BOREFT: MANIFOLD STEERING OF LANGUAGE MODELS FOR BLACK-BOX OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Language models are increasingly used as proposal models for black-box search, from program optimization to molecular design. Existing approaches typically improve proposals through iterative prompting or parameter updates, offering limited control over how completely and efficiently the model's search space is explored. Continuous optimization methods, such as Bayesian optimization, provide a principled way to search but require a suitable domain to operate over. To address this, we introduce BOReFT, which learns a compact, low-dimensional space of hidden-state interventions in a frozen language model, and uses this space as the search domain for Bayesian optimization with an external scoring function. Empirically, we find that the learned domain spans semantic regions and exhibits smoothness properties that support search. Theoretically, we show that semantic coverage and interpolation control the best score available in the learned space, and that decoding from this space yields a standard stochastic-bandit observation model for adaptive search. We evaluate BOReFT on the interpretable word search task "Semantle" and on three more real-world discovery tasks in *de novo* molecule property optimization. Compared to strong LLM baselines, BOReFT finds in Semantle a higher number of hidden targets and, on two out of three molecular objectives, achieves higher property scores. Consequently, our method provides a principled new bridge between discrete proposal spaces of LLM-based search and continuous black-box optimization.

1 INTRODUCTION

Generative search and discovery using language models proposes new discrete solutions, such as molecules or scientific hypotheses, under a limited budget of external evaluations. In several settings, the evaluation yields only scalar feedback; thus, efficiently finding a high-scoring proposal becomes a black-box optimization problem. Bayesian optimization provides a principled way to perform such a search, but operates over a continuous decision domain, whereas a language model generates discrete, stochastic sequences.

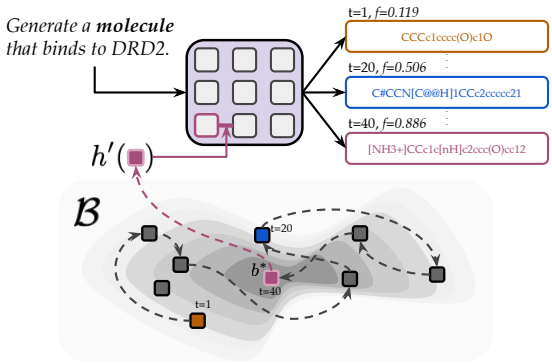


Figure 1: **BOReFT overview.** Bayesian optimization searches a learned intervention manifold $\mathcal{B}$, where each code b steers an LLM activation h' to generate a discrete candidate scored by objective f.

Existing approaches bridge this mismatch indirectly, either through prompting or by searching representations of candidate solutions (Yang et al., 2024; Agarwal et al., 2025a; Chen et al., 2024; Kristiadi et al., 2024; Ranković & Schwaller, 2025). They do not search a continuous space that can be decoded directly into solutions.

BOReFT learns such a search domain inside the frozen language model. From target sequences paired with descriptions, it learns a compact, low-dimensional manifold of hidden-state interventions. Each code in the learned manifold specifies an intervention that the frozen decoder maps to a distribution over discrete sequences, and the learned codes define a bounded, continuous domain

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

that Bayesian optimization searches over using only external scores (Fig. 1). This separates learning a useful generative search domain from the downstream procedure used to explore it.

Contributions. We make three contributions: **(i)** we introduce BOReFT, which learns a compact intervention manifold in a frozen language model and searches that fixed space with Bayesian optimization (§3); **(ii)** we decompose performance into a representational gap determined by the learned space and a search gap determined at inference, and characterize both through semantic coverage, interpolation, and standard stochastic-bandit observations (§4); **(iii)** we evaluate BOReFT on Semantle and molecular property optimization, and analyze how representation-training data, loss components, coverage, and interpolation affect downstream search (§5).

2 Related work

BO with language models. Classical latent-space BO searches continuous latent variables of a generative model and decodes them into structured objects (Gómez-Bombarelli et al., 2018; Maus et al., 2022). Kristiadi et al. (2024) use frozen or parameter-efficiently finetuned LLMs as feature extractors for molecular BO, and GOLLuM learns an LLM deep kernel jointly with a Gaussian process (Ranković & Schwaller, 2025). BOPRO fits a surrogate over external embeddings and proposes by prompting (Agarwal et al., 2025a), and GGOLLuM trains a LoRA actor from preferences supplied by a GP critic (Ranković & Schwaller, 2026). Notably, InstructZero applies BO to a low-dimensional soft-prompt that a frozen LLM consumes (Chen et al., 2024), but has subsequently been shown to be ineffective in practice (Agarwal et al., 2025a). BOReFT learns the continuous coordinates from target solutions and searches them inside the frozen generator.

Activation steering and representation interventions. Mechanistic-interpretability methods recover structure in activations, including sparse autoencoder features (Huben et al., 2024; Templeton et al., 2024), and steering methods intervene on those activations to change generation (Subramani et al., 2022; Rimsky et al., 2024; Zou et al., 2023). ReFT and LoReFT learn low-rank interventions for task adaptation (Wu et al., 2024). BOReFT uses that intervention form as the decision variables of black-box search. §C discusses test-time search and related interpretability methods.

3 BOReFT: Bayesian optimization via Representation Finetuning

BOReFT is a two-phase method for learning and searching a continuous intervention space in a frozen language model. During training, it learns a low-dimensional space from target sequences paired with auxiliary descriptions. At inference, the learned domain is fixed, and Bayesian optimization searches over it to optimize an external black-box objective. We view the learned space of hidden-state interventions as an *intervention manifold*, whose coordinates provide a continuous search interface to the language model.

3.1 Low-rank interventions as a search space

Rather than searching over prompts or model parameters, BOReFT finds continuous interventions on an internal representation built from LoReFT (Wu et al., 2024; Geiger et al., 2024). Let M be a frozen transformer language model with hidden dimension d, and let $h \in \mathbb{R}^d$ be the residual stream activation at a chosen layer and prompt position. For intervention rank $r \ll d$, LoReFT maps h to

$$h' \;=\; h + R^\top(Wh + b - Rh), \qquad R, W \in \mathbb{R}^{r\times d}, \quad b \in \mathbb{R}^r. \tag{1}$$

The rows of R are the subspace in which the intervention acts, while the base model is kept frozen. In BOReFT, we share R and W across a set of targets and treat the bias b as a target-dependent code. For fixed h, Eq. (1) is then an affine transformation in that subspace as a function of b,

$$h'(b) = h_0 + R^\top b, \qquad h_0 := h + R^\top(Wh - Rh), \tag{2}$$

The intervention is applied once, at the chosen prompt position, and the remaining layers decode from $h'(b)$ as usual. We write P_b^τ for the output distribution over sequences induced by $h'(b)$ at temperature τ (or P_b, when $\tau = 1$). The layer and position for each experiment are in §5.

3.2 LEARNING THE SEARCH SPACE

We train on pairs $\{(s_i, d_i)\}_{i=1}^n$, where s_i is a target sequence and d_i is an auxiliary natural-language description of the same target. For tasks in this work, on word search, s_i is a word and d_i is its dictionary-like natural language definition, while for molecular search, s_i is a SMILES string and d_i is its natural-language description from ChEBI-20 (Edwards et al., 2021).

Posterior over codes. Distribution-wise interventions cover a neighborhood in representation space and control behavior more reliably than pointwise interventions (Deng et al., 2025). We therefore use a VAE-style parameterization (Kingma & Welling, 2014), where each target–description pair is first passed through a frozen **semantic encoder**[1] ϕ, then through a shared trainable projection g_ψ that maps it to the parameters of a diagonal Gaussian over intervention codes,

$$q_i(b) = \mathcal{N}\big(\mu_i, \operatorname{diag}(\sigma_i^2)\big), \qquad (\mu_i, \sigma_i) = g_\psi\big(\phi(s_i, d_i)\big), \qquad \mu_i, \sigma_i \in \mathbb{R}^r. \tag{3}$$

Notice that since g_ψ predicts all posterior parameters, the target codes are not learned as unrelated free vectors but inherit structure from the semantic representations of ϕ. We use this as a geometric inductive bias, although we do not assume that semantic distances are preserved exactly. During training we sample $b_i = \mu_i + \sigma_i \odot \epsilon$ with $\epsilon \sim \mathcal{N}(0, I_r)$ using the reparameterization trick.

Trainable parameters. The base model M and semantic encoder ϕ remain frozen, while the intervention matrices R, W and the projection parameters ψ are set as trainable parameters. For each target, μ_i and σ_i are produced by the shared map $g_\psi(\phi(s_i, d_i))$, and the training code b_i is sampled from the resulting posterior q_i.

Training objective. For effective search in the continuous intervention space, we would like the observed targets to be recoverable from their codes, the search space to remain compact, and codes to retain useful related behavior rather than act dictionary lookups. We encourage these properties by constructing a loss objective that combines reconstruction, variational prior regularization, and on-policy self-distillation, i.e., for target i,

$$\mathcal{L}^{(i)} = \mathcal{L}^{(i)}_{\text{rec}} + \beta\, \mathcal{L}^{(i)}_{\text{prior}} + \lambda\, \mathcal{L}^{(i)}_{\text{distill}}. \tag{4}$$

Here, reconstruction anchors each training target in the learned space, the variational prior penalizes posteriors that move far from a shared reference distribution, and self-distillation helps learn a richer distribution of output sequences beyond the target itself. In §4.2, we analyze how the three terms help shape the space that search later uses.

Reconstruction. Let $s_i = (s_{i,1}, \ldots, s_{i,T_i})$ be the tokens of target i. For a code b sampled from q_i, we use teacher-forced next-token cross-entropy to reproduce that target,

$$\mathcal{L}^{(i)}_{\text{rec}} = \mathbb{E}_{b\sim q_i}\left[-\frac{1}{T_i}\sum_{t=1}^{T_i} \log P_b(s_{i,t} \mid s_{i,<t})\right], \tag{5}$$

where $s_{i,<t}$ is the prefix before position t.

Variational prior. We use an isotropic Gaussian prior $p_0 = \mathcal{N}(0, \sigma_0^2 I_r)$ to keep the posteriors at a shared scale, and penalize the KL divergence between each posterior and this prior,

$$\mathcal{L}^{(i)}_{\text{prior}} = D_{\text{KL}}(q_i \,\|\, p_0). \tag{6}$$

This discourages posteriors from moving arbitrarily far apart, thus limiting the spread of the learned space and keeping search efficient.

[1] Depending on the experiment, ϕ is either an external embedding model or a representation extracted from the frozen language model itself; the exact choice is given in §5.

Algorithm 1 BOReFT: learning and searching a low-rank generative space

Require: Frozen LM M; pairs $\{(s_i, d_i)\}_{i=1}^{n}$
encoder ϕ; objective f; budget Q
Phase I: learn the search space
for each training update **do**
  $(\mu_i, \sigma_i) \leftarrow g_\psi(\phi(s_i, d_i))$
  $b_i = \mu_i + \sigma_i \odot \epsilon,\ \epsilon \sim \mathcal{N}(0, I_r)$
  Sample rollouts from P_{b_i}
    (optionally include s_i) to form $\mathcal{T}_i$
  Update R, W, ψ on Eqs. (5) to (7)
$\mathcal{B} \leftarrow \prod_{j=1}^{r} [\min_i \mu_{ij}, \max_i \mu_{ij}]$
  from the trained posterior means
Phase II: search the learned space
for $t = 1, \ldots, Q$ **do**
  Fit surrogate on observed code–score pairs
  Optimize acquisition function
    $b_t \in \arg\max_{b \in \mathcal{B}} a_t(b)$
  Decode $S_t \sim P_{b_t}^{\tau_{\text{search}}}$
  Evaluate $y_t = f(S_t)$
return the highest-scoring sequence

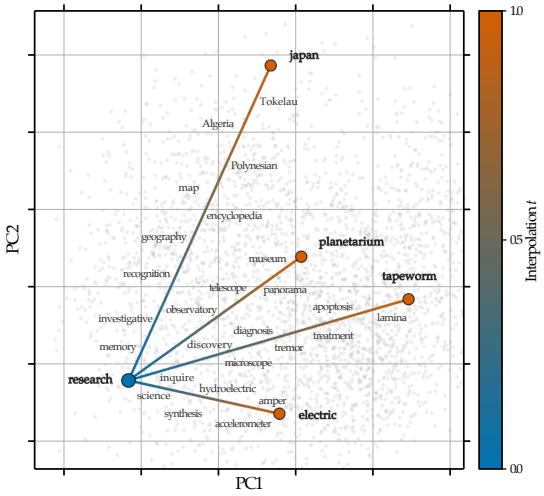


Figure 2: **Interpolation in the learned code space.** PCA of the 3072 Semantle training posterior means. Color denotes interpolation weight t.

