Contextual Bayesian Optimisation with Large Language Models via In-Context Learning

Siddartha Nath [*] [1] **Shyam Sundhar Ramesh** [1] **Ilija Bogunovic** [1]

Abstract

Contextual Bayesian Optimization (CBO) improves Bayesian Optimization (BO) by incorporating context, expanding its usability. Crucially, it strikes a balance between exploration and exploitation in complex scenarios. While Large Language Models (LLMs) excel in contextual understanding, their use in CBO is uncharted. We introduce the **C**ontext **A**ware **L**arge **L**anguage **S**ystem (CALLS) and the novel CLLM-UCB algorithm, utilizing an LLM as a surrogate model with a Contextual Upper Confidence Bound as the acquisition function. Designed to assess LLMs in optimization, our scalable CALLS framework requires no fine-tuning. Empirically validated in Reinforcement Learning's Multi-Armed-Problem for molecular optimization, our results show that an LLM within a CBO framework consistently outperforms their counterpart in context-absent BO setup, achieving sublinear contextual regret.

1. Introduction

Bayesian optimization (BO) is a powerful approach for optimising *black-box* functions i.e., unknown functions that are computationally expensive to evaluate. This phenomena appears across many different disciplines, including robotic design, vaccine design and engineering design (Calandra et al., 2016; Rosa et al., 2022; Do & Zhang, 2023). Broadly speaking, BO incorporates a *surrogate model* and *acquisition* function. The surrogate model serves as a proxy for the unknown function. It is first trained on an initial dataset via a warm up process. The remaining set of points to query are placed into a pool and the acquisition function is then used to propose the next point from this pool, which is then selected and evaluated by the surrogate model. The surrogate model is iteratively updated with these selected points, until a desired criteria is met.

A key architectural limitation with BO is that when additional information is available and desired to be used in decision-making, it is not possible to leverage this within the current setup and thus, BO is extended to Contextual Bayesian optimization (CBO). In this setting, prior to proposing a point from the pool, the environment provides a context and given this constraint, the surrogate model must select a point. This framework is analogous to Contextual Bandits (CB) in Reinforcement Learning (RL) (Krause & Ong, 2011). In the realm of CBO, the reward entails evaluating the surrogate model's performance at each time-step, contingent upon the observed context preceding the action selection. Subsequently, this action contributes to the quantification of loss, commonly referred to as *regret*, which is the difference in the true function value, between a given context-action pair and the optimal context-action pair, at time t. This can then be used to calculate various *objective metrics* e.g., average regret, cumulative regret and minmax regret.

In this paper, we seek to understand whether LLMs have any capacity of qualifying as a reliable surrogate model in CBO and not necessarily whether they outperform existing surrogate models. We formalise this through our architecture CALLS, where we build an end-to-end CBO pipeline, incorporating sophisticated prompt engineering techniques, namely constrained dynamic role-playing and ICL via maximal marginal relevance (MMR). The foundational backbone of CALLS is the proposal of a novel algorithm, CLLM-UCB, which facilitates the use of an LLM as a surrogate model and a Contextual Upper Confidence Bound (C-UCB) as an acquisition function. We evaluate this method on a molecular optimization task and find that an LLM within a CBO framework consistently outperforms their counterpart in context-absent BO setup, by achieving sublinear contextual regret.

2. Related Work

In the realm of optimization, our research aligns with several established algorithms, such as Bayesian optimization

[*] Equal contribution [1] Department of Electronic and Electric Engineering, University of College London, London, United Kingdom. Correspondence to: Siddartha Nath <ucabsn4@ucl.ac.uk>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

(BO) (Kirschner et al., 2020), Contextual Bayesian optimization (CBO) (Shyam Sundhar Ramesh & Bogunovic, 2022; Char et al., 2019), and Contextual Bandit (CB) optimization (Krause & Ong, 2011). Significant strides have been made in extending BO to CBO, providing thorough regret bound analyses and empirical validations, yet these developments have largely focused on traditional surrogate models like Gaussian Processes (GPs) or Neural Networks (NNs). Concurrently, the incorporation of Large Language Models (LLMs) into these optimization frameworks has started to attract attention. Notable efforts include research on prompt optimization (Chengrun Yang & XinyunChen, 2023; LLM, 2024a) and In-Context Learning (ICL) in BO with GPs and NNs (PFN, 2023), alongside innovative applications such as (Ramos et al., 2023; Tennison Liu & van der Schaar, 2024) that integrate LLMs into BO's protocol in varying capacities.

Regarding both setups, it is crucial to obtain a protocol which efficiently finds points that best describe the unknown function, whilst needing limited observations to train on. This ultimately comes down to the surrogate model, acquisition function and their synergy. Traditionally, either Gaussian Process (GP) or Neural Network (NN) have been used as surrogate models, with Thompson Sampling (TS), Upper Confidence Bound (UCB) or Expected Improvement (EI) as acquisition functions, with many success in different combinations of these (Frazier, 2018). However, there is a significant obstacle in their use - they encounter scalability issues; GPs struggle with computational intensiveness for large datasets, and NNs often require extensive data to learn effectively. This scenario is typically framed as the *few-shot* paradigm, where there is a necessity for rapid adaptation and generalization based on a minimal number of examples (Wang et al., 2020). Interestingly, such challenges of the few-shot paradigm align with the proficiencies of Large Language Models (LLMs). Modern LLMs such as GPT4 and Claude 3 (OpenAI, 2024; Anthropic, 2024), which have undergone pre-training on vast internet-scale datasets, demonstrate a notable ability to generalize from limited data. This ability enables their strong performance in tasks requiring zero to few-shot learning, such as prediction and content generation (Takeshi Kojima & Iwasawa, 2023; Brown et al., 2020; Wei et al., 2022b), and in grasping contextual nuances (Wei et al., 2022a; Yilun Zhu & Tseng, 2024). This has been further strengthened through training mechanisms such as *prompt engineering* (PE) (Amatriain, 2024; Pranab Sahoo & AmanChadha, 2024). The main techniques within PE include *role-playing* (Banghao Chen & Zhu, 2023), which involves designing precise instructions to guide LLMs and, *in-context learning* (ICL) (Aaron Mueller & Linzen, 2023; Qingxiu Dong & Sui, 2023), which involves designing templates containing examples that are contextually related for a specified task. Despite these hypothesized advantages of LLMs, effectively capitalizing on them in an iterative optimization framework like BO or CBO is challenging.

Our study is particularly inspired by (Krause & Ong, 2011) and (Ramos et al., 2023); the former introduces context into the acquisition function alongside GPs, forming the basis for CGP-UCB, while the latter pioneers the use of LLMs as surrogate models in BO with the integration of ICL. Melding these concepts, we extend their application into the CBO domain, incorporating prompt engineering to enhance the LLM's performance. As such, our work is poised to be a pioneering endeavor in applying LLMs within a contextual optimization framework.

