

When the surrogate maps a region of input space to a near-constant predictive distribution, or when the posterior reverts to the prior away from the data, $\mathrm{tr}(\mathbf{G}_t(\cdot))$ collapses in that region. Proposition 4.2 then shows that bounded acquisition-side sensitivity forces the acquisition gradient to become small there. This effect is visible in Figure 1 and Figure 2: the AF gradient vanishes where the pullback trace collapses and remains active where the local predictive geometry is non-degenerate. Table 1 gives acquisition-specific constants for Gaussian predictives.

Gradient Bound using Eigenvalues and Geometric Interpretation at Stationary Points For a BO iteration t and an input $\mathbf{x}$, with predictive parameters $\boldsymbol{\theta}_{t,\mathbf{x}} = \varphi_t(\mathbf{x})$, the proof in Appendix D.2 formulates the AF gradient as $\nabla_{\mathbf{x}}\alpha(\mathbf{x}) = \mathbf{A}^\top \mathbf{u}$, where

$$\mathbf{A} = \mathcal{I}_{\mathcal{S}}(\boldsymbol{\theta}_{t,\mathbf{x}})^{1/2}\, J_t(\mathbf{x}), \qquad \mathbf{u} = \mathcal{I}_{\mathcal{S}}(\boldsymbol{\theta}_{t,\mathbf{x}})^{-1/2}\, \mathbb{E}_{\boldsymbol{\zeta}}\left[\nabla_{\boldsymbol{\theta}_{t,\mathbf{x}}} \bar{\alpha}(\boldsymbol{\theta}_{t,\mathbf{x}}, \boldsymbol{\zeta})\right]. \tag{8}$$

Let $\mathbf{M} = \mathbf{A}\mathbf{A}^\top$, and $\nu_{\min}$ and $\nu_{\max}$ denote the smallest and largest eigenvalues of $\mathbf{M}$. One can also write $\|\nabla_{\mathbf{x}}\alpha(\mathbf{x})\|_2^2 = \mathbf{u}^\top \mathbf{M}\mathbf{u}$. For a scalar Gaussian predictive, $\mathbf{M}$ is 2×2, the matrix $\mathbf{M}$ is a Hermitian matrix, so using Rayleigh quotient we can also show that the acquisition gradient norm is bounded by the eigenvalues of $\mathbf{M}$ as

$$\nu_{\min}\|\mathbf{u}\|_2^2 \;\leq\; \|\nabla_{\mathbf{x}}\alpha(\mathbf{x})\|_2^2 \;\leq\; \nu_{\max}\|\mathbf{u}\|_2^2 \;\leq\; C_\alpha(\boldsymbol{\theta}_{t,\mathbf{x}})\,\mathrm{tr}(\mathbf{G}_t(\mathbf{x}))\,. \tag{9}$$

For a scalar Gaussian predictive, $\mathbf{M} = \mathbf{A}\mathbf{A}^\top$ and the pullback tensor, $\mathbf{G}_t(\mathbf{x}) = \mathbf{A}^\top\mathbf{A}$, has the same trace. Since $\nu_{\max} \leq \mathrm{tr}(\mathbf{M}) \leq 2\nu_{\max}$, we can say that the trace upper bounds has a slack of at most $\sqrt{2}$ times the largest eigenvalue, with the same acquisition factor, independent of the input dimension. As shown in the Appendix D.2, $\|\mathbf{u}\|_2^2 \leq C_\alpha(\boldsymbol{\theta}_{t,\mathbf{x}})$, where the equality exists for the analytic acquisition functions. For a scalar Gaussian predictive with variance $\Sigma_{t,\mathbf{x}} > 0$ and nonzero mean and variance gradients, the two rows of $\mathbf{A}$ in (8) are $(\nabla_{\mathbf{x}}\mu_{t,\mathbf{x}})^\top/\sqrt{\Sigma_{t,\mathbf{x}}}$ and $(\nabla_{\mathbf{x}}\Sigma_{t,\mathbf{x}})^\top/(\sqrt{2}\Sigma_{t,\mathbf{x}})$.

At an interior stationary point, the gradient $\nabla_{\mathbf{x}}\alpha(\mathbf{x}) = \mathbf{A}^\top\mathbf{u} = 0$, which implies $\mathbf{M}\mathbf{u} = 0$. If the acquisition-side factor $\mathbf{u} \neq \mathbf{0}$, this implies that $\mathbf{M}$ is singular and has a nonzero nullspace. This can happen only if the mean and variance gradients are collinear, as shown in Figure 3(a). That is, at an interior stationary point, which may be a saddle point or a local or global optimum, $\nabla_{\mathbf{x}}\mu_{t,\mathbf{x}} = c\nabla_{\mathbf{x}}\Sigma_{t,\mathbf{x}}$, where $c \in \mathbb{R}$ is determined by the acquisition derivatives at that point. For an analytic AF with nonzero mean derivative, $c = -\frac{\partial\bar{\alpha}(\boldsymbol{\theta}_{t,\mathbf{x}})/\partial\Sigma_{t,\mathbf{x}}}{\partial\bar{\alpha}(\boldsymbol{\theta}_{t,\mathbf{x}})/\partial\mu_{t,\mathbf{x}}}$. This collinearity condition is necessary, not sufficient, for stationarity. This is an important insight as the inequality in (9) defines the necessary conditions for an input location $\mathbf{x}$ to be an optimizer of AF for univariate Gaussian predictive posterior. This can motivate new optimization methods.

4.2 Unified FIM-based Framework for HDBO analysis

Lengthscale Scaling and Random Axis Aligned Subspace Perturbations For GP surrogates with AFs such as EI, UCB, or PI, the sensitivity $C_\alpha(\varphi_t(\mathbf{x}))$ remains uniformly bounded as D_x grows, as shown in Appendix D.4. The experiments use LogEI, for which the same local regularity argument applies in the strictly positive-variance regime considered here. As shown in Appendix D.5, for i.i.d. inputs drawn uniformly from $[0,1]^{D_x}$, the expected Euclidean distance $d = \|\mathbf{x} - \mathbf{x}'\|$ is $\Theta(\sqrt{D_x})$. We can write $\lambda \propto D_x^p$, where p controls the lengthscale growth. The SE kernel then satisfies $k_\lambda(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp[-\Theta(D_x^{1-2p})]$. The $\sqrt{D_x}$ scaling was identified by Hvarfner et al. [6], Xu et al. [7], Papenmeier et al. [8]. The manifold view also explains why $\sqrt{D_x}$ scaling is the optimal dimension

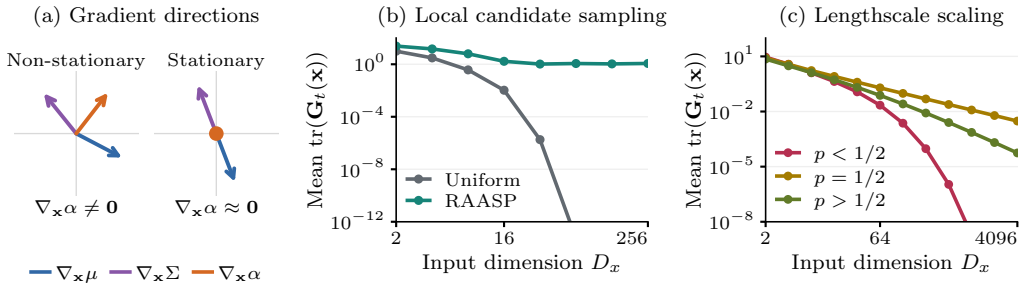

Figure 3: Unified view of HDBO. (a) Collinear mean and variance gradients at a stationary point with $\nabla_{\mathbf{x}}\alpha \approx 0$. (b) RAASP candidates have nonzero trace of pullback tensor $\mathrm{tr}(\mathbf{G}_t(\mathbf{x}))$. (c) $p = 1/2$ scaling gives the slowest asymptotic decay of $\mathrm{tr}(\mathbf{G}_t(\mathbf{x}))$.

informed scaling for lengthscale priors. As shown in Figure 3(c) and Appendix D.5, $p = 1/2$ gives the slowest asymptotic trace decay, $\Theta(D_x^{-1})$, among other scalings. Isotropic Matérn kernels depend on the same normalized distance d/λ and have the same critical scaling. The trace-collapse argument also helps explain why RAASP around observed data [9] improves multi-start AF optimization (Figure 3(b)). The pullback Fisher tensor therefore provides a common language for both failure modes and fixes in HDBO.

TuRBO connection Figure 2 shows that the pullback trace and the AF gradient remain non-degenerate near the observed data. TuRBO develops on a similar principle, where it centers a local axis-aligned TR at the incumbent and adapts its size over time using the learned lengthscales [11]. The dimensions with longer lengthscales have a longer TR length. Appendix D.6 shows that the metric tensor at a local optimum $\mathbf{x}_{\text{best}}$ is often dominated by the variance-gradient term, which is inversely proportional to the lengthscales. The ellipsoid made using the pullback metric would be longer in the dimensions with longer lengthscales. This is similar to how TuRBO uses the learned lengthscales to create TR. This motivates the use of pullback-metric-based TR for BO, which we call the Fisher-Information Trust Region (FITR). The FITR is a location-dependent TR that can be extended to any surrogate for which the derivative of the predictive distribution with respect to the input variables can be calculated. Figure 6 in Appendix C illustrates the local geometry for different surrogates.

5 Fisher-Information Trust Region (FITR)

As explained in Appendix B.1, TuRBO centers the trust region at the best observed location $\mathbf{x}_{\text{best}}$ and scales the dimension-wise side lengths using kernel lengthscales [11]. This construction is effective because it avoids performing the inner AF optimization in large regions where the surrogate has little information. Since the kernels are isotropic, the notion of distance underlying TuRBO remains Euclidean. TuRBO has proven effective in practice, but is also limited to SE or Matérn kernels.

Algorithm 1: FITR weight construction

Input: Center $\mathbf{x}_{t,c}$, data $\mathcal{D}_t$
Output: Raw FITR weights $\bar{w}_{t,j}$ for $j = 1, \ldots, D_x$
Fit surrogate on $\mathcal{D}_t$;
Draw $\tilde{y}_s \sim p(y; \varphi_t(\mathbf{x}_{t,c}))$ for $s = 1, \ldots, S$;
Estimate diagonal pullback Fisher entries as mentioned in (11);
Compute ε_t using estimated diagonal entries $\widehat{G}_{t,j}(\mathbf{x}_{t,c})$;
Form $\bar{w}_{t,j} \leftarrow (\widehat{G}_{t,j}(\mathbf{x}_{t,c}) + \varepsilon_t)^{-1/2}$ for each $j = 1, \ldots, D_x$;
return $\bar{\boldsymbol{w}}_t = (\bar{w}_{t,j})_{j=1,\ldots,D_x}$;

Section 4 suggests a different local notion of distance. The acquisition depends on $\mathbf{x}$ through $\varphi_t(\mathbf{x})$, so the relevant local geometry is defined by the pullback Fisher tensor $\mathbf{G}_t(\mathbf{x})$ in (5). This tensor can be computed for any differentiable probabilistic surrogate that satisfies Assumption 4.1. The ideal local Fisher region is the ellipsoid

$$\left\{\mathbf{x} \in \mathcal{X} : (\mathbf{x} - \mathbf{x}_{t,c})^\top \mathbf{G}_t(\mathbf{x}_{t,c})(\mathbf{x} - \mathbf{x}_{t,c}) \leq \kappa_t^2\right\}. \tag{10}$$

Limitations. The trust region, local ellipsoid, requires inversion or whitening of the pullback tensor. For scalar predictive families, the pullback rank is at most two, so the tensor is rank-deficient when $D_x > 2$. Without regularization, near-null directions would produce arbitrarily large radii. A separate issue is sampling: popular BO frameworks such as BoTorch largely support sampling from axis-aligned trust regions. One can create a hyperrectangle from the pullback tensor by using the infinity norm and whitening to make it axis-aligned. However, once such a region intersects the constraints/bounds of $\mathcal{X}$, the feasible set becomes a clipped polytope rather than a simple box. Polytope sampling is substantially slower and less efficient than simple axis-aligned Sobol sampling for multi-start AF optimization.

