CLUE-BO: ADAPTIVE LANGUAGE PRIORS FOR HIGH-DIMENSIONAL BAYESIAN OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Bayesian optimization (BO) becomes challenging in high dimensions, where limited evaluations make it difficult for Gaussian process surrogates to reliably infer dimension-wise relevance. However, many optimization problems provide semantic descriptions of their decision variables, offering structural information that standard BO does not directly exploit. We introduce CLUE-BO, a language-guided BO framework that uses an LLM to translate task semantics into dimension-specific priors over ARD lengthscales. Rather than treating these semantic priors as fixed or inherently reliable, CLUE-BO continually evaluates them through predictive comparisons between semantic and neutral GPs. Predictive evidence determines when to trigger reflection, whether a revised prior should be activated, and whether optimization should follow semantic or neutral guidance. Across six high-dimensional benchmarks ranging from 30 to 1,003 dimensions, CLUE-BO achieves the best mean final objective on five benchmarks and remains competitive on the sixth. Ablation and trajectory analyses further show that semantic priors complement sparse numerical observations, while evidence-driven control allows unreliable semantic hypotheses to be revised during optimization. These results demonstrate that task semantics can provide useful structural information for high-dimensional BO when observations alone are insufficient.

1 INTRODUCTION

Bayesian optimization (BO) is a sample-efficient method for optimizing expensive black-box functions and has been widely used in hyperparameter tuning, engineering design, and robotic control (Shahriari et al., 2016). BO typically fits a probabilistic surrogate, such as a Gaussian process (GP) (Rasmussen & Williams, 2006), to the evaluations collected so far and uses an acquisition function to decide where to evaluate next. However, BO becomes much more difficult in high-dimensional spaces, where limited evaluation budgets lead to sparse observations and make it harder for the surrogate to reliably capture the structure of the objective (Eriksson & Jankowiak, 2021).

To address this challenge, many high-dimensional BO methods rely on structural assumptions about the objective or the input space, including additive decompositions (Kandasamy et al., 2015), low-dimensional embeddings (Nayebi et al., 2019; Letham et al., 2020), and sparsity over input dimensions (Eriksson & Jankowiak, 2021). These assumptions can reduce the effective complexity of high-dimensional optimization, but their effectiveness depends on how well the assumed structure matches the objective being optimized.

Recent work takes a different route by revisiting standard GPs with automatic relevance determination (ARD) kernels in high dimensions. These studies observe that typical pairwise distances grow on the order of $\sqrt{d}$ as dimensionality increases. Without appropriate calibration, kernel correlations can become very small, leading to vanishing gradients for lengthscale optimization and unreliable GP fitting (Xu et al., 2025). Scaling the lengthscale prior or initialization on the order of $\sqrt{d}$ can substantially improve the behavior of standard ARD GPs in high dimensions (Hvarfner et al., 2024; Xu et al., 2025). However, such calibration mainly addresses the numerical stability of high-dimensional GP modeling. The lengthscale associated with each dimension is still learned primarily from sparse function evaluations, and under limited budgets, the marginal likelihood may not provide enough information to reliably estimate a large number of dimension-wise lengthscales. This motivates a key question: *"Can we provide useful prior information about dimension-wise sen-*

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

sitivity to improve surrogate modeling and thereby improve BO performance in high-dimensional spaces?"

In many practical optimization problems, task descriptions contain semantic clues about how different inputs may affect the objective. Variable meanings, system descriptions, and domain knowledge can provide information about the relative sensitivity of different dimensions. However, standard BO primarily learns from numerical inputs and function evaluations, making such task-level information difficult to incorporate directly. Large language models (LLMs) can interpret these descriptions and use their pretrained knowledge to form hypotheses about the relative sensitivity of different dimensions.

Motivated by this observation, we propose **CLUE-BO**, a language-guided framework that turns semantic clues into dimension-specific priors over ARD lengthscales. Rather than directly proposing query points, the LLM forms hypotheses about the relative sensitivity of different dimensions and expresses them as lengthscale preferences relative to a dimension-aware neutral scale. These preferences are incorporated into a standard full-dimensional ARD GP, allowing task-level semantic information to shape the surrogate while leaving the underlying BO procedure largely unchanged.

Because semantic guidance may not always reflect the true structure of the objective, CLUE-BO continually evaluates its reliability during optimization. We compare a semantic GP against a neutral GP using a sequential e-process (Ramdas et al., 2023; Choe & Ramdas, 2024) built from predictive evidence. When the accumulated evidence provides sufficient support for the neutral model, CLUE-BO triggers LLM reflection to revise the current semantic prior. The revised prior is not activated immediately; instead, it is evaluated against subsequent observations and used to guide acquisition only after receiving sufficient predictive support. This evidence-driven monitoring and validation mechanism allows CLUE-BO to take advantage of useful semantic information while adapting when the current semantic prior becomes unreliable.

Experimental results show that CLUE-BO achieves the highest mean final objective on five of six high-dimensional optimization problems with task semantics, while remaining competitive on the remaining benchmark. Ablation and trajectory analyses further show that semantic guidance provides useful dimension-wise prior information when observations are scarce, while evidence-driven reflection helps revise semantic assumptions that become less consistent with the data. These results suggest that CLUE-BO benefits from the complementarity between semantic guidance and statistical control. The contributions of this work can be summarized as follows:

- We propose **CLUE-BO**, a language-guided framework that leverages semantic clues in task descriptions to construct dimension-specific priors over ARD lengthscales and adapt them as optimization proceeds.
- We develop an evidence-driven reflection mechanism that uses a sequential e-process to continually validate semantic priors against a neutral surrogate, triggering prior revision when needed and requiring revised priors to receive predictive support before guiding acquisition.
- Across six high-dimensional optimization problems ranging from 30 to 1,003 dimensions, CLUE-BO achieves the highest mean final objective on five tasks. Ablation and trajectory analyses further demonstrate the complementary roles of semantic guidance and statistical control.

2 Background

2.1 Bayesian Optimization with ARD Gaussian Processes

We consider the maximization of an expensive black-box function $f : \mathcal{X} \to \mathbb{R}$ over a d-dimensional domain $\mathcal{X} \subseteq \mathbb{R}^d$. At iteration t, given the observations $\mathcal{D}_{t-1} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{t-1}$, Bayesian optimization fits a probabilistic surrogate for f and selects the next query by maximizing an acquisition function,

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

After evaluating $y_t = f(\mathbf{x}_t)$, the new observation is added to the dataset and the surrogate is updated.

We use a Gaussian process (GP) surrogate,

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

with an automatic relevance determination (ARD) Matérn-$5/2$ kernel. Let $\boldsymbol{\ell} = (\ell_1, \ldots, \ell_d)$ denote the dimension-wise lengthscales and define

$$r(\mathbf{x}, \mathbf{x}') = \sqrt{\sum_{j=1}^{d} \frac{(x_j - x_j')^2}{\ell_j^2}}.$$

The kernel is

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \left(1 + \sqrt{5}r + \frac{5}{3}r^2\right) \exp(-\sqrt{5}r).$$

Each ℓ_j controls how rapidly the GP varies along dimension j: a smaller lengthscale allows faster variation, whereas a larger one implies smoother variation. The ARD lengthscales therefore provide a natural way to encode beliefs about dimension-wise sensitivity.

Given observations $\mathbf{y}$, GP hyperparameters are typically estimated by maximizing the marginal likelihood. In high dimensions, however, estimating a large number of ARD lengthscales from limited observations can be difficult. Recent work further suggests that the overall scale of the lengthscales should scale with dimensionality, with dimension-aware priors or initialization on the order of $\sqrt{d}$ substantially improving standard BO; see (Hvarfner et al., 2024; Xu et al., 2025) for detailed motivation and analysis. We use this dimension-aware scale as a neutral reference and introduce semantic information through dimension-specific deviations from it.

2.2 Sequential e-Processes

A p-value measures the compatibility of the observed data with a specified null hypothesis (Wasserstein & Lazar, 2016). In contrast, an e-value is a nonnegative random variable whose expectation is at most one under the null (Vovk & Wang, 2021). An e-process extends this idea to the sequential setting, providing anytime-valid evidence as observations arrive (Ramdas et al., 2023). This makes e-processes particularly useful when the number of observations or stopping time is not fixed in advance.

In sequential model comparison, predictors make forecasts before observing the next outcome, and their predictive performance can be compared as observations arrive (Choe & Ramdas, 2024). CLUE-BO adopts this perspective to compare the semantic and neutral GP surrogates online and accumulate relative predictive evidence between them. The specific e-process used for prior monitoring and validation is introduced in Section 3.3.

3 Methodology

3.1 Overview

CLUE-BO consists of three main components: a semantic prior constructor, an evidence-driven Bayesian optimization loop with active and shadow GPs, and an evidence-driven prior controller, as illustrated in Figure 1. Given textual descriptions of the optimization task, objective, and decision variables, the LLM reasons about the relative sensitivity of different dimensions and produces a lengthscale ratio vector. These preferences are converted into LogNormal priors over ARD lengthscales and incorporated into a standard full-dimensional GP. The same constructor is reused during reflection, where optimization feedback is provided to the LLM to revise the current semantic hypothesis.

Because semantic guidance may not always align with the structure of the objective, CLUE-BO continually evaluates it through sequential predictive comparisons between active and shadow GPs. The resulting evidence drives transitions among three states: semantic monitoring, candidate validation, and neutral monitoring. It triggers reflection when the current semantic prior loses support, validates revised priors before activation, and allows dormant semantic guidance to be reactivated when later observations support it.

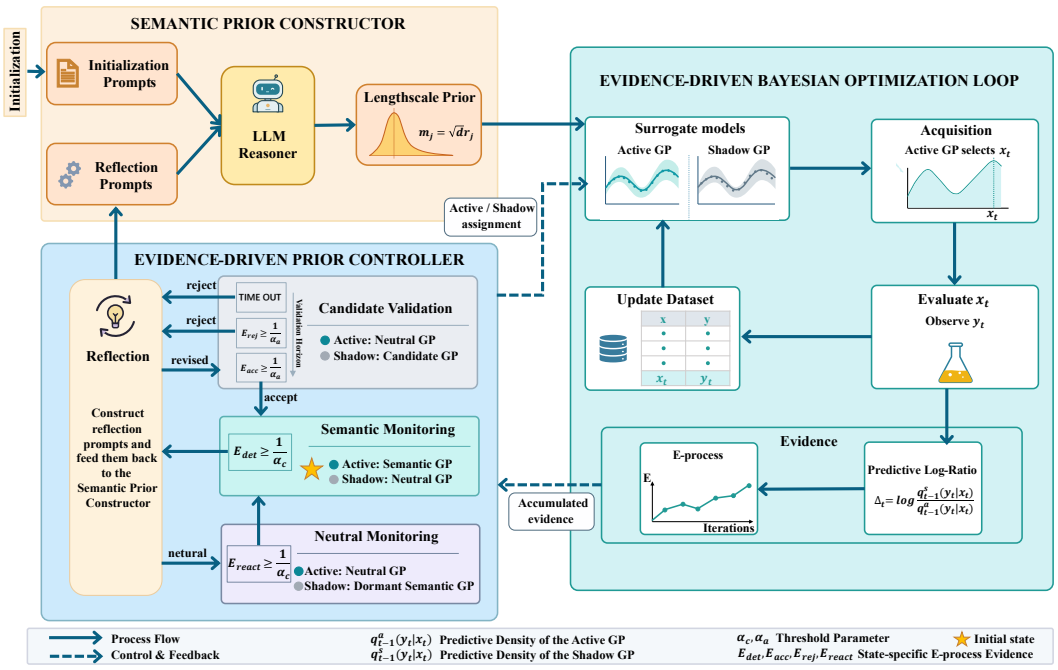

Figure 1: Overview of CLUE-BO. Task semantics are first used to construct dimension-wise priors over ARD lengthscales, which can later be revised by the LLM based on optimization feedback. During BO, an active GP drives acquisition while a shadow GP provides predictive comparisons used to construct a sequential evidence statistic inspired by e-processes. This evidence controls transitions among semantic monitoring, candidate validation, and neutral monitoring, allowing CLUE-BO to revise unreliable semantic priors and activate new ones only after they receive sufficient predictive support.

