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

Siddartha Nath [1] **Shyam Sundhar Ramesh** [2]

Abstract

Contextual Bayesian optimization (CBO) is a powerful framework for sequential decision-making w.r.t. an unknown function. Similar to Bayesian optimization (BO), it utilizes surrogate models to approximate the unknown function, but with access to additional contextual information. Recent literature has highlighted the potential of using a Large Language Model (LLM) in optimisation frameworks - we contribute to this research area by introducing **C**ontext **A**ware **L**arge **L**anguage **S**ystem (`CALLS`), an architecture which translates the optimisation protocol into a natural language interface and utilizes the novel `CLLM-UCB` algorithm. The algorithm combines a LLM, as the surrogate model via in-context learning (ICL) and, a Contextual Upper Confidence Bound (C-UCB), as the acquisition function. Our algorithm demonstrates that a LLM in a CBO framework, outperforms their counterpart in context-absent BO setup, by achieving sublinear contextual regret.

1. Introduction

Solubility plays a pivotal role in organic chemistry and carries considerable importance in medicinal chemistry. Current techniques involve Quantitative Structure-Property Relationship (QSPR) modeling for estimating solubility to minimize the need for expensive experimental procedures. However, fundamental drawbacks to QSPR involve 1) assuming a multiple linear regression system, which prohibits non-linear functional representations and 2) the inability to search and extrapolate decisions in a large combinatorial feature space. To address these challenges, one promising strategy is Bayesian optimization (BO).

Bayesian optimization (BO) is an effective method for optimizing black-box functions, which are complex and computationally demanding to evaluate, and finds application in diverse fields such as robotics, vaccine design, and engineering (Calandra et al., 2016; Rosa et al., 2022; Do & Zhang, 2023). The process uses a *surrogate model*, which serves as a proxy for the unknown function. It is initially trained on a preliminary dataset and then iteratively selects and evaluates new data points from a *pool*, using an *acquisition function* that balances exploration and exploitation to choose points that are expected to provide the most informative feedback, thus refining the model until a predetermined performance criterion is met. A key architectural limitation with BO is that when additional information is available and beneficial to be used in decision-making, it is not possible to leverage this and thus, BO is extended to Contextual Bayesian optimization (CBO), where point selection is conditional on the context observed, analogous to the Contextual Bandits (CB) approach in Reinforcement Learning (RL) (Langford & Zhang, 2008; Ali Baheri, 2023).

Regarding both setups, 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). Yet, there is a significant obstacle in their use - they encounter scalability issues; GPs struggle with computational intensiveness in high-dimensional space, and NNs often require extensive data to learn effectively and are uninterpretable. This scenario is typically framed as the *few-shot* paradigm, where there is a necessity for adaptation and generalization based on minimal high-dimensional data (Wang et al., 2020). Interestingly, such challenges align with the proficiencies of Large Language Models (LLMs).

In light of this, we introduce `CALLS`, an automated CBO system, using the in-context learning (ICL) capability of LLMs, via the `CLLM-UCB` algorithm. We hypothesize that LLMs, in the presence of context, (1) possess contextual understanding and (2) reliably provide decision adjustments, offering an alternative choice for a surrogate model. Our experiments across a broad combinatorial feature space with restricted data access support our hypotheses.

2. Related Work

In the realm of optimization, our research aligns with several established algorithms, such as Bayesian optimization (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; Qingyan Guo & Yang, 2024) and In-Context Learning (ICL) in BO with GPs and NNs (Samuel Muller & Hutter, 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.

In the realm of LLMs, such as GPT4 and Claude 3 (OpenAI, 2024; Anthropic, 2024), which have undergone pre-training on vast internet-scale datasets, these have demonstrated 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.

Our research 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. We extend this by investigating an unobserved research avenue - using an LLM as a surrogate model, within CBO, in an automated, dynamic and scalable way.