Proposed method. The experiments in this paper use a diagonal approximation of the pullback Fisher tensor rather than the full ellipsoid in (10). At the trust-region center $\mathbf{x}_{t,c}$, we first estimate the diagonal entries of the unregularized pullback Fisher tensor by Monte Carlo (MC) samples from the predictive posterior:

$$\widehat{G}_{t,j}(\mathbf{x}_{t,c}) = \frac{1}{S}\sum_{s=1}^{S}\left(\frac{\partial}{\partial x_j}\log p(\tilde{y}_s\,;\varphi_t(\mathbf{x}))\Big|_{\mathbf{x}=\mathbf{x}_{t,c}}\right)^2, \qquad \tilde{y}_s \sim p(y;\varphi_t(\mathbf{x}_{t,c})), \tag{11}$$

where S is the number of MC samples and $j \in \{1,\dots,D_x\}$. Appendix D.8 shows that Equation 11 is an unbiased MC estimator of the corresponding diagonal entry [25]. For a scalar Gaussian predictive, the diagonal is also available exactly from (25), without score samples. The reported BO runs use the MC estimator. Regularization is added after the estimation step.

We use diagonal Tikhonov-style regularization [25] to avoid unbounded stretching in directions associated with zero or near-zero eigenvalues of the pullback tensor:

$$\bar{w}_{t,j} = \left(\widehat{\mathbf{G}}_{t,j}(\mathbf{x}_{t,c}) + \varepsilon_t\right)^{-1/2}.$$

The regularization parameter is chosen adaptively from the local Fisher scale as $\varepsilon_t = \frac{1}{D_x}\sum_{j=1}^{D_x}\widehat{\mathbf{G}}_{t,j}(\mathbf{x}_{t,c}) + \varepsilon_{\text{jit}}$, where ε_{jit} is a small numerical jitter to prevent division by zero. This damping adds an isotropic curvature floor and interpolates between induced geometry and Euclidean geometry when the pullback tensor is degenerate [25]. The ℓ^2 norm gives an ellipsoid from (10). When we use the ℓ_∞ norm, we obtain a box-shaped TR. We implement axis-aligned FITR using only the diagonal of the regularized pullback tensor. This yields a location-dependent axis-aligned box that remains compatible with standard BO toolchains and avoids the clipped-polytope sampling problem at the domain boundary. The diagonal approximation is $\mathrm{O}(D_x)$ time and storage; compared to $\mathrm{O}(D_x^2)$ for a dense whitening matrix, keeping the method practical in higher dimensions. TuRBO computes trust-region weights as in Algorithm 2 and then uses volume normalization and trust-region-bounded AF optimization to select the next candidate, as outlined in Algorithm 3. FITR replaces only the local weight construction in Algorithm 2 and keeps the same volume normalization and outer control logic. The resulting trust region is a box in (12).

$$\mathcal{X}_t^{\mathcal{TR}} = \left\{\mathbf{x}\in\mathcal{X} : \left\|\mathrm{diag}(\bar{\boldsymbol{w}}_t)^{-1}(\mathbf{x}-\mathbf{x}_{t,c})\right\|_\infty \le \frac{\kappa_t}{2}\right\}, \tag{12}$$

where $\bar{\boldsymbol{w}}_t$ denotes the volume-normalized weights obtained from the raw values returned by Algorithm 1. The only algorithmic change relative to TuRBO is therefore the local weight construction: ARD lengthscales are replaced by regularized pullback-Fisher sensitivities, while the trust-region scaling rule and outer control logic remain unchanged. We refer to the full whitening-based construction as FullFITR and treat it as an ablation rather than the default method.

6 Experiments

Problems and baselines. We evaluate FITR with a GP using an SE kernel against strong baselines such as TuRBO [11], DSP [6], and BOUNCE [36] on nine continuous benchmark problems, as shown in Figure 4. These are control and robotics benchmarks commonly used in high-dimensional BO [11, 36]. HPA102-1 is a human-powered aircraft design task from the HPA benchmark suite [37]. MOPTA08 is an automotive design benchmark [11]; LassoDNA and SVM test sparse regression and hyperparameter tuning behavior. Implementation details are given in Appendix C. We also evaluate the same FITR construction with a non-lengthscale kernel, namely IBNN [21], as a transferability check beyond standard ARD kernels. Figures 4 and 5 report the best observed objective value in the benchmark convention, while the BO procedure itself is written in maximization form.

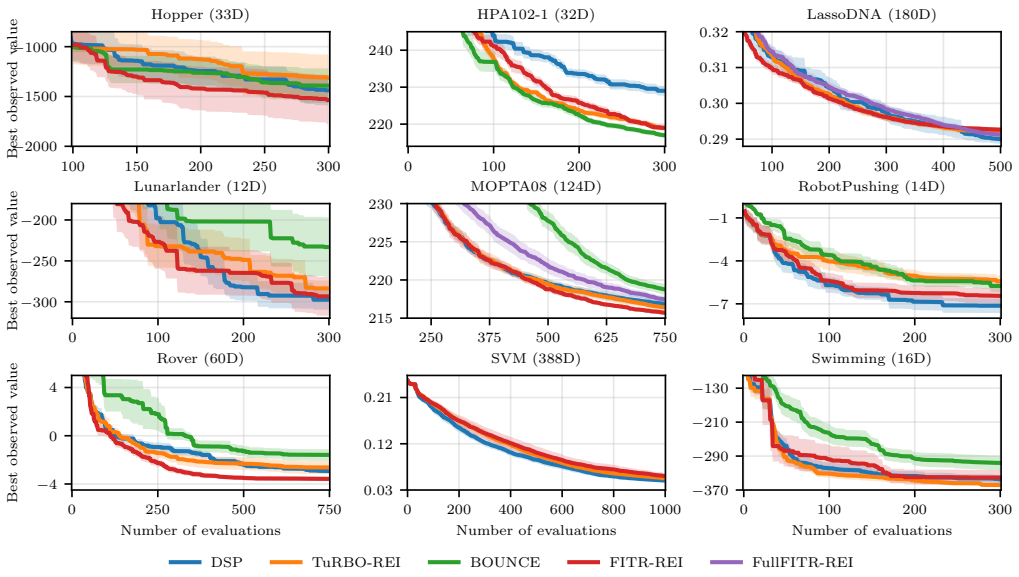

Figure 4: BO performance (lower is better) comparison for different methods with a GP using an SE kernel. Solid curves show means over 11 trials, and shaded bands show standard error. FITR-REI and TuRBO-REI use the same GP modeling pipeline and regional trust-region center selection; their trust-region shapes use ARD lengthscales and the pullback Fisher tensor, respectively.

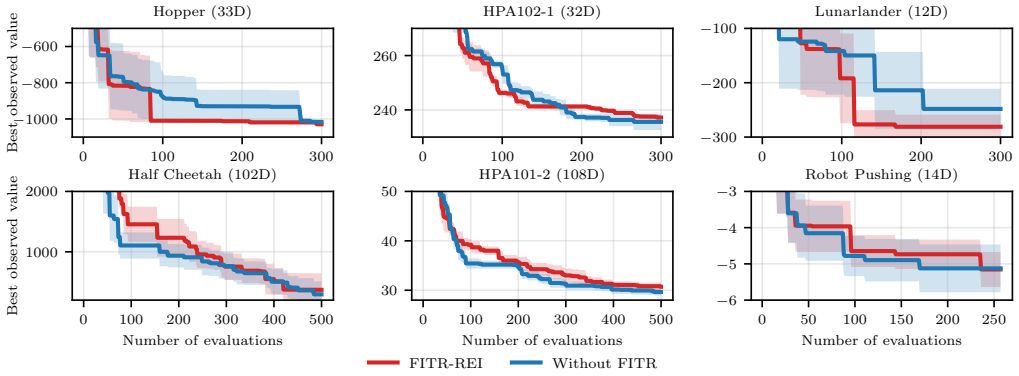

Figure 5: BO performance comparison (lower is better) with a GP using an IBNN kernel of depth 3. Solid curves show means over 5 trials, and shaded bands show standard error.

Results Figure 4 shows that, for a GP with an SE kernel, FITR has lower observed mean than TuRBO-REI on several benchmarks and remains competitive with strong baselines such as DSP and BOUNCE on the remainder. Table 2 reports the final best observed values for different benchmarks. Figure 7 in Appendix C reports the corresponding trust-region diagnostics. With diagonal Tikhonov regularization, the anisotropy remains bounded while adapting to local predictive geometry. The trust-region length traces for FITR and TuRBO are qualitatively similar. Appendix C.1 reports the time to compute the Fisher diagonal used for the FITR weights.

On the subset of problems where we evaluated FullFITR, it did not outperform diagonal FITR. A likely reason is that the pullback geometry is local. If the geometry varies rapidly near the trust-region center, the rotated region estimated at the center need not model nearby candidate locations better than the diagonal approximation. Boundary clipping and the higher numerical cost of the full tensor may also contribute. We therefore use diagonal FITR as the practical default.

The IBNN experiments in Figure 5 use the same pipeline with a depth-3 IBNN kernel [21]. In this setting we compare FITR only against standard BO with acquisition optimization over the full input

Table 2: Final best observed values for GP(SE), reported as mean ± standard error (lower is better). Relative improvement of FITR-REI over TuRBO-REI has also been shown (higher is better).

Problem	TuRBO-REI	FITR-REI	Improvement (%)
Hopper (33D)	-1312.41 ± 231.02	-1536.81 ± 242.86	$+17.10$
HPA102-1 (32D)	219.01 ± 0.82	218.99 ± 0.60	$+0.01$
LassoDNA (180D)	0.29195 ± 0.00093	0.29263 ± 0.00066	-0.24
Lunarlander (12D)	-283.66 ± 9.69	-293.36 ± 23.62	$+3.42$
MOPTA08 (124D)	216.40 ± 0.46	215.70 ± 0.27	$+0.32$
RobotPushing (14D)	-5.43 ± 0.40	-6.45 ± 0.54	$+18.79$
Rover (60D)	-2.63 ± 0.55	-3.58 ± 0.19	$+36.18$
SVM (388D)	0.05336 ± 0.00306	0.05780 ± 0.00613	-8.32
Swimming (16D)	-358.25 ± 1.80	-340.55 ± 17.87	-4.94

domain $\mathcal{X}$, so the figure should be read as a transferability check rather than as a full non-lengthscale benchmark comparison. FITR improves on the non-trust-region baseline on a subset of tasks and remains comparable on the remainder. The results show that the same FITR pipeline can be used for a non-lengthscale kernel, although the performance gain is task-dependent.

7 Limitations

Proposition 4.2 bounds the acquisition gradient during inner optimization; it does not imply a regret guarantee or a global convergence statement for the outer BO loop. One could also replace the trust region by a KL-divergence constraint and solve the resulting problem with an Augmented Lagrangian Method (ALM) [38]. During our experiments, however, the ALM variant became highly exploitative and less exploratory, which degraded performance. For non-isotropic kernels like IBNNs, the pullback trace need not collapse with D_x in the same way as for many lengthscale-dominated GPs [21], so the vanishing-gradient narrative from Section 4.2 applies less sharply. The improvement in performance using FITR becomes task-dependent. A precise analysis of when local Fisher geometry is advantageous beyond isotropic kernels is left for future work.

8 Conclusion

We propose an information-geometric perspective on Bayesian optimization (BO). The surrogate is treated as a smooth map from the input space to the manifold of predictive distributions, and reparameterizable acquisition functions depend on the input only through these predictive parameters. Proposition 4.2 bounds the acquisition gradient using an acquisition-side sensitivity and the trace of the pullback Fisher tensor. For bounded acquisition-side sensitivity, the gradient vanishes when the pullback trace collapses. This provides a common explanation for several observations in high-dimensional BO, including dimension-scaled lengthscales and local initialization around observed data. Using the gradient bound, we also identify necessary conditions for an input location to be an optimizer of AF for a univariate Gaussian predictive posterior.

Using these insights, we propose the Fisher-Information Trust Region (FITR), a simple trust-region method for differentiable probabilistic surrogates that satisfy the stated regularity assumptions. FITR shapes the trust region using local pullback-Fisher weights at the current center and replaces the lengthscale-based scaling in TuRBO by a diagonal approximation to the regularized pullback Fisher tensor. In our experiments, this diagonal FITR variant improves on TuRBO-REI on several GP benchmarks and remains competitive elsewhere. FITR does not introduce task-specific tuning and can be extended to other predictive families and to batch or multi-objective settings.

This perspective opens several research directions in BO. These could include new theoretical insights, manifold-based AF optimization, or new regret bounds. The present work studies a theoretical aspect of BO; we do not foresee any broader societal impacts.

9 Acknowledgment

The research in this paper was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - 459291153 (FOR5399).

References

[1] Sebastian Ament, Samuel Daulton, David Eriksson, Maximilian Balandat, and Eytan Bakshy. Unexpected improvements to expected improvement for bayesian optimization. *Advances in Neural Information Processing Systems*, 36:20577–20612, 2023.