3.2 Semantic Prior Construction

Estimating a large number of dimension-wise ARD lengthscales from sparse function evaluations is difficult, while many optimization variables come with textual descriptions that provide clues about how they may affect the objective. LLMs provide a natural way to reason over such unstructured task information and form hypotheses about the relative sensitivity of different dimensions. We encode these hypotheses as relative lengthscale preferences, rather than asking the LLM to directly predict absolute GP hyperparameters.

Given a task description $\mathcal{T}$ and variable semantics $\mathcal{S} = \{s_j\}_{j=1}^d$, the LLM produces a positive lengthscale ratio vector

$$\mathbf{r} = (r_1, \ldots, r_d) \in \mathbb{R}_+^d,$$

where r_j specifies the preferred lengthscale of dimension j relative to the neutral scale $L_{\text{base}} = \sqrt{d}$. Here, $r_j = 1$ recovers the neutral scale; a smaller r_j corresponds to a shorter lengthscale and hence greater expected variation along dimension j, whereas a larger r_j indicates smoother variation. The ratio vector therefore captures the relative structure suggested by the task semantics, while retaining the dimension-aware neutral scale as an anchor. We map each semantic ratio to the mode of the corresponding ARD lengthscale prior:

$$m_j = L_{\text{base}} r_j = \sqrt{d}\, r_j.$$

We then use m_j as the mode of a LogNormal prior over the corresponding ARD lengthscale,

$$\ell_j \sim \text{LogNormal}\left(\mu_j, \sigma^2_{\text{prior}}\right), \qquad \mu_j = \log m_j + \sigma^2_{\text{prior}},$$

where σ_{prior} controls the prior spread. This parameterization makes m_j the most favored lengthscale under the prior, directly aligning the semantic preference with MAP fitting. Given observations

$(\mathbf{X}_t, \mathbf{y}_t)$, the GP hyperparameters are fitted using standard MAP estimation:

$$\hat{\boldsymbol{\theta}}_t = \arg\max_{\boldsymbol{\theta}} \left[\log p(\mathbf{y}_t \mid \mathbf{X}_t, \boldsymbol{\theta}) + \sum_{j=1}^{d} \log p(\ell_j) \right].$$

Here, terms unrelated to the semantic lengthscale priors are omitted for clarity. The observed data can subsequently adjust these semantic preferences through MAP fitting.

3.3 Evidence-Driven Reflection and Prior Switching

The semantic prior reflects an LLM hypothesis about dimension-wise sensitivity derived from task descriptions, and this hypothesis may not always match the structure of the objective. CLUE-BO therefore uses optimization data to assess whether the current prior remains supported. We maintain an *active GP*, which controls acquisition, and a *shadow GP*, which is fitted on the same observations but used only for predictive comparison. Based on their accumulated predictive evidence, the controller transitions among semantic monitoring, candidate validation, and neutral monitoring, as illustrated in Figure 1.

Sequential predictive evidence. Because a single-step predictive comparison can be noisy, CLUE-BO accumulates log predictive ratios over time. At BO iteration t, both the active and shadow GPs are fitted on $\mathcal{D}_{t-1}$, and the active GP selects the next query $\mathbf{x}_t$. Before observing y_t, the two models provide predictive densities $q^a_{t-1}(y \mid \mathbf{x}_t)$ and $q^s_{t-1}(y \mid \mathbf{x}_t)$. Once y_t is observed, we compute

$$\Delta_t = \log \frac{q^s_{t-1}(y_t \mid \mathbf{x}_t)}{q^a_{t-1}(y_t \mid \mathbf{x}_t)}.$$

Thus, $\Delta_t > 0$ means that the realized observation has higher predictive density under the shadow model at the queried point, whereas $\Delta_t < 0$ indicates the opposite. Since both predictive distributions are fixed before y_t is revealed, this comparison is prequential (Dawid, 1984). Let $\mathcal{G}_t$ denote the information available before observing y_t, including the selected $\mathbf{x}_t$ and the two fitted predictors, and let g_t denote the conditional density of the next observation. Whenever the divergences are finite, the logarithmic scoring rule (Gneiting & Raftery, 2007) gives

$$\mathbb{E}_{g_t}[\Delta_t \mid \mathcal{G}_t] = D_{\mathrm{KL}}(g_t \| q^a_{t-1}) - D_{\mathrm{KL}}(g_t \| q^s_{t-1}),$$

so positive expected increments indicate a smaller KL divergence for the shadow predictive distribution at the queried point.

Because a predictive mismatch may emerge after an episode has already begun, we maintain evidence accumulators for multiple possible onset times, following multi-start constructions from sequential change detection (Ramdas et al., 2023; Shin et al., 2024). For a possible onset τ, we accumulate the relative predictive evidence from τ to the current local step n, and aggregate over possible onsets within an episode of horizon L:

$$R_{\tau,n} = \exp\left(\sum_{i=\tau}^{n} \Delta_i\right), \qquad E^{(L)}_n = \frac{L - n + \sum_{\tau=1}^{n} R_{\tau,n}}{L}, \qquad 0 \le n \le L.$$

The term $L - n$ accounts for onset components that have not yet been reached, so that $E^{(L)}_0 = 1$. Under the corresponding model-based null, this normalized likelihood-ratio mixture admits an e-process interpretation (Ramdas et al., 2023). In CLUE-BO, we use $E^{(L)}_n$ as a sequential statistic for relative predictive evidence. For semantic and neutral monitoring, L is set to the remaining BO evaluation budget B_{rem} at the start of the episode, whereas candidate validation uses $L = \min(H, B_{\mathrm{rem}})$, where H denotes the prescribed candidate-validation horizon.

Since evidence monitoring may restart multiple times during optimization, we use a decreasing alpha-spending schedule across restarts (Tian & Ramdas, 2021):

$$\alpha_c = \frac{\alpha}{c(c+1)}, \qquad \alpha_a = \frac{\alpha}{a(a+1)}.$$

Here, c indexes newly started semantic or neutral monitoring episodes, whereas a indexes reflection-generated proposals. Semantic and neutral monitoring use the threshold $1/\alpha_c$, while both directions of candidate validation use $1/\alpha_a$.

Semantic monitoring and reflection. While a semantic prior guides acquisition, its dimension-wise allocation remains a hypothesis derived from task semantics. We track its predictive support against a neutral reference as optimization proceeds. During semantic monitoring, the GP fitted under the current semantic prior serves as the active model, while the GP fitted under the neutral prior, with $r_j = 1$ for all dimensions, serves as the shadow model. Denoting their predictive densities by q_{t-1}^{sem} and q_{t-1}^{neu}, respectively, we monitor

$$\Delta_t^{\mathrm{det}} = \log \frac{q_{t-1}^{\mathrm{neu}}(y_t \mid \mathbf{x}_t)}{q_{t-1}^{\mathrm{sem}}(y_t \mid \mathbf{x}_t)}.$$

As long as $E_n^{\mathrm{det}} < 1/\alpha_c$, CLUE-BO remains in semantic monitoring and the current semantic prior continues to guide acquisition. Once the threshold is crossed, the accumulated predictive evidence favors the neutral reference over the current semantic allocation, prompting CLUE-BO to reflect on the current prior.

The reflection prompt combines the semantic basis of the current prior with optimization feedback, including: **(i)** the task semantics, **(ii)** the original and current dimension-wise prior allocations, **(iii)** the accumulated predictive evidence, **(iv)** the current optimization trajectory and progress, and **(v)** the fitted ARD lengthscales together with their deviations from the corresponding prior modes.

The LLM then either proposes a revised non-neutral allocation or returns to the neutral configuration. A revised allocation is converted into a new semantic ARD prior using the procedure described in Section 3.2, and the controller then moves to candidate validation rather than activating the prior immediately. If the LLM instead returns the neutral configuration, the controller moves to neutral monitoring.

Candidate validation of revised priors. Reflection produces a revised semantic hypothesis from task semantics and accumulated optimization feedback. Because this prior is proposed using past observations, we do not activate it immediately. Instead, it is evaluated on subsequent observations in *candidate validation*. During this stage, the neutral-prior GP serves as the active model and continues to select queries, while the GP fitted under the revised semantic prior serves as the shadow model, referred to as the *candidate GP*. Their predictive densities are denoted by q_{t-1}^{neu} and q_{t-1}^{cand}, respectively.

For each new observation, we compute predictive evidence in both directions:

$$\Delta_t^{\mathrm{acc}} = \log \frac{q_{t-1}^{\mathrm{cand}}(y_t \mid \mathbf{x}_t)}{q_{t-1}^{\mathrm{neu}}(y_t \mid \mathbf{x}_t)}, \qquad \Delta_t^{\mathrm{rej}} = \log \frac{q_{t-1}^{\mathrm{neu}}(y_t \mid \mathbf{x}_t)}{q_{t-1}^{\mathrm{cand}}(y_t \mid \mathbf{x}_t)}.$$

The corresponding statistics E_n^{acc} and E_n^{rej} are both compared with $1/\alpha_a$. If E_n^{acc} crosses the threshold first, the candidate prior becomes active and CLUE-BO returns to semantic monitoring. If E_n^{rej} crosses the threshold first, or if neither statistic crosses within the validation horizon, CLUE-BO reflects again using the evidence collected during validation. A new non-neutral prior is evaluated in another candidate-validation episode, while a neutral outcome moves the controller to neutral monitoring.

Neutral monitoring and reactivation. We argue that falling back to the neutral prior should not permanently disable semantic guidance, since its relative predictive support may change as new observations are collected. Inspired by the principle of reversible fallback in runtime assurance (Peng et al., 2023), CLUE-BO treats the neutral configuration as a reversible fallback. The controller enters *neutral monitoring*, where the neutral-prior GP serves as the active model and continues to select queries, while the most recent non-neutral prior is retained as the shadow model, referred to as the *dormant semantic GP*. We retain the most recent prior because it represents the latest available non-neutral semantic hypothesis informed by optimization feedback.

To assess whether this dormant prior regains predictive support, we continue to compare it against the active neutral GP. Denoting its predictive density by q_{t-1}^{dorm}, we compute

$$\Delta_t^{\mathrm{react}} = \log \frac{q_{t-1}^{\mathrm{dorm}}(y_t \mid \mathbf{x}_t)}{q_{t-1}^{\mathrm{neu}}(y_t \mid \mathbf{x}_t)}.$$

The resulting E_n^{react} is compared with $1/\alpha_c$. While it remains below the threshold, the controller stays in neutral monitoring: the neutral GP continues to guide acquisition, while the dormant semantic GP is used only for predictive comparison. Once the threshold is crossed, the dormant semantic

prior has regained sufficient predictive support and is directly reactivated as the active model, moving the controller back to semantic monitoring. This reversible fallback preserves a pathway for semantic guidance to re-enter the optimization process when it regains sufficient predictive support.

Overall, CLUE-BO treats LLM-derived semantic guidance as a revisable hypothesis rather than a fixed optimization rule, with predictive evidence controlling when it is retained, revised, or reactivated.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETTINGS

We evaluate CLUE-BO on six real-world optimization benchmarks that span robotic control, neural architecture search, trajectory planning, and hyperparameter optimization. The tasks range from 30 to 1,003 dimensions and provide semantically meaningful decision variables, allowing us to evaluate whether semantic information can provide useful structural priors for high-dimensional BO. For consistency, all tasks are formulated as maximization problems.

Benchmarks. We consider six real-world optimization problems. Except for Hopper-v4, our benchmark implementations are based on those of Xu et al. (2025). **Hopper-v4 (33)** is a MuJoCo continuous-control task (Todorov et al., 2012), for which we adopt the 33-dimensional linear-policy parameterization of Wang et al. (2020) in the Hopper-v4 environment. **NAS201 (30)** is a neural architecture search problem based on NAS-Bench-201 (Dong & Yang, 2020). **Rover (60)** is a trajectory planning problem (Wang et al., 2018), where the trajectory is determined by 30 waypoints in a two-dimensional environment. **DNA (180)** is a hyperparameter optimization problem from LassoBench (Šehić et al., 2022), involving 180 regularization parameters for the weighted Lasso regression. **SVM (388)** is a high-dimensional hyperparameter optimization problem for support vector regression (Eriksson & Jankowiak, 2021), involving 3 regularization parameters and 385 kernel lengthscale parameters. **HumanoidStandup-v4 (1,003)** is a high-dimensional MuJoCo control problem (Todorov et al., 2012); we retain the 1,003-dimensional parameterization used by Xu et al. (2025) in the HumanoidStandup-v4 environment. Detailed task definitions, parameterizations, and variable semantics are provided in Appendix B.1.