3. Problem Statement

3.1. Mathematical Formulation

Let $f : \mathcal{C} \times \mathcal{A} \to \mathbb{R}$ be an unknown function, where $\mathcal{C}$ represents a finite convex and compact space of contexts i.e., $\mathcal{C} \subset \mathbb{R}$ and $\mathcal{A}$ is a finite set of actions i.e., $|\mathcal{A}| = n$. For each step t, a context $\mathbf{c}_t \in \mathcal{C}$ is uniformly sampled from the environment and observed. Given $\mathbf{c}_t$, the most suitable action $\mathbf{a}_t \in \mathcal{A}$ is selected from the data $\mathcal{D}$, determined by the surrogate model $\mathcal{M}$, which is abstractly defined to capture the unknown dynamics of f, and acquisition function C, which determines the level of exploration and exploitation of $\mathcal{A}$. The environment then establishes a reward, which is modeled as a realisation of f,

$$r_t = f\left(\mathbf{c}_t, \mathbf{a}_t\right) + \epsilon_t \in \left[0, f\left(\mathbf{c}_t, \mathbf{a}_t^*\right)\right], \tag{1}$$

where $\epsilon_t \sim \mathcal{N}\left(0, \sigma^2\right)$ represents zero-mean stochastic noise. This reward is in turn used to calculate the loss function known as *regret*,

$$\mathcal{L}_t = f\left(\mathbf{c}_t, \mathbf{a}_t^*\right) - r_t \in \left[0, f\left(\mathbf{c}_t, \mathbf{a}_t^*\right)\right], \tag{2}$$

where $\mathbf{a}_t^*$ represents the optimal action that maximizes the expected reward for a given context $\mathbf{c}_t$. The chosen $(\mathbf{c}_t, \mathbf{a}_t)$ is augmented to the pool $\mathcal{P}$ of already chosen context-action pairs, and the procedure continues iteratively till a convergence criterion is met. The context-specific best action is a more demanding benchmark than the best action used in the (context-free) definition regret. The final performance evaluation is calculated through the expected *cumulative regret*:

$$R_T = \sum_{t=1}^{T} \mathbb{E}_{\mathcal{M}}\left[\mathcal{L}_t\right] = \sum_{t=1}^{T} \mathbb{E}_{\mathcal{M}}\left[f\left(\mathbf{c}_t, \mathbf{a}_t^*\right) - f\left(\mathbf{c}_t, \mathbf{a}_t\right)\right]$$

The ultimate goal is to optimize the a priori f - thus, by carefully selecting $\mathcal{M}$, we seek a protocol P, whose cumulative contextual regret grows sublinearly in T,

$$\lim_{T\to\infty} \frac{R_T}{T} = 0 \equiv O\left(P\right) << O\left(T\right) \tag{3}$$

This framework generalises the multi-armed bandit settting and ultimately combines CB and BO to give CBO.

3.2. Mathematical Assumptions

Function Structure. In a traditional BO and CBO setup, $f : \mathcal{C} \times \mathcal{A} \rightarrow \mathbb{R}$ is typically assumed to be sampled from a GP distribution and hence many approaches use GPs as the surrogate model. In our setup, we do not make this assumption - instead, we allow f to be any unknown finite-space function. This makes modelling f more difficult but we trade this off with the ability to appropriately test whether LLMs are simply autoregressive models or if they can act as function approximators.

Single Context Selection. Regarding the context, we have that,

$$\forall t \in \mathbb{R}, \quad \mathbf{c}_t \in \mathbb{R} \Rightarrow \dim\left(\mathbf{c} = \mathbf{c_t}\right) = 1.$$

This means that $|\mathcal{C}| = m$. Ideally, we should use the notation c instead of $\mathbf{c}$, but we stick to the conventions outlined in literature.

Distinct Action Maximizers. Regarding the actions across all contexts,

$$\text{If } \exists \mathbf{a}^* \in \mathcal{A} : \forall \mathbf{c} \in \mathcal{C}, \quad \mathbf{a}^* = \arg\max_{\mathbf{a} \in \mathcal{A}} f\left(\mathbf{c}, \mathbf{a}\right),$$

then it will be difficult to assess whether the context plays a role in the optimization process as the protocol can simply always sample $\mathbf{a}^*$ i.e. the global maximum, which essentially downgrades CBO to BO. Hence, we aim to make sure that,

$$\forall i, j \in \{1, 2, ..., m\}, i \not= j :$$
$$\arg\max_{\mathbf{a} \in \mathcal{A}} f\left(\mathbf{c}^i, \mathbf{a}\right) \neq \arg\max_{\mathbf{a} \in \mathcal{A}} f\left(\mathbf{c}^j, \mathbf{a}\right).$$

MMR Retrieval Quality. In traditional BO and CBO, once the action is selected, we refit $\mathcal{M}$ on the augmented $\mathcal{P}$ - however, in our approach, this does not happen i.e., no fine tuning occurs. We simply use $\mathcal{P}$ as a source of history inference for the LLM and we query over $\mathcal{D}$ on each iteration - this is to make sure that we benefit from prompt engineering, specifically from ICL and that the optimal action for each context is available to select on every iteration, mimicking environments that naturally occur in real life scenarios. As a result, once an optimal action a_c^* for a particular context c is identified and included in $\mathcal{P}$, its representation within the few-shot learning template for similar contexts $c' \approx c$ in future iterations increases, formalised as,

$$\forall c' \approx c, a_c^* \in T(x_{t'}) \Rightarrow \phi_{t'}(a_c^*|c') \geq \phi_t(a_c^*|c),$$

where $\phi_{t'}(a_c^*|c') = \Pr(a_{t'} = a_c^*|c_{t'} = c', P_{t'})$.

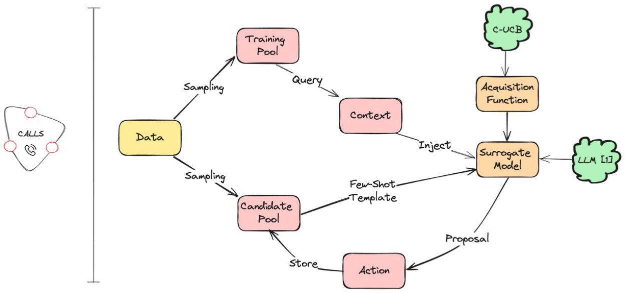


Figure 1. CALLS: Context Aware Large Language System.