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

[3] Jia Wu, Xiu-Yun Chen, Hao Zhang, Li-Dong Xiong, Hang Lei, and Si-Hao Deng. Hyperparameter optimization for machine learning models based on bayesian optimization. *Journal of Electronic Science and Technology*, 17(1):26–40, 2019.

[4] Noémie Jaquier, Leonel Rozo, Sylvain Calinon, and Mathias Bürger. Bayesian optimization meets riemannian manifolds in robot learning. In *Conference on Robot Learning*, pages 233–246. PMLR, 2020.

[5] Rémi Lam, Matthias Poloczek, Peter Frazier, and Karen E Willcox. Advances in bayesian optimization with applications in aerospace engineering. In *2018 AIAA Non-Deterministic Approaches Conference*, page 1656, 2018.

[6] Carl Hvarfner, Erik Orm Hellsten, and Luigi Nardi. Vanilla bayesian optimization performs great in high dimensions. *arXiv preprint arXiv:2402.02229*, 2024.

[7] Zhitong Xu, Haitao Wang, Jeff M Phillips, and Shandian Zhe. Standard gaussian process is all you need for high-dimensional bayesian optimization. *arXiv preprint arXiv:2402.02746*, 2024.

[8] Leonard Papenmeier, Matthias Poloczek, and Luigi Nardi. Understanding high-dimensional bayesian optimization. *arXiv preprint arXiv:2502.09198*, 2025.

[9] Leonard Papenmeier, Luigi Nardi, and Matthias Poloczek. Increasing the scope as you learn: Adaptive bayesian optimization in nested subspaces. *Advances in Neural Information Processing Systems*, 35:11586–11601, 2022.

[10] Colin Doumont, Donney Fan, Natalie Maus, Jacob R Gardner, Henry Moss, and Geoff Pleiss. We still don't understand high-dimensional bayesian optimization. *arXiv preprint arXiv:2512.00170*, 2025.

[11] David Eriksson, Michael Pearce, Jacob Gardner, Ryan D Turner, and Matthias Poloczek. Scalable global optimization via local bayesian optimization. *Advances in neural information processing systems*, 32, 2019.

[12] Moritz Herrmann, F Julian D Lange, Katharina Eggensperger, Giuseppe Casalicchio, Marcel Wever, Matthias Feurer, David Rügamer, Eyke Hüllermeier, Anne-Laure Boulesteix, and Bernd Bischl. Position: Why we must rethink empirical research in machine learning. *arXiv preprint arXiv:2405.02200*, 2024.

[13] Shun-ichi Amari. *Information geometry and its applications*. Springer, 2016.

[14] John M Lee. *Introduction to Riemannian manifolds*, volume 2. Springer, 2018.

[15] Maximilian Balandat, Brian Karrer, Daniel Jiang, Samuel Daulton, Ben Letham, Andrew G Wilson, and Eytan Bakshy. Botorch: A framework for efficient monte-carlo bayesian optimization. *Advances in neural information processing systems*, 33:21524–21538, 2020.

[16] Roman Garnett. *Bayesian Optimization*. Cambridge University Press, 2023.

[17] James Wilson, Frank Hutter, and Marc Deisenroth. Maximizing acquisition functions for bayesian optimization. *Advances in neural information processing systems*, 31, 2018.

[18] Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer school on machine learning*. Springer, 2003.

[19] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

[20] Henry B Moss, Sebastian W Ober, and Victor Picheny. Inducing point allocation for sparse gaussian processes in high-throughput bayesian optimisation. In *International Conference on Artificial Intelligence and Statistics*, pages 5213–5230. PMLR, 2023.

[21] Yucen Lily Li, Tim GJ Rudner, and Andrew Gordon Wilson. A study of bayesian neural network surrogates for bayesian optimization. *arXiv preprint arXiv:2305.20028*, 2023.

[22] Jonas Mockus. The application of bayesian methods for seeking the extremum. *Towards global optimization*, 2:117, 1998.

[23] James T Wilson, Riccardo Moriconi, Frank Hutter, and Marc Peter Deisenroth. The reparameterization trick for acquisition functions. *arXiv preprint arXiv:1712.00424*, 2017.

[24] Shun-ichi Amari and Hiroshi Nagaoka. *Methods of information geometry*, volume 191. American Mathematical Soc., 2000.

[25] James Martens. New insights and perspectives on the natural gradient method. *Journal of Machine Learning Research*, 21(146):1–76, 2020.

[26] Shun-ichi Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2): 251–276, 1998.

[27] Mohammad Emtiyaz Khan and Håvard Rue. The bayesian learning rule. *Journal of Machine Learning Research*, 24(281):1–50, 2023.

[28] Minyoung Kim, Da Li, Shell Xu Hu, and Timothy Hospedales. Fisher SAM: Information geometry and sharpness aware minimisation. In *International Conference on Machine Learning*, pages 11148–11161. PMLR, 2022.

[29] Hadi Beik-Mohammadi, Søren Hauberg, Georgios Arvanitidis, Gerhard Neumann, and Leonel Rozo. Learning riemannian manifolds for geodesic motion skills. *arXiv preprint arXiv:2106.04315*, 2021.

[30] Noémie Jaquier and Leonel Rozo. High-dimensional bayesian optimization via nested riemannian manifolds. *Advances in Neural Information Processing Systems*, 33:20939–20951, 2020.

[31] Rosen Ting-Ying Yu, Cyril Picard, and Faez Ahmed. Git-bo: High-dimensional bayesian optimization with tabular foundation models. *arXiv preprint arXiv:2505.20685*, 2025.

[32] Alessandra Tosi, Søren Hauberg, Alfredo Vellido, and Neil D Lawrence. Metrics for probabilistic geometries. *arXiv preprint arXiv:1411.7432*, 2014.

[33] Georgios Arvanitidis, Lars Kai Hansen, and Søren Hauberg. Latent space oddity: on the curvature of deep generative models. *arXiv preprint arXiv:1710.11379*, 2017.

[34] Georgios Arvanitidis, Miguel González-Duque, Alison Pouplin, Dimitris Kalatzis, and Søren Hauberg. Pulling back information geometry. *arXiv preprint arXiv:2106.05367*, 2021.

[35] Leonel Rozo, Miguel González-Duque, Noémie Jaquier, and Søren Hauberg. Riemann^2: Learning riemannian submanifolds from riemannian data. *arXiv preprint arXiv:2503.05540*, 2025.

[36] Leonard Papenmeier, Luigi Nardi, and Matthias Poloczek. Bounce: Reliable high-dimensional bayesian optimization for combinatorial and mixed spaces. *Advances in Neural Information Processing Systems*, 36:1764–1793, 2023.

[37] Nobuo Namura. Single and multi-objective optimization benchmark problems focusing on human-powered aircraft design. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 195–210. Springer, 2025.

[38] Ernesto G Birgin and José Mario Martínez. *Practical augmented Lagrangian methods for constrained optimization*. SIAM, 2014.

[39] Jialei Wang, Scott C Clark, Eric Liu, and Peter I Frazier. Parallel bayesian global optimization of expensive functions. *Operations Research*, 68(6):1850–1865, 2020.

[40] Nobuo Namura and Sho Takemori. Regional expected improvement for efficient trust region selection in high-dimensional bayesian optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 19624–19632, 2025.

[41] Stephen P Smith. Differentiation of the cholesky algorithm. *Journal of Computational and Graphical Statistics*, 4(2):134–147, 1995.

[42] Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias W Seeger. Information-theoretic regret bounds for gaussian process optimization in the bandit setting. *IEEE transactions on information theory*, 58(5):3250–3265, 2012.

[43] Donald R Jones, Matthias Schonlau, and William J Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998.

A Gaussian Process

A Gaussian process (GP) is a distribution over functions, where we place a prior on the unknown objective and update our belief using the noisy observations [18].

We model the latent function as

$$f \sim \mathcal{GP}\big(m(\cdot), k(\cdot, \cdot')\big),$$

where $m(\cdot)$ is the prior mean and $k(\cdot, \cdot')$ is the covariance kernel. For observed input-output pairs $\mathcal{D}_t = \{(\mathbf{x}_n, y_n)\}_{n=1}^N$, we use the Gaussian observation model

$$y_n = f(\mathbf{x}_n) + \varepsilon_n, \qquad \varepsilon_n \sim \mathcal{N}(0, \sigma_\varepsilon^2).$$

The variance σ_ε^2 captures observation noise and appears as a diagonal correction to the kernel matrix.

Given $\mathcal{D}_t = \{(\mathbf{x}_n, y_n)\}_{n=1}^N$, the predictive distribution at a query point $\mathbf{x}$ is Gaussian

$$p(y \mid \mathbf{x}, \mathcal{D}_t) = \mathcal{N}(\mu_t(\mathbf{x}),\, \Sigma_t(\mathbf{x}))\,,$$

where the posterior mean and variance are given by standard Gaussian conditioning identities [18]

$$\mu_t(\mathbf{x}) = m(\mathbf{x}) + k(\mathbf{x}, \mathbf{X}) \left(k(\mathbf{X}, \mathbf{X}) + \sigma_\varepsilon^2 \mathbf{I}\right)^{-1} (\mathbf{y} - m(\mathbf{X}))\,, \tag{13}$$

$$\Sigma_t(\mathbf{x}) = k(\mathbf{x}, \mathbf{x}) - k(\mathbf{x}, \mathbf{X}) \left(k(\mathbf{X}, \mathbf{X}) + \sigma_\varepsilon^2 \mathbf{I}\right)^{-1} k(\mathbf{X}, \mathbf{x}). \tag{14}$$

and the GP hyperparameters (for example, lengthscales in Λ and the noise level σ_ε^2) are estimated by maximizing the log marginal likelihood [18]

$$\log p(\mathbf{y} \mid \mathbf{X}, \Lambda) = -\frac{1}{2} (\mathbf{y} - m(\mathbf{X}))^\top K_\Lambda^{-1} (\mathbf{y} - m(\mathbf{X})) - \frac{1}{2} \log|K_\Lambda| - \frac{N}{2} \log(2\pi). \tag{15}$$

where $K_\Lambda = k(\mathbf{X}, \mathbf{X}) + \sigma_\varepsilon^2 \mathbf{I}$ [18].

B Common Acquisition Functions: Analytic and MC Forms

Section 2 introduces AF and its reparameterized form (2). When the acquisition is available in closed form, $\bar{\alpha}$ does not depend on $\boldsymbol{\zeta}$ and (2) collapses to

$$\alpha(\mathbf{x}) = \mathbb{E}_{\boldsymbol{\zeta}}[\bar{\alpha}(\boldsymbol{\theta}_{t,\mathbf{x}}; \boldsymbol{\xi})] = \bar{\alpha}(\boldsymbol{\theta}_{t,\mathbf{x}}; \boldsymbol{\xi})\,. \tag{16}$$

Table 3 shows some closed-form AF for a scalar Gaussian predictive, where $z = (\mu_{t,\mathbf{x}} - y^\star)/\Sigma_{t,\mathbf{x}}^{1/2}$, $\Omega_\mathcal{N}$ is the Cumulative Distribution Function (CDF) and $\omega_\mathcal{N}$ is Probability Density Function (PDF) of normal distribution.

Table 3: Closed-form AF $\bar{\alpha}(\boldsymbol{\theta}_{t,\mathbf{x}}; \boldsymbol{\xi})$ under (16) for a scalar Gaussian predictive $\boldsymbol{\theta}_{t,\mathbf{x}} = (\mu_{t,\mathbf{x}}, \Sigma_{t,\mathbf{x}})$. EI and PI use incumbent $y^\star$; UCB uses exploration weight $\kappa_{\mathrm{UCB}} > 0$.

Acquisition	**Closed-form** $\bar{\alpha}(\mu_{t,\mathbf{x}}, \Sigma_{t,\mathbf{x}}; \boldsymbol{\xi})$
EI	$\big(\mu_{t,\mathbf{x}} - y^\star\big)\, \Omega_\mathcal{N}(z) + \Sigma_{t,\mathbf{x}}^{1/2}\, \omega_\mathcal{N}(z)$
PI	$\Omega_\mathcal{N}(z)$
UCB	$\mu_{t,\mathbf{x}} + \sqrt{\kappa_{\mathrm{UCB}}}\, \Sigma_{t,\mathbf{x}}^{1/2}$

Following Wilson et al. [23], the Monte Carlo form of EI can be written as

$$\bar{\alpha}_{\mathrm{EI}}(\boldsymbol{\theta}_{t,\mathbf{x}},\, \boldsymbol{\zeta};\, \boldsymbol{\xi}) = \mathrm{ReLU}\Big(\mu_{t,\mathbf{x}} + \Sigma_{t,\mathbf{x}}^{1/2} \boldsymbol{\zeta} - y^\star\Big)\,. \tag{17}$$

For parallel BO [39] at batch input $\boldsymbol{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_q\}$, we write the reparameterized sample as $\boldsymbol{y} = \boldsymbol{\mu}_{t,\boldsymbol{X}} + \boldsymbol{L}_{t,\boldsymbol{X}} \boldsymbol{\zeta}$, with $\boldsymbol{L}_{t,\boldsymbol{X}} = \mathrm{chol}(\boldsymbol{\Sigma}_{t,\boldsymbol{X}})$ and $\boldsymbol{\zeta} \sim \mathcal{N}(0, \boldsymbol{I})$. Table 4 uses this batch notation because it matches the BoTorch implementation. In the scalar sequential setting of the main text, $q = 1$ and these expressions reduce to the usual one-point acquisitions.