Baselines. We compare CLUE-BO with representative high-dimensional black-box optimization methods spanning embedding-based, trust-region, and full-dimensional GP approaches. *HeSBO* (Nayebi et al., 2019) performs BO in a low-dimensional random embedding; we use embedding dimensions of 10 and 20, denoted by HeSBO-10 and HeSBO-20. We also include *Bounce* (Papenmeier et al., 2023), which searches through nested embeddings of increasing dimensionality. For trust-region optimization, we evaluate two variants of *REI* (Namura & Takemori, 2025): TuRBO-1-LogEI-qREI and TuRBO-1-LogEI-qREI(restart). We also compare with three full-dimensional GP baselines. *M5-UCB* uses a standard ARD Matérn-$5/2$ GP with UCB and BoTorch's default lengthscale initialization. *SBO* (Xu et al., 2025) uses the same Matérn-$5/2$ kernel and UCB acquisition function but adopts dimension-aware initialization. *VBO* (Hvarfner et al., 2024) instead introduces dimension-scaled GP lengthscale priors. Finally, we include *CMA-ES* (Hansen & Ostermeier, 2001) as a non-Bayesian derivative-free optimizer. Unless otherwise specified, we use the original implementations and their recommended settings for the competing methods.

Implementation Details. We use Deep-v4-Pro for semantic prior construction and reflection. For CLUE-BO, we use an ARD Matérn-$5/2$ kernel and adopt UCB with $\beta = 1.5$ as the acquisition function. We set the base evidence level to $\alpha = 0.01$ and use a candidate-validation horizon of $H = 40$. Each run is initialized using Latin hypercube sampling (LHS), with 20 initial observations for all tasks except HumanoidStandup-v4, for which we use 50. We run all methods for 300 optimization iterations across 10 random seeds. Additional implementation details and hyperparameter settings are provided in Appendix B.2.

4.2 MAIN RESULTS

Figure 2 reports the distributions of the final best-so-far objective values on the six benchmarks. CLUE-BO achieves the highest mean final objective on five of the six tasks, with particularly clear gains on DNA, Rover, and HumanoidStandup-v4. It also achieves the highest mean final objec-

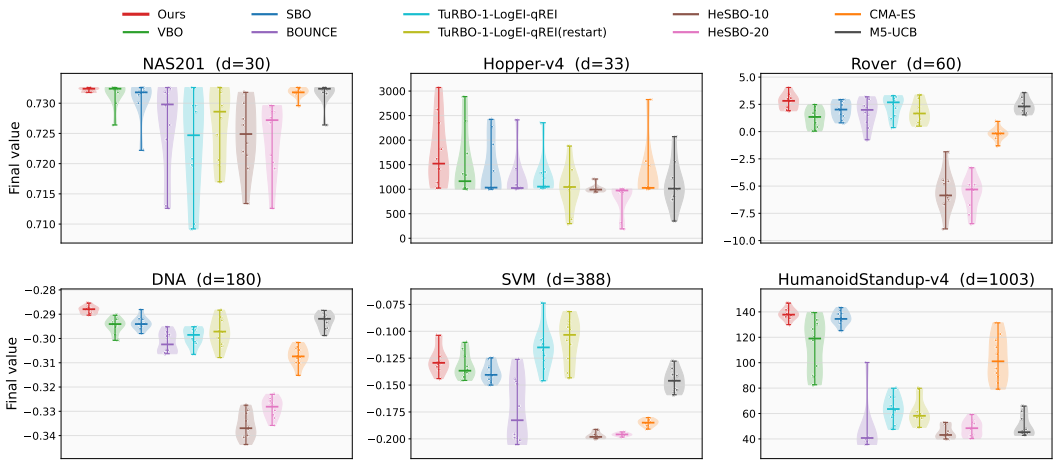

Figure 2: Final optimization performance on six high-dimensional benchmarks. Each violin shows the distribution of the final best-so-far objective value across runs. Higher values are better.

tive on Hopper-v4 and NAS201, although the margins are smaller. In NAS201, several methods reach similar final objective values, while CLUE-BO exhibits a relatively concentrated terminal distribution. SVM is the only exception: the two REI variants achieve higher final objective values, while CLUE-BO still outperforms most of the remaining baselines. In particular, explicit semantic information is available for only a small subset of the SVM dimensions, leaving less task-specific information for constructing dimension-wise semantic priors. This is consistent with the smaller gains of CLUE-BO on this benchmark. Overall, the violin plots suggest that the gains of CLUE-BO are observed across runs rather than being driven by a few favorable outcomes.

These results are consistent with the motivation behind CLUE-BO. In high-dimensional BO, limited evaluation budgets provide only sparse observations for estimating many dimension-wise ARD lengthscales, motivating the use of complementary prior information on relative sensitivity. The comparison with SBO and VBO is particularly informative: both are strong full-dimensional GP baselines that improve high-dimensional lengthscale calibration, whereas CLUE-BO additionally incorporates task-specific, dimension-wise semantic information into the lengthscale priors. The observed gains across several benchmarks provide evidence that such semantic clues can offer a useful complementary source of structure when numerical observations are sparse. Complete best-so-far optimization trajectories are provided in Appendix C.1. They further show that CLUE-BO does not necessarily outperform the baselines from the earliest iterations, but often becomes increasingly competitive as optimization proceeds, which is consistent with its design of continually reassessing semantic guidance as new observations accumulate.

4.3 Ablation Studies

We consider nine ablations in two groups. The first group targets the core semantic adaptation loop. *w/o LLM* removes language guidance entirely and uses the neutral prior throughout optimization; *w/o Reflection* uses the LLM only to construct the initial semantic prior and disables subsequent revisions; *w/o Validation* retains reflection but activates each revised semantic prior immediately, bypassing prospective validation; and *w/o Neutral Recovery* disables reactivation once the controller falls back to the neutral prior. The second group probes the reflection context while the rest of the controller unchanged: *w/o Task Semantics*, *w/o Prior Allocations*, *w/o Trajectory Progress*, *w/o Predictive Evidence*, and *w/o ARD Displacement* each remove the corresponding information source. Figure 3 reports the resulting best-so-far optimization trajectories.

Effect of the core adaptation mechanisms. The full CLUE-BO achieves a higher terminal mean than all four mechanism ablations on all six tasks, although their relative performance varies across tasks and optimization stages. Notably, *w/o LLM* achieves a higher terminal mean than *w/o Reflection* on Hopper-v4 and HumanoidStandup-v4, suggesting that an inaccurate or stale semantic prior may be less effective than the neutral reference when it is never revised. Reflection mitigates this

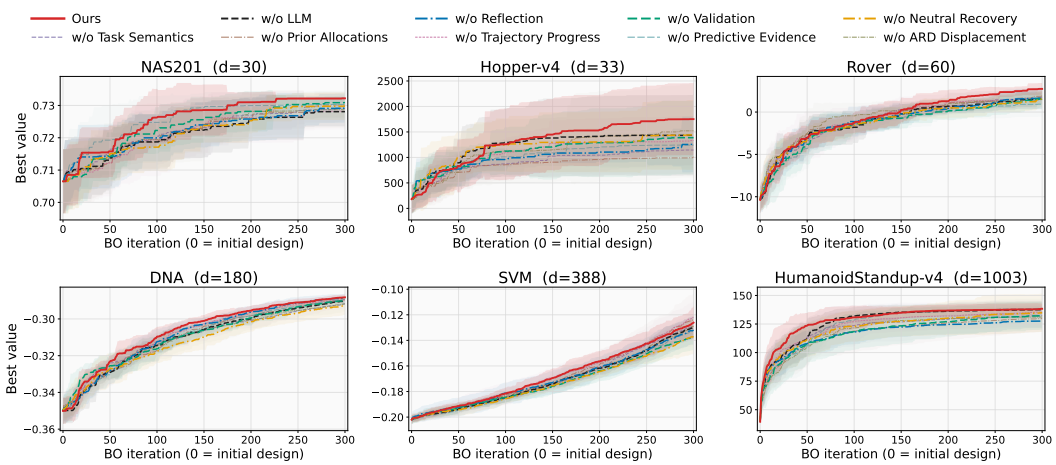

Figure 3: Best-so-far optimization trajectories of CLUE-BO and its ablations on six high-dimensional benchmarks. Each curve shows the mean best-so-far objective value across runs as optimization proceeds, and the shaded region denotes one standard deviation. Higher values are better.

failure mode, yet *w/o Validation* still underperforms the full method, particularly on Hopper-v4 and HumanoidStandup-v4, suggesting that reflected proposals are not consistently reliable enough for immediate activation. On Rover, the variants remain close early in optimization, while CLUE-BO gradually separates later; disabling neutral recovery also leads to a lower late-stage plateau. The differences are smaller on NAS201, DNA, and SVM, suggesting that the value of adaptation depends partly on the quality of the initial semantic hypothesis. Overall, these results indicate that semantic guidance is useful but not uniformly reliable. CLUE-BO benefits from coordinating mechanisms for retaining, revising, validating, and reactivating semantic guidance, rather than relying on any single mechanism alone.

Effect of the reflection context. Compared with the five prompt ablations, CLUE-BO with the full reflection context achieves the highest terminal mean on five of the six tasks, supporting the general benefit of combining semantic information with feedback from the ongoing optimization process. However, the contribution of individual information sources varies across tasks. SVM provides the clearest exception: removing either trajectory progress or prior-allocation history yields a slightly higher terminal mean than the full context. This indicates that providing more information during reflection does not guarantee better decisions on every task; depending on the optimization state, some signals may be redundant or less informative than others. Identifying which feedback should be emphasized across different tasks and at different optimization stages therefore represents a promising direction for improving the reflection mechanism.

5 Conclusion

We introduced **CLUE-BO**, a language-guided framework for high-dimensional Bayesian optimization that uses task semantics to construct dimension-specific priors over ARD lengthscales. CLUE-BO complements sparse function evaluations with semantic clues about relative variable sensitivity, while treating this guidance as a revisable hypothesis rather than fixed prior knowledge. An evidence-driven controller continuously evaluates the resulting priors, triggers reflection when they lose predictive support, and validates revised priors before activation. Across six high-dimensional optimization tasks, CLUE-BO achieves the best mean final objective on five tasks while remaining competitive on the sixth, with ablations confirming the complementary roles of semantic guidance and evidence-driven adaptation. These results demonstrate that task semantics can provide useful structural information when sparse observations alone are insufficient to reliably infer dimension-wise relevance. Future work could extend this principle beyond ARD lengthscale priors to richer surrogate structures and broader black-box optimization settings.

AI USE STATEMENT

Large Language Models (LLMs) were used to assist with the writing and polishing of the manuscript, as well as with code implementation and debugging. Specifically, LLMs were used to refine language, improve readability and clarity, rephrase sentences, check grammar, and improve the overall flow of the text. They were also used to provide assistance with code implementation and debugging.

The LLMs were not used for research ideation, methodology design, experimental design, interpretation of results, or formulation of scientific conclusions. All research ideas, methodological decisions, experimental settings, analyses, and scientific conclusions were independently determined by the authors. All AI-assisted text was reviewed by the authors, and AI-assisted code was manually reviewed and tested for correctness.

The authors take full responsibility for the final content of this work, including any text or code produced or polished with the assistance of LLMs, as well as the reported experimental results and conclusions. All AI-assisted content was reviewed to ensure consistency with the authors' intended meaning and compliance with relevant ethical and academic standards.

