In-Context Learning for Latent Space Bayesian Optimization

Tuan A. Vu [1] **Harri Lähdesmäki** [1] **Julien Martinelli** [1]

Abstract

Bayesian optimization (BO) is a central tool for sample-efficient design, and latent-space Bayesian optimization (LSBO) extends it to structured objects such as molecules and proteins. In parallel, tabular foundation models such as TabPFN and TabICL now achieve state-of-the-art regression performance and are increasingly used as BO surrogates. Because their Bayesian behavior is induced by large synthetic pretraining collections, the composition of this pretraining distribution is crucial. LSBO creates a distinctive mismatch: the induced map from latent code to objective value differs markedly from the regression tasks used to train current in-context models. We address this mismatch by complementing the pretraining stage of tabular foundation model surrogates with synthetic optimization tasks defined on the latent space of a molecular VAE. The continued-pretraining objective features a regularizer that anchors the model to the original checkpoint, preserving its broad regression prior while avoiding overspecialization to the adaptation tasks. On held-out molecular optimization benchmarks, the resulting model achieves strong performance, supporting the relevance of LSBO-specific adaptation for in-context surrogates.

1. Introduction

Many scientific design problems require optimizing expensive black-box objectives over large structured spaces. Examples include molecular design, protein engineering, and sequence optimization, where each candidate may require a simulator, wet-lab assay, or human-in-the-loop evaluation (Gomez-Bombarelli et al., 2018; Stanton et al., 2022; Gruver et al., 2024). Bayesian optimization (BO) is well suited to this setting because it uses a probabilistic surrogate to balance exploration and exploitation under a limited evaluation budget (Garnett, 2023). The main difficulty is that many design spaces are combinatorial and highly structured.

Latent-space Bayesian optimization (LSBO) addresses this by learning a continuous representation of the structured design space and performing BO in that latent space before decoding candidates back to the original domain (Gomez-Bombarelli et al., 2018; Maus et al., 2022; Lee et al., 2025; Vu et al., 2026). This recipe has become standard in molecular design and increasingly relevant in protein design. The LSBO literature also shows that strong performance depends on more than a single global surrogate: successful methods rely on locality, trust regions, elite-biased updates, or representation adaptation to handle latent-geometry mismatch and decoder-induced errors (Tripp et al., 2020; Maus et al., 2022; Chu et al., 2024; Lee et al., 2025; Moss et al., 2025).

In parallel, tabular foundation models (TFMs) and prior-data fitted networks (PFNs) have emerged as strong probabilistic predictors for tabular regression and classification (Muller et al., 2021; Hollmann et al., 2025; Qu et al., 2026). These models are trained on large synthetic task collections and learn to perform approximate Bayesian prediction in context. Recent works present PFNs as a broad paradigm for Bayesian prediction (Muller et al., 2025; Zhang et al., 2025a). This development is now reaching BO: PFNs4BO uses PFNs as surrogates, GIT-BO scales TFMs to high-dimensional BO (Yu et al., 2026), and in-context optimization policies are beginning to appear (Zhang et al., 2026).

The difficulty is that BO, and especially LSBO, places specific demands on the surrogate. Good decisions depend mainly on high-value regions of the objective and on the local neighborhoods defined by the best observations seen so far (Maus et al., 2022; Yu et al., 2026). For in-context models, this makes pretraining crucial: they inherit useful Bayesian behavior only when their synthetic training tasks resemble those seen at deployment. In LSBO, this resemblance often fails. The map from latent code to objective value is shaped by the decoder and by structured properties of the underlying object. Moreover, the objectives themselves are often domain-specific property scores such as logP, QED, or similarity-based criteria (Brown et al., 2019), rather than the synthetic targets used in TFM pretraining (Qu et al., 2026). Current BO uses of TFMs treat them as generic

[1]Aalto University. Correspondence to: Tuan A. Vu <tuan.t.vu@aalto.fi>.

Accepted at ICML 2026 Workshop on Decision-Making from Offline Datasets to Online Adaptation: Black-Box Optimization to Reinforcement Learning. (DEMO@ICML 2026), Seoul, South Korea. *Copyright 2026 by the author(s).*

plug-in regressors, leaving this mismatch unresolved.

Contributions. We address this mismatch through LSBO-specific adaptation of a pretrained tabular foundation-model surrogate. We construct synthetic objectives from a library of molecular base tasks derived from the GuacaMol database and sample contexts with a value-biased distribution that emphasizes promising regions of the VAE latent space. We use these episodes to continue pretraining a selected tabular foundation model checkpoint, with a regularizer that anchors the adapted model to the original weights and preserves its broad regression prior. We also introduce latent-space and objective-space diagnostics of the proposed synthetic distribution. The resulting model, LILBO (Latent In-context Learning for Bayesian Optimization), can be plugged directly into modern LSBO pipelines with local search, elite mechanisms, and periodic VAE retraining. On held-out molecular optimization tasks, LILBO achieves strong performance, supporting the relevance of LSBO-specific adaptation for in-context surrogates.

2. Related work

Latent-space Bayesian optimization. LSBO extends BO to structured domains by introducing a learned continuous representation in which candidate generation and acquisition optimization become tractable (Gomez-Bombarelli et al., 2018). Much of the recent literature focuses on the mismatch between latent geometry and the objective. Weighted retraining changes the latent data distribution toward promising regions (Tripp et al., 2020); LOLBO combines latent BO with local trust-region search (Maus et al., 2022); CoBO and InvBO improve local alignment or invert observations into better latent codes (Lee et al., 2023; Chu et al., 2024); NF-BO uses normalizing flows to mitigate reconstruction-induced value discrepancy (Lee et al., 2025); and COWBOYS argues for a stronger decoupling between the generative backbone and the surrogate (Moss et al., 2025). Beyond LSBO, agentic and language-based optimizers are also reshaping the biological design landscape (Maus et al., 2026).

In-context learning for BO. PFNs learn posterior prediction from synthetic task distributions in a single forward pass (Muller et al., 2021; Hollmann et al., 2025; Qu et al., 2026; Grinsztajn et al., 2026). PFNs4BO showed that such models can serve as BO surrogates (Muller et al., 2023), while α-PFN moved toward amortizing information-theoretic acquisition functions (Viering et al., 2025). GIT-BO showed that TFM surrogates can scale to hundreds of dimensions when coupled with active-subspace search (Yu et al., 2026). More recently, fully amortized policies suggest that surrogate fitting and acquisition design may eventually be replaced end to end in some BO settings (Maraval et al., 2023; Zhang et al., 2025b; 2026; Blumer et al., 2026). These results motivate our focus on LSBO, where the pretraining distribution becomes especially important.