For MC-UCB in the scalar case, the reparameterized form uses the identity $\mathbb{E}|Z| = \sqrt{2/\pi}$ for $Z \sim \mathcal{N}(0, 1)$, so $\sqrt{\kappa_{\mathrm{UCB}}\pi/2}\, \Sigma_{t,\mathbf{x}}^{1/2} |Z|$ has expectation $\sqrt{\kappa_{\mathrm{UCB}}}\, \Sigma_{t,\mathbf{x}}^{1/2}$.

Table 4: Utility a and integrand $\bar{\alpha}(\cdot, \boldsymbol{\zeta}; \cdot)$ inside $\mathbb{E}_{\boldsymbol{\zeta}}$ for reparameterized acquisitions under (2) for a Gaussian predictive [23, 39]. $\mathbf{1}^{+/-}$ denote the right/left-continuous Heaviside step functions; $\mathrm{ReLU}(u) = \max(u, 0)$; $\mathrm{sigmoid}(u) = (1 + e^{-u})^{-1}$; H is Shannon entropy. EI (Expected Improvement) and PI (Probability of Improvement) use incumbent threshold $y^\star$; PI uses temperature $\tau > 0$; the PI row is a sigmoid-smoothed surrogate for Monte Carlo gradients and is not the analytic PI functional $\Omega_{\mathcal{N}}(z)$. UCB uses exploration weight $\kappa_{\mathrm{UCB}} > 0$. For ES (Entropy Search), subscripts a and b mark the query set and discretization points; $\boldsymbol{\mu}_{b|a}$ and $\boldsymbol{L}_{b|a}$ are the conditional posterior mean and Cholesky factor at b given an outcome at a. SR (Simple Regret) does not use extra parameters.

Acquisition	**Utility** $a(y; \boldsymbol{\theta}_{t,\mathbf{x}}, \boldsymbol{\xi})$	**Reparameterized form** $\bar{\alpha}(\boldsymbol{\theta}_{t,\mathbf{x}}, \boldsymbol{\zeta}; \boldsymbol{\xi})$
EI	$\max(\mathrm{ReLU}(\boldsymbol{y} - y^\star))$	$\max(\mathrm{ReLU}(\boldsymbol{\mu}_{t,\boldsymbol{X}} + \boldsymbol{L}_{t,\boldsymbol{X}}\boldsymbol{\zeta} - y^\star))$
PI	$\max\big(\mathbf{1}^-(\boldsymbol{y} - y^\star)\big)$	$\max\Big(\mathrm{sigmoid}\Big(\frac{\boldsymbol{\mu}_{t,\boldsymbol{X}} + \boldsymbol{L}_{t,\boldsymbol{X}}\boldsymbol{\zeta} - y^\star}{\tau}\Big)\Big)$
UCB	$\max\big(\boldsymbol{\mu}_{t,\boldsymbol{X}} + \sqrt{\kappa_{\mathrm{UCB}}\pi/2}\,\lvert\boldsymbol{y} - \boldsymbol{\mu}_{t,\boldsymbol{X}}\rvert\big)$	$\max\big(\boldsymbol{\mu}_{t,\boldsymbol{X}} + \sqrt{\kappa_{\mathrm{UCB}}\pi/2}\,\lvert\boldsymbol{L}_{t,\boldsymbol{X}}\boldsymbol{\zeta}\rvert\big)$
SR	$\max(\boldsymbol{y})$	$\max(\boldsymbol{\mu}_{t,\boldsymbol{X}} + \boldsymbol{L}_{t,\boldsymbol{X}}\boldsymbol{\zeta})$
ES	$-\mathrm{H}\big(\mathbb{E}_{\boldsymbol{y}_b \mid \boldsymbol{y}_a}\big[\mathbf{1}^+(\boldsymbol{y}_b - \max(\boldsymbol{y}_b))\big]\big)$	$-\mathrm{H}\Big(\mathbb{E}_{\boldsymbol{\zeta}_b}\Big[\mathrm{softmax}\Big(\frac{\boldsymbol{\mu}_{b\mid a} + \boldsymbol{L}_{b\mid a}\boldsymbol{\zeta}_b}{\tau}\Big)\Big]\Big)$

Algorithm 2: TuRBO TR weights calculation

Input: Center $\mathbf{x}_{t,c}$, data $\mathcal{D}_t$
Output: TR length weights $\bar{\boldsymbol{w}}_t$, trained surrogate GP $\mathcal{M}_t$
Fit GP on $\mathcal{D}_t$ and extract ARD lengthscales $\lambda_{t,j}$ for all $j = 1, \ldots, D_x$;
$\bar{w}_{t,j} \leftarrow \lambda_{t,j}$ for each $j = 1, \ldots, D_x$;
return $\bar{\boldsymbol{w}}_t = (\bar{w}_{t,j})_{j=1,\ldots,D_x}$ and $\mathcal{M}_t$;

B.1 Trust-Region Bayesian Optimization (TuRBO)

TuRBO replaces a single global search step with local BO updates inside the trust region [11]. At each iteration t, a trust region (TR), $\mathcal{X}_t^{\mathcal{TR}} \subseteq \mathcal{X}$, is defined by its center $\mathbf{x}_{t,c}$ and length κ_t. The TR is an axis-aligned box for which the dimension-wise length is calculated using the GP surrogate $\mathcal{M}_t$ with Automatic Relevance Determination (ARD) kernel. The next candidate is selected by maximizing an AF with bounds as

$$\mathbf{x}_{t+1} = \arg\max_{\mathbf{x} \in \mathcal{X}_t^{\mathcal{TR}}} \alpha(\mathbf{x}; \mathcal{D}_t).$$

The TR length κ_t grows or shrinks using two counters. τ_{succ} tracks the number of consecutive improvements over the incumbent $y^\star$, and τ_{fail} tracks the number of consecutive failures. If τ_{succ} reaches the threshold τ_{succ}, the region expands; if τ_{fail} reaches τ_{fail}, it contracts; both counters reset on any length change as

$$\kappa_{t+1} = \begin{cases} \min(2\kappa_t, \kappa_{\max}), & \text{if success count reaches } \tau_{\mathrm{succ}}, \\ \frac{1}{2}\kappa_t, & \text{if failure count reaches } \tau_{\mathrm{fail}}, \\ \kappa_t, & \text{otherwise.} \end{cases}$$

When $\kappa_t < \kappa_{\min}$ at any iteration, TR is said to have collapsed. It is then restarted around a new center location. Namura and Takemori [40] provide a Regional Expected Improvement (REI) acquisition function that can be used to select the next TR center location when the TR collapses. Eriksson et al. [11] also present a multi-region variant with multiple trust-region centers; however, we omit it here.

Algorithm 3: Trust-region length control and next candidate selection

Input: TR-weight routine `compute_tr_weights`, budget T, thresholds $\tau_{\text{succ}}^{\max}, \tau_{\text{fail}}^{\max}$, bounds $\kappa_{\min}, \kappa_{\max}$ and initial length κ_{init}

Output: Best input $\mathbf{x}^\star$ and value $y^\star$

Initialize $\mathbf{x}_{0,c}$, $\kappa_0 \leftarrow \kappa_{\text{init}}$, $\tau_{\text{succ}} \leftarrow 0$, $\tau_{\text{fail}} \leftarrow 0$, $\mathcal{D}_0$, incumbent $y^\star$;

for $t = 0, \dots, T-1$ **do**

  Get TR length weights $\bar{w}_j$ and $\mathcal{M}_t$ from `compute_tr_weights` with $(\mathbf{x}_{t,c}, \mathcal{D}_t)$;

  Volume normalization;

  $\bar{w}_j \leftarrow \bar{w}_j \big/ \big(\prod_{k=1}^{D_x} \bar{w}_k\big)^{1/D_x}$ for each $j = 1, \dots, D_x$;

  TR construction using upper and lower limits of $\mathcal{X}$;

  for $j = 1, \dots, D_x$ **do**

    $\alpha_j \leftarrow \max\Big(x_j^c - \frac{\bar{w}_j \kappa_t}{2},\ x_j^{\min}\Big)$;

    $\beta_j \leftarrow \min\Big(x_j^c + \frac{\bar{w}_j \kappa_t}{2},\ x_j^{\max}\Big)$;

  Define $\mathcal{X}_t^{\mathcal{TR}}$ by the dimension-wise limits $[\alpha_j,\ \beta_j]$ for all $j = 1, \dots, D_x$;

  Select next candidate $\mathbf{x}_{t+1} \leftarrow \arg\max_{\mathbf{x} \in \mathcal{X}_t^{\mathcal{TR}}} \alpha(\mathbf{x}; \mathcal{D}_t)$;

  Evaluate $y_{t+1} = f_{\text{true}}(\mathbf{x}_{t+1})$ and update $\mathcal{D}_{t+1}$;

  if $y_{t+1} > y^\star$ **then**

    $\tau_{\text{succ}} \leftarrow \tau_{\text{succ}} + 1$; $\tau_{\text{fail}} \leftarrow 0$;

    Update $y^\star$;

  else

    $\tau_{\text{succ}} \leftarrow 0$; $\tau_{\text{fail}} \leftarrow \tau_{\text{fail}} + 1$;

  $$\kappa_{t+1} \leftarrow \begin{cases} \min(2\kappa_t, \kappa_{\max}), & \tau_{\text{succ}} = \tau_{\text{succ}}^{\max} \\ \frac{1}{2}\kappa_t, & \tau_{\text{fail}} = \tau_{\text{fail}}^{\max} \\ \kappa_t, & \text{otherwise} \end{cases};$$

  Reset $\tau_{\text{succ}}, \tau_{\text{fail}} \leftarrow 0$ if κ changed;

  if $\kappa_{t+1} < \kappa_{\min}$ **then**

    Restart around a new center; $\kappa_{t+1} \leftarrow \kappa_{\text{init}}$;

return $\mathbf{x}_{t+1}^\star$ and $y^\star$;

C Implementation Details

For the experiments with a GP using an SE kernel, we use the Dimensionally Scaled LogNormal Priors (DSP) with mean $\sqrt{2} + \log(D_x) * 0.5$ and standard deviation $\sqrt{3}$. DSP [6] uses dimensionally scaled LogNormal priors on the lengthscales and optimizes AF over the full input domain $\mathcal{X}$, which is also the default implementation for SingleTaskGP in BoTorch v0.16.1 [15]. For the experiments with the GP using an IBNN kernel, we keep the BoTorch default depth 3.

TuRBO-REI combines TuRBO's lengthscale-weighted trust region [11] with Regional Expected Improvement for choosing the trust-region center when the TR collapses [40]. Namura and Takemori [40] showed that REI improved TuRBO by selecting a new center in a region with high expected utility. BOUNCE [36] uses an adaptive subspace decomposition and can optimize mixed and combinatorial structures as well. FITR keeps the same trust-region scaling logic as TuRBO-REI (Algorithm 3) and also uses REI center selection at trust-region collapse. It replaces ARD lengthscale weights by the regularized diagonal pullback-Fisher weights from Algorithm 1. Section 5 makes clear that the main method in this paper is the diagonal axis-aligned approximation; we also test a full-tensor ablation (FullFITR). FullFITR uses whitening for Sobol sampling and a simple boundary-clipping strategy instead of polytope MCMC. Due to its additional computational cost, we evaluate FullFITR only on a subset of problems.

DSP, TuRBO-REI, FITR, and FullFITR are implemented in BoTorch [15] and use LogEI [1]. Inputs are normalized to $[0, 1]^{D_x}$, and outputs are standardized before fitting the GP. For each experiment, we start from 30 randomly sampled observed points. We use sequential evaluations ($q = 1$). The SE-kernel experiments are repeated over 11 independently seeded trials, whereas the IBNN experiments

are repeated over 5 independently seeded trials. AF is optimized using L-BFGS with 5 restarts and 256 raw samples. For FITR, the diagonal entries of the pullback Fisher tensor $G_t(\mathbf{x}_{t,c})$ are estimated with $S = 256$ MC samples using (11). These settings are fixed across tasks rather than tuned per benchmark.