On-policy self-distillation. Sampling from proposal distributions has been shown to improve LLM search (Brown et al., 2024; Wang et al., 2023); so the learned codes should emit a distribution of outputs instead of a single sequence. We enable this via knowledge distillation (Hinton et al., 2015) on on-policy rollouts from the posterior[2], where the student is the intervened model P_b and the teacher Q_i is a frozen copy of the unintervened model conditioned on the description d_i as privileged information (Zhao et al., 2026; Hübotter et al., 2026). The teacher then supplies dense token-level supervision via

$$\mathcal{L}_{\text{distill}}^{(i)} = \mathbb{E}_{\mathcal{T}_i} \mathbb{E}_{b \sim q_i} \left[\frac{1}{\sum_{S \in \mathcal{T}_i} |S|} \sum_{S \in \mathcal{T}_i} \sum_{t=1}^{|S|} D_{\text{KL}}(Q_i(\cdot \mid S_{<t}) \,\|\, P_b(\cdot \mid S_{<t})) \right]. \tag{7}$$

Unlike in previous work, we use forward KL here to cover all behaviors exhibited by the teacher.

Search domain. After training, we no longer sample from the target posteriors to propose search points. Instead, we construct a fixed search domain from the learned posterior means using their axis-aligned bounding box (Frazier, 2018; Siivola et al., 2021), which is the Cartesian product of r component-wise intervals,

$$\mathcal{B} = \prod_{j=1}^{r} \left[\min_i \mu_{i,j},\ \max_i \mu_{i,j} \right] \subset \mathbb{R}^r. \tag{8}$$

Training therefore produces a generative map $b \mapsto P_b^\tau$ together with a bounded continuous domain $\mathcal{B}$ over which that map can be queried, rather than just the n training means.

3.3 Searching the Learned Space with Bayesian Optimization

At inference, the domain $\mathcal{B}$ is fixed and search varies only the code $b \in \mathcal{B}$. A query chooses a code, decodes at temperature τ_{search}, and returns a score using an external black-box objective f,

$$b_t \in \mathcal{B} \longrightarrow S_t \sim P_{b_t}^{\tau_{\text{search}}} \longrightarrow y_t = f(S_t). \tag{9}$$

To choose the next code, we use Bayesian optimization over $\mathcal{B}$ (Shahriari et al., 2016; Frazier, 2018) as a principled framework for budgeted black-box search over continuous domains. After t queries, a Gaussian process surrogate (Rasmussen, 2003) is fit to the observed pairs $\{(b_j, y_j)\}_{j=1}^{t}$, and the next code is selected by maximizing an acquisition function a_t built from the surrogate,

$$b_{t+1} \in \arg\max_{b \in \mathcal{B}} a_t(b), \tag{10}$$

[2]In practice, we include target sequence s_i resulting in a mixed policy (Yan et al., 2026; Phan et al., 2025).

which defines the search strategy. In our main experiments, a_t is log expected improvement (Ament et al., 2023). After Q evaluations, we return the highest-scoring sequence observed during the run. Algorithm 1 summarizes the full training and search procedure.

4 What can search over the learned code space return?

At inference, BOReFT is the search procedure of §3 in which each query proposes a code $b \in \mathcal{B}$, the frozen decoder generates a sequence, and the task objective scores it. We ask how training shapes that space (§4.2) and when adaptive search over it is sample-efficient (§4.3). The usual i.i.d. generalization framework does not apply, because held-out discovery candidates are not drawn from the training distribution, and success is measured by finding *some* code in $\mathcal{B}$ that makes the frozen decoder emit a high-scoring sequence.

4.1 The search problem and a decomposition of its regret

Each query draws b from the box $\mathcal{B}$ of Eq. (8), then samples $S \sim P_b^\tau$ to return $f(S)$. For $\tau > 0$, this is a noisy evaluation of the expected sampled score,

$$F_\tau(b) \;:=\; \mathbb{E}_{S\sim P_b^\tau}\big[f(S)\big] \;=\; \sum_s P_b^\tau(s)\, f(s), \tag{11}$$

where the sum runs over the finitely many sequences the decoder can emit under the length cap.[3] We write $F_\tau^\star := \max_{b\in\mathcal{B}} F_\tau(b)$ for the best expected score available in the learned space, and $f^\star := \max_s f(s)$ for the best score of any sequence attained at a sequence $s^\star$.

Regret decomposition. The regret of a query at a code $b \in \mathcal{B}$, measured against the optimum, then splits into two nonnegative parts,

$$f^\star - F_\tau(b) \;=\; \underbrace{f^\star - F_\tau^\star}_{\text{representational gap}} \;+\; \underbrace{F_\tau^\star - F_\tau(b)}_{\text{search gap}}. \tag{12}$$

The model, rank, and training fix the representational gap, and further search cannot reduce it. The search gap, on the other hand, is controlled by the inference procedure. §4.2 asks when $F_\tau^\star$ is large, and §4.3 asks how efficiently can adaptive search over $\mathcal{B}$ find it.

4.2 How training shapes the space that is searched

Effect of the loss components. We first describe the effect of each loss component on the learned space. While reconstruction makes each training target searchable from its code, it does not by itself guarantee that a held-out sequence is reachable in $\mathcal{B}$ (Proposition A.1)[4]. Next, as discussed in §3.2, the VAE prior limits how far the means spread; Lemma A.2(i) shows that it bounds the resultant search domain $\mathcal{B}$ by the rank and the scale of the prior. Finally, when the expected forward KL to the teacher is small, self-distillation makes the student track the teacher on every set of sequences, after averaging over the posterior (Lemma A.2(ii)).

Semantic interpolation and coverage. A property that makes the learned space useful for search is that moving between learned anchors changes the decoded semantics gradually, so a code between anchors can express meaning that no single training target provides. Figure 2 shows this on a 2D PCA projection of the learned Semantle space: temperature-sampled decodes along linear interpolations of the training posterior means follow different semantic directions for different anchors.

Whether this property helps a specific search task depends on that task respecting the same semantic geometry. We make this link explicit below by requiring the task score to vary smoothly with distance in the semantic representation. Without such a condition, semantic interpolation alone need not imply similar task value. Let $\phi(s)$ be the semantic embedding from §3.2, and let μ_i be

[3] Under greedy decoding, the observation is the score of a deterministic sequence, which is piecewise constant in $\mathcal{B}$. Sampling at $\tau > 0$ replaces it with a noisy observation whose expectation is continuous in b.

[4] In the spirit of the No Free Lunch theorems (Wolpert, 1996; Wolpert & Macready, 1997).

the learned mean of training target s_i. For convex weights α in the simplex $\mathcal{S}_n$, we compare the semantic interpolant of the targets with the interpolant of their codes,

$$x_\alpha \;:=\; \sum_i \alpha_i\, \phi(s_i), \qquad b_\alpha \;:=\; \sum_i \alpha_i\, \mu_i. \tag{13}$$

The intervention is affine in the code (Eq. (2)), so $h'(b_\alpha) = \sum_i \alpha_i h'(\mu_i)$, an instantiation that some prior work has also used (Park et al., 2024; Subramani et al., 2022). However, the nonlinear decoder may still result in sequences that deviate from the interpolant, which must be measured empirically. Over a nonempty set of interpolation weights $\mathcal{A} \subseteq \mathcal{S}_n$, and for any sequence s, we therefore define

$$\epsilon_{\text{int}} \;:=\; \sup_{\alpha\in\mathcal{A}} \mathbb{E}_{S\sim P^\tau_{b_\alpha}} \left\|\phi(S) - x_\alpha\right\|_2, \qquad \epsilon_{\text{cov}}(s) \;:=\; \inf_{\alpha\in\mathcal{A}} \left\|\phi(s) - x_\alpha\right\|_2, \tag{14}$$

as the **interpolation error** of the learned space and the **semantic coverage error** of s, respectively. The first measures how far decoded sequences lie from the semantic interpolant of the anchors in the embedding space, and the second measures how far a given sequence lies from the semantic interpolants made available by the training set.

Proposition 4.1 (Semantic coverage and interpolation bound the representational gap)**.** *Suppose the task score f is Lipschitz with respect to the semantic embedding, in that $|f(s) - f(s')| \le L_f \left\|\phi(s) - \phi(s')\right\|_2$ for all sequences s, s' and some constant $L_f \ge 0$ set by the task, and let the interpolation weights $\mathcal{A} \subseteq \mathcal{S}_n$ be nonempty. Then, for any sequence s,*

$$F^\star_\tau \;\ge\; f(s) \;-\; L_f\big(\epsilon_{\text{cov}}(s) + \epsilon_{\text{int}}\big). \tag{15}$$

(Proof in §A.3.)

The Lipschitz condition, a property of the task relative to ϕ, links interpolation to search, by which sequences that are close under ϕ have similar task scores. Under this, taking $s = s^\star$ bounds the representational gap of Eq. (12) by $L_f(\epsilon_{\text{cov}}(s^\star) + \epsilon_{\text{int}})$. The Lipschitz condition holds for Semantle, whose score is cosine similarity between normalized embeddings, so $L_f = 1$. For the molecular predictors it remains an assumption. In §5 we measure coverage of held-out targets and the displacement of the mean decoded semantics along interpolated codes, which is at most ϵ_{int} (§A.3).

Takeaway

Reconstruction anchors known behaviors at codes, the prior keeps them inside a compact search region, and self-distillation distributes the teacher outputs on codes around the anchors. If nearby embeddings have similar task scores, the representational gap is bounded by how well the optimal sequence is covered by the train set and how close the decoder stays to semantic interpolants (Proposition 4.1).

4.3 Guarantees for adaptive search

After training, the remaining question is how efficiently an adaptive procedure closes the *search gap* of Eq. (12). In each round, a new query is selected based on the task scores observed so far, returning a noisy evaluation of the expected sampled score F_τ of Eq. (11). Here we assume the task score is bounded in an interval $[f_{\min}, f_{\max}]$ and write $\Delta_f := f_{\max} - f_{\min}$ for its range.

Proposition 4.2 (Queries are standard stochastic bandit observations)**.** *Let each query $b_t \in \mathcal{B}$ be proposed by any rule that depends on the past queries and observations $\mathcal{H}_{t-1}$, for example by maximizing an acquisition function fit to them, and let $Y_t := f(S_t)$ with $S_t \sim P^\tau_{b_t}$. Then $\mathbb{E}[Y_t \mid \mathcal{H}_{t-1}, b_t] = F_\tau(b_t)$, and the observation error $\xi_t := Y_t - F_\tau(b_t)$, the deviation of the sampled score from its conditional expectation, is conditionally $(\Delta_f/2)$-sub-Gaussian; i.e., its tails decay at least as fast as those of a Gaussian with standard deviation $\Delta_f/2$. (Proof in §A.4.)*

In words, each query is an unbiased, conditionally sub-Gaussian observation of F_τ, for any policy that proposes the code. Standard bandit and Bayesian optimization guarantees apply to this observation model only under their own further assumptions. Discovery depends on the best sequence a run emits, which the next corollary controls for any policy.

Corollary 4.3 (Cumulative regret to discovery guarantee)**.** *Let $R_Q := \sum_{t\le Q} \big(F^\star_\tau - F_\tau(b_t)\big)$ be the cumulative regret of any policy against the best expected score $F^\star_\tau$ over $\mathcal{B}$, and let $S_1, \ldots, S_Q$ be the sequences its queries return. Then, with probability at least $1-\delta$,*

$$\max_{t\le Q} f(S_t) \;\ge\; F^\star_\tau \;-\; \frac{R_Q}{Q} \;-\; \Delta_f \sqrt{\frac{\log(1/\delta)}{2Q}}. \tag{16}$$

(Proof in §A.4.)

In words, if a cumulative-regret bound holds, it translates into a guarantee on the best sequence the run emits (§A.4), and subtracting the coverage and interpolation errors additionally requires the Lipschitz condition of Proposition 4.1 to hold.

Takeaway

Each query is an unbiased, conditionally sub-Gaussian observation of F_τ (Proposition 4.2), so standard bandit and Bayesian optimization guarantees apply when their remaining regularity assumptions hold, and those guarantees control the best sequence the run emits (Corollary 4.3).

5 EXPERIMENTS

We evaluate BOReFT on two tasks, Semantle word search and molecular property optimization, and organize the experiments around the following questions. First, whether search over the learned code space finds high-scoring sequences unseen during representation training. Second, how that comparison holds when other methods share the same representation-training data. Third, what components of BOReFT's training contribute to its performance, and last, whether the coverage and interpolation errors are consistent with the theory. We also provide additional analyses and details in §B, including the effect on search quality of alternative search domain construction, decoding temperature, kernels, acquisition functions, and intervention rank.