4. `CALLS`

4.1. Architectural Motivation

Background. In the formulation above, when using the term *environment*, we refer to the dataset. In our case, this is on molecular optimization, specifically chemical solubility optimization. The context in this case is the *temperature* variable and the features are *SMILES* and *SMILES SOLVENT* pairs. In traditional CBO approaches, the context, such as the temperature variable in our molecular optimization case, is usually integrated into the surrogate models via the acquisition function. However, these models often lack the capability to interpret raw context information, particularly when it doesn't conform to structured numerical or categorical data formats, as they are not designed to parse and understand natural language. This limitation underscores the advantage of leveraging LLMs in such scenarios. With LLMs, we can convert tabular data into a natural language format, enabling the LLM to contextualize and process the data effectively (Tuan Dinh & Lee, 2022). Nevertheless, merely inputting context and feature variables into the LLM does not suffice. Applying *prompt engineering* (Pranab Sahoo & AmanChadha, 2024), is essential to tailor the LLM's focus and output towards the specific requirements of the CBO task at hand.

Assistant Prompt Injection. In specialised tasks, such as predicting solubility values, generic prompts often fall short in guiding LLMs effectively. If LLMs have already been fine-tuned on these tasks, then it is less of a problem however still of concern as there are chances that the LLMs may *hallucinate*, without external verfication. Hence, constrained dynamic role-playing is motivated by the need to tailor prompts specifically to the intricacies of the task at hand, whilst allowing the template style to be dynamically altered. By carefully crafting aligned instructions at the start of the process, we can enhance the model's understanding and performance, thus further assisting in decision-making. We coin this as *ASSISTANT prompt injection*.

MMR Prompt Injection. Providing instructions to the

LLM about the task is not enough to acquire the exact output needed. It is crucial to note that the goal in CBO is to navigate a complex decision space efficiently, identifying optimal points for evaluation with limited computational resources. Traditional approaches often struggle to balance breadth and depth in exploration, particularly when the number of potential evaluations exceeds the practical limits of experimentation. If we were to simply pass all the observed points into the LLM, we would encounter issues of information overload and diminished model effectiveness. This overload, termed *token explosion*, would not only impede the LLM's ability to process the vast amount of data but also degrade the quality of its predictions due to diluted attention across too many data points. Hence, ICL with MMR is motivated by the need to curate a concise yet informative few-shot example prompt template for LLMs, after the assitant prompt injection, to facilitate the prediction of the next optimal point in the CBO process. We coin this as *MMR prompt injection*.

Modified Acquisition Function. The cornerstone of BO is its surrogate model, which is often guided by acquisition functions that balance the trade-off between exploration and exploitation of the search space. Traditionally, Expected Improvement (EI) has been the standard acquisition function in BO given by,

$$\mathrm{E}_{\mathrm{in}}(x) := \mathrm{E}_n \left[f(x) - f_n^*\right]^+$$

Here, $\mathrm{E}_n[\cdot] = \mathrm{E}[\cdot | x_{1:n}, y_{1:n}]$ indicates the expectation taken under the posterior distribution given evaluations of f at $x_1, \ldots, x_n$ and $f_n^* = \max_{m \leq n} f(x_m)$ be the value of this point, where n is the number of times we have evaluated f thus far (Frazier, 2018). While EI has been effective for many applications, it inherently does not allow for an explicit investigation into the confidence intervals of the sample predictions and the cumulative regret bounds are unknown. Hence, we turn out attention to the Upper Confidence Bound (UCB), given by,

$$\mathrm{UCB}\left(\mathbf{a}_t\right) = \mu(\mathbf{a}_t) + \lambda_t \sigma(\mathbf{a}_t)$$

where $\mathbf{a}_t \in \mathcal{A}$ denotes the action variable, indicating sample points devoid of contextual influence, $\mu(\mathbf{a}_t)$ is the predictive mean, and $\sigma(\mathbf{a}_t)$ is the standard deviation representing uncertainty, with λ_t as the exploration-exploitation trade-off coefficient, at time t (Krause & Ong, 2011). Upon examination, it is clear that this acquisition function lacks the capacity to incorporate contextual information, which is crucial in many real-world scenarios, as well as the cornerstone in testing LLMs contextual understanding.

4.1.1. Modified Lift

Before the CBO protocol is executed, we need to initialise the process. We do this by using a modified version of LIFT, a technique used as the bedrock in (Ramos et al., 2023), for which we built atop of.

Assistant Prompt Injection. Our framework allows for the task, features and goal to be dynamically altered depending on the dataset. For our particular use case, we setup a minimal dynamic JSON configuration for the *assistant_prompt*:

This JSON was then parsed through a default *assistant template*: It is important to note that we do not exactly define what the context and action is through the template, in the efforts to not *bias* or assist the LLM in anyway, as we concerned with how it utilises the provided information at each time-step. Additionally, we are not overly concerned with the optimisation of the assistant prompt or template prompt, so long as we align it clearly towards our task, since empirical studies have already been covered on this (Tennison Liu & van der Schaar, 2024).

4.2. CLLM-UCB Algorithm

Once the CBO is initialised by 4.1.1, we can commence the iterative procedure of the novel algorithm, CLLM-UCB.

Algorithm 1 Contextual Large Language Model - Upper Confidence Bound (CLLM-UCB)

```
**Input:** Dataset $\mathcal{D}$, context space $\mathcal{C}$, action space $\mathcal{A}$, surrogate model $\mathcal{M}$, MMR templates, C-UCB acquisition function, unknown objective function $f$, number of iterations $T$
Initialize pool: $\mathcal{P} \subset \mathcal{D}$
Initialize regrets: $\mathcal{L} = []$
**for** $t = 1$ to $T$ **do**
    $\mathbf{c}_t \sim \mathcal{C}$ (uniformly sampled)
    Generate $|\mathcal{D}|$ MMR templates: $\mathcal{T}\left(\{\mathbf{a} : \mathbf{a} \in \mathcal{P}\}\right) = \mathrm{MMR}\left(\mathcal{P}\right)$
    $\mathbf{a}_t = \arg\max_{\mathbf{a}_t \in \mathcal{A}} \text{C-UCB}\left(\mathbf{a}_t \mid \mathbf{c}_t, \mathcal{M}\left(\mathcal{T}\right)\right)$
    $r_t \leftarrow f\left(\mathbf{c}_t, \mathbf{a}_t\right) + \epsilon_t$, where $\epsilon_t \sim \mathcal{N}\left(0, \sigma^2\right)$
    $\mathcal{L}_t \leftarrow f\left(\mathbf{c}_t, \mathbf{a}_t^*\right) - r_t$
    $\mathcal{P} \leftarrow \mathcal{P} \cup \{\mathbf{c}_t, \mathbf{a}_t\}$
    $\mathcal{L} \leftarrow \mathcal{L} \cup \mathcal{L}_t$
**end for**
$R_T = \sum_{t=1}^{T} \mathbb{E}_{\mathcal{M}}\left[\mathcal{L}_t\right]$, where $\mathbf{a}_t^*$ is the action that maximizes $f$ given $\mathbf{c}_t$
**Output:** Pool $\mathcal{P}$, cumulative regret $R_T$
```

MMR Prompt Injection Our framework integrates MMR into the CBO process to generate few-shot example prompts tailored for each available query point. To begin with, after storing an initial subset of points from the data into a pool $\mathcal{P}$ i.e., $n << |\mathcal{D}| = N$, the challenge lies in proposing the next point from the dataset $\mathcal{D}$ for evaluation. For each candidate in this pool, MMR is applied to select a size $k \leq n$ example prompt template, which is most relevant and diverse, relative to $\mathcal{P}$. This template, generated for each candidate in $\mathcal{P}$, is a distilled representation of the decision space and is used by the LLM to predict the value of the next optimal candidate point in context.