The damping parameter is chosen automatically as $\varepsilon_t = D_x^{-1} \sum_{j=1}^{D_x} \widehat{G}_{t,j} + 10^{-12}$. The resulting weights are volume-normalized before constructing the trust region. Using Algorithm 3, the trust-region length κ_t controls the search volume while the pullback tensor controls the local anisotropy. The trust-region length follows the standard TuRBO rule: it doubles after 10 consecutive improvements, halves after $\min\{\lceil \max(\frac{4}{q}, \frac{D_x}{q}) \rceil, 20\}$ consecutive failures, and restarts when the length drops below 0.5^7. FullFITR uses the full regularized pullback Fisher tensor and optimizes LogEI in whitened coordinates. Candidate points are mapped back to the input cube and clipped at the boundary, which keeps the implementation compatible with standard BoTorch box optimization while exposing the additional cost of the full tensor. The experiments were run on an Nvidia GPU Tesla V100-SXM2-32GB.

C.1 Runtime for computing the Fisher diagonal

Computing the Fisher diagonal used to form the FITR weights took 242–261 ms with 256 MC samples, compared to 2.5–4.1 ms using the exact Gaussian expression in Corollary D.2. These isolated CPU timings use synthetic fixed-kernel posteriors with 30 observations for 32, 124, and 388 inputs dimensions.

C.2 Geometry illustrations

Figure 3(a) uses an SE GP with eight observations in $[0, 1]^2$, lengthscales $(0.22, 0.24)$, and observation-noise variance 0.0025. We compare the EI gradient at $(0.70, 0.30)$ with a stationary point found numerically.

For Figure 3(b,c), we use a zero-mean GP with SE kernel with and observation-noise variance 0.1. In (c), $\lambda = 0.35 D_x^p$ with $p \in \{1/4, 1/2, 3/4\}$, using the same uniform candidates for all three powers.

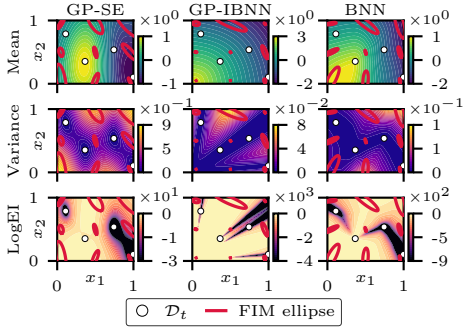

Figure 6: Ellipses using $G(\mathbf{x})$ from (5) for different surrogates: GP with squared exponential (SE) kernel, GP with Infinite Width Bayesian Neural Network (IBNN) kernel [21] and Bayesian Neural Network (BNN).

C.3 Trust-region diagnostics

Figure 7 complements the objective-value plots with internal trust-region diagnostics. The trust-region length traces for FITR and TuRBO-REI remain qualitatively similar across problems, so the performance differences in Figure 4 are not explained by a different restart or length-adaptation schedule. For the diagonal FITR implementation, the logged anisotropy is the ratio between the largest and smallest volume-normalized coordinate weights used to shape the axis-aligned trust region, so values close to 1 indicate a nearly isotropic local region and larger values indicate stronger directional stretching. The diagnostics are shown using the experiment logs whose results are reported in Figure 4.

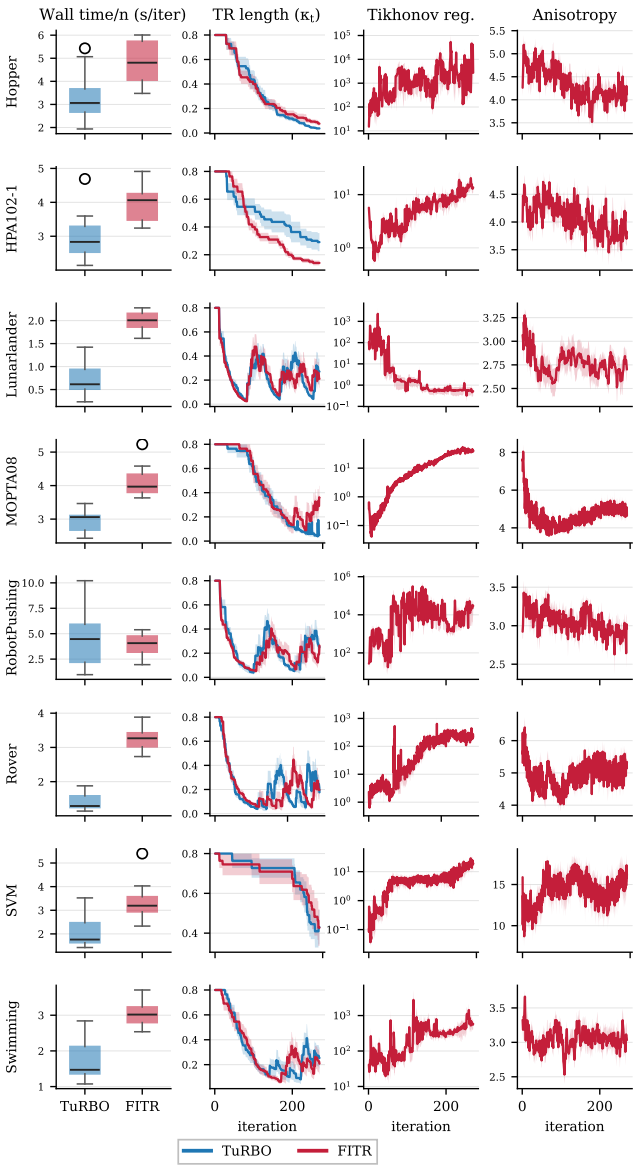

Figure 7: Per-problem diagnostics for TuRBO and FITR on the GP-SE benchmark suite. The first column reports the average time for getting the next candidate in BO iterations. The second column shows the trust-region length κ_t. The last two columns show the FITR-specific Tikhonov regularization and anisotropy diagnostics. Solid lines show means over the 11 seeded runs and shaded bands show standard error.

D Proofs

D.1 Interchange Conditions for Gaussian Reparameterizations

Differentiating under the expectation in Proposition 4.2 requires a valid interchange of gradient and expectation. We use standard reparameterization conditions from Wilson et al. [17]. The lemma below provides a sufficient condition for scalar Gaussian reparameterizations, where the AF integrand depends on the sampled output through the predictive distribution parameters $\boldsymbol{\theta}$. This is the case used later for MC-EI and the smoothed MC-PI surrogate.

Lemma D.1 (Generic Gaussian Reparameterization Condition)**.** *Let Assumption 4.1 hold and let*

$$\bar{\alpha}(\boldsymbol{\theta}_{t,\mathbf{x}}, \boldsymbol{\zeta}; \boldsymbol{\xi}) = a\Big(\mu_{t,\mathbf{x}} + \Sigma_{t,\mathbf{x}}^{1/2}\boldsymbol{\zeta};\, \boldsymbol{\xi}\Big)$$

be a scalar Gaussian reparameterized acquisition, with $\boldsymbol{\zeta} \sim \mathcal{N}(0, I)$ independent of $\mathbf{x}$ [17, 39]. We assume

(a) *$\bar{\alpha}(\cdot, \boldsymbol{\zeta}; \boldsymbol{\xi})$ is locally Lipschitz on $\mathcal{S}$ and differentiable almost everywhere for almost every $\boldsymbol{\zeta}$;*

(b) *$a(y; \boldsymbol{\xi})$ is globally Lipschitz in y.*

Then $\mathbb{E}_{\boldsymbol{\zeta}}\Big[\big\|\nabla_{\boldsymbol{\theta}_{t,\mathbf{x}}}\bar{\alpha}(\boldsymbol{\theta}_{t,\mathbf{x}}, \boldsymbol{\zeta}; \boldsymbol{\xi})\big\|^2\Big] < \infty$ for all $\boldsymbol{\theta}_{t,\mathbf{x}} \in \mathcal{S}$.

Proof. Assumption (a) is the pathwise differentiability condition used in reparameterized acquisition optimization: the integrand admits gradients almost everywhere, which is sufficient for piecewise-smooth utilities such as MC-EI. For square-integrability, let L_a be the Lipschitz constant of a from assumption (b). The chain rule gives, for almost every $\boldsymbol{\zeta}$,

$$\nabla_{\boldsymbol{\theta}_{t,\mathbf{x}}}\bar{\alpha} = \nabla_y a\Big(\mu_{t,\mathbf{x}} + \Sigma_{t,\mathbf{x}}^{1/2}\boldsymbol{\zeta};\, \boldsymbol{\xi}\Big) \cdot \nabla_{\boldsymbol{\theta}_{t,\mathbf{x}}}\Big(\mu_{t,\mathbf{x}} + \Sigma_{t,\mathbf{x}}^{1/2}\boldsymbol{\zeta}\Big). \tag{18}$$

Since a is L_a-Lipschitz in y, the first factor satisfies $\|\nabla_y a\| \le L_a$. The second factor is affine in $\boldsymbol{\zeta}$, so $\|\nabla_{\boldsymbol{\theta}_{t,\mathbf{x}}}(\mu_{t,\mathbf{x}} + \Sigma_{t,\mathbf{x}}^{1/2}\boldsymbol{\zeta})\| \le A + B\|\boldsymbol{\zeta}\|$ for finite constants A, B at a fixed $\boldsymbol{\theta}_{t,\mathbf{x}}$. $\mathrm{chol}(\cdot)$ is smooth on the positive-definite cone, since $\Sigma_{t,\mathbf{x}} \succ 0$ by Assumption 4.1(ii) [41]. Therefore $\|\nabla_{\boldsymbol{\theta}_{t,\mathbf{x}}}\bar{\alpha}\|^2 \le L_a^2(A + B\|\boldsymbol{\zeta}\|)^2$, and since $\boldsymbol{\zeta} \sim \mathcal{N}(0, I)$ has finite second moment the expectation is finite. □

Remark (Necessity of $\boldsymbol{\Sigma} \succ 0$)**.** *If the predictive variance degenerates, derivatives of the square-root factor can blow up, and the bound above would not hold. In the scalar case, $\nabla\Sigma^{1/2} = (\nabla\Sigma)/(2\Sigma^{1/2})$ is singular as $\Sigma \to 0$. A strictly positive noise level rules out this singularity for GP regression. Similar reasoning applies to the batch BO setting; however, we limit our discussion to the scalar case for simplicity in this work.*

Remark (LogEI in the experimental regime)**.** *The experiments use the numerically stabilized LogEI implementation from BoTorch [1]. In the scalar Gaussian setting with strictly positive predictive variance, this implementation is smooth in the predictive parameters on any compact subset of $\mathbb{R} \times (0, \infty)$. The local interchange argument above therefore applies in the positive-variance regime considered in the experiments. Appendix D.4 distinguishes the analytic EI and LogEI sensitivities from those of Monte Carlo representations.*

D.2 Proof of Proposition 4.2 (Acquisition Gradient Bound)

Let $\boldsymbol{\zeta} \sim \mathcal{N}(0, \mathbf{I})$ be the independent base samples in the reparameterized acquisition from Lemma D.1. $\alpha(\mathbf{x}) = \mathbb{E}_{\boldsymbol{\zeta}}[\bar{\alpha}(\boldsymbol{\theta}_{t,\mathbf{x}}, \boldsymbol{\zeta}; \boldsymbol{\xi})]$, where $\boldsymbol{\theta}_{t,\mathbf{x}} = (\mu_{t,\mathbf{x}}, \Sigma_{t,\mathbf{x}})$ are the GP posterior mean and variance at $\mathbf{x}$. $J_t(\mathbf{x}) = \partial\boldsymbol{\theta}_{t,\mathbf{x}}/\partial\mathbf{x} \in \mathbb{R}^{D_\theta \times D_x}$ is the Jacobian of the posterior map.

Step 1 (Chain rule and interchange). By the assumed interchange of gradient and expectation in Proposition 4.2, justified for scalar Gaussian reparameterizations by Lemma D.1, the reparameterized acquisition satisfies

$$\nabla_{\mathbf{x}}\,\alpha(\mathbf{x}) = J_t(\mathbf{x})^\top\, \mathbb{E}_{\boldsymbol{\zeta}}\big[\nabla_{\boldsymbol{\theta}_{t,\mathbf{x}}}\bar{\alpha}(\varphi_t(\mathbf{x}), \boldsymbol{\zeta}; \boldsymbol{\xi})\big], \tag{19}$$

which also aligns with the discussion around Equation 5 of Section 3.2 in Wilson et al. [17]. For ease of notation, we write $\mathbb{E}_{\boldsymbol{\zeta}}\big[\nabla_{\boldsymbol{\theta}_{t,\mathbf{x}}}\bar{\alpha}(\varphi_t(\mathbf{x}), \boldsymbol{\zeta}; \boldsymbol{\xi})\big]$ as $\mathbb{E}_{\boldsymbol{\zeta}}[\nabla_{\boldsymbol{\theta}}\bar{\alpha}(\cdot,\, \boldsymbol{\zeta})]$ in this proof, where the argument $\cdot$ denotes $\varphi_t(\mathbf{x})$.

