High-dimensional Level Set Estimation with Trust Regions and Double Acquisition Functions

Giang Ngo **Dat Phan Trong** **Dang Nguyen** **Sunil Gupta**
Applied Artificial Intelligence Initiative, Deakin University

Abstract

Level set estimation (LSE) classifies whether an unknown function's value exceeds a specified threshold for given inputs, a fundamental problem in many real-world applications. In active learning settings with limited initial data, we aim to iteratively acquire informative points to construct an accurate classifier for this task. In high-dimensional spaces, this becomes challenging where the search volume grows exponentially with increasing dimensionality. We propose TRLSE, an algorithm for high-dimensional LSE, which identifies and refines regions near the threshold boundary with dual acquisition functions operating at both global and local levels. We provide a theoretical analysis of TRLSE's accuracy and show its superior sample efficiency against existing methods through extensive evaluations on multiple synthetic and real-world LSE problems.

1 INTRODUCTION

Given a threshold h and a black-box function f, the *superlevel* (or *sublevel*) set of f comprises all points $\mathbf{x}$ where $f(\mathbf{x}) \geq h$ (or $f(\mathbf{x}) < h$). Building a classifier to classify points into either set is challenging when f is an expensive-to-evaluate black-box function leading to limited training data. Examples of such applications include computational chemistry (Tran et al., 2021), environmental monitoring (Gotovos et al., 2013), and vehicle structure design (Ngo et al., 2024), where data collection is costly. An active learning solution to LSE needs to iteratively evaluate informative points to improve the classifier. Thus, sample efficiency is a key objective when designing LSE algorithms, alongside achieving high classification accuracy.

LSE algorithms select the next point by optimizing an *acquisition function* (AF), which quantifies utility for the classifier using a surrogate model (typically a Gaussian Process (GP)) fitted on existing data points (Zanette et al., 2018). In high-dimensional LSE (HDLSE), this scheme is challenged by GP's performance degradation in high-dimensional spaces, but a recent work by Hvarfner et al. (2024) shows that adjusting GP kernel lengthscales can significantly alleviate this issue. However, challenges in HDLSE go beyond GP modeling. First, optimizing the AF, often a non-convex problem, becomes harder as the search space expands exponentially. Second, GP inference scales cubically with data size, making AF computation slow as HDLSE accumulates more points.

Bayesian Optimization (BO), a related problem which optimizes f instead, faces similar challenges in a high-dimensional setting. Local BO methods (Eriksson et al., 2019; Daulton et al., 2022) have become a popular solution in this setting. They optimize the AF over local trust regions (TRs) instead of the entire global search space, and each TR is accurately modeled by its own local GP. Multiple TRs are first randomly initialized, and their expansion/shrinkage is dictated by whether they can improve upon the incumbent solution. A TR without improvement will be replaced by a randomly initialized one to ensure eventual global optimization. An advantage of local BO is that optimizing the AF over a bounded local region is easier than over the entire space, with less over-exploration and faster computation.

However, adapting the local BO framework to HDLSE is not straightforward due to fundamental differences between the two problems. First, defining the success of a trust region (TR) for LSE is non-trivial. TR performance in BO is tied to improvement over the incumbent solution, but measuring such TR performances in LSE is unclear, complicating the ability to expand, shrink, or replace underperforming TRs ef-

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300. Copyright 2026 by the author(s).

fectively. Second, it is also ambiguous where a TR's center should be located. In BO, TRs are naturally centered at the incumbent solutions or the optima of a confidence bound. No such concepts exist in LSE, leaving the placement of TRs' centers uncertain. Finally, local BO only focuses on promising regions to find the next best solution inside those regions, while LSE demands accurate classification across the entire space. This creates a significant gap: while points within TRs might be well-modeled and classified, those outside the TRs, despite being the majority since TRs are often small, are neglected.

We address the challenges above of adapting local BO for HDLSE tasks with a novel method, referred to as **Trust-region Level Set Estimation** (TRLSE). To reveal the location of the level sets, it is important to identify the *threshold boundary* region where $f(\mathbf{x}) \approx h$ helps reveal the location of the level sets. Therefore, we focus the TRs at this boundary and expand a region if it is located around the threshold boundary. TRLSE employs two levels of acquisition functions: a *global* AF to identify regions likely to contain the threshold boundary, and a *local* AF within each TR to capture boundary details with finer resolution. The global AF helps revealing the threshold boundary over the entire space, extending accurate classification to outside the local TRs. Optimizing a local AF over bounded local regions enjoys the benefits mentioned above, including less over-exploration, faster computation, and a better model for the threshold boundary, which is often the most uncertain region to classify in an LSE problem.

Our contributions are summarized as follows:

1. We propose TRLSE, a novel algorithm for HDLSE with multiple trust regions to reveal the threshold boundary and two levels of acquisition functions to guide region initialization and local sampling.
2. We theoretically analyze TRLSE, guaranteeing its classification accuracy outside the trust regions, which accounts for most of the search space.
3. We demonstrate TRLSE's superior performance on a variety of synthetic and real-world LSE problems across both low and high-dimensional search spaces (from 2 - 1000 dimensions).

2 RELATED WORKS

Level Set Estimation Most LSE methods focus on designing AFs that reveal information about the level sets of f. The Straddle heuristic (Bryan et al., 2005), LSE (Gotovos et al., 2013) (later referred to as ActiveLSE to avoid confusion with the setting's abbreviation), and C2LSE (Ngo et al., 2024) utilize confidence intervals to prioritize evaluating where the classification by the surrogate model is the most uncertain. These points are often near high-uncertainty regions and/or potential threshold boundaries. One-step lookahead mechanisms are also used; for example, TruVar (Bogunovic et al., 2016) considers the potential reduction in truncated variance while unifying LSE and BO, and RMILE (Zanette et al., 2018) prioritizes points expected to expand the superlevel set volume. While the idea sounds promising, AFs using one-step lookahead mechanism often suffer from high computational complexity. Other approaches include an information-theoretic acquisition (Nguyen et al., 2021) and sampling strategy based on experimental design (Mason et al., 2022). Despite these alternatives, the Straddle heuristic is popularly employed in many real-world applications due to strong empirical performance and adaptability to both discrete and continuous LSE problems. Nevertheless, optimizing AFs becomes increasingly challenging in high-dimensional spaces due to the exponential growth of the search space and the computational demands of GP surrogate models.

High-dimensional LSE HLSE (Ha et al., 2021) used BNN with Monte Carlo dropout as the surrogate model instead of a GP. While a BNN generally scales better to high-dimensional inputs than a GP, HLSE suffers from remarkably higher runtime due to an extensive, ongoing hyperparameter tuning and lacks theoretical guarantees for solution accuracy. Though not specifically targeting HDLSE, Shekhar and Javidi (2019) proposed a hierarchical space-partitioning method to explore the search space at different levels of detail with potentially smaller computational costs in higher dimensionalities. However, in truly high-dimensional spaces, the exponential growth in number of partitioned regions limits its effectiveness. The method also offers only theoretical convergence guarantees, lacking both rigorous empirical evaluation and practical guidance for hyperparameter selection.

High-dimensional BO Many high-dimensional BO methods rely on certain assumptions such as the existence of a representative low-dimensional subspace (Wang et al., 2016; Kirschner et al., 2019), an additive structure (Han et al., 2021; Rolland et al., 2018), or proximity of the optimum to the search space center (Oh et al., 2018). Without such assumptions, adjusting kernel lengthscales (Rana et al., 2017; Hvarfner et al., 2024) is also a common strategy which assumes a reduced complexity over f. Local BO (Zhao et al., 2022; Daulton et al., 2022) has become prominent recently with its scalability to large evaluation budgets while retaining the advantages of GPs as surrogate

models, with TuRBO (Eriksson et al., 2019) being one of the most widely adopted algorithms. Using multiple local trust regions avoids over-exploration due to smaller bounded search spaces which allows faster, more accurate acquisition optimization. However, local BO methods, designed for optimization, do not directly apply to LSE, which requires accurate classification of the entire input space rather than finding a single optimum.

3 METHODOLOGY

3.1 Problem Definition

Let $f : \mathcal{X} \longrightarrow \mathbb{R}$ be a black-box function where $\mathcal{X} \subset \mathbb{R}^d$ is a compact set. An LSE problem aims to classify any $\mathbf{x} \in \mathcal{X}$ into either the superlevel set $\mathcal{H} = \{\mathbf{x} \in \mathcal{X} : f(\mathbf{x}) \geq h\}$ and the sublevel set $\mathcal{L} = \{\mathbf{x} \in \mathcal{X} : f(\mathbf{x}) < h\}$ for a given threshold h. More information about f can be obtained through noisy function evaluations $y = f(\mathbf{x}) + \eta$ at $\mathbf{x}$ with observation noise $\eta \sim \mathcal{N}(0, \sigma^2)$.

3.2 Trust-region Level Set Estimation

We now present TRLSE, a method utilizing multiple local trust regions for high-dimensional LSE. The method steps are summarized in Algorithm 1, with a detailed description for each component presented in the following sections. TRLSE simultaneously maintains R trust regions at every time step, and in the beginning, they are initialized by creating R hypercubes of volume V_{init} centered at R data points sampled uniformly within the search space $\mathcal{X}$. At each iteration, the TRs are updated to center at the points deemed closest to the threshold boundary and to change their volume according to how well their function values range contains the threshold (see Section 3.2.1). Further, if a region is too "small", it will be replaced by a new region initialized by optimizing the global AF (see Section 3.2.2). The algorithm then proceeds to refine the local regions by choosing the next evaluation point through jointly optimizing the local AF over all TRs (see Section 3.2.2). Both AFs share the goal of balancing between exploring the search space and identifying the threshold boundary. The local AF allows a granular surrogate model of the threshold boundary, while the global one ensures coverage on the entire search space. It should be noted that the classification rules in the end of Algorithm 1 can be applied at any time step of the process, not necessarily at the end. These rules use the global GP to classify outside of the TRs and the local GPs to classify inside the TRs. If a point belongs to more than one TRs, it will be classified by the local GP that gives the lowest posterior variance.

Below is a notation list for the reader's convenience:

Algorithm 1: Trust-region level set estimation

Input: threshold h, budget B, V_{init}, V_{max}, #trust regions R, confidence parameter β.
Initialize R regions $[\mathcal{T}^i]_{i=1}^R$ of volume V_{init} centered at $D = [\mathcal{C}_0^i]_{i=1}^R$ with $\mathcal{C}_0^i \sim \text{Uniform}(\mathcal{X})$;
for $i = 1$ **to** R **do**
  Fit $\mathcal{GP}_0^i$ for $\mathcal{T}^i$ with $[\mathcal{C}_0^i, y_i]$, $y_i = f(\mathcal{C}_0^i) + \eta$;
$t \leftarrow 1; n \leftarrow R$; Fit $\mathcal{GP}_g$ with D;
while $|D| \leq B$ **do**
  for $i = 1$ **to** R **do**
    Update centroid $\mathcal{C}_t^i$, volume V_t^i, and length L_t^i of $\mathcal{T}^i$; `// See Section 3.2.1`
    Update $\mathcal{GP}_t^i$ with $\{\mathbf{x} \in D \cap [\mathcal{C}_t^i \pm L_t^i]\}$;
  for $i = 1$ **to** R **do**
    if $V_t^i < V_{init}/2$ **then**
      Discard $\mathcal{T}^i$;
      Select the new centroid $\mathcal{C}_t^i$ at $\bar{\mathbf{x}}_n = \text{argmax}_{\mathbf{x} \in \bar{\mathcal{U}}_t} a_g(\mathbf{x})$; `// See Eq.5`
      Evaluate f at $\bar{\mathbf{x}}_n : \bar{y}_n = f(\bar{\mathbf{x}}_n) + \eta$;
      Re-initialize $\mathcal{T}_i$ at $\mathcal{C}_t^i$ with volume V_{init};
      $D = D \cup (\bar{\mathbf{x}}_n, \bar{y}_n)$, update $\mathcal{GP}_g$ with D;
      $n \leftarrow n + 1$;
  Select $\mathbf{x}_t = \text{argmax}_{\mathbf{x} \in \mathcal{U}^t} a_l(\mathbf{x})$; `// See Eq.7`
  Get evaluation y_t of f at $\mathbf{x}_t$; $D = D \cup (\mathbf{x}_t, y_t)$;
  Update $\mathcal{GP}_g$ with D; $t \leftarrow t + 1$;

Output: Classification rules:
$\mathbf{x} \in \hat{\mathcal{H}}_T$ (predicted superlevel set) if $\mu_g(\mathbf{x}) \geq h$ if $\mathbf{x} \in \bar{\mathcal{U}}_T$ or $\mu_T^i(\mathbf{x}) \geq h$ if $\mathbf{x} \in \mathcal{T}^i$;
$\mathbf{x} \in \hat{\mathcal{L}}_T$ (predicted sublevel set) if $\mu_g(\mathbf{x}) < h$ if $\mathbf{x} \in \bar{\mathcal{U}}_T$ or $\mu_T^i(\mathbf{x}) < h$ if $\mathbf{x} \in \mathcal{T}^i$;

- $\mathcal{T}^i$: the i-th TR.
- $\mathcal{C}_t^i, V_t^i, L_t^i$: the centroid, volume, and lengths of $\mathcal{T}^i$ at the end of iteration t respectively.
- $\mathcal{GP}_t^i$: the local GP of $\mathcal{T}^i$ at iteration t (with posterior mean $\mu_t^i(\mathbf{x})$ and variance $(\sigma_t^i(\mathbf{x}))^2$).
- $\mathcal{U}_t = \bigcup_{i=1}^{R} \mathcal{T}^i$ and $\bar{\mathcal{U}}_t = \mathcal{X} \setminus \mathcal{U}_t$.
- D: the set of all evaluated data points.
- n: total number of TRs ever initialized.
- $\mathcal{GP}_g$: the global GP fitted with D (with posterior mean $\mu_g(\mathbf{x})$ and variance $\sigma_g^2(\mathbf{x})$ at $\mathbf{x}$).

