GOLLUM: GAUSSIAN PROCESS OPTIMIZED LLMS – REFRAMING LLM FINETUNING THROUGH BAYESIAN OPTIMIZATION

Bojana Ranković[1,2], **Philippe Schwaller**[1,2]
[1]École Polytechnique Fédérale de Lausanne (EPFL)
[2]National Centre of Competence in Research (NCCR) Catalysis
{bojana.rankovic,philippe.schwaller}@epfl.ch

ABSTRACT

Large Language Models (LLMs) can encode complex relationships in their latent spaces, yet harnessing them for optimization under uncertainty remains challenging. We address this gap with a novel architecture that reframes LLM finetuning as Gaussian process (GP) marginal likelihood optimization via deep kernel methods. We introduce LLM-based deep kernels, jointly optimized with GPs to preserve the benefits of both – LLMs to provide a rich and flexible input space for Bayesian optimization and – GPs to model this space with predictive uncertainty for more efficient sampling. Applied to Buchwald-Hartwig reaction optimization, our method nearly doubles the discovery rate of high-performing reactions compared to static LLM embeddings (from 24% to 43% coverage of the top 5% reactions in just 50 optimization iterations). We also observe a 14% improvement over domain-specific representations without requiring specialized features. Extensive empirical evaluation across 19 benchmarks – ranging from general chemistry to reaction and molecular property optimization – demonstrates our method's robustness, generality, and consistent improvements across: (1) tasks, (2) LLM architectures (encoder, decoder, encoder-decoder), (3) pretraining domains (chemistry-related or general-purpose) and (4) hyperparameter settings (tuned once on a single dataset). Finally, we explain these improvements: joint LLM-GP optimization through marginal likelihood implicitly performs contrastive learning, aligning representations to produce (1) better-structured embedding spaces, (2) improved uncertainty calibration, and (3) more efficient sampling – without requiring any external loss. This work provides both practical advances in sample-efficient optimization and insights into what makes effective Bayesian optimization.

1 INTRODUCTION

Large Language Models (LLMs) have demonstrated remarkable capabilities in natural language understanding and generation (Vaswani et al., 2017; Devlin et al., 2018; Radford et al., 2018; Wei et al., 2022). Their success stems from an ability to learn rich representations of text that capture subtle patterns, relationships, and domain-specific knowledge (Petroni et al., 2019; Brown et al., 2020). This representational power has naturally led to growing interest in adapting LLMs beyond general language tasks to specialized domains – from scientific discovery to reasoning tasks (White, 2023; Boiko et al., 2023; M. Bran et al., 2024; Hayes et al., 2025; Guo et al., 2025). However, despite their expressive capabilities, LLMs exhibit fundamental limitations in reliability. Even in their primary domain of text generation, they can produce overconfident yet factually incorrect outputs through hallucination (Maynez et al., 2020; Peng et al., 2023; Huang et al., 2025). In high-stakes fields like drug discovery, materials design, or automated reasoning, such failures carry critical real-world risks, making principled uncertainty quantification essential (Schweidtmann et al., 2018; Tavazza et al., 2021; Mervin et al., 2021; Pomberger et al., 2022; Müller et al., 2022; Torres et al., 2022; Hickman et al., 2022).

In this context, Bayesian optimization (BO) (Kushner, 1962; 1964; Garnett, 2023) has emerged as a powerful strategy for optimizing expensive-to-evaluate functions by efficiently balancing explo-

ration and exploitation (Eyke et al., 2020; Shields et al., 2021; Felton et al., 2021; Häse et al., 2021a; Guo et al., 2023). BO typically employs Gaussian Processes (Williams & Rasmussen, 2006) (GPs) due to their principled uncertainty estimates, interpretable confidence bounds and well-calibrated predictions even in low-data regimes. This property of GPs is particularly valuable in domains that demand sample-efficient optimization of complex (e.g., analytically intractable) objectives – as is often the case in chemistry. Recent works have begun exploring LLMs for Bayesian optimization, either as feature extractors for surrogate models (Ranković & Schwaller, 2023) or through sequential LLM finetuning with post-hoc uncertainty quantification (Kristiadi et al., 2024). While promising, these methods either underutilize LLMs' adaptation capabilities or decouple predictive performance from uncertainty estimation, limiting their optimization effectiveness.

We introduce GOLLuM (Gaussian Process Optimized LLMs), a framework that seamlessly integrates LLMs into the GP architecture through Deep Kernel Learning (DKL) (Wilson & Nickisch, 2015; Wilson et al., 2016). Rather than using LLMs as sophisticated encoding tools (via embeddings) or lookup-tables (via prompting), our approach directly employs the GP marginal likelihood as LLM finetuning objective. The resulting method provides a bidirectional feedback loop: the GP guides updates to LLM weights to produce more effective embeddings, which in turn enhance the GP's probabilistic modeling. Through GP marginal likelihood optimization, the LLM embeddings adapt to follow a key principle: points with similar function values should be close in the embedding space, aligning with the similarity structure imposed by the GP kernel. This strategy inherently induces a contrastive learning effect (Kaya & Bilge, 2019) enabling the model to naturally organize the design space points into distinct regions – the good (high objective function values), the bad (low objectives), and the ugly [1]. The clear separation in the latent space improves the optimization performance by enabling more effective exploration while the joint training helps maintain well-calibrated uncertainty estimates, despite potential kernel misspecifications.

Our framework is agnostic to the specific LLM architecture or its pretraining, effectively adapting any general-purpose LLM into a powerful optimization tool with rigorous uncertainty quantification. We evaluate our approach within the chemistry domain, where expensive evaluations and vast design spaces pose crucial bottlenecks to progress in drug discovery and materials science (Guo et al., 2023; Taylor et al., 2023). In this context, our method's ability to efficiently navigate complex optimization spaces with principled uncertainty guidance has the potential to accelerate scientific discovery.

Our key contributions include:

1. **GOLLuM: The first jointly trained LLM-based deep kernel GP architecture.** We introduce the first end-to-end framework that integrates LLM capabilities with Gaussian processes via marginal likelihood, enabling principled Bayesian optimization. We demonstrate and formalize how GP marginal likelihood optimization induces contrastive structure in the embedding space – separating regions by performance – without any explicit contrastive loss. This unified approach provides an alternative to domain-specialized models or representations while producing interpretable latent space structures.

2. **Empirical insights into representation factors that enable successful high-dimensional BO.** Through systematic analysis of 14 fixed LLM and chemistry-specific representations, we quantify how representation structure influences BO success. We find that optimization success strongly correlates ($r = 0.92$) with a normalized smoothness metric capturing the alignment between the GP's inductive bias and the structure of the representation space. Representations that support smooth yet calibrated surrogate fits enable more principled exploration and lead to better BO outcomes.

3. **Robust generalization across diverse chemical tasks.** Our approach demonstrates consistently strong performance across 19 diverse chemistry benchmarks, using hyperparameters tuned on a single dataset. Compared to chemistry-specific representations, fixed LLM embeddings, and disjoint finetuning approaches, our method achieves superior exploration of chemical spaces, better sample efficiency, and generalization. This conclusion is supported by more than 8,000 experiments.

[1] Reference to the 1966 film The Good, the Bad and the Ugly, directed by Sergio Leone.

2 METHODS

2.1 BAYESIAN OPTIMIZATION OVERVIEW

Bayesian optimization is a sample-efficient method for optimizing black-box functions, potentially expensive to evaluate – a setting common in chemistry, where each experiment incurs substantial costs. BO works by training a probabilistic surrogate model (typically a GP) on previously observed data and using it to select new, informative queries via an acquisition function. We provide a detailed technical overview in Appendix B. The effectiveness of BO depends on the choice of representations and the quality of the surrogate model – both of which we improve in this work.

2.2 DATA REPRESENTATION

BO performance in chemistry is highly sensitive to the choice of data representation, especially given the heterogeneous data types (e.g., categorical reagents, numeric conditions, molecular structures), combinatorial design spaces, and variable numbers of parameters involved – making representation a critical challenge (Ranković et al., 2024). Natural language offers a flexible medium for expressing such optimization problems as textual descriptions, while LLMs can transform these inputs – regardless of type – into unified continuous embeddings. We construct these embeddings through a two-step process:

1. Template Construction: We define each task t as a standardized template: $t = \text{template}(\{\textit{parameters}, \textit{values}\})$ where values define the actual conditions of the problem (e.g., reagents used in a chemical synthesis). For single-variable tasks (e.g., molecular optimization), the template reduces to a single textual identifier such as a molecular SMILES string (Anderson et al., 1987; Weininger, 1988). This approach provides a consistent format applicable across a wide range of optimization problems.

2. LLM Embedding: We process the templated description through LLMs to obtain a fixed-dimensional embedding: $\mathbf{x} = \text{LLM}(t) \in \mathbb{R}^d$. This embedding unifies heterogeneous parameter types and enables compatibility with standard continuous kernels (e.g., Matérn), while preserving inter-parameter relationships and scaling to variable-length inputs.

The resulting embedding vector $\mathbf{x}$ captures both the individual parameter values and their interactions, providing a unified representation for subsequent GP modeling. This approach circumvents the need for designing specialized kernels, as the LLM embedding space naturally encodes meaningful distances – enabling optimization over mixed categorical and numerical inputs within a continuous space. Moreover, it generalizes to tasks with arbitrary combinations of categorical, numerical or structural parameters – making it broadly applicable beyond chemistry to domains where design spaces can be expressed through text.

2.3 GAUSSIAN PROCESS WITH FIXED LLM EMBEDDINGS

LLM embeddings can be directly used as input vectors to GPs, which model the output based on observed data. In this setup, the embeddings remain fixed throughout the optimization process – following the approach outlined in BoChemian (Ranković & Schwaller, 2023) – and the model's adaptability comes solely from learning the GP hyperparameters θ. We use a GP prior with a Matérn-5/2 kernel with trainable hyperparameters $\theta = \{\ell, \sigma^2, \sigma_n^2, c\}$ representing the lengthscale, signal variance, observation noise variance, and constant mean. This approach relies entirely on the pretrained LLM's embedding space to define input structure – specifically, the relative positioning of points based on their underlying features. The GPs with stationary kernels (such as Matérn-5/2) assume this structure reflects meaningful relationships: points close together in the embedding space are expected to have similar outcomes. However, general pretrained LLMs may not reflect chemical similarities and their representations may not encode the right inductive biases for the task. As a result, the GP can struggle to model the objective effectively in the fixed-feature setting, unless the embedding space already captures relevant patterns. This limitation can be addressed through deep kernel methods, which we describe next.

2.4 DEEP KERNEL GAUSSIAN PROCESS

Deep kernel Gaussian processes combine the flexibility of deep neural networks with the principled uncertainty quantification of Gaussian processes. In this approach, the kernel function is composed with a learned feature transformation:

$$k_{\theta,\,\phi}(\mathbf{x}, \mathbf{x}') = k_\theta(g_\phi(\mathbf{x}), g_\phi(\mathbf{x}')),$$

where g_ϕ is a parameterized feature extractor with parameters ϕ. This composition allows the model to learn task-specific feature representations while maintaining the probabilistic properties of the GP framework. The learned transformation and the GP parameters are jointly optimized through the marginal likelihood where $\mathbf{K}_{\theta,\phi}$ is the kernel matrix computed using the transformed features.

2.5 LLM-BASED DEEP KERNEL

In our framework, we explore different approaches to constructing the feature transformation $g_\phi(\cdot)$.

1. Projection Layer: A learned transformation consisting of a linear projection $\mathbf{P} \in \mathbb{R}^{m \times d}$ followed by a non-linear activation function (ELU), applied to fixed LLM embeddings: $g_\phi(\mathbf{x}) = \mathbf{P}\text{LLM}(t)$ where m is the projection dimension. This setup closely follows standard deep kernel learning with a trainable transformation applied on top of fixed features before kernel evaluation. It is particularly useful in settings where LLM weights cannot be accessed, as in the case of closed-source models from OpenAI. The projection layer learns to emphasize or suppress different aspects of fixed LLM embeddings, effectively creating a task-specific representation.

2. PEFT-Adapted LLM: Low-rank adaptation of LLM parameters: $g_\phi(\mathbf{x}) = \text{LLM}\phi(t)$ where ϕ represents the trainable adapter parameters. Parameter efficient finetuning (PEFT) (Houlsby et al., 2019; Hu et al., 2022; Li et al., 2023a) addresses the challenge of adapting large language models by updating a smaller (often several orders of magnitude fewer) number of parameters, typically inserted into or alongside the LLM architecture. We employ Low-Rank Adaptation (LoRA) (Hu et al., 2022) to preserve potential chemical knowledge captured during pretraining and learn task-specific adaptations, while avoiding catastrophic forgetting or compromising general capabilities.

3. Combined Approach: Sequential application of LoRA and projection: $g_\phi(\mathbf{x}) = \mathbf{P}\text{LLM}\phi(t)$, thus combining the benefits of both worlds. The LoRA adapters allow the LLM to adapt its internal representations to the optimization task, while the projection layer provides an additional degree of freedom to reshape the embedding space.

With any of these methods, we optimize the parameters ϕ (projection matrix and/or LoRA parameters) jointly with the GP hyperparameters through the marginal likelihood. In other words, we are finetuning the LLM through the GP loss which allows the model to learn transformations that both preserve relevant chemical information, organize the latent space to better reflect the structure of the optimization objective, and provide well-calibrated uncertainty measures.

2.6 LLM FINETUNING AS GP MARGINAL LIKELIHOOD OPTIMIZATION

Let $\mathcal{L}(\theta, \phi)$ denote the GP marginal likelihood of observing targets $\mathbf{y}$ given inputs $\mathbf{X}$, LLM parameters ϕ, and GP hyperparameters θ:

$$\mathcal{L}(\theta, \phi) = \log p(\mathbf{y}|\mathbf{X}, \theta, \phi) = -\frac{1}{2}(\mathbf{y}^\top \mathbf{K}_{\theta,\phi}^{-1}\mathbf{y} + \log|\mathbf{K}_{\theta,\phi}| + n\log 2\pi) \tag{1}$$

To optimize the embedding parameters ϕ jointly with the GP hyperparameters θ, we maximize the marginal likelihood using gradient-based optimization:

$$\theta^*, \phi^* = \arg\max_{\theta,\phi} \mathcal{L}(\theta, \phi). \tag{2}$$

We compute the gradients of the marginal likelihood with respect to the parameters via standard backpropagation:

$$\nabla_{\theta,\phi}\mathcal{L}(\theta, \phi) = \frac{1}{2}\mathbf{y}^\top \mathbf{K}_{\theta,\phi}^{-1}\left(\nabla_{\theta,\phi}\mathbf{K}_{\theta,\phi}\right)\mathbf{K}_{\theta,\phi}^{-1}\mathbf{y} - \frac{1}{2}\text{Tr}\left(\mathbf{K}_{\theta,\phi}^{-1}\nabla_{\theta,\phi}\mathbf{K}_{\theta,\phi}\right). \tag{3}$$

We perform the joint optimization with separate learning rates for embedding parameters (ϕ) and GP hyperparameters (θ) to encourage stable convergence and avoid overfitting of either component.

2.7 Implicit Metric Learning

The explicit feature of GPs to evaluate the similarities in the output based on the distances in the input space creates a contrastive learning effect for LLM embeddings. This beneficial consequence arises from the two-fold utilization of the GP marginal likelihood. The kernel function $k_{\theta,\phi}(\mathbf{x}, \mathbf{x}') = k_\theta(g_\phi(\mathbf{x}), g_\phi(\mathbf{x}'))$ measures similarity between points, and optimizing the marginal likelihood (Eq. 1) encourages embedding distances to decrease between points with similar outputs and increase between points with dissimilar outputs. The contrastive learning effect comes directly from the GP marginal likelihood optimization. For a kernel based on distances (like Matérn) we can rewrite the term $\mathbf{y}^\top \mathbf{K}_{\theta,\phi}^{-1}\mathbf{y}$ as a weighted sum of pairwise interactions (with weights w_{ij} defined by the inverse kernel matrix) inducing implicit contrastive learning objective $\mathcal{L}_{\text{implicit}}$:

$$\mathcal{L}_{\text{implicit}}(\theta, \phi) \propto \sum_{i,j} w_{ij} \cdot \|g_\phi(\mathbf{x}_i) - g_\phi(\mathbf{x}_j)\|_2, \quad \begin{cases} \|g_\phi(\mathbf{x}_i) - g_\phi(\mathbf{x}_j)\|_2 \downarrow & \text{if } \|y_i - y_j\| \text{ is small} \\ \|g_\phi(\mathbf{x}_i) - g_\phi(\mathbf{x}_j)\|_2 \uparrow & \text{if } \|y_i - y_j\| \text{ is large} \end{cases} \tag{4}$$

In other words, the joint GP optimization induces high kernel values (small distances) between points with similar outputs and low kernel values (large distances) between points with different outputs, therefore separating the embedding space into distinct categories. This reorganization in the latent space happens automatically through the optimization of the deep kernel parameters, adapting the feature space to better align with outcomes without requiring explicit contrastive loss terms.