5.1 TASKS AND EVALUATION PROTOCOL

Semantle word search. In Semantle (Agarwal et al., 2025a), the generator proposes a single word and receives the cosine similarity of that word to a hidden target under a frozen embedding model, Qwen3-Embedding-0.6B (Zhang et al., 2025), as the black-box function f. BOReFT uses Llama-3.2-1B-Instruct (Meta AI, 2024) both as the generator and as the source of ϕ, sets rank $r = 64$, and uses for representation training 3072 words ($\sim$2 tokens/word) paired with synthetically-generated definitions from Claude Opus 5 (Anthropic, 2026). We score five hidden targets each from the train and held-out sets, each with three different warmstart sets of ten words each, and a budget of 500 evaluations. The complete setup and protocol are in §B.1 and §B.3.

Molecular property optimization. The second task asks for a new molecule as a SMILES string (Weininger, 1988) with high predicted activity on a protein target, a standard goal in drug discovery. We use DRD2, GSK3β, and JNK3 predictors (Huang et al., 2021; Gao et al., 2022) as black-box objective functions, where each score is a class-1 probability in $[0, 1]$, and an invalid SMILES string receives 0. BOReFT uses MiST (Bran et al., 2025), a chemistry-pretrained Qwen2.5-3B model, as the generator and Qwen3-Embedding-0.6B as ϕ, sets rank $r = 64$, and trains once on 1024 description–molecule pairs from ChEBI-20 (Edwards et al., 2021) from molecules within the 90th percentile of activity for any target (i.e., DRD2 ≤ 0.029, GSK3$\beta \leq 0.10$, JNK3 ≤ 0.04; $\sim$31 tokens/molecule). The budget is 500 evaluations, with five warmstart sets of ten molecules.

Baselines and SFT evaluation. We compare with LLM-search baselines OPRO (Yang et al., 2024), BOPRO (Agarwal et al., 2025a), MiGrATe (Phan et al., 2025), AutoDiscovery (Agarwal et al., 2025b), SDPO-TTT (Hübotter et al., 2026), and random repeated sampling. These cover prevalent search strategies, including in-context learning, Bayesian optimization, tree-search, and test-time training. We also include as a reference discrete BO (Ranković & Schwaller, 2025) over frozen embeddings, which can only propose sequences from the training set. Lastly, to strictly compare the quality of search, we also evaluate the baselines after supervised fine-tuning (SFT) on the same train set that BOReFT uses, which puts the methods on a shared data prior. We do this by training a LoRA adapter (Hu et al., 2022) for each base model for ten epochs. The search mechanism for each baseline is in §B.4.

5.2 RESULTS AND ANALYSES

Semantle. Figure 3 follows the best similarity found at each verification step on the held-out words, and Table 1 gives the final exact-match (EM) count, mean of the best-found similarity, and

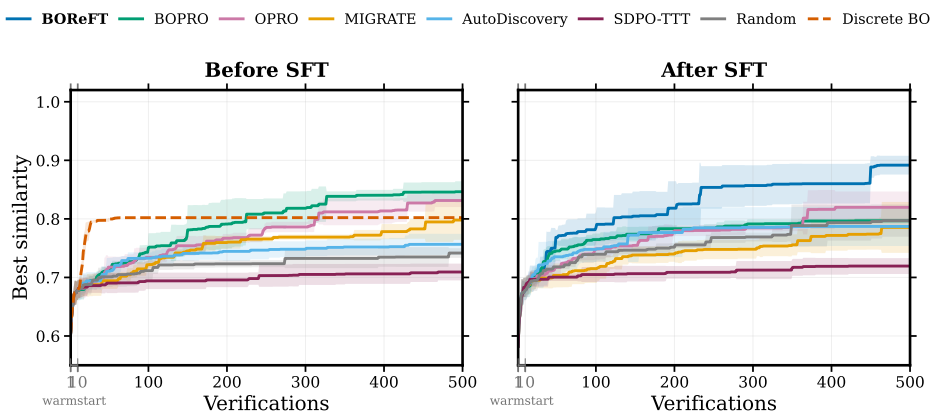


Figure 3: **Best-so-far performance on held-out Semantle search.** Best semantic similarity found at each verification step, averaged over 5 targets and 3 warmstart runs, with a band of ±1 SD. We plot BOReFT only in the after-SFT block to group together methods that all use the training data.

	Before SFT			After SFT		
Method	EM (↑)	Sim. (↑)	Novelty (↑)	EM (↑)	Sim. (↑)	Novelty (↑)
Discrete BO	*0/15*	*0.802*	*0.000*	*0/15*	*0.802*	*0.000*
Random	0/15	0.710	0.561	1/15	0.797	0.119
SDPO-TTT	0/15	0.709	0.679	0/15	0.719	<u>0.676</u>
AutoDiscovery	0/15	0.757	<u>0.736</u>	1/15	0.787	0.154
MiGrATe	1/15	0.798	0.630	1/15	0.785	0.621
BOPRO	<u>3/15</u>	<u>0.846</u>	**0.798**	0/15	0.797	0.203
OPRO	<u>3/15</u>	0.831	0.708	<u>2/15</u>	<u>0.820</u>	0.212
BOReFT	**8/15**	**0.892**	0.691	**8/15**	**0.892**	**0.691**

Table 1: **Held-out Semantle search.** Exact-match counts and mean best similarity over 15 runs (5 held-out targets and 3 warmstarts) with the rate of novelty w.r.t. the train set. Bold and underline mark the best and second-best. BOReFT scores are repeated under before-SFT for clarity.

novelty rate (w.r.t. training) across 5 seeds. BOReFT recovers $8/15$ held-out targets, at mean similarity 0.892, with novelty rate 0.691, while the next-best EM count is only $3/15$ before SFT alignment and $2/15$ after it. Further, discrete BO finds 0 held-out words by construction, but it finds every training word in the fewest number of evaluations (Table 4). Fine-tuning the baselines on the training words raises the held-out EM and similarity scores of AutoDiscovery and random sampling, while still trailing BOReFT, but also lowers scores on both metrics for BOPRO and OPRO. We discuss reasons for baseline performance further in §B.5.

Molecular property optimization. Table 2 reports for each property the mean and best activation probability found over 5 seeds for BOReFT and baselines, before and after SFT. Despite the percentile-90 training, BOReFT generalizes well beyond the scores it is trained on, and, notably, supports multiple task objectives simultaneously. Before SFT, BOReFT leads JNK3 with mean 0.370 and best score 0.940, the highest by a margin ≥ 0.5, and also finds the highest GSK3β score of 0.670. On DRD2, however, it trails OPRO, MiGrATe, and BOPRO. After SFT, all baselines except MiGrATe show a drop in performance, promoting BOReFT to the top-2 on all properties.

Takeaway

On both Semantle and molecule property optimization, searching a learned manifold with Bayesian optimization is able to generalize beyond the task performance of its training data, and can outperform other LLM search strategies (ICL, TTT, tree-search). BOReFT's performance, however, remains coupled to the quality of training data (the representation gap of §4.1), which may prevent it from accessing high-scoring distributions, such as on DRD2, and motivates the need for continual learning to update the learned manifold as new observations are collected (De Santi et al., 2026), which we leave as future work.

Method	Before SFT			After SFT		
	DRD2 (↑)	GSK3β (↑)	JNK3 (↑)	DRD2 (↑)	GSK3β (↑)	JNK3 (↑)
Random	0.346/0.717	0.310/0.590	0.106/0.110	0.235/0.403	0.220/0.260	0.094/0.120
SDPO-TTT†	–/0.981	–/0.280	–/0.100	–/0.656	–/0.270	–/0.190
AutoDiscovery	0.534/0.929	0.316/0.380	0.158/0.160	0.345/0.801	0.226/0.290	0.094/0.160
MiGrATe	0.880/**1.000**	0.452/0.580	0.216/0.260	**0.844**/**0.958**	**0.526**/**0.710**	0.218/0.260
BOPRO	0.253/**1.000**	**0.470**/0.630	0.284/0.420	0.086/0.152	0.322/0.520	0.100/0.130
OPRO	**0.965**/**1.000**	0.402/0.430	0.356/0.410	0.145/0.425	0.275/0.360	0.180/0.320
BOReFT	0.645/0.782	0.446/**0.670**	**0.370**/**0.940**	0.645/0.782	0.446/0.670	**0.370**/**0.940**

Table 2: **Molecular property optimization.** Mean and best (`mean/max`) activation probability over 5 seeds at a budget of 500 evaluations. Bold and underline mark the best and second-best. BOReFT scores are repeated under before-SFT for clarity. †*Only one seed finished for SDPO-TTT.*

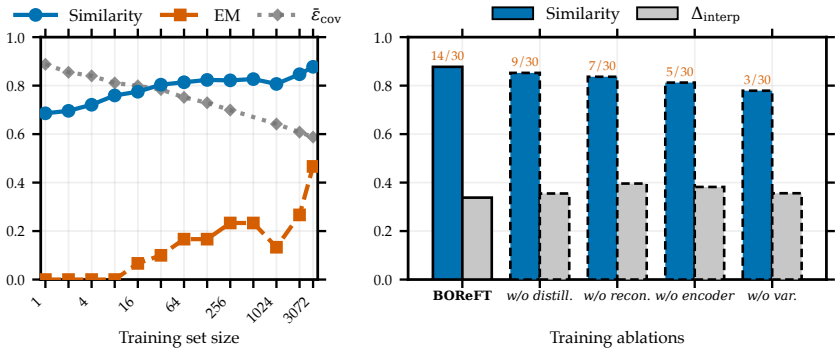


Figure 4: **Training-set size and method ablations. Left:** Mean best similarity and EM on Semantle, along with median coverage error over 30 runs across train and test. **Right:** Mean best similarity and median interpolation distance Δ_{interp} on the same 30 runs, with EM counts in orange.

What components of BOReFT help search? Full BOReFT finds $14/30$ hidden targets across train and test. Removing self-distillation, reconstruction, or variational training lowers that count to $9/30$, $7/30$, and $3/30$, respectively, and removing the shared encoder lowers it to $5/30$ (Fig. 4, right). This underscores the importance of each loss component, as well as the geometric bias from the encoder, in learning a suitable space for search (see Table 10 and Fig. 7).

Do coverage and interpolation move with search? Proposition 4.1 bounds the representational gap by $L_f(\epsilon_{\text{cov}} + \epsilon_{\text{int}})$. Figure 4(left) therefore plots coverage error[5] beside the search curves. On 928 held-out words, the median coverage error falls from 0.89 at $N = 1$ to 0.59 at $N = 3072$, while search improves correspondingly. The interpolation error is the expected distance of one decoded sequence from the semantic interpolant x_α. Figure 4(right) reports a related but smaller quantity Δ_{interp}, the distance from the average decoded embedding to x_α[6]. We measure this halfway between pairs of training words, where the interpolant is farthest from either word. On 40 such pairs, Δ_{interp} is 0.338, while removing self-distillation, variational training, reconstruction, or the shared encoder all raise it to between 0.355 and 0.396 and reduce search quality (Table 10).

6 Conclusion

We introduce BOReFT, a generative search method for language models that first learns a low-dimensional continuous space of representation interventions, then uses Bayesian optimization to search that space a black-box objective. Our analysis characterizes when the space contains high-quality solutions and when adaptive search over it is well behaved, while experiments on Semantle and molecular property optimization show gains over baselines. Future work may consider how to expand the learned space with new observations.

[5]We report an upper-bound on ϵ_{cov} (Eq. (14)) using the distance of a sequence to its nearest train embedding.
[6]By Jensen's inequality, it is at most ϵ_{int} (§A.3).

AI USE STATEMENT

In this work, we used generative AI tools to generate the Semantle word definitions and the ChEBI-20 semantic categories, to assist with the theory proofs, and to help write and edit code. All generated definitions, categories, proofs, and code were manually reviewed and edited before being incorporated. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Assumptions and complete proofs of the theoretical claims are given in §4 and §A. The method is specified in §3. Models, representation-training data, and evaluation protocols are described in §B. Anonymized source code is included in the supplementary material.

REFERENCES

Dhruv Agarwal, Manoj Ghuhan Arivazhagan, Rajarshi Das, Sandesh Swamy, Sopan Khosla, and Rashmi Gangadharaiah. Searching for optimal solutions with llms via bayesian optimization. In *International Conference on Learning Representations*, 2025a.

Dhruv Agarwal, Bodhisattwa Prasad Majumder, Reece Adamson, Megha Chakravorty, Satvika Reddy Gavireddy, Aditya Parashar, Harshit Surana, Bhavana Dalvi Mishra, Andrew McCallum, Ashish Sabharwal, and Peter Clark. Autodiscovery: Open-ended scientific discovery via bayesian surprise. In *Advances in Neural Information Processing Systems*, 2025b.

Ekin Akyürek, Mehul Damani, Adam Zweiger, Linlu Qiu, Han Guo, Jyothish Pari, Yoon Kim, and Jacob Andreas. The surprising effectiveness of test-time training for few-shot learning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 942–963. PMLR, 2025.