3.2.1 Trust Region Update

Moving Region Centroid To solve the centering problem for TRs, TRLSE updates each TR to cover the threshold boundary as much as possible given the

available information about f. Thus, the first step is to move the centroid of $\mathcal{T}^i$ to the point deemed closest to the threshold boundary by $\mathcal{GP}^i_{t-1}$ as follows:

$$\mathcal{C}^i_t = \underset{\mathbf{x}\in\mathcal{T}^i}{\operatorname{argmin}} |\mu^i_{t-1}(\mathbf{x}) - h|. \tag{1}$$

Updating Volume and Length Next, we define the update rule for the volume of $\mathcal{T}^i$. TuRBO simply doubles the TR's volume when it experiences a series of consecutive progress upon the incumbent solution (and halves the TR's volume with a series of failures). To address the challenge of defining the success of a TR for LSE, we reward a TR if it centers at the threshold boundary (and shrinks it if it deviates from this boundary). To determine if $\mathcal{T}^i$ successfully brackets the threshold h, we define a penalty based on whether the confidence interval of the function's range within the TR contains h or not. Specifically, we first define the penalty for a region $\mathcal{T}^i$ at iteration t as:

$$\mathcal{P}_t(\mathcal{T}^i) = \Phi\left(\left|\frac{\bar{l}^i_{t-1} + \bar{u}^i_{t-1} - 2h}{2\bar{\sigma}^i_{t-1}}\right|\right) \tag{2}$$

with Φ being the standard normal CDF and:

$$\begin{aligned}
\bar{l}^i_{t-1} &= \min_{\mathbf{x}\in\mathcal{T}^i} \left(\mu^i_{t-1}(\mathbf{x}) - \beta\sigma^i_{t-1}(\mathbf{x})\right)\\
\bar{u}^i_{t-1} &= \max_{\mathbf{x}\in\mathcal{T}^i} \left(\mu^i_{t-1}(\mathbf{x}) + \beta\sigma^i_{t-1}(\mathbf{x})\right)\\
\bar{\sigma}^i_{t-1} &= \frac{\bar{u}^i_{t-1} - \bar{l}^i_{t-1}}{2\beta}
\end{aligned}$$

where β is the confidence parameter. Using both $\bar{l}^i_{t-1}$ and $\bar{u}^i_{t-1}$ allows the update to take uncertainty (within $\mathcal{T}^i$) into account while giving a rigorous estimate of the function value's range. We use Φ to limit the penalty to at most 1 to avoid a sudden, drastic change in region volume. Though this limit can slow down the discard of underperforming regions, it also protects newly created regions from being discarded prematurely, leaving a chance for local exploration. $\mathcal{P}_t(\mathcal{T}^i)$ increases when function values within $\mathcal{T}^i$ are far from h and decreases when they are close to h.

Instead of simply doubling/halving region volumes, we adjust them according to their penalty as follows:

$$V^i_t = \min(V^i_{t-1} S(\mathcal{P}_{t-1}(\mathcal{T}^i)), V_{max}) \tag{3}$$

where $S(\cdot)$ smoothly decreases from around 2 to 0 as the penalty $\mathcal{P}_t$ increases from 0.5 to 1 (the feasible value range of $\mathcal{P}_t$) so that $S(\mathcal{P}_t)$ serves as an adjustment factor for updating region volume. We discuss different choices for $S(\cdot)$ in Section D.5. Each TR is allowed to expand up to V_{max} to ensure it is sufficiently small for accurate acquisition optimization. Similar to TuRBO, the region lengths are updated as:

$$L^i_t = (V^i_t / \prod_j^d \lambda_j)^{1/d} \cdot (\lambda_1, \lambda_2, ..., \lambda_d)^T \tag{4}$$

where λ_k is the lengthscale for the k-th dimension of $\mathcal{GP}^i_{t-1}$. This allows the region to expand more along dimensions where the function is not so sensitive.

Updating Local GP Finally, $\mathcal{GP}^i_t$ is fitted with any existing training data point $\mathbf{x} \in D$ s.t. $\mathbf{x} \in [\mathcal{C}^i_t - L^i_t, \mathcal{C}^i_t + L^i_t]$ (i.e. the hyperrectangle of length $2L^i_t$ centered at $\mathcal{C}^i_t$). By using data points from immediately outside the TR, $\mathcal{GP}^i_t$ models the TR's border better, which can be inaccurate if only data points inside the TR are used.

3.2.2 Two Levels of Acquisition Functions

Finally, TRLSE employs a dual AF system to ensure global accuracy. The local AF refines the boundary within TRs, while the global AF helps re-initialize failing TRs at informative locations, preventing the algorithm from neglecting any part of the search space.

Global AF for TR suggestion A TR is discarded when its volume falls below $V_{init}/2$, suggesting it is unlikely to contain the threshold boundary. The data points collected within this region are retained in the training set. Given $n-1$ regions initialized so far, the centroid $\mathcal{C}^i_t$ of the new TR will be at:

$$\bar{\mathbf{x}}_n = \underset{\mathbf{x}\in\bar{\mathcal{U}}_t}{\operatorname{argmax}}\ a_g(\mathbf{x}) \tag{5}$$

with a_g being an AF (calculated using $\mathcal{GP}_g$) that increases its value at locations that are either associated with high uncertainty (i.e. exploration) or deemed close to the threshold boundary (i.e. exploitation). We choose the Straddle heuristic (Bryan et al., 2005), which can be easily optimized over continuous spaces with a low computational cost:

$$a_g(\mathbf{x}) = \beta\sigma_g(\mathbf{x}) - |\mu_g(\mathbf{x}) - h|. \tag{6}$$

The function value $\bar{y}_n$ at $\bar{\mathbf{x}}_n$ is evaluated, and $\mathcal{GP}^i_t$ is re-fitted as in the region update step. This replacement ensures no underperforming region remains for too long and becomes wasteful in the local acquisition optimization step, and that eventually all TRs remaining will focus on the threshold boundary.

Local AF for refining regions Local modeling is further enriched by evaluating points from within $\mathcal{U}_t$:

$$\mathbf{x}_t = \underset{x^i_l : 1\le i\le R}{\operatorname{argmax}}\ a^i_l \tag{7}$$

$$\text{where } a^i_l = \max_{\mathbf{x}\in\mathcal{T}_i} a_l(\mathbf{x}) \text{ and } x^i_l = \underset{\mathbf{x}\in\mathcal{T}_i}{\operatorname{argmax}}\, a_l(\mathbf{x})$$

where $a_l(\mathbf{x})$ is defined as $a_g(\mathbf{x})$ but uses the local posterior mean $\mu_t^i(\mathbf{x})$ and standard deviation $\sigma_t^i(\mathbf{x})$. Sampling inside the TRs allows a better model for the detected threshold boundary.

4 THEORETICAL ANALYSIS

We now theoretically analyze TRLSE's accuracy. All proofs can be found in Appendix A.

4.1 Ensuring Progress via Trust Region Re-initialization

TRLSE initializes regions for global coverage and samples within them for local accuracy, so its performance depends on both the number of regions initialized and samples collected. While the latter inherently increases over the iterations, it is not straightforward that the former will increase (i.e. new regions may not be initialized). We first show that underperforming TRs are surely discarded, which means initialization of new regions will happen.

Lemma 4.1. *Let $\beta \geq \Phi^{-1}(\psi)$ with $\psi \in (0.5, 1)$ and $S(u) = 2/(1 + \exp(au - b))$ with $b = \psi a$. Any region $\mathcal{T}^i$ where either $\bar{l}_{t-1}^i \geq h$ or $\bar{u}_{t-1}^i \leq h$ and remains so for the next iterations will be replaced after at most ζ iterations with $\zeta = \log(V_{init}/V_{max}^2)/(\log(2) - \log(1 + \exp(-b + a\Phi(\beta))))$.*

Since new TRs will be initialized, n does not get stuck. Suppose that when TRLSE terminates at iteration T, it has initialized N TRs. The accuracy of TRLSE can now be estimated by N and T.

4.2 Guarantee for Classification Accuracy

4.2.1 Accuracy Definition

We restate the definition for the accuracy of level set classification as follows:

Definition 4.2. Let $\hat{\mathcal{H}}$ and $\hat{\mathcal{L}}$ be the predicted superlevel and sublevel sets of an LSE algorithm $\mathbb{A}$ for a black-box function f over the search space $\mathcal{X}$ such that they are disjoint and that $\hat{\mathcal{H}} \cup \hat{\mathcal{L}} = \mathcal{X}$. Given a non-negative constant ϵ, the level set classification of a point $\mathbf{x} \in \mathcal{X}$ is said to be ϵ-accurate (or $\mathbf{x}$ is classified with ϵ-accuracy) if $h - f(\mathbf{x}) \leq \epsilon \ \forall \mathbf{x} \in \hat{\mathcal{H}}$ and $f(\mathbf{x}) - h \leq \epsilon \ \forall \mathbf{x} \in \hat{\mathcal{L}}$.

The constant ϵ is often predefined per application to indicate an acceptable error margin (e.g., if $h = 1.0$, an ϵ of 0.01 means that the practitioner accepts classification errors for points whose function values are within $[0.99, 1.01]$). For example, in safety-critical applications, ϵ may be set to a very small value to ensure high accuracy, while in less critical applications, a larger value may be acceptable to reduce the evaluation budget.

A point-wise definition of accuracy is a necessary adaptation for the continuous setting, moving beyond the definition by Gotovos et al. (2013) for predefined finite sets. It serves as a fundamental building block from which one can aggregate performance over a finite test set drawn from anywhere inside $\mathcal{X}$. Furthermore, a guarantee on the continuous space (e.g., "$\mathcal{X}$ is classified with ϵ-accuracy if the error for every point is less than ϵ"), while seemingly stronger, does not provide a clear mechanism for performance analysis on a finite test set, leading to a lack of practical utility.

4.2.2 Accuracy Guarantee for TRLSE

To analyze the accuracy convergence of TRLSE, we follow Srinivas et al. (2010) and assume that the unknown black-box function $f : \mathcal{X} \to \mathbb{R}$ is a sample path drawn from a Gaussian Process (GP) prior, denoted as $\mathcal{GP}(\mu_0(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$, where $\mu_0(\mathbf{x})$ is the mean function and $k(\mathbf{x}, \mathbf{x}')$ is a positive definite covariance (kernel) function. Without loss of generality, we assume the mean function is zero.

The following result shows the theoretical convergence for the level set accuracy of TRLSE:

Theorem 4.3 (Accuracy guarantee of TRLSE)**.** *Let TRLSE be run with the Straddle heuristic. Assume that there exist constants $\kappa_g \in (0, 1]$ such that $a_g(\bar{\mathbf{x}}_n) \geq \kappa_g \max_{\mathbf{x} \in \bar{\mathcal{U}}_t} a_g(\mathbf{x})$ for all $n \geq 1$. For any N that satisfies the condition $a_g(\bar{\mathbf{x}}_N) \leq \frac{1}{N} \sum_{n=1}^{N} a_g(\bar{\mathbf{x}}_n)$, it holds with a probability of at least $\Phi(\beta)$ that the classification of a point $\mathbf{x}$ outside the trust regions ($\mathbf{x} \in \bar{\mathcal{U}}_T$) is ϵ-accurate, where ϵ satisfies:*

$$\epsilon \leq \frac{\beta}{\kappa_g} \sqrt{\frac{C_1 \gamma_N}{N}}, \text{ with } C_1 = \frac{2}{\log(1 + \sigma^{-2})}. \quad (8)$$

Assuming the use of common kernels as in Theorem 5 in Srinivas et al. (2010) (i.e. linear, squared exponential, and Matérn kernels), the right hand side of Eq.8 converges to zero as N increases, which means the accuracy of TRLSE outside the TRs improves over time. Bounds for the maximum information gain γ_N w.r.t. the number of dimensions d and the amount of data points are given by Srinivas et al. (2010).