3 Results & Discussion

3.1 Bayesian Optimization with Fixed LLM Features

Building on top of BoChemian (Ranković & Schwaller, 2023), we first evaluate the effectiveness of LLM embeddings as fixed feature extractors for Bayesian optimization of Buchwald-Hartwig (BH) reactions. Our objective is to optimize the yield of this chemical reaction. The parameters include reaction compounds – 15 reactants, 22 additives, 3 bases, and 4 ligands – totaling a design space of 3955 evaluated reactions, split across five distinct products. We use a variety of publicly available LLMs selected through their base architecture: **Encoder-based** — ModernBERT (Warner et al., 2024), UAE (Li & Li, 2023), MXBAI (Lee et al., 2024); **Encoder–Decoder** — Instructor (Su et al., 2022), T5 (Raffel et al., 2020) and its chemistry-related variant T5Chem (Christofidellis et al., 2023); **Decoder–only** — Llama series (Grattafiori et al., 2024; BehnamGhader et al., 2024), Mistral series (BehnamGhader et al., 2024), Qwen series (Bai et al., 2023; Li et al., 2023b), and OpenAI embeddings (OpenAI, 2024).

In Figure 1 we show the performance of all LLM-based representations alongside chemistry-specific baselines: DRFP(Probst et al., 2022) – a reaction fingerprint, and T5Chem-SMILES(Christofidellis et al., 2023) – a domain-specialized LLM leveraging reaction SMILES inputs, aligned with its pretraining. All other representations, including T5Chem, employ templated textual procedures, as described previously.

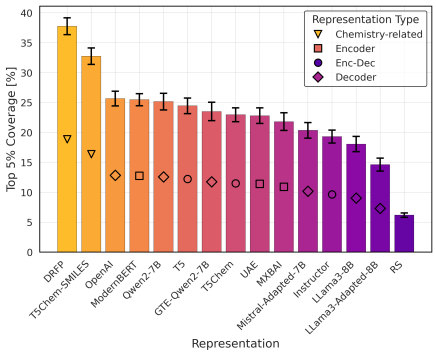

Figure 1: **BO performance with fixed LLM features as input to GP.** Average discovery of high-impact regions of the design space. We show percentage of the top 5% reactions found during the optimization, across all five Buchwald-Hartwig reactions. Domain-specific representations include T5Chem-SMILES(Christofidellis et al., 2023), a pretrained chemistry-related LLM with SMILES input, and DFRP(Probst et al., 2022), a reaction fingerprint.

We use the top 5% coverage metric following 50 BO iterations to evaluate the success of BO in uncovering entire regions of high-valued reactions rather than a single optimum. The motivation is two-fold: (1) identifying a single top-performing reaction is of limited practical value, as the yield difference between the absolute best and other high-performing reactions may be negligible; (2) discovering entire regions of successful reaction conditions enables chemists to select reactions that satisfy additional practical constraints such as cost (Schoepfer et al., 2024), environmental impact (Anastas & Eghbali, 2010) or availability of reagents (Griffiths & Hernández-Lobato, 2020). While the chemistry-specialized baselines excel at discovering high-yield reactions, they require SMILES notation as input, limiting their ability to represent diverse reaction conditions beyond molecular structures.

3.2 What makes a good representation for Bayesian optimization?

We observe substantial variation in BO performance across LLM embeddings, prompting analysis of the underlying factors. T5Chem only performs well with inputs resembling its pretraining data (e.g., reaction SMILES), highlighting the limited generality of domain-specialized LLMs (Kristiadi et al., 2024) and the critical role of representation choice (Ranković et al., 2024). This observation suggests that input representation influences LLM-based BO through two mechanisms: (1) contextual alignment with pretraining helps models better leverage their learned weights, and (2) the resulting embedding structure affects how well the GP can model the objective under a fixed kernel.

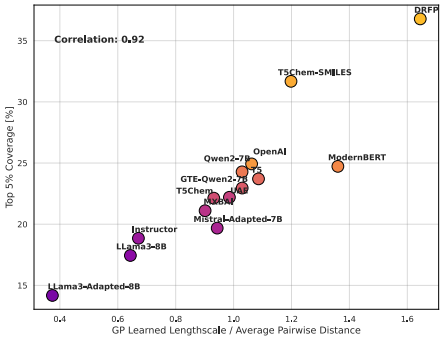

Figure 2: **Data representations and their success rates in BO.** BO performance correlates with GP smoothness, measured as the ratio of learned lengthscale to average pairwise embedding distance.

To investigate the second effect, we examine how the embedding space structure interacts with the GP's inductive bias. Specifically, we compute the ratio between the GP's learned lengthscale and the average pairwise distance between points in the embedding space. This normalized smoothness ratio reflects how far the GP is willing to generalize relative to the scale of the data distribution. As shown in Figure 2 this metric correlates strongly with BO performance ($r = 0.92$). A higher ratio indicates that the GP can model the objective with a smoother fit (see Figure 10 in the Appendix), as the embedding space provides a coherent structure that aligns with the kernel's assumptions. This alignment allows the GP to generalize across broader regions while still resolving performance differences, leading to more effective acquisition decisions. Our analysis complements Papenmeier et al. (2025), who argue that longer lengthscales only help when the objective varies slowly enough to be captured by a smooth surrogate. We extend this insight to the representation level, showing that representations inducing such smoothness naturally – without priors or initialization tricks – enable more effective optimization. One might alternatively expect that better GP fit, particularly in high-performing regions, is the primary driver of BO success. However, as shown in Appendix E.3, standard and weighted R^2 correlate less strongly with performance. While predictive accuracy helps, modeling the objective function and efficiently discovering its optimum are distinct challenges. Our results show that alignment between the representation space and the GP's inductive bias more directly enables principled optimization.

3.3 Bayesian Optimization with LLM-based Deep Kernels

These findings motivate moving beyond fixed embeddings toward joint optimization of both the representation and surrogate model. To address the misalignment between embedding spaces and GP inductive biases, we explore three variants of our LLM-based deep kernel architecture:

1. PLLM: A projection layer applied to fixed LLM embeddings, allowing task-specific transformations with frozen LLM weights.

2. LLMϕ: Direct adaptation of the LLM through parameter-efficient fine-tuning (LoRA), modifying the internal representation without an additional projection.

3. PLLMϕ: A combined approach that leverages both LoRA adaptation and a projection layer, providing maximum flexibility in representation learning.

In each variant, the deep kernel LLM-GP optimization through shared marginal likelihood enables the representation space to dynamically align with the GP's assumptions. We now present results supporting this alignment.

GP optimized LLMs as part of the deep kernel GP architecture lead to substantial improvements in performance, increasing the discovery rate of high-performing reactions. Our method consistently outperforms static embeddings across all LLM architectures (Figure 3). PLLMϕ adaptation of base T5 model now achieves the highest coverage at 42.6%, representing a 74% relative improvement over the fixed-LLM features using the same model (24.45%). Moreover, we observe a 14% increase over the previous best representation (DRFP 37.75%) while not requiring constrained input to reaction SMILES or any domain-specific parametrization. General-purpose LLMs with GP-guided finetuning also surpass chemistry-specialized pretrained LLMs such as T5Chem-SMILES, while using general input through procedural texts. In contrast to previous work (Kristiadi et al., 2024), these results show that promising BO results can be achieved without requiring domain-specialized models or inputs aligned to their pretraining data. Our model can adapt the input embeddings on-the-fly allowing for increased flexibility. The improvements of GP-finetuned LLMs are also consistent across encoder-only (ModernBERT), encoder-decoder (T5), and decoder-only (Qwen2-7B) architectures. These results validate our earlier analysis on the importance of the alignment between GP's inductive bias and embedding space structure. In the following section, we analyze how we achieve such alignment.

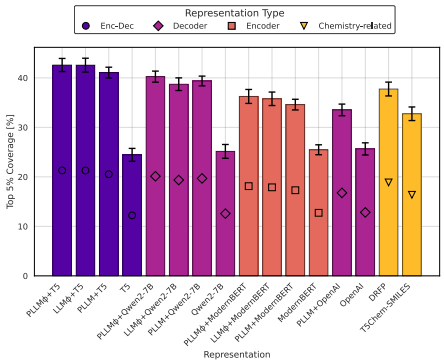

Figure 3: **Comparative analysis of GP-based LLM finetuning.** The finetuned models are arranged by the overall performance and relative improvements to their base (fixed embeddings) LLM-GP variants. Chemistry baselines (previous best) included for comparison.

3.4 Implicit Contrastive Learning and Chemical Interpretability in the Latent Space

The evolution of the embedding space (Figure 4 A and B) illustrates how the GP's marginal log-likelihood objective adapts the LLM's representations during finetuning. Initially, high and low-performing points are mixed in the embedding space. These unstructured representations would typically induce a non-smooth GP fit, as the objective function varies quickly even across nearby points. As optimization progresses, however, the space gradually reorganizes into clearer performance regions (Figure 4 B), achieving a contrastive learning effect through distance-sensitive GP marginal likelihood. The pairwise distance distributions reflect this process – initially overlapping across high-high, high-low, and low-low regions, they gradually separate. This separation is both mathematical and semantical, reflecting the model's ability to learn chemical relationships that improve optimization (Figure 4 D).

The joint adaptation – with updated LLM embeddings and smoother GP fit, offers greater flexibility than using frozen features. As the structure forms, high-yield reactions become clustered, enabling the GP to guide the acquisition function toward promising regions. Variance estimates remain well-calibrated across the space (visualized with point sizes in Figure 4B), supporting reliable exploration.

Moreover, the learned structure supports interpretability. For instance, in Figure 4D, high-yield iodide-based reactions are consistently separated from lower-yield chloride-based ones. This separation reveals meaningful chemical patterns learned by the model, and highlights how the embedding space captures domain-relevant knowledge that can aid downstream decision-making.

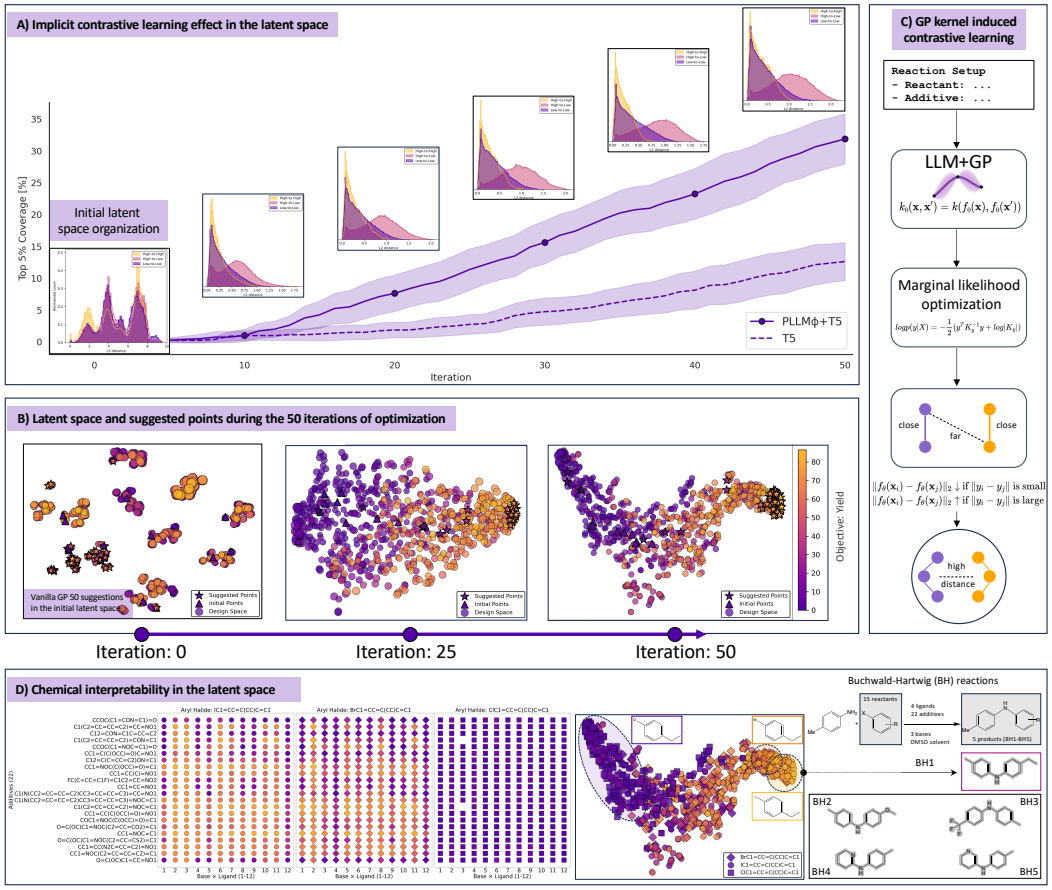

Figure 4: **Implicit contrastive learning effects with LLM-based deep kernel GPs.** A) We observe the progression of the embeddings through pairwise distance histograms of points in separate output classified regions (high yield reactions, low yield reactions, in between). B) Additionally we visualize the new latent space during the optimization procedure in the first, 25th, and last iterations. C) General approach to implicit contrastive learning with GP+LLM architecture. D) Chemical interpretability of the learned latent space. The first three panels show the distribution of reaction outcomes for different reactants (I, Br, Cl aryl halides) across the whole design space (including base-ligand and additive conditions) with colors indicating reaction performance (yield). Following is the projection of the latent space, where we observe reactions clustering based on their aryl halide identity, suggesting that the model captures meaningful chemical relationships. The rightmost section represents the chemical design space of Buchwald-Hartwig reactions used in experiments.

3.5 Additional Benchmarks and Uncertainty Calibration

Finally, we evaluate our model on a variety of optimization tasks in chemistry, from reaction and molecular optimization to the optimization of chemical processes. We provide additional details on the datasets in the Appendix D. All experiments in this and previous sections were run with 10 initial points (randomly selected from the lower median) and 50 BO iterations of batch size 1 (Appendix F).

We compare our results to several related benchmarks: (1) Standard GP with domain-specific representations (DRFP for reactions, molecular fingerprints for property optimization, and vectorized parameters where neither of these are available, as in the case of general chemistry processes optimization tasks). (2) BoChemian (Ranković & Schwaller, 2023) which uses fixed LLM embeddings as input to a standard GP. (3) LAPEFT (Kristiadi et al., 2024) which updates the LLM through supervised finetuning with MSE loss and employs Laplace approximation (LA) for probabilistic modeling. A comprehensive visualization of the different methods is available in Figure 5 A. Hav-

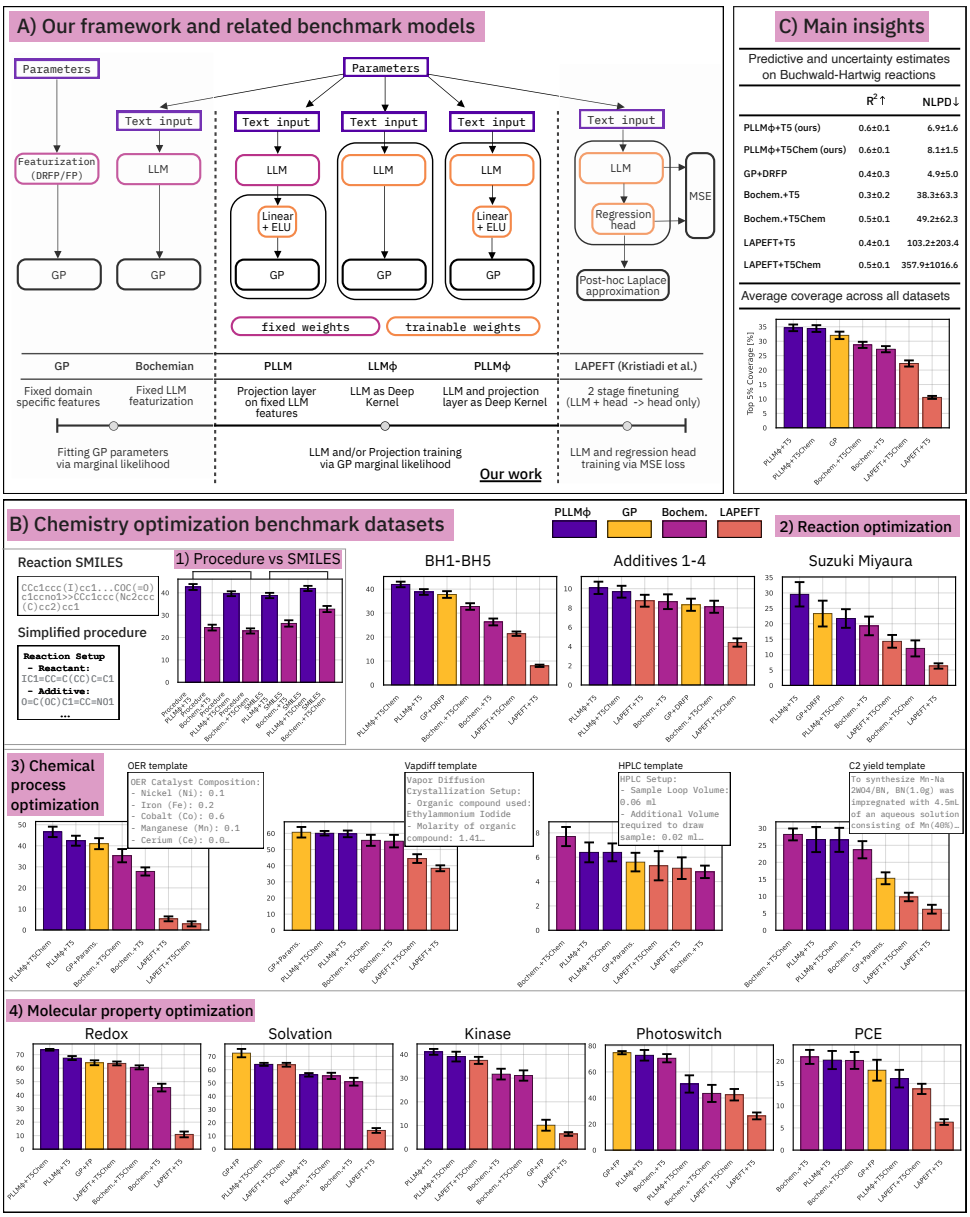

Figure 5: **Benchmarking on various chemistry-related optimization tasks with comparisons to related approaches.** A) Visualization of our proposed architectures alongside available previous works (BoChemian (Ranković & Schwaller, 2023), LAPEFT (Kristiadi et al., 2024)) and standard GP baseline. B) Optimization results on 19 chemistry-related optimization tasks – 2) reaction optimization (5 Buchwald Hartwig reactions, 4 additive screening reactions, Suzuki Miyaura cross-coupling); 4) general chemistry benchmarks (high-performance liquid chromatography – HPLC setup, oxygen evolution reaction catalysts – OER, vapor diffusion crystallization – Vapdiff, C2 yield optimization (Ramos et al., 2023)) and 4) molecular property optimization. To ensure a fair comparison to LAPEFT we add their best base model (T5Chem with SMILES input) and fix the textual representation to SMILES where possible (BH1-BH5, Additives 1-4, Suzuki-Miyaura, Molecular optimization). However, we note our model's robustness to a variety of textual input and compare different textual representation of BH reactions (reaction SMILES or textual procedure in B1). For datasets where reaction SMILES are not available (C2 yield, HPLC setup, OER, Vapdiff) we show an example of applied template. C) Predictive (R^2) and uncertainty estimates (negative log predictive density – NLPD) on BH reactions (60 training points, 20 repeats) together with average (across all tasks) optimization performance (top 5% coverage) against benchmarked models.

ing previously shown PLLMϕ outperforming other variants (PLLM, LLMϕ) we select this approach for all subsequent benchmarking. To ensure a fair comparison to LAPEFT, which states T5Chem with SMILES as the best-performing model, we additionally include this model next to its base variant T5.