REFERENCES

Virginia Aglietti, Ira Ktena, Jessica Schrouff, Eleni Sgouritsa, Francisco Ruiz, Alan Malek, Alexis Bellot, and Silvia Chiappa. FunBO: Discovering acquisition functions for bayesian optimization with FunSearch. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 639–661. PMLR, 2025.

Yo Joong Choe and Aaditya Ramdas. Comparing sequential forecasters. *Operations Research*, 72: 1368–1387, 2024.

A. P. Dawid. Present position and potential developments: Some personal views: Statistical theory: The prequential approach. *Journal of the Royal Statistical Society: Series A (General)*, 147: 278–290, 1984.

Xuanyi Dong and Yi Yang. NAS-Bench-201: Extending the scope of reproducible neural architecture search. In *International Conference on Learning Representations*, 2020.

David Eriksson and Martin Jankowiak. High-dimensional bayesian optimization with sparse axis-aligned subspaces. In *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, pp. 493–503, 2021.

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

Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102:359–378, 2007.

Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9:159–195, 2001.

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

Kirthevasan Kandasamy, Jeff Schneider, and Barnabas Poczos. High dimensional bayesian optimisation and bandits via additive models. In *Proceedings of the 32nd International Conference on Machine Learning*, pp. 295–304. PMLR, 2015.

Benjamin Letham, Roberto Calandra, Akshara Rai, and Eytan Bakshy. Re-examining linear embeddings for high-dimensional bayesian optimization. In *Advances in Neural Information Processing Systems*, 2020.

Tennison Liu, Nicolás Astorga, Nabeel Seedat, and Mihaela van der Schaar. Large language models to enhance bayesian optimization. In *International Conference on Learning Representations*, 2024.

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, pp. 19624–19632, 2025.

Amin Nayebi, Alexander Munteanu, and Matthias Poloczek. A framework for bayesian optimization in embedded subspaces. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 4752–4761. PMLR, 2019.

Giang Ngo, Dat Phan Trong, Dang Nguyen, Sunil Gupta, and Svetha Venkatesh. Adaptive acquisition selection for bayesian optimization with large language models. In *International Conference on Learning Representations*, 2026.

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

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

Yanfei Peng, Guozhen Tan, and Huaiwei Si. Rta-ir: A runtime assurance framework for behavior planning based on imitation learning and responsibility-sensitive safety model. *Expert Systems with Applications*, 232:120824, 2023.

Aaditya Ramdas, Peter Grünwald, Vladimir Vovk, and Glenn Shafer. Game-theoretic statistics and safe anytime-valid inference. *Statistical Science*, 38:576–601, 2023.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, 2006.

James Requeima, John Bronskill, Dami Choi, Richard E. Turner, and David Duvenaud. LLM processes: Numerical predictive distributions conditioned on natural language. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Kenan Šehić, Alexandre Gramfort, Joseph Salmon, and Luigi Nardi. Lassobench: A high-dimensional hyperparameter optimization benchmark suite for lasso. In *Proceedings of the First International Conference on Automated Machine Learning*, volume 188 of *Proceedings of Machine Learning Research*, pp. 2/1–24. PMLR, 2022.

Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams, and Nando de Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1): 148–175, 2016.

Jaehyeok Shin, Aaditya Ramdas, and Alessandro Rinaldo. e-detectors: A nonparametric framework for sequential change detection. *The New England Journal of Statistics in Data Science*, 2:229–260, 2024.

Richard Cornelius Suwandi, Feng Yin, Juntao Wang, Renjie Li, Tsung-Hui Chang, and Sergios Theodoridis. Adaptive kernel design for bayesian optimization is a piece of CAKE with LLMs. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Jinjin Tian and Aaditya Ramdas. Online control of the familywise error rate. *Statistical Methods in Medical Research*, 30:976–993, 2021.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012.

Vladimir Vovk and Ruodu Wang. e-values: Calibration, combination and applications. *The Annals of Statistics*, 49:1736–1754, 2021.

Linnan Wang, Rodrigo Fonseca, and Yuandong Tian. Learning search space partition for black-box optimization using monte carlo tree search. In *Advances in Neural Information Processing Systems*, volume 33, pp. 19511–19522, 2020.

Zi Wang, Clement Gehring, Pushmeet Kohli, and Stefanie Jegelka. Batched large-scale bayesian optimization in high-dimensional spaces. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84, pp. 745–754. PMLR, 2018.

Ronald L. Wasserstein and Nicole A. Lazar. The asa's statement on p-values: Context, process, and purpose. *The American Statistician*, 70:129–133, 2016.

Zhitong Xu, Haitao Wang, Jeff M. Phillips, and Shandian Zhe. Standard gaussian process is all you need for high-dimensional bayesian optimization. In *The Thirteenth International Conference on Learning Representations*, 2025.

A Interpretation of Sequential Predictive Evidence

This section provides further justification for the sequential predictive evidence used in CLUE-BO, using the notation introduced in the main text. The one-step log predictive ratio admits a relative KL-divergence interpretation without requiring either GP to be correctly specified. Under an idealized model-based null, the multi-start accumulation further admits an e-process interpretation. These two perspectives respectively motivate the predictive comparison and its sequential accumulation.

A.1 Relative Predictive Interpretation

Recall that, before observing y_t, the active and shadow GPs provide predictive densities $q^a_{t-1}(y \mid \mathbf{x}_t)$ and $q^s_{t-1}(y \mid \mathbf{x}_t)$, and CLUE-BO computes

$$\Delta_t = \log \frac{q^s_{t-1}(y_t \mid \mathbf{x}_t)}{q^a_{t-1}(y_t \mid \mathbf{x}_t)}.$$

Let $\mathcal{G}_t$ and g_t be defined as in the main text. Whenever the relevant divergences are finite,

$$\begin{aligned}\mathbb{E}_{g_t}[\Delta_t \mid \mathcal{G}_t] &= \int g_t(y) \log \frac{q^s_{t-1}(y \mid \mathbf{x}_t)}{q^a_{t-1}(y \mid \mathbf{x}_t)}\, dy \\ &= D_{\mathrm{KL}}\big(g_t \| q^a_{t-1}\big) - D_{\mathrm{KL}}\big(g_t \| q^s_{t-1}\big)\,.\end{aligned}$$

Hence, a positive expected increment indicates that the shadow predictive distribution is closer to the conditional observation law than the active predictive distribution in KL divergence, whereas a negative expected increment indicates the opposite (Gneiting & Raftery, 2007). This interpretation does not require either GP predictive distribution to coincide with the true conditional law, and therefore remains meaningful under model misspecification.

A.2 Model-Based e-Process Interpretation

We next justify the finite-horizon accumulation $E^{(L)}_n$. Within a monitoring episode, we follow the local indexing of the main text, where n is the current local step and τ is a possible onset time. Consider the idealized model-based null under which the denominator predictive distribution in the log ratio coincides with the conditional observation law. For the generic active–shadow comparison, this means

$$H_0 : \qquad g_t(y) = q^a_{t-1}(y \mid \mathbf{x}_t)$$

for the iterations within the episode. Under H_0, assuming that the density ratio is well defined,

$$\begin{aligned}\mathbb{E}\left[e^{\Delta_t} \mid \mathcal{G}_t\right] &= \int q^a_{t-1}(y \mid \mathbf{x}_t) \frac{q^s_{t-1}(y \mid \mathbf{x}_t)}{q^a_{t-1}(y \mid \mathbf{x}_t)}\, dy \\ &= 1.\end{aligned}$$

For each possible onset $\tau \in \{1, \dots, L\}$, define

$$M_{\tau,n} = \begin{cases} 1, & n < \tau, \\ R_{\tau,n}, & n \geq \tau, \end{cases}$$

where $R_{\tau,n}$ is defined in the main text as

$$R_{\tau,n} = \exp\left(\sum_{i=\tau}^{n} \Delta_i\right).$$

Since each one-step likelihood ratio has a conditional expectation one under H_0,

$$\mathbb{E}\left[M_{\tau,n} \mid \mathcal{G}_n\right] = M_{\tau,n-1},$$

so $M_{\tau,n}$ is a nonnegative likelihood-ratio martingale initialized at one.

Averaging over all L possible onset times gives the following.

$$\begin{aligned} \frac{1}{L}\sum_{\tau=1}^{L} M_{\tau,n} &= \frac{\sum_{\tau=1}^{n} R_{\tau,n} + \sum_{\tau=n+1}^{L} 1}{L} \\ &= \frac{L - n + \sum_{\tau=1}^{n} R_{\tau,n}}{L} \\ &= E_n^{(L)}. \end{aligned}$$

Therefore, under the corresponding model-based null, $E_n^{(L)}$ is a nonnegative martingale initialized at one and hence is an e-process (Ramdas et al., 2023). This derivation also explains the term $L - n$: it corresponds to onset components that have not yet been reached and therefore remain equal to one.

Ville's inequality then yields

$$\Pr_{H_0}\left(\max_{0 \leq n \leq L} E_n^{(L)} \geq \frac{1}{\alpha}\right) \leq \alpha.$$

Thus, when the denominator predictive distribution correctly specifies the conditional observation law, the threshold $1/\alpha$ has the standard sequential-testing interpretation.

The same argument applies to each directional statistic in CLUE-BO by treating the denominator of its log predictive ratio as the corresponding model-based null: the semantic GP for Δ_t^{det}, the neutral GP for Δ_t^{acc}, the candidate GP for Δ_t^{rej}, and the neutral GP for $\Delta_t^{\mathrm{react}}$. Adaptive BO query selection does not alter the argument because $\mathbf{x}_t$ and both predictive distributions are fixed before y_t is observed and are included in $\mathcal{G}_t$, consistent with the prequential setting (Dawid, 1984).

A.3 Scope and Restarted Monitoring

The preceding e-process interpretation is deliberately model-based. In practical black-box optimization, neither GP predictive distribution is generally expected to coincide exactly with the true conditional observation law. CLUE-BO therefore does not use the martingale argument as an unconditional or model-free statistical guarantee. Instead, the KL identity provides a relative predictive interpretation under model misspecification, while the likelihood-ratio construction motivates a principled way to accumulate such comparisons over time and across possible onset points.

Consequently, a threshold crossing should be interpreted as strong accumulated predictive evidence in favor of one surrogate relative to another, rather than as proof that either surrogate is the true data-generating model. This is precisely the role required by the controller: the statistic determines when semantic guidance should be reconsidered, validated, or reactivated.

Because monitoring may restart during optimization, CLUE-BO uses the decreasing level schedule

$$\alpha_k = \frac{\alpha}{k(k+1)}, \qquad k = 1, 2, \dots,$$

inspired by alpha-spending principles (Tian & Ramdas, 2021). Since

$$\sum_{k=1}^{\infty} \alpha_k = \alpha,$$

later restarts receive smaller nominal levels and larger evidence thresholds $1/\alpha_k$, requiring stronger accumulated evidence for a threshold-triggered transition. In CLUE-BO, this schedule is instantiated as α_c for semantic and neutral monitoring episodes and as α_a for candidate proposals generated by reflections. The fixed-horizon timeout in candidate validation is an algorithmic trigger for further reflection, rather than a statistical rejection event.

B Experimental Details

B.1 Benchmark Details

We provide additional details on the six high-dimensional benchmarks used in our experiments. Except for Hopper-v4, we follow the benchmark implementations of Xu et al. (2025). For CLUE-BO, each benchmark is accompanied by a task description and variable-level semantics derived from its parameterization. These descriptions specify the role of each decision variable in the underlying problem and are used as the semantic inputs for constructing dimension-wise lengthscale priors.

Hopper-v4 (33D). Hopper-v4 is a MuJoCo continuous-control task (Todorov et al., 2012). Following the linear-policy formulation of Wang et al. (2020), the policy is parameterized by a 3×11 matrix, yielding 33 decision variables. The matrix maps the normalized 11-dimensional observation to the three-dimensional motor action. Each optimization variable therefore corresponds to one state-to-action coefficient, with its semantics determined by the associated observation coordinate and action dimension. We use the Hopper-v4 environment and maximize the cumulative rollout reward.