3. Problem statement

Let $\mathcal{X}$ denote a structured design space, such as molecules or protein sequences, and let $f : \mathcal{X} \to \mathbb{R}$ be an expensive black-box objective. We consider LSBO with a learned encoder-decoder representation, given by an encoder $e_\psi : \mathcal{X} \to \mathcal{Z}$ and a decoder $p_\psi(\boldsymbol{x} \mid \boldsymbol{z})$ over a latent space $\mathcal{Z} \subset \mathbb{R}^d$. LSBO operates by searching over latent codes rather than directly over structured objects. When the decoder is stochastic, the latent optimization problem is

$$\boldsymbol{z}^\star \in \operatorname*{argmax}_{\boldsymbol{z}\in\mathcal{Z}} g(\boldsymbol{z}), \qquad g(\boldsymbol{z}) := \mathbb{E}_{\boldsymbol{x}\sim p_\psi(\cdot\mid\boldsymbol{z})}[f(\boldsymbol{x})]. \quad (1)$$

In the deterministic case, this reduces to $g(\boldsymbol{z}) = f(h_\psi(\boldsymbol{z}))$, where h_ψ is the decoder map. Given the history

$$\mathcal{H}_t = \{(\boldsymbol{z}_i, y_i)\}_{i=1}^t, \;\; \boldsymbol{z}_i = e_\psi(\boldsymbol{x}_i), \;\; y_i = f(\boldsymbol{x}_i) + \varepsilon_i, \quad (2)$$

LSBO fits a probabilistic surrogate $q_t(y \mid \boldsymbol{z}, \mathcal{H}_t)$ and uses it to define an acquisition rule α_t, for instance expected improvement or Thompson sampling (Garnett, 2023). The next latent query is selected as

$$\boldsymbol{z}_{t+1} \in \operatorname*{argmax}_{\boldsymbol{z}\in\mathcal{Z}} \alpha_t(\boldsymbol{z}), \quad (3)$$

and is then decoded and evaluated:

$$\boldsymbol{x}_{t+1} \sim p_\psi(\cdot \mid \boldsymbol{z}_{t+1}), \qquad y_{t+1} = f(\boldsymbol{x}_{t+1}) + \varepsilon_{t+1}. \quad (4)$$

Classically, the surrogate q_t is taken to be the posterior predictive distribution of a Gaussian process fitted on $\mathcal{H}_t$ (Rasmussen and Williams, 2006). In practice, this surrogate is often embedded in a broader pipeline that includes trust-region heuristics and retraining of the generative model.

Our goal is to replace q_t with a pretrained in-context model

$$q_\theta(y \mid \boldsymbol{z}, \mathcal{H}_t), \quad (5)$$

used as the predictive model inside the LSBO loop. The parameters θ are learned offline through pretraining on a distribution of synthetic regression tasks. At test time, given the observed history $\mathcal{H}_t$ and a set of query latent points, the model outputs predictive posterior quantities for those queries. In analogy with Bayesian inference, the synthetic pretraining distribution plays the role of a flexible implicit prior over regression problems. The key question is therefore how to choose this distribution so that the resulting predictions are adapted to LSBO.

4. Method

Our approach has two components: a BO-aware synthetic adaptation distribution on the latent space of a pretrained molecular VAE, and a continued-pretrained TabPFN-3 surrogate used inside a standard LSBO loop.

4.1. BO-aware synthetic task generation

We start from the GuacaMol molecular corpus $\mathcal{M} = \{\boldsymbol{x}_m\}_{m=1}^M$ encoded by a pretrained SELFIES VAE, yielding latent codes $\boldsymbol{z}_m = e_\psi(\boldsymbol{x}_m)$ (Brown et al., 2019; Maus et al., 2022).

Synthetic target property creation. Rather than drawing synthetic objectives from generic regression priors, we construct them from a small library of molecular *base tasks*, such as logP, QED, similarity, and rediscovery. For each synthetic episode, we sample a small subset of these tasks and evaluate them on the molecular pool. Writing

$$\boldsymbol{s}(\boldsymbol{x}) = \big(s_1(\boldsymbol{x}), \dots, s_K(\boldsymbol{x})\big) \in \mathbb{R}^K \tag{6}$$

for the resulting vector of selected base-task values, we then construct a synthetic scalar objective by applying either a linear or nonlinear combiner:

$$f_\tau^{\text{lin}}(\boldsymbol{x}) = \boldsymbol{w}_\tau^\top \boldsymbol{s}_\tau(\boldsymbol{x}), \qquad \boldsymbol{w}_\tau \sim \text{Dirichlet}(\mathbf{1}_K), \tag{7}$$

$$f_\tau^{\text{nlin}}(\boldsymbol{x}) = g_\tau\big(\boldsymbol{s}_\tau(\boldsymbol{x})\big), \tag{8}$$

where $g_\tau : \mathbb{R}^K \to \mathbb{R}$ is a randomly sampled MLP. This follows the general philosophy of synthetic task generation for TFMs while grounding the prior in molecular objectives that are closer to downstream LSBO than generic synthetic targets. Additional combiner families, including formula trees, and the full base-task library are described in Appendix A. Importantly, the evaluation benchmarks are excluded from the base-task library, so downstream objectives do not appear in the pretraining distribution.

Context dataset sampler. The context sampler is designed to reflect BO rather than i.i.d. regression. Uniformly sampled contexts contain mostly mediocre molecules and provide limited information about high-value objective regions. We therefore bias sampling toward promising points. Given a synthetic objective f_τ, we evaluate it on the GuacaMol pool $\mathcal{M}$ and induce a Boltzmann distribution

$$p_\tau(m) = \frac{\exp\big(f_\tau(\boldsymbol{x}_m)/T_\tau\big)}{\sum_j \exp\big(f_\tau(\boldsymbol{x}_j)/T_\tau\big)}, \tag{9}$$

where T_τ is sampled per episode (Appendix A.2). This biases contexts toward high-value regions while retaining diversity across tasks.

4.2. LSBO-continued-pretrained TabPFN-3 surrogate

Rather than pre-training from scratch using our proposed molecular synthetic prior, our plug-in surrogate is initialized from a pretrained TabPFN-3 checkpoint (Grinsztajn et al., 2026). The rationale is that molecular LSBO episodes are highly specialized: they involve a complex mapping involving VAE latent codes and objective ladnscapes induced by molecular property scores, and value-biased contexts.