3. Problem Statement

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 aims to approximate 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,

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

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

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

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*:

$$R_T = \sum_{t=1}^{T} \mathcal{L}_t \tag{3}$$

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

$$\lim_{T\to\infty} \frac{R_T}{T} = 0 \equiv O(R_T) << O(T) \tag{4}$$

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

4. CALLS

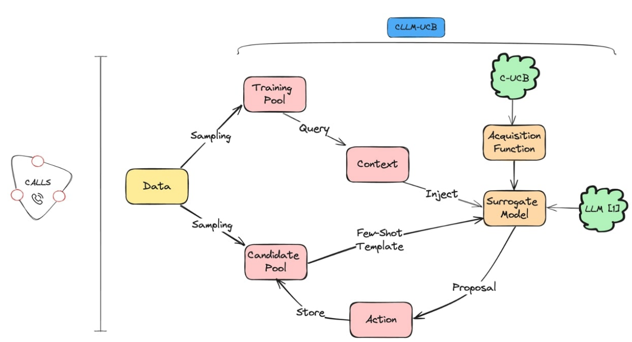

Figure 1. CALLS: Context Aware Large Language System.

4.1. Modified-Lift

Before the CBO protocol is executed, we need to initialise the process. We do this by using a modified version of `LIFT` (Tuan Dinh & Lee, 2022).

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 is 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 are concerned with how it utilizes the provided information at each time-step. Additionally, we are not overly concerned with the optimisation of the assistant or template prompt, so long as it is aligned 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, 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 $\epsilon_t \sim \mathcal{N}\left(0, \sigma^2\right)$)
  $\mathcal{L}_t \leftarrow f\left(\mathbf{c}_t, \mathbf{a}_t^*\right) - r_t$ (where $\mathbf{a}_t^*$ is the action that maximizes f given $\mathbf{c}_t$)
  $\mathcal{P} \leftarrow \mathcal{P} \cup \{\mathbf{c}_t, \mathbf{a}_t\}$
  $\mathcal{L} \leftarrow \mathcal{L} \cup \mathcal{L}_t$
end for
Output: Pool $\mathcal{P}$, $\sum \mathcal{L}$

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,

$$\text{C-UCB}\left(\mathbf{c}_t, \mathbf{a}_t\right) = \mu\left(\mathbf{c}_t, \mathbf{a}_t\right) + \lambda_t \sigma\left(\mathbf{c}_t, \mathbf{a}_t\right), \tag{5}$$

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 λ_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).

In summary, the `CLLM-UCB` algorithm optimizes an unknown objective function f using a contextual LLM $\mathcal{M}$ over a dataset $\mathcal{D}$. It initializes a pool $\mathcal{P}$ with a subset of $\mathcal{D}$ and starts with an empty regrets list. In each iteration, it uniformly samples a context, generates MMR templates for candidates in $\mathcal{D}$, selects the action that maximizes the acquisition function, evaluates the objective function, calculates regret, and updates the pool and regrets list. The algorithm concludes by outputting the final pool and the cumulative regret.

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 account for 101 combinations, which collectively represent 57% of all the measured data points (Krasnov et al., 2023).

5.2. Dataset Preprocessing

The preprocessing of BigSolDB, with $54,273$ entries across 5 columns, begins by loading the dataframe and performing extensive cleaning: 1) Removing rows with missing values, 2) Eliminating duplicates, and 3) Renaming the *T,K* column to *Temperature*. Then, the dataset is sorted by the *SMILES* column, reduced to unique *SMILES* strings and their temperatures, and filtered by a frequency threshold for *SMILES* occurrences. We also limit the length of *SMILES* strings to address token length issues in LLMs. Finally, to ensure equal numbers of unique contexts and action maximizers, the dataset is artificially modified.

Table 1. Different CBO datasets.

# Distinct Actions	# Distinct Contexts	# Unique 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

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 4096, whereas GPT-3 models reach 2049. The MMR size is set to $\min(\#initial\ train, 10)$ - this is primarily a result of accommodating for the increase in dimensionality of the feature space, when transitioning from BO to CBO.

Table 2. CBO experimental setup.

Components	Values
Dataset	100 examples
λ	$[1, 5, 10]$
# training points	%of the dataset
# iterations	$[20, 30, 50]$
K	$\min(\#initial\ train, 10)$
Model	OpenAI GPT-3.5-Turbo

5.4. Results

In each figure, the thick colored lines show the mean maximum solubility or mean cumulative regret up to $t = T$, while the 5 lighter lines depict the maximum solubility or cumulative regret across 5 different experimental runs. Note that the we do not concern ourselves with outperforming any baselines.