Theorem 4.3 guarantees the accuracy *outside* the TRs (i.e. $\forall \mathbf{x} \in \bar{\mathcal{U}}_T$), not inside them. In high-dimensional problems, the total volume occupied by the TRs maintained by TRLSE is often insignificant, as can be seen in Appendix B. Deriving a formal accuracy guarantee for points within the trust regions (i.e. for all $\mathbf{x} \in \mathcal{U}_T$) is a significant challenge, as their position, size, and local models change at every iteration. Given this

complexity and the limited practical interest in the accuracy over such insignificant volume inside the TRs, the focus remains on providing a foundational accuracy guarantee for the vast global space (as delivered by Theorem 4.3), which is the most critical measure of the algorithm's performance.

Proof sketch We first need to establish that the acquisition values $a_g(\bar{\mathbf{x}}_n)$ and $a_t(\mathbf{x}_t)$ are always non-negative under a mild assumption that all data points are not yet confidently classified by the algorithm. With $A_N = \frac{1}{N}\sum_{n=1}^{N} a_g(\bar{\mathbf{x}}_n)$, we can link the sum of the global acquisition values to the sum of the global posterior variances and eventually maximum information gain over sampling any N points using the submodularity property of information gain. This allows us to upper bound A_N by a constant proportional to $\sqrt{\gamma_N/N}$. After revealing the connection between the acquisition value and the classification accuracy on the same search space, we can now derive the accuracy outside the trust regions.

4.2.3 Practical Considerations

The following lemma formalizes that the condition $a_g(\bar{\mathbf{x}}_N) \leq A_N$ in Theorem 4.3 occurs infinitely often.

Lemma 4.4. *There are infinitely many integers N s.t. $a_g(\bar{\mathbf{x}}_N) \leq A_N$.*

Lemma 4.4 means this condition is not a restrictive or rare event, which is crucial for the convergence of the global accuracy. In practice, with the magnitude of $\sigma_g(\mathbf{x})$ decreasing with more data points, the global acquisition value $a_g(\bar{\mathbf{x}}_N)$ will also decrease, which means that it will be less than or equal to its running average. Therefore, while the theorem is stated conditionally, the convergence of the global accuracy is primarily dictated by the rate shown in Eq.8.

Other considerations relate to the optimizer's behavior in a continuous setting and how β affects the balance between sample efficiency and accuracy. Theorem 4.3 explicitly incorporates the optimizer's efficiency via a constant κ_g. Previous works often assume perfect AF optimization as a result of operating on a finite set. Incorporating κ_g makes the result more realistic with a formal acknowledgement of the practical difficulty of optimizing the AF directly impacts the achievable classification accuracy. A less efficient optimizer (smaller κ_g) leads to a worse classification accuracy, correctly reflecting real-world performance. The constant β was introduced earlier in Algorithm 1 as the confidence parameter. For the context of Theorem 4.3, demanding a higher confidence level (i.e., a larger β) would require more samples to achieve the same accuracy.

Though Theorem 4.3 is shown with the Straddle heuristic as both global and local AFs, one can use other AFs for TRLSE in practice. We show the experimental results using other AFs in Appendix E.1. In addition, although our focus is on high-dimensional spaces with the assumption of Gaussian noise, it would be interesting to extend this work and our theoretical analysis to non-Gaussian ones. For sub-Gaussian noise assumptions, the results in Theorem 4.3 still hold with minor modifications involving the replacement of the standard Gaussian concentration inequalities with Hoeffding-type bounds (Chowdhury and Gopalan, 2017). Heavy-tailed noise, however, necessitates specialized design choices for GP robustness against outliers; consequently, theoretical guarantees depend on the particular robust GP formulation used.

4.3 A Conjectured Tighter Bound

While the preceding analysis provides a formal guarantee for TRLSE's global accuracy, its convergence rate of $O(\sqrt{\gamma_N/N})$ only accounts for the N points sampled by the global AF. This is a notable limitation, as $\mathcal{GP}_g$ is updated using the entire evaluation budget of $N+T$ points. As discussed in Section 4.1, TRLSE's accuracy should intuitively reflect the total number of samples. Based on this observation, we conjecture that a tighter bound dependent on the total sample size exists. However, a formal proof for such a bound is highly challenging without first making a plausible, yet unproven, assumption about the behavior of the global AF at locally sampled points, which we will now discuss.

Assumption 4.5 (Non-negativity of Global Acquisition Values at Local Points)**.** The global acquisition value at $\mathbf{x}_t$, calculated using $\mathcal{GP}_g$ just before evaluating $\mathbf{x}_t$, is non-negative. Specifically:

$$a_g(\mathbf{x}_t) = \beta\sigma_g(\mathbf{x}_t) - |\mu_g(\mathbf{x}_t) - h| \geq 0$$

This assumption is grounded in the interplay between the local and global models. $\mathcal{GP}^i_{t-1}$ is a specialized and accurate model for its TR $\mathcal{T}^i$, trained on the most relevant local data. $\mathbf{x}_t$ is selected from $\mathcal{T}^i$ because of its high local acquisition value, meaning $\mathcal{GP}^i_{t-1}$ is uncertain about the point's classification. Given that this specialized local model is uncertain, it is improbable that the more general global model $\mathcal{GP}_g$ would be simultaneously highly confident in its classification of this point, which is necessary for $a_g(\mathbf{x}_t)$ to be negative. While such disagreements are possible, they become less frequent with more data.

With this assumption in place, we can explore the implications for a tighter convergence rate of TRLSE through a similar proof strategy. Let A'_{N+T} be the running average over the global acquisition values of all $N+T$ sampled points. A'_{N+T} can be upper bounded

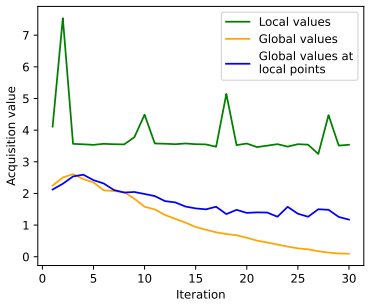


Figure 1: An illustration of Assumption 4.5 on AA33D. The global acquisition values at the local points remain non-negative. Global values constantly decrease with more points sampled as shown in the proof for Theorem 4.3, while local values often spike when new uncertain areas are explored.

by a constant proportional to $\sqrt{\gamma_{N+T}/(N+T)}$ when we also have that $a_g(\mathbf{x}_t) \geq 0, \forall t$. This tighter average can be connected to the global accuracy where two conditions are met: (1) the most recently sampled point is a global point, $\bar{\mathbf{x}}_N$, and (2) its acquisition value is less than or equal to the current running average, $a_g(\bar{\mathbf{x}}_N) \leq A'_{N+T}$. The first condition simply specifies the point in the algorithm's execution that we are analyzing. The second one is guaranteed to be met infinitely often, following a similar logic to that in Lemma 4.4. The global accuracy is then bounded by:

$$\epsilon \leq \frac{\beta}{\kappa_g}\sqrt{\frac{C_1\gamma_{N+T}}{N+T}}. \tag{9}$$

This tighter bound hinges on an assumption that is highly plausible in practice as seen in Figure 1. This bound depends on the total evaluation budget $N+T$ which more accurately reflects the algorithm's operation. It reveals a key synergy in TRLSE's design: local exploration, driven by a separate objective, accelerates the convergence of the global model, providing a more complete theoretical explanation for the sample efficiency observed in our experiments.

5 EXPERIMENTS

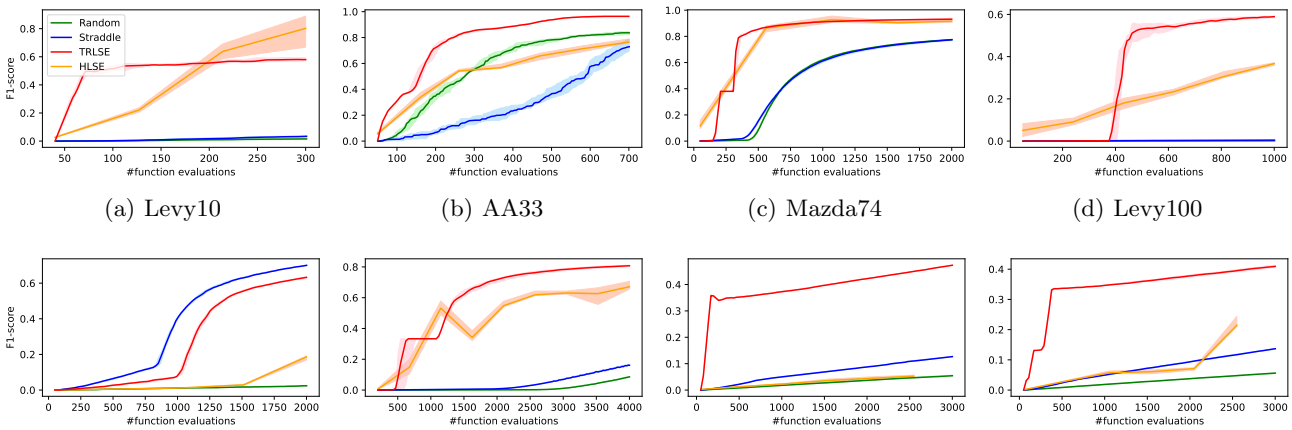
We evaluate TRLSE on 8 synthetic and real-world problems, comparing it against: 1) Random sampling (Random), 2) The Straddle heuristic (Bryan et al., 2005) (STR), and 3) HLSE (Ha et al., 2021). Random sampling and STR use a single GP as their surrogate model, and we use ExpHLSE (Ha et al., 2021) as the implementation of HLSE to match our problem setting of known threshold. These baselines provide a comprehensive evaluation against naive, popular, and state-of-the-art methods. STR is a commonly used method and has a comparable performance to other popular methods including ActiveLSE and TruVar (Zanette et al., 2018), and its inherent adaptability to continuous spaces makes it a highly relevant benchmark for our setting. Since STR is also highly prone to over-exploration in high-dimensional spaces, it is an ideal test for TRLSE's ability to mitigate this issue, especially when it is used as both the global and local AFs in TRLSE. We also consider other AF choices for TRLSE including C2LSE and Thompson sampling and make explicit comparison between these methods and the corresponding TRLSE version in Appendix E.1. In addition, a key focus of the comparison with HLSE is to evaluate which algorithm achieves a better trade-off between classification accuracy and computational efficiency. While HLSE is a state-of-the-art method, its high computational demands and hardware requirements are well-documented, so this comparison is crucial for demonstrating that TRLSE offers a more practical and scalable solution without sacrificing performance.

All experiments use search spaces scaled to $[0,1]^d$. Methods with GPs employ a Matérn kernel with hyperparameter priors adjusted according to Hvarfner et al. (2024). Results using other kernels are shown in Appendix E.3. Descriptions of the benchmarks, along with their corresponding hyperparameters, are presented in Appendix B. To remove the bias of zero-mean prior by GP in classification, we standardize the function values and shift their mean to the threshold. We show results with TRLSE using STR as both global and local AFs. We use $S_1(u) = 2/(1+\exp(8u\text{-}6))$ for $S(\cdot)$ and ablate this choice in Section D.5. The digit suffix in the name of each benchmark shows its dimensionality. We report F1-scores as the measure of classification accuracy. Each method is repeated 10 times, where the initializations are similar among all methods for each repetition. The median and interquartile range among 10 repetitions are plotted. Code to reproduce all experiments is available at `https://github.com/giang-n-ngo/TRLSE`.

5.1 Main Results

5.1.1 Classification performance

Figure 2 shows that TRLSE is at least as good as the baselines on most problems. TRLSE's accuracy also improves consistently throughout the process while HLSE's accuracy may fluctuate, as seen for Ackley200 (and also as reported by Ha et al. (2021)). HLSE also experiences out-of-memory issues and cannot complete the experiments on 1000D problems. STR, however, struggles as a result of almost zero recall on both experiments. In truly high-dimensional spaces, heuristic AFs like STR tend to overly explore due to the depen-



(a) Levy10 (b) AA33 (c) Mazda74 (d) Levy100

(e) Vehicle124 (f) Ackley200 (g) Trid1000 (h) Rosenbrock1000

Figure 2: Results on synthetic functions and real-world benchmark problems

dence on the posterior variance which increases greatly with more dimensions. While STR in this case, with the help of a reduced complexity assumption using adjusted kernel hyperparameters similar to Hvarfner et al. (2024), may not necessarily explore only the border of the search space (which often comes with the largest posterior variance), it becomes similar to random sampling on many experiments because the acquisition values become almost constant as a result of large posterior variance. TRLSE reduces the effect of over-exploration by gradually updating the TRs closer to the threshold boundary and only exploring a new region if a current TR is not promising.