Sebastian Ament, Samuel Daulton, David Eriksson, Maximilian Balandat, and Eytan Bakshy. Unexpected improvements to expected improvement for Bayesian optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Anthropic. System card: Claude Opus 5. `https://www.anthropic.com/claude-opus-5-system-card`, 2026. July 24.

Andres M Bran, Tong Xie, Shai Pranesh, Jeffrey Meng, Xuan Vu Nguyen, Jeremy Goumaz, David Ming Segura, Ruizhi Xu, Dongzhan Zhou, Wenjie Zhang, et al. Mist: Understanding the role of mid-stage scientific training in developing chemical reasoning models. *arXiv preprint arXiv:2512.21231*, 2025.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*, 2024.

Lichang Chen, Jiuhai Chen, Tom Goldstein, Heng Huang, and Tianyi Zhou. InstructZero: Efficient instruction optimization for black-box large language models. In *International Conference on Machine Learning (ICML)*, 2024.

Riccardo De Santi, Bruce Lee, Cristian Perez Jensen, Kimon Protopapas, Sophia Tang, Cheng-Hao Liu, Pranam Chatterjee, Yisong Yue, and Andreas Krause. Active flow expansion for out-of-distribution discovery: from theory to molecules. *arXiv preprint arXiv:2606.08802*, 2026.

Chunyuan Deng, Ruidi Chang, and Hanjie Chen. Learning distribution-wise control in representation space for language models. In *International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 13044–13068, 2025.

Carl Edwards, ChengXiang Zhai, and Heng Ji. Text2Mol: Cross-modal molecule retrieval with natural language queries. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 595–607. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-main.47.

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

Wenhao Gao, Tianfan Fu, Jimeng Sun, and Connor W. Coley. Sample efficiency matters: A benchmark for practical molecular optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Atticus Geiger, Zhengxuan Wu, Christopher Potts, Thomas Icard, and Noah D. Goodman. Finding alignments between interpretable causal variables and distributed neural representations. In *Proceedings of the Third Conference on Causal Learning and Reasoning (CLeaR)*, 2024.

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

Wes Gurnee, Nicholas Sofroniew, Adam Pearce, Mateusz Piotrowski, Isaac Kauvar, Runjin Chen, Anna Soligo, Paul Bogdan, Euan Ong, Rowan Wang, Ben Thompson, David Abrahams, Subhash Kantamneni, Emmanuel Ameisen, Joshua Batson, and Jack Lindsey. Verbalizable representations form a global workspace in language models. *arXiv preprint arXiv:2607.15495*, 2026.

Moritz Hardt and Yu Sun. Test-time training on nearest neighbors for large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

Kexin Huang, Tianfan Fu, Wenhao Gao, Yue Zhao, Yusuf Roohani, Jure Leskovec, Connor W. Coley, Cao Xiao, Jimeng Sun, and Marinka Zitnik. Therapeutics data commons: Machine learning datasets and tasks for drug discovery and development. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021.

Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *International Conference on Learning Representations (ICLR)*, 2024.

Jonas Hübotter, Frederike Lübeck, Lejs Deen Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, and Andreas Krause. Reinforcement learning via self-distillation. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=QkfkxyRizZ.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR)*, 2014.

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

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Rätsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *International Conference on Machine Learning (ICML)*, 2019.

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

Meta AI. Llama 3.2 model card. Hugging Face model card, 2024. URL https://huggingface.co/meta-llama/Llama-3.2-1B-Instruct.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 39643–39666. PMLR, 2024.

Peter Phan, Dhruv Agarwal, Kavitha Srinivas, Horst Samulowitz, Pavan Kapanipathi, and Andrew McCallum. Migrate: Mixed-policy grpo for adaptation at test-time. *arXiv preprint arXiv:2508.08641*, 2025.

Yury Polyanskiy and Yihong Wu. *Information Theory: From Coding to Learning*. Cambridge University Press, 2025.

Bojana Ranković and Philippe Schwaller. GOLLuM: Gaussian process optimized LLMs – reframing LLM finetuning through Bayesian optimization. In *ICLR Workshop on World Models*, 2025.

Bojana Ranković and Philippe Schwaller. Large language models as generative Bayesian policies. In *ICML Workshop on AI for Science*, 2026. OpenReview: OSlXZGcur9.

Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer school on machine learning*, pp. 63–71. Springer, 2003.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15504–15522. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.828.

Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams, and Nando de Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1): 148–175, 2016. doi: 10.1109/JPROC.2015.2494218.

Eero Siivola, Andrei Paleyes, Javier González, and Aki Vehtari. Good practices for bayesian optimization of high dimensional structured spaces. *Applied AI Letters*, 2(2):e24, 2021.

Nishant Subramani, Nivedita Suresh, and Matthew E. Peters. Extracting latent steering vectors from pretrained language models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 566–581. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.findings-acl.48.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei A. Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9229–9248. PMLR, 2020.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L. Turner, Callum McDougall, Monte MacDiarmid, Alex Tamkin, Esin Durmus, Tristan Hume, Francesco Mosconi, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. Scaling monosemanticity: Extracting interpretable features from Claude 3 Sonnet. *Transformer Circuits Thread*, 2024.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

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

Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P. Xing. Deep kernel learning. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pp. 370–378. PMLR, 2016.

David H. Wolpert. The lack of a priori distinctions between learning algorithms. *Neural Computation*, 8(7):1341–1390, 1996.

David H. Wolpert and William G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997.

Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Dan Jurafsky, Christopher D. Manning, and Christopher Potts. ReFT: Representation finetuning for language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Daniel Wurgaft, Can Rager, Matthew Kowal, Vasudev Shyam, Sheridan Feucht, Usha Bhalla, Tal Haklay, Eric Bigelow, Raphael Sarfati, Thomas McGrath, Owen Lewis, Jack Merullo, Noah D. Goodman, Thomas Fel, Atticus Geiger, and Ekdeep Singh Lubana. Manifold steering reveals the shared geometry of neural network representation and behavior. *arXiv preprint arXiv:2605.05115*, 2026.

Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. Learning to reason under off-policy guidance. *Advances in Neural Information Processing Systems*, 38:117157–117186, 2026.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. Large language models as optimizers. In *International Conference on Learning Representations*, 2024.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=Jpxfof0EaS.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. Representation engineering: A top-down approach to AI transparency. *arXiv preprint arXiv:2310.01405*, 2023.

A Proofs and experimental definitions

This appendix proves the results stated in §4, gives the formal versions of the statements that §4.2 summarizes in words, and defines the diagnostics reported in §5.

Conventions. The search domain $\mathcal{B} \subset \mathbb{R}^r$ is compact with nonempty interior. Each logit map $b \mapsto L_{p,u}(b)$ is continuous in b, because every frozen operation after the intervention site is continuous. Sampling at temperature $\tau > 0$ draws $s_t \sim p_b^\tau(\cdot \mid s_{<t}) \propto e^{L_{s_{<t},\cdot}(b)/\tau}$ and stops at the first `[EOS]` or at a length cap $T_{\max}$. At $\tau = 0$, sampling reduces to greedy decoding, so $P_b^0(s)$ equals one when greedy decoding at b emits s and zero otherwise. Every target sequence is `[EOS]`-terminated: $s = (s_1, \ldots, s_T)$ with $s_T =$ `[EOS]`, $s_t \neq$ `[EOS]` for $t < T$, and $T \leq T_{\max}$. With this convention the event that a run emits exactly s coincides with the trajectory event, so $P_b^\tau(s) = \prod_{t=1}^{T} p_b^\tau(s_t \mid s_{<t})$ with the terminal `[EOS]` factor included, and sequences truncated at the length cap are covered by treating $t = T_{\max}$ as a terminal step. Two consequences used below are that no target is a prefix of another, and that distinct terminated sequences correspond to disjoint emission events. Wherever a greedy argmax appears we break exact ties by a fixed total order on $\mathcal{V}$, so that the greedy decode map Φ_0 is defined everywhere. We call a code b *nondegenerate for* s if no logit difference $L_{s_{<t},s_t}(b) - L_{s_{<t},v}(b)$ with $v \neq s_t$ vanishes along the trajectory of s at b, and statements about greedy decoding are made at such codes.

Reliable reachability. Some statements below concern what the learned space can emit with confidence rather than on average, so we record that notion here. At $\tau > 0$ every sequence has positive probability at every code, so the support of P_b^τ does not distinguish what the space can express. We instead fix a confidence level $\theta \in (0, 1)$ and define the *reachable set*

$$\mathcal{R}_\tau(\theta) \;:=\; \big\{s : \sup_{b \in \mathcal{B}} P_b^\tau(s) \geq \theta\big\}, \tag{17}$$

the sequences that some code in the domain emits with probability at least θ in a single sample. When that set is nonempty we write $f^\star_{\mathrm{reach}}(\tau, \theta) := \max\{f(s) : s \in \mathcal{R}_\tau(\theta)\}$ for the best score among its members. The two views are related. A code that emits a sequence s with probability at least θ has $F_\tau(b) \geq \theta f(s) + (1 - \theta) f_{\min}$, so taking the best such sequence gives

$$F^\star_\tau \;\geq\; \theta\, f^\star_{\mathrm{reach}}(\tau, \theta) \;+\; (1 - \theta)\, f_{\min}. \tag{18}$$

The main text works with F_τ throughout, because that is the quantity a query observes.

A.1 Reconstruction alone gives no held-out coverage (Proposition A.1)

§4.2 states that fidelity to the training targets carries no information about held-out sequences, and this subsection makes that statement precise. Coverage is measured through the reachable set of Eq. (17), because at positive temperature every held-out sequence has positive probability at every code and coverage in terms of support alone would be trivial. For a finite set U of held-out targets we define the *oracle coverage* $\chi_{\mathrm{oracle}}(U; \tau, \theta) := |\mathcal{R}_\tau(\theta) \cap U|/|U|$, which upper-bounds the fraction of U that any search over $\mathcal{B}$ could reliably elicit at confidence θ.

Proposition A.1 (No coverage guarantee from reconstruction alone)**.** *Fix rank r, a compact $\mathcal{B}$ with nonempty interior, distinct codes $\mu_1, \ldots, \mu_n \in \mathcal{B}$, targets $s_1, \ldots, s_n$, and a finite held-out set U of sequences disjoint from the targets. For any $\tau \geq 0$ and $\theta \in (1/2, 1)$, there exists no $g(n, r) > 0$ that lower-bounds $\chi_{\mathrm{oracle}}(U; \tau, \theta)$ over the class of decoders with smooth logit maps that satisfy $P^\tau_{\mu_i}(s_i) \geq \theta$ for all i.*

The number of training targets and the rank therefore give no guarantee of held-out coverage. The construction uses the prefix tree of the training targets, so it applies to sequences that share prefixes and does not assume that the first token determines the output.

Proof of Proposition A.1. Let E be the set of prefix–token edges along the training sequences, and write Z for the first edge at which each held-out sequence leaves that tree. Such an edge exists: a path that stayed on the tree through the terminal token would equal a training target. At that node the tree has some other continuation.

Fix a scale $a > 0$, small distinct tie-breaking offsets $c_v \in [0, a/4]$, and smooth bumps $\varphi_i(b) := \exp(-\|b - \mu_i\|_2^2/\sigma^2)$. Define

$$L_{p,v}(b) \;:=\; c_v \;-\; a\,\mathbf{1}\{(p, v) \in Z\} \;+\; 2a \sum_{i\,:\,(p,v) \in E_i} \varphi_i(b), \tag{19}$$

where E_i is the set of edges along training target s_i. These maps are smooth in b.

At μ_i, the correct token of s_i receives a bump of $2a$ at every step. Once σ is small enough that every other bump is at most $1/(8n)$ there, each competitor is at most $a/2$, so the correct token leads by at least $3a/2$. A softmax step with that lead has conditional probability at least $1 - (K - 1)e^{-(3a/2)/\tau}$ when $\tau > 0$, and a product over at most $T_{\max}$ steps is at least θ once $a \geq \tau \log \frac{T_{\max}(K-1)}{1-\theta}$. At $\tau = 0$ the same lead makes greedy decoding emit s_i. Thus every training target is reconstructed at confidence θ.

A held-out leaving token is penalized by a at every code and receives no bump, while the tree's alternative continuation is not penalized. The leaving token therefore trails by at least $3a/4$ everywhere, so its conditional probability is below $1/2$. The probability of the whole held-out sequence is at most that factor, hence below θ at every code. At $\tau = 0$, greedy decoding takes the alternative continuation and leaves the held-out trajectory. No positive function of n and r can therefore lower-bound χ_{oracle} over this class. □

The decoder in the proof is a smooth map from codes to logits, a class larger than a transformer with a rank-r intervention. The proposition says that smoothness together with high-confidence reconstruction leaves held-out coverage undetermined. The penalty is applied at the first prefix where a held-out sequence leaves the training tree, so the same construction covers targets that share a long prefix with a training sequence.