Algorithm 1 Synthetic BO episode generation for LILBO

Require: Molecular pool $\mathcal{M} = \{\boldsymbol{x}_m\}_{m=1}^M$, fixed encoder e_ψ, descriptor map $\boldsymbol{s}$ in Eq. (6)

1: Encode the pool once: $\boldsymbol{z}_m \leftarrow e_\psi(\boldsymbol{x}_m)\ \forall \boldsymbol{x}_m \in \mathcal{M}$
2: Sample a synthetic objective f_τ from Eqs. (7)–(8)
3: Evaluate pool labels: $y_m^\tau \leftarrow f_\tau(\boldsymbol{x}_m)$
4: Sample indices $I_\tau \sim p_\tau^{\otimes n_{\text{ctx}}+n_{\text{qry}}}$ via Eq. (9), and split $I_\tau = I_\tau^{\text{ctx}} \cup I_\tau^{\text{qry}}, \quad |I_\tau^{\text{ctx}}| = n_{\text{ctx}}, \quad |I_\tau^{\text{qry}}| = n_{\text{qry}}$.
5: Form the context and query sets $\mathcal{C}_\tau = \{(\boldsymbol{z}_i, y_i^\tau) : i \in I_\tau^{\text{ctx}}\}, \mathcal{Q}_\tau = \{(\boldsymbol{z}_i, y_i^\tau) : i \in I_\tau^{\text{qry}}\}$.
6: Emit the training episode $\mathcal{E}_\tau = (\mathcal{C}_\tau, \mathcal{Q}_\tau)$.
7: Update q_θ with the TabPFN-3 predictive loss and L_2-SP regularization toward θ_0 via Eq. (10).

Training only on such episodes might give full control over the molecular distribution, but could also discard the broad regression prior already learned by large-scale TFM pretraining. Continued pretraining provides a compromise: the model can adapt to the geometry and target distributions of molecular LSBO while retaining its general-purpose notion of regression tasks. This reasoning is akin to that of Real-TabPFN (Garg et al., 2025), which shows that further pretraining can improve tabular foundation models on downstream data distributions. We operationalize this idea with their continued-pretraining objective. Let θ_0 denote the original checkpoint. The continued-pretraining objective is

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{CE}}(\theta) + \frac{\lambda_{\text{L2-SP}}}{2}\|\theta - \theta_0\|_2^2, \tag{10}$$

where the second term limits drift from the pretrained model and mitigates catastrophic forgetting (Kirkpatrick et al., 2017). Full optimization details are given in Appendix B.1. Algorithm 1 summarizes the training loop. We train for 8,000 continued-pretraining steps, exposing the model to approximately 500k molecular LSBO episodes. Full optimization details are given in Table 3.

At test time we keep the LSBO loop unchanged and replace only the surrogate. For each candidate batch $\mathcal{Z}_t^{\text{cand}}$, typically produced by local search, we retrieve up to n_{ctx} nearby or high-value observations from the BO history and score the candidates with the pretrained model. The next query is then selected by Thompson sampling,

$$\widetilde{f}_t(\boldsymbol{z}) \sim q_\theta(\cdot \mid \boldsymbol{z}, \widetilde{\mathcal{H}}_t), \qquad \boldsymbol{z}_{t+1} = \operatorname*{argmax}_{\boldsymbol{z} \in \mathcal{Z}_t^{\text{cand}}} \widetilde{f}_t(\boldsymbol{z}). \tag{11}$$

In the current instantiation, continued-pretraining is performed on the latent space of a frozen VAE. At test time, however, we keep the LSBO pipeline unchanged and allow periodic VAE retraining, since this yields substantial benefits (Maus et al., 2022). Hence, the latent space drifts during optimization, creating a mismatch between the implicit prior learned during pretraining and the tasks encountered at deployment. This is a current limitation of our approach.

5. Experiments

Benchmarks and protocol. We use the SELFIES VAE of Maus et al. (2022), pretrained on the 1.27M GuacaMol molecules (Brown et al., 2019), as the base generator for synthetic pretraining. Crucially, downstream evaluation is performed on held-out GuacaMol benchmark objectives that were not seen during pretraining, thereby testing generalization to unseen objectives. Our main suite contains eight molecular multi-property optimization tasks: Osimertinib, Fexofenadine, Ranolazine, Zaleplon, Amlodipine, Perindopril, Median Molecules 1 and Median Molecules 2 (Brown et al., 2019). Each run starts from an initial set of 100 randomly sampled molecules, and we report the best value found versus oracle calls over 10 random seeds. The setup follows (Lee et al., 2023) and is designed to emulate real-world lead optimization.

Baselines and implementation details. We compare against representative LSBO baselines, including GP-LSBO (Eriksson et al., 2019; Gomez-Bombarelli et al., 2018), LOLBO (Maus et al., 2022), CoBO (Lee et al., 2023), InvBO (Chu et al., 2024), NF-BO (Lee et al., 2025) and random latent search. To isolate the effect of LSBO-specific continued pretraining, we also include an off-the-shelf TabPFN-3 plug-in baseline (Grinsztajn et al., 2026). The latter uses the same BO loop and candidate batches as our method, but without our molecular LSBO adaptation. All baselines select the next query via Thompson sampling over a common candidate set $\mathcal{Z}_t^{\mathrm{cand}}$, whose generation is detailed in Appendix B.3. Moreover, all baselines except LSBO and random perform VAE retraining. For non-TFM baselines this entails joint retraining of the surrogate and the VAE, whereas our TFM-based methods replace only the surrogate. This tests whether a pretrained in-context model remains effective under the latent drift induced by strong LSBO pipelines.

5.1. Pretraining-prior diagnostics

Before turning to BO performance, we introduce two diagnostics of the synthetic adaptation distribution, designed to assess whether it covers the latent-space regions and objective-space variability relevant to LSBO.

Latent-space coverage of sampled episodes. For a fixed objective f_τ, we define three sets of latent points:

1. *Uniform pool samples:* $\mathcal{Z}_\tau^{\mathrm{uni}} = \{\boldsymbol{z}_l\}_{l=1}^L$, where the corresponding molecules are sampled uniformly from the GuacaMol pool $\mathcal{M}$.
2. *Value-biased pool samples:* $\mathcal{Z}_\tau^{\mathrm{bias}} = \{\boldsymbol{z}_l\}_{l=1}^L$, where the corresponding molecules are sampled from the same pool $\mathcal{M}$ with the value-biased distribution in Equation 9.
3. *Local perturbations:* $\mathcal{Z}_\tau^{\mathrm{pert}} = \{\boldsymbol{z}_i + r\boldsymbol{\xi}_i : \boldsymbol{z}_i \in \mathcal{Z}_\tau^{\mathrm{uni}} \cup \mathcal{Z}_\tau^{\mathrm{bias}}\}$, with $\boldsymbol{\xi}_i \overset{\text{i.i.d.}}{\sim} \mathrm{Unif}(\mathbb{B}_d)$ and r small.