5.1.2 Runtime

Table 1: Average runtimes in minutes for experiments in Figure 2. * indicates out-of-memory experiments.

Problem	Random	STR	HLSE	TRLSE
Levy10	0.1	0.1	368.2	4.2
AA33	1.0	1.7	661.4	11.5
Mazda74	6.1	9.8	599.8	24.4
Levy100	1.9	1.9	1128.1	24.3
Vehicle124	4.9	5.5	673.2	58.7
Ackley200	18.0	18.6	1568.9	992.2
Trid1000	6.3	8.2	2200.9*	433.9
Rosen...1000	6.1	6.9	3867.2*	450.7

Table 1 lists the average runtime for the experiments in Figure 2. HLSE's much higher runtimes and crashes at higher dimensionalities due to memory issues are expected, given its demanding hyperparameter tuning. Note that HLSE ran on three Tesla V100 GPUs while other methods ran on only one. The runtimes of TRLSE are increasing mostly due to a combination of increasing dimensionality (more computation and more test points) and increasing number of TRs. Because the implementation use sequential region updates, TRLSE's runtime on Ackley200 (with 200 TRs) is much higher than those of 1000D problems (which were run with 50 TRs as explained in Appendix B), which means the increased number of TRs outweighs the increased dimensionality in this case.

5.2 Ablation Studies

Due to space constraints, we present the ablation studies in Appendix D, with a summary of the findings below:

- Setting the initial TR volume V_{init} too big or too small can reduce the performance of TRLSE, but this is only noticeable in truly high-dimensional spaces (e.g., 100D). TRLSE achieves a good local-global balance with $V_{init} \approx 0.5^d$ in practice. (Appendix D.1)
- Extremely small R can hurt the accuracy of TRLSE, as it limits the pool for local acquisition optimization. However, large R yields minimal benefits, as the fixed number of points selected at the local level constrains performance improvements. (Appendix D.2)
- Using random new TRs instead of the global AF for exploration significantly degrades performance in truly high-dimensional problems, highlighting the importance of informed re-initialization. (Appendix D.3)
- Using local GPs also contributes to improved performance, especially in higher dimensions, by pro-

viding more accurate local modeling for acquisition optimization and TR updates. (Appendix D.4)

- No volume update reduces performance on both 10D and 100D problems, demonstrating the necessity of adjusting TR volumes based on their proximity to the threshold boundary. In addition, TRLSE is robust to different choices of $S(\cdot)$, as long as it is a monotonically decreasing function that maps large penalties to small adjustment factors and vice versa. Both too large and too small adjustment factors can degrade performance. (Appendix D.5)

6 CONCLUSION

This work addresses high-dimensional level set estimation using multiple local regions to capture threshold boundary. Our algorithm, TRLSE, updates regions based on their boundary centering while using local acquisition to improve threshold boundary modeling. Additionally, a global acquisition function spawns new regions to replace underperforming ones, ensuring global coverage. The algorithm's theoretical classification accuracy is rigorously analyzed, ensuring an accuracy based on the number of points sampled locally and the number of regions initialized. Extensive experiments across multiple synthetic and real-world problems demonstrate TRLSE's superior performance against the baselines with satisfactory performance on up to 1000 dimensions.

References

M. Anjos. The mopta 2008 benchmark. `http://www.miguelanjos.com/jones-benchmark`, 2008. Accessed: 2025-05-05.

Ilija Bogunovic, Jonathan Scarlett, Andreas Krause, and Volkan Cevher. Truncated variance reduction: A unified approach to bayesian optimization and level-set estimation. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

Brent Bryan, Robert C. Nichol, Christopher R Genovese, Jeff Schneider, Christopher J. Miller, and Larry Wasserman. Active learning for identifying function threshold boundaries. In *Advances in Neural Information Processing Systems*, volume 18, 2005.

Sayak Ray Chowdhury and Aditya Gopalan. On kernelized multi-armed bandits. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 844–853. JMLR.org, 2017.

Samuel Daulton, David Eriksson, Maximilian Balandat, and Eytan Bakshy. Multi-objective bayesian optimization over high-dimensional search spaces. In *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, volume 180, pages 507–517, 2022.

Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7(1):1–30, 2006.

David Eriksson and Martin Jankowiak. High-dimensional bayesian optimization with sparse axis-aligned subspaces. In *Uncertainty in Artificial Intelligence*, pages 493–503. PMLR, 2021.

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

Alkis Gotovos, Nathalie Casati, Gregory Hitz, and Andreas Krause. Active learning for level set estimation. In *International Joint Conference on Artificial Intelligence*, 2013.

Huong Ha, Sunil Gupta, Santu Rana, and Svetha Venkatesh. High dimensional level set estimation with bayesian neural network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12095–12103, 2021.

Eric Han, Ishank Arora, and Jonathan Scarlett. High-dimensional bayesian optimization via tree-structured additive models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35: 7630–7638, 2021.

Carl Hvarfner, Erik Orm Hellsten, and Luigi Nardi. Vanilla Bayesian optimization performs great in high dimensions. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pages 20793–20817, 2024.

Johannes Kirschner, Mojmir Mutny, Nicole Hiller, Rasmus Ischebeck, and Andreas Krause. Adaptive and safe Bayesian optimization in high dimensions via one-dimensional subspaces. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 3429–3438, 2019.

Takehisa Kohira, Hiromasa Kemmotsu, Oyama Akira, and Tomoaki Tatsukawa. Proposal of benchmark problem based on real-world car structure design optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, page 183–184, 2018.

Andreas Krause and Carlos Guestrin. Near-optimal nonmyopic value of information in graphical models. In *Proceedings of the Twenty-First Conference on*

Uncertainty in Artificial Intelligence, UAI'05, page 324–331, Arlington, Virginia, USA, 2005. AUAI Press. ISBN 0974903914.

Blake Mason, Lalit Jain, Subhojyoti Mukherjee, Romain Camilleri, Kevin Jamieson, and Robert Nowak. Nearly optimal algorithms for level set estimation. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151, pages 7625–7658, 2022.

Giang Ngo, Dang Nguyen, Dat Phan-Trong, and Sunil Gupta. Active level set estimation for continuous search space with theoretical guarantee. In *Proceedings of the 15th Asian Conference on Machine Learning*, volume 222, pages 943–958, 2024.

Giang Ngo, Dang Nguyen, and Sunil Gupta. Robust transfer learning for active level set estimation with locally adaptive gaussian process prior. In *Proceedings of the 16th Asian Conference on Machine Learning*, volume 260, pages 607–622, 2025.

Quoc Phong Nguyen, Bryan Kian Hsiang Low, and Patrick Jaillet. An information-theoretic framework for unifying active learning problems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 9126–9134, 2021.

ChangYong Oh, Efstratios Gavves, and Max Welling. BOCK : Bayesian optimization with cylindrical kernels. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 3868–3877, 2018.

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.

Santu Rana, Cheng Li, Sunil Gupta, Vu Nguyen, and Svetha Venkatesh. High dimensional Bayesian optimization with elastic Gaussian process. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 2883–2891, 2017.

Paul Rolland, Jonathan Scarlett, Ilija Bogunovic, and Volkan Cevher. High-dimensional bayesian optimization via additive models with overlapping groups. In *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics*, volume 84, pages 298–307, 2018.

Shubhanshu Shekhar and Tara Javidi. Multiscale gaussian process level set estimation. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics*, volume 89, pages 3283–3291, 2019.

Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: no regret and experimental design. In *Proceedings of the 27th International Conference on Machine Learning*, page 1015–1022, 2010.

Sonja Surjanovic and Derek Bingham. Virtual library of simulation experiments: Test functions and datasets. `https://www.sfu.ca/~ssurjano/`, 2013. Accessed: 2025-05-05.

Kevin Tran, Willie Neiswanger, Kirby Broderick, Eric Xing, Jeff Schneider, and Zachary W. Ulissi. Computational catalyst discovery: Active classification through myopic multiscale sampling. *The Journal of Chemical Physics*, 154:124118, 2021.

Ziyu Wang, Frank Hutter, Masrour Zoghi, David Matheson, and Nando De Freitas. Bayesian optimization in a billion dimensions via random embeddings. *J. Artif. Int. Res.*, 55(1):361–387, 2016.

Andrea Zanette, Junzi Zhang, and Mykel J Kochenderfer. Robust super-level set estimation using gaussian processes. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 276–291, 2018.

Yiyang Zhao, Linnan Wang, Kevin Yang, Tianjun Zhang, Tian Guo, and Yuandong Tian. Multi-objective optimization by learning space partition. In *International Conference on Learning Representations*, 2022.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [**Yes**/No/Not Applicable]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [**Yes**/No/Not Applicable]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [**Yes**/No/Not Applicable]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [**Yes**/No/Not Applicable]
   (b) Complete proofs of all theoretical results. [**Yes**/No/Not Applicable]
   (c) Clear explanations of any assumptions. [**Yes**/No/Not Applicable]

3. For all figures and tables that present empirical results, check if you include:

(a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [**Yes**/No/Not Applicable]

(b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [**Yes**/No/Not Applicable]

(c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [**Yes**/No/Not Applicable]

(d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [**Yes**/No/Not Applicable]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:

   (a) Citations of the creator If your work uses existing assets. [**Yes**/No/Not Applicable]

   (b) The license information of the assets, if applicable. [Yes/No/**Not Applicable**]

   (c) New assets either in the supplemental material or as a URL, if applicable. [Yes/No/**Not Applicable**]

   (d) Information about consent from data providers/curators. [Yes/No/**Not Applicable**]

   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Yes/No/**Not Applicable**]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

   (a) The full text of instructions given to participants and screenshots. [Yes/No/**Not Applicable**]

   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Yes/No/**Not Applicable**]

   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Yes/No/**Not Applicable**]

Appendices

A PROOF FOR THEORETICAL RESULTS

A.1 Proof of Lemma 4.1

Proof. When $\bar{l}^i_{t-1} \geq h$, we know that $\mathcal{P}_t(\mathcal{T}^i) = \Phi(|\frac{\bar{l}^i_{t-1}+\bar{u}^i_{t-1}-2h}{2\bar{\sigma}^i_{t-1}}|) \geq \Phi(\beta) \geq \psi$ if the confidence level β is set to be higher than $\Phi^{-1}(\psi)$. It follows that the volume of $\mathcal{T}^i$ will reduce at iteration t since $S(\mathcal{P}_t(\mathcal{T}^i)) < 1$ if $S(u) = 2/(1+\exp(au-b))$ with $b = \psi a$.

Given that the maximum and minimum volumes possible for a TR are V_{max} and $V_{min}/2$, the maximum number of consecutive iterations that a region has its lower-confidence bound $\bar{l}^i_{t-1} \geq h$ is:

$$\zeta = \frac{\log(V_{init}) - 2\log(V_{max})}{\log(2) - \log(1+\exp(-b+a\Phi(\beta)))} \tag{10}$$

The proof when $\bar{u}^i_{t-1} \leq h$ is similar. □

A.2 Proof of Theorem 4.3

A proof for Theorem 4.3 requires several auxiliary results. The first assumption is a necessary condition for the algorithm to continue sampling points, ensuring that there is always at least one point in the search space that remains uncertain with respect to the threshold h.