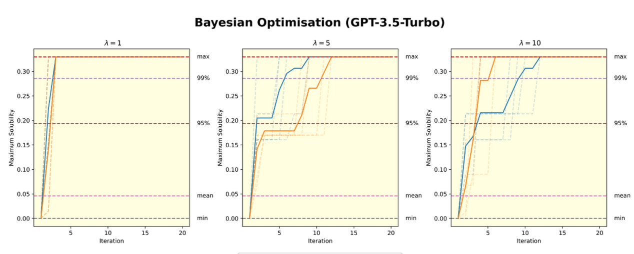

Figure 2. BO maximum solubility function across varying λ values. Initial training size is 15, MMR selection size is $\min(\#initial\ train, 10)$, and iteration size is 20. 1 temperature with 1 unique action maximizer.

When considering Figure 2, across all λ values, both BO (without context) and CBO (with context) frameworks converge to the maximum solubility i.e., observing optimal action, at similar rates. Given that there is a single value for the single context involved, we would assume that the performance would be closer since the context does not change and so in CBO, the LLM is not utilising more information about the regression environment, in order to make better informed decisions, compared to BO.

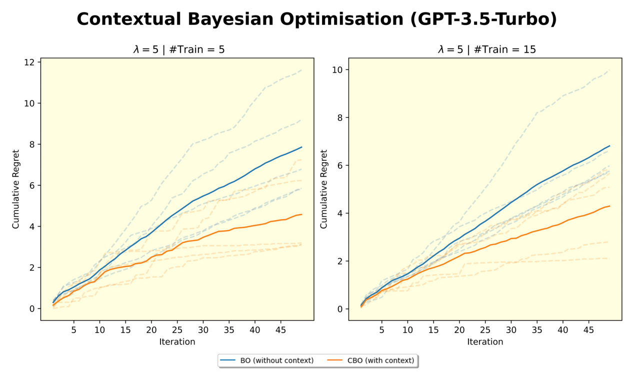

Figure 3. Contextual cumulative regret for $\lambda = 5$. Initial training size is 15, MMR selection size is $\min(\#initial\ train, 10)$, and iteration size is 50. 4 temperatures with 4 unique action maximizers.

Prior experiments using grid-search hyperparameter optimization show that a sensible value for our exploration-exploitation balance is $\lambda = 5$. Figure 3 demonstrates that with a fixed λ, training size becomes less relevant as iteration number increases. This suggests that for some value T, convergence can be guaranteed. The convergence rate is steeper for experiments with more than 4 temperatures due to higher complexity and the need for more iterations. Coupling this with Table 3 reveals the difference in decision-making between frameworks, showing that CBO is asymptotically approaching sublinear contextual regret.

6. Conclusion

This paper addresses optimizing an unknown cost function by choosing optimal actions based on time-varying contextual information, focusing on *molecular solubility optimization*. The goal is to maximize solubility outcomes from chemical interactions between solvents and solutes at various temperatures. We introduce `CALLS`, a framework incorporating the novel `CLLM-UCB` algorithm, which uses LLMs and advanced prompt engineering to enhance CBO. Our results highlight the potential of LLMs beyond their traditional next token prediction, showing their value in optimization algorithms, especially in scenarios requiring nuanced understanding and contextual decision-making.

References

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.

Ali Baheri, C. B. Llms-augmented contextual bandit, 2023. URL https://arxiv.org/pdf/2311.02268.

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.

Krasnov, L., Mikhaylov, S., Fedorov, M., and Sosnin, S. 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.

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

Langford, J. and Zhang, T. The epoch-greedy algorithm for contextual multi-armed bandits, 2008. URL https://proceedings.neurips.cc/paper/2007/file/4b04a686b0ad13dce35fa99fa4161c65-Paper.pdf.

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.

Qingyan Guo, Rui Wang, J. G. B. L. K. S. X. T. G. L. J. B. and Yang, Y. Connecting large language models with evolutionary algorithms yields powerful prompt optimizers, 2024. arXiv:2210.08087.

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.