A.2 The prior and distillation terms (Lemma A.2)

This subsection states and proves the two consequences summarized in §4.2. Throughout, $q_i = \mathcal{N}(\mu_i, \Sigma_i)$ is the Gaussian variational posterior for target s_i, with diagonal covariance in our implementation, $p_0 = \mathcal{N}(0, \sigma_0^2 I_r)$ is the shared prior, and $\mathcal{B}$ is the axis-aligned bounding box of the means $\{\mu_1, \dots, \mu_n\}$. As in §3, P_b denotes the decoder distribution P_b^τ at the training temperature $\tau = 1$: teacher-forced reconstruction evaluates exactly $-\log P_b(s_i)$ under the termination convention, and the distillation term scores the student at the same temperature (the implementation exposes a separate softmax temperature for distillation, which we take at its default value of one). We use two standard facts (Polyanskiy & Wu, 2025): Pinsker's inequality $\mathrm{TV}(P, Q) \le \sqrt{D_{\mathrm{KL}}(P\|Q)/2}$ in natural logarithms, and joint convexity of both D_{KL} and TV in their arguments.

The first consequence follows from the closed form of the Gaussian divergence, and the second from Pinsker's inequality together with convexity of relative entropy in its second argument.

Lemma A.2 (Consequences of the prior and distillation terms)**.** *Write $\bar{P}_i(\cdot) := \mathbb{E}_{b\sim q_i}[P_b(\cdot)]$ for the student sequence distribution of target i averaged over its posterior. Then:*

- *(i) if $D_{\mathrm{KL}}(q_i\|p_0) \le \kappa_i$ for every target i, then each mean satisfies $\|\mu_i\|_2 \le \sigma_0\sqrt{2\kappa_i}$, and with $\kappa_{\max} := \max_i \kappa_i$ the search domain satisfies $\mathcal{B} \subseteq [-R, R]^r$ for $R := \sigma_0\sqrt{2\kappa_{\max}}$ and $\mathrm{diam}(\mathcal{B}) \le 2\sigma_0\sqrt{2r\kappa_{\max}}$;*
- *(ii) if $\mathbb{E}_{b\sim q_i}[D_{\mathrm{KL}}(Q_i\|P_b)] \le \eta_i$ for a target i, then $\mathrm{TV}(Q_i, \bar{P}_i) \le \sqrt{\eta_i/2}$. Hence $\bar{P}_i(A) \ge Q_i(A) - \sqrt{\eta_i/2}$ for every set A of terminated sequences, for instance the sequences that express a given meaning, and $\frac{1}{2}\sum_{k\le m} |Q_i(A_k) - \bar{P}_i(A_k)| \le \sqrt{\eta_i/2}$ for any disjoint sets $A_1, \dots, A_m$.*

In words, part (i) says that the prior term controls the geometric extent of the domain that is later searched, which is the statement summarized in §4.2. Part (ii) says that the distillation term makes the student's output distribution, averaged over the posterior, track the teacher's on every set of sequences simultaneously. A teacher that spreads its mass over several disjoint sets of sequences therefore forces that average to assign mass to each of them. The quantity η_i is an expected sequence-level forward KL. The implemented distillation loss averages per-token divergences on a mixture of teacher samples, student rollouts, and the target, so the logged loss is not η_i.

Proof of Lemma A.2. (i). For the Gaussians $q_i = \mathcal{N}(\mu_i, \Sigma_i)$ and $p_0 = \mathcal{N}(0, \sigma_0^2 I_r)$ the divergence has the closed form

$$D_{\mathrm{KL}}(q_i \,\|\, p_0) \;=\; \tfrac{1}{2}\Big(\|\mu_i\|_2^2/\sigma_0^2 \;+\; \mathrm{tr}(\Sigma_i)/\sigma_0^2 \;-\; r \;-\; \log\det(\Sigma_i/\sigma_0^2)\Big), \tag{20}$$

whose covariance contribution equals $\frac{1}{2}\sum_{j\le r}(\lambda_j - \log\lambda_j - 1)$ in terms of the eigenvalues λ_j of Σ_i/σ_0^2 and is therefore nonnegative, since $x - \log x - 1 \ge 0$ for $x > 0$. Hence $\|\mu_i\|_2^2 \le 2\sigma_0^2\kappa_i$ for every target. Every coordinate of every mean then satisfies $|\mu_{ij}| \le \|\mu_i\|_2 \le \sigma_0\sqrt{2\kappa_{\max}} = R$, so the bounding box of the means lies in $[-R, R]^r$, whose Euclidean diameter is $2R\sqrt{r}$, and the box's diameter is at most that. The same covariance term is nonnegative and diverges as any eigenvalue of Σ_i/σ_0^2 tends to 0 or to ∞, so a small divergence also keeps the posterior spread comparable to the prior's.

(ii). The averaged distribution $\bar{P}_i$ is a mixture of the P_b over the mixing measure q_i, and the teacher Q_i does not depend on b, so convexity of relative entropy in its second argument gives

$$D_{\mathrm{KL}}(Q_i \,\|\, \bar{P}_i) \;\le\; \mathbb{E}_{b\sim q_i}\big[D_{\mathrm{KL}}(Q_i \,\|\, P_b)\big] \;\le\; \eta_i. \tag{21}$$

Pinsker's inequality then gives $\mathrm{TV}(Q_i, \bar{P}_i) \le \sqrt{\eta_i/2}$. The bound $\bar{P}_i(A) \ge Q_i(A) - \sqrt{\eta_i/2}$ is the definition of total variation applied to the event A. For disjoint sets $A_1, \dots, A_m$, the map sending a

sequence to the index of the set containing it (with one extra index for sequences in none of them) is a deterministic function of the sample, so the induced distributions on indices are pushforwards of Q_i and $\bar{P}_i$. Total variation cannot increase under a pushforward, and for distributions on a finite set it equals half the ℓ_1 distance, which gives $\frac{1}{2}\sum_{k\le m}|Q_i(A_k)-\bar{P}_i(A_k)| \le \mathrm{TV}(Q_i,\bar{P}_i) \le \sqrt{\eta_i/2}$. □

A.3 Proof of Proposition 4.1

Proof of Proposition 4.1. Fix a sequence s and a weight $\alpha \in \mathcal{A}$, and abbreviate $\mathbb{E}_S$ for the expectation over $S \sim P^{\tau}_{b_\alpha}$. Since $F_\tau(b_\alpha) = \mathbb{E}_S[f(S)]$, the Lipschitz hypothesis applied inside the expectation gives

$$f(s) - F_\tau(b_\alpha) \;=\; \mathbb{E}_S\big[f(s) - f(S)\big] \;\le\; L_f\, \mathbb{E}_S\, \|\phi(s) - \phi(S)\|_2 \,. \tag{22}$$

For every sequence S the triangle inequality in the embedding space gives $\|\phi(s) - \phi(S)\|_2 \le \|\phi(s) - x_\alpha\|_2 + \|\phi(S) - x_\alpha\|_2$, and taking expectations gives

$$f(s) - F_\tau(b_\alpha) \;\le\; L_f \Big(\|\phi(s) - x_\alpha\|_2 + \mathbb{E}_S\, \|\phi(S) - x_\alpha\|_2 \Big). \tag{23}$$

Each b_α is a convex combination of the means μ_i, and the axis-aligned bounding box $\mathcal{B}$ of those means is convex and contains them, so $b_\alpha \in \mathcal{B}$ and $F^\star_\tau \ge F_\tau(b_\alpha)$. The second term of Eq. (23) is at most $L_f \epsilon_{\mathrm{int}}$, because ϵ_{int} is a supremum over $\mathcal{A}$, so

$$F^\star_\tau \;\ge\; f(s) - L_f\, \|\phi(s) - x_\alpha\|_2 - L_f \epsilon_{\mathrm{int}} \tag{24}$$

for every $\alpha \in \mathcal{A}$. The left-hand side does not depend on α, so taking the supremum of the right-hand side over $\mathcal{A}$ replaces $\|\phi(s) - x_\alpha\|_2$ by its infimum, which is $\epsilon_{\mathrm{cov}}(s)$. □

The bound is loose by construction, since it passes through a worst-case Lipschitz constant and discards the direction in which the decoded semantics deviate from the interpolant. It is a decomposition of the representational gap into two measurable parts, not a tight estimate.

Remark (scores that are linear in the embedding). When the score reads the embedding linearly, the interpolation term can be replaced by a smaller quantity. Semantle is of this form: its embeddings are normalized, the hidden target $s^\star$ has embedding $x^\star := \phi(s^\star)$, and $f(s) = \langle x^\star, \phi(s)\rangle$, so $L_f = 1$ and $f^\star = 1$ by the Cauchy–Schwarz inequality. Writing

$$m_\tau(b) \;:=\; \mathbb{E}_{S\sim P^\tau_b}\big[\phi(S)\big] \tag{25}$$

for the mean semantics of what a code emits, we have $F_\tau(b) = \langle x^\star, m_\tau(b)\rangle$. Adding and subtracting $\langle x^\star, x_\alpha\rangle$, and applying the Cauchy–Schwarz inequality to the second term, gives

$$1 - F^\star_\tau \;\le\; \Big(1 - \sup_{\alpha\in\mathcal{A}} \langle x^\star, x_\alpha\rangle\Big) \;+\; \sup_{\alpha\in\mathcal{A}} \|m_\tau(b_\alpha) - x_\alpha\|_2 \,. \tag{26}$$

The interpolation term of Eq. (26) is the displacement of the mean decoded semantics rather than the expected spread around the interpolant, and by Jensen's inequality it is the smaller of the two. It is the quantity that the interpolation paths of §5 report. The coverage term of Eq. (26) is linear in α, so its optimum on the simplex is attained at a vertex and reduces to the nearest training target. The distance-based coverage error of Eq. (14) can credit a mixture. The experiments report the vertex form.

Remark (the shared encoder as an inductive bias). Nothing in Eq. (4) makes ϵ_{int} small. An isotropic prior in particular does not guarantee that individual code directions correspond to distinct semantic attributes of the decoded text, and alignment of that kind requires additional inductive biases on the model or the data (Locatello et al., 2019). The shared encoder of §3, which produces each posterior from a semantic encoding of a privileged description of its target, is an inductive bias of this kind, because it ties the geometry of the codes to the geometry of those inputs.

Measuring the two errors. Proposition 4.1 holds for every nonempty set $\mathcal{A}$ of interpolation weights. A smaller set lowers ϵ_{int} and raises ϵ_{cov}, so any such choice keeps the bound valid, and the two errors move in opposite directions as weights are added. Restricting coverage to the vertices, the nearest training embedding, can only increase ϵ_{cov}. The protocol used in the experiments is in §B.10.

A.4 Proofs of Proposition 4.2 and Corollary 4.3

This subsection proves the two properties of the expected sampled score $F_\tau(b) := \sum_s P_b^\tau(s) f(s)$ used in §4.3. A query is an unbiased, conditionally sub-Gaussian observation of F_τ, and cumulative regret controls the best sequence sampled during the run. The sum defining F_τ runs over the finitely many terminated sequences of length at most $T_{\max}$, and the task score is bounded, $f(s) \in [f_{\min}, f_{\max}]$ with range $\Delta_f := f_{\max} - f_{\min}$.

Observation model (Proposition 4.2). Conditionally on the history $\mathcal{H}_{t-1}$ and the chosen query b_t, the sequence S_t is drawn from $P_{b_t}^\tau$, so $\mathbb{E}[Y_t \mid \mathcal{H}_{t-1}, b_t] = \sum_s P_{b_t}^\tau(s) f(s) = F_\tau(b_t)$ directly from the definition. For the noise $\xi_t := Y_t - F_\tau(b_t)$, the observation Y_t takes values in an interval of width Δ_f and has conditional mean $F_\tau(b_t)$, so Hoeffding's lemma gives

$$\mathbb{E}\big[e^{\lambda \xi_t} \,\big|\, \mathcal{H}_{t-1}, b_t\big] \;\le\; e^{\lambda^2 \Delta_f^2/8} \qquad \text{for every } \lambda \in \mathbb{R}, \tag{27}$$

so ξ_t is conditionally $(\Delta_f/2)$-sub-Gaussian. The bound holds conditionally on the query, so it is unaffected by the fact that b_t is chosen adaptively from the history.

From cumulative regret to the best sampled sequence.