We fit a two-dimensional UMAP to $\mathcal{Z}_\tau^{\mathrm{all}} := \mathcal{Z}_\tau^{\mathrm{uni}} \cup \mathcal{Z}_\tau^{\mathrm{bias}} \cup \mathcal{Z}_\tau^{\mathrm{pert}}$ and display a single embedding in which marker shape indicates provenance and color indicates objective value. This visualization checks whether value-biased sampling and local perturbations populate latent regions that are relevant to BO, and whether these regions align with high-value parts of the landscape.

Figure 1 shows the representation for $L = 200$, so that 800 points are displayed. The main question is whether value-biased samples and their local perturbations are enriched in higher-value regions relative to uniform samples. In the current plots, the objective-value coloring does suggest that the landscapes differ across tasks, with some objectives exhibiting more localized high-value regions than others. At the same time, provenance differences remain moderate, which indicates that the perturbation mechanism stays close to the latent manifold. This also means that the figure alone does not fully characterize the gap between pretraining episodes and test-time BO candidate distributions.

Objective-space coverage of synthetic tasks.

We also visualize the diversity of the synthetic objective family. Let $\mathcal{P} = \{\boldsymbol{x}_j\}_{j=1}^J$ be a fixed molecular probe set. It is not necessary that $\mathcal{P} \subset \mathcal{M}$. Each objective f_τ is represented by its vector of values on this probe set,

$$\boldsymbol{v}_\tau = \big(f_\tau(\boldsymbol{x}_1), \ldots, f_\tau(\boldsymbol{x}_J)\big) \in [0,1]^m. \tag{12}$$

We compute these vectors for both synthetic pretraining objectives and held-out benchmark objectives, then embed them in two dimensions with UMAP. The comparison is inherently probe-set dependent and should be interpreted carefully. Still, it shows whether the synthetic prior covers a broad region of molecular objective space.

Figure 2 displays the representation for $|\mathcal{P}| = 5000$ and 5000 synthetic tasks, as well as the held-out and base tasks. The synthetic tasks form continuous clusters around the base objectives, as expected from linear and nonlinear combinations of molecular descriptors. Importantly, the held-out GuacaMol benchmarks do not appear isolated, lying close to the synthetic task mass. This supports the intended role of the prior: exposing the model to related molecular optimization landscapes. For completeness, base objective distributions and representative synthetic objective distributions are shown in Figures 4 and 5.

5.2. Results

We next evaluate whether our proposed pretraining distribution improves downstream optimization on held-out molecular objectives. Figure 3 reports the best value found as a

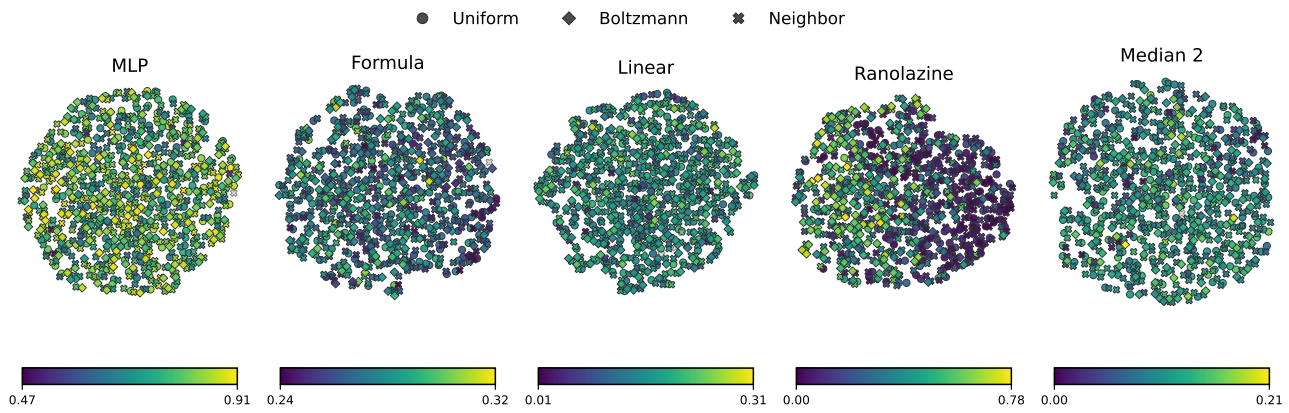

Figure 1. **Latent-space coverage of pretraining episodes.** UMAP embeddings for 3 synthetic objectives and 2 held-out benchmarks.

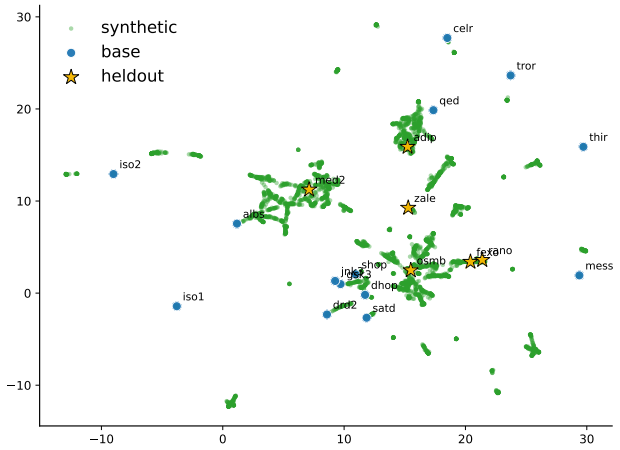

Figure 2. **Objective-space coverage of the synthetic prior.** Each objective is evaluated on the same probe set $\mathcal{P}$ and represented by the resulting value vector (Equation 12), embedded with UMAP.

function of oracle evaluations, and Table 1 shows baseline ranking aggregated over tasks and seeds.

Tabular foundation models are strong plug-in surrogates for LSBO. Figure 3 shows that tabular foundation models can already serve as competitive surrogates for LSBO: both off-the-shelf TabPFN-3 and LILBO are in the leading group across the held-out molecular objectives. Importantly, the optimization pipeline is otherwise kept fixed, including candidate generation, acquisition rule, and VAE retraining schedule, so the comparison isolates the surrogate replacement itself. These results suggest that in-context probabilistic surrogates are a viable alternative to GPs in modern LSBO pipelines.