Motivated by the insights outlined in (Kristiadi et al., 2024) we first analyze how the different textual representations influence our model's performance. For this analysis we compare the results on inputs represented through reaction SMILES or template procedure on BH1-BH5 reactions. We do observe a slight preference of general T5 model to general procedure text and a similar improvement of T5Chem with the structured reaction SMILES compared to a more general input (simplified procedure – Figure 5 B1). However, these differences are still negligible compared to the improvement over fixed features (BoChemian) demonstrating the robustness of our approach to different textual representations (SMILES vs procedure) and LLM pretraining (T5 vs T5Chem). Our method aligns the input through the joint LLM-GP optimization resulting in both adaptive LLM weights – removing the need for domain-pretrained models, and adaptive representations – removing the dependency on the pretraining data format.

Averaged across all benchmark datasets, our approach yields superior performance compared to all baseline models: (1) standard GP optimization in parameter space with domain-specific representations, (2) fixed LLM features (BoChemian (Ranković & Schwaller, 2023)) and (3) Bayesian neural network (BNN) surrogates with decoupled supervised finetuning (LAPEFT(Kristiadi et al., 2024)). These improvements are particularly evident in reaction optimization tasks, where our method consistently outperforms across all reaction types (Buchwald-Hartwig, Suzuki-Miyaura, and additive screening). These tasks typically involve complex combinatorial spaces, which highlights our method's ability to effectively model structured chemical domains. Moreover, by representing the data through text, we eliminate the need for specialized featurization techniques, reducing reliance on expert-designed features that may be costly to compute. Unlike standard approaches that require careful handling of mixed parameter spaces – deciding between categorical, continuous or domain-engineered features – our method integrates all information in a unified, flexible representation. Compared to the previous approach with fixed LLM features we observe an overall 23% increase in covering the high-output regions demonstrating that joint GP-LLM optimization results in a better strategy. We are also able to select more than double high-performing design set points (114% more) compared to LAPEFT (Kristiadi et al., 2024). This substantial improvement suggests that the approach of supervised finetuning combined with post-hoc BNNs is actually detrimental to performance, as we observe fixed features with T5 outperforming LAPEFT-T5 in 85% of benchmarks. On the other hand, LAPEFT tends to leverage domain-specialized models (pretrained T5Chem) showing better performance between the two in almost all benchmarks. Our model, however, outperforms both the LAPEFT-T5 and LAPEFT-T5Chem in all optimization tasks while not requiring pretrained domain-specific models. This result demonstrates the power of LLM finetuning through GP marginal likelihood, as we are able to transform any general LLM to a domain-specific optimizer without task-related pretraining. We support this claim with a comparable performance between PLLMϕ method on both T5Chem and its base T5 version in almost all benchmarks while only using domain-specific textual representation (SMILES). Moreover, our method performs well across all tasks (reaction, molecular, and process optimization) ranking first in 50% of benchmarks or second in the remaining 50%. In optimization problems where we rank second, the gap can often be attributed to a strong alignment between the task and domain-specific features. For example, molecular fingerprints perform well for solvation energy prediction, where structure-based encodings may better suit the objective, while fixed embeddings from pretrained chemistry models (T5Chem) outperform on HPLC and C2 yield – likely due to prior knowledge already embedded in the model. Nonetheless, our method remains consistently competitive across all problems, demonstrating its versatility with (1) different LLM models, (2) textual input formats, and (3) a wide range of optimization tasks – all under a single, robust set of hyperparameters.

4 Conclusion

This work presents a novel method that reframes LLM finetuning through Bayesian optimization, demonstrating how joint training with Gaussian processes can substantially improve the utility of LLM embeddings for optimization tasks. By leveraging the GP marginal likelihood optimization alongside the representational power of LLMs, we achieve four key benefits: implicit metric learning

in the embedding space, principled uncertainty quantification, more effective sampling of promising regions, and seamless adaptability across diverse domains and tasks using readily available LLMs.

Our approach directly addresses two fundamental challenges in BO: (1) designing meaningful representations, which we solve through adaptive LLM embeddings that evolve during optimization, and (2) selecting appropriate kernels, which we handle via deep kernel learning that jointly adapts both the representation space and kernel parameters to the specific task.

By merging LLMs as an integrated part of GPs through deep kernel learning, we enable a structured organization in the embedding space, creating representations that better support sample-efficient optimization. This process occurs without explicit contrastive learning objectives, emerging instead from the GP's need to model the objective function under uncertainty with trainable LLM input. The consistent improvement across different LLM architectures suggests we have identified a fundamental principle for adapting general pretrained models to specific optimization tasks.

With GP optimized LLMs we offer a principled alternative to prompt-based methods that currently dominate LLM applications. Rather than relying on closed-source model behavior and instruction heuristics that often lack reproducibility, we align the internal representations of LLMs with the requirements of uncertainty-aware decision-making – a critical need in real-world scientific and engineering applications.

Our results across 19 diverse chemical optimization problems demonstrate practical benefits over (1) GP optimization with domain-specialized features (2) static LLM embeddings and (3) decoupled supervised finetuning approaches. These improvements, combined with maintained uncertainty calibration, suggest promising applications beyond chemistry in domains where sample efficiency is crucial and data collection is expensive.

ACKNOWLEDGMENTS

We would like to acknowledge Ryan-Rhys Griffiths and Joshua Sin for their invaluable insights, feedback, and constructive discussions throughout this study.

This publication was created as part of NCCR Catalysis (grant number 225147), a National Centre of Competence in Research funded by the Swiss National Science Foundation.

CODE AND DATA AVAILABILITY

Code and data available at https://github.com/schwallergroup/gollum.

REFERENCES

Dhruv Agarwal, Manoj Ghuhan Arivazhagan, Rajarshi Das, Sandesh Swamy, Sopan Khosla, and Rashmi Gangadharaiah. Searching for optimal solutions with llms via bayesian optimization. In *The Thirteenth International Conference on Learning Representations*.

Garvit Agarwal, Hieu A. Doan, Lily A. Robertson, Lu Zhang, and Rajeev S. Assary. Discovery of energy storage molecular materials using quantum chemistry-guided multiobjective bayesian optimization. *Chemistry of Materials*, 33(20):8133–8144, October 2021. ISSN 1520-5002. doi: 10.1021/acs.chemmater.1c02040.

Derek T Ahneman, Jesús G Estrada, Shishi Lin, Spencer D Dreher, and Abigail G Doyle. Predicting reaction performance in C–N cross-coupling using machine learning. *Science*, 360(6385):186–190, 2018.

Paul Anastas and Nicolas Eghbali. Green chemistry: principles and practice. *Chemical Society Reviews*, 39(1):301–312, 2010.

Eric Anderson, Gilman D Veith, and David Weininger. *SMILES, a line notation and computerized interpreter for chemical structures*. US Environmental Protection Agency, Environmental Research Laboratory, 1987.

Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3(Nov):397–422, 2002.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

Maximilian Balandat, Brian Karrer, Daniel R. Jiang, Samuel Daulton, Benjamin Letham, Andrew Gordon Wilson, and Eytan Bakshy. BoTorch: A Framework for Efficient Monte-Carlo Bayesian Optimization. In *Advances in Neural Information Processing Systems 33*, 2020. URL http://arxiv.org/abs/1910.06403.

Christoph Bannwarth, Sebastian Ehlert, and Stefan Grimme. GFN2-xTB—An accurate and broadly parametrized self-consistent tight-binding quantum chemical method with multipole electrostatics and density-dependent dispersion contributions. *Journal of Chemical Theory and Computation*, 15(3):1652–1671, 2019.

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. Llm2vec: Large language models are secretly powerful text encoders. *arXiv preprint arXiv:2404.05961*, 2024.

Lukas Biewald. Experiment tracking with weights and biases, 2020. URL https://www.wandb.com/. Software available from wandb.com.

Daniil A Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. Autonomous chemical research with large language models. *Nature*, 624(7992):570–578, 2023.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Alice Capecchi, Daniel Probst, and Jean-Louis Reymond. One molecular fingerprint to rule them all: drugs, biomolecules, and the metabolome. *Journal of cheminformatics*, 12(1):1–15, 2020.

Adrià Cereto-Massagué, María José Ojeda, Cristina Valls, Miquel Mulero, Santiago Garcia-Vallvé, and Gerard Pujadas. Molecular fingerprint similarity search in virtual screening. *Methods*, 71: 58–63, 2015.

Wenlin Chen, Austin Tripp, and José Miguel Hernández-Lobato. Meta-learning adaptive deep kernel gaussian processes for molecular property prediction. *arXiv preprint arXiv:2205.02708*, 2022.

Dimitrios Christofidellis, Giorgio Giannone, Jannis Born, Ole Winther, Teodoro Laino, and Matteo Manica. Unifying molecular and textual representations via multi-task language modelling. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 6140–6157. PMLR, 2023. URL https://proceedings.mlr.press/v202/christofidellis23a.html.

Kangway V Chuang and Michael J Keiser. Comment on "predicting reaction performance in C–N cross-coupling using machine learning". *Science*, 362(6416):eaat8603, 2018.

Erik Daxberger, Agustinus Kristiadi, Alexander Immer, Runa Eschenhagen, Matthias Bauer, and Philipp Hennig. Laplace redux–effortless Bayesian deep learning. In *NeurIPS*, 2021.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Natalie S Eyke, William H Green, and Klavs F Jensen. Iterative experimental design based on active machine learning reduces the experimental burden associated with reaction screening. *Reaction Chemistry & Engineering*, 5(10):1963–1972, 2020.

William Falcon and The PyTorch Lightning team. PyTorch Lightning, 3 2019. URL https://github.com/Lightning-AI/lightning.

Kobi C Felton, Jan G Rittig, and Alexei A Lapkin. Summit: benchmarking machine learning methods for reaction optimisation. *Chemistry-Methods*, 1(2):116–122, 2021.

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

Jacob R Gardner, Geoff Pleiss, David Bindel, Kilian Q Weinberger, and Andrew Gordon Wilson. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. In *Advances in Neural Information Processing Systems*, 2018.

Roman Garnett. *Bayesian optimization*. Cambridge University Press, 2023.

David E. Graff, Eugene I. Shakhnovich, and Connor W. Coley. Accelerating high-throughput virtual screening through molecular pool-based active learning. *Chemical Science*, 12(22):7866–7881, 2021. ISSN 2041-6539. doi: 10.1039/d0sc06805e.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Ryan-Rhys Griffiths and José Miguel Hernández-Lobato. Constrained bayesian optimization for automatic chemical design using variational autoencoders. *Chemical science*, 11(2):577–586, 2020.

Ryan-Rhys Griffiths, Jake L. Greenfield, Aditya R. Thawani, Arian R. Jamasb, Henry B. Moss, Anthony Bourached, Penelope Jones, William McCorkindale, Alexander A. Aldrick, Matthew J. Fuchter, and Alpha A. Lee. Data-driven discovery of molecular photoswitches with multioutput gaussian processes. *Chemical Science*, 13(45):13541–13551, 2022. ISSN 2041-6539. doi: 10.1039/d2sc04306h.

Ryan-Rhys Griffiths, Leo Klarner, Henry Moss, Aditya Ravuri, Sang T. Truong, Yuanqi Du, Samuel Don Stanton, Gary Tom, Bojana Ranković, Arian Rokkum Jamasb, Aryan Deshwal, Julius Schwartz, Austin Tripp, Gregory Kell, Simon Frieder, Anthony Bourached, Alex James Chan, Jacob Moss, Chengzhi Guo, Johannes P. Dürholt, Saudamini Chaurasia, Ji Won Park, Felix Strieth-Kalthoff, Alpha Lee, Bingqing Cheng, Alan Aspuru-Guzik, Philippe Schwaller, and Jian Tang. GAUCHE: A library for gaussian processes in chemistry. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL `https://openreview.net/forum?id=vzrA6uqOis`.

Antoine Grosnit, Rasul Tutunov, Alexandre Max Maraval, Ryan-Rhys Griffiths, Alexander I Cowen-Rivers, Lin Yang, Lin Zhu, Wenlong Lyu, Zhitang Chen, Jun Wang, et al. High-dimensional bayesian optimisation with variational autoencoders and deep metric learning. *arXiv preprint arXiv:2106.03609*, 2021.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Jeff Guo, Bojana Ranković, and Philippe Schwaller. Bayesian optimization for chemical reactions. *Chimia*, 77(1/2):31–38, 2023.

Florian Häse, Matteo Aldeghi, Riley J Hickman, Loïc M Roch, Melodie Christensen, Elena Liles, Jason E Hein, and Alán Aspuru-Guzik. Olympus: a benchmarking framework for noisy optimization and experiment planning. *Machine Learning: Science and Technology*, 2(3):035021, 2021a.

Florian Häse, Matteo Aldeghi, Riley J. Hickman, Loïc M. Roch, Melodie Christensen, Elena Liles, Jason E. Hein, and Alán Aspuru-Guzik. Olympus: a benchmarking framework for noisy optimization and experiment planning. *Machine Learning: Science and Technology*, 2(3):035021, July 2021b. ISSN 2632-2153. doi: 10.1088/2632-2153/abedc8.

Thomas Hayes, Roshan Rao, Halil Akin, Nicholas J Sofroniew, Deniz Oktay, Zeming Lin, Robert Verkuil, Vincent Q Tran, Jonathan Deaton, Marius Wiggert, et al. Simulating 500 million years of evolution with a language model. *Science*, pp. eads0018, 2025.

Riley Hickman, Jurgis Ruža, Loïc Roch, Hermann Tribukait, and Alberto García-Durán. Equipping data-driven experiment planning for self-driving laboratories with semantic memory: case studies of transfer learning in chemical reaction optimization. *ChemRxiv*, 2022.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pp. 2790–2799. PMLR, 2019.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55, 2025.

John PA Ioannidis. Why most published research findings are false. *PLoS medicine*, 2(8):e124, 2005.

Kevin Maik Jablonka, Philippe Schwaller, Andres Ortega-Guerrero, and Berend Smit. Leveraging large language models for predictive chemistry. *Nature Machine Intelligence*, 6(2):161–169, 2024.

Donald R Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of global optimization*, 21:345–383, 2001.

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

Mahmut Kaya and Hasan Şakir Bilge. Deep metric learning: A survey. *Symmetry*, 11(9):1066, 2019.

Steven Kearnes, Kevin McCloskey, Marc Berndl, Vijay Pande, and Patrick Riley. Molecular graph convolutions: moving beyond fingerprints. *Journal of computer-aided molecular design*, 30:595–608, 2016.

Agustinus Kristiadi, Felix Strieth-Kalthoff, Marta Skreta, Pascal Poupart, Alán Aspuru-Guzik, and Geoff Pleiss. A sober look at LLMs for material discovery: Are they actually good for Bayesian optimization over molecules? In *ICML*, 2024.

Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*, 2018.

Harold J Kushner. A versatile stochastic model of a function of unknown and time varying form. *Journal of Mathematical Analysis and Applications*, 5(1):150–167, 1962.

Harold J Kushner. A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. 1964.

Sean Lee, Aamir Shakir, Darius Koenig, and Julius Lipp. Open source strikes bread - new fluffy embeddings model, 2024. URL `https://www.mixedbread.ai/blog/mxbai-embed-large-v1`.

Xianming Li and Jing Li. Angle-optimized text embeddings. *arXiv preprint arXiv:2309.12871*, 2023.

Yixiao Li, Yifan Yu, Chen Liang, Pengcheng He, Nikos Karampatziakis, Weizhu Chen, and Tuo Zhao. Loftq: Lora-fine-tuning-aware quantization for large language models. *arXiv preprint arXiv:2310.08659*, 2023a.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*, 2023b.

Steven A. Lopez, Edward O. Pyzer-Knapp, Gregor N. Simm, Trevor Lutzow, Kewei Li, Laszlo R. Seress, Johannes Hachmann, and Alán Aspuru-Guzik. The harvard organic photovoltaic dataset. *Scientific Data*, 3(1), September 2016. ISSN 2052-4463. doi: 10.1038/sdata.2016.86.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D White, and Philippe Schwaller. Augmenting large language models with chemistry tools. *Nature Machine Intelligence*, 6(5):525–535, 2024.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. On faithfulness and factuality in abstractive summarization. *arXiv preprint arXiv:2005.00661*, 2020.

Lewis H Mervin, Simon Johansson, Elizaveta Semenova, Kathryn A Giblin, and Ola Engkvist. Uncertainty quantification in drug design. *Drug discovery today*, 26(2):474–489, 2021.

Pia Müller, Adam D Clayton, Jamie Manson, Samuel Riley, Oliver S May, Norman Govan, Stuart Notman, Steven V Ley, Thomas W Chamberlain, and Richard A Bourne. Automated multi-objective reaction optimisation: which algorithm should i use? *Reaction Chemistry & Engineering*, 7(4):987–993, 2022.

Tung Nguyen, Qiuyi Zhang, Bangding Yang, Chansoo Lee, Jorg Bornschein, Sagi Perel, Yutian Chen, and Xingyou Song. Predicting from strings: Language model embeddings for bayesian optimization. 2024.

Sebastian W Ober, Carl E Rasmussen, and Mark van der Wilk. The promises and pitfalls of deep kernel learning. In *Uncertainty in Artificial Intelligence*, pp. 1206–1216. PMLR, 2021.

OpenAI. New embedding models and api updates, 2024. URL `https://openai.com/index/new-embedding-models-and-api-updates/`.

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

Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, et al. Check your facts and try again: Improving large language models with external knowledge and automated feedback. *arXiv preprint arXiv:2302.12813*, 2023.

Damith Perera, Joseph W Tucker, Shalini Brahmbhatt, Christopher J Helal, Ashley Chong, William Farrell, Paul Richardson, and Neal W Sach. A platform for automated nanomole-scale reaction screening and micromole-scale synthesis in flow. *Science*, 359(6374):429–434, 2018.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*, 2019.

Alexander Pomberger, AA Pedrina McCarthy, Ahmad Khan, Simon Sung, CJ Taylor, MJ Gaunt, Lucy Colwell, David Walz, and AA Lapkin. The effect of chemical representation on active machine learning towards closed-loop optimization. *Reaction Chemistry & Engineering*, 7(6): 1368–1379, 2022.

Cesar N Prieto Kullmer, Jacob A Kautzky, Shane W Krska, Timothy Nowak, Spencer D Dreher, and David WC MacMillan. Accelerating reaction generality and mechanistic insight through additive mapping. *Science*, 376(6592):532–539, 2022.

Daniel Probst, Philippe Schwaller, and Jean-Louis Reymond. Reaction classification and yield prediction using the differential reaction fingerprint drfp. *Digital Discovery*, 1(2):91–97, 2022.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL `http://jmlr.org/papers/v21/20-074.html`.

Mayk Caldas Ramos, Shane S. Michtavy, Marc D. Porosoff, and Andrew D. White. Bayesian optimization of catalysts with in-context learning, 2023.

Bojana Ranković and Philippe Schwaller. Bochemian: Large language model embeddings for bayesian optimization of chemical reactions. In *NeurIPS 2023 Workshop on Adaptive Experimental Design and Active Learning in the Real World*, 2023.

Bojana Ranković, Ryan-Rhys Griffiths, Henry B Moss, and Philippe Schwaller. Bayesian optimisation for additive screening and yield improvements–beyond one-hot encoding. *Digital Discovery*, 3(4):654–666, 2024.

David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, 50(5):742–754, 2010.

Nadine Schneider, Daniel M Lowe, Roger A Sayle, and Gregory A Landrum. Development of a novel fingerprint for chemical reactions and its application to large-scale reaction classification and similarity. *Journal of Chemical Information and Modeling*, 55(1):39–53, 2015.

Alexandre A Schoepfer, Jan Weinreich, Ruben Laplaza, Jerome Waser, and Clemence Corminboeuf. Cost-informed bayesian reaction optimization. *Digital Discovery*, 3(11):2289–2297, 2024.

Philippe Schwaller, Daniel Probst, Alain C Vaucher, Vishnu H Nair, David Kreutter, Teodoro Laino, and Jean-Louis Reymond. Mapping the space of chemical reactions using attention-based neural networks. *Nature Machine Intelligence*, 3(2):144–152, 2021.

Artur M Schweidtmann, Adam D Clayton, Nicholas Holmes, Eric Bradford, Richard A Bourne, and Alexei A Lapkin. Machine learning meets continuous flow chemistry: Automated optimization towards the Pareto front of multiple objectives. *Chemical Engineering Journal*, 352:277–282, 2018.

Benjamin J Shields, Jason Stevens, Jun Li, Marvin Parasram, Farhan Damani, Jesus I Martinez Alvarado, Jacob M Janey, Ryan P Adams, and Abigail G Doyle. Bayesian reaction optimization as a tool for chemical synthesis. *Nature*, 590(7844):89–96, 2021.

Sukriti Singh and José Miguel Hernández-Lobato. Deep kernel learning for reaction outcome prediction and optimization. *Communications Chemistry*, 7(1):136, 2024.

Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*, 2009.

Samuel Stanton, Wesley Maddox, Nate Gruver, Phillip Maffettone, Emily Delaney, Peyton Greenside, and Andrew Gordon Wilson. Accelerating bayesian optimization for biological sequence design with denoising autoencoders. In *International conference on machine learning*, pp. 20459–20478. PMLR, 2022.

Hongjin Su, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A Smith, Luke Zettlemoyer, Tao Yu, et al. One embedder, any task: Instruction-finetuned text embeddings. *arXiv preprint arXiv:2212.09741*, 2022.

Francesca Tavazza, Brian DeCost, and Kamal Choudhary. Uncertainty prediction for machine learning models of material properties. *ACS omega*, 6(48):32431–32440, 2021.

Connor J Taylor, Alexander Pomberger, Kobi C Felton, Rachel Grainger, Magda Barecka, Thomas W Chamberlain, Richard A Bourne, Christopher N Johnson, and Alexei A Lapkin. A brief introduction to chemical reaction optimization. *Chemical Reviews*, 123(6):3089–3126, 2023.

William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

Jose Antonio Garrido Torres, Sii Hong Lau, Pranay Anchuri, Jason M Stevens, Jose E Tabora, Jun Li, Alina Borovika, Ryan P Adams, and Abigail G Doyle. A multi-objective active learning platform and web app for reaction optimization. *Journal of the American Chemical Society*, 144 (43):19999–20007, 2022.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Nathan Cooper, Griffin Adams, Jeremy Howard, and Iacopo Poli. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference, 2024. URL `https://arxiv.org/abs/2412.13663`.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022.

David Weininger. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences*, 28(1):31–36, 1988.

Andrew D White. The future of chemistry is language. *Nature Reviews Chemistry*, pp. 1–2, 2023.

Christopher K Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*. MIT press Cambridge, MA, 2006.

Andrew Wilson and Hannes Nickisch. Kernel interpolation for scalable structured gaussian processes (kiss-gp). In *International conference on machine learning*, pp. 1775–1784. PMLR, 2015.

Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P Xing. Deep kernel learning. In *Artificial intelligence and statistics*, pp. 370–378. PMLR, 2016.

Tong Xie, Yuwei Wan, Wei Huang, Zhenyu Yin, Yixuan Liu, Shaozhou Wang, Qingyuan Linghu, Chunyu Kit, Clara Grazian, Wenjie Zhang, et al. Darwin series: Domain specific large language models for natural science. *arXiv preprint arXiv:2308.13565*, 2023.

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

Adam X Yang, Maxime Robeyns, Xi Wang, and Laurence Aitchison. Bayesian low-rank adaptation for large language models. *arXiv preprint arXiv:2308.13111*, 2023.

A Related work

Adapting LLMs to specialized tasks through finetuning typically optimizes for predictive accuracy (Jablonka et al., 2024; Xie et al., 2023). Such domain adaptation neglects the dimension of epistemic uncertainty to indicate when model outputs should not be trusted. In this context, the challenge of extracting reliable uncertainties from LLMs for efficient BO has introduced diverse approaches that fall into four categories.

(1) Prompt-based methods like BOLIFT (Ramos et al., 2023), estimate uncertainty by aggregating multiple LLM responses, while LLAMBO (Nguyen et al., 2024) queries LLMs as optimizers. (2) Embedding-based methods, such as BoChemian (Ranković & Schwaller, 2023), model GPs on static LLM representations. (3) Surrogate conversion models either predict uncertainty through pretrained in-context regressor (Nguyen et al., 2024) or transform LLMs into Bayesian neural networks through PEFT and Laplace approximation (LAPEFT (Daxberger et al., 2021) following Bayesian LoRA approach (Yang et al., 2023)). (4) Hybrid approaches like BOPRO (Agarwal et al.) combine

fixed LLM embeddings for GP-based acquisition optimization with prompt-conditioned LLM generation, using in-context examples to explore the solution space. While promising, these methods face limitations from heavy prompt engineering to post-hoc uncertainty fixes. Our approach offers a solution by integrating uncertainty modeling *during* training within both the LLM and the surrogate model (GP) through LLM-based deep kernel strategy.

Previous works on integrating LLMs and GPs (Ranković et al., 2024; Ramos et al., 2023; Ranković & Schwaller, 2023; Kristiadi et al., 2024) for chemical optimization primarily use LLMs as fixed feature extractors. Despite LLM's flexibility in converting diverse parameters into fixed-dimensional representations, these approaches essentially reduce their power to sophisticated encoding tools. The static embeddings struggle to match, let alone surpass, carefully engineered domain-specific features, creating an artificial barrier between LLM expressiveness and GP rigor.

The closest related approach to ours is the one by Kristiadi et al. (2024). It incorporates supervised LLM finetuning during optimization (for better prediction), followed by post-hoc Laplace approximation of learned weights (for uncertainty estimates). The sequential approach however, decouples uncertainty quantification from the learning process and optimizes for prediction accuracy (lower MSE loss) rather than optimization itself. Our method fundamentally differs by integrating LLMs directly into the GP framework as a deep kernel, making uncertainty quantification an integral part of the optimization objective through the GP marginal likelihood.

Existing deep kernel methods (Singh & Hernández-Lobato, 2024; Chen et al., 2022) in chemistry constrain to using graph neural network kernels and domain specific representations. Advancing this concept, we demonstrate that LLM-based kernels provide richer embedding space and easier adaptation to various domains. In the latent space optimization, our approach relates to BO with variational autoencoders (VAEs). For example, (Griffiths & Hernández-Lobato, 2020) use VAE-BO for constrained optimization of molecules while (Grosnit et al., 2021) introduce explicit deep metric learning to structure the latent space. We elevate this approach by replacing pretrained VAEs with general purpose LLMs. Moreover, our joint optimization induces implicit metric learning, directly structuring the latent space without the need for explicit contrastive objectives. Importantly, deep kernel learning in low-data regime has long been considered challenging due to issues like overfitting and training instability (Ober et al., 2021), with some approaches opting for coordinate ascent or bi-level optimization strategies to mitigate these issues (Stanton et al., 2022; Chen et al., 2022). In contrast, we build a robust and stable training pipeline using standard backpropagation with separate learning rates for the LLM and GP components, avoiding the need for complex optimization schedules or custom regularization.

Operating in high-dimensional BO (HDBO) spaces (768 features with BERT-based models, 4096 with larger decoder types), our work demonstrates successful optimization that challenges conventional assumptions about HDBO limitations (Frazier, 2018). Through our empirical analysis, we identify key principles for effective HDBO: embeddings should structure following the GP's inductive bias. This organization of the embedding space plays a crucial role in overcoming dimensionality challenges that have traditionally constrained BO approaches. Recent work on HDBO has identified vanishing gradients in GP training and acquisition optimization as a key challenge, and proposed solutions based on informed lengthscale priors and local search heuristics (Papenmeier et al., 2025). Our approach complements this line of work by focusing on the structure of the representation space itself, showing that smoothness and generalization can emerge naturally when the embedding is well-aligned with the GP kernel. Our observations contribute to emerging approaches in understanding and improving HDBO performance (Papenmeier et al., 2025; Xu et al., 2025).

Within chemistry, high-dimensional features are the default (one-hot encodings (Chuang & Keiser, 2018), fingerprints (Rogers & Hahn, 2010; Schneider et al., 2015; Capecchi et al., 2020; Probst et al., 2022; Schwaller et al., 2021) quantum mechanical descriptors (Ahneman et al., 2018; Shields et al., 2021)) but their selection remains challenging (Ranković et al., 2024). We overcome this problem through unified representation learning as an implicit objective of the optimization strategy itself. As a result, we achieve adaptive features for any optimization at hand, without requiring traditionally used expert-based descriptors.

B TECHNICAL BACKGROUND

B.1 BAYESIAN OPTIMIZATION

Bayesian optimization (BO) is a suitable method for optimizing expensive-to-evaluate functions with unknown analytic form or gradients. Such problems are common in chemistry where experimental evaluations are often costly, time-consuming, and only available through real-world lab experiments. The primary objective of BO is to find:

$$\mathbf{x}^* = \arg\max_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}) \tag{5}$$

where $f : \mathcal{X} \to \mathbb{R}$ is the objective function over domain $\mathcal{X}$. In practical chemistry applications, we often work in a constrained domain $\mathcal{X}_{\text{pool}}$ with limited set of possible experimental conditions or molecular structures (e.g., feasible reaction conditions, available reagents or compound libraries). The optimization objective may involve maximizing reaction yield or selectivity, or minimizing properties such as reaction time, cost or toxicity.

B.2 SEQUENTIAL DECISION PROCESS

Bayesian optimization operates as a sequential decision-making process that balances exploration and exploitation. Key components include (1) a probabilistic surrogate model of the underlying objective function f and (2) an acquisition function α. The acquisition function guides the optimization process by proposing subsequent evaluation points. Common choices include expected improvement (EI (Jones et al., 1998)), probability of improvement (PI (Jones, 2001)), upper confidence bound (UCB (Auer, 2002), (Srinivas et al., 2009)), and Thompson sampling (TS (Thompson, 1933)). Acquisition function selection over the points in the design space relies on the surrogate model and its predictive and uncertainty estimates. For example EI selects points that, in expectation, improve upon the current best observed value $f(\mathbf{x}_{\text{best}})$:

$$\alpha_{\text{EI}}(\mathbf{x}|\mathcal{D}_t) = \mathbb{E}_{p(f|\mathcal{D}_t)}[\max(f(\mathbf{x}) - f(\mathbf{x}_{\text{best}}), 0)] \tag{6}$$

In that sense, the choice of a surrogate model is critical to the success of BO. Gaussian Processes (GPs) are the most common choice due to their flexibility and ability to quantify uncertainty, making them particularly suitable for guiding the exploration of vast chemical spaces while operating in low-data regimes.

B.3 GAUSSIAN PROCESSES AND MARGINAL LIKELIHOOD OPTIMIZATION

GPs provide a flexible non-parametric method for modeling unknown functions. A GP places a prior distribution

$$f(\mathbf{x}) \sim \mathcal{GP}(c, k(\mathbf{x}, \mathbf{x}')), \tag{7}$$

defined by a mean function c (typically 0 or constant) and a kernel function k encoding pairwise similarity between inputs and prior assumptions about function smoothness and variability. A common choice is the Matérn-5/2 kernel

$$k_{\text{Matérn-5/2}}(\mathbf{x}, \mathbf{x}') = \sigma^2 \left(1 + \frac{\sqrt{5}d}{\ell} + \frac{5d^2}{3\ell^2}\right) \exp\left(-\frac{\sqrt{5}d}{\ell}\right), \tag{8}$$

where $d = \|\mathbf{x} - \mathbf{x}'\|_2$, ℓ is the lengthscale, and σ^2 is the signal variance.