Samuel Muller, Matthias Feurer, N. H. and Hutter, F. Pfns4bo: In-context learning for bayesian optimization, 2023. arXiv:2305.17535.

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. Problem Statement

Regarding Section 3, we outline the assumptions behind our environmental setup:

i) **Function Structure:** In a traditional BO and CBO setup, $f : \mathcal{C} \times \mathcal{A} \to \mathbb{R}$ 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.

ii) **Single Context Selection:** Regarding the context, we have that,

$$\forall t \in \mathbb{R}, \quad \mathbf{c}_t \in \mathbb{R} \Rightarrow \dim\left(\mathbf{c} = \mathbf{c_t}\right) = 1. \tag{6}$$

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

iii) **Unique 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,

$$\begin{gathered} \forall i, j \in \{1, 2, ..., m\}, i \neq j : \\ \arg\max_{\mathbf{a} \in \mathcal{A}} f\left(\mathbf{c}_i, \mathbf{a}\right) \neq \arg\max_{\mathbf{a} \in \mathcal{A}} f\left(\mathbf{c}_j, \mathbf{a}\right). \end{gathered} \tag{7}$$

iv) **MMR Retrieval Quality:** In traditional BO and CBO, after selecting an action, the model $\mathcal{M}$ is refitted on the augmented dataset $\mathcal{P}$. In contrast, our approach skips this fine-tuning step, avoiding the associated computational overhead. This trade-off affects model learning, but for small datasets, it proves beneficial compared to other surrogate models that require refitting. Instead, we 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 $\mathbf{a}_c^*$, the optimal action for a particular context, is identified and included in $\mathcal{P}$, we hypothesise that,

$$\forall \mathbf{c} \in \mathcal{C}, \quad \lim_{t \to \infty} P\left(\mathbf{a}_t = \mathbf{a}_c^* | \mathbf{a}_c^* \in \mathcal{P}\right) = 1 \tag{8}$$

This notion is the bedrock for sublinear contextual regret as it guarantees that the optimal action for any context is always found in the prompt template of the optimal action.

B. Architectural Motivation

Regarding Section 4.2, we outline the motivation of the CALLS architecture:

i) **Background:** 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.

ii) **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*.

iii) **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*.

iv) **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, which is typically a constant (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.