Step 2 (Fisher metric insertion). Using Assumption 4.1(ii), let $\mathbf{G}_{\mathcal{S}}(\theta_{t,\mathbf{x}}) := \mathcal{I}_{\mathcal{S}}(\varphi_t(\mathbf{x})) \succ 0$. Using the identity $\mathbf{I} = \mathbf{G}_{\mathcal{S}}(\theta_{t,\mathbf{x}})^{1/2}\mathbf{G}_{\mathcal{S}}(\theta_{t,\mathbf{x}})^{-1/2}$ from Amari and Nagaoka [24], we can decompose the gradient into a geometry-dependent factor $\mathbf{A}$ and an acquisition-dependent factor $\mathbf{u}$:

$$\nabla_{\mathbf{x}}\,\alpha(\mathbf{x}) = \underbrace{\big(\mathbf{G}_{\mathcal{S}}(\theta_{t,\mathbf{x}})^{1/2} J_t(\mathbf{x})\big)^{\top}}_{=:\,\mathbf{A}^{\top}} \underbrace{\mathbf{G}_{\mathcal{S}}(\theta_{t,\mathbf{x}})^{-1/2}\mathbb{E}_{\boldsymbol{\zeta}}[\nabla_{\boldsymbol{\theta}}\bar{\alpha}(\cdot,\,\boldsymbol{\zeta})]}_{=:\,\mathbf{u}}, \tag{20}$$

Step 3 (Cauchy–Schwarz). For any matrix $\mathbf{A} \in \mathbb{R}^{p\times q}$ and vector $\mathbf{u} \in \mathbb{R}^p$, $\|\mathbf{A}^{\top}\mathbf{u}\|_2^2 \le \|\mathbf{A}\|_F^2\|\mathbf{u}\|^2$. Applying it to the decomposition from Step 2 gives

$$\|\nabla_{\mathbf{x}}\,\alpha(\mathbf{x})\|_2 \le \sqrt{\underbrace{\Big\|\mathbf{G}_{\mathcal{S}}(\theta_{t,\mathbf{x}})^{1/2} J_t(\mathbf{x})\Big\|_F^2}_{=\,\mathrm{tr}(\mathbf{G}_t(\mathbf{x}))} \cdot \Big\|\mathbf{G}_{\mathcal{S}}(\theta_{t,\mathbf{x}})^{-1/2}\mathbb{E}_{\boldsymbol{\zeta}}[\nabla_{\boldsymbol{\theta}}\bar{\alpha}(\cdot,\,\boldsymbol{\zeta})]\Big\|^2}, \tag{21}$$

where using (5) we have $\mathrm{tr}(\mathbf{G}_t(\mathbf{x})) = \mathrm{tr}\big(J_t^{\top}\,\mathcal{I}_{\mathcal{S}}(\varphi_t(\mathbf{x}))\,J_t\big) = \Big\|\mathcal{I}_{\mathcal{S}}^{1/2}(\varphi_t(\mathbf{x}))\,J_t\Big\|_F^2$. And

$$\Big\|\mathbf{G}_{\mathcal{S}}(\theta_{t,\mathbf{x}})^{-1/2}\mathbb{E}_{\boldsymbol{\zeta}}[\nabla_{\boldsymbol{\theta}}\bar{\alpha}(\cdot,\,\boldsymbol{\zeta})]\Big\|^2 = \mathbb{E}_{\boldsymbol{\zeta}}[\nabla_{\boldsymbol{\theta}}\bar{\alpha}(\cdot,\,\boldsymbol{\zeta})]^{\top}\;\mathbf{G}_{\mathcal{S}}(\theta_{t,\mathbf{x}})^{-1}\,\mathbb{E}_{\boldsymbol{\zeta}}[\nabla_{\boldsymbol{\theta}}\bar{\alpha}(\cdot,\,\boldsymbol{\zeta})]\,. \tag{22}$$

Step 4 (Jensen's inequality). Considering $v = \mathbb{E}_{\boldsymbol{\zeta}}[\nabla_{\boldsymbol{\theta}}\bar{\alpha}(\cdot,\,\boldsymbol{\zeta})]$, (22) can be rewritten as $f : v \mapsto v^{\top}\mathbf{G}_{\mathcal{S}}(\theta_{t,\mathbf{x}})^{-1}v$, where f is convex for $\mathbf{G}_{\mathcal{S}}(\theta_{t,\mathbf{x}}) \succ 0$. Using Jensen's inequality we can write $f(\mathbb{E}_{\boldsymbol{\zeta}}[\nabla_{\boldsymbol{\theta}}\bar{\alpha}(\cdot,\,\boldsymbol{\zeta})]) \le \mathbb{E}_{\boldsymbol{\zeta}}[f(\nabla_{\boldsymbol{\theta}}\bar{\alpha}(\cdot,\,\boldsymbol{\zeta}))]$. Hence, we have

$$\Big\|\mathbf{G}_{\mathcal{S}}(\theta_{t,\mathbf{x}})^{-1/2}\mathbb{E}_{\boldsymbol{\zeta}}[\nabla_{\boldsymbol{\theta}}\bar{\alpha}(\cdot,\,\boldsymbol{\zeta})]\Big\|^2 \le \mathbb{E}_{\boldsymbol{\zeta}}\Big[\nabla_{\boldsymbol{\theta}}\bar{\alpha}(\cdot,\,\boldsymbol{\zeta})^{\top}\;\mathbf{G}_{\mathcal{S}}(\theta_{t,\mathbf{x}})^{-1}\,\nabla_{\boldsymbol{\theta}}\bar{\alpha}(\cdot,\,\boldsymbol{\zeta})\Big] = C_{\alpha}(\varphi_t(\mathbf{x}))\,. \tag{23}$$

Combining (21) and (23) gives (6). However, it is to noted that for analytical AFs, this step (Jensen's inequality) is an equality since that integrand is independent of the random base samples $\boldsymbol{\zeta}$. Any remaining slack is bound comes from the Cauchy–Schwarz step. Equality holds if and only if $\mathbf{u} = 0$ or every column of $\mathbf{A}$ is a scalar multiple of $\mathbf{u}$.

D.3 Gaussian Predictive Distribution

Corollary D.2 (Pullback FIM for a Gaussian Predictive Distribution)**.** *Let Assumption 4.1 hold and suppose the predictive distribution is univariate Gaussian, $p(y|\mathbf{x},\mathcal{D}_t) = \mathcal{N}\big(\mu_{t,\mathbf{x}},\Sigma_{t,\mathbf{x}}\big)$, where $\mu_{t,\mathbf{x}} \in \mathbb{R}$ is the posterior mean and $\Sigma_{t,\mathbf{x}} > 0$ is the posterior variance, parameterized by $\boldsymbol{\theta}_{t,\mathbf{x}} = \big(\mu_{t,\mathbf{x}},\Sigma_{t,\mathbf{x}}\big) \in \mathcal{S} = \mathbb{R}\times\mathbb{R}_+$. The Fisher–Rao metric at this point on the predictive distribution manifold is*

$$\mathcal{I}_{\mathcal{S}}\big(\boldsymbol{\theta}_{t,\mathbf{x}}\big) = \mathrm{diag}\big(\Sigma_{t,\mathbf{x}}^{-1},\;\tfrac{1}{2}\,\Sigma_{t,\mathbf{x}}^{-2}\big)\,, \tag{24}$$

and the pullback tensor (5) *reduces to*

$$\mathbf{G}_t(\mathbf{x}) = \frac{\nabla\mu_{t,\mathbf{x}}\,\nabla\mu_{t,\mathbf{x}}^{\top}}{\Sigma_{t,\mathbf{x}}} + \frac{\nabla\Sigma_{t,\mathbf{x}}\,\nabla\Sigma_{t,\mathbf{x}}^{\top}}{2\,\Sigma_{t,\mathbf{x}}^2}, \tag{25}$$

where ∇ denotes the gradient with respect to $\mathbf{x}$.

The first term, $\nabla\mu_{t,\mathbf{x}}\nabla\mu_{t,\mathbf{x}}^{\top}/\Sigma_{t,\mathbf{x}}$, is large where the posterior mean changes rapidly relative to uncertainty, so it captures the local exploitation geometry. The second term, $\nabla\Sigma_{t,\mathbf{x}}\nabla\Sigma_{t,\mathbf{x}}^{\top}/(2\Sigma_{t,\mathbf{x}}^2)$, is large where the posterior variance changes rapidly, so it captures the local exploration geometry.

Proof of Corollary D.2. The log-likelihood of $\mathcal{N}(y;\mu_{t,\mathbf{x}},\Sigma_{t,\mathbf{x}})$ is $\log p = -\frac{1}{2}\log(2\pi\Sigma_{t,\mathbf{x}}) - (y-\mu_{t,\mathbf{x}})^2/(2\Sigma_{t,\mathbf{x}})$. Differentiating with respect to the distribution parameters $\boldsymbol{\theta} = (\mu_{t,\mathbf{x}},\Sigma_{t,\mathbf{x}})$ gives the score functions

$$\frac{\partial\log p}{\partial\mu_{t,\mathbf{x}}} = \frac{y-\mu_{t,\mathbf{x}}}{\Sigma_{t,\mathbf{x}}}, \qquad \frac{\partial\log p}{\partial\Sigma_{t,\mathbf{x}}} = \frac{(y-\mu_{t,\mathbf{x}})^2}{2\Sigma_{t,\mathbf{x}}^2} - \frac{1}{2\Sigma_{t,\mathbf{x}}}.$$

The off-diagonal terms would be zero [13], and the FIM is a diagonal matrix

$$\mathcal{I}_{\mathcal{S}}(\boldsymbol{\theta}_{t,\mathbf{x}}) = \operatorname{diag}\big(\Sigma_{t,\mathbf{x}}^{-1},\ \tfrac{1}{2}\Sigma_{t,\mathbf{x}}^{-2}\big)\,.$$

The posterior map φ_t has Jacobian $J_t(\mathbf{x}) = \big[\nabla\mu_{t,\mathbf{x}},\ \nabla\Sigma_{t,\mathbf{x}}\big]^\top \in \mathbb{R}^{2\times D_x}$, where each row is the gradient of one parameter with respect to $\mathbf{x}$. Substituting into (5):

$$\begin{aligned}\mathbf{G}_t(\mathbf{x}) &= J_t^\top \operatorname{diag}\big(\Sigma_{t,\mathbf{x}}^{-1},\ \tfrac{1}{2}\Sigma_{t,\mathbf{x}}^{-2}\big)\, J_t \\ &= \Sigma_{t,\mathbf{x}}^{-1}\,\nabla\mu_{t,\mathbf{x}}\,\nabla\mu_{t,\mathbf{x}}^\top + \tfrac{1}{2}\Sigma_{t,\mathbf{x}}^{-2}\,\nabla\Sigma_{t,\mathbf{x}}\,\nabla\Sigma_{t,\mathbf{x}}^\top.\end{aligned}$$

□

D.4 Acquisition Sensitivity under a Scalar Gaussian Predictive

Substituting $\mathcal{I}_{\mathcal{S}}^{-1} = \operatorname{diag}(\Sigma_{t,\mathbf{x}},\, 2\Sigma_{t,\mathbf{x}}^2)$ from (24) into the definition of C_α in (23) gives

$$C_\alpha(\boldsymbol{\theta}_t) = \mathbb{E}_{\boldsymbol{\zeta}}\left[\Sigma_{t,\mathbf{x}}\Big(\frac{\partial\bar{\alpha}}{\partial\mu_{t,\mathbf{x}}}\Big)^2 + 2\Sigma_{t,\mathbf{x}}^2\Big(\frac{\partial\bar{\alpha}}{\partial\Sigma_{t,\mathbf{x}}}\Big)^2\right]. \tag{26}$$

For both analytic and Monte Carlo AFs, boundedness of C_α reduces to controlling partial derivatives of $\bar{\alpha}$ with respect to $(\mu_{t,\mathbf{x}}, \Sigma_{t,\mathbf{x}})$.

Analytic acquisition functions. Let $z_t = (\mu_{t,\mathbf{x}} - y^\star)/\sqrt{\Sigma_{t,\mathbf{x}}}$, Ω and ω be the cumulative distribution function (CDF) and probability density function (PDF) of the standard normal distribution.