LSBO-aware continued pretraining improves average performance against TabPFN-3. Table 1 shows that LILBO slightly improves over the TabPFN-3 plug-in baseline overall, with average ranks of 2.64 and 2.94, respectively. This gain is modest but notable, as our adaptation is intentionally lightweight compared with the original TabPFN-3 pretraining. The checkpoint was trained on more than 8 trillion synthetic tokens (Grinsztajn et al., 2026), whereas our molecular adaptation uses only about 500k LSBO episodes of length 1024. On a similar note, TabICLv2 was pretrained for 550K synthetic-data steps with batch size 64, corresponding to roughly 35M synthetic datasets (Qu et al., 2026). Moreover, our adaptation distribution is far more specialized, operating on 256-dimensional SELFIES-VAE latents with 100–600 context rows, compared with the broad row–feature regimes targeted by TabPFN-3. Overall, this motivates anchoring the adapted model to the original checkpoint: the goal is to specialize the surrogate to molecular LSBO while preserving the broad regression prior learned during large-scale pretraining.

Method	Average rank
LILBO	2.64 ± 1.56
InvBO	2.88 ± 1.62
TabPFN-3	2.94 ± 1.64
NFBO	2.96 ± 1.69
LSBO	5.86 ± 1.75
LolBO	5.90 ± 1.37
Random	6.25 ± 1.68
CoBO	6.34 ± 1.19

Table 1. **Average rank of the methods across all tasks.** Ranks were computed separately for each seed, the table reports the mean rank $\pm$ 1 std across seeds.

6. Conclusion

We studied whether in-context surrogates can be adapted to latent-space Bayesian optimization through task-matched pretraining. Our main empirical message is that tabular foun-

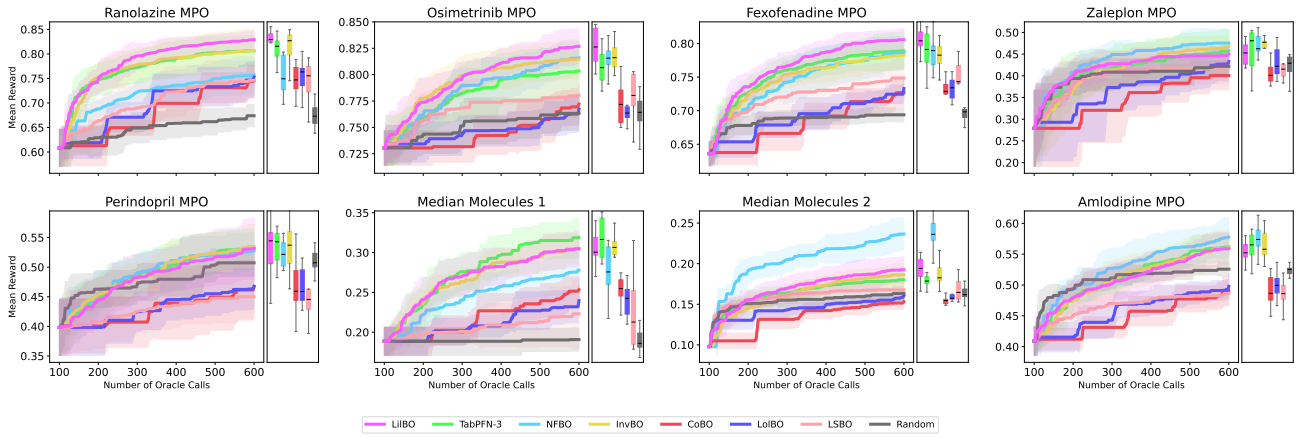

Figure 3. **Main results.** Best value found over oracle calls and distribution over the final recommendation on held-out molecular objectives. Mean $\pm$ 1 std computed across 10 seeds.

dation models are already competitive surrogates for LSBO when inserted into an otherwise fixed optimization pipeline, and that LSBO-aware continued pretraining yields additional gains. While LILBO does not uniformly dominate every task or every specialized LSBO baseline, it achieves the best average rank in our benchmark suite, supporting the central hypothesis of this work: For tabular foundation models to be effective in LSBO, their adaptation distribution should reduce the mismatch between generic tabular pretraining tasks and the structure of latent space objectives and BO histories encountered at deployment.

Perspectives. Several limitations suggest natural next steps. First, pretraining episodes should be made more BO-like by simulating short optimization trajectories rather than sampling only from the molecular pool, so that training contexts include high-value molecules beyond those already present in GuacaMol. This would better match deployment histories and is consistent with recent work (Li et al., 2025). Second, the prior should account for the latent drift induced by periodic VAE retraining, so that the in-context surrogate is pretrained not only on synthetic objectives but also on evolving latent representations. Third, multi-objective LSBO is a natural extension, particularly in drug design where one must balance properties such as potency, toxicity, and solubility. Finally, the same framework should extend beyond small-molecule design to other structured domains, including protein and antimicrobial peptide design, provided a suitable generative model supplies a latent representation.

References

Nicolas Samuel Blumer, Julien Martinelli, and Samuel Kaski. In-context black-box optimization with unreliable feedback. *arXiv preprint arXiv:2605.06187*, 2026.

Nathan Brown, Marco Fiscato, Marwin H. S. Segler, and Alain C. Vaucher. Guacamol: Benchmarking models for de novo molecular design. *Journal of Chemical Information and Modeling*, 2019.

Jaewon Chu, Jinyoung Park, Seunghun Lee, and Hyunwoo J. Kim. Inversion-based latent bayesian optimization. In *Advances in Neural Information Processing Systems*, 2024.

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

Anurag Garg, Muhammad Ali, Noah Hollmann, Lennart Purucker, Samuel Müller, and Frank Hutter. Realtabpfn: Improving tabular foundation models via continued pre-training with real-world data. In *1st ICML Workshop on Foundation Models for Structured Data*, 2025. URL https://openreview.net/forum?id=BtEiqKsIMw.

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

Rafael Gomez-Bombarelli, Jennifer N. Wei, David Duvenaud, Jose Miguel Hernandez-Lobato, Benjamin Sanchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D. Hirzel, Ryan P. Adams, and Alan Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, 4(2):268–276, 2018.

Léo Grinsztajn, Klemens Flöge, Oscar Key, Felix Birkel, Philipp Jund, Brendan Roof, Mihir Manium, Shi Bin, Magnus Bühler, Anurag Garg, et al. Tabpfn-3: Technical report. *arXiv preprint arXiv:2605.13986*, 2026.

Nate Gruver, Samuel Stanton, Nathan Frey, Tim G. J. Rudner, Isidro Hotzel, Julien Lafrance-Vanasse, Arvind Rajpal, Kyunghyun Cho, and Andrew Gordon Wilson. Protein design with guided discrete diffusion. In *Advances in Neural Information Processing Systems*, 2024.