C. CLLM-UCB Algorithm

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

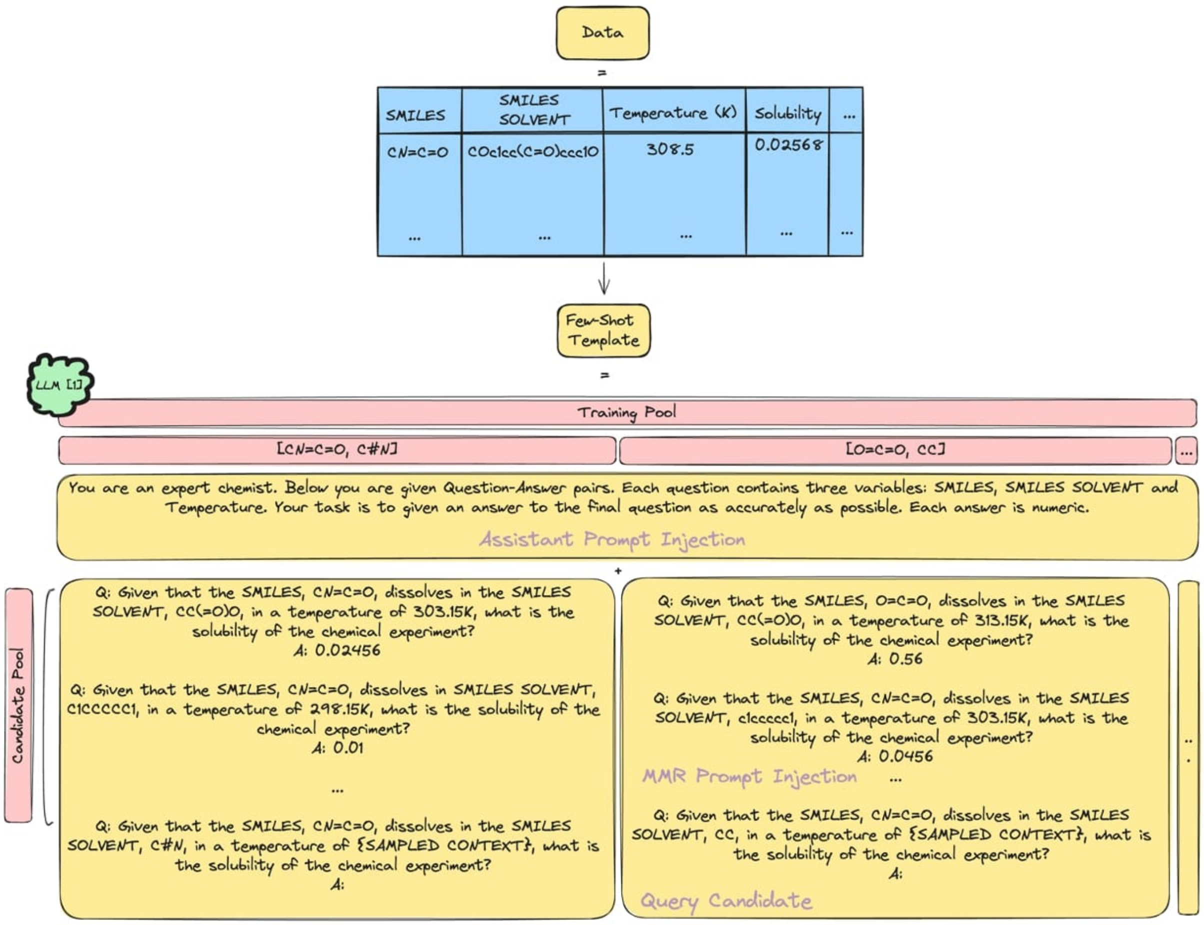

Figure 4. Few-Shot prompt template generation. The tabular data is converted into natural language, allowing the LLM to iteratively suggest and evaluation solutions, informed by the CBO problem description and search history. In comparison, the BO (without context) is where the LLM does not have access to the temperature variable.

The original BigSolDB dataset is preprocessed into the different CBO datasets through sampling, into a training pool and candidate pool. Note that in this setup, the training pool includes the candidate pool.

D. Results

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

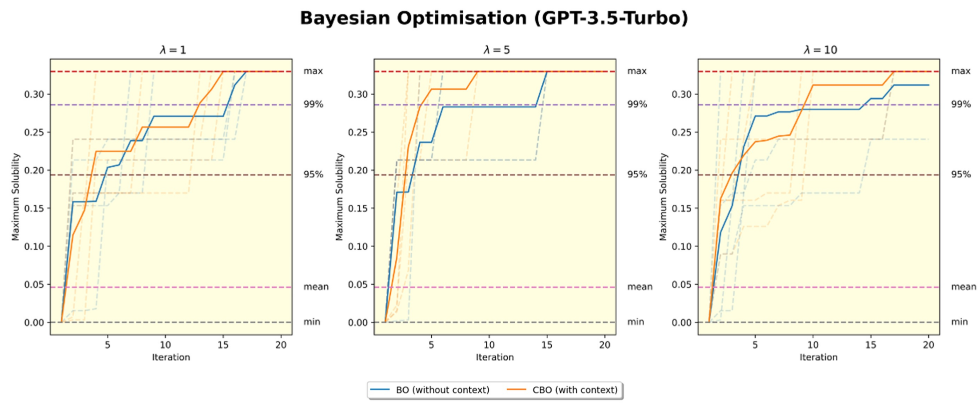

Figure 5. BO maximum solubility function across varying λ values. Initial training size is 5, MMR selection size is $\min(\#initial\ train, 10)$, and iteration size is 20. 1 temperature with 1 unique action maximizer.