Proof of Corollary 4.3. We write $R_Q := \sum_{t \le Q} \big(F_\tau^\star - F_\tau(b_t)\big)$. Averaging the identity $Y_t = F_\tau(b_t) + \xi_t$ over $t \le Q$ gives

$$\frac{1}{Q}\sum_{t \le Q} Y_t \;=\; F_\tau^\star \;-\; \frac{R_Q}{Q} \;+\; \frac{1}{Q}\sum_{t \le Q} \xi_t. \tag{28}$$

The noise variables form a martingale difference sequence with respect to the filtration generated by the histories, and each is conditionally $(\Delta_f/2)$-sub-Gaussian by the observation-model paragraph above, so the Azuma–Hoeffding inequality gives, with probability at least $1 - \delta$,

$$\frac{1}{Q}\sum_{t \le Q} \xi_t \;\ge\; -\frac{\Delta_f}{2}\sqrt{\frac{2\log(1/\delta)}{Q}} \;=\; -\Delta_f\sqrt{\frac{\log(1/\delta)}{2Q}}. \tag{29}$$

Since $\max_{t \le Q} Y_t \;\ge\; \frac{1}{Q}\sum_{t \le Q} Y_t$ and $Y_t = f(S_t)$, combining the two displays gives the stated inequality. □

The corollary takes a cumulative-regret bound as an input. Combining it with Proposition 4.1 further requires that the score be Lipschitz in the embedding, which holds for Semantle and remains an assumption for the molecular predictors.

B EXPERIMENTAL DETAILS

This appendix records implementation and protocol details summarized in §5.

B.1 MODELS AND REPRESENTATION TRAINING

Semantle. BOReFT uses Llama-3.2-1B-Instruct (Meta AI, 2024) as the frozen generative model and as the source of the semantic representation used to predict code posteriors. Each target–description pair is formatted as `The meaning of '{word}' is: {definition}`. To construct the posterior representation, we run the sequence through the frozen backbone up to the penultimate decoder layer, apply a copied final decoder block and the model's final RMSNorm, and read the representation at the last token of the definition span. A rank-8 LoRA on the copied final block is trained jointly with the posterior projection and is used only to construct code posteriors; it is not used by the generative model during search. The representation-training set contains 3072 target words paired with dictionary-like descriptions. For this run the three terms in Eq. (4) are weighted by 1, $\beta = 1$, and $\lambda = 1$.

Molecular property optimization. BOReFT uses the chemistry-pretrained Qwen2.5-3B checkpoint released with MiST (Bran et al., 2025) as the frozen generative model; in our experiments this checkpoint is identified as `qwen_pretranined_v6`. Representation training uses ChEBI-20 (Edwards et al., 2021), which pairs canonical SMILES strings with natural-language descriptions. Our prepared corpus contains 8000 pairs, shuffled once, and the primary experiments use a random $N = 1024$ subsample drawn with seed 42. For the checkpoint reported in Table 2, reconstruction, the prior, and self-distillation are weighted by 1, $\beta = 10^{-3}$, and $\lambda = 1$.

Generation follows the MiST completion format in §B.2. The prompt ends in `[START_SMILES]`, and the gold continuation ends in `[END_SMILES]`. The semantic encoder ϕ is the frozen Qwen3-Embedding-0.6B model (Zhang et al., 2025). Each molecule–description pair is formatted as `The molecule '{SMILES}' is: {description}`, and we L2-normalize the resulting embeddings before applying the learned posterior projection g_ψ.

Intervention site. On both tasks the edit in Eq. (1) is applied once, at rank $r = 64$. The layer is the first transformer block, and the edited vector is that block's residual-stream output. Later blocks decode from the edited activation. The token is a dedicated marker written at the start of the task instruction. On Semantle the marker is `<|reserved_special_token_0|>`, and it is the first token of the user turn inside the chat template. On molecular optimization the marker is `<|boreft_0|>`. It is prefixed to the completion string, the tokenizer then adds its beginning-of-sequence token, and `[START_SMILES]` remains the last token of the prompt.

On both tasks, g_ψ is a multilayer perceptron. A linear layer of width 256, a ReLU, and a layer norm are followed by a linear layer of width 128 and a ReLU. Separate linear heads then predict the posterior mean and log-variance in $\mathbb{R}^r$.

B.2 PROMPTS

Semantle uses Llama-3.2-1B-Instruct, an instruction-tuned model. Reconstruction and self-distillation are user turns of its chat template, and generation continues from the assistant header. Molecular property optimization uses the MiST Qwen2.5-3B checkpoint as a completion model. Those prompts are plain prefixes, and the model continues the same string. Each generative instruction below is prefixed with the marker token of §B.1. Embedding strings are plain text on both tasks.

Reconstruction. The student is trained to continue a task prompt with the target. The description is reserved for the teacher and the embedding string.

Semantle, the user turn:

```
Generate an English word (only the word, without any
decoration or formatting).
```

Molecular optimization, the completion prefix. The gold continuation is the SMILES string followed by `[END_SMILES]`.

```
Here is a valid SMILES string for a molecule (only the
SMILES string; no additional text): [START_SMILES]
```

Self-distillation. The teacher is the unintervened frozen model. Its prompt includes the description of the training target. The student is still conditioned on the reconstruction prompt.

Semantle, the user turn. {definition} is the dictionary text of the target word.

```
Here is the definition of an English word: {definition}
Generate a word that matches this definition (only the word,
without any decoration or formatting).
```

Molecular optimization. {definition} is the molecule description.

```
Here is the description of a molecule: {definition}
A valid SMILES string for such a molecule (only the SMILES
string; no additional text): [START_SMILES]
```

Embedding. These strings are the inputs used to predict code posteriors.

Semantle. The string is read by the copied final block of Llama-3.2-1B-Instruct, at the last token of the definition span.

```
The meaning of '{word}' is: {definition}
```

Molecular optimization. Qwen3-Embedding-0.6B encodes the string, and we L2-normalize the result before g_ψ.

```
The molecule '{SMILES}' is: {description}
```

Definition generation. The Semantle definitions paired with the representation-training words were written by Claude Opus 5 (Anthropic, 2026).

Search. Semantle decodes from the reconstruction prompt above. The molecular runs in Table 2 replace it with an objective line followed by the same completion prefix, including `[START_SMILES]`. For DRD2 the prompt is

```
The task is to optimize for DRD2 binding.
Here is a valid SMILES string for a molecule (only the
SMILES string; no additional text): [START_SMILES]
```

GSK3β and JNK3 keep that second line. Their first lines are "The task is to optimize for GSK3β (GSK3B) inhibition." and "The task is to optimize for JNK3 inhibition."

B.3 Evaluation protocols

Semantle. We evaluate ten hidden targets: five sampled from the 3072 representation-training words and five from the held-out reconstruction-test pool, with three independent runs per target. Target selection uses seed 42, and these words are distinct from the evaluation targets used in BO-PRO and MiGrATe. Each run has a budget of 500 objective evaluations. For each target and seed, all methods receive the same ten labeled warm-start word–score pairs, drawn from the training vocabulary and excluding the hidden target. BOReFT uses greedy decoding during Semantle search. The surrogate is a Gaussian process with an ARD Matérn-2.5 kernel and log expected improvement, unless varied in §B.12. Before the kernel, each code is normalized to the search box and passed through one linear layer of width 64 followed by an ELU. That feature map is trained jointly with the Gaussian process, the deep-kernel construction of Wilson et al. (2016) in the form used by GOLLuM (Ranković & Schwaller, 2025). Molecular search uses the same surrogate. Unless varied in §B.8 and §B.11, BOReFT searches the axis-aligned bounding box of the learned posterior means in Eq. (8).

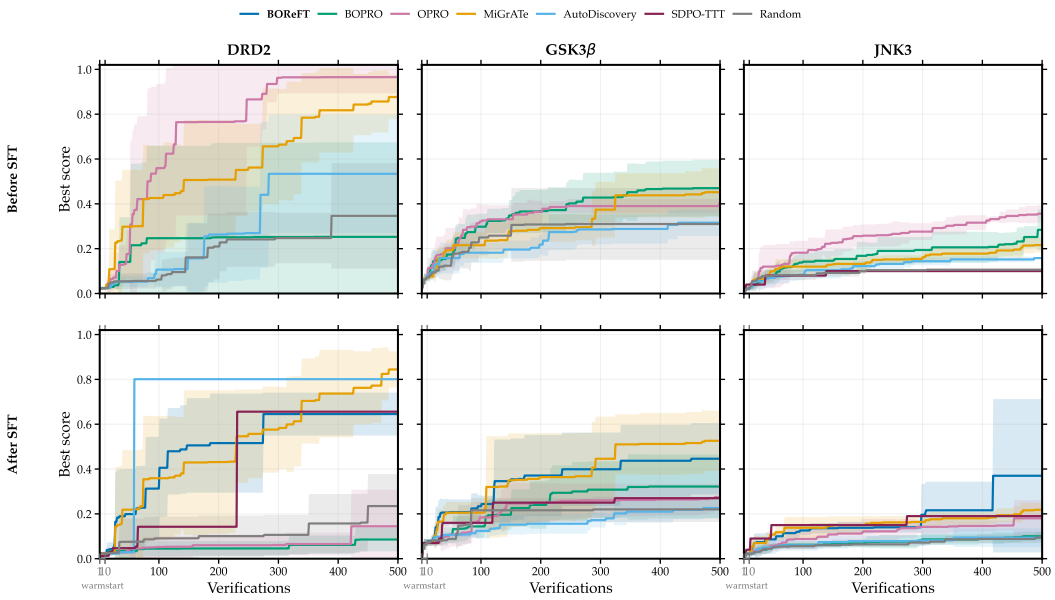


Figure 5: **Search against baselines on molecular property optimization.** Best-so-far property at each verification, mean $\pm$ 1 SD across seeds. The top row is before SFT and the bottom row is after SFT. The first ten evaluations are shared warm starts. We plot BOReFT only after SFT; its scores are repeated in the before-SFT columns of Table 2. SDPO-TTT, AutoDiscovery, and MiGrATe are the no-instruction runs. A curve with no band is a single finished seed.

Molecular property optimization. We evaluate DRD2, GSK3β, and JNK3 using the molecular property scores described in §5.1. Invalid SMILES receive score 0. All three oracles are class-1 probabilities in $[0, 1]$ (sklearn 1.4+ no longer normalizes TDC's stored leaf counts, so we restore that normalization at load time). Each method receives a budget of 500 property evaluations and is evaluated over five independent seeds. We draw ten Sobol points in the learned BOReFT domain, decode them once at temperature 1, and use the resulting molecules as the common warm-start proposals; their scores are computed separately under each objective. BOReFT samples at temperature 1 during molecular search. Figure 5 shows the before- and after-SFT best-so-far curves for the runs in Table 2.

B.4 Search baselines

Table 3 lists the proposal rule and per-round query count used by each search method. Unless a row says otherwise, methods share the objective-evaluation budget and the same warm-start observations. On Semantle the base model is Llama-3.2-1B-Instruct; on molecular optimization it is the MiST chemistry-pretrained Qwen2.5-3B checkpoint of §B.1. Prompt-based methods sample at temperature 1. Molecular completions use the MiST `[START_SMILES]` / `[END_SMILES]` format described above.

B.5 Proposal behavior on Semantle

Table 1 shows that SDPO-TTT, AutoDiscovery, and MiGrATe recover at most one held-out target. This section describes the proposals behind those results. All three methods use the settings in Table 3 and the same warm starts as the other Semantle runs. Averaged over the 30 runs, the best warm-start similarity is 0.683. Averages below are over those runs, and they exclude the warm-start proposals.

SDPO-TTT. Each proposal is sampled from the task description. That prompt does not list previously scored words. The reverse-KL update conditions the teacher on the most recent proposal and its score. Across these runs the most common proposal is the word "cloud", which is not a warm-start word. It accounts for 0.28 of the first 30 proposals and 0.32 of the last 100. In every

Method	Update	Proposal rule	Scored / round
Random (base)	none	task prompt only	1
Random (post-SFT)	none	same, after LoRA-SFT on the training set	1
OPRO	none	scored-history in-context prompt	1
BOPRO	GP over embeddings	OPRO prompt over 5 nearest neighbors	1
AutoDiscovery	UCB1 MCTS	OPRO prompt on the selected branch	1 of 8
MiGrATe	LoRA GRPO	mixed on-policy / greedy / neighborhood group	4
SDPO-TTT	LoRA self-distillation	task prompt after a reverse-KL update	1
Discrete BO	GP over the training set	acquisition over that finite pool	1
BOReFT	GP over codes	decode from a queried code b	1

Table 3: **Search-method settings.** All methods use the same base model within a task. AutoDiscovery expands eight completions per node and scores one; the rest remain as untried children. MiGrATe scores four new completions per round (two on-policy and two neighborhood); greedy group members are reused from history. BOPRO's GP is fit in the Qwen3 embedding space of observed solutions, not in the BOReFT code space.

run, at least 0.29 of the proposals after the warm start are this word. The mean proposal similarity is 0.633 over the first 50 proposals and 0.630 over the last 50, below the warm-start best of 0.683. Later proposals do not score higher than earlier ones. Table 4 reports a best similarity of 0.763 on training targets, and Table 1 reports 0.709 on held-out targets, with no exact matches.