Modified Acquisition Function. Our framework extends UCB to C-UCB,

$$\text{C-UCB}\left(\mathbf{c}_t, \mathbf{a}_t\right) = \mu\left(\mathbf{c}_t, \mathbf{a}_t\right) + \beta_t \sigma\left(\mathbf{c}_t, \mathbf{a}_t\right), \qquad (4)$$

where $\mu\left(\mathbf{c}_t, \mathbf{a}_t\right)$ is the mean prediction, $\sigma\left(\mathbf{c}_t, \mathbf{a}_t\right)$ is the standard deviation of the prediction, and β_t controls the trade-off between exploration and exploitation. This allows us to directly inject the context variable inside the acquisition function and conduct bayesian inference through the action selection (Krause & Ong, 2011).

4.2.1. Theoretical Guarantees

Our main theorem bounds the cumulative regret of `CLLM-UCB`.

Lemma 1: *(Convergence of K-shot Templates for Optimal Action to the Ideal Template)* For each context c, the k-shot template $T_t(a^*(c))$ for the optimal action $a^*(c)$ converges to the ideal template $T_t^*(a^*(c))$ as t tends to infinity. The ideal template $T_t^*(a^*(c))$ is defined as the template that maximizes the LLM's predictive accuracy for $a^*(c)$.

Lemma 2: *(Enhancement of LLM Prediction Accuracy for Optimal Action as Templates Converge)* As the k-shot template $T_t(a^*(c))$ for the optimal action $a^*(c)$ in context c converges to the ideal template $T_t^*(a^*(c))$, the LLM's prediction accuracy for $a^*(c)$ improves, leading to a decrease in prediction error and an increase in the likelihood of accurately predicting the value of $a^*(c)$.

Lemma 3: *(Decrease in Instantaneous Regret as t Increases)* As t increases, the instantaneous regret $r_t(c)$ experienced for choosing any action in context c, especially as the algorithm increasingly favors the optimal action due to improved prediction accuracy, decreases, contributing to the overall reduction in cumulative regret.

Main Theorem. *(Sublinear Cumulative Regret of CLLM-UCB)* The cumulative regret R_T of the `CLLM-UCB` algorithm over T iterations is sublinear, i.e.,

$$\lim_{T \to \infty} \frac{R_T}{T} = 0$$

.

5. Experiments

5.1. Dataset Background

BigSolDB is an expansive and diverse solubility dataset encompassing a wide range of organic compounds. BigSolDB consists of 54273 individual solubility values for 830 unique molecules and 13888 individual solvents. The temperature range covered in the dataset spans from 243.15 to 403.15 K at atmospheric pressure. Notably, the top 8 solvents (ethanol, methanol, isopropanol, ethyl acetate, acetone, n-propanol, water, n-butanol, acetonitrile) account for 101 combinations, which collectively represent 57% of all the measured data points. Surprisingly, water, with 3168 records, only constitutes 5.8% of the total and ranks 7^{th} in frequency. This observation suggests that these 8 solvents are the most commonly used in organic chemistry. A similar trend is observed for individual compounds as well, with a majority of experiments conducted in alcohols rather than water. However, this bias may prove undesirable for medicinal chemistry, where water solubility of drug candidates holds greater importance (Big, 2023).

5.2. Dataset Preprocessing

The data preprocessing stage begins with loading the BigSolDB dataframe into our environment, which contains 54273 entries, spread across 5 columns. To ensure the integrity and consistency of the dataset, we carry out an extensive data cleaning process. This involves the removal of rows which have missing values and the elimination of duplicate entries. Following this, we proceed to rename the column labeled *T,K* to *Temperature* to enhance clarity and simplify future referencing. The data is then meticulously sorted by the *SMILES* column, a step that sets the stage for more effective data manipulation. We then reduce its size by focusing exclusively on unique *SMILES* strings and their corresponding temperatures, in pursuit of a dataset that is both concentrated and manageable. Finally, we apply a threshold that guarantees a minimum frequency of *SMILES* occurrences for each temperature value. This is complemented by imposing restrictions on the length of the *SMILES* strings, thereby mitigating any potential token length issues when these strings are used within the context windows of LLMs.

Table 1. CBO dataset setup.

# Distinct Actions	**# Distinct Contexts**	**# Distinct Action maximizers**
100	1 (313.15°)	1
50	2 (313.15°, 308.15°)	1
50	2 (313.15°, 308.15°)	2
25	4 (313.15°, 308.15°, 303.15°, 298.15°)	4
25	8 (313.15°, 308.15°, 303.15°, 298.15°, . . .)	...

5.3. Component Configuration

We use GPT-3.5-Turbo as running inference is much cheaper than the GPT-3 models (Curie and Davinci) and the token length can reach up to $4,096$k, whereas GPT-3 models reach $2,049$. The MMR size is set to $\min\left(\#train, 10\right)$ - this is primarily a result of accommodating for the increased complexity when transitioning from single-context to multiple-context datasets, i.e. from BO to CBO.

Table 2. CBO experimental setup.

Components	**Values**
Dataset	100 examples
λ	$[1, 5, 10]$
# training points	%of the dataset
# iterations	$[30, 50]$
K	$\min(\#initial\ train, 10)$
Model	OpenAI GPT-3.5-Turbo

5.4. Results

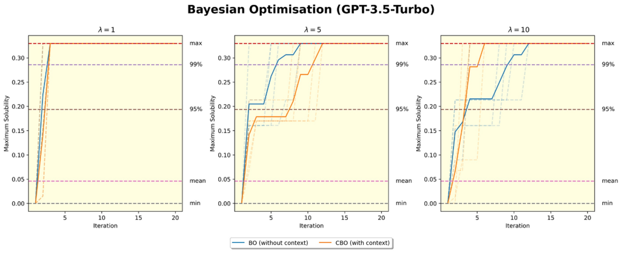


Figure 2. BO maximum solubility function across varying λ values. Initial training size is 15, MMR selection size is $\min\left(\#initial\ train, 10\right)$, and iteration size is 20. 1 temperature with 1 unique action maximizer.