Given training data $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, the GP posterior allows closed-form prediction of the function at new points, along with uncertainty estimates. Crucially, the GP hyperparameters $\theta = \{c, \ell, \sigma^2, \sigma_n^2\}$ are learned by maximizing the marginal likelihood of the data:

$$\mathcal{L}(\theta) = \log p(\mathbf{y}|\mathbf{X}, \theta) = -\frac{1}{2}(\mathbf{y}^\top \mathbf{K}_\theta^{-1}\mathbf{y} + \log|\mathbf{K}_\theta| + n\log 2\pi) \tag{9}$$

where $\mathbf{K}_\theta$ is the kernel matrix incorporating observation noise σ_n^2, evaluated on the training inputs using kernel parameters (ℓ, σ^2). If a constant mean function c is used, the targets $\mathbf{y}$ are centered as $\mathbf{y} - c\mathbf{1}$ during marginal likelihood computation. In the standard fixed-feature setting, input $\mathbf{x}$ is mapped to a feature vector via a static transformation (e.g., molecular fingerprints or frozen LLM embeddings), and the GP operates solely on these representations. The optimization updates θ, adapting the GP's inductive bias to the fixed feature space.

B.4 Deep Kernel Gaussian Processes

Deep Kernel Gaussian Processes (DKGPs) introduce an additional parameter set ϕ to the optimization objective by integrating neural network-based feature transformations into the GP kernel. Formally, the kernel function becomes:

$$k_{\theta,\phi}(\mathbf{x}, \mathbf{x}') = k_\theta(g_\phi(\mathbf{x}), g_\phi(\mathbf{x}')),$$

where $g_\phi(\cdot)$ is a learned data representation parameterized by ϕ. This formulation enables the model to adapt the input space to the task at hand while preserving the uncertainty modeling properties of the GP.

The transformation g_ϕ can take the form of any neural architecture suitable for the data modality. The original paper applied the DKGP architecture on regression tasks with images using convolutional neural networks (Wilson et al., 2016), while subsequent works have extended it to structured chemical domains using graph neural networks (Singh & Hernández-Lobato, 2024). As detailed in the main section, we apply this framework to textual chemical representations by using large language models (LLMs) as the deep kernel feature extractor. This allows us to incorporate both pretrained domain knowledge and task-specific adaptation within the BO loop.

B.5 Large Language Models

LLMs process textual inputs by converting them into dense vector representations through a sequence of tokenization (), embedding () and attention-based () transformations. Tokenization involves the process of splitting the input text into subword units (tokens) using a model-specific vocabulary (e.g., SentencePiece (Kudo & Richardson, 2018), Byte-Pair Encoding (Radford et al., 2019)). The tokens are mapped to continuous vectors via learned embedding layers and passed through multiple self-attention layers that capture contextual relationships between tokens.

LLMs can follow different architectural designs: encoder-only (e.g., BERT (Devlin et al., 2018)), decoder-only (e.g., Qwen (Bai et al., 2023)), and encoder-decoder (e.g., T5 (Raffel et al., 2020)). Encoder-based models process the full input bidirectionally and are suited for classification and regression (). Decoder-only models generate text autoregressively with causal masking (). Encoder-decoder models combine both components and are often used for sequence-to-sequence tasks (). The architecture choices impact the structure and pooling strategies used to extract unified representations from the variable-length token sequences.

Pooling refers to the process of aggregating a sequence of token-level representations produced by a language model into a single fixed-dimensional embedding. Encoder-based models often use the hidden state corresponding to the special [CLS] token or apply mean-pooling across token embeddings. Decoder-only models typically use the final hidden state of the last non-padding token. For encoder-decoder models, pooling is applied over the encoder-side hidden states.

B.6 Parameter-efficient LLM Finetuning

Although pretrained LLM embeddings encode rich semantic information, they are not tailored to specific downstream tasks. In that sense, adapting LLMs through finetuning allows for better task-specific capabilities. However, updating LLM weights can be computationally prohibitive due to their large size (often billions of parameters in modern LLMs). Parameter-efficient finetuning (PEFT), however, provides a recipe for LLM task alignment by adapting a smaller subset of parameters while leaving the majority of the model unchanged.

One such approach is Low-Rank Adaptation (LoRA) (Hu et al., 2022), which injects trainable low-rank matrices into existing weight layers. Instead of updating a weight matrix $W \in \mathbb{R}^{d\times k}$, LoRA

learns a low-rank update of the form:

$$\Delta W = AB, \quad \text{where } A \in \mathbb{R}^{d\times r}, \quad B \in \mathbb{R}^{r\times k}, \quad r \ll \min(d,k)$$

The adapted weight becomes $W' = W + \Delta W$, allowing task-specific learning with a parameter count that scales with r, the rank of the decomposition. This method allows efficient finetuning and mitigates the risk of catastrophic forgetting by preserving the pretrained weights.

B.7 PSEUDOCODES

Algorithm 1 Constrained Bayesian Optimization

Require: Initial dataset $\mathcal{D}_0 = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n_0}$, candidate pool $\mathcal{X}_{\text{pool}}$, budget T, objective function f
Initialize surrogate model (e.g., GP) using $\mathcal{D}_0$
for $t = 1$ to T **do**
  Fit surrogate model to current data $\mathcal{D}_{t-1}$
  for all $\mathbf{x} \in \mathcal{X}_{\text{pool}}$ **do**
    Compute acquisition value $\alpha(\mathbf{x} \mid \mathcal{D}_{t-1})$
  end for
  Select next point: $\mathbf{x}_t = \arg\max_{\mathbf{x}\in\mathcal{X}_{\text{pool}}} \alpha(\mathbf{x} \mid \mathcal{D}_{t-1})$
  Evaluate objective function: $y_t = f(\mathbf{x}_t)$
  Update dataset: $\mathcal{D}_t = \mathcal{D}_{t-1} \cup \{(\mathbf{x}_t, y_t)\}$
  Remove $\mathbf{x}_t$ from $\mathcal{X}_{\text{pool}}$
end for
return Best observed point: $\mathbf{x}^* = \arg\max_{(\mathbf{x},y)\in\mathcal{D}_T} y$

Algorithm 2 Bayesian Optimization with LLM-based Deep Kernel GP

Require: Initial dataset $\mathcal{D}_0 = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n_0}$, candidate pool $\mathcal{X}_{\text{pool}}$, budget T
for $t = 1$ to T **do**
  Initialize parameters ϕ (LLM) and θ (GP)
  Train LLM-GP model:
  repeat
    Compute embeddings: $\mathbf{z}_i = g_\phi(\mathbf{x}_i)$ for all $(\mathbf{x}_i, y_i) \in \mathcal{D}_{t-1}$
    Evaluate GP marginal log-likelihood $\log p(\mathbf{y} \mid \mathbf{z}, \theta)$
    Update ϕ, θ
  until convergence
  Compute acquisition on candidate pool:
  for all $\mathbf{x}_j \in \mathcal{X}_{\text{pool}}$ **do**
    $\mathbf{z}_j = g_\phi(\mathbf{x}_j)$
    Compute $\alpha(\mathbf{z}_j; \theta)$
  end for
  Select next input: $\mathbf{x}_t = \arg\max_{\mathbf{x}_j} \alpha(\mathbf{z}_j)$
  Observe outcome: $y_t = f(\mathbf{x}_t)$
  Update dataset: $\mathcal{D}_t = \mathcal{D}_{t-1} \cup \{(\mathbf{x}_t, y_t)\}$
  Remove $\mathbf{x}_t$ from $\mathcal{X}_{\text{pool}}$
end for
return Best input: $\arg\max_{(\mathbf{x},y)\in\mathcal{D}_T} y$

C LLM AND DOMAIN-SPECIFIC REPRESENTATIONS

C.1 MOLECULAR REPRESENTATIONS

Molecular representations have been extensively studied in chemistry, leading to (1) fingerprints (Rogers & Hahn, 2010; Cereto-Massagué et al., 2015; Capecchi et al., 2020), (2) Simplified Molecular Input Line Entry System (SMILES) strings (Anderson et al., 1987; Weininger, 1988), (3) molecular graph-based features (Kearnes et al., 2016) or (4) more physics-informed descriptors derived

from electronic structure calculations (Bannwarth et al., 2019). Each of these representations encodes different aspects of molecular structure and properties making their utility task-dependent.

For example, molecular fingerprints can be effective for tasks involving structural similarity or substructure-driven properties, while quantum-derived features may be better suited for tasks involving electronic properties. For applications in BO, these representations typically require specialized kernel functions to capture relevant similarity (Griffiths et al., 2023).

To compare domain-specific representation to general LLM-based ones, we set molecular fingerprints as input to a GP in all molecular property optimization benchmarks.

C.2 Reaction Representations

Chemical reactions, on the other hand, attach an additional layer of complexity beyond molecular representation including reaction conditions and procedural descriptions. In that sense, they present a unique challenge for machine learning due to an increased complexity and heterogeneous nature. Reaction conditions typically comprise multiple parameter types: numerical values (temperature, concentration, time), categorical variables (catalyst type, solvent choice), and detailed procedural descriptions, making their featurization challenging. Reaction representations used in ML range from simple one-hot encodings (Chuang & Keiser, 2018) to more elaborate reaction fingerprints (Rogers & Hahn, 2010; Schneider et al., 2015; Capecchi et al., 2020; Probst et al., 2022), quantum mechanical descriptors (Ahneman et al., 2018; Shields et al., 2021) and learned representations (Schwaller et al., 2021).

We make extensive use of Differential Reaction Fingerprints (DRFPs) (Probst et al., 2022), previously shown to achieve state-of-the-art results on reaction optimization tasks compared to a variety of molecular and reaction descriptors (Ranković et al., 2024). Generated by first computing circular fingerprints for each reactant and product and then taking their symmetric difference, DRFPs highlight the structural changes during the reaction while remaining computationally cheap. We input DRFPs to GP in reaction optimization tasks with available reaction SMILES. For other chemical optimization tasks we generate features through one-hot encoding of categorical variables, concatenated with numerical parameter values.

C.3 LLM representations

C.3.1 Template Construction

We define each task t through a standardized template: $t = \text{template}(\{\text{parameters}, \text{values}\})$ where the template converts various parameter types into a structured text format:

```
The reaction was prepared with:
temperature: {numerical_value}°C
solvent: {solvent_smile}
ligand: {ligand_smile}
```

C.3.2 Encoder-Based Models

Encoder-based

Encoder-based language models, originally developed for natural language understanding tasks, have long been used to generate fixed-dimensional vector representations of text. These models, typically based on the transformer encoder architecture, process input sequences bidirectionally. In that sense, they have been widely adopted in various downstream tasks such as classification, clustering, and semantic similarity.

With the recent evolution toward larger-scale pretraining, encoder-only models have also followed the trajectory of large language models, yielding high-capacity embedding models suitable for diverse domains beyond natural language. These modern embedding models are trained on massive corpora with contrastive or retrieval-oriented objectives, making them particularly effective for extracting general-purpose sentence and document embeddings.

In this work, we evaluate three encoder-based embedding models on tasks of representing chemical procedures and reaction descriptions for BO.

ModernBERT (Warner et al., 2024): A compact and efficient embedding model trained with a retrieval objective, designed for high-speed and high-quality sentence representations.

UAE-Large (Li & Li, 2023): The Universal Alignment Embedding model, trained for multilingual and multimodal generalization with a strong emphasis on alignment across domains.

MXBAI-Embed (Lee et al., 2024): A large-scale embedding model from Mixedbread-AI, trained to preserve semantic similarity across a broad range of tasks, including code, math, and text.

C.3.3 Encoder-Decoder Models

Encoder-decoder architectures, such as the T5 family (Raffel et al., 2020), consist of two transformer modules: an encoder that processes the input sequence and a decoder that generates output sequences, typically in an autoregressive fashion. For embedding tasks, representations are typically extracted from the encoder side, which embeds the input text into a fixed-length latent representation. Compared to encoder-only models, encoder-decoder architectures are often pretrained with sequence-to-sequence objectives such as masked span prediction or denoising, making them well-suited for tasks involving paraphrasing, summarization, or input–output alignment.

We evaluate the encoder outputs from three encoder-decoder models:

T5 (base variant) (Raffel et al., 2020): A widely-used general-purpose model pretrained on a multi-task mixture of unsupervised and supervised NLP tasks. We use the encoder outputs as text embeddings.

T5Chem (Christofidellis et al., 2023) : A domain-adapted variant of T5, finetuned on chemical tasks using the GT4SD framework. It is trained on a multitask mixture involving molecular property prediction, retrosynthesis, and chemical text modeling, making it more specialized for chemistry-related input sequences.

Instructor (Su et al., 2022): An instruction-tuned encoder-decoder model trained on natural language–task pairs. It learns to produce embeddings guided by a task description (e.g., "Represent the reaction for similarity search"), making it suitable for alignment-sensitive downstream applications.

C.3.4 Decoder-Only Models

Decoder-only architectures, exemplified by models in the GPT (Radford et al., 2018) family, generate outputs autoregressively by predicting each token conditioned on all previous ones. While traditionally used for generation tasks, these models can also produce dense representations of input text by extracting hidden states from specific tokens (e.g., the final token or special marker tokens). Decoder-only models are typically pretrained with causal language modeling objectives and operate unidirectionally, which distinguishes their contextual encoding behavior from encoder-based models.

In this work, we evaluate several decoder-style models for embedding chemical procedures and reaction descriptions:

OpenAI Embeddings (OpenAI, 2024) A widely-used commercial API that provides text embeddings via proprietary transformer models. While the architectural details are not public, we assign them to decoder-style GPT family.

Qwen2-7B-Instruct (Bai et al., 2023) A large-scale instruction-tuned language model from Alibaba, based on a decoder-only architecture. We use this model in embedding mode by extracting the hidden state of the last non-padding token.

GTE-Qwen2-7B-Instruct (Li et al., 2023b) A retrieval-optimized variant of Qwen2, finetuned to produce sentence-level embeddings with improved performance on similarity and ranking tasks.

LLaMA 3–8B (Grattafiori et al., 2024) Meta's open LLaMA 3 model in its original instruction-tuned form, without additional adaptation for embeddings.

LLM2Vec Models (BehnamGhader et al., 2024) We also evaluate decoder-only LLMs adapted for embedding tasks using the LLM2Vec framework (**?**). These models, such as LLM2Vec–Meta-Llama-3 and LLM2Vec–Mistral-7B, are trained with masked next token prediction (MNTP) to enable bidirectional context modeling and use supervised mean pooling over selected internal layers. This adaptation allows decoder-only transformers to behave similarly to encoder models in embedding quality, while preserving their original architecture.

C.4 Representations Overview

We build LLM representations with models from HuggingFace (HF) selected through their base architecture and the results on MTEB in summarization task. In Table 1 we give an overview of all representations used in this paper alongside specifics on the dimensionality, pooling, architecture, pretraining, and connections to chemistry. We also include HF sources for LLM-based representations and links to chemistry-related molecular fingerprints and DRFP featurization methods.

Model	Arch.	Pretraining Objective	Pooling	Chem.	Dim.	Source
Molecular						
Fingerprints	/	/	/	Yes	2048	Morgan
Reaction						
DRFP	/	/	/	Yes	2048	DRFP
Encoder						
ModernBERT	Encoder	Retrieval contrastive	CLS	No	768	HF
MXBAI-Embed	Encoder	General-purpose	CLS	No	1024	HF
UAE-Large	Encoder	Alignment / Multimodal	CLS	No	1024	HF
Enc-Dec						
T5-Base	Enc-Dec	Masked span prediction	Mean	No	768	HF
T5Chem	Enc-Dec	Chem multitask	Mean	Yes	768	HF
Instructor	Enc-Dec	Instruction alignment	Weighted Mean	Part.[†]	768	HF
Decoder						
OpenAI Embedding	Decoder*	Proprietary	Unk.	Unk.	3072	OpenAI
Qwen2-7B-Instruct	Decoder	Instruction tuning	Last	No	3584	HF
GTE-Qwen2	Decoder	Contrastive retrieval	Last	No	3584	HF
LLM2Vec–LLaMA3	Decoder	Supervised pooling	Last	No	4096	HF
LLM2Vec–Mistral	Decoder	Supervised pooling	Last	No	4096	HF
LLaMA 3–8B	Decoder	Instruction tuning	Last	No	4096	HF

Table 1: Overview of data representations used in our experiments, including architecture, pretraining objective, pooling strategy, chemistry adaptation, and embedding dimensionality and direct link to the source.

D Benchmarking Datasets

D.1 Buchwald-Hartwig reactions

We performed all initial investigations on a set of Buchwald-Hartwig (BH) reactions with the task of optimizing yield (0-100%). This dataset consists of 3955 reactions spanning across five distinct products (BH1-BH5). The data originates from a high-throughput experimentation (HTE) study published by Ahneman et al. (Ahneman et al., 2018). For each product, reactions were evaluated based on their percentage yield as determined by HPLC analysis. The design space is combinatorial across 15 reactants (aryl halides), 22 additives, 4 ligands, and 3 bases in DMSO solvent.

[1]* While OpenAI Embedding model architecture is not publicly disclosed, we assign a decoder-style structure based on the GPT family.

[2]† Partially chemistry-aligned through prompting instructions like "Represent the chemical reaction."

In all initial BO experiments (unless explicitly stated as in T5Chem-SMILES), we represented reactions through procedural text template describing the reaction conditions in natural language. Moving forward to benchmarking against other models, we featurized reactions based on reaction SMILES, to ensure fair comparison to models that report best performance when using this representation (LAPEFT). Moreover, this dual representation allowed us to evaluate and establish robustness to the impact of different input formats on model performance.