Noah Hollmann, Samuel Muller, Lennart Purucker, Arjun Krishnakumar, Max Korfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637 (8045):319–326, 2025.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Seunghun Lee, Jaewon Chu, Sihyeon Kim, Juyeon Ko, and Hyunwoo J. Kim. Advancing bayesian optimization via learning correlated latent space. In *Advances in Neural Information Processing Systems*, 2023.

Seunghun Lee, Jinyoung Park, Jaewon Chu, Minseo Yoon, and Hyunwoo J. Kim. Latent bayesian optimization via autoregressive normalizing flows. In *International Conference on Learning Representations*, 2025.

Diantong Li, Kyunghyun Cho, and Chong Liu. None to optima in few shots: Bayesian optimization with mdp priors. *arXiv preprint arXiv:2511.01006*, 2025.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

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

Alexandre Maraval, Matthieu Zimmer, Antoine Grosnit, and Haitham Bou Ammar. End-to-end Meta-Bayesian Optimisation with Transformer Neural Processes. *Advances in Neural Information Processing Systems*, 2023.

Natalie Maus, Haydn Jones, Juston Moore, Matt J. Kusner, John Bradshaw, and Jacob Gardner. Local latent space bayesian optimization over structured inputs. In *Advances in Neural Information Processing Systems*, 2022.

Natalie Maus, Yimeng Zeng, Haydn Thomas Jones, Yining Huang, Gaurav Ng Goel, Alden Rose, Kyurae Kim, Hyun-Su Lee, Marcelo Der Torossian Torres, Fangping Wan, Cesar de la Fuente-Nunez, Mark Yatskar, Osbert Bastani, and Jacob R. Gardner. Purely agentic black-box optimization for biological design. *arXiv preprint arXiv:2601.22382*, 2026.

Henry B. Moss, Sebastian W. Ober, and Tom Diethe. Return of the latent space cowboys: Re-thinking the use of vaes for bayesian optimisation of structured spaces. In *International Conference on Machine Learning*, 2025.

Samuel Muller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do bayesian inference. *arXiv preprint arXiv:2112.10510*, 2021.

Samuel Muller, Matthias Feurer, Noah Hollmann, and Frank Hutter. PFNs4BO: In-context learning for Bayesian optimization. In *International Conference on Machine Learning*, 2023.

Samuel Muller, Arik Reuter, Noah Hollmann, David Rügamer, and Frank Hutter. Position: The future of bayesian prediction is prior-fitted. *arXiv preprint arXiv:2505.23947*, 2025.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. Tabiclv2: A better, faster, scalable, and open tabular foundation model. *arXiv preprint arXiv:2602.11139*, 2026.

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

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*, 2022.

Austin Tripp, Erik Daxberger, and Jose Miguel Hernandez-Lobato. Sample-efficient optimization in the latent space of deep generative models via weighted retraining. In *Advances in Neural Information Processing Systems*, 2020.

Tom Viering, Steven Adriaensen, Herilalaina Rakotoarison, Samuel Muller, Carl Hvarfner, Frank Hutter, and Eytan Bakshy. α-pfn: In-context learning entropy search. In *ICLR Workshop on Frontiers in Probabilistic Inference*, 2025.

Tuan A. Vu, Julien Martinelli, and Harri Lähdesmäki. Time-aware latent space bayesian optimization. *arXiv preprint arXiv:2603.00935*, 2026.

Li Xuhong, Yves Grandvalet, and Franck Davoine. Explicit inductive bias for transfer learning with convolutional networks. In *International conference on machine learning*, pages 2825–2834. PMLR, 2018.

Rosen Ting-Ying Yu, Cyril Picard, and Faez Ahmed. GIT-BO: High-dimensional bayesian optimization with

tabular foundation models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=9iTdKS4SRQ.

Qiong Zhang, Yan Shuo Tan, Qinglong Tian, and Pengfei Li. Tabpfn: One model to rule them all? *arXiv preprint arXiv:2505.20003*, 2025a.

Xinyu Zhang, Daolang Huang, Samuel Kaski, and Julien Martinelli. PABBO: Preferential Amortized Black-Box Optimization. *International Conference on Learning Representations*, 2025b.

Xinyu Zhang, Conor Hassan, Julien Martinelli, Daolang Huang, and Samuel Kaski. In-context multi-objective optimization. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=odmeUlWta8.

A. Synthetic pretraining distribution

A.1. Descriptor library and objective families

To construct a synthetic objective, we first draw $K \sim \mathcal{U}(\{1, 2, 3, 4\})$, and then sample K base objectives from Table 2.

Index	Objective function
1	albuterol_similarity
2	mestranol_similarity
3	celecoxib_rediscovery
4	thiothixene_rediscovery
5	troglitazone_rediscovery
6	isomer_c7h8n2o2
7	isomer_c9h10n2o2pf2cl
8	sa_tdc
9	drd2_docking
10	rdkit_qed
11	deco_hop
12	scaffold_hop
13	gsk3_beta
14	jnk3

Table 2. Descriptors used for synthetic prior tasks.

For a molecule $\boldsymbol{x}$, let $\boldsymbol{s}(\boldsymbol{x}) = \big(s_1(\boldsymbol{x}), \ldots, s_K(\boldsymbol{x})\big)$ denote the selected objective values. We then combine them using a linear-weighted combiner, a random MLP, or a formula tree.

Linear combiner. The linear-weighted family combines the selected molecular objectives through a random weighted average. We draw the weights from a symmetric Dirichlet distribution,

$$\mathbf{w} \sim \mathrm{Dirichlet}(\mathbf{1}_K), \qquad f_{\text{lin}}(\boldsymbol{x}) = \sum_{i=1}^{K} w_i s_i(\boldsymbol{x}).$$

Random MLP combiner. The random MLP family introduces smooth nonlinear interactions between the selected molecular objectives by applying a sampled function $g_\tau : \mathbb{R}^K \to \mathbb{R}$ to the vector $\boldsymbol{s}(\boldsymbol{x})$. Concretely,

$$f_{\text{mlp}}(\boldsymbol{x}) = g_\tau\big(\boldsymbol{s}(\boldsymbol{x})\big),$$

where g_τ is an L-hidden-layer MLP with width H, activation ϕ, and block-sparse Gaussian weights. We sample

$$L \in \{1, 2, 3\}, \qquad H \in \{16, 32, 64\}, \qquad \phi \in \{\tanh, \text{ReLU}, \text{ELU}, \sigma\}.$$