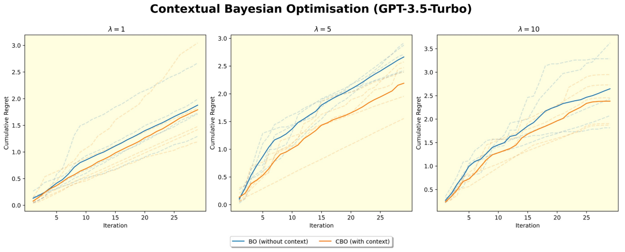


Figure 3. CBO maximum solubility function across varying λ values. Initial training size is 15, MMR selection size is $\min\left(\#initial\ train, 10\right)$, and iteration size is 30. 2 temperatures with 2 distinct action maximizers.

Number of Initial Training Points and Cumulative Regret. The amount of initial training data does not seem to have a clinical impact on the performance of both BO and CBO. This might because we set a minimum ....

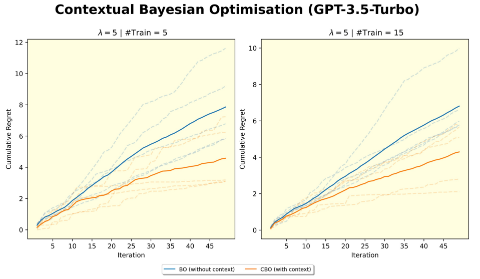


Figure 4. CBO maximum solubility function across varying λ values. Initial training size is 15, MMR selection size is $\min\left(\#initial\ train, 10\right)$, and iteration size is 50. 4 temperatures with 4 distinct action maximizers.

λ and Cumulative Regret. ...

BO Vs CBO Cumulative Regret. CBO tends to outperform BO when the context is a factor in the decision-making process. This is evident in scenarios with multiple contexts and distinct action maximizers, where CBO's ability to account for the additional contextual information allows it to more effectively navigate the solution space and identify optimal actions, thereby minimizing cumulative regret more efficiently than BO. Applying the least square best fit further demonstrated that BO tended to $O\left(T\right)$ whereas CBO tended to $O\left(T^{\frac{1}{2}}\right)$.

STILL BEING COMPLETED!!!

6. Discussions

6.1. Conclusion

In this paper, we tackle the challenge of optimizing an unknown cost function, by selecting optimal actions under the influence of time-varying contextual information, with a specific focus on *molecular solubility optimization*. This task involves maximizing solubility outcomes from chemical interactions between solvents and solutes across various temperatures. We introduced `CALLS`, an innovative framework that integrates the novel `CLLM-UCB` algorithm, leveraging LLMs alongside advanced prompt engineering, to refine CBO. Our findings illuminate the potential of LLMs beyond their traditional use as stochastic autoregressive models, demonstrating their value within optimization algorithm frameworks, specifically in scenarios requiring nuanced contextual understanding.

6.2. Limitations and Future Work

Our study, while pioneering in its integration of LLMs with CBO, encounters several limitations. A significant constraint is the absence of theoretical guarantees, a challenge compounded by the inherent complexity of the analysis involved. Given that we have minimal assumptions on f and that we use LLM as priors, it may be difficult to put forward regret bounds - one possible way is by utilising the theory from information gain. Furthermore, the process of conducting inference through LLM across all templates is resource-intensive, highlighting a critical need for efficient parallelization to enhance time and cost efficiency - note that this could be an issue due to the request limit set by OpenAI. Finally, the method's success is intricately linked to the quality of the MMR mechanism and the performance of the chosen LLM, underscoring the dependency on external factors that could influence the optimization outcome.

Looking ahead, there are several avenues to extend and enhance this research. One potential direction is to experiment with an annealing λ_t value, which could dynamically optimize the trade-off between exploration and exploitation - alternatively, we could capture this in a new objective metric such as *minimax* regret. Expanding the number of contexts to explore a combinatorial context space presents another opportunity, although this may necessitate the implementation of feature selection strategies, particularly if scalability becomes an issue with larger context spaces. Such strategies were beneficial in previous regression analyses on similar datasets, suggesting a viable path to maintaining effectiveness. Additionally, leveraging advancements in LLMs, such as utilizing GPT-4, could harness improved contextual understanding and processing capabilities, offering a richer and more informed decision-making framework.

6.3. Ethics and Reproducibility

In this work, we evaluate a public open-source dataset revolving the field of medicinal chemistry. Additionally, we adhere to the guidelines provided by OpenAI when running GPT-3.5 Turbo. All experimental investigations were conducted on a M1 Macbook Air 2020, throughout August 2023 to December 2023. Given the stochasticity of the LLMs, it is possible that attempting to reproduce the results may yield similar but not exact figures as demonstrated in this paper.

References

Bigsoldb: Solubility dataset of compounds in organic solvents and water in a wide range of temperatures, 2023. URL https://chemrxiv.org/engage/chemrxiv/article-details/6426c1d8db1a20696e4c947b.

Reason for future, act for now: A principled framework for autonomous llm agents with provable sample efficiency, 2023. arXiv:2309.17382.

Pfns4bo: In-context learning for bayesian optimization, 2023. arXiv:2305.17535.

Connecting large language models with evolutionary algorithms yields powerful prompt optimizers, 2024a. arXiv:2210.08087.

Do llm agents have regret? acase study in online learning and games, 2024b. arXiv:2403.16843.

Aaron Mueller, Albert Webson, J. P. and Linzen, T. In-context learning generalizes, but not always robustly: The case of syntax, 2023. arXiv:2311.07811.

Amatriain, X. Prompt design and engineering: Introduction and advanced methods, 2024. arXiv:2401.14423.

Anthropic. The claude 3 model family: Opus, sonnet, haiku, 2024. URL https://www-cdn.anthropic.com/de8ba9b01c9ab7cbabf5c33b80b7bbc618857627/Model_Card_Claude_3.pdf.

Banghao Chen, Zhaofeng Zhang, N. L. and Zhu, S. Unleashing the potential of prompt engineering in large language models: a comprehensive review, 2023. arXiv:2310.14735.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33: 1877–1901, 2020.

Calandra, R., Seyfarth, A., Peters, J., and Deisenroth, M. P. Bayesian optimization for learning gaits under uncertainty: An experimental comparison on a dynamic bipedal walker. *Annals of Mathematics and Artificial Intelligence*, 76:5–23, 2016.