NAS201 (30D). NAS201 (Dong & Yang, 2020) is a tabular neural architecture search benchmark. Each cell is represented as a directed acyclic graph with four nodes and six edges, where each edge selects one of five candidate operations: zero, skip connection, 1×1 convolution, 3×3 convolution, or 3×3 average pooling. Following the 30-dimensional parameterization used by Xu et al. (2025), the six edge-wise operation choices are represented by 6×5 decision variables. The variable semantics identify both the corresponding graph edge and candidate operation, providing architectural information beyond the numerical encoding. We use the CIFAR-100 benchmark and maximize the architecture accuracy.

Rover (60D). Rover is a trajectory-planning benchmark adapted from Wang et al. (2018). A trajectory is specified by 30 waypoints in a two-dimensional environment, resulting in 60 decision variables. Consecutive pairs of variables determine the horizontal and vertical coordinates of each waypoint, and a smooth trajectory is constructed through these points. The objective rewards trajectories that connect the prescribed start and goal while avoiding costly regions and obstacles. Accordingly, the semantic description of each dimension contains its waypoint index and coordinate axis, exposing the spatial structure of the trajectory parameterization.

DNA (180D). DNA is the 180-dimensional real-world benchmark from LassoBench (Šehić et al., 2022), which formulates hyperparameter optimization for weighted Lasso regression on a DNA dataset. The decision vector contains 180 feature-wise regularization parameters controlling the amount of shrinkage applied to the corresponding regression coefficients. The dimensions therefore share the common role of regularization while remaining associated with distinct input features. LassoBench reports that the effective structure of this task is substantially lower-dimensional than its ambient dimension, making it a useful test case for dimension-wise relevance modeling.

SVM (388D). SVM is a 388-dimensional support vector regression hyperparameter optimization benchmark (Eriksson & Jankowiak, 2021). Its decision variables consist of three regularization-related parameters and 385 kernel lengthscale parameters. The former have explicit hyperparameter-level roles, whereas the latter form a much larger group of feature-specific kernel parameters. Consequently, compared with the other benchmarks, SVM provides relatively limited heterogeneous dimension-level semantics, making it a challenging case for semantic prior construction.

HumanoidStandup-v4 (1003D). HumanoidStandup-v4 is a high-dimensional trajectory-optimization problem based on the MuJoCo humanoid simulator (Todorov et al., 2012). Following the parameterization of Xu et al. (2025), we optimize a sequence of 59 actions, each containing 17 motor-control variables, yielding $59 \times 17 = 1{,}003$ dimensions. Unlike a policy-based formulation, these variables directly specify the action trajectory. We retain this 1,003-dimensional parameteri-

zation while using the HumanoidStandup-v4 environment. Each dimension is therefore associated with both a specific actuator and a trajectory step, providing explicit temporal and control semantics for semantic prior construction.

B.2 Implementation Details

Our implementation builds on the sequential BO framework of the SBO codebase (Xu et al., 2025). We retain its full-dimensional GP optimization procedure, while introducing LLM-derived ARD lengthscale priors and evidence-driven prior adaptation. Inputs are normalized to $[0, 1]^d$, and objective values are standardized using the mean and standard deviation of all observations available at the current iteration. At each BO iteration, an exact GP with an ARD Matérn-$5/2$ kernel and a learnable output scale is refitted using all accumulated observations. GP hyperparameters are optimized by MAP estimation with BoTorch's marginal-likelihood fitting routine using L-BFGS-B, with Adam used as a fallback when fitting fails.

For the semantic GP, the dimension-wise lengthscale priors introduced in Section 3.2 are registered as LogNormal priors. Following the prior-scale choice in VBO (Hvarfner et al., 2024), we set $\sigma_{\text{prior}} = \sqrt{3}$. Given prior mode m_j, we use

$$\ell_j \sim \text{LogNormal}\left(\log m_j + \sigma^2_{\text{prior}}, \sigma^2_{\text{prior}}\right),$$

and initialize ℓ_j at m_j. The prior parameters remain fixed during each GP fit. We select one query per iteration using UCB,

$$\alpha(\mathbf{x}) = \mu(\mathbf{x}) + \sqrt{\beta}\,\sigma(\mathbf{x}), \qquad \beta = 1.5.$$

The acquisition function is optimized over $[0, 1]^d$ using BoTorch's `optimize_acqf` with $q = 1$, 10 restarts, and 1,000 raw samples. If acquisition optimization fails, we fall back to uniform random sampling over the search domain.

When the controller requires a predictive comparison, the active and shadow GPs are fitted independently to the same accumulated observations. The active GP determines the next query, while the shadow GP is evaluated only at the selected point; therefore, maintaining the shadow model does not consume additional black-box evaluations.

Each run begins with a Latin hypercube design of 20 points, except for HumanoidStandup-v4, which uses 50. The initial design is followed by 300 additional sequential BO iterations, yielding total evaluation budgets of 320 and 350, respectively. We use ten initial-design seeds, $s \in \{0, \ldots, 9\}$. CLUE-BO and the matched SBO baseline use the same LHS construction and initial-design size for each benchmark and seed.

B.3 Ablation Configurations

We consider nine ablation variants in two groups: four ablations of the core adaptation mechanisms and five ablations of the reflection context. Unless otherwise specified, all variants use the same optimization protocol and hyperparameter settings as the full CLUE-BO. For the reflection-context ablations, only the specified information source is removed from the reflection prompt, while semantic-prior initialization and all other controller components remain unchanged. Table 1 summarizes the configurations.

C Additional Results

C.1 Optimization Trajectories

Figure 4 presents the complete best-so-far optimization trajectories corresponding to the main results. These trajectories complement the terminal performance distributions in Figure 2 by showing how performance evolves over the optimization budget.

The trajectories show that CLUE-BO does not necessarily dominate from the earliest iterations, but often becomes increasingly competitive as optimization proceeds. On some tasks, the methods remain comparable during the early stages of optimization before CLUE-BO gradually establishes an

Table 1: Configurations of the nine ablation variants.

(a) Core adaptation mechanisms

Variant	Modification
w/o LLM	Uses the neutral prior, $r_j = 1$, throughout optimization.
w/o Reflection	Fixes the initial LLM-derived semantic prior without subsequent reflection.
w/o Validation	Activates each reflected semantic prior immediately without candidate validation.
w/o Neutral Recovery	Disables reactivation of dormant semantic guidance after neutral fallback.

(b) Reflection context

Variant	Removed Information
w/o Task Semantics	Task description and variable-level semantics.
w/o Prior Allocations	Original and current dimension-wise prior allocations.
w/o Trajectory Progress	Optimization trajectory and progress information.
w/o Predictive Evidence	Accumulated sequential predictive evidence.
w/o ARD Displacement	Fitted ARD lengthscales and their deviations from the corresponding prior modes.

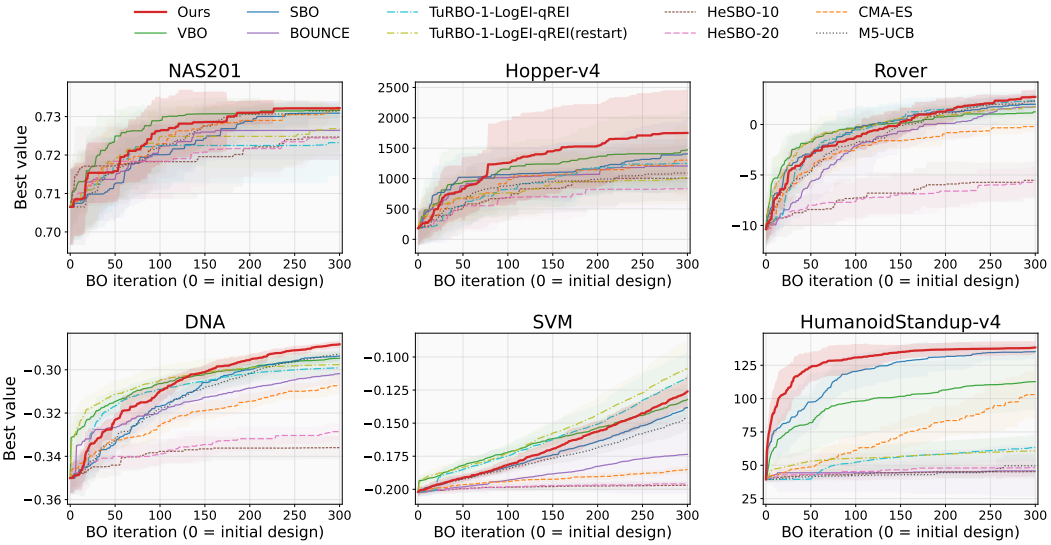

Figure 4: Best-so-far optimization trajectories on the six high-dimensional benchmarks. Each curve shows the mean best objective value across runs, and the shaded region denotes one standard deviation. Iteration 0 corresponds to the initial design. Higher values are better.

advantage, whereas on others it benefits from semantic guidance earlier and maintains this advantage throughout optimization. SVM remains the main exception, consistent with the terminal results reported in the main text. These patterns are consistent with the adaptive design of CLUE-BO: when the initial semantic prior is well aligned with the objective structure, it can provide useful guidance early in optimization; when it is less well aligned, accumulated predictive evidence can trigger reflection and revise the prior as additional observations become available. CLUE-BO can therefore exploit useful semantic structure without relying on the initial LLM hypothesis throughout the entire optimization process.

C.2 Sensitivity to the Evidence Level

We further study the sensitivity of CLUE-BO to the evidence level α, which controls the decision threshold of the evidence-driven controller. A larger α corresponds to a more permissive evidence

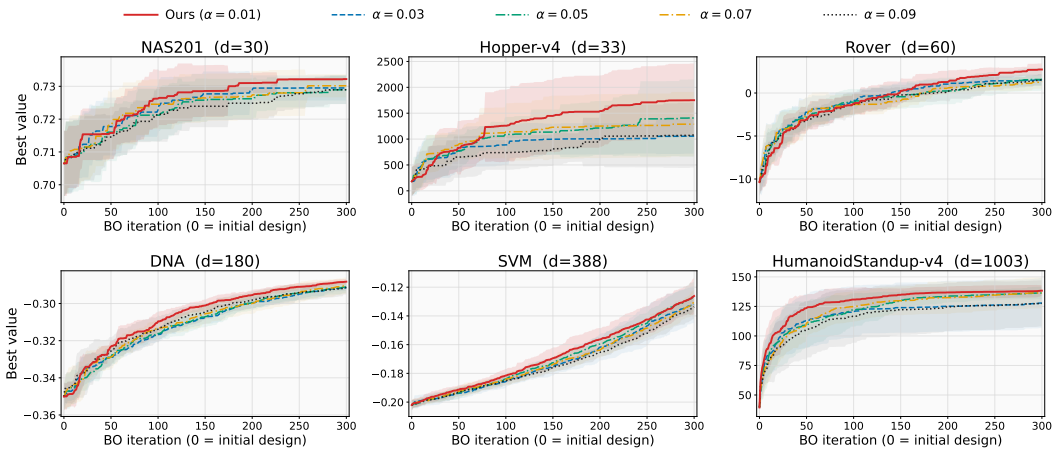

Figure 5: Runtime performance with different values of α.

threshold, allowing the controller to trigger reflection or prior switching with less accumulated predictive evidence. We consider

$$\alpha \in \{0.01, 0.03, 0.05, 0.07, 0.09\},$$

and use a single default value of $\alpha = 0.01$ throughout the main experiments without task-specific tuning.

As shown in Fig. 5, $\alpha = 0.01$ provides stable and competitive performance across all six tasks. On NAS201, DNA, and SVM, the differences across α values are relatively small, and the overall optimization trajectories remain similar, suggesting that these tasks are not particularly sensitive to the exact evidence threshold. Rover and HumanoidStandup-v4 show a similar overall trend, although the more conservative setting tends to achieve better final or early-stage performance.

Hopper-v4 shows a stronger dependence on α, with performance decreasing more noticeably as α increases. This is consistent with the role of α in the controller. A larger α lowers the amount of evidence required for a state transition, making the controller more likely to revise the current semantic prior based on limited short-term predictive feedback. When these predictive comparisons are noisy, a permissive threshold may therefore lead to premature reflection or prior switching. In contrast, a smaller α requires stronger accumulated evidence before taking these actions, reducing the chance of reacting to transient predictive fluctuations.