Analytic UCB. For Upper Confidence Bound (UCB) [16, 42], $\bar{\alpha} = \mu_{t,\mathbf{x}} + \sqrt{\kappa_{\mathrm{UCB}}\Sigma_{t,\mathbf{x}}}$ gives

$\partial\bar{\alpha}/\partial\mu_{t,\mathbf{x}} = 1$ and $\partial\bar{\alpha}/\partial\Sigma_{t,\mathbf{x}} = \sqrt{\kappa_{\mathrm{UCB}}}/(2\sqrt{\Sigma_{t,\mathbf{x}}})$, so

$$C_\alpha^{\mathrm{UCB}} = \Sigma_{t,\mathbf{x}}\big(1 + \tfrac{\kappa_{\mathrm{UCB}}}{2}\big)\,. \tag{27}$$

Analytic EI. For Expected Improvement (EI) [22, 43], $\bar{\alpha}^{\mathrm{EI}} = (\mu_{t,\mathbf{x}} - y^\star)\Omega(z_t) + \sqrt{\Sigma_{t,\mathbf{x}}}\,\omega(z_t)$ gives

$\partial z_t/\partial\Sigma_{t,\mathbf{x}} = -z_t/(2\Sigma_{t,\mathbf{x}})$, $\partial\bar{\alpha}^{\mathrm{EI}}/\partial\mu_{t,\mathbf{x}} = \Omega(z_t)$ and $\partial\bar{\alpha}^{\mathrm{EI}}/\partial\Sigma_{t,\mathbf{x}} = \omega(z_t)/(2\sqrt{\Sigma_{t,\mathbf{x}}})$. Using $\Omega \le 1$ and $\omega \le 1/\sqrt{2\pi}$.

$$C_\alpha^{\mathrm{EI}} = \Sigma_{t,\mathbf{x}}\big(\Omega(z_t)^2 + \tfrac{1}{2}\omega(z_t)^2\big) \le \Sigma_{t,\mathbf{x}}\big(1 + \tfrac{1}{4\pi}\big)\,, \tag{28}$$

Analytic PI. For Probability of Improvement (PI) [22], $\bar{\alpha}^{\mathrm{PI}} = \Omega(z_t)$ gives

$\partial\bar{\alpha}^{\mathrm{PI}}/\partial\mu_{t,\mathbf{x}} = \omega(z_t)/\sqrt{\Sigma_{t,\mathbf{x}}}$ and $\partial\bar{\alpha}^{\mathrm{PI}}/\partial\Sigma_{t,\mathbf{x}} = -z_t\omega(z_t)/(2\Sigma_{t,\mathbf{x}})$. Since $e^{-z^2}(1 + z^2/2) \le 1$ for all $z \in \mathbb{R}$,

$$C_\alpha^{\mathrm{PI}} = \omega(z_t)^2\Big(1 + \tfrac{z_t^2}{2}\Big) \le \tfrac{1}{2\pi}, \tag{29}$$

For bounded $\Sigma_{t,\mathbf{x}}$, the AF sensitivities in (27), (28), and (29) are bounded.

Analytic LogEI. For analytic LogEI [1], applying the chain rule to the logarithm of EI gives

$$C_\alpha^{\mathrm{LogEI}}(\boldsymbol{\theta}) = \frac{C_\alpha^{\mathrm{EI}}(\boldsymbol{\theta})}{\bar{\alpha}^{\mathrm{EI}}(\boldsymbol{\theta})^2}. \tag{30}$$

Monte Carlo sensitivity bounds. The MC column of Table 1 follows from bounds on the sample-wise parameter derivatives. The next lemma isolates the shared calculation for MC-EI and smoothed MC-PI.

Lemma D.3 (Monte Carlo sensitivity bound)**.** *Let Assumption 4.1 hold. At a fixed scalar Gaussian predictive state $\boldsymbol{\theta} = (\mu, \Sigma)$ with $\Sigma > 0$, let ζ be a scalar standard-normal base sample. Suppose there is a finite scalar constant $L_{\bar{\alpha}} > 0$, independent of ζ, such that almost surely*

$$\left|\frac{\partial\bar{\alpha}(\boldsymbol{\theta},\zeta)}{\partial\mu}\right| \le L_{\bar{\alpha}}, \qquad \left|\frac{\partial\bar{\alpha}(\boldsymbol{\theta},\zeta)}{\partial\Sigma}\right| \le \frac{L_{\bar{\alpha}}\,|\zeta|}{2\sqrt{\Sigma}}. \tag{31}$$

Then

$$C_\alpha(\boldsymbol{\theta}) \le \tfrac{3}{2}L_{\bar{\alpha}}^2\Sigma. \tag{32}$$

Proof. Substituting the two derivative bounds into (26) gives

$$C_\alpha(\boldsymbol{\theta}) \le L_{\bar{\alpha}}^2 \Sigma + \tfrac{1}{2} L_{\bar{\alpha}}^2 \Sigma\, \mathbb{E}_\zeta[\zeta^2] = \tfrac{3}{2} L_{\bar{\alpha}}^2 \Sigma,$$

since a scalar standard-normal variable has second moment one. □

MC-EI. The ReLU utility in Table 4 has derivative bounded by one almost everywhere [17, 23]. Its Gaussian reparameterization satisfies both conditions with $L_{\bar{\alpha}} = 1$, giving $C_\alpha^{\text{MC-EI}}(\boldsymbol{\theta}) \le 3\Sigma/2$.

Smoothed MC-PI. The sigmoid utility with temperature τ in Table 4 has derivative bounded by $1/(4\tau)$ [17]. Taking $L_{\bar{\alpha}} = 1/(4\tau)$ gives $C_\alpha^{\text{MC-PI}}(\boldsymbol{\theta}) \le 3\Sigma/(32\tau^2)$. This is the sensitivity of the smoothed utility, not analytic PI.

MC-UCB. For the representation in Table 4, direct differentiation gives

$$\frac{\partial \bar{\alpha}^{\text{UCB}}(\boldsymbol{\theta}, \zeta)}{\partial \mu} = 1, \qquad \frac{\partial \bar{\alpha}^{\text{UCB}}(\boldsymbol{\theta}, \zeta)}{\partial \Sigma} = \frac{\sqrt{\kappa_{\text{UCB}}\pi/2}\,|\zeta|}{2\sqrt{\Sigma}}.$$

The lemma applies with $L_{\bar{\alpha}} = \max\{1, \sqrt{\kappa_{\text{UCB}}\pi/2}\}$. However, substituting these derivatives directly into (26) gives the sharper identity $C_\alpha^{\text{MC-UCB}}(\boldsymbol{\theta}) = \Sigma(1 + \kappa_{\text{UCB}}\pi/4)$.

Remark (Extension to multivariate outputs and parallel BO)**.** *For a multivariate Gaussian predictive distribution $\mathcal{N}(\boldsymbol{\mu}_t,\, \boldsymbol{\Sigma}_t)$ with $\boldsymbol{\mu}_t \in \mathbb{R}^{D_y}$ and $\boldsymbol{\Sigma}_t \in \mathbb{R}^{D_y \times D_y}$ (as in Multi-Objective Bayesian Optimization (MOBO) with D_y correlated outputs), the Fisher–Rao metric takes the block-diagonal form $\mathcal{I}_{\mathcal{S}} = \operatorname{diag}\big(\boldsymbol{\Sigma}_t^{-1},\, \frac{1}{2}\boldsymbol{\Sigma}_t^{-1} \otimes \boldsymbol{\Sigma}_t^{-1}\big)$ [13], and* (25) *extends to*

$$\mathbf{G}_t(\boldsymbol{X}) = J_{\boldsymbol{\mu}}^\top\, \boldsymbol{\Sigma}_t^{-1}\, J_{\boldsymbol{\mu}} + \tfrac{1}{2}\, J_{\boldsymbol{\Sigma}}^\top \big(\boldsymbol{\Sigma}_t^{-1} \otimes \boldsymbol{\Sigma}_t^{-1}\big)\, J_{\boldsymbol{\Sigma}}, \tag{33}$$

where $J_{\boldsymbol{\mu}}$ and $J_{\boldsymbol{\Sigma}}$ are the Jacobians of the vectorized mean and covariance with respect to $\boldsymbol{X}$.

We limit our discussion to the scalar case for simplicity. However, following Wang et al. [39] or Balandat et al. [15], one can apply a similar analysis to obtain a bound for the acquisition sensitivity $C_\alpha^{(q)}(\boldsymbol{\theta}_{t,\boldsymbol{X}})$ for Multi-Objective Bayesian Optimization or parallel BO with $q > 1$.

D.5 Pullback FIM Trace Under the Critical Lengthscale

Distance concentration. We assume that the initial training points $\mathbf{x}_i$ for $i = 1, \dots, N$ are drawn i.i.d. from $\text{Uniform}[0,1]^{D_x}$. Let $d := \|\mathbf{x} - \mathbf{x}_n\|$. Since $\mathbb{E}\big[(x_j - x_{n,j})^2\big] = (x_j - \frac{1}{2})^2 + \frac{1}{12}$ for $x_{n,j} \sim \text{Uniform}[0,1]$, we have

$$\frac{d}{\sqrt{D_x}} \xrightarrow{\text{a.s.}} \sqrt{v_{\mathbf{x}}}, \qquad v_{\mathbf{x}} := \frac{1}{D_x} \sum_{j=1}^{D_x} \big[(x_j - \tfrac{1}{2})^2 + \tfrac{1}{12}\big] \in \big[\tfrac{1}{12},\, \tfrac{1}{3}\big]\,.$$

Hence $d = \Theta(\sqrt{D_x})$ almost surely. Here $f = \Theta(g)$ means there exist $c_1, c_2 > 0$ and d_0 such that $c_1 g(d) \le f(d) \le c_2 g(d)$ for all $d \ge d_0$.

FIM trace for the SE kernel. For a GP with SE kernel $k_\lambda(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\big(-\|\mathbf{x} - \mathbf{x}'\|^2/(2\lambda^2)\big)$, let $K_{nn'} = k_\lambda(\mathbf{x}_n, \mathbf{x}_{n'})$ and $\boldsymbol{\alpha} = (K + \sigma_\varepsilon^2 I)^{-1}\mathbf{y}$. Then

$$\nabla_{\mathbf{x}} \mu_t(\mathbf{x}) = \sum_{n=1}^{N} \alpha_n\, \nabla_{\mathbf{x}} k_\lambda(\mathbf{x}, \mathbf{x}_n), \qquad \nabla_{\mathbf{x}} k_\lambda(\mathbf{x}, \mathbf{x}_n) = -\frac{k_\lambda(\mathbf{x}, \mathbf{x}_n)}{\lambda^2}\,(\mathbf{x} - \mathbf{x}_n).$$

By Corollary D.2,

$$\operatorname{tr}(\mathbf{G}_t) = \frac{\|\nabla \mu_t\|^2}{\Sigma_t} + \frac{\|\nabla \Sigma_t\|^2}{2\Sigma_t^2},$$

and both terms scale with k_λ/λ^2. Substituting $\lambda = c D_x^p$ and $d = \Theta(\sqrt{D_x})$ gives

$$k_\lambda(\mathbf{x}, \mathbf{x}_n) \xrightarrow{\text{a.s.}} \sigma_f^2 \exp\Big(-\frac{v_{\mathbf{x}}}{2c^2}\, D_x^{1-2p}\Big)\,.$$

This results in three cases:

Case $p < 1/2$**.** The exponent $v_{\mathbf{x}} D_x^{1-2p}/(2c^2) \to \infty$, so $k_\lambda \to 0$ exponentially. Hence $\nabla_{\mathbf{x}} k_\lambda \to 0$ exponentially, and therefore $\nabla \mu_t \to 0$ and $\mathrm{tr}(\mathbf{G}_t) \to 0$ exponentially.

Case $p > 1/2$**.** The exponent $\to 0$, so $k_\lambda \to \sigma_f^2$ (constant). The gradient $\nabla_{\mathbf{x}} k_\lambda \propto k_\lambda \cdot (\mathbf{x} - \mathbf{x}_n)/\lambda^2$. Since each coordinate difference is $\Theta(1)$ while $\lambda^2 = c^2 D_x^{2p}$, each entry scales as $\Theta(D_x^{-2p})$ and the full gradient norm scales as $\Theta(D_x^{1/2-2p}) \to 0$. Hence $\nabla \mu_t \to 0$ and $\mathrm{tr}(\mathbf{G}_t) \to 0$ at a polynomial rate.