Char, I., Chung, Y., Neiswanger, W., Kandasamy, K., Nelson, A. O., Boyer, M., Kolemen, E., and Schneider, J. Offline contextual bayesian optimization. Conference on Neural Information Processing Systems (NeurIPS), 2019.

Chengrun Yang, Xuezhi Wang, Y. L. H. L. Q. V. L. D. Z. and XinyunChen. Large language models as optimizers, 2023. arXiv:2309.03409.

Do, B. and Zhang, R. Multi-fidelity bayesian optimization in engineering design, 2023. arXiv:2311.13050.

Frazier, P. I. A tutorial on bayesian optimization, 2018. arXiv:1807.02811.

Kirschner, J., Bogunovic, I., Jegelka, S., and Krause, A. Distributionally robust bayesian optimization. International Conference on Artificial Intelligence and Statistics (AISTATS), 2020.

Krause, A. and Ong, C. S. Contextual gaussian process bandit optimization. Conference on Neural Information Processing Systems (NeurIPS), 2011.

OpenAI. Gpt-4 technical report, 2024. arXiv:2303.08774.

Pranab Sahoo, Ayush Kumar Singh, S. S. V. J. S. M. and AmanChadha. A systematic survey of prompt engineering in large language models: Techniques and applications, 2024. arXiv:2402.07927.

Qingxiu Dong, Lei Li, D. D. C. Z. Z. W. B. C. X. S. J. X. L. L. and Sui, Z. A survey on in-context learning, 2023. arXiv:2301.00234.

Ramos, M. C., Michtavy, S. S., Porosoff, M. D., and White, A. D. Bayesian optimization of catalysts with in-context learning, 2023. arXiv:2304.05341.

Rosa, S. S., Nunes, D., Antunes, L., Prazeres, D. M. F., Marques, M. P. C., and Azevedo, A. M. Maximizing mrna vaccine production with bayesian optimization. *PubMed*, 2022.

Shyam Sundhar Ramesh, Andreas Krause, P. G. S. and Bogunovic, I. Movement penalized bayesian optimization with application to wind energy systems, 2022. arXiv:2210.08087.

Takeshi Kojima, Machel Reid, Y. M. S. S. G. and Iwasawa, Y. Large language models are zero-shot reasoners, 2023. arXiv:2205.11916v4.

Tennison Liu, Nicolas Astorga, N. S. and van der Schaar, M. Large language models to enhance bayesian optimization, 2024. arXiv:2402.03921.

Tuan Dinh, Yuchen Zeng, R. Z. Z. L. M. G. S. R. J.-y. S. D. P. and Lee, K. Lift: Language-interfaced fine-tuning for non-language machine learning tasks, 2022. arXiv:2206.06565.

Wang, Y., Yao, Q., Kwok, J. T., and Ni, L. M. Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys (CSUR)*, 53(3):1–34, 2020.

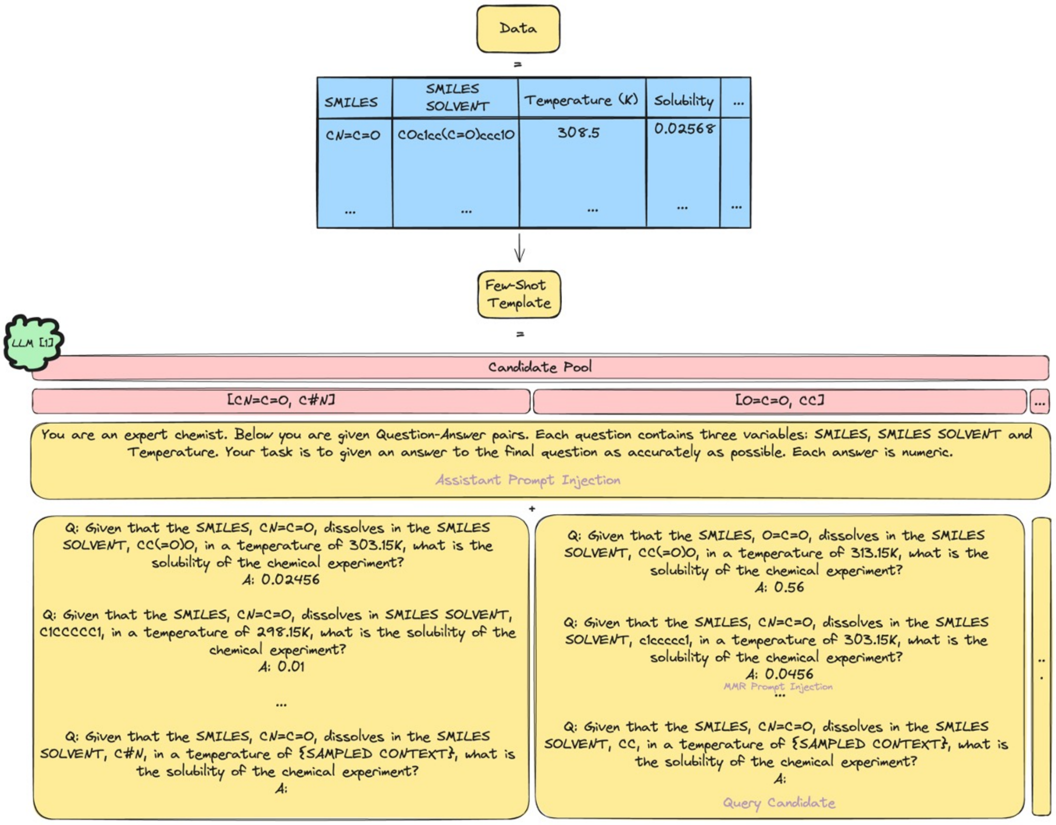
Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., et al. Emergent abilities of large language models, 2022a. arXiv:2206.07682.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35: 24824–24837, 2022b.

Yilun Zhu, Joel Ruben Antony Moniz, S. B. J. L. D. P. S. L. Y. Z. H. Y. and Tseng, B.-H. Can large language models understand context?, 2024. arXiv:2402.00858.

A. CLLM-UCB Algorithm

Regarding Section 4.2, we outline the data representation for the LLM within the `CLLM-UCB` algorithm:



SMILES	SMILES SOLVENT	Temperature (K)	Solubility	...
CN=C=O	COc1cc(C=O)ccc1O	308.5	0.02568	
...	...	...	...	...

You are an expert chemist. Below you are given Question-Answer pairs. Each question contains three variables: SMILES, SMILES SOLVENT and Temperature. Your task is to given an answer to the final question as accurately as possible. Each answer is numeric.