Overall, CLUE-BO remains reasonably stable under moderate changes in α on most tasks, while some tasks benefit from more conservative evidence control. These results support our use of a fixed default $\alpha = 0.01$ across all experiments.

C.3 Sensitivity to the Validation Horizon

We further examine the sensitivity of CLUE-BO to the candidate-validation horizon H. The horizon determines how long a reflected candidate prior is evaluated before the controller decides whether to activate it. A shorter horizon allows the controller to make validation decisions using fewer subsequent observations, while a longer horizon evaluates the candidate over a broader sequence of predictive comparisons before activation. We evaluate

$$H \in \{20, 40, 60, 80, 100\},$$

and use $H = 40$ as the default throughout the main experiments.

As shown in Fig. 6, the overall performance is fairly stable across different horizon values on most tasks. On NAS201, DNA, and SVM, the optimization trajectories remain close throughout the evaluation budget, suggesting that the method is not particularly sensitive to the exact candidate-validation horizon in these cases. A similar pattern is observed on Rover and HumanoidStandup-v4, where $H = 40$ remains consistently competitive and generally provides stronger early-stage or final performance.

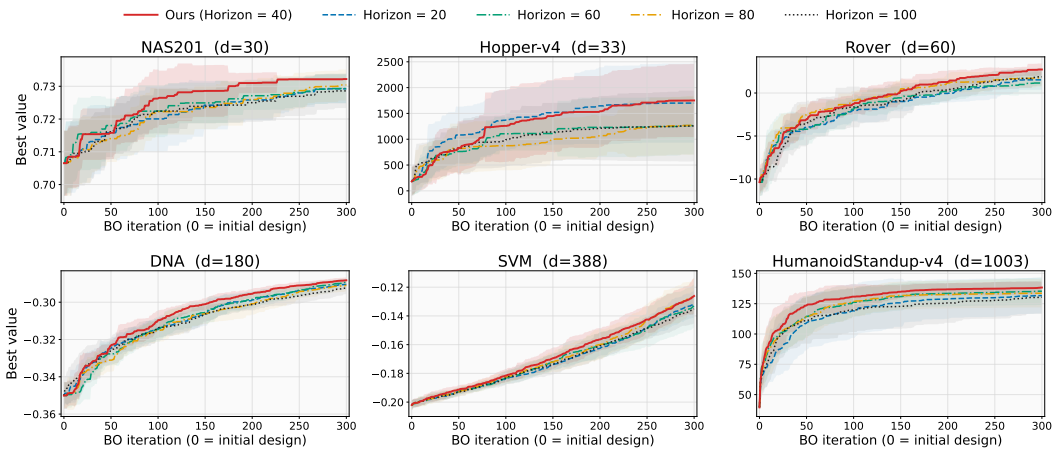

Figure 6: Runtime performance with different values of H.

Table 2: Information provided to the LLM during evidence-triggered reflection.

Context component	**Information provided**
Task semantics	The same benchmark-specific objective, parameterization, and variable semantics used for initial semantic prior construction in Appendix D.1.
Prior allocations	The original cold-start prior and the prior under consideration immediately before reflection, including their dimension-wise mode ratios relative to L_{base}.
Predictive evidence	The current sequential evidence statistic and its trigger threshold, together with the dominant evidence segment and the accumulated predictive log-score advantage over that segment.
Trajectory progress	The current BO iteration and budget fraction, the best objective values before and after the dominant evidence segment, the improvement over this segment, and summary statistics of the predictive comparison.
ARD displacement	Dimension-wise fitted ARD lengthscales and their relationship to the current prior modes and neutral baseline, including standardized displacement and the consistency of its direction over the dominant evidence segment.

Hopper-v4 shows a clearer dependence on the validation horizon. The performance with $H = 20$ is very close to that with $H = 40$, whereas larger horizons such as $H = 60$, 80, and 100 tend to yield weaker optimization performance. This behavior is consistent with the role of H in candidate validation: a short horizon may cause the controller to make activation decisions based on relatively limited validation evidence, whereas an overly long horizon may delay the activation of a useful reflected prior even when its predictive advantage becomes apparent relatively early.

Overall, these results suggest that a moderate candidate-validation horizon provides a useful balance between timely adaptation and reliable candidate assessment. We therefore use $H = 40$ as a fixed default across all experiments, without task-specific tuning.

D Prompts

D.1 Initial Semantic Prior Prompt

At initialization, CLUE-BO prompts the LLM to construct a dimension-wise semantic hypothesis before any BO observations are available. The prompt consists of a shared instruction defining the meaning of the relative lengthscale ratios, an objective-free baseline specification, and a benchmark-specific task payload describing the objective, parameterization, and variable semantics. The LLM is explicitly restricted to task semantics and general scientific knowledge, and is not provided with benchmark optima, previous runs, fitted importance estimates, acquisition values, or other optimization outcomes. The resulting structured output specifies relative ARD lengthscale-prior modes rather than BO query points.

Shared instruction. The following instruction is shared across all benchmarks. For readability, minor whitespace is normalized.

```
You allocate ARD lengthscale-prior modes before Bayesian optimization.

All inputs are normalized to [0,1]. The compiler supplies an
objective-free Matern-5/2 baseline L_base. Return only positive mode
ratios relative to L_base: below one indicates a shorter prior
lengthscale and greater sensitivity to changes along that dimension;
one is neutral; and above one indicates a longer prior lengthscale
and smoother variation.

Use task semantics and general scientific knowledge only. Do not use
benchmark optima, previous runs, hidden feature rankings, acquisition
values, or absolute lengthscales.

Dimensions remain independently assignable even when they are grouped
for readability. Keep overrides disjoint, prefer conservative
mechanism-based allocations.

Selected baseline:
- L_base: {base_lengthscale}

{benchmark-specific task description and variable semantics}
```

Benchmark-specific task specifications. The benchmark-specific context appended to the shared instruction is reported below. We preserve the objective descriptions, parameterization rules, and variable semantics used in the implementation as closely as possible. Repetitive per-dimension entries are abbreviated with "..." only for presentation; the corresponding complete specifications are provided to the LLM during optimization.

Table 3: Benchmark-specific task specifications provided to the LLM for initial semantic prior construction. Repetitive dimension entries are abbreviated only for presentation.

Problem	Task-specific context
Hopper-v4	**Objective.** Optimize a 33-dimensional linear feedback policy for Gymnasium Hopper-v4. A candidate vector in $[0,1]^{33}$ is mapped elementwise to controller weights in $[-1.4, 1.4]$ and reshaped in row-major order into a 3×11 policy matrix. Each row produces one of the thigh, leg, and foot joint actions, and each column corresponds to one of the 11 listed observations. At each simulation step, the policy matrix multiplies an observation vector standardized using fixed benchmark-provided mean and standard-deviation statistics, which remain unchanged throughout optimization. The resulting actions are passed directly to the environment. The objective is the mean cumulative reward over five rollouts and is maximized. **Parameterization.** The three actuator outputs are `thigh`, `leg`, and `foot`. The observations are `torso_height`, `torso_pitch`, `thigh_angle`, `leg_angle`, `foot_angle`, `torso_x_velocity`, `torso_z_velocity`, `torso_pitch_velocity`, `thigh_angular_velocity`, `leg_angular_velocity`, and `foot_angular_velocity`. The flattening rule is $\texttt{dimension_index} = 11 \times \texttt{action_index} + \texttt{observation_index}.$ **Variable semantics.** Each dimension is one independently assignable entry of the 3×11 policy matrix and represents a linear-feedback weight from one observation to one actuator, e.g., `weight_thigh_from_torso_height`, `weight_thigh_from_torso_pitch`, ..., `weight_foot_from_foot_angular_velocity`. Actuator blocks organize the task description and are not atomic groups.

Continued on the next page.

Table 3 continued from the previous page.

Problem	Task-specific context
NAS-Bench-201	**Objective.** Optimize a NAS-Bench-201 cell architecture for CIFAR-100 validation accuracy. The cell is a four-node directed acyclic graph with six ordered edges and is reused throughout the benchmark network. The optimizer supplies 30 continuous scores in $[0, 1]$, arranged as six five-score edge blocks. Within each block, argmax selects exactly one operation. Consequently, many continuous points decode to the same discrete architecture and the objective is piecewise constant in score space. The tabular benchmark returns final-epoch CIFAR-100 validation accuracy divided by 100, which is maximized. **Parameterization.** The six directed edges are $(1, 2)$, $(1, 3)$, $(1, 4)$, $(2, 3)$, $(2, 4)$, and $(3, 4)$. For each edge, the five operations are ordered as `skip_connect`, `none`, `nor_conv_3x3`, `nor_conv_1x1`, and `avg_pool_3x3`. For each edge independently, the operation with the largest score is selected; ties select the lowest operation index. The dimension indexing rule is $$\texttt{dimension_index} = 5 \times \texttt{edge_index} + \texttt{operation_index}.$$ **Variable semantics.** All 30 score coordinates remain independently assignable ARD dimensions, e.g., `score_edge_1_to_2_skip_connect`, `score_edge_1_to_2_none`, ..., `score_edge_3_to_4_avg_pool_3x3`. Only the within-edge score ranking affects the decoded architecture. Edge blocks organize the prompt and are not atomic groups. General operation and graph semantics may support conservative hypotheses, but they do not establish empirical sensitivity or the best operation for a particular edge. No queried accuracies, fitted importance estimates, previous optimization results, best architecture, or benchmark optimum are provided.
Rover	**Objective.** Optimize a two-dimensional rover trajectory represented by 30 two-dimensional cubic B-spline control points. The desired start is $(0.05, 0.05)$ and the goal is $(0.95, 0.95)$. Endpoints are not hard-fixed and incur 10 times their L_1 error. Free-space travel costs 0.05 per unit length, while fixed axis-aligned obstacles cost 20 per unit length. The objective is to maximize 5 minus the total trajectory cost. **Parameterization.** Each of the 30 ordered control points contains an x- and a y-coordinate. Normalized coordinates in $[0, 1]$ are mapped to the physical range $[-0.1, 1.1]$. Cubic B-spline bases have compact overlapping support, so each control point affects a local trajectory segment together with nearby points. **Variable semantics.** The 60 dimensions are `control_point_00_x`, `control_point_00_y`, ..., `control_point_29_x`, and `control_point_29_y`. For each point, the prompt specifies its relative position in the ordered B-spline sequence. Its actual spatial location is unknown before evaluation, and the LLM is explicitly instructed not to assume that an intermediate point lies near an obstacle. Control-point groups organize the semantics only; x and y remain independently assignable. No trajectory, observations, fitted sensitivity estimates, or optimum are supplied.
DNA	**Objective.** Tune feature-wise Weighted-Lasso penalties for classifying whether the center of a 60-nucleotide DNA window is a splice junction. Positions range from -30 to 29 relative to the window center. Each position is encoded by three indicators, A, C, and G, while T is the all-zero reference. The 180 optimization variables independently control feature regularization penalties, rather than the DNA sequence itself. The objective is to maximize negative the cross-validation loss. **Parameterization.** Each relative DNA position contains three independently tunable penalty controls corresponding to the A, C, and G indicator features. A larger normalized control maps to stronger regularization. **Variable semantics.** Representative dimensions are `penalty_pos_-30_A_indicator`, `penalty_pos_-30_C_indicator`, `penalty_pos_-30_G_indicator`, ..., `penalty_pos_29_A_indicator`, `penalty_pos_29_C_indicator`, and `penalty_pos_29_G_indicator`. Each parameter tunes the Weighted-Lasso penalty of a feature; it does not directly choose or mutate a nucleotide.

Continued on the next page.

Table 3 continued from the previous page.