SDPO forms its teacher by conditioning the current model on feedback from the attempt, then distills that teacher into the student (Hübotter et al., 2026). The method is designed for rich environment feedback, such as a runtime error or a written judgment, which tells the teacher something the student did not see. When the environment returns only a scalar reward, the same work uses a successful attempt as feedback for the failures. Semantle returns only a scalar cosine similarity, and these runs never produce the hidden word for the teacher to condition on. The scalar attached to the latest word is the teacher's only additional context. The teacher distribution can therefore stay close to the unconditioned proposal distribution, and the distillation target supplies little additional supervision. This reading is consistent with proposal similarities that stay flat across the budget.

AutoDiscovery. Each proposal is drawn from a prompt built on the selected branch. The prompt lists scored words on that branch and asks for a word with a higher score. The mean proposal similarity is 0.624 over the first 50 proposals and 0.624 over the last 50, below the mean best warm-start similarity. A proposal exceeds the best score observed so far on 0.006 of search steps. The repetition rate in Table 5 is 0.364, so most proposals are new words. No run records an exact match. The final similarities are 0.772 on training targets and 0.757 on held-out targets.

Exploration is strong relative to exploitation. The search keeps introducing new words, and those words rarely improve on the best score already observed. That balance can suit the setting AutoDiscovery was introduced for, open-ended search over scientific hypotheses, where continued exploration is part of the goal (Agarwal et al., 2025b). Semantle asks for one hidden word, so search needs proposals that improve on the best score already observed. The runs here already move toward exploitation relative to the published defaults. The exploration constant is $C = 0.5$ rather than $C = 2.0$, and the prompt includes 20 nodes from the selected branch rather than 3. Even so, the mean proposal similarity does not rise. A different acquisition function, or a further change in these hyperparameters, might exploit high-scoring guesses more directly. We evaluate one fixed configuration and do not sweep those choices.

MiGrATe. MiGrATe updates a LoRA adapter after each group of proposals. The group mixes samples from the task prompt with variants of high-scoring words already observed (Phan et al., 2025). The mean proposal similarity rises from 0.637 over the first 50 proposals to 0.742 over the last 50. The fraction of proposals equal to "cloud" falls from 0.27 in the first 30 proposals to 0.03 in the last 100. The test-time GRPO update does change the proposal distribution on this task. Exact-match recovery remains low, at $0/15$ on training targets and $1/15$ on held-out targets, and the repetition rate is 0.678.

The same update makes test-time search slower than BOReFT. Each round samples a group and takes a gradient step on the adapter. BOReFT fits a Gaussian process and decodes one code, with no gradient step. On these runs the mean elapsed time is 0.35 hours for MiGrATe and 0.10 hours for

	Before SFT		After SFT	
Method	Exact↑	Sim.↑	Exact↑	Sim.↑
Discrete BO	*15/15*	*1.000*	–	–
Random	$0/15$	0.758	$3/15$	0.825
SDPO-TTT	$0/15$	0.763	$0/15$	0.766
AutoDiscovery	$0/15$	0.772	$2/15$	0.811
MiGrATe	$0/15$	0.788	$0/15$	0.782
BOPRO	$5/15$	$\underline{0.878}$	$\underline{5/15}$	0.857
OPRO	$\mathbf{7/15}$	$\mathbf{0.886}$	$4/15$	$\underline{0.858}$
BOReFT	$\underline{6/15}$	0.863	$\mathbf{6/15}$	$\mathbf{0.863}$

Table 4: **Semantle search on training targets, before and after supervised fine-tuning.** The table reports exact-match counts and mean best semantic similarity on five targets from the representation-training set, over fifteen runs. The left block uses the frozen generator. The right block uses the LoRA adapter of §B.6. BOReFT does not use that adapter. Discrete BO searches the training vocabulary, so the right block has no entry for it. In each column, bold marks the best value and underline marks the second best. The italic row is a reference, and it is not included in that ranking.

BOReFT, so MiGrATe takes about $3.5\times$ as long at the same budget of 500 scores. The comparison is search time only. It does not include the cost of BOReFT's representation training.

B.6 Search on training targets

Table 4 reports exact match and mean best similarity when the hidden word is one of the five targets drawn from the representation-training set. The protocol matches Table 1: three seeds, budget 500, and the same ten warm starts. The after-training columns use one LoRA adapter, trained by supervised fine-tuning on the 3072 representation-training words for 10 epochs, and shared by Random, SDPO-TTT, AutoDiscovery, MiGrATe, BOPRO, and OPRO. BOReFT does not use that adapter, so its entries repeat.

Held-out results for these runs are in Table 1. None of the fine-tuned baselines matches BOReFT's $8/15$ held-out exact matches. On training targets, discrete BO reaches $15/15$ because those words are its candidate set. After fine-tuning, OPRO's training-target exact-match count falls from $7/15$ to $4/15$, while random sampling rises from $0/15$ to $3/15$.

Novelty on the pooled runs drops for AutoDiscovery, BOPRO, and OPRO, from 0.736, 0.798, and 0.708 before fine-tuning to 0.154, 0.203, and 0.212 after it (Table 1). More of their proposals are words from the representation-training set. SDPO-TTT and MiGrATe stay near the base novelty rates.

B.7 Proposal repetition

Table 5 collects the repetition rate for the main comparisons on both tasks. A proposal counts as a repeat when the same solution already occurred earlier in that run, including during the warm start. Warm-start proposals are excluded from the denominator. On Semantle we lower-case each word and collapse repeated whitespace, and the rate pools every post-warm-start proposal from the thirty runs on the training and held-out targets. On the molecular task, valid SMILES strings are canonicalized before this comparison, so two strings for the same molecule are not counted as distinct. Invalidity is tracked separately from repetition. Molecular rates pool the finished seeds on DRD2, GSK3β, and JNK3 from the runs in Table 2. Novelty, which asks whether a proposal falls outside the representation-training set, remains in Table 1 and is not repeated here.

B.8 Search-domain construction

The primary BOReFT runs search the axis-aligned bounding box (AABB) of the learned posterior means, Eq. (8). The variational posterior also produces a per-target standard deviation σ_i, so the same checkpoint admits other compact domains that still cover those means. We evaluate two alternatives on Semantle, keeping the checkpoint, warm starts, budget, and acquisition unchanged.

Method	Semantle		Molecules	
	Before SFT	After SFT	Before SFT	After SFT
Discrete BO	*0.000*	*0.000*	–	–
Random	0.791	**0.201**	0.212	<u>0.068</u>
SDPO-TTT	0.750	0.747	0.176	0.183
AutoDiscovery	**0.364**	<u>0.218</u>	<u>0.075</u>	**0.066**
MiGrATe	0.678	0.650	0.112	0.102
BOPRO	0.527	0.342	**0.039**	0.094
OPRO	0.540	0.368	0.223	0.541
BOReFT	<u>0.481</u>	0.481	0.133	0.133

Table 5: **Proposal repetition before and after supervised fine-tuning.** Each entry is the fraction of post-warm-start proposals that repeat an earlier solution in the same run. Lower is better. Semantle pools the training-target and held-out runs. Molecular rates pool finished seeds on DRD2, GSK3β, and JNK3. SDPO-TTT before SFT is the one finished JNK3 seed. After SFT, SDPO-TTT is one finished seed on each property, and AutoDiscovery omits four unfinished DRD2 seeds. Bold and underline mark the best and second-best rate in each column. The italic row is a reference, and it is not ranked. BOReFT does not use the fine-tuning adapter, so its entries repeat.

The first generalizes the AABB to smoothly expand each coordinate proportional to its standard deviation,

$$\prod_{j=1}^{r} \left[\min_i (\mu_{ij} - k\sigma_{ij}),\ \max_i (\mu_{ij} + k\sigma_{ij}) \right], \tag{30}$$

where $k = 0$ recovers the original bounds. The second replaces the box with a covering ellipsoid (Siivola et al., 2021) of the same means. That ellipsoid is the Mahalanobis ball of their sample covariance, scaled so that every μ_i lies inside, and the acquisition is optimized over this convex set.

Table 6 reports the results together with each domain's Lebesgue volume relative to the mean box V_0 and the repetition rate.

Expanding the box by $k = 0.1$ multiplies the volume by 4.71×10^{5}. Exact-match recovery falls to $3/15$ on training targets and $4/15$ on held-out targets, and the repetition rate rises to 0.627 and 0.577. At $k = 0.5$ the volume is 1.08×10^{21} times larger, no run recovers an exact match, and the repetition rate is 0.851 on training targets and 0.844 on held-out targets. A full standard-deviation expansion ($k = 1$) yields a volume 7.31×10^{32} times that of the mean box and likewise recovers no exact matches. Mean best similarity falls to 0.726 on training targets and 0.707 on held-out targets, which is comparable to random sampling from the frozen base model in Table 1. The repetition rate is then 0.943 on training targets and 0.913 on held-out targets. On this checkpoint, the typical posterior standard deviation is on the order of the coordinate-wise span of the means, so even a modest k in Eq. (30) enlarges the domain substantially.

The covering ellipsoid has smaller volume than the mean box ($4.08\times10^{-4}\,V_0$) and remains closer to it in search performance. It finds $5/15$ training targets and $5/15$ held-out targets, against $6/15$ and $8/15$ for the mean box, with mean similarities 0.873 and 0.852 and repetition rates 0.594 and 0.605. On this protocol, replacing the box with that covering ellipsoid does not improve exact-match recovery, while enlarging the box to include posterior spread reduces exact-match recovery and raises the repetition rate.

B.9 Geometric prior

The shared map g_ψ predicts each code posterior from a frozen semantic representation of the representation-training input. We call the geometry of these representations the geometric prior. Figure 6 shows the first two principal components of that prior. The top row is the ChEBI-20 set of molecules with their descriptions, and the bottom row is the Semantle set of words with their definitions. Within each row, points that share a semantic category use the same color. The left column is Qwen3-Embedding-0.6B. That encoder supplies ϕ on molecular optimization, and it is also the model used for the Semantle task score. The right column is the task language model: the chemistry-pretrained MiST checkpoint on molecular optimization, and Llama-3.2-1B-Instruct on

Search domain	V/V_0	Train			Held out		
		Exact↑	Sim.↑	Rep.↓	Exact↑	Sim.↑	Rep.↓
AABB							
at $k = 0$	1	6/15	0.863	0.505	8/15	0.892	0.452
at $k = 0.1$	4.71×10^{5}	3/15	0.819	0.627	4/15	0.836	0.577
at $k = 0.5$	1.08×10^{21}	0/15	0.758	0.851	0/15	0.746	0.844
at $k = 1$	7.31×10^{32}	0/15	0.726	0.943	0/15	0.707	0.913
Covering ellipsoid	4.08×10^{-4}	5/15	0.873	0.594	5/15	0.852	0.605

Table 6: **Semantle search under alternative domains.** Exact matches and mean best semantic similarity, using the protocol of Table 1 on the same checkpoint. Repetition (↓) uses the definition in Table 5, pooled within each split: the fraction of post-warm-start proposals whose solution repeats an earlier proposal in the same run. V/V_0 is the Lebesgue volume of the search domain relative to the mean box ($k = 0$). The axis-aligned bounding boxes follow Eq. (30), and the covering ellipsoid is as described in §B.8 over the posterior means.

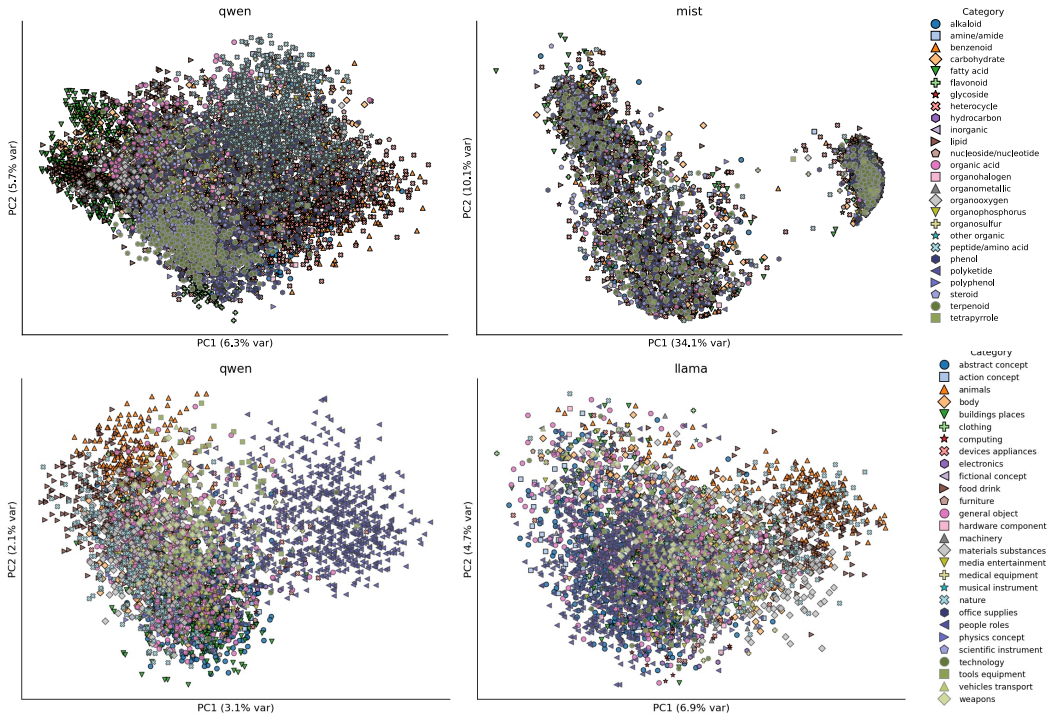


Figure 6: **Geometric prior via semantic representations.** First two principal components of the frozen semantic representations that supply the geometric prior. The top row is molecular optimization and the bottom row is Semantle. The left column is Qwen3-Embedding-0.6B. The right column is the MiST chemistry-pretrained checkpoint on the top row and Llama-3.2-1B-Instruct on the bottom row. Points are colored by semantic category, with one palette per row. These are the representations passed to g_ψ. The learned Semantle codes are shown separately in Fig. 2.