Q: Given that the SMILES, CN=C=O, dissolves in the SMILES SOLVENT, CC(=O)O, in a temperature of 303.15K, what is the solubility of the chemical experiment?
A: 0.02456

Q: Given that the SMILES, CN=C=O, dissolves in SMILES SOLVENT, C1CCCCC1, in a temperature of 298.15K, what is the solubility of the chemical experiment?
A: 0.01

...

Q: Given that the SMILES, CN=C=O, dissolves in the SMILES SOLVENT, C#N, in a temperature of {SAMPLED CONTEXT}, what is the solubility of the chemical experiment?
A:

Q: Given that the SMILES, O=C=O, dissolves in the SMILES SOLVENT, CC(=O)O, in a temperature of 313.15K, what is the solubility of the chemical experiment?
A: 0.56

Q: Given that the SMILES, CN=C=O, dissolves in the SMILES SOLVENT, c1ccccc1, in a temperature of 303.15K, what is the solubility of the chemical experiment?
A: 0.0456

...

Q: Given that the SMILES, CN=C=O, dissolves in the SMILES SOLVENT, CC, in a temperature of {SAMPLED CONTEXT}, what is the solubility of the chemical experiment?
A:

Figure 5. Few-Shot prompt template generation. The tabular data is converted into natural language, allowing the LLM to iteratively suggest and evaluation solutions, informed by the CBO problem description and search history.

A.1. Theoretical Guarantees

Regarding Section 4.2.1, we outline the proofs for each of the lemmas and then the main theorem.

Proof of Lemma 1. In Lemma 1, we examine the behavior of the k-shot template $T_t(a^*(c))$ for the optimal action $a^*(c)$ in context c as it evolves over iterations t. Specifically, we quantify how $T_t(a^*(c))$ approaches the ideal template $T_t^*(a^*(c))$, measured using the Hausdorff distance d_H, and discuss the significance of this convergence in the context of the action space (A, d):

1. **Template Convergence**:
$$\lim_{t\to\infty} d_H(T_t(a^*(c)), T_t^*(a^*(c))) = 0,$$
indicating that the actual k-shot template $T_t(a^*(c))$ becomes indistinguishable from the ideal template $T_t^*(a^*(c))$ in the limit.

2. **Ideal Template Characteristics**: The ideal template $T_t^*(a^*(c))$ is such that for any action a in $T_t^*(a^*(c))$, the LLM's prediction accuracy for $a^*(c)$ is maximized, implying $T_t^*(a^*(c))$ likely contains $a^*(c)$ or actions very close to it.

3. **Implications for LLM Prediction Accuracy**: As $T_t(a^*(c))$ converges to $T_t^*(a^*(c))$, the LLM's predictions based on $T_t(a^*(c))$ become increasingly accurate, approaching the accuracy achieved with the ideal template.

□

Proof of Lemma 2. In Lemma 2, we explore the consequences of the convergence of the k-shot template $T_t(a^*(c))$ for the optimal action $a^*(c)$ in context c to the ideal template $T_t^*(a^*(c))$ as t tends to infinity. This convergence, quantified by the Hausdorff distance d_H, has significant implications for the LLM's prediction accuracy and the likelihood of correctly predicting the value of $a^*(c)$. Here we detail how this convergence enhances the LLM's predictive performance:

1. **Template Convergence and Prediction Accuracy:** Given the optimal action $a^*(c)$ for context c, as $T_t(a^*(c))$ converges to the ideal template $T_t^*(a^*(c))$ (i.e., $\lim_{t\to\infty} d_H(T_t(a^*(c)), T_t^*(a^*(c))) = 0$), the LLM's prediction error for $a^*(c)$ decreases:
$$\lim_{t\to\infty} |f(c, a^*(c)) - \hat{f}(c, a^*(c))| = 0.$$

2. **Probability of Accurate Prediction:** The probability of the LLM accurately predicting the value of $a^*(c)$ approaches 1:
$$\lim_{t\to\infty} P(\hat{f}(c, a^*(c)) = f(c, a^*(c)) | T_t(a^*(c))) \geq 1 - \epsilon_t,$$
where ϵ_t is a small error term that diminishes as t increases.

3. **Implications for LLM Predictive Performance:**
   - The increasing resemblance of $T_t(a^*(c))$ to $T_t^*(a^*(c))$ enhances the template's representativeness for $a^*(c)$, leading to more accurate LLM predictions.
   - The convergence of $T_t(a^*(c))$ to $T_t^*(a^*(c))$ ensures that the actions within the template are highly representative for predicting $a^*(c)$, thereby minimizing the prediction error and maximizing the likelihood of correct predictions.
   - This improved predictive accuracy directly translates to a reduction in the prediction error for $a^*(c)$, showcasing the LLM's enhanced capability to model and predict outcomes accurately over iterations.

□

Proof of Lemma 3. In Lemma 3, we explore the implications of the increasing representativeness of the k-shot template $T_t(a^*(c))$ for the optimal action $a^*(c)$ in context c. We demonstrate how this evolution, quantified by the convergence to the ideal template $T_t^*(a^*(c))$, leads to a decrease in instantaneous regret $r_t(c)$ as the number of iterations t increases:

1. **Reduction in Instantaneous Regret:** Let $r_t(c)$ denote the instantaneous regret at iteration t for context c, defined as $r_t(c) = f(c, a^*(c)) - f(c, a_t)$, where a_t is the action chosen by the algorithm at time t. As the LLM's prediction accuracy for $a^*(c)$ improves, the algorithm is more likely to choose actions close to $a^*(c)$, thus reducing $r_t(c)$:
$$\lim_{t\to\infty} r_t(c) = 0.$$

2. **Enhanced Prediction Accuracy and Regret Reduction:** The convergence of $T_t(a^*(c))$ to the ideal template $T_t^*(a^*(c))$ enhances the LLM's prediction accuracy for $a^*(c)$, as established in Lemma 2. This improved accuracy directly influences the algorithm's ability to select the optimal action, thereby minimizing $r_t(c)$.

3. **Cumulative Impact on Regret:** The decrease in $r_t(c)$ for each context c over time contributes to the reduction of the cumulative regret R_T, supporting the theorem that the cumulative regret grows sublinearly with respect to T.

□

Proof of Main Theorem. In this theorem, we synthesize the insights gained from the preceding lemmas to establish that the CLLM-UCB algorithm exhibits sublinear growth in cumulative regret R_T over T iterations:

1. **Instantaneous Regret Convergence:** Based on Lemma 3, for each context c, the instantaneous regret $r_t(c)$ decreases as t increases, ultimately tending toward zero: $\lim_{t\to\infty} r_t(c) = 0$.

2. **Aggregate Instantaneous Regret:** The cumulative regret R_T is the sum of instantaneous regrets over all contexts and iterations:

$$R_T = \sum_{t=1}^{T} \sum_{c \in C} r_t(c).$$

Given that each $r_t(c)$ is decreasing due to the enhanced prediction accuracy (from Lemma 2) and convergence of k-shot templates (from Lemma 1), the overall sum R_T grows at a sublinear rate.