Writing $\boldsymbol{h}^{(0)} = \boldsymbol{s}(\boldsymbol{x})$, the hidden states satisfy

$$\boldsymbol{h}^{(\ell)} = \phi_\ell\Big(W_\ell \boldsymbol{h}^{(\ell-1)} + \boldsymbol{b}_\ell\Big), \qquad \ell = 1, \ldots, L,$$

and the final output is

$$f_{\text{mlp}}(\boldsymbol{x}) = g_\tau\big(\boldsymbol{s}(\boldsymbol{x})\big) = \boldsymbol{w}_{\text{out}}^\top \boldsymbol{h}^{(L)} + b_{\text{out}}.$$

Formula-tree combiner. The formula-tree family introduces compositional objectives with gates, sharp thresholds, and multiplicative interactions. We sample a random binary expression tree f_{tree} whose leaves are indexed by the selected objectives $s_1, \ldots, s_K$. Internal nodes are assigned binary operations

$$\mathcal{B} = \{+, \times, \text{gate}\}, \qquad \text{gate}(a, b) = \mathbf{1}[a > 0] \cdot b,$$

and unary operations

$$\mathcal{U} = \{\mathrm{id},\ |\cdot|,\ \sin,\ \sigma,\ \mathrm{ReLU},\ \sqrt{\cdot}\}.$$

Writing $F_v(\boldsymbol{s})$ for the value at node v, we define the tree recursively as

$$F_v(\boldsymbol{s}) = \begin{cases} s_i, & \text{if } v \text{ is a leaf labeled } i, \\ u_v\Big(b_v\big(F_{v_{\mathrm{left}}}(\boldsymbol{s}),\ F_{v_{\mathrm{right}}}(\boldsymbol{s})\big)\Big), & \text{if } v \text{ is an internal node,} \end{cases}$$

where $b_v \in \mathcal{B}$ and $u_v \in \mathcal{U}$. The final objective is then

$$f_{\mathrm{tree}}(\boldsymbol{x}) = F_{\mathrm{root}}(\boldsymbol{s}(\boldsymbol{x})).$$

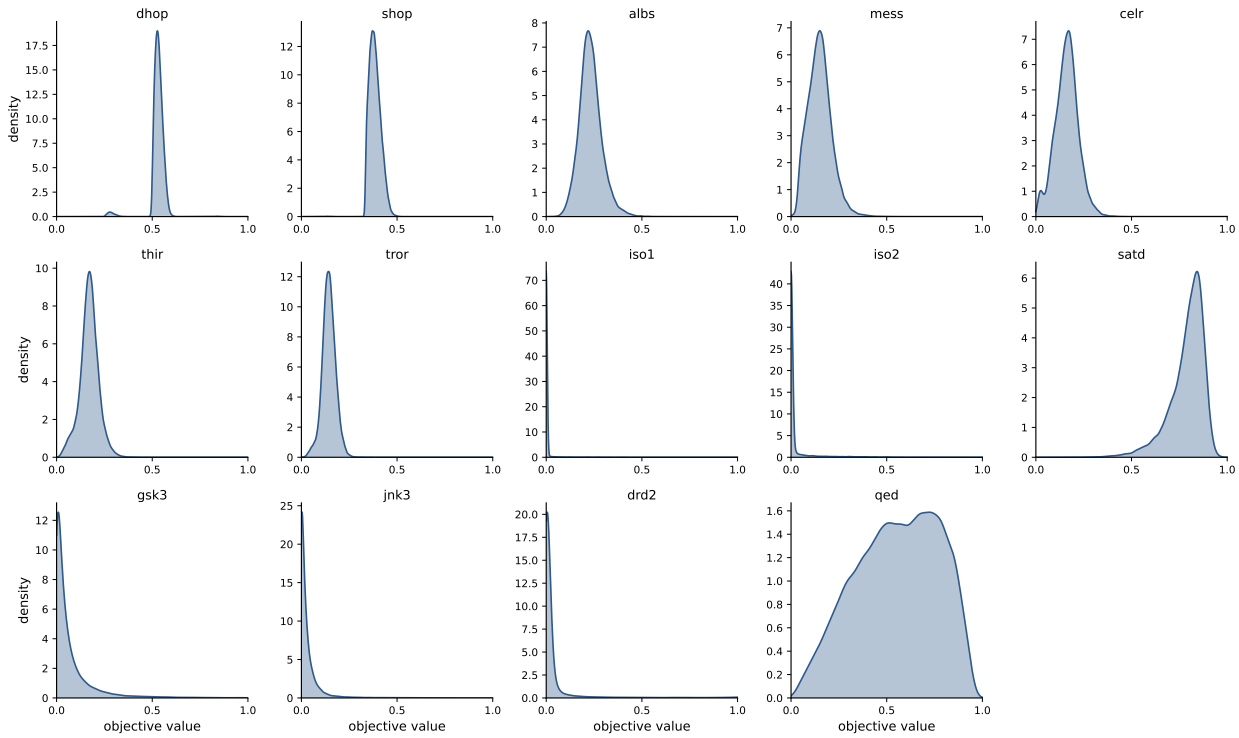

Figure 4. **Base objective distributions.** Kernel density estimates of the normalized base objectives used to construct the synthetic prior.

A.2. Context and query sampling

We evaluate the synthetic objective on the GuacaMol molecular pool to obtain

$$y_n = f\big(\boldsymbol{s}(\boldsymbol{x}_n)\big), \qquad n = 1, \ldots, N,$$

which induces a Boltzmann distribution over the molecular pool. To generate a synthetic dataset, we draw a temperature T log-uniformly from

$$\log T \sim \mathrm{Uniform}(\log t_{\min}, \log t_{\max}), \qquad [t_{\min}, t_{\max}] = [0.01,\ 10].$$

The induced distribution is

$$\pi_n(T) = \frac{\exp(y_n/T)}{\sum_{m=1}^{N} \exp(y_m/T)}, \qquad n = 1, \ldots, N.$$

The resulting synthetic dataset is

$$\mathcal{D} = \{(\boldsymbol{x}_{i_r},\ y_{i_r})\}_{r=1}^{m}.$$

This dataset is then partitioned into context and target points for PFN training.

