Steering Sequence Generation in Protein Language Models through Iterative Lookback Monte Carlo Sampling

Anonymous Authors[1]

Abstract

Protein language models (pLMs) leverage large-scale evolutionary data to generate novel sequences, but steering generation toward desired physicochemical properties without sacrificing diversity remains a major challenge. Existing approaches often induce severe diversity loss or require computationally expensive retraining. We introduce **Iterative Lookback Monte Carlo** (ILMC), a training-free inference-time sampling strategy that interleaves autoregressive elongation with Metropolis–Hastings refinement to approximate sampling from a maximum-entropy target distribution balancing generative quality and steering objectives. We show theoretically that this target distribution is entropy-maximizing under fixed generative quality and steering constraints, and empirically that ILMC produces more diverse samples than standard autoregressive baselines at matched generative quality. Using simple steering potentials, ILMC improves desired molecular properties, including generating proteins with up to $12^{\circ}\mathrm{C}$ higher predicted melting temperature than compute-matched alternative strategies. ILMC naturally applies to classifier-guided steering, where it outperforms purely autoregressive guidance in diversity while maintaining comparable enrichment of target properties. We validate ILMC on family-specific pLMs and on the multi-family model ProGen3.

1. Introduction

Protein Language Models (pLMs) trained on millions of natural proteins have demonstrated a remarkable ability to capture complex evolutionary rules governing protein structure and function (Madani et al., 2023; Lin et al., 2023). These models learn a probability landscape where regions of high probability correspond to biologically active proteins. Sampling from the model can result in novel sequences whose function has been verified in experiments. Higher-probability samples are more likely to be functional in experiments (Russ et al., 2020; Fields et al., 2025), and practitioners often "steer" generation to artificially enrich for such high-probability variants (Lambert et al., 2025; Fernandez-de Cossio-Diaz et al., 2025). In textual large language models (LLMs), high-probability sampling (also called *distribution sharpening*) can elicit reasoning-like capabilities (Karan & Du, 2025), but existing methods often result in critical diversity loss (He et al., 2025; Song et al., 2025; Calvanese et al., 2025) or require computationally expensive retraining (Kim et al., 2024; Stocco et al., 2025). Furthermore, beyond sampling viable sequences, a critical challenge in protein engineering is to design specific attributes important in real-world applications, such as thermostability, or finding variants within the mutational vicinity of a wild-type sequence to preserve existing compatibilities, rather than generating totally new sequences.

To bridge the gap between unconditional generation and functional design, standard approaches rely on retraining or fine-tuning (Nijkamp et al., 2023). While effective, these methods are computationally demanding and prone to "catastrophic forgetting," where the model over-optimizes the target property but neglects broader functional priors learned during pre-training. More recently, Reinforcement Learning (RL) has been employed to align models with desired rewards. Yet, as noted in recent literature, RL-based distribution sharpening often leads to a collapse in generation diversity (Song et al., 2025).

Our contribution. We introduce Iterative Lookback Monte Carlo Sampling (ILMC), a training-free inference-time method for controllable generation in protein language models. ILMC targets a maximum-entropy distribution that balances the generative quality of the base model with a user-defined steering potential. Algorithmically, it interleaves standard autoregressive elongation with Metropolis–Hastings (MH) refinement steps that revisit and resample recent segments of the sequence. This correction allows ILMC to steer generation toward desired regions of sequence space

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

without gradient updates or model retraining.

We show theoretically that distribution targeted by ILMC is the unique maximum-entropy distribution under fixed generative quality and steering constraints. Empirically, ILMC yields more diverse samples than standard autoregressive baselines at matched generative quality. We demonstrate steering toward enhanced thermostability and toward the mutational neighborhoods of a wild-type sequence. ILMC extends naturally to classifier-guided steering for non-decomposable objectives, and evaluate this setting on a homodimerization structural-class task. Across Chorismate Mutase, Phage Lysozyme, and Response Regulator families, and in both family-specific pLMs and ProGen3, ILMC consistently improves the diversity–quality trade-off for steered generation. Code and data available at the project repository and on Zenodo.

2. Generation Steering and Maximum Entropy

Consider a generative model defining a probability distribution $P(\mathbf{a})$ over protein sequences $\mathbf{a} = (a_1, a_2, \ldots, a_L)$, where $a_i \in \mathcal{A}$, the set of 20 amino acids. In statistical physics terms, this model characterizes an energy landscape $E(\mathbf{a}) = -\log P(\mathbf{a})$. Sequences with lower energy are more probable under the model and are more likely to be functional in experiments (Russ et al., 2020; Lambert et al., 2025; Fernandez-de Cossio-Diaz et al., 2025).

Besides sampling functional variants, one may be