Assumption A.1. At any iteration t where the algorithm has not yet terminated, there exists at least one point whose classification, given the current surrogate model, remains uncertain with respect to the threshold h. Specifically:

- Global Uncertainty: The set of points within $\bar{\mathcal{U}}_t$ whose confidence intervals from $\mathcal{GP}_g$ contains the threshold h is non-empty. Formally:

$$\{\mathbf{x} \in \bar{\mathcal{U}}_t \mid \mu_g(\mathbf{x}) - \beta\sigma_g(\mathbf{x}) \leq h \leq \mu_g(\mathbf{x}) + \beta\sigma_g(\mathbf{x})\} \neq \emptyset.$$

- Local Uncertainty: The set of points within $\mathcal{U}_t$ whose confidence intervals from their corresponding local Gaussian Process ($\mathcal{GP}^i_t$) contains the threshold h is non-empty. Formally:

$$\{\mathbf{x} \in \mathcal{U}_t \mid \exists i \in \{1,\ldots,R\} \text{ s.t. } \mathbf{x} \in \mathcal{T}^i \text{ and } \mu_i(\mathbf{x}) - \beta\sigma_i(\mathbf{x}) \leq h \leq \mu_i(\mathbf{x}) + \beta\sigma_i(\mathbf{x})\} \neq \emptyset.$$

Assumption A.1 pertains to the continued informativeness that drives the active learning process for acquiring new data points, rather than the final classification assignment. TRLSE actively seeks out points where there is still significant uncertainty about the function's true value relative to the threshold, specifically where the confidence interval $[\mu(\mathbf{x}) - \beta\sigma(\mathbf{x}), \mu(\mathbf{x}) + \beta\sigma(\mathbf{x})]$ still contains the threshold h. Therefore, Assumption A.1 is a practical statement that as long as the algorithm has not fully achieved its goal of confidently classifying the entire search space, there remains at least one point that is still *uncertain* enough to warrant further sampling. In the setting with a discrete search space, this is equivalent to the termination criterion in ActiveLSE (Gotovos et al., 2013) which stops the algorithm when all points are classified with sufficient confidence. In high-dimensional continuous spaces, where fully resolving uncertainty across all points with a limited budget is highly improbable, this assumption remains valid until the problem is effectively solved and the model is sufficiently confident everywhere. This assumption would be violated only in the extreme case where the threshold h lies entirely outside the range of observed function values, causing all acquisition functions to yield negative values and rendering further active sampling trivial for TRLSE's objectives, but we do not consider such extreme scenarios in this theoretical analysis.

Lemma A.2 (Non-negativity of Acquisition Values)**.** *It holds that $a_g(\bar{\mathbf{x}}_n) \geq 0$ and $a_l(\mathbf{x}_t) \geq 0$ for all $t \geq 1, n \geq 1$.*

Proof. According to Assumption A.1, at any non-terminal iteration of the algorithm, $\bar{\mathcal{U}}_t$ is guaranteed to contain at least one point $\mathbf{x}$ s.t. $\mu_g(\mathbf{x}) - \beta\sigma_g(\mathbf{x}) \leq h \leq \mu_g(\mathbf{x}) + \beta\sigma_g(\mathbf{x})$. By definition, this means: $|\mu_g(\mathbf{x}) - h| \leq \beta\sigma_g(\mathbf{x})$, so we can see that for this point $\mathbf{x}$, the acquisition value is non-negative: $a_g(\mathbf{x}) = \beta\sigma_g(\mathbf{x}) - |\mu_g(\mathbf{x}) - h| \geq 0$. The algorithm selects the next point to evaluate, $\bar{\mathbf{x}}_n$, by maximizing its respective acquisition function over its domain (i.e. $\bar{\mathbf{x}}_n = \underset{\mathbf{x}\in\bar{U}_t}{\operatorname{argmax}}\ a_g(\mathbf{x})$), so $a_g(\bar{\mathbf{x}}_n) = \underset{\mathbf{x}\in\bar{U}_t}{\operatorname{argmax}}\ a_g(\mathbf{x}) \geq a_g(\mathbf{x}) \geq 0$. The same reasoning applies to the local acquisition function, $a_l(\mathbf{x}_t)$, thus proving that both global and local acquisition values are non-negative. □

Lemma A.3 (Upper bound the average global acquisition value)**.** *It holds that:*

$$\frac{1}{N}\sum_{n=1}^{N} a_g(\bar{\mathbf{x}}_n) \leq \sqrt{\frac{\beta^2 C_1 \gamma_N}{N}} \tag{11}$$

where γ_N is the maximum information gain over all possible sets of N noisy observations.

Proof. By definition, the global acquisition function is defined as:

$$a_g(\mathbf{x}) = \beta\sigma_g(\mathbf{x}) - |\mu_g(\mathbf{x}) - h|$$

We then have that:

$$\begin{aligned}
\left[\frac{1}{N}\sum_{n=1}^{N} a_g(\bar{\mathbf{x}}_n)\right]^2 &\leq \frac{1}{N}\sum_{n=1}^{N} a_g(\bar{\mathbf{x}}_n)^2 \quad \text{(Cauchy-Schwarz inequality)} \\
&\leq \frac{1}{N}\sum_{n=1}^{N} (\beta\sigma_g(\bar{\mathbf{x}}_n))^2 \quad \text{(given } 0 \leq a_g(\bar{\mathbf{x}}_n) \leq \beta\sigma_{n-1}(\bar{\mathbf{x}}_n)\text{)} \\
&\leq \frac{\beta^2 C_1}{N}\sum_{n=1}^{N} \frac{log(1+\sigma^{-2}\sigma_g(\bar{\mathbf{x}}_n)^2)}{2} \quad \text{(see to Lemma 5.4 in Srinivas et al. (2010))}
\end{aligned} \tag{12}$$

Next, we will try to upper bound the right hand side of the Ineq.12 using γ_N. With $\sigma_g(\mathbf{x})$ being the posterior standard deviation at $\mathbf{x}$ of $\mathcal{GP}_g$ before $\mathbf{x}$ is evaluated, let us denote $\triangle I(\mathbf{x}) = log(1+\sigma^{-2}\sigma_g(\mathbf{x})^2)/2$. From Lemma 5.3 in Srinivas et al. (2010), the information gain from noisy observations $\mathbf{y}_D = [y_t]_{t=1}^{T} \cup [\bar{y}_n]_{n=1}^{N}$, which is used to fit $\mathcal{GP}_g$, is given by:

$$I(\mathbf{y}_D, f) = \sum_{n=1}^{N} \triangle I(\bar{\mathbf{x}}_n) + \sum_{t=1}^{T} \triangle I(\mathbf{x}_t),$$

where $\triangle I(\mathbf{x})$ can be seen as the incremental information gain from observing the noisy function evaluation at $\mathbf{x}$. With $\sum_{n=1}^{N} \triangle I(\bar{\mathbf{x}}_n) \leq I(\mathbf{y}_D, f)$, we can upper bound the right hand side of Ineq.12 using γ_{N+T}, the maximum information gain over all possible sets of $N+T$ noisy observations, but this is larger than our target upper bound γ_N. In order to obtain a tighter bound, we need to relate $\sum_{n=1}^{N} \triangle I(\bar{\mathbf{x}}_n)$ to $I([\bar{y}_n]_{n=1}^{N}, f)$, which is the information gain from just observing $[\bar{y}_n]_{n=1}^{N}$ and can be upper bounded by γ_N.

Because $[y_t]_{t=1}^{T}$ and $[\bar{y}_n]_{n=1}^{N}$ are intermingled in $\mathbf{y}_D$ due to the ordering of observations (as can be seen in the two function evaluation steps for the global and local phases in Algorithm 1), we know that $I([\bar{y}_n]_{n=1}^{N}, f) \neq \sum_{n=1}^{N} \triangle I(\bar{\mathbf{x}}_n)$. However, $I([\bar{y}_n]_{n=1}^{N}, f)$, which is an information gain, satisfies submodularity (Krause and Guestrin, 2005). Since $[\bar{y}_n]_{n=1}^{N}$ are not necessarily the first N observations in $\mathbf{y}_D$, it follows that $I([\bar{y}_n]_{n=1}^{N}, f) \geq \sum_{n=1}^{N} \triangle I(\bar{\mathbf{x}}_n)$. To put it simply, the information gain from observing $[\bar{y}_n]_{n=1}^{N}$ immediately at the beginning of the observation process is at least as large as the sum of the incremental information gains from observing these noisy function values later in the process. Therefore, we have that:

$$\sum_{n=1}^{N} \triangle I(\bar{\mathbf{x}}_n) \leq I([\bar{y}_n]_{n=1}^{N}, f) \leq \gamma_N. \tag{13}$$

Combining Ineq.12 and Ineq.13, the lemma holds as a result. □

Lemma A.4 (Linking Acquisition Value to Classification Accuracy)**.** *Let $a(\mathbf{x}^*) = \max_{\mathbf{x}\in\mathcal{S}} a(\mathbf{x})$ for a set $\mathcal{S}$ with $\hat{\mathcal{H}}_{\mathcal{S}}$ and $\hat{\mathcal{L}}_{\mathcal{S}}$ being the predicted superlevel and sublevel sets using the classification rules as in Algorithm 1, respectively. If $a(\mathbf{x}^*) \leq \epsilon$, the probability that a point $\mathbf{x} \in \mathcal{S}$ satisfies ϵ-accuracy requirements is at least $\Phi(\beta)$.*

Proof. Let $\mathbf{x}^+ \in \hat{\mathcal{H}}_{\mathcal{S}}$ and let $\mathcal{N}(\mu(\mathbf{x}^+), \sigma^2(\mathbf{x}^+))$ be the posterior distribution used for level set classification at $\mathbf{x}^+$. Given the classification rules, we know that $\mu(\mathbf{x}^+) > h$. Having $a(\mathbf{x}^*) \leq \epsilon$ means $a(\mathbf{x}) \leq \epsilon \ \forall \mathbf{x} \in \mathcal{S}$. This means $\beta\sigma(\mathbf{x}^+) - |\mu(\mathbf{x}^+) - h| \leq \epsilon$, or $\mu(\mathbf{x}^+) - \beta\sigma(\mathbf{x}^+) \geq h - \epsilon$ given $\mu(\mathbf{x}^+) > h$ due to the classification rules. Conditioned on $f(\mathbf{x}^+) \sim \mathcal{N}(\mu(\mathbf{x}^+), \sigma^2(\mathbf{x}^+))$, we have that:

$$\Pr(f(\mathbf{x}^+) \leq \mu(\mathbf{x}^+) + \beta\sigma(\mathbf{x}^+)) = \Pr(f(\mathbf{x}^+) \geq \mu(\mathbf{x}^+) - \beta\sigma(\mathbf{x}^+)) = \Phi(\beta).$$

Therefore, $\Pr(h - f(\mathbf{x}^+) \leq \epsilon) \geq \Phi(\beta)$ if $\mathbf{x}^+ \in \hat{\mathcal{H}}_{\mathcal{S}}$. Similarly, we have that $\Pr(f(\mathbf{x}^+) - h \leq \epsilon) \geq \Phi(\beta)$ if $\mathbf{x}^+ \in \hat{\mathcal{L}}_{\mathcal{S}}$. The lemma holds as a result. □

With all auxiliary results established, we can now prove Theorem 4.3.

Let N be an iteration that satisfies the theorem's condition, $a_g(\bar{\mathbf{x}}_N) \leq A_N$. Our goal is to find a uniform accuracy bound *epsilon* for all $\mathbf{x} \in \bar{\mathcal{U}}_N$.

According to Lemma A.5, if we can show that $\max_{\mathbf{x}\in\bar{\mathcal{U}}_N} a_g(\mathbf{x}) \leq \epsilon$, then any point in that domain is classified with *epsilon*-accuracy with probability at least $\Phi(\beta)$. We bound this maximum acquisition value by chaining our preceding results. We know that $\max_{\mathbf{x}\in\bar{\mathcal{U}}_N} a_g(\mathbf{x}) \leq a_g(\bar{\mathbf{x}}_N)/\kappa_g$. Applying the theorem's condition to the numerator, followed by the upper bound on the average from Lemma A.3, we have:

$$\frac{a_g(\bar{\mathbf{x}}_N)}{\kappa_g} \leq \frac{A_N}{\kappa_g} \leq \frac{1}{\kappa_g}\sqrt{\frac{\beta^2 C_1 \gamma_N}{N}}$$

This gives a bound on the maximum acquisition value. Setting our accuracy bound ϵ to this value, $\epsilon = \frac{\beta}{\kappa_g}\sqrt{\frac{C_1\gamma_N}{N}}$, satisfies the condition for Lemma A.4.

A.3 Proof of Lemma 4.4

Proof by Contradiction. Let $B_N = \sqrt{\frac{\beta^2 C_1 \gamma_N}{N}}$. From Lemma A.3, we know that $A_N \leq B_N$, and from Lemma A.2, we know that $A_N \geq 0$ for all N. Assuming common kernels with their corresponding maximum information gain as stated in Theorem 5 in Srinivas et al. (2010), we have that B_N converges to 0 as $N \to \infty$. By the Squeeze Theorem, this implies that $\lim_{N\to\infty} A_N = 0$.

Now, assume for contradiction that the set $I = \{N \in \mathbb{N} \mid a_g(\bar{\mathbf{x}}_N) \leq A_N\}$ is finite. This means there must exist some integer K_0 such that for all $N > K_0$, the opposite condition holds:

$$a_g(\bar{\mathbf{x}}_N) > A_N$$

We can express the running average recursively as $NA_N = (N-1)A_{N-1} + a_g(\bar{\mathbf{x}}_N)$. Substituting our assumption for $N > K_0$:

$$\begin{aligned} NA_N &> (N-1)A_{N-1} + A_N \\ (N-1)A_N &> (N-1)A_{N-1} \\ A_N &> A_{N-1} \end{aligned}$$

This shows that for all $N > K_0$, the sequence $\{A_N\}$ is strictly increasing. Furthermore, since all $a_g(\cdot)$ are non-negative, $A_N > 0$ for all N.