Semantle. The Llama states are read from the last layer at the last instruction token, which is how ϕ is constructed on Semantle.

B.10 Interpolation in the learned code space

Figure 2 shows qualitative interpolations in a PCA projection of the 3072 Semantle training posterior means. Paths run from "research" to four distant anchors, and temperature-sampled decodes along each path follow distinct semantic transitions.

	Train			Held out		
Temperature	Exact↑	Sim.↑	Rep.↓	Exact↑	Sim.↑	Rep.↓
$T = 0$	$6/15$	0.863	0.505	$8/15$	0.892	0.452
$T = 0.25$	$3/15$	0.830	0.455	$5/15$	0.860	0.412
$T = 0.50$	$5/15$	0.861	0.333	$4/15$	0.839	0.345
$T = 0.75$	$1/15$	0.807	0.234	$3/15$	0.839	0.231

Table 7: **Semantle search under decoding temperature.** Exact matches, mean best semantic similarity, and repetition rate, using the protocol of Table 1 on the same checkpoint and the mean-box domain of Eq. (8). Repetition is defined as in Table 6. $T = 0$ is greedy decoding as in the primary runs.

	Train			Held out		
Kernel	Exact↑	Sim.↑	Rep.↓	Exact↑	Sim.↑	Rep.↓
Matérn-2.5	$6/15$	0.863	0.505	$8/15$	0.892	0.452
Matérn-1.5	$3/15$	0.834	0.533	$6/15$	0.865	0.454
Matérn-0.5	$5/15$	0.857	0.321	$4/15$	0.857	0.365
RBF	$4/15$	0.837	0.493	$4/15$	0.832	0.498

Table 8: **Semantle search under GP kernel.** Exact matches, mean best semantic similarity, and repetition rate, using the protocol of Table 1 on the same checkpoint and the mean-box domain of Eq. (8). Repetition is defined as in Table 6. All kernels use ARD lengthscales. Matérn-2.5 is the primary setting.

To measure interpolation across the space, we decode temperature-1 samples along linear paths between 40 training-target pairs at the mid-path weights $t \in \{0.4, 0.5, 0.6\}$. At each weight we take the mean decoded embedding and its Euclidean distance to the semantic interpolant x_α of Eq. (13). Table 10 reports the median of these distances, written Δ_{interp}. An endpoint-only reference that always emits the nearer of the two training words, with no interpolation, has $\Delta_{\text{interp}} = 0.365$. BOReFT obtains 0.338. Removing self-distillation or variational training yields 0.355 and 0.356. Removing reconstruction or the shared encoder yields 0.396 and 0.382, both larger than the endpoint-only reference.

B.11 Effect of temperature on search

The primary Semantle protocol decodes greedily ($T = 0$). We also search the same mean box at sampling temperatures $T \in \{0.25, 0.50, 0.75\}$, keeping the checkpoint, warm starts, budget, and acquisition unchanged.

Table 7 reports the results. Raising the temperature lowers the repetition rate, from 0.505 and 0.452 under greedy decoding to 0.234 and 0.231 at $T = 0.75$. Exact-match recovery is highest at $T = 0$, with $6/15$ training targets and $8/15$ held-out targets. At $T = 0.50$, training recovery is $5/15$ and held-out recovery is $4/15$. Held-out similarity is also highest at $T = 0$.

B.12 Effect of GP kernel on search

The primary Semantle runs use an ARD Matérn-2.5 covariance in the Gaussian-process surrogate. We also compare Matérn-1.5, Matérn-0.5, and the squared-exponential (RBF) kernel on the same checkpoint, warm starts, budget, acquisition, and mean-box domain, changing only the kernel family.

Table 8 reports the results. Matérn-2.5 recovers the most targets on both splits ($6/15$ training, $8/15$ held out). The rougher Matérn kernels and the smoother RBF kernel each reduce exact-match recovery, with the largest drop on held-out targets ($4/15$ for Matérn-0.5 and RBF). Mean best similarity follows the same ordering. Under Matérn-0.5 the repetition rate falls to 0.321 on training targets and 0.365 on held-out targets, and held-out exact-match recovery is lower than under Matérn-2.5. On this protocol, Matérn-2.5 remains the strongest default among the kernels we tested.

	Train			Held out		
Acquisition	Exact↑	Sim.↑	Rep.↓	Exact↑	Sim.↑	Rep.↓
LogEI	6/15	0.863	0.505	8/15	0.892	0.452
UCB	4/15	0.845	0.461	6/15	0.865	0.442
Thompson sampling	2/15	0.813	0.201	3/15	0.826	0.205

Table 9: **Semantle search under acquisition function.** Exact matches, mean best semantic similarity, and repetition rate, using the protocol of Table 1 on the same checkpoint and the mean-box domain of Eq. (8). Repetition is defined as in Table 6. LogEI is the primary setting. UCB uses coefficient $\beta = 0.2$.

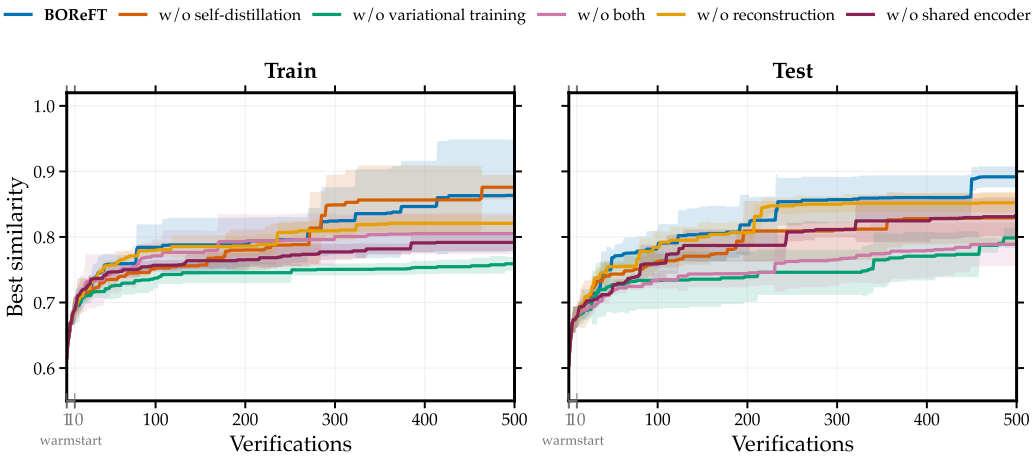


Figure 7: **Search under training ablations.** Best-so-far semantic similarity (mean ± 1 SD across 3 seeds) on five training and five held-out Semantle targets. The first ten objective evaluations are shared warm starts.

B.13 Effect of acquisition function on search

The primary Semantle runs select the next code by log expected improvement. We also compare upper confidence bound, with coefficient $\beta = 0.2$, and Thompson sampling on the same checkpoint, warm starts, budget, ARD Matérn-2.5 kernel, and mean-box domain.

Table 9 reports the results. Log expected improvement recovers the most targets on both splits ($6/15$ training, $8/15$ held out). Upper confidence bound recovers $4/15$ and $6/15$. Thompson sampling recovers $2/15$ and $3/15$, with repetition rates 0.201 and 0.205. On this protocol, log expected improvement remains the strongest default among the acquisition functions we tested.

B.14 Training ablation numbers

Figure 7 shows the anytime curves for the runs summarized on the right of Fig. 4. Table 10 gives the train and held-out split. Its interpolation column is the median distance defined in §B.10: temperature-1 decodes along 40 training-target pairs at $t \in \{0.4, 0.5, 0.6\}$, compared with an endpoint-only reference at 0.365. Similarity in that panel is the unweighted mean of the two split means below, and the exact-match label is their sum.

B.15 Effect of intervention rank

The primary Semantle runs use intervention rank $r = 64$. We also vary the rank at fixed training-set size $N = 3072$, keeping the rest of the protocol in §5.2 unchanged. Figure 8 reports mean best semantic similarity and exact-match rate, pooled over the same ten targets and three seeds as Fig. 4. Rank 64 gives the strongest exact-match result ($14/30$). Performance is not monotone in rank, with a pronounced dip at rank 32.

Method	Train		Held out		
	Exact↑	Sim.↑	Exact↑	Sim.↑	Δ_{interp} ↓
BOReFT	**6/15**	0.863	**8/15**	**0.892**	**0.338**
w/o self-distillation	**6/15**	**0.876**	3/15	0.829	0.355
w/o reconstruction	2/15	0.821	5/15	0.852	0.396
w/o shared encoder	1/15	0.792	4/15	0.833	0.382
w/o variational training	0/15	0.759	3/15	0.799	0.356

Table 10: **Search and interpolation in ablated spaces.** Exact matches and mean best semantic similarity on Semantle, split by whether the hidden target was included in representation training (15 runs per split: 5 targets × 3 seeds). Δ_{interp} is the median distance from the average embedding of temperature-1 decodes to the semantic interpolant x_α, over 40 target pairs at $t \in \{0.4, 0.5, 0.6\}$. An endpoint-only reference that always emits the nearer training word has $\Delta_{\text{interp}} = 0.365$.

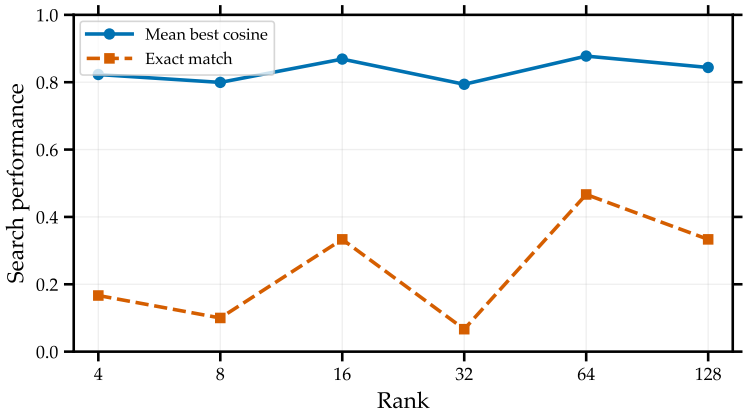


Figure 8: **Search vs. intervention rank.** Mean best semantic similarity (solid) and exact-match rate (dashed) at $N = 3072$, pooled over 10 targets × 3 seeds.

C Additional related work

This section records comparisons that sit next to §2 but are not needed to follow it.

Test-time search and adaptation. OPRO conditions generation on scored candidates (Yang et al., 2024), Tree of Thoughts searches intermediate states (Yao et al., 2023), and AutoDiscovery searches a tree of hypotheses (Agarwal et al., 2025b). Test-time training updates parameters at inference (Sun et al., 2020; Hardt & Sun, 2024; Akyürek et al., 2025). MiGrATe uses mixed-policy GRPO (Phan et al., 2025), and SDPO distills from feedback on hard problems (Hübotter et al., 2026). BOReFT learns its search domain before inference and then holds the generator and the domain fixed.

Verbalizable representations. Beyond sparse autoencoders, the Jacobian lens and its associated J-space identify representations that can be verbalized as text (Gurnee et al., 2026). These results describe structure that is already present in a model's activations. BOReFT does not try to recover that structure. It learns a separate set of interventions from target solutions.

Monitoring and manifold steering. Representation Engineering uses population-level representations for monitoring as well as for manipulation (Zou et al., 2023). Manifold-steering methods go beyond linear directions by fitting activation manifolds and intervening along paths that respect their learned geometry (Wurgaft et al., 2026). Both lines of work aim to describe or follow geometry that the model already has. The BOReFT intervention is instead a learned search domain, and its coordinates are chosen by an external optimizer.