Case $p = 1/2$**.** The exponent converges to $v_{\mathbf{x}}/(2c^2) \in (0, \infty)$, so $k_\lambda \to \gamma_c := \sigma_f^2 e^{-v_{\mathbf{x}}/(2c^2)} \in (0, \sigma_f^2)$. The Gram matrix concentrates to $K_\infty = (\sigma_f^2 + \sigma_\varepsilon^2 - \bar{\gamma})I + \bar{\gamma}\,\mathbf{1}\mathbf{1}^\top$, where $\bar{\gamma} = \sigma_f^2 e^{-1/(12c^2)}$ is the pairwise limit for i.i.d. training pairs. Since $\lambda_{\min}(K_\infty) > 0$, we have $\boldsymbol{\alpha} = \Theta(1)$. Each entry of $\nabla_{\mathbf{x}} k_\lambda$ scales as $\Theta(D_x^{-1})$, so the full gradient norm scales as $\Theta(D_x^{-1/2})$ and summing D_x squared terms gives $\mathrm{tr}(\mathbf{G}_t) = \Theta(D_x^{-1})$.

In all three cases, the trace of $\mathbf{G}_t$ is governed solely by whether k_λ is able to model meaningful correlation or not. $\lambda = \Theta(\sqrt{D_x})$ is the unique scaling for which $\mathrm{tr}(\mathbf{G}_t)$ stays at the critical $\Theta(D_x^{-1})$ order instead of decaying faster.

FIM trace at $p = 1/2$ and acquisition gradient bound. With $\boldsymbol{\alpha} = \Theta(1)$ and $\gamma_c = \Theta(1)$, substituting into Proposition 4.2 gives

$$\|\nabla_{\mathbf{x}} \alpha\|^2 \;\le\; C_\alpha \cdot \Theta(D_x^{-1}).$$

Since C_α is uniformly bounded in $\mathbf{x}$ and D_x for standard acquisitions (Appendix D.3), the universal bound scales as $\Theta(D_x^{-1})$ rather than collapsing exponentially. For $p < 1/2$, the trace collapses exponentially, while for $p > 1/2$ it decays polynomially. In both off-critical regimes, the AF gradient norm still vanishes.

Matérn kernels. The same argument applies to isotropic Matérn-ν kernel, since it depends on inputs only through $d = \|\mathbf{x} - \mathbf{x}'\|/\lambda$ and the same concentration gives $d \to \sqrt{v_{\mathbf{x}} D_x}/\lambda$. Thus the non-degenerate scaling remains $\lambda = \Theta(\sqrt{D_x})$. The difference appears when $\lambda \ll \sqrt{D_x}$: using $k_\nu(d) \sim d^{\nu-1/2} e^{-\sqrt{2\nu}\,d}$ as $d \to \infty$, the trace decays as $\exp(-c_\nu \sqrt{D_x}/\lambda)$, whereas for the SE kernel it decays as $\exp(-c_1 D_x/\lambda^2)$. Since $D_x/\lambda^2 \gg \sqrt{D_x}/\lambda$ in that regime, the SE trace collapses strictly faster, matching the empirical and theoretical results documented by Xu et al. [7].

D.6 Connection between TuRBO and FITR

Consider a GP with SE kernel with Automatic Relevance Determination (ARD) $k_\lambda(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\!\big(-\sum_j (x_j - x_j')^2/(2\lambda_j^2)\big)$ evaluated at the trust-region center $\mathbf{x}_c$.

At a local optimum of μ_t, $\nabla \mu_t(\mathbf{x}_c) \approx 0$, so the mean-gradient term in the pullback FIM (Corollary D.2) vanishes. The diagonal of $G_t(\mathbf{x}_c)$ is then dominated by the variance-gradient term:

$$[G_t(\mathbf{x}_c)]_{jj} \approx \frac{1}{2\Sigma_t(\mathbf{x}_c)^2} \left(\frac{\partial \Sigma_t}{\partial x_j}\right)^2.$$

For the SE kernel, $\partial k_\lambda(\mathbf{x}, \mathbf{x}')/\partial x_j = -(x_j - x_j')k_\lambda(\mathbf{x}, \mathbf{x}')/\lambda_j^2$, so $\partial \Sigma_t/\partial x_j$ scales proportionally to λ_j^{-2}. Using the definition of the pullback tensor, the diagonal entry $[G_t(\mathbf{x}_c)]_{jj}$ then scales like λ_j^{-4}. The ellipsoid induced by the pullback tensor is therefore longer in dimensions with longer lengthscales and shorter in dimensions with shorter lengthscales:

$$\{(\mathbf{x} - \mathbf{x}_c)^\top \mathrm{diag}(G_t(\mathbf{x}_c))(\mathbf{x} - \mathbf{x}_c) \le \delta^2\}.$$

The radius in dimension j is proportional to $[G_t]_{jj}^{-1/2}$. TuRBO's trust region $\{|x_j - x_{c,j}| \le L\lambda_j\}$ is not an exact diagonal-FIM box, but it preserves the same qualitative ordering: directions with larger ARD lengthscales receive wider search lengths. This motivates extending TuRBO to a location-dependent trust region whose scaling is determined by predictive sensitivities rather than explicit kernel lengthscales.

D.7 Local KL Expansion for Pullback-Fisher Trust Regions

This subsection gives the local probabilistic interpretation of the trust-region ellipsoid from Section 5. The key point is that the full pullback Fisher tensor defines the second-order term of the KL divergence between nearby predictive posteriors, so a trust region based on this tensor limits local predictive change rather than raw Euclidean displacement.

Fix an iteration t and center $\mathbf{x}_{t,c} \in \mathcal{X}$. Let

$$\boldsymbol{\theta}_{t,c} := \varphi_t(\mathbf{x}_{t,c}), \qquad \delta := \mathbf{x} - \mathbf{x}_{t,c}.$$

Consider the KL divergence between the predictive posterior at the center and the predictive posterior at a nearby input:

$$\mathcal{K}_t(\delta) := D_{\mathrm{KL}}(p(y; \boldsymbol{\theta}_{t,c}) \,\|\, p(y; \varphi_t(\mathbf{x}_{t,c} + \delta))) \,. \tag{34}$$

Step 1: KL expansion in parameter space. For a smooth parametric family $p(y; \boldsymbol{\theta})$, the KL divergence admits the standard second-order expansion around $\boldsymbol{\theta}_{t,c}$ [13, 24]:

$$D_{\mathrm{KL}}(p(y; \boldsymbol{\theta}_{t,c}) \,\|\, p(y; \boldsymbol{\theta}_{t,c} + \Delta\boldsymbol{\theta})) = \tfrac{1}{2}\Delta\boldsymbol{\theta}^\top \mathcal{I}_{\mathcal{S}}(\boldsymbol{\theta}_{t,c})\Delta\boldsymbol{\theta} + o(\|\Delta\boldsymbol{\theta}\|^2). \tag{35}$$

Step 2: Pull back to input space. Since φ_t is C^1 by Assumption 4.1, we have the first-order expansion

$$\varphi_t(\mathbf{x}_{t,c} + \delta) = \boldsymbol{\theta}_{t,c} + J_t(\mathbf{x}_{t,c})\,\delta + o(\|\delta\|). \tag{36}$$

Substituting

$$\Delta\boldsymbol{\theta} = J_t(\mathbf{x}_{t,c})\,\delta + o(\|\delta\|)$$

into (35) gives

$$\begin{aligned} \mathcal{K}_t(\delta) &= \tfrac{1}{2}\delta^\top J_t(\mathbf{x}_{t,c})^\top \mathcal{I}_{\mathcal{S}}(\boldsymbol{\theta}_{t,c}) J_t(\mathbf{x}_{t,c})\delta + o(\|\delta\|^2) \\ &= \tfrac{1}{2}\delta^\top \mathbf{G}_t(\mathbf{x}_{t,c})\delta + o(\|\delta\|^2), \end{aligned} \tag{37}$$

where the second line uses Definition 4.1.

Equation (37) shows that the ellipsoid

$$\delta^\top \mathbf{G}_t(\mathbf{x}_{t,c})\delta \le \kappa_t^2 \tag{38}$$

is, to second order, a local KL trust region around the predictive posterior at $\mathbf{x}_{t,c}$. This interpretation applies to the full pullback Fisher tensor. The FITR implementation in Section 5 uses a diagonal, regularized approximation of (38) to obtain an axis-aligned box that is computationally compatible with TuRBO-style trust-region updates.

D.8 Pullback FIM Diagonal via Backpropagation

The estimator (11) computes the diagonal of the pullback Fisher tensor by differentiating the log predictive likelihood with respect to the input $\mathbf{x}$ rather than the distribution parameters $\boldsymbol{\theta}$. The connection follows from the chain rule and the definition of the FIM.

By the chain rule, the partial derivative of the log predictive likelihood with respect to input coordinate x_j is

$$\frac{\partial}{\partial x_j} \log p\big(y; \varphi_t(\mathbf{x})\big) = \big(\nabla_{\boldsymbol{\theta}} \log p(y; \boldsymbol{\theta})\big)^\top \frac{\partial \varphi_t(\mathbf{x})}{\partial x_j}. \tag{39}$$

Squaring both sides and taking the expectation over $y \sim p(y; \varphi_t(\mathbf{x}))$,

$$\mathbb{E}\left[\left(\frac{\partial}{\partial x_j} \log p\big(y; \varphi_t(\mathbf{x})\big)\right)^2\right] = \left(\frac{\partial \varphi_t}{\partial x_j}\right)^\top \underbrace{\mathbb{E}\big[\nabla_{\boldsymbol{\theta}} \log p \, \nabla_{\boldsymbol{\theta}} \log p^\top\big]}_{=\mathcal{I}_{\mathcal{S}}(\varphi_t(\mathbf{x}))} \left(\frac{\partial \varphi_t}{\partial x_j}\right) = \big[\mathbf{G}_t(\mathbf{x})\big]_{jj}, \tag{40}$$

where the last equality uses the definition of the FIM (3) and the pullback tensor (5). The diagonal entry of the pullback Fisher tensor is therefore the expected squared partial derivative of the log predictive likelihood with respect to x_j, computed at the predictive posterior $\varphi_t(\mathbf{x})$.

Replacing the expectation by an average over S draws $\tilde{y}_s \sim p(y; \varphi_t(\mathbf{x}))$ and evaluating the partial derivatives by backpropagation through the log-likelihood gives the estimator (11). This is the outer-product-of-gradients form of the FIM, similar to [25], applied to the input coordinates $\mathbf{x}$ instead of the distribution parameters $\boldsymbol{\theta}$.

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 abstract and introduction clearly state the claims made, including the contributions made in the paper. The Figure 1 and lines 41-52 in the introduction section and lines 2-3, 6-10 in the abstract specify the main claims of the paper.

   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 paper discusses the limitations of the work in the section 5 and also in the Limitations section.

   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: [Yes]

Justification: The paper provides the full set of assumptions and a complete proof for the theoretical results. We provide a proof for the main theorem in the appendix.

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: The pseudocode and the experiment setup fully discloses all the information needed to reproduce the main experimental results of the paper. The used benchmark datasets are publicly available and very often used in the BO literature.

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: [No]

   Justification: The code is currently undergoing institutional legal review and will be released open-source (with the repository linked on arXiv) once cleared. All experiments rely entirely on open-access benchmarks and the open-source library BoTorch. Furthermore, the proposed methods, bounds, and experimental setups can be fully reproduced using the detailed pseudocode and implementation specifications provided in the paper.

   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: The paper specifies all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results. The experiment used the BoTorch package primarily. The default hyperparameters and training routines from BoTorch were used for the surrogate model training and AF optimization. The hyperparameters that were changed, have been specified in the paper.

   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 experiment plots show an average and min-max shaded region for the the different methods. The shaded region is calculated as the standard error from 11 runs for each method.

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: The experiments were run on an Nvidia GPU Tesla V100-SXM2-32GB. The experiments were run for 11 independent runs for each method. The BO budget for each iteration is provided in the experiment plots.

   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 conducted in the paper conforms, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`. The paper does not involve any human subjects or any sensitive data. All the experiments were run on a publicly available dataset and code was developed using open-source libraries.

   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: [N/A]

    Justification: This work discusses a theoretical aspect of BO and is useful for black-box optimization in general. However, we do not see any broader societal impact of this work.

    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: This work does not release any models or datasets that have a high risk for misuse.

    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: The paper properly credits the papers, opensource libraries and datasets used in the paper. The credit to the authors of code is explicitly mentioned in the code repository as well.

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: [N/A]

    Justification: This work does not introduce any new assets.

    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 work does not involve crowdsourcing nor research with 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 work does not involve research with 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: This work does not involve LLMs as any important, original, or non-standard components.

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.