We have thus established that $\{A_N\}_{N>K_0}$ is a strictly increasing sequence of positive numbers. However, an increasing sequence of positive numbers cannot converge to 0. This contradicts our initial finding that $\lim_{N\to\infty} A_N = 0$.

Therefore, our initial assumption must be false, and the set I must be infinite. □

Problem (Source)	Domain	% of superlevel set	V_{init}	V_{max}	R
MC2D (Ngo et al., 2024)	$[0, 9]^2$	12%	10^{-3}	5×10^{-2}	10
Mishra03 (Ngo et al., 2025)	$[-5, 5]^2$	61.5%	10^{-4}	5×10^{-2}	10
Levy10 (Ha et al., 2021)	$[-10, 10]^{10}$	20%	10^{-5}	10^{-1}	40
AA33 (Ha et al., 2021)	See source paper	20%	10^{-4}	10^{-1}	50
Mazda74 (Kohira et al., 2018)	See source paper	24.35%	10^{-20}	10^{-2}	50
Levy100 (Extension of Levy10)	$[-10, 10]^{100}$	20%	10^{-30}	10^{-2}	50
Vehicle124 (Anjos, 2008)	See source paper	3.3%	10^{-35}	10^{-2}	50
Ackley200 (Extension of Ackley10)	$[-5, 10]^{200}$	20%	10^{-60}	10^{-2}	200
Trid1000 (Surjanovic and Bingham, 2013)	$[-10^6, 10^6]^{1000}$	20%	10^{-300}	10^{-2}	50
Rosenbrock1000 (Surjanovic and Bingham, 2013)	$[-5, 10]^{1000}$	20%	10^{-300}	10^{-2}	50

Table 2: TRLSE's data sources and hyperparameters

B DETAILS ON EXPERIMENTS

B.1 Experiment Setup

Table 2 lists the hyperparameters of TRLSE for all synthetic and real-world benchmark problems used in this paper. As discussed earlier in Section D.1, V_{max} is often set to be from 1-10% of the search space's volume, much larger than V_{init}, which is usualy 0.5^d, only to stop a TR from getting infinitely large. The number of regions R maintained simultaneously is also often quite similar between the problems, regardless of the dimensionality of the underlying function. Although Lemma 4.1 only requires the confidence parameter β to be higher than $\Phi^{-1}(0.75)(\approx 0.6745)$, we keep β fixed at 1.96 to demand higher confidence for all experiments. We originally ran the 1000D problems with $R \geq 200$ (at least on par with the number of regions for Ackley200), but prohibitively slow runtimes and memory issues prevent us from setting R higher for these problems, which require much more computational resources with their higher number of dimensions.

Details about Levy10, Ackley10, Alpine10, Protein20, and AA33 are described in Ha et al. (2021). For Mazda74, we first generate 100,000 uniformly random input points, with 74 dimensions for each point, within the allowed input ranges for SUV (as specified by Kohira et al. (2018)). We then run the approximation code (available at https://ladse.eng.isas.jaxa.jp/benchmark/) over these points to obtain the side impact measurement of the SUV, which is used as the output value. We use 0 as the threshold for Mazda74 to align with the constraint for side impact shown by Kohira et al. (2018). Vehicle124 is originally called MOPTA08 by Anjos (2008), which is a minimization problem with hard constraints. We follow the practice of Eriksson and Jankowiak (2021) to obtain an objective value using soft constraints. The specific processing code was taken from the source code in Papenmeier et al. (2022). The threshold for this problem was determined by raising it until random sampling achieves near zero in F1-score.

In practice, initializing TR with $V_{init} = 0.5^d$ can lead to potential floating-point precision issues. In fact, the smallest value for V_{init} shown in Table 2 is 10^{-300} for the 1000D problems, which nearly reaches the limit of double-precision floating-point representation. To test the effect of floating-point precision, we compare using power and multiplication operations with logarithmic operations. We observe up to 10^{-11}% difference in volume calculations and 10^{-13}% difference in TR side length calculations, with higher differences for higher dimensions, being highest in the 1000D case. These differences are negligible in practice, given the lengths of around 0.5. To prevent potential floating-point precision issues, future implementation should consider using logarithmic operations for volume and side length calculations.

B.2 The Effect of Random Sampling on Evaluation Results

Our test set for each problem comprised of 100,000 points randomly sampled across the input space. While random sampling may not yield the most comprehensive test set for evaluation, in high-dimensional spaces without prior knowledge of the function, it is the best approach to test classification accuracy throughout the

Test size	Mazda74	Ackley200	Rosenbrock1000
10000	0.8677 (±0.003499)	0.8379 (±0.004048)	0.4202 (±0.005498)
100000	0.8686 (±0.002103)	0.8379 (±0.001500)	0.4185 (±0.001230)
1000000	0.8684 (±0.000282)	0.8379 (±0.000469)	0.4180 (±0.000461)
10000000	0.8685 (±0.000196)	0.8380 (±0.000193)	0.4179 (±0.000148)
100000000	0.8686 (±0.000050)	0.8380 (±0.000050)	0.4178 (±0.000056)

Table 3: Resulting F1-scores with different test sizes.

Table 4: Results of the statistical tests to compare TRLSE with the baselines (with mean F1-score across all iterations).

Benchmark Problem	Friedman test's p-value	Wilcoxon signed-ranked test's p-value between TRLSE and			Best Algorithm
		Random	Straddle	HLSE	
Levy10	<0.001	0.0117	0.0117	0.0117	TRLSE
AA33	<0.001	0.0117	0.0117	0.0117	TRLSE
Mazda74	<0.001	0.0117	0.0117	0.0391	HLSE
Levy100	<0.001	0.0117	0.0117	0.0117	TRLSE
Vehicle124	<0.001	0.0117	0.0117	0.0117	Straddle
Ackley200	<0.001	0.0117	0.0117	0.0117	TRLSE
Trid1000	<0.001	0.0117	0.0117	0.0117	TRLSE
Rosenbrock1000	<0.001	0.0117	0.0117	0.0117	TRLSE

entire input space. The alternative of using a grid-based test set is simply infeasible in high-dimensional spaces. That said, we have tested the effect of random sampling on the evaluation results of TRLSE at the final iteration by varying the size of the test set from 10,000 to 100,000,000 points (and repeating 10 times for each size) on some benchmarks. From Table 3, although increasing the test set size does lead to more stable F1-scores, it is clear that the results reported in Figure 2 are already quite close to the results with much larger test sets.

C STATISTICAL TESTS

We perform statistical tests to compare the performance of TRLSE with the baselines, previously shown in Figure 2, based on two aggregated metrics derived from each run: (1) the mean F1-score across all iterations, and (2) the final F1-score at the last iteration. For each of these two metrics, following the procedure suggested by Demšar (2006), we first use the Friedman test to assess the significance of the differences in performance across the different methods on each benchmark problem. If the Friedman test indicates significant differences, we conduct post-hoc analysis using pairwise Wilcoxon signed-ranked tests to identify specific differences, with p-values adjusted via the Holm-Bonferroni method. For any pair found to have a statistically significant difference, we then compare their respective means across all repetitions to identify the superior method. A significance level of 0.05 is used for all tests.

Tables 4 and 5 present the results of the statistical tests comparing TRLSE with the baselines, using the mean and final F1-scores, respectively. The Friedman test's p-values indicate significant differences in performance across the methods for all benchmark problems. With the mean F1-score as the aggregated performance metric, the Wilcoxon signed-rank test's p-values reveal that TRLSE significantly outperforms the baselines on 6/8 problems. With the final F1-score as the aggregated performance metric, we observe that TRLSE significantly outperforms the baselines on 5/8 problems, with a tie with HLSE on Mazda74. This confirms the superiority of TRLSE over the baselines in terms of both mean and final F1-scores.

Table 5: Results of the statistical tests to compare TRLSE with the baselines (with final F1-score at last iteration). TRLSE has a higher mean than the HLSE on Mazda74, but the difference is not statistically significant.

Benchmark Problem	Friedman test's p-value	Wilcoxon signed-ranked test's p-value between TRLSE and			Best Algorithm
		Random	Straddle	HLSE	
Levy10	<0.001	0.0117	0.0117	0.0117	HLSE
AA33	<0.001	0.0117	0.0117	0.0117	TRLSE
Mazda74	<0.001	0.0117	0.0117	0.4316	TRLSE/HLSE
Levy100	<0.001	0.0117	0.0117	0.0117	TRLSE
Vehicle124	<0.001	0.0117	0.0117	0.0117	Straddle
Ackley200	<0.001	0.0117	0.0117	0.0117	TRLSE
Trid1000	<0.001	0.0117	0.0117	0.0117	TRLSE
Rosenbrock1000	<0.001	0.0117	0.0117	0.0117	TRLSE

D ABLATION STUDIES

D.1 Trust Region Volume

While V_{max} is often set extremely large at 1-10% of the total search volume simply as a stopping criterion for the TRs, V_{init} directly affects the performance of TRLSE as all TRs are initialized at this volume. Figures 3(a) and 3(b) indicate that setting V_{init} either too big or too small can reduce the performance of the model but is only noticeable in a truly high-dimensional space. Following experimental results, TRLSE generally works best with $V_{init} \approx 0.5^d$ in practice, striking a local-global balance. When the TRs are initialized too big, local AFs cannot be optimized accurately. When initialized too small, they need more iterations to expand, and the pool for local acquisition optimization is also small, hindering the detection of the threshold boundary at the local level.

D.2 Number of Simultaneous Regions R

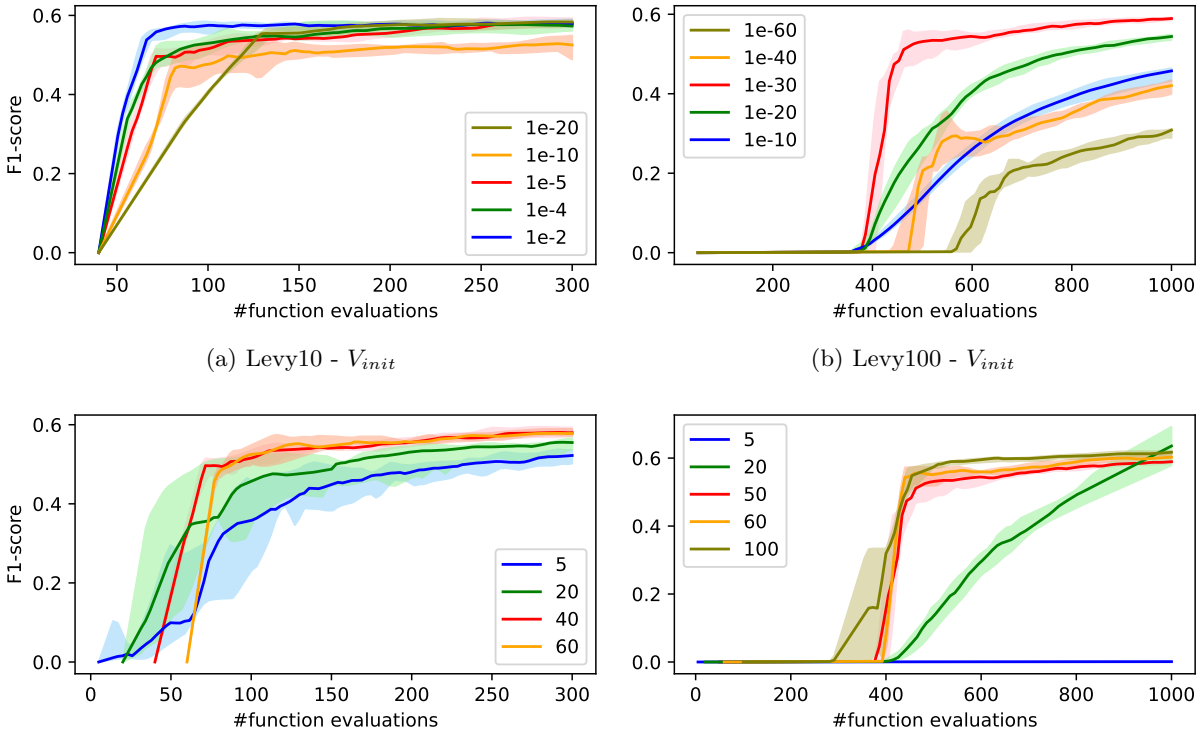
Figures 3(c) and 3(d) show TRLSE has minimal benefits from an increase of R and that an extremely small R can hurt the accuracy of TRLSE. Smaller R means that the pool for local acquisition optimization is also smaller. Thus, the point selected by the local AF can be uninformative if R is small. On the other hand, a fixed number of points selected at the local level means that an ever-increasing local pool can only select that many informative data points within the TRs, so performance is not significantly improved.