3. **Demonstrating Sublinearity:** To show sublinearity, we analyze the rate of growth of R_T with respect to T:

$$\lim_{T \to \infty} \frac{R_T}{T} = 0.$$

This limit signifies that the average regret per iteration decreases over time, confirming the sublinear growth of cumulative regret.

□

(LLM, 2024b; 2023)

B. Results

Regarding Section 5.4, we establish additional results for both BO and CBO:

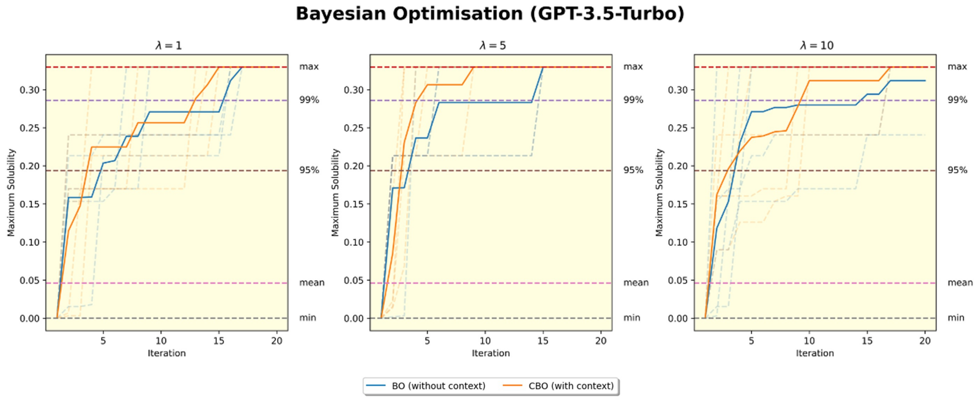


Figure 6. BO maximum solubility function across varying λ values. Initial training size is 5, MMR selection size is $\min(\#initial\ train, 10)$, and iteration size is 20. 1 temperature with 1 unique action maximizer.

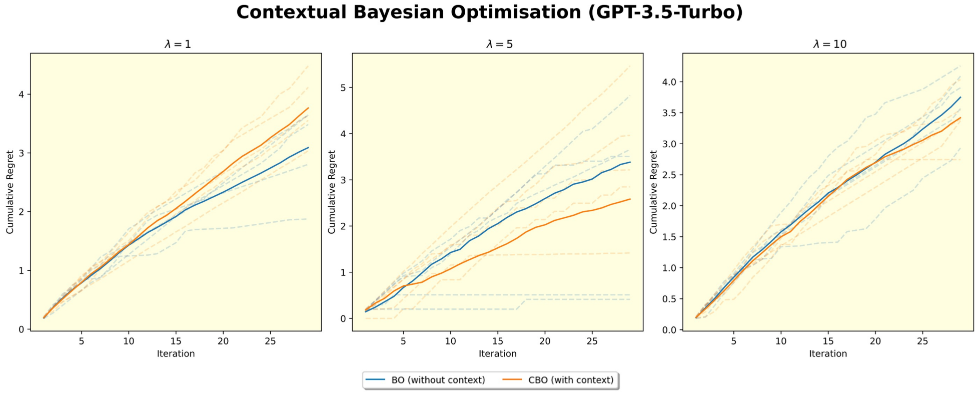


Figure 7. BO maximum solubility function across varying λ values. Initial training size is 5, MMR selection size is $\min(\#initial\ train, 10)$, and iteration size is 30. 2 temperatures with 1 distinct action maximizer.

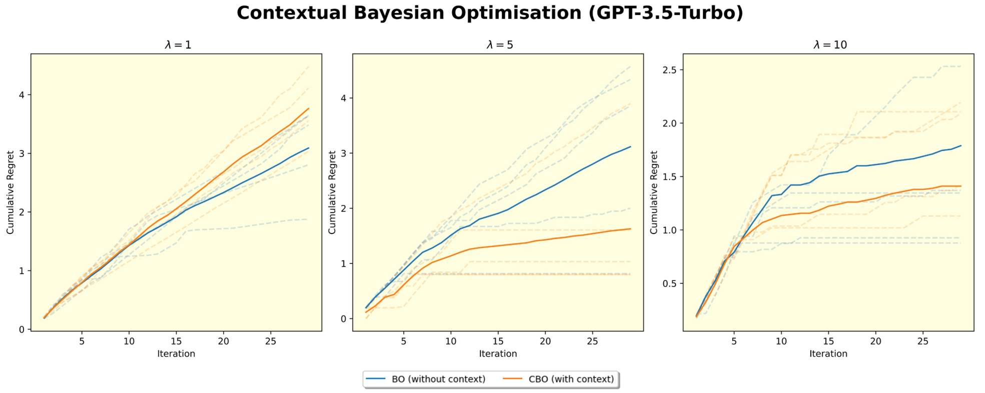


Figure 8. BO maximum solubility function across varying λ values. Initial training size is 15, MMR selection size is $\min{(\#initial\ train, 10)}$, and iteration size is 30. 2 temperatures with 1 distinct maximizer.

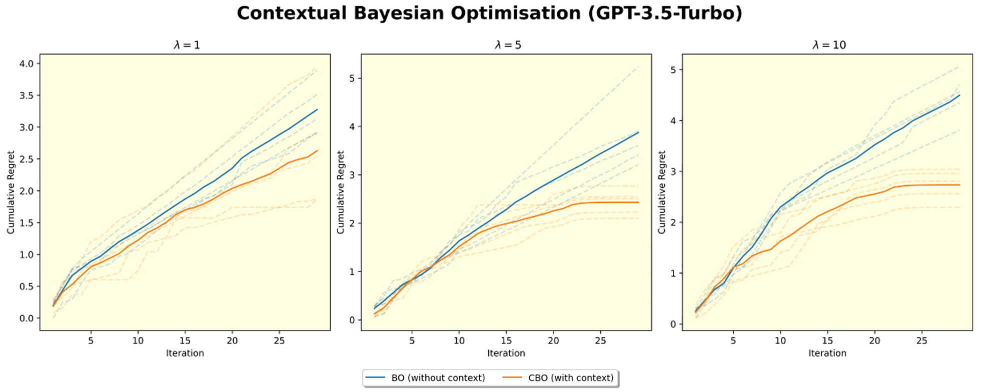


Figure 9. BO maximum solubility function across varying λ values. Initial training size is 5, MMR selection size is $\min{(\#initial\ train, 10)}$, and iteration size is 30. 2 temperatures with 2 distinct action maximizers.