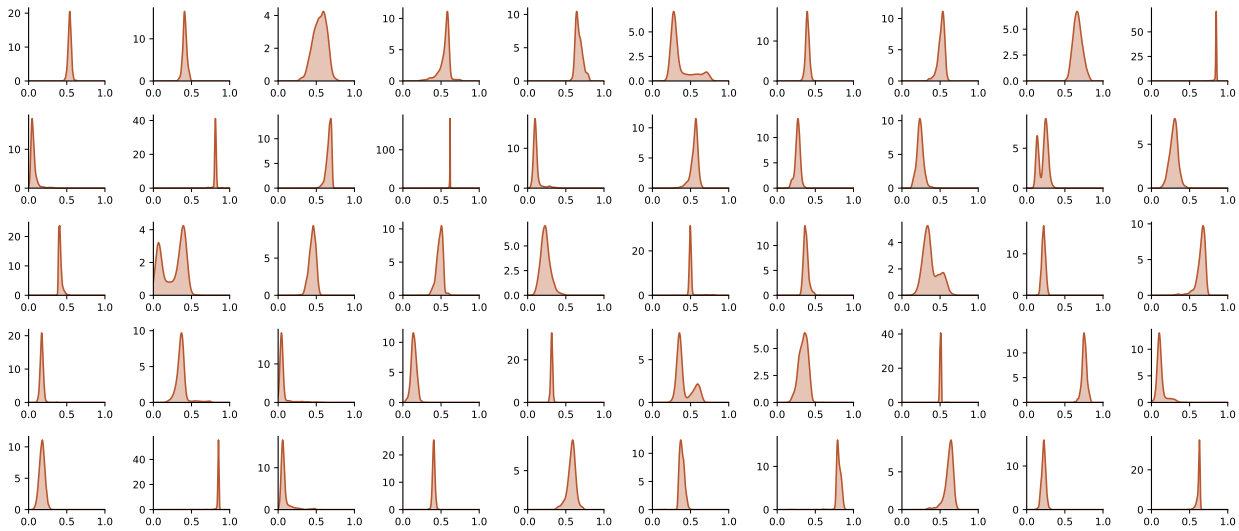

Figure 5. **Synthetic objective distributions.** Kernel density estimates of representative normalized synthetic objectives sampled from the pretraining prior.

B. Implementation details

B.1. Continued-pretraining configuration

Following (Garg et al., 2025), we adopt a two-stage approach. In the first stage, we start from the original TabPFN-3 (Grinsztajn et al., 2026) checkpoint, which has already been pretrained on a broad collection of synthetic tabular datasets. We then continue pretraining from this checkpoint using our own synthetic datasets described earlier. We keep the same architecture as TabPFN-3 and use a learning rate of 3×10^{-7} with the AdamW optimizer (Loshchilov and Hutter, 2017), combined with a linear warm-up phase and a subsequent cosine annealing schedule (Loshchilov and Hutter, 2016). The complete training configuration is provided in Table 3. To mitigate catastrophic forgetting (Kirkpatrick et al., 2017), we incorporate an L2-SP regularization term (Xuhong et al., 2018) into the training objective, preventing the model parameters from drifting too far from the weights of the original pretrained model:

$$\mathcal{L} = \mathcal{L}_{CE} + \frac{\lambda_{L2-SP}}{2} \|\theta - \theta_0\|_2^2,$$

where θ_0 denotes the parameters of the initial checkpoint and $\| \cdot \|_2$ denotes the ℓ_2 norm. The final model was trained for 8,000 steps, during which it was exposed to approximately 500k datasets.

Parameter	Value
Base model	Pretrained TabPFN-v3 checkpoint
Training data	Molecular synthetic datasets
Input features	256-d SELFIES-VAE latents
Sequence length	1024
Context rows	100–600
Optimizer	AdamW
LR	3×10^{-7}
Weight decay	0.01
Batch size	64
Steps	8000
$\lambda_{\mathrm{L2-SP}}$	0.003

Table 3. **Training configuration used for continued pretraining.**

B.2. Baselines.

We compare LILBO to representative latent-space optimization baselines:

- **Random search**: samples latent codes from the proposal distribution and decodes them without fitting a surrogate.
- **LSBO** (Eriksson et al., 2019): latent-space BO with a Gaussian-process surrogate over latent codes,

$$g(\boldsymbol{z}) \sim \mathcal{GP}(0, k(\boldsymbol{z}, \boldsymbol{z}')),$$

  combined with adaptive local trust regions that restrict candidate generation to a neighborhood of the current incumbent.
- **LOL-BO** (Maus et al., 2022): local latent-space BO with periodic retraining of the generative model, designed to improve the latent representation around promising regions.
- **CoBO** (Lee et al., 2023): an LSBO method that learns a correlated latent space to better align latent geometry with the objective.
- **InvBO** (Chu et al., 2024): an inversion-based LSBO method that reduces mismatch between latent points and decoded molecules.
- **NF-BO** (Lee et al., 2025): a latent BO method based on autoregressive normalizing flows, designed to reduce the reconstruction-gap mismatch between latent points and decoded molecules; it also uses a candidate sampling strategy that adapts exploration to token importance.
- **TabPFN-3 plug-in** (Grinsztajn et al., 2026): replaces the GP surrogate with the released TabPFN-3 model, while using the same candidate batches, acquisition rule, and host LSBO loop as LILBO.

B.3. Candidate Sampling

For all probabilistic surrogates, candidates are selected by Thompson sampling over the same candidate set $\mathcal{Z}_t^{\text{cand}}$:

$$\tilde{g}_t \sim q_t(\cdot \mid \mathcal{H}_t), \qquad \boldsymbol{z}_{t+1} \in \operatorname*{argmax}_{\boldsymbol{z} \in \mathcal{Z}_t^{\text{cand}}} \tilde{g}_t(\boldsymbol{z}). \tag{13}$$

We adopt a multi-center trust-region-based local search BO method (Eriksson et al., 2019; Maus et al., 2022) in latent space. At each BO iteration, we select M high-scoring anchor points, each of which defines a promising local region for trust-region-based local search. A separate trust region is constructed around each center so that the acquisition step can explore multiple promising regions simultaneously.

A trust region centered at $\boldsymbol{\kappa} \in \mathbb{R}^d$ with side length $\rho > 0$ and weight $\boldsymbol{\omega} \in \mathbb{R}^d$ is

$$\mathcal{T}(\boldsymbol{\kappa}; \rho, \boldsymbol{\omega}) = \left\{ z \in \mathbb{R}^d : \boldsymbol{\kappa} - \tfrac{1}{2}\boldsymbol{\omega}\rho \leq z \leq \boldsymbol{\kappa} + \tfrac{1}{2}\boldsymbol{\omega}\rho \right\}. \tag{14}$$

Given centers $\{\boldsymbol{\kappa}_m\}_{m=1}^M$, this defines a collection of local regions $\{\mathcal{T}(\boldsymbol{\kappa}_m; \rho, \boldsymbol{\omega})\}_{m=1}^M$.

For each center, we generate the same number of candidates within its trust region by perturbing a subset of latent dimensions. The per-center candidate sets are pooled, and Thompson sampling from the surrogate predictive distribution selects the final batch of latent points.