D.3 Region Re-initialization

Using the global acquisition function, TRLSE aims to re-initialize underperforming TRs at more informative locations. We ablate this design choice by replacing it with random re-initialization where a new TR is spawned at a random location in the search space (still outside the existing TRs). In Figure 4, no clear improvement is seen on a 10D problem when TRLSE uses the global AF, but the difference widens significantly on a 100D problem. This is expected since the search for informative regions is more challenging on search spaces with higher dimensionality where random re-initialization becomes ineffective.

D.4 Using Local GPs

We verify the necessity of using local GPs for local acquisition optimization and TR updates using a modified version of TRLSE with only the global GP for local computation. Figure 4 again shows that the difference is small on the 10D problem but becomes significant on the 100D problem, which means using local GPs clearly improves TRLSE's performance. This can be explained by a fact demonstrated in Section 3.6 of the TuRBO paper by Eriksson et al. (2019): local GPs, with better fitted hyperparameters, model the local regions better than the global GP. More accurate local modeling improves the accuracy of the local AF, which results in better local sampling and eventually better refinement for the boundary region.



(a) Levy10 - V_{init}

(b) Levy100 - V_{init}

(c) Levy10 - R

(d) Levy100 - R

Figure 3: Ablation studies varying V_{init} and R

D.5 Other Choices for S

We explore other choices for $S(\cdot)$, which converts the penalty $\mathcal{P}_t$ to the adjustment factor for updating the volume of a TR. Firstly, Figures 5(a) and 5(b) show that maintaining constant TR volume (i.e., $S(u) = 1, \forall u$) degrades TRLSE's performance. This demonstrates the necessity of updating region volume according to how it centers at the threshold boundary.

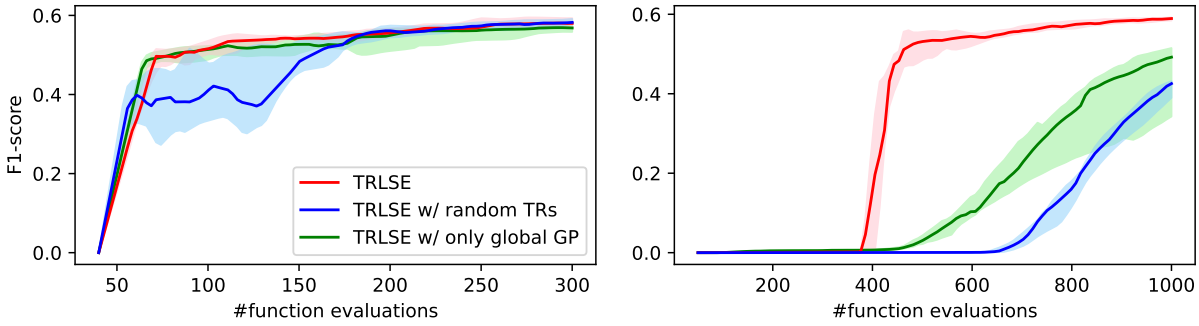
Next, we modify S_1 to still take the sigmoid form but output different ranges for the adjustment factor. Figures 5(a) and 5(b) show that S_1 and S_3 have similar performances, which aligns with the similarity of the two functions compared to S_2 and S_4. Both small (i.e. S_4) and large factors (i.e. S_2) lead to a minor decrease in performance, though they still converge to the same results. Slow TR updates (i.e. small adjustment factors) retain underperforming TRs longer than necessary before getting discarded. With too fast TR updates (i.e. large adjustment factors), premature discards may limit the ability to further explore locally at new promising regions. However, this also indicates that TRLSE is robust to a wide range of adjustment factors.

Finally, we tried some linear functions for $S(\cdot)$. The gap is not significant in Figure 5(c), but in 5(d), both too slow and too fast region updates (i.e. S_5 and S_7) again lead to slightly worse performance than S_6, which has a similar ranges compared to S_1. However, the overall observation is that a linear form for $S(\cdot)$ performs almost as well as the sigmoid form, indicating that $S(u)$ only needs to be smoothly decreasing from around 2 to 0 as u increases from 0.5 to 1.

E ADDITIONAL RESULTS

E.1 Using Different Acquisition Functions

Besides Straddle, we also evaluate TRLSE when it uses other functions for the global and local acquisition functions to test the robustness of the algorithm w.r.t different types of acquisition functions. We employ two



(a) Levy10 (b) Levy100

Figure 4: Ablation studies with: 1) random new TRs, and 2) only the global GP for computations.

acquisition functions, namely C2LSE (Ngo et al., 2024) and Thompson Sampling (TS). Although popularly adopted in BO, TS has never been used in the context of LSE. In this experiment, we define the TS acquisition function for LSE as $a_{TS}(\mathbf{x}) = |\tilde{f}(\mathbf{x}) - h|$ with $\tilde{f}(\mathbf{x}) \sim \mathcal{N}(\mu(\mathbf{x}), \sigma^2(\mathbf{x}))$ where $\mathcal{N}(\mu(\mathbf{x}), \sigma^2(\mathbf{x}))$ is the posterior distribution at $\mathbf{x}$ given by either current global or local GP (depending on where the TS acquisition function is optimized on).

As seen in Figures 6 and 7, TRLSE maintains a robust improvement over the baselines for both acquisition functions on most problems. An additional observation is that TS exhibits a comparable performance with the other two acquisition functions, making it a viable option for the LSE context.

E.2 Low-dimensional Problems and Sampling Behavior

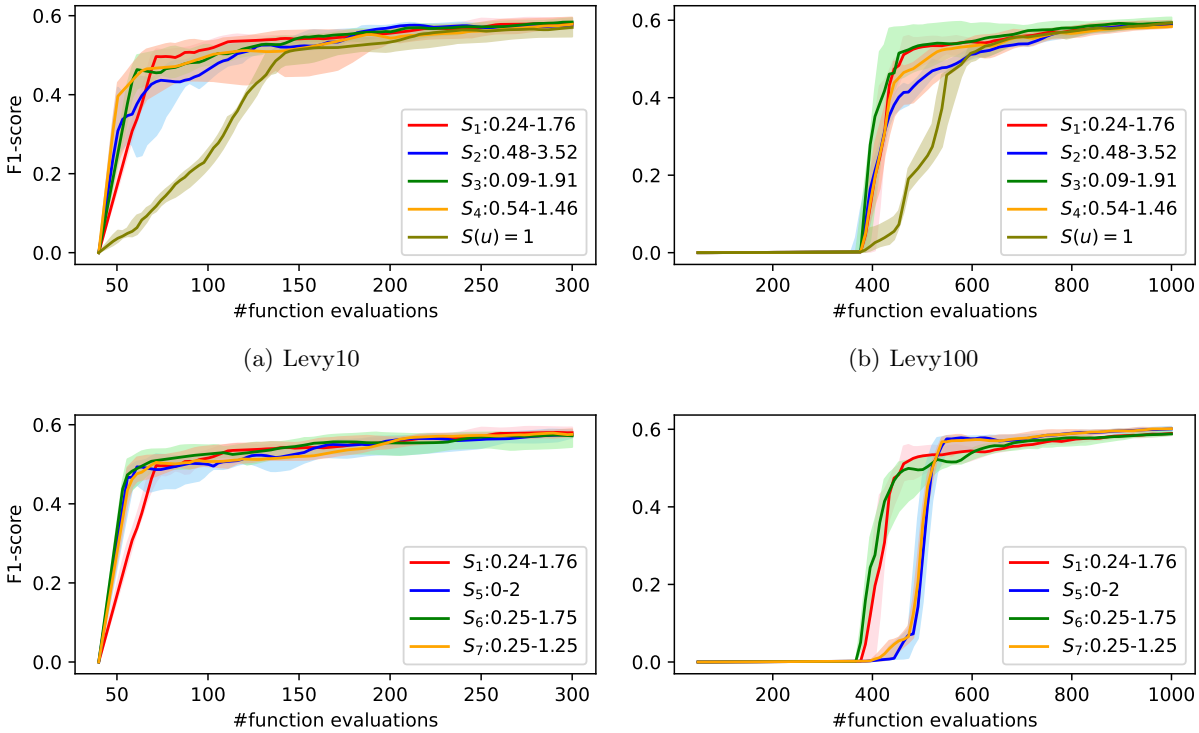
Despite not being specifically tailored for low-dimensional problems, TRLSE still demonstrates a competitive accuracy compared to the baselines on two synthetic functions MC2D (Ngo et al., 2024) and Mishra03 (Ngo et al., 2025), as can be seen in Figure 8. These results suggest that TRLSE is a robust algorithm suitable for most LSE algorithms regardless of the dimensionality of the underlying function. Contrary to the claim that HLSE performs worse than GP-based methods in the low-dimensional settings by Ha et al. (2021), it actually exhibits significantly better performances than Straddle on both MC2D and Mishra03. We believe this claim was from the results on Branin (d=2) and Hartman3 (d=3) functions, which are not good benchmarks for LSE algorithms due to the fact that Straddle only needs a few data points for high F1-scores for both functions.

E.3 Different Kernels

We ablate the kernel choice for GP modeling, which can greatly influence the performance of GP-based methods. We rerun the main experiments in Section 5.1, where everything remains the same, but the kernel is replaced by the radial basis function kernel (RBF) and the rational quadratic kernel (RQ). Note that this is applied to both TRLSE and Straddle, which are GP-based methods, not for HLSE. As seen on both Figures 9 and 10, TRLSE still remains better than the baselines on most experiments for both RBF and RQ, asserting the robustness of TRLSE across different kernels.

E.4 Visualizing the Behavior of TRLSE

From Figure 11, one can observe how TRLSE behaves on a 2D problem. A noticeable observation is that TRs far from the threshold boundary quickly get discarded, and as a result, most TRs in later iterations are near the boundary. Secondly, due to the use of Straddle - a myopic acquisition function, the locally sampled points show explorative behaviors but are still constrained to be inside the TRs, which are relatively close to the boundary. While these points are not exactly on the boundary, they are near it, which helps with modeling the boundary region, not just locating the boundary, and this is more beneficial for level set classification than simply identifying the boundary's location. Another observation is that some TRs get discarded prematurely when they have few surrounding data points. However, this is not essentially bad for the ultimate goal of classifying the whole space.



(a) Levy10

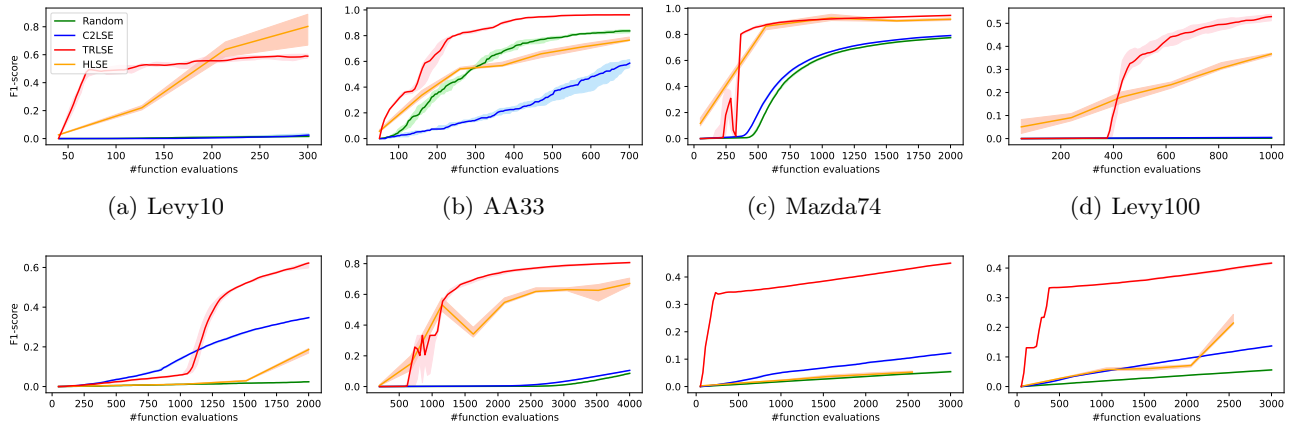
(b) Levy100

(c) Levy10

(d) Levy100

Figure 5: Ablation studies for $S(\cdot)$: (a,b) No volume update + sigmoid variants, and (c,d) linear variants.