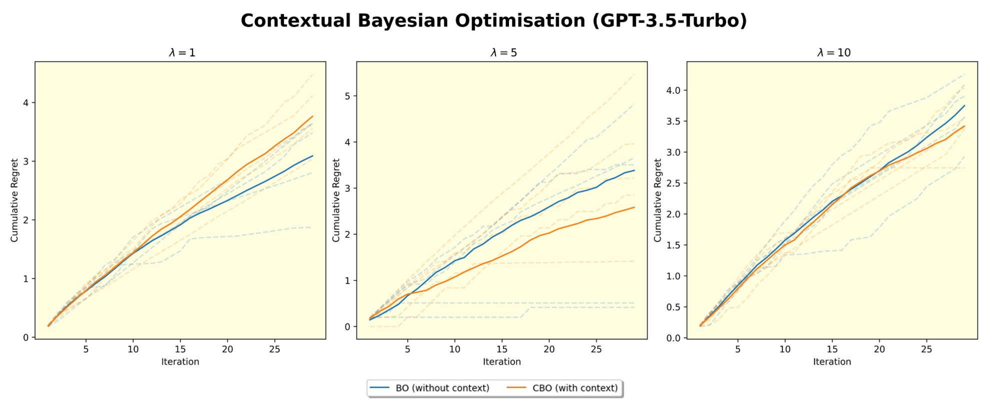

Figure 6. Contextual cumulative regret across varying λ values. Initial training size is 5, MMR selection size is $\min(\#initial\ train, 10)$, and iteration size is 30. 2 temperatures with 1 unique action maximizer.

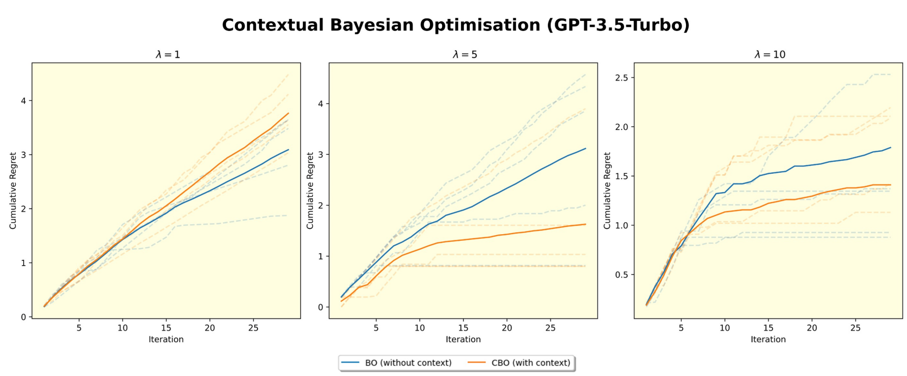

Figure 7. Contextual cumulative regret across varying λ values. Initial training size is 15, MMR selection size is $\min(\#initial\ train, 10)$, and iteration size is 30. 2 temperatures with 1 unique maximizer.

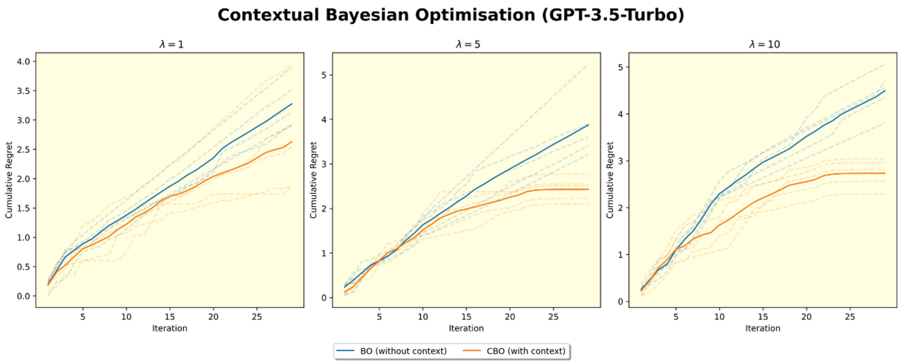

Figure 8. Contextual cumulative regret across varying λ values. Initial training size is 5, MMR selection size is $\min(\#initial\ train, 10)$, and iteration size is 30. 2 temperatures with 2 unique action maximizers.