D.2 Additive screening

This dataset originates from a study on organic additives' influence on the reactivity of complex Ni-catalysed reactions in a high-throughput experimentation (HTE) setup (Prieto Kullmer et al., 2022). It covers a wide range of screened additives (720) across four different reactions (Additives 1-4) and measures their effect on UV210 product area absorption. The challenge with traditional featurization methods in this dataset lies in the sole variability of the additive in the design space, while the other reaction parameters remain fixed. In such a setup, traditional one-hot encoding techniques would yield results similar to random search. Previous approaches report success with DRFP representation, comparing its performance to a set of molecular descriptors (Ranković et al., 2024). Compared to this method, we achieve better results while representing the reaction SMILES through LLM embeddings and optimizing with our GP-guided finetuning approach.

D.3 Suzuki-Miyaura reactions

Suzuki-Miyaura dataset is a reaction optimization benchmark that includes high-throughput evaluated Suzuki-Miyaura cross-coupling reactions, originally introduced by Perera et al. (Perera et al., 2018). The dataset was generated using an automated nanomole-scale synthesis platform designed to explore large combinatorial reaction spaces efficiently. It contains 5760 reactions across varied combinations of 7 unique electrophile, 4 nucleophile, 11 ligands (plus one blank), 7 bases (plus one blank), 4 solvents and Pd(OAc)2 as precatalyst, while the optimization objective is maximizing yield.

D.4 Catalyst optimization – C2 yield

We evaluate our method on optimizing methane oxidative coupling (OCM) using a subset of 1180 reactions from (Ramos et al., 2023). Each reaction involves synthesizing a supported catalyst (e.g., $Mn-Na_2WO_4/BN$) by impregnating a solid support (typically BN) with a solution of up to three metal precursors in defined molar ratios. Additional parameters include reaction temperatures (typically ˜900 °C), with controlled gas flows (CH_4, O_2, Ar) and contact times. The objective is to maximize C2 yield, a measure of desirable product formation. The original study already provides a textual template of reactions in this dataset, which we used for testing our method on diverse input formatting. For the standard GP baseline we featurize the data by concatenating numerical parameters and one-hot encoded categorical values.

D.5 Molecular optimization

We selected the benchmark datasets from Kristiadi et al. (Kristiadi et al., 2024) to both test our model on molecular property optimization and compare directly to their approach (LAPEFT). Moreover, we use molecular SMILES as the textual representation of the data following their best practice and offering a fair comparison between the methods. The five datasets we present in this work span diverse scientific applications and optimization objectives:

- Redox (1,407 samples): minimize redox potential for flow battery materials (Agarwal et al., 2021),
- Solvation (1,407): minimize solvation energy (Agarwal et al., 2021),
- Kinase (10,449): minimize docking score in kinase inhibitors (Graff et al., 2021),
- Photoswitch (392): maximize the $\pi - \pi^*$ transition wavelength* in organic photoswitches (Griffiths et al., 2022),

- PCE (10,000): maximize power conversion efficiency of photovoltaic materials (Lopez et al., 2016).

All objectives are continuous and we use molecular fingerprints as a chemistry-related featurization for the baseline GP comparison.

D.6 General Chemistry Benchmarks

To move beyond reaction and molecular optimization we include three datasets from Olympus (Häse et al., 2021b), spanning catalysis, crystallization, and process optimization, all of which involve continuous or mixed-variable optimization objectives. These datasets are commonly used in autonomous discovery and closed-loop optimization studies.

OER (Oxygen Evolution Reaction Catalysts). This dataset comprises 2,121 samples describing compositions of high-throughput screened catalysts for the oxygen evolution reaction. Each data point represents a combination of elemental loadings (Ni, Fe, Co, Mn, Ce, La) constrained to sum to 1. The optimization goal is to minimize the overpotential, a key descriptor of catalytic efficiency.

- **Target:** Overpotential (continuous)
- **Features:** 6 discrete fractional loadings

Vapdiff Crystallization (Crystal Score). This dataset reports the outcomes of vapor diffusion crystallization experiments across 918 combinations of organic, solvent, and inorganic conditions. The target is an *ordinal* score representing crystallization quality, with categorical and continuous inputs describing the experiment setup.

- **Target:** Crystal score (ordinal)
- **Features:** 10 variables (categorical, continuous, discrete)

Similarly to C2 yield optimization, we featurize the categorical variables through one-hot encoding and concatenate these vectors to the remaining numerical parameters for the input to the standard GP baseline.

HPLC (High-Performance Liquid Chromatography). This dataset includes 1,386 data points measuring peak response from an automated HPLC system as a function of six continuous process parameters such as flow rate, sample volume, and wait time. The objective is to maximize the peak signal (measured by photo degradation response).

- **Target:** Photo degradation (continuous)
- **Features:** 6 continuous parameters

E EXTENDED RESULTS

E.1 TOKENIZATION PER LLM TYPE

We investigate the impact of different pooling strategies on the quality of LLM embeddings for Bayesian optimization. Since LLMs produce variable-length token sequences, pooling plays a critical role in converting these sequences into fixed-size representations used by the GP surrogate. Our ablation reveals a clear interaction between model architecture and pooling choice. For encoder-only models, CLS token pooling outperforms alternatives – last token pooling dilutes the informative signal captured in the CLS token – specifically trained to represent global context. In contrast, decoder-only models tend to collapse all inputs to similar representations when the starting token is pooled, leading to duplicates and unsuccessful optimization. Here, last-token pooling aligns with the autoregressive structure and yields substantially better results. For encoder-decoder models, we observe lower differences in performance across pooling strategies, though mean pooling shows a slight edge in consistency.

Based on these findings, we adopt CLS pooling for encoder models, last-token pooling for decoder models, and mean pooling for encoder-decoder models throughout the main experiments. Figure 6 summarizes the performance differences across model types and pooling methods. The pooling choices ensure meaningful input representations across LLM types, and deviations from optimal setup can lead to noticeable performance drops within the fixed-feature setting (e.g., up to 40% in top-5 discovery rate between CLS and last-token pool for encoder models).

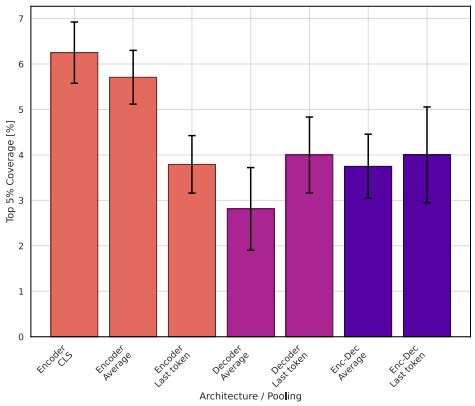

Figure 6: **Tokenization pool strategy per LLM type.** We compare the pooling strategies across LLM types. Encoder-based models benefit from utilizing the CLS token, unlike decoder architectures where this token collapses all inputs to duplicated representation (hence not appearing in the results). For decoder-based architectures, last token pooling improves results over token averaging. For encoder-decoder models, the difference between average and last token pool is less pronounced, however with a lower variability for mean pooling. The bars represent the standard error and we compare results on BH1 reaction with 25 BO iterations repeating the experiments over 10 seeds.

E.2 WHICH LLM LAYERS CARRY THE MOST INFORMATION?

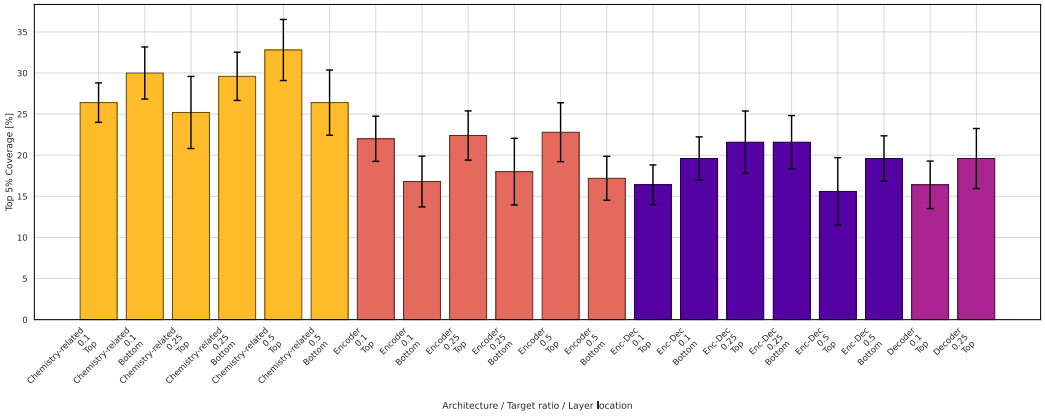

Figure 7: **Performance comparison of PEFT strategies across LLM architectures.** We vary the proportion (10%, 25%, 50%) and location (top vs. bottom) of targeted linear layers using LoRA. Results show that top-layer finetuning consistently outperforms bottom-layer updates for encoder-only and decoder-only∗ models. For encoder-decoder models, performance is more consistent across layer locations. Based on these findings, we fix the default to targeting the top 25% of linear layers.

We further examine which subset of LLM layers is most effective to target during PEFT. Since full model finetuning within the GP kernel is infeasible for large LLMs, we use LoRA to adapt only a fraction of the model weights. The choice of which layers to modify has a strong effect on optimization performance. Inspired by the intuition that higher (deeper) transformer layers encode more task-relevant semantics, we evaluate LoRA targeting strategies by varying both the location (top vs. bottom layers) and proportion (10%, 25%, 50%) of modified linear layers.

Figures 7 and 8 show the results across four (we now separate the encoder-decoder models into chemistry-related–T5Chem and base T5) LLM architectures. For encoder-only, targeting the top layers consistently outperforms bottom-layer finetuning. This aligns with the well-established view that deeper layers in such models encode more abstract and domain-specific representations. Importantly, we omit bottom-layer results for decoder-only models (e.g., Qwen2-7B) due to instability and numerical issues encountered during training, likely caused by incompatible LoRA insertions in early layers, and only show results of the different ratio of LoRA adapted top layers (10% and 25%).

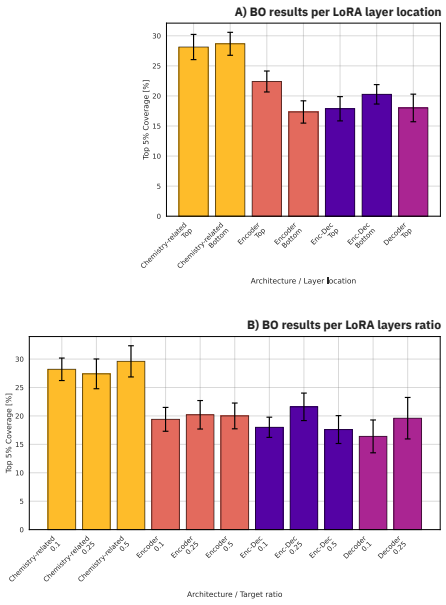

Figure 8: Breakdown of results from Figure 7, highlighting the effect of LoRA layer location (top vs. bottom) and proportion (10%, 25%, 50%) on BO performance. Targeting bottom layers in decoder-only models (e.g., Qwen2-7B) resulted in numerical instabilities during optimization while higher proportion (50%) resulted in out-of-memory issues.

Interestingly, encoder-decoder models (both T5 and chemistry-specialized T5Chem) show competitive performance even when targeting bottom layers, though the top 25% for T5 still yields slightly better or equally stable results. This suggests that meaningful information may be distributed across layers in encoder-decoder setups, potentially due to the dual role of encoding and decoding steps. For T5Chem, targeting the bottom 50% of linear layers with LoRA yield the best results in the ablation (Figure 7). This observation could potentially justify targeting all layers with LoRA in chemistry-related architectures which

could contribute to even better results. Nevertheless, to ensure a consistent and generalizable comparison across LLM sizes and types, we adopt a default strategy of targeting the top 25% of linear layers for all models. This decision achieves a balance between performance and computational efficiency, while also avoiding the overhead of architecture-specific tuning.

E.3 Structure vs Fit and BO Performance

Selecting suitable priors is one of the core challenges in BO, especially when limited information is available about the underlying objective function. The first modeling choice in BO is how to represent the input design space. In chemistry, this space can be expressed in various ways, such as SMILES strings, reaction templates, or molecular fingerprints. Often, this representation is predetermined by the constraints of the problem setting, such as one-hot encodings in combinatorial screens. However, the choice of representation imposes downstream consequences on other components of the BO pipeline.

Surrogate models, which map the input $\mathbf{x}$ to the output y in a probabilistic manner, come with their own inductive biases. GPs rely on a kernel function to define similarity between points. The choice of kernel encodes assumptions about smoothness, differentiability, and the geometry of the function to be modeled. For example, the Tanimoto kernel (Griffiths et al., 2023) might be better suited for binary fingerprint inputs, while Matérn kernels are broadly applicable to continuous Euclidean representations. In our study, we fix the surrogate kernel to Matérn 5/2 due to its balance between smoothness and flexibility, and its prevalence in chemical BO applications. All related analysis in this section is, therefore, built on the basis of this kernel. Future work may explore kernel-specific behaviors in structure/fit alignment. With the selected design space representation and the surrogate kernel, we are left with a set of assumptions about the structure of the objective function itself. If these assumptions are misspecified – e.g., if the representation induces a geometry not aligned with the kernel – then BO performance can degrade.

We now analyze how the choice of data representation impacts BO performance under a fixed surrogate model and acquisition function. To do so, we introduce a single normalized smoothness metric: the ratio between the GP's learned lengthscale and the average pairwise distance in the embedding space. This metric reflects how far the GP generalizes relative to the data distribution, serving as a proxy for the compatibility between the representation space and the kernel's inductive bias.

We observe a strong correlation ($r = 0.92$) between this normalized lengthscale and BO performance (Figure 2), suggesting that representations that allow the GP to maintain broader, smoother fits tend to support more successful optimization. A higher ratio indicates that the GP can generalize over broader regions while still resolving performance differences, ultimately leading to better acquisition decisions. This trend is consistent with the intuition that smoother fits – enabled by coherent, well-structured representation spaces – support more principled exploration and reduce overfitting to local noise.

We further show that while standard and weighted R^2 measures correlate with BO performance, their predictive power is consistently lower than the normalized smoothness metric. This supports the view that while accurate fit helps, smooth fits – enabled by representations that align well with the GP kernel – are even more important for effective acquisition. This finding motivates our proposed deep kernel learning approach. By jointly training the LLM and GP via marginal likelihood, we allow the representation to adapt to the GP's inductive assumptions, resulting in smoother surrogate fits and more structured latent spaces – as shown in Figure 10. This ultimately enables better optimization.

We also observe that the best-performing fixed-feature baseline (DRFP) already exhibits relatively structured embedding space, reflected in clear pairwise L2 and kernel similarity histograms. In contrast, the fixed T5 model produces less organized latent structure. With our adaptive method (PLLMϕ+T5), however, the latent space becomes substantially more structured, resulting in smoother GP fits and better-aligned similarity distributions. This confirms the key role of aligning learned representations with the GP kernel's inductive bias.

[1]* Targeting bottom layers in decoder-only models (Qwen2-7B) led to numerical instabilities in optimization.

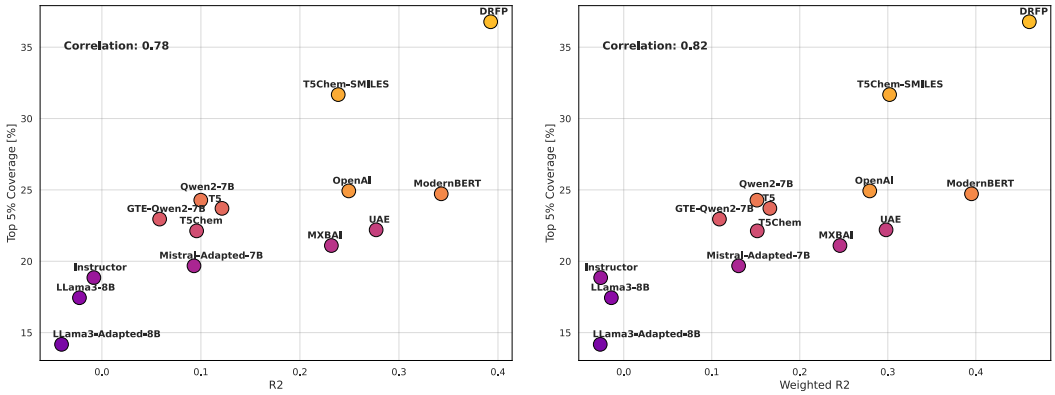

Figure 9: We compare (left) R^2 over the entire design space, (right) weighted R^2 that upweights points in the top 5% (using a 3:1 weighting scheme). While fit alone is informative ($r = 0.78$ unweighted, $r = 0.82$ weighted), the smoothness metric based on normalized lengthscale achieves a stronger correlation with BO success ($r = 0.92$, main text Figure 2), highlighting the importance of representation-structure alignment.