Problem	Task-specific context
SVM	**Objective.** Tune an RBF support-vector regressor for CT-slice localization and maximize negative held-out RMSE. Inputs comprise one patient-identifier feature, 240 bone-histogram features, and 144 air-histogram features. The first 385 controls divide the corresponding features by $\exp(4x^2 - 2)$, while the final three controls determine ϵ, γ, and C. All optimization controls lie in $[0, 1]$. **Parameterization and variable semantics.** Dimension 0 is a continuous scale control for a fixed patient-identifier feature column. It does not select a patient, and the identifier has no meaningful ordinal geometry. Dimensions 1–240 independently control the scales of anonymous bone-structure polar-histogram bins. Dimensions 241–384 independently control the scales of anonymous air-inclusion polar-histogram bins. Individual bin geometry is unavailable, and the LLM is instructed not to infer semantics from anonymous bin numbers. The remaining dimensions are explicit SVR hyperparameters: $x_{385}: \quad \epsilon = 0.01 \times 100^{x_{385}^2}, \quad \epsilon \in [0.01, 1],$ $x_{386}: \quad \gamma = 0.1 \times 30^{x_{386}^2}, \quad \gamma \in [0.1, 3],$ $x_{387}: \quad C = 0.01 \times 500^{x_{387}^2}, \quad C \in [0.01, 5].$ All dimensions remain independently assignable. Individual histogram-bin geometry, fitted importance estimates, prior runs, and the benchmark optimum are not supplied.
HumanoidStandup-v4	**Objective.** Optimize a 1,003-dimensional open-loop action sequence for humanoid stand-up control. The sequence contains 59 time steps with 17 joint actions at each step. Each optimization dimension specifies one joint action at one time step. Because different stages of the stand-up process may have different motion states and control requirements, actions at different time steps are explicitly distinguished by their temporal positions. The objective is cumulative stand-up reward and is maximized. **Parameterization.** The action sequence has shape 59×17. The 17 action variables are `abdomen_y`, `abdomen_z`, `abdomen_x`, `right_hip_x`, `right_hip_z`, `right_hip_y`, `right_knee`, `left_hip_x`, `left_hip_z`, `left_hip_y`, `left_knee`, `right_shoulder_1`, `right_shoulder_2`, `right_elbow`, `left_shoulder_1`, `left_shoulder_2`, and `left_elbow`. The internal flattening rule is $\texttt{dimension_index} = 17 \times \texttt{time_step} + \texttt{variable_index}.$ **Variable semantics.** Each dimension is one continuous open-loop joint action at one time step, e.g., `action_abdomen_y_at_time_00`, . . ., `action_left_elbow_at_time_58`. An ARD lengthscale describes objective sensitivity when one torque variable is perturbed; it does not directly encode smoothness between neighboring actions in the time sequence. Because the objective is cumulative reward, terminal controls are not assumed to be especially important without a task-specific semantic reason.

D.2 Reflection Prompt

When accumulated predictive evidence triggers reflection, CLUE-BO asks the LLM to reassess the current semantic prior using both the original task semantics and feedback collected during optimization. In addition to the semantic information used at initialization, the reflection prompt provides the current prior allocation, predictive evidence, optimization progress, and dimension-wise GP fitting information. The LLM is instructed to reason about these signals jointly rather than treating any individual statistic as direct evidence of variable importance or prior failure. The benchmark-specific task semantics are identical to those used for initial prior construction and are therefore not repeated here.

Shared reflection instruction. The core instruction used for evidence-triggered reflection is shown below. Interface-only instructions for enforcing structured output are omitted for readability.

```
You are revising an ARD lengthscale prior during Bayesian optimization.

The supplied five-part context includes task semantics, the original and active prior decisions, sequential predictive evidence, optimization statistics, and fitted-prior displacement.

Reason before deciding. Use the role-based reflection state, current champion, evidence challenger, and role assignment as authoritative.

Predictive evidence against the current semantic prior only establishes that its predictive performance is worse relative to the challenger. It does not establish that the fully neutral prior is superior to every possible revised non-neutral prior.

Fitted displacement is diagnostic evidence, not feature importance or causal proof of prior failure. Interpret it jointly with task semantics and predictive evidence, accounting for possible estimation noise in high-dimensional, data-sparse settings.

Use the coordinates exactly:
- current_effective_prior_mode_ratio =
  current compiled prior mode / L_base;
- fitted_to_current_prior_ratio =
  fitted lengthscale / current prior mode;
- fitted_to_neutral_base_ratio =
  fitted lengthscale / L_base.

Interpret fitted displacement relative to the current prior center, not relative to one.

There is no requirement to review every dimension individually. Semantic groups and position labels provide organization but do not imply that all members should receive the same update. When grouped dimensions support different directions or substantially different target ratios, treat them separately.

You may choose one of the following reflection actions:
- recenter: propose a new non-neutral semantic prior;
- neutralize: remove semantic shifts so that all prior-mode ratios equal one;

Candidate ratios specify target prior-mode centers relative to L_base, rather than absolute lengthscales or fitted-to-prior ratios. A recentered candidate is validated prospectively on future observations before adoption.

{five-part reflection context}
```

Reflection context. The reflection context is organized into five complementary sources of information, summarized in Table 2. These are also the five information sources examined in the reflection-context ablations.

The five context components play complementary roles in reflection. Predictive evidence determines when the current hypothesis warrants reconsideration, while task semantics, prior history, optimization progress, and dimension-wise displacement provide additional information for deciding whether and how the prior should be revised. The prompt does not prescribe which source should dominate, allowing the reflection decision to adapt to the evidence available at the current optimization state.

E LLM RESPONSE EXAMPLES

We provide representative LLM responses to illustrate how CLUE-BO forms semantic lengthscale hypotheses and revises them using optimization feedback. For initial prior construction, we reproduce the raw model outputs with only whitespace normalization. For reflection, we report the structured LLM response recorded after schema validation, while preserving its reasoning text and proposed allocation. These examples are intended to illustrate the behavior of the semantic prior constructor and reflection mechanism rather than provide additional quantitative evidence.

E.1 INITIAL SEMANTIC PRIOR CONSTRUCTION

We first show two cold-start responses from Rover and HumanoidStandup-v4. The former provides a compact example of dimension-wise semantic allocation, whereas the latter illustrates how the LLM can exploit both actuator identity and temporal structure in a 1,003-dimensional search space.

Rover. The raw LLM response for the initial Rover prior is:

```
{
  "action": "allocate",
  "default_mode_ratio": 1.0,
  "overrides": [
    {
      "dimensions": [0, 1],
      "mode_ratio": 0.5,
      "hypothesis": "Start region: control point 0 is near the desired
      start (0.05,0.05). The trajectory must pass close to this point,
      so the function varies more rapidly here, requiring a shorter
      lengthscale."
    },
    {
      "dimensions": [58, 59],
      "mode_ratio": 0.5,
      "hypothesis": "Goal region: control point 29 is near the desired
      goal (0.95,0.95). The trajectory must end close to this point,
      so the function varies more rapidly here, requiring a shorter
      lengthscale."
    }
  ],
  "rationale": "The task is a trajectory optimization with 30 control
  points. The objective description specifies a desired start near
  (0.05,0.05) and a desired goal near (0.95,0.95). The endpoints are
  not hard-fixed, but deviations are penalized heavily (10x L1 error).
  This creates a strong local optimum near the start and goal regions,
  meaning the objective function will change rapidly with small
  adjustments to the first and last control points. Therefore, the
  lengthscale for the first and last control point coordinates
  (dimensions 0,1 and 58,59) should be shorter than the baseline to
  capture this sensitivity. The intermediate control points have no
  such strong local constraints, so the default baseline lengthscale
  is appropriate. No other dimensions have a clear semantic
  justification for a lengthscale override."
}
```

The response uses a neutral default and introduces non-neutral preferences only for the four endpoint coordinates directly associated with the stronger start and goal penalties. In particular, both endpoint groups receive $r_j = 0.5$, whereas all intermediate coordinates retain $r_j = 1$. This example illustrates the intended role of the LLM at initialization: it converts task semantics into a sparse hypothesis about relative dimension-wise sensitivity rather than attempting to infer an optimum or directly propose BO queries. Importantly, CLUE-BO treats this reasoning as a hypothesis rather than ground truth; as shown in Appendix E.2, part of this allocation is later revised by optimization evidence.

HumanoidStandup-v4. The 1,003-dimensional HumanoidStandup-v4 task yields a more structured allocation. The raw LLM response is:


```
{
  "action": "allocate",
  "default_mode_ratio": 1.0,
  "overrides": [
    {
      "hypothesis": "Weight-bearing lower-limb and trunk torques are
      direct determinants of lifting the body, extending from the
      ground-contact transition into upright support, so individual
      torque perturbations are plausibly somewhat more consequential
      than a generic action coordinate throughout the horizon.",
      "mode_ratio": 0.85,
      "time_steps": [
        0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,
        20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,
        36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,
        52,53,54,55,56,57,58
      ],
      "variables": [
        "abdomen_y", "abdomen_z", "abdomen_x",
        "right_hip_x", "right_hip_z", "right_hip_y",
        "right_knee",
        "left_hip_x", "left_hip_z", "left_hip_y",
        "left_knee"
      ]
    },
    {
      "hypothesis": "Upper-limb torques can contribute to initial
      momentum and body reorientation while rising from the ground,
      but their direct role is generally less central than lower-limb
      support torques; assign only a mild early-horizon sensitivity
      adjustment.",
      "mode_ratio": 0.92,
      "time_steps": [
        0,1,2,3,4,5,6,7,8,9,10,11,12,
        13,14,15,16,17,18,19,20,21,22,23,24
      ],
      "variables": [
        "right_shoulder_1", "right_shoulder_2", "right_elbow",
        "left_shoulder_1", "left_shoulder_2", "left_elbow"
      ]
    }
  ],
  "rationale": "Use a neutral default where task mechanics do not
  distinguish a coordinate. Apply conservative shorter modes only to
  biomechanically direct weight-bearing trunk and leg controls, and
  mildly to early arm controls that may assist the rise; no claims are
  made about unobserved rewards or trajectory-specific optima."
}
```


Unlike the Rover response, this allocation is structured jointly over actuator roles and temporal positions. Weight-bearing trunk and lower-limb controls receive a moderately shorter ratio of 0.85 throughout the trajectory, while shoulder and elbow controls receive a weaker adjustment of 0.92 only during the first 25 time steps. All remaining coordinates stay at the neutral ratio. The response therefore exhibits a graded semantic hypothesis rather than a binary partition of important and unimportant variables, while keeping all deviations relatively close to the neutral baseline.

E.2 Reflection and Prior Revision

We next examine a representative Rover run with seed 0 to illustrate how an initial LLM hypothesis is subsequently challenged, reconsidered, and selectively revised. The initial response above assigns

$$r_{0,1} = 0.5, \qquad r_{58,59} = 0.5, \qquad r_{\text{others}} = 1.$$

The resulting prior is not treated as fixed knowledge. At iteration 20, predictive evidence in favor of the neutral reference reaches 220.39, exceeding the current threshold of 200, and the reflection decision temporarily neutralizes the semantic prior. The previous semantic allocation is retained as a dormant challenger. At iteration 31, renewed predictive support for this dormant prior crosses the neutral-monitoring threshold, causing it to be reactivated without an additional LLM call.

After reactivation, the semantic prior is again monitored against the neutral reference. At iteration 37, the evidence statistic reaches

$$E_n^{(L)} = 1240.32 > 1200,$$

triggering another reflection. Over iterations 32–37, the neutral model attains a lower predictive RMSE than the current semantic model (0.727 versus 0.980). More importantly, the fitted ARD displacement differs across the two endpoint groups: the goal dimensions $\{58, 59\}$ have fitted lengthscales approximately 1.7–2.1 times their current prior modes, whereas the evidence for the start dimensions $\{0, 1\}$ is mixed.