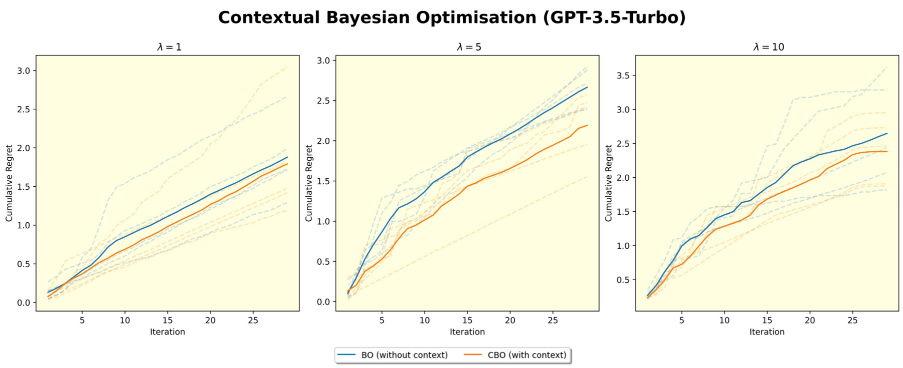

Figure 9. Contextual cumulative regret across varying λ values. Initial training size is 15, MMR selection size is $\min(\#initial\ train, 10)$, and iteration size is 30. 2 temperatures with 2 unique action maximizers.

Table 3. Context, Action and Regret for Iterations $46 - 50$ for BO (without context) and CBO (with context) methods. Initial training size is 15, MMR selection size is $\min(\#initial\ train, 10)$ and $\lambda = 5$. 4 temperatures with 4 unique action maximisers.

Method	Context	Action	Regret
BO	Iteration 46: 313.15 Iteration 47: 313.15 Iteration 48: 308.15 Iteration 49: 303.15 Iteration 50: 298.15	Iteration 46: {"SMILES": "O=c1cc(CO)occ1O", "SMILES Solvent": "CN(C)C=O"} Iteration 47: {"SMILES": "O=c1cc(CO)occ1O", "SMILES Solvent": "CN(C)C=O"} Iteration 48: {"SMILES": "O=c1cc(CO)occ1O", "SMILES Solvent": "CN(C)C=O"} Iteration 49: {"SMILES": "Oc1ccc(OCc2ccccc2)cc1", "SMILES Solvent": "O"} Iteration 50: {"SMILES": "Oc1ccc(OCc2ccccc2)cc1", "SMILES Solvent": "O"}	Iteration 46: 0.2668 Iteration 47: 0.2668 Iteration 48: 0.1866 Iteration 49: 0.3013 Iteration 50: 0.2772
CBO	Iteration 46: 313.15 Iteration 47: 313.15 Iteration 48: 308.15 Iteration 49: 303.15 Iteration 50: 298.15	Iteration 46: {"SMILES": "Oc1ccc(OCc2ccccc2)cc1", "SMILES Solvent": "C1COCCO1"} Iteration 47: {"SMILES": "Oc1ccc(OCc2ccccc2)cc1", "SMILES Solvent": "C1COCCO1"} Iteration 48: {"SMILES": "c1ccc2c(c1)oc1ccccc12", "SMILES Solvent": "Cc1ccccc1"} Iteration 49: {"SMILES": "Oc1ccc(OCc2ccccc2)cc1", "SMILES Solvent": "CCCO"} Iteration 50: {"SMILES": "Oc1ccc(OCc2ccccc2)cc1", "SMILES Solvent": "C1COCCO1"}	Iteration 46: 0.0 Iteration 47: 0.0 Iteration 48: 0.0 Iteration 49: 0.2371 Iteration 50: 0.0872

E. Conclusion

E.1. Limitations and Future Work

Our study, while pioneering in its integration of LLMs with CBO, encounters several limitations. 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. Firstly, we can experiment with a context specific annealing λ_t^c parameter, which could dynamically optimize the trade-off between exploration and exploitation for different contexts. Secondly, levelling from single to multiple contexts and objectives presents another opportunity - the current system supports this, exemplfying its novelty, however, if scalability becomes an issue, this may necessitate the implementation of other strategies. Lastly, leveraging advancements in LLMs, such as utilizing GPT-4 with an agentic approach, by incorporating a feedback loop, could harness improved contextual understanding and processing capabilities, offering a richer and more informed decision-making framework.

E.2. 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. Due to accessibility difficulties, all experiments were conducted locally via a M1 Macbook Air 2020, throughout August 2023 to December 2023 - the code to generate the results can be found in the following repository: CALLS - Context Aware Large Language System.
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.