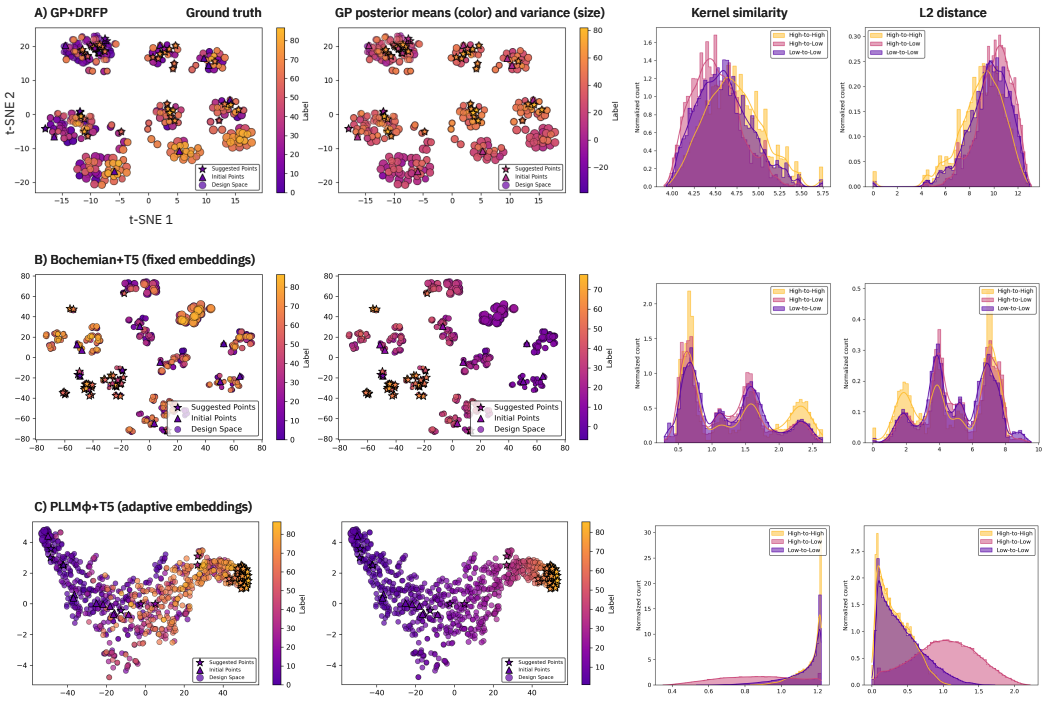

Figure 10: **Visual analysis of latent space structure, GP behavior, and similarity metrics across different representations.** We compare three models: (A) GP+DRFP, the best-performing fixed-feature baseline; (B) BoChemian+T5, using frozen LLM embeddings from natural language templates; and (C) PLLMϕ+T5, our best adaptive embedding model. The left two columns visualize the latent space with ground truth (left) and GP posterior mean/variance (middle) colored by yield. We mark the suggested and initial points. The right two columns show pairwise kernel similarities and L2 distances for high–high, high–low, and low–low yielding regions. DRFP exhibits mild structural organization even in its fixed feature space, contributing to strong performance. T5 without finetuning lacks this structure, while PLLMϕ+T5 learns a highly structured latent space, enabling smoother fits and more effective acquisition decisions.

E.4 BO Results per Representation Type

With the BO traces in Figure 11 we show aggregated results per LLM or chemistry-related representation types. Additionally, we provide an overview of the BO performance for each individual representation (LLM or chemistry related) during the 50 optimization steps in Figure 12. We observe that performance varies in different BH reactions, with no representation consistently outperforming others across all tasks – including chemistry-specialized ones. All LLM types, however, show similar distributions of suggested point evaluations. In comparison to other LLM architectures, encoder-based models tend to achieve higher R^2 values during optimization. However, the improved function approximation does not necessarily translate to better BO performance, as modeling the function and identifying its optimal points are two fundamentally distinct, though complementary, objectives.

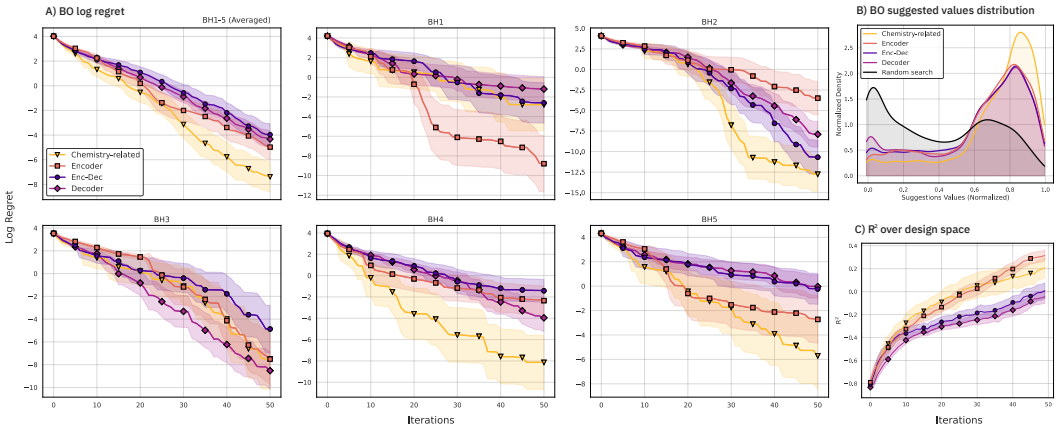

Figure 11: **BO metrics per LLM type.** A) We show optimization paths for all BH reactions (BH1-5 Averaged) across different LLM types (Encoder only, Encoder-Decoder, Decoder only) and chemistry-related representations (DRFP, T5Chem-SMILES) together with optimization results on individual reactions (BH1-BH5). B) Distribution of evaluated suggestions generated throughout the entire optimization process (50 iterations) for 20 seed runs and all BH reactions. C) R^2 scores per LLM type over the evolving design space, averaged across all BH reactions.

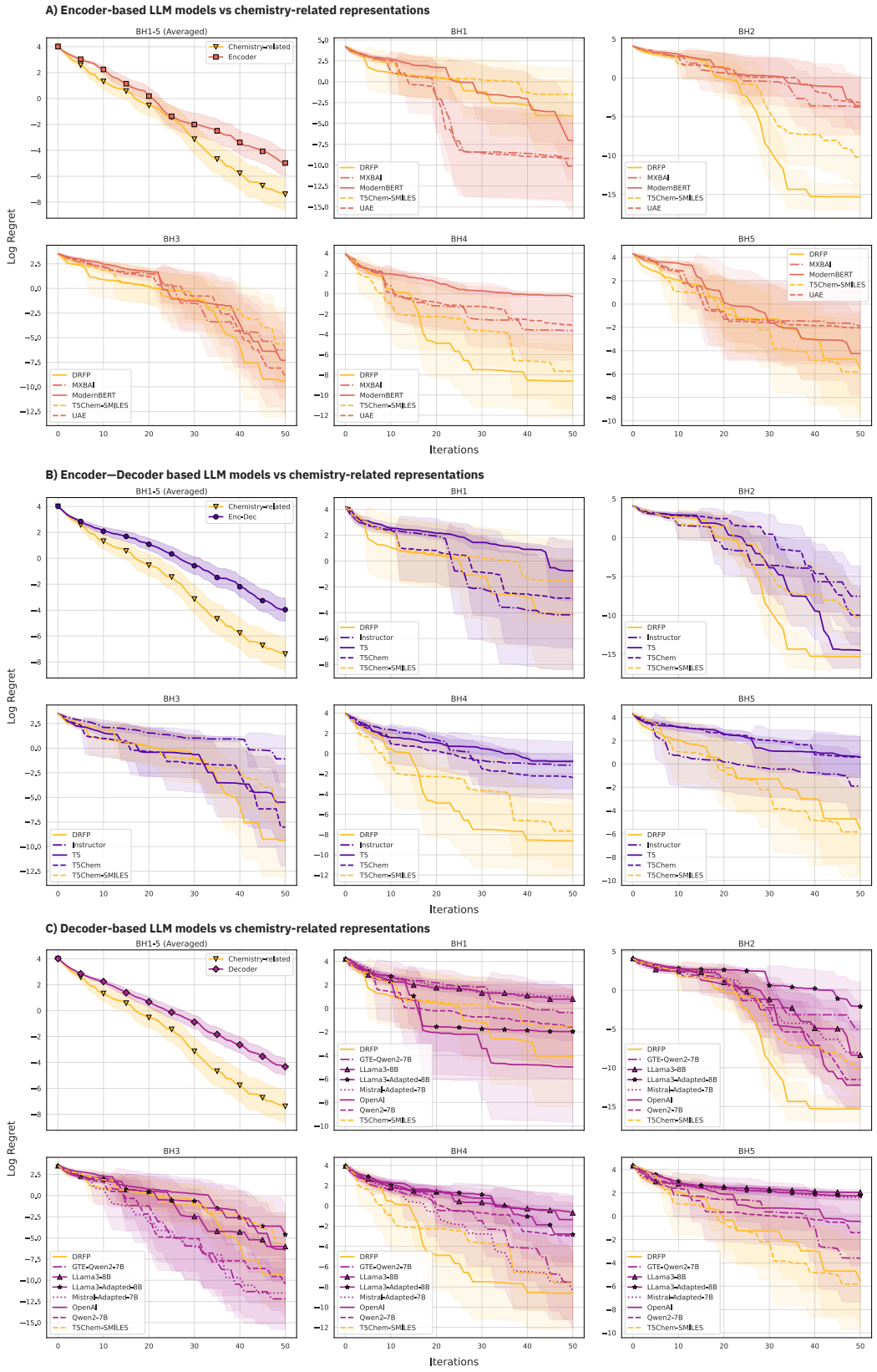

Figure 12: Individual LLM BO tracelines in Buchwald-Hartwig optimization.

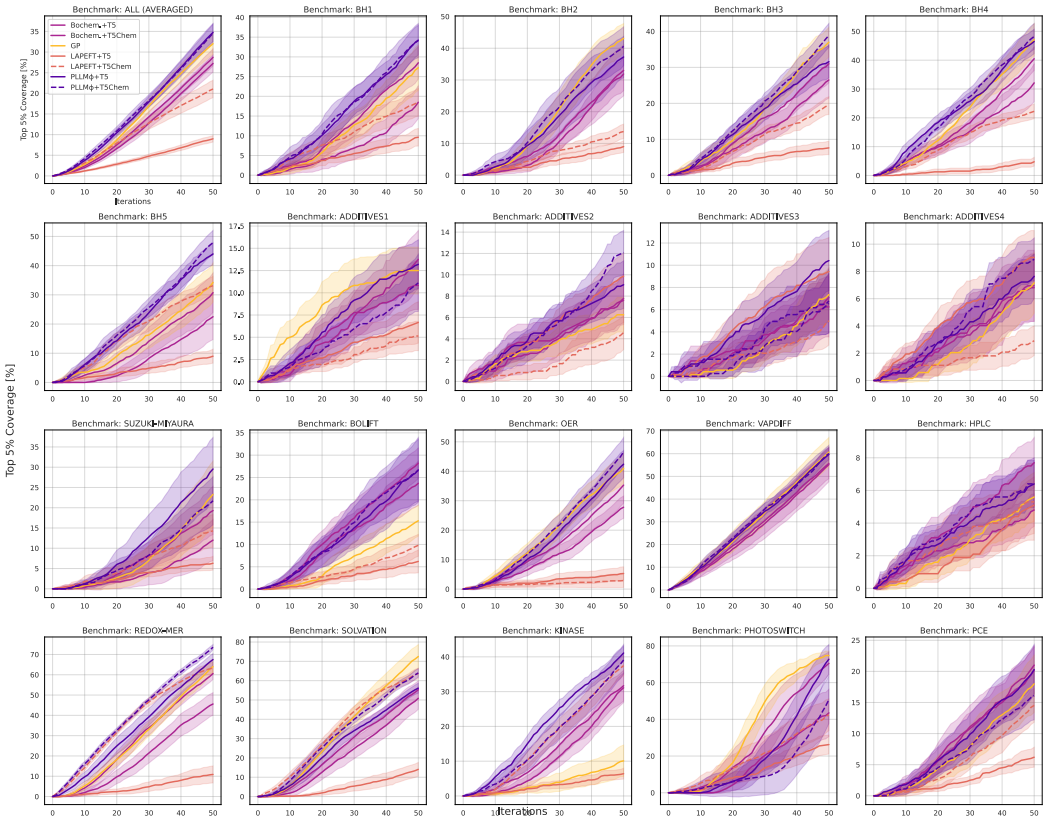

Figure 13: Top 5% coverage per iteration for all benchmarks.

F REPRODUCIBILITY

F.1 BO INITIALIZATION

Bayesian optimization is in practice initialized with a small number of pre-existing datapoints, either from prior experimentation or simulation. In reaction optimization, for example, these initial points often correspond to unsuccessful or low-yielding reactions. In scientific discovery settings, this setup is not only realistic but expected – optimization typically begins from sparse and suboptimal conditions, with the goal of efficiently identifying high-performing regions.

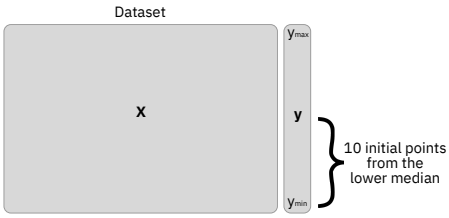

Figure 14: **Illustrative example of initial data selection.** For all benchmark datasets, we constrain the selection of initial points for BO to the lower median points based on the objective values. This initialization strategy better reflects real-world scenarios where optimization starts from suboptimal conditions.

This starting point contrasts with scenarios where good conditions are already known, in which case optimization reduces to local exploitation rather than global search. Reaching high-yielding conditions from poor initial data is substantially more challenging and better reflects real-world discovery pipelines. Compounding this difficulty is the negative bias in the scientific literature: failed or low-yield experiments are rarely reported, making published datasets inherently skewed toward successful outcomes (Ioannidis, 2005).

To simulate this setting, we initialize the BO algorithm (Algorithm 1) with 10 points sampled from the lower median range of the objective value distribution. Figure 14 illustrates this strategy. For each benchmark dataset, the candidate pool is first sorted by objective value, and initial points are randomly selected from those with values below the median. This initialization requires the model to reason under uncertainty and efficiently navigate toward high-performing regions with minimal prior knowledge.

F.2 BO SETUP

Following the initialization, we run the BO loop for 50 iterations with batch size 1. We repeat each experiment configuration for 20 times with different seed values (1-20) to obtain robust performance metrics. Our choice for the GP kernel is Matérn-5/2, based on its demonstrated effectiveness on both continuous and discrete design spaces (Ranković et al., 2024). For balancing the exploration and exploitation we employ the expected improvement acquisition function.

F.3 IMPLEMENTATION DETAILS

Surrogate Models. We implement both fixed-feature and finetuned surrogate models as subclasses of `SingleTaskGP` from `botorch` (Balandat et al., 2020). For fixed-feature GPs, the inputs are LLM embeddings, chemistry-related representations (fingerprints, DRFP) or default parameters (one-hot encoded categorical variables, numerical values), while we learn the GP kernel hyperparameters $\theta = \{\ell, \sigma^2, \sigma_n^2, c\}$ by maximizing the marginal likelihood using the L-BFGS-B optimizer provided by BoTorch's `fit_gpytorch_mll` routine. We employ the Matérn-5/2 kernel (Gardner et al., 2018) with initialization: $\ell = 1.0$, $\sigma^2 = 1.0$, and $\sigma_n^2 = 1.0 \times 10^{-4}$. The optimizer runs with multiple restarts, as part of BoTorch's default behavior.

For deep kernel GPs, we extend the surrogate to jointly optimize both GP and LLM parameters. We build a custom `DeepGP` class that incorporates a finetuning model (PEFT adapter and/or projection head) jointly trained via AdamW (Loshchilov & Hutter, 2017). We optimize GP and LLM parameters using separate learning rates (2×10^{-1} for GP, 2×10^{-3} for LLM) with a shared weight decay of 1×10^{-3}. We apply gradient clipping with a max norm of 1.0 and decay the learning rates using a StepLR scheduler with a decay factor of 0.95.

PEFT Configuration. We insert LoRA adapters into the top 25% of linear layers, using the following configuration: rank $r = 4$, $\alpha = 16$, no bias updates and dropout of 0.2.

Projection Layer. We define the projection layer as:

$$\mathbf{z} = \text{ELU}(\text{Dropout}(W\mathbf{x} + b)), \tag{10}$$

where $W \in \mathbb{R}^{d \times 64}$ and input d is model-dependent. We initialize the parameters of this model via Xavier uniform initialization.

Featurization. We extract LLM embeddings from Hugging Face models using their default tokenizers and truncating the input to maximum of 512 tokens. Pooling strategies depend on the model architecture: CLS token for encoder-based models, last-token for decoder-only models, and mean-pooling for encoder-decoder models.

Training Setup. Fixed-feature models can be run on CPU or a single GPU (e.g., RTX 3090 with 24GB VRAM). Finetuned experiments involving large LLMs (e.g., PLLMϕ+Qwen2-7B) are run on NVIDIA H100 GPU and 96GB RAM. Lightweight models (e.g., PLLMϕ+T5, PLLMϕ+ModernBERT) are trainable on local hardware, with under 300k trainable parameters in total (e.g., 230k out of 149M for ModernBERT, 165k out of 109M for T5; less than 0.2% of model weights updated).