LLM reflection response. Below we reproduce the structured response recorded for this reflection. The reasoning text and proposed ratios are unchanged; only formatting is normalized.

```
{
  "evidence_assessment":
    "The neutral challenger shows a substantial predictive advantage
    over the current semantic prior. The mixture e-value (1240.32)
    far exceeds the trigger threshold (1200), and the dominant segment
    (iterations 32-37) has a large log-score advantage for neutral
    (12.68). The neutral prior has a lower RMSE (0.727 vs 0.980) and
    a lower mean squared standardized residual (4.59 vs 8.68),
    indicating better calibrated predictions. The current semantic
    prior, which uses shorter lengthscales for the start and goal
    control points, is underperforming. The fitted lengthscale
    displacement shows that for the start point (dims 0,1), the fitted
    lengthscale is close to the prior mode (ratio ~1.1 for x, ~0.21
    for y), but for the goal point (dims 58,59), the fitted
    lengthscale
    is much larger than the prior mode (ratio ~1.7 for x, ~2.1 for y).
    This suggests the prior's short lengthscale for the goal region is
    too restrictive, causing poor predictions. The neutral prior's
    uniform lengthscale is performing better overall.",

  "conflicts": [
    {
      "target": "goal_region_prior",
      "dimensions": [58, 59],
      "description":
        "The semantic prior specifies a short lengthscale (ratio 0.5)
        for the goal control point (dims 58,59), hypothesizing rapid
        variation. However, the fitted lengthscale is much larger than
        the prior mode (ratio ~1.7-2.1), and the neutral prior
        (ratio 1.0) outperforms the semantic prior. This indicates the
        prior is too short for the goal region."
    }
  ],

  "uncertainties": [
    "The fitted displacement for the start region (dims 0,1) is mixed:
    dim 0 is close to the prior, dim 1 is much shorter. It is unclear
    if the short prior for the start region is beneficial or
    harmful.",
```

```
    "The neutral prior's advantage may be due to the goal region prior
    being too short, but the start region prior might still be
    appropriate. A fully neutral prior might not be optimal for the
    start region.",

    "The optimization has only used 12.3% of the budget, so the fitted
    lengthscales may still be noisy estimates."
  ],

  "summary":
    "The neutral prior is clearly outperforming the current semantic
    prior, primarily because the semantic prior's short lengthscale
    for
    the goal region is too restrictive. The fitted lengthscale for the
    goal point is much larger than the prior mode, and the neutral
    prior's uniform lengthscale captures the function better. The
    start
    region prior may still be useful, but the goal region prior is
    harmful. A revised semantic prior should keep the short
    lengthscale
    for the start region but relax the goal region to neutral or
    longer.",

  "confidence": "high",

  "changes": [
    {
      "target": "goal_region_prior",
      "evidence_dimensions": [58, 59],
      "update_basis": "combined",
      "reason":
        "The fitted lengthscale for the goal point (dims 58,59) is
    much
        larger than the current prior mode (ratio ~1.7-2.1), and the
        neutral prior outperforms the semantic prior. The prior's
    short
        lengthscale for the goal region is causing poor predictions.
        Relaxing it to neutral (ratio 1.0) aligns with the fitted
        evidence and the neutral prior's success."
    }
  ],

  "reflection_action": "recenter",

  "candidate_prior": {
    "default_mode_ratio": 1.0,
    "overrides": [
      {
        "dimensions": [0, 1],
        "mode_ratio": 0.5,
        "hypothesis":
          "Start region: control point 0 is near the desired start
          (0.05,0.05). The trajectory must pass close to this point,
          so the function varies more rapidly here, requiring a
          shorter lengthscale."
      }
    ],
    "rationale":
      "The neutral prior outperforms the current semantic prior, but
      the start region prior may still be useful. The goal region
    prior
      is harmful, so it is removed. The start region prior is retained
      because its fitted displacement is mixed and it may still
    provide
```

```
        benefit. The default for all other dimensions is neutral
        (ratio 1.0)."
    }
}
```

The resulting revision is therefore selective:

$$\underbrace{r_{0,1} = 0.5, \qquad r_{58,59} = 0.5}_{\text{prior before reflection}} \quad \longrightarrow \quad \underbrace{r_{0,1} = 0.5, \qquad r_{58,59} = 1.0}_{\text{reflected candidate}}.$$

Rather than interpreting the neutral model's predictive advantage as evidence that all semantic structure should be discarded, the LLM combines the global predictive comparison with dimension-wise displacement and retains the part of the original semantic hypothesis that remains plausible.

The example highlights the distinction between semantic reasoning and evidence-controlled adaptation in CLUE-BO. The LLM provides an interpretable structural hypothesis, but neither its initial allocation nor a reflected revision is trusted unconditionally. Predictive evidence first determines when the current hypothesis warrants reconsideration; task semantics and dimension-wise GP feedback then help determine which components should be preserved or revised. Finally, a newly proposed non-neutral prior must receive prospective predictive support before it is allowed to control acquisition. Thus, the LLM acts as a hypothesis generator and reviser, while the evidence controller determines when those hypotheses are allowed to influence BO.

F Related Works

We discuss three areas most closely related to CLUE-BO: structural methods for high-dimensional Bayesian optimization, LLM-guided black-box optimization, and sequential predictive model assessment. The common thread is how to introduce additional knowledge into BO when numerical observations are scarce, without giving up the statistical structure of the underlying optimizer.

F.1 Structural Approaches to High-Dimensional Bayesian Optimization

Many high-dimensional BO methods make the problem easier by imposing structure on the objective or the search space. Additive BO decomposes the objective into lower-dimensional components (Kandasamy et al., 2015), while embedding-based methods optimize through a lower-dimensional representation of the original domain (Nayebi et al., 2019; Letham et al., 2020). Other approaches assume sparsity over the input coordinates (Eriksson & Jankowiak, 2021), or confine the search to local or adaptively constructed regions. TuRBO, for example, performs BO within adaptive trust regions (Eriksson et al., 2019); BAxUS and Bounce instead adapt the subspace used for optimization as the search proceeds (Papenmeier et al., 2022; 2023). Although these methods encode structure in different ways, they all reduce the burden of modeling the full high-dimensional objective.

A complementary line of work has revisited whether this reduction is always necessary. Hvarfner et al. (Hvarfner et al., 2024) show that standard full-dimensional BO can work surprisingly well when GP lengthscale priors are scaled appropriately with dimension. Xu et al. (Xu et al., 2025) further show that suitable kernel choices and lengthscale initialization can prevent degenerate GP fitting and make standard ARD GPs competitive in high dimensions. These results motivate retaining the original search space while paying closer attention to how the GP is parameterized.

CLUE-BO adopts this full-dimensional view, but focuses on an information problem that remains after dimension-aware calibration. A global lengthscale scale does not say how sensitivity should differ from one coordinate to another. When hundreds of ARD lengthscales must be inferred from a small number of expensive evaluations, the observations may be insufficient to resolve these differences reliably. CLUE-BO supplements this numerical evidence with task semantics. An LLM maps variable meanings and domain knowledge to prior information about relative dimension-wise sensitivity, expressed as ARD lengthscale-prior ratios around a dimension-aware neutral scale. The resulting semantic prior is not fixed throughout optimization; it is treated as a hypothesis that can be revised or suspended when later predictive evidence no longer supports it.

Table 4: State evolution in the representative Rover reflection example.

Iteration	State/Event	Evidence and decision	Effective allocation
Initial	Semantic monitoring	The LLM assigns shorter lengthscales to both endpoint control points based on task semantics.	$r_{0,1} = 0.5,\ r_{58,59} = 0.5$.
20	Neutral fallback	Neutral-over-semantic evidence reaches $220.39 > 200$; reflection neutralizes the current semantic prior.	$r_j = 1$ for all j.
31	Semantic reactivation	The dormant semantic challenger regains sufficient predictive support and crosses the neutral-monitoring threshold; the previous semantic prior is reactivated.	$r_{0,1} = 0.5,\ r_{58,59} = 0.5$.
37	Reflection	Neutral-over-semantic evidence reaches $1240.32 > 1200$. The LLM retains the start override but removes the goal override.	Candidate: $r_{0,1} = 0.5,\ r_{58,59} = 1.0$.
38	Candidate accepted	The candidate acceptance evidence reaches $2.40 \times 10^5 > 600$, so the reflected candidate is promoted to active use.	$r_{0,1} = 0.5,\ r_{58,59} = 1.0$.

F.2 LLM-Guided Black-Box Optimization

LLMs have recently been used at several points in the black-box optimization pipeline. LLAMBO (Liu et al., 2024) uses language models for warm-starting, surrogate modeling, and candidate generation. LLM Processes (LLMP) (Requeima et al., 2024) elicit numerical predictive distributions directly from an LLM and use them for regression and optimization. Other work assigns the model a more strategic or structural role: FunBO (Aglietti et al., 2025) uses LLM-driven program search to discover acquisition functions, CAKE (Suwandi et al., 2025) uses LLMs to evolve GP kernel structures, and LMABO (Ngo et al., 2026) uses an LLM to select acquisition functions online.

The published experiments of these methods have mostly been carried out in low- to moderate-dimensional regimes relative to the setting studied here. LLMP demonstrates black-box optimization on functions with only a few input dimensions. LLAMBO includes a 15-dimensional synthetic setting, while the controller-tuning problems considered by CAKE are 12- and 14-dimensional. More recent work such as LMABO extends synthetic experiments to tens of dimensions. CLUE-BO instead studies problems ranging from 30 to 1,003 dimensions, including settings in which hundreds of independently parameterized coordinates must be modeled from a limited evaluation budget.

This change in dimensionality affects how an LLM can be used effectively. The issue is not simply that a candidate vector becomes longer. For methods that repeatedly condition an LLM on optimization history, every new evaluation may introduce another high-dimensional configuration and its outcome into the context. As optimization proceeds, the amount of numerical state to represent can therefore grow with both dimension and evaluation count, even though the statistical evidence available for any individual coordinate remains sparse. Asking the LLM to reason directly over candidates or numerical histories then places an increasingly difficult inference problem inside the language model itself.

CLUE-BO uses a different interface. The LLM operates at the level of *structural prior reasoning* rather than numerical search. It is called once to construct the initial semantic prior and is subsequently invoked only when accumulated predictive evidence triggers reflection. At reflection time, the optimization history is summarized through structured signals such as predictive evidence, optimization progress, the current prior allocation, and dimension-wise ARD displacement, rather than exposing the full sequence of high-dimensional query vectors. The LLM then proposes a compact hypothesis about relative dimension-wise sensitivity, which is encoded through ARD lengthscale priors; the actual acquisition optimization remains entirely within BO.

This separation is also useful because semantic reasoning is not guaranteed to match the unknown objective. CLUE-BO therefore separates proposing a prior from deciding whether that prior should control the search. An LLM-derived hypothesis enters through a probabilistic prior whose predictive consequences can be evaluated on subsequent BO observations. Accumulated counter-evidence may trigger another revision, while a newly proposed non-neutral prior must receive prospective predictive support before it is activated. The LLM thus contributes structural knowledge without being treated as an unquestioned optimizer.

F.3 Sequential Predictive Model Assessment

The evidence mechanism in CLUE-BO is related to sequential predictive model assessment. The prequential principle evaluates a probabilistic model through predictions made before the corresponding outcomes are observed (Dawid, 1984). Proper scoring rules, including the logarithmic score, provide a principled basis for comparing such probabilistic forecasts (Gneiting & Raftery, 2007). E-values and e-processes provide tools for accumulating evidence under sequential monitoring (Ramdas et al., 2023), with related work considering sequential forecaster comparison (Choe & Ramdas, 2024) and sequential change detection (Shin et al., 2024).

CLUE-BO uses these ideas to compare the predictive consequences of different ARD priors. At each BO iteration, competing GPs are fitted to the same accumulated observations but use different prior structures. Their one-step-ahead predictive densities are fixed before the newly queried objective value is observed, after which the predictive log ratio contributes to the sequential evidence statistic. Under the corresponding model-based null, the likelihood-ratio mixture used by CLUE-BO admits an e-process interpretation. In the optimization algorithm, we use it as a measure of relative predictive evidence without assuming that either fitted GP is the true data-generating model.

The purpose of this comparison is not merely to choose between two fixed predictors. In CLUE-BO, predictive evidence regulates a semantic hypothesis that can itself change during optimization. It determines when the current prior warrants reconsideration and whether a newly proposed prior has received enough prospective support to influence acquisition. In this sense, sequential predictive assessment provides the interface between semantic reasoning and BO: the LLM proposes a possible structure for the surrogate, while subsequent observations determine how much influence that structure should retain.