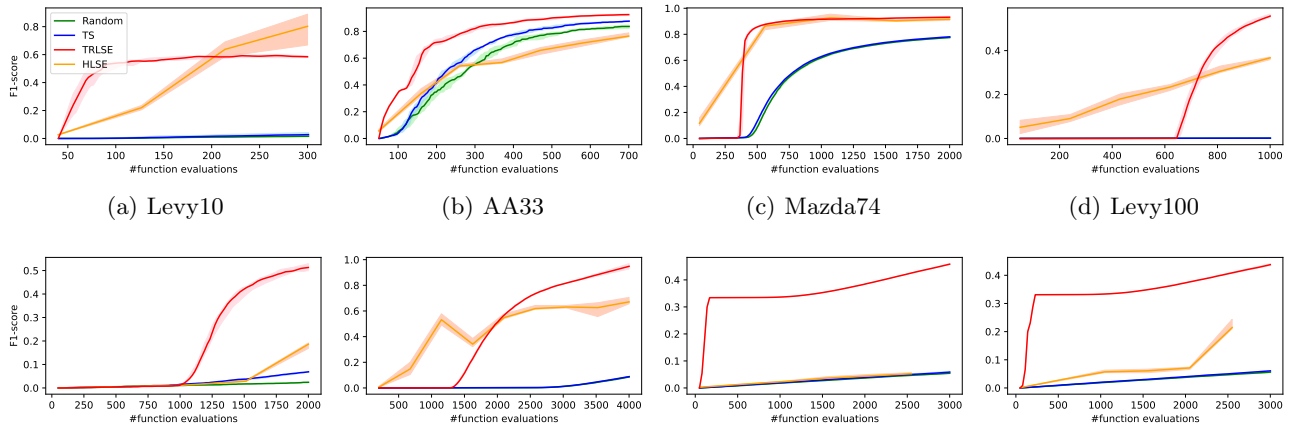
For example, from iteration 8 to 9, though some TRs, which are close to the boundary, get discarded early, the TRs that replace them discover the boundary region in the center (i.e. five new TRs are spawned in the center in iteration 9). This explains the sharp jump in recall (from 0.75 to 0.89), showing evidence that this is beneficial for global coverage.



(a) Levy10 (b) AA33 (c) Mazda74 (d) Levy100

(e) Vehicle124 (f) Ackley200 (g) Trid1000 (h) Rosenbrock1000

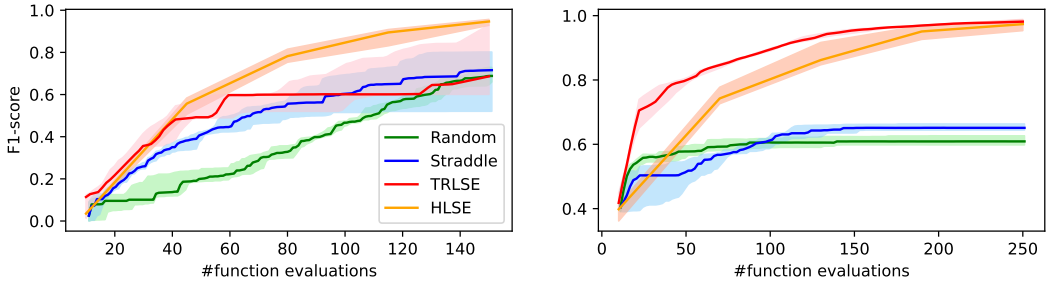
Figure 6: TRLSE using C2LSE as both global and local acquisition functions



(a) Levy10 (b) AA33 (c) Mazda74 (d) Levy100

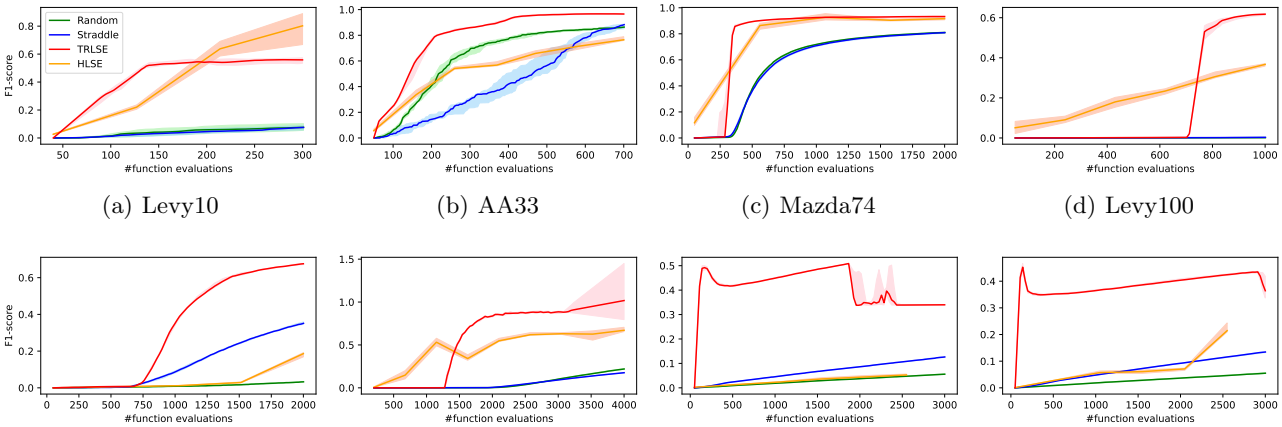
(e) Vehicle124 (f) Ackley200 (g) Trid1000 (h) Rosenbrock1000

Figure 7: TRLSE using TS as both global and local acquisition functions



(a) MC2D (b) Mishra03

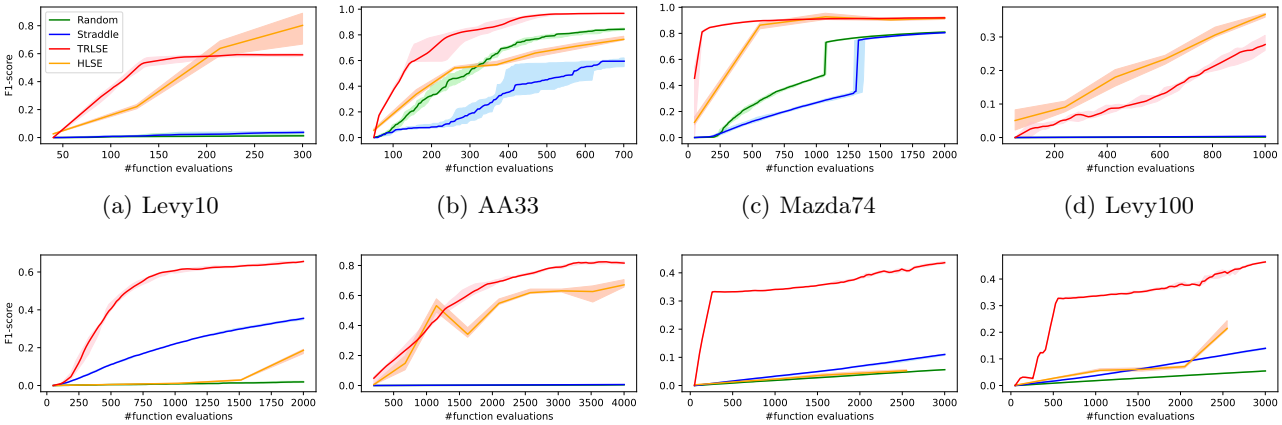
Figure 8: Results on low-dimensional synthetic functions



(a) Levy10 (b) AA33 (c) Mazda74 (d) Levy100

(e) Vehicle124 (f) Ackley200 (g) Trid1000 (h) Rosenbrock1000

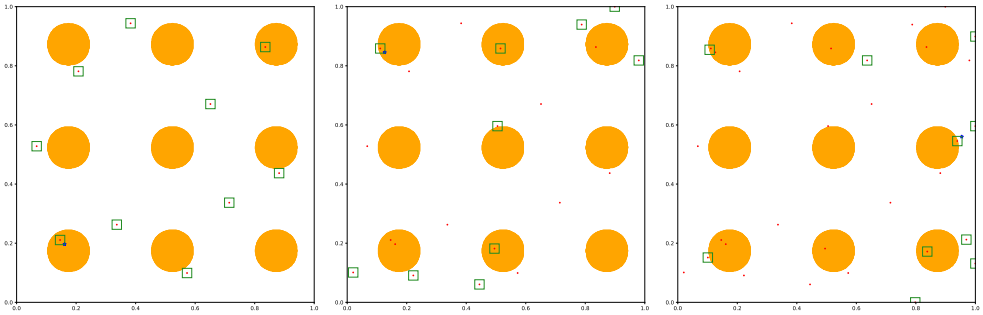
Figure 9: Using RBF kernel for TRLSE and Straddle



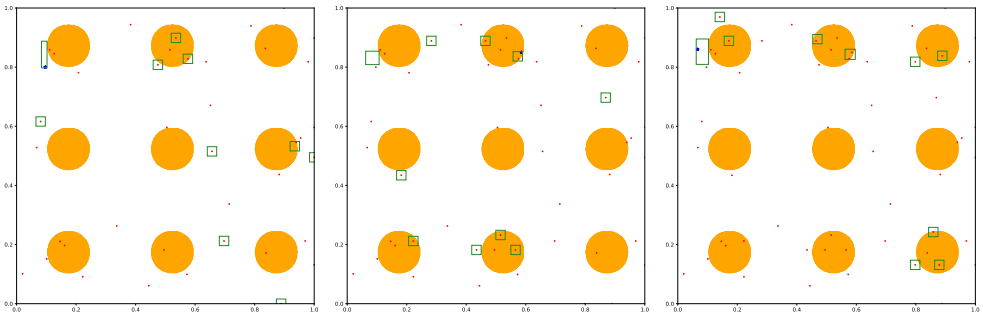
(a) Levy10 (b) AA33 (c) Mazda74 (d) Levy100

(e) Vehicle124 (f) Ackley200 (g) Trid1000 (h) Rosenbrock1000

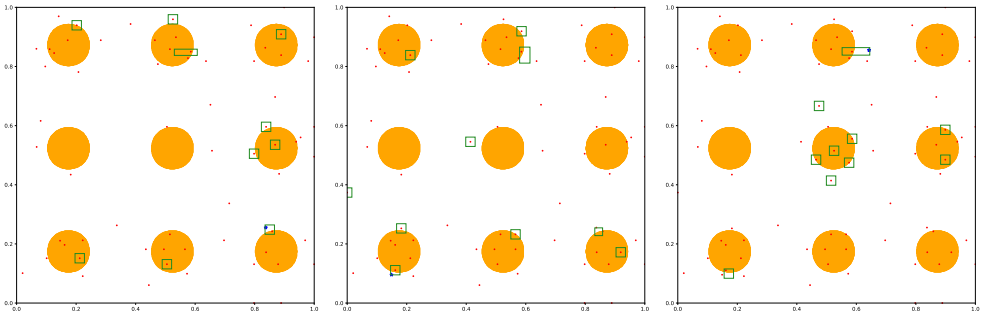
Figure 10: Using RQ kernel for TRLSE and Straddle



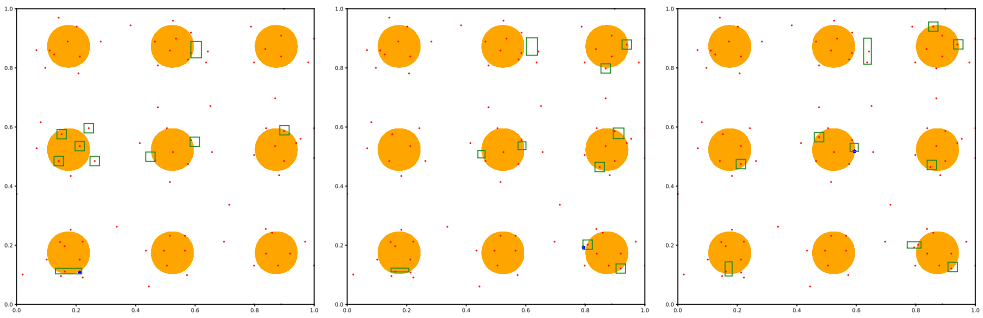
(a) Iteration 1 - recall=0.21 (b) Iteration 2 - recall=0.39 (c) Iteration 3 - recall=0.41



(d) Iteration 4 - recall=0.51 (e) Iteration 5 - recall=0.51 (f) Iteration 6 - recall=0.64



(g) Iteration 7 - recall=0.76 (h) Iteration 8 - recall=0.75 (i) Iteration 9 - recall=0.89



(j) Iteration 10 - recall=1.0 (k) Iteration 11 - recall=1.0 (l) Iteration 12 - recall=1.0

Figure 11: TRLSE's behavior on MC2D with the corresponding recall for each iteration. The superlevel and sublevel sets are in orange and white, respectively. Red dots represent the training data at the current iteration, and the blue star is the point sampled locally. All TRs are represented by the green boxes.