Tracking and Seeding. All experiments are seeded using `seed_everything` from `pytorch_lighting` (Falcon & The PyTorch Lightning team, 2019) with values 1-20. Acquisition optimization is deterministic over the candidate pool. We do not augment the data or use any stochastic featurization. We use `wandb` (Biewald, 2020) for tracking experiments, managing seeds, logging losses, metrics, learning rates, images, and running configuration sweeps.

G Supporting Tables

Representation	Top 5% Coverage [%]
DRFP	**37.750 ± 13.983**
T5Chem-SMILES	32.750 ± 13.728
OpenAI	25.650 ± 12.287
ModernBERT	25.475 ± 10.061
Qwen2-7B	25.150 ± 13.934
T5	24.450 ± 13.005
GTE-Qwen2-7B	23.500 ± 15.288
T5Chem	22.950 ± 11.571
UAE	22.800 ± 13.144
MXBAI	21.800 ± 14.733
Mistral-Adapted-7B	20.350 ± 13.046
Instructor	19.300 ± 10.935
LLama3-8B	18.056 ± 12.745
LLama3-Adapted-8B	14.625 ± 10.803
Random Search	6.176 ± 3.751

Table 2: BO with fixed LLM features.

Representation	Top 5% Coverage [%]
PLLMϕ+T5	**42.602 ± 13.111**
LLMϕ+T5	42.577 ± 13.817
PLLM+T5	41.075 ± 10.844
PLLMϕ+Qwen2-7B	40.250 ± 11.189

Representation	Top 5% Coverage [%]
PLLM+Qwen2-7B	39.375 ± 9.850
LLMϕ+Qwen2-7B	38.725 ± 12.780
PLLMϕ+ModernBERT	36.250 ± 14.100
LLMϕ+ModernBERT	35.775 ± 13.618
PLLM+ModernBERT	34.600 ± 10.805
PLLM+OpenAI	33.525 ± 11.791
OpenAI	25.650 ± 12.287
ModernBERT	25.475 ± 10.061
Qwen2-7B	25.150 ± 13.934
T5	24.450 ± 13.005

Table 3: BO results with LLM-based deep kernels.

Benchmark	Method	$R^2 \uparrow$	$NLPD \downarrow$
ADDITIVES-1	PLLMϕ+T5 (ours)	-0.00 ± 0.10	21.19 ± 2.77
	PLLMϕ+T5Chem (ours)	0.07 ± 0.09	27.02 ± 3.93
	GP+DRFP	0.09 ± 0.06	11.34 ± 0.11
	Bochem.+T5	0.14 ± 0.05	11.66 ± 0.92
	Bochem.+T5Chem	0.17 ± 0.09	11.87 ± 1.25
	LAPEFT+T5	-0.15 ± 0.14	723.23 ± 1625.77
	LAPEFT+T5Chem	-0.04 ± 0.13	329.75 ± 566.60
ADDITIVES-2	PLLMϕ+T5 (ours)	-0.22 ± 0.13	25.72 ± 8.98
	PLLMϕ+T5Chem (ours)	-0.11 ± 0.09	36.73 ± 8.29
	GP+DRFP	0.01 ± 0.02	10.00 ± 0.24
	Bochem.+T5	0.02 ± 0.03	10.58 ± 1.56
	Bochem.+T5Chem	0.05 ± 0.03	10.21 ± 0.67
	LAPEFT+T5	-0.15 ± 0.11	392.67 ± 516.44
	LAPEFT+T5Chem	-0.12 ± 0.10	292.60 ± 346.04
ADDITIVES-3	PLLMϕ+T5 (ours)	-0.30 ± 0.31	33.75 ± 15.29
	PLLMϕ+T5Chem (ours)	-0.19 ± 0.21	52.34 ± 28.62
	GP+DRFP	-0.01 ± 0.02	10.36 ± 0.60
	Bochem.+T5	-0.01 ± 0.02	10.50 ± 0.80
	Bochem.+T5Chem	-0.00 ± 0.02	10.48 ± 0.92
	LAPEFT+T5	-0.21 ± 0.16	641.89 ± 1356.60
	LAPEFT+T5Chem	-0.12 ± 0.12	386.27 ± 431.61
ADDITIVES-4	PLLMϕ+T5 (ours)	-0.34 ± 0.23	22.36 ± 4.87
	PLLMϕ+T5Chem (ours)	-0.21 ± 0.14	28.60 ± 9.03
	GP+DRFP	0.01 ± 0.03	10.27 ± 0.13
	Bochem.+T5	0.01 ± 0.05	11.12 ± 2.58
	Bochem.+T5Chem	0.00 ± 0.05	10.83 ± 1.41
	LAPEFT+T5	-0.22 ± 0.15	186.94 ± 177.56
	LAPEFT+T5Chem	-0.14 ± 0.15	389.52 ± 715.90
BH-1	PLLMϕ+T5 (ours)	0.68 ± 0.05	5.57 ± 0.63
	PLLMϕ+T5Chem (ours)	0.69 ± 0.07	7.11 ± 0.90
	GP+DRFP	0.31 ± 0.34	6.14 ± 7.20
	Bochem.+T5	0.14 ± 0.23	19.96 ± 30.81
	Bochem.+T5Chem	0.52 ± 0.14	32.23 ± 25.51

Benchmark	Method	$R^2 \uparrow$	$NLPD \downarrow$
	LAPEFT+T5	0.51 ± 0.11	65.15 ± 83.74
	LAPEFT+T5Chem	0.60 ± 0.04	216.41 ± 276.61
BH-2	PLLMϕ+T5 (ours)	0.65 ± 0.08	5.96 ± 1.00
	PLLMϕ+T5Chem (ours)	0.65 ± 0.09	7.23 ± 1.17
	GP+DRFP	0.50 ± 0.27	4.06 ± 0.77
	Bochem.+T5	0.22 ± 0.22	60.34 ± 102.47
	Bochem.+T5Chem	0.50 ± 0.08	64.15 ± 58.29
	LAPEFT+T5	0.32 ± 0.18	235.69 ± 415.15
	LAPEFT+T5Chem	0.53 ± 0.06	249.48 ± 192.73
BH-3	PLLMϕ+T5 (ours)	0.47 ± 0.13	6.69 ± 1.28
	PLLMϕ+T5Chem (ours)	0.50 ± 0.10	8.34 ± 1.58
	GP+DRFP	0.38 ± 0.30	5.72 ± 8.59
	Bochem.+T5	0.16 ± 0.14	22.18 ± 47.62
	Bochem.+T5Chem	0.41 ± 0.13	50.64 ± 99.46
	LAPEFT+T5	0.27 ± 0.10	71.06 ± 53.91
	LAPEFT+T5Chem	0.40 ± 0.07	350.63 ± 528.16
BH-4	PLLMϕ+T5 (ours)	0.57 ± 0.08	7.86 ± 1.15
	PLLMϕ+T5Chem (ours)	0.59 ± 0.12	8.72 ± 1.47
	GP+DRFP	0.48 ± 0.29	4.13 ± 0.57
	Bochem.+T5	0.46 ± 0.15	39.56 ± 43.63
	Bochem.+T5Chem	0.46 ± 0.13	41.01 ± 54.16
	LAPEFT+T5	0.32 ± 0.09	80.44 ± 89.41
	LAPEFT+T5Chem	0.40 ± 0.08	581.34 ± 2146.82
BH-5	PLLMϕ+T5 (ours)	0.50 ± 0.10	8.48 ± 1.42
	PLLMϕ+T5Chem (ours)	0.53 ± 0.08	9.08 ± 1.48
	GP+DRFP	0.42 ± 0.28	4.31 ± 0.59
	Bochem.+T5	0.33 ± 0.17	49.31 ± 62.92
	Bochem.+T5Chem	0.43 ± 0.10	57.96 ± 51.71
	LAPEFT+T5	0.37 ± 0.11	63.98 ± 44.88
	LAPEFT+T5Chem	0.42 ± 0.10	391.56 ± 543.81
C2-YIELD	PLLMϕ+T5 (ours)	0.44 ± 0.09	4.83 ± 2.30
	PLLMϕ+T5Chem (ours)	0.42 ± 0.12	4.56 ± 1.73
	GP+Num.Params	0.16 ± 0.09	154792.41 ± 578478.07
	Bochem.+T5	0.41 ± 0.11	0.99 ± 0.94
	Bochem.+T5Chem	0.42 ± 0.14	5.57 ± 18.91
	LAPEFT+T5	0.29 ± 0.09	180.61 ± 220.08
	LAPEFT+T5Chem	0.24 ± 0.09	1196.45 ± 1636.29
HPLC	PLLMϕ+T5 (ours)	-0.13 ± 0.07	20.13 ± 5.14
	PLLMϕ+T5Chem (ours)	-0.21 ± 0.07	20.18 ± 5.66
	GP+Num.Params	-0.04 ± 0.05	1444.58 ± 6420.47
	Bochem.+T5	-0.03 ± 0.03	18.57 ± 14.46
	Bochem.+T5Chem	-0.02 ± 0.03	49.10 ± 66.46
	LAPEFT+T5	-0.22 ± 0.14	229.99 ± 255.04
	LAPEFT+T5Chem	-0.18 ± 0.09	2427.96 ± 8532.05
KINASE	PLLMϕ+T5 (ours)	0.35 ± 0.06	14.55 ± 2.16
	PLLMϕ+T5Chem (ours)	0.41 ± 0.05	12.82 ± 2.09
	GP+FP	-0.02 ± 0.03	1.41 ± 0.11
	Bochem.+T5	0.02 ± 0.11	3.44 ± 6.44

Benchmark	Method	$R^2 \uparrow$	$NLPD \downarrow$
	Bochem.+T5Chem	-0.02 ± 0.02	1.67 ± 0.11
	LAPEFT+T5	0.33 ± 0.07	12.42 ± 3.44
	LAPEFT+T5Chem	0.45 ± 0.04	21.59 ± 5.58
OER	PLLMϕ+T5 (ours)	0.43 ± 0.15	1.95 ± 1.61
	PLLMϕ+T5Chem (ours)	0.42 ± 0.15	0.88 ± 1.31
	GP+Num.Params	0.59 ± 0.04	189.97 ± 218.23
	Bochem.+T5	0.44 ± 0.09	17.26 ± 23.03
	Bochem.+T5Chem	0.44 ± 0.11	154.70 ± 206.00
	LAPEFT+T5	0.47 ± 0.08	27.27 ± 34.99
	LAPEFT+T5Chem	0.47 ± 0.07	174.81 ± 286.01
PCE	PLLMϕ+T5 (ours)	0.25 ± 0.12	16.12 ± 4.78
	PLLMϕ+T5Chem (ours)	0.27 ± 0.11	15.43 ± 4.42
	GP+FP	-0.02 ± 0.02	2.52 ± 0.08
	Bochem.+T5	0.10 ± 0.15	4.51 ± 6.22
	Bochem.+T5Chem	-0.02 ± 0.03	2.57 ± 0.15
	LAPEFT+T5	0.16 ± 0.15	29.69 ± 46.25
	LAPEFT+T5Chem	0.26 ± 0.13	41.33 ± 21.10
PHOTOSWITCH	PLLMϕ+T5 (ours)	0.59 ± 0.12	11.53 ± 2.13
	PLLMϕ+T5Chem (ours)	0.65 ± 0.10	9.86 ± 1.62
	GP+FP	0.57 ± 0.26	5.13 ± 0.58
	Bochem.+T5	0.58 ± 0.07	41238535.40 ± 32024292.47
	Bochem.+T5Chem	0.46 ± 0.28	151181279.47 ± 202238451.62
	LAPEFT+T5	0.52 ± 0.08	7.78 ± 1.20
	LAPEFT+T5Chem	0.64 ± 0.07	18.50 ± 11.90
REDOX-MER	PLLMϕ+T5 (ours)	0.86 ± 0.02	0.48 ± 0.47
	PLLMϕ+T5Chem (ours)	0.89 ± 0.02	-0.02 ± 0.35
	GP+FP	0.73 ± 0.25	-0.51 ± 0.44
	Bochem.+T5	0.85 ± 0.02	19.72 ± 16.26
	Bochem.+T5Chem	0.85 ± 0.20	47.01 ± 40.57
	LAPEFT+T5	0.73 ± 0.06	6.45 ± 2.53
	LAPEFT+T5Chem	0.86 ± 0.03	16.55 ± 6.20
SOLVATION	PLLMϕ+T5 (ours)	0.74 ± 0.03	2.84 ± 0.98
	PLLMϕ+T5Chem (ours)	0.73 ± 0.04	2.86 ± 0.97
	GP+FP	0.77 ± 0.03	-0.36 ± 0.47
	Bochem.+T5	0.79 ± 0.03	77.92 ± 58.50
	Bochem.+T5Chem	0.69 ± 0.30	174.24 ± 192.44
	LAPEFT+T5	0.71 ± 0.05	14.28 ± 20.01
	LAPEFT+T5Chem	0.79 ± 0.02	73.66 ± 38.83
SUZUKI-MIYAURA	PLLMϕ+T5 (ours)	0.10 ± 0.13	8.16 ± 1.45
	PLLMϕ+T5Chem (ours)	0.12 ± 0.13	9.65 ± 2.37
	GP+DRFP	0.29 ± 0.15	1.20 ± 2.57
	Bochem.+T5	0.07 ± 0.08	8.76 ± 28.10
	Bochem.+T5Chem	0.12 ± 0.10	1.11 ± 3.25
	LAPEFT+T5	-0.10 ± 0.18	107.12 ± 105.69
	LAPEFT+T5Chem	0.18 ± 0.09	231.35 ± 181.75
VAPDIFF	PLLMϕ+T5 (ours)	-0.01 ± 0.06	12.40 ± 2.73
	PLLMϕ+T5Chem (ours)	-0.05 ± 0.09	14.25 ± 3.41
	GP+Num.Params	0.12 ± 0.06	608876.27 ± 906151.79

Benchmark	Method	$R^2 \uparrow$	$NLPD \downarrow$
	Bochem.+T5	0.07 $\pm$ 0.06	1.69 $\pm$ 0.28
	Bochem.+T5Chem	0.08 $\pm$ 0.07	1.72 $\pm$ 0.46
	LAPEFT+T5	-0.17 $\pm$ 0.11	288.12 $\pm$ 284.60
	LAPEFT+T5Chem	-0.04 $\pm$ 0.08	164.06 $\pm$ 142.57

Table 4: Predictive and uncertainty estimates for all benchmark datasets and methods. Each model is trained on 60 points and evaluated on the remaining data, emulating a 10+50 BO iteration setup. This fixed train/validation split ensures fair comparison by avoiding divergence in design sets during BO due to different selection of candidate points during optimization, even when starting with the same initial points. We run 20 repeats and report mean and standar deviation values.

Arch.	Pool	Model	Quant. 95 [cnt]
Enc	CLS	MXBAI	2.70 $\pm$ 2.15
		ModernBERT	2.40 $\pm$ 2.09
		UAE	2.40 $\pm$ 2.11
	Avg	MXBAI	2.30 $\pm$ 2.00
		ModernBERT	2.30 $\pm$ 1.72
		UAE	2.25 $\pm$ 1.86
	Last	MXBAI	1.80 $\pm$ 1.82
		ModernBERT	0.25 $\pm$ 0.55
		UAE	2.50 $\pm$ 2.35
Dec	Avg	LLama3-8B	0.75 $\pm$ 1.48
		Qwen2-7B	1.05 $\pm$ 2.28
	Last	LLama3-8B	1.55 $\pm$ 1.64
		Qwen2-7B	1.65 $\pm$ 2.54
Enc-Dec	Avg	T5	1.10 $\pm$ 1.41
		T5Chem	1.90 $\pm$ 2.05
	Last	T5	1.65 $\pm$ 2.48
		T5Chem	1.55 $\pm$ 2.91

Table 5: Tokenization influence to sampling from the 5th percentile.

Architecture	Target layers	Target ratio	Deep Kernel + Model	Quantile 95 [cnt]
Chem-related	Top	0.1	PLLMϕ+T5Chem-SMILES	6.60 ± 1.90
		0.25		6.30 ± 3.47
		0.5		8.20 ± 2.94
	Bottom	0.1		7.50 ± 2.51
		0.25		7.40 ± 2.32
		0.5		6.60 ± 3.13
Encoder	Top	0.1	PLLMϕ+ModernBERT	5.50 ± 2.17
		0.25		5.60 ± 2.37
		0.5		5.70 ± 2.83
	Bottom	0.1		4.20 ± 2.44
		0.25		4.50 ± 3.21
		0.5		4.30 ± 2.11
Enc-Dec	Top	0.1	PLLMϕ+T5	4.10 ± 1.91
		0.25		5.40 ± 2.99
		0.5		3.90 ± 3.25
	Bottom	0.1		4.90 ± 2.08
		0.25		5.40 ± 2.55
		0.5		4.90 ± 2.18
Decoder	Top	0.1	PLLMϕ+Qwen2-7B	4.10 ± 2.28
		0.25		4.90 ± 2.88

Table 6: GP-LLM finetuning per LLM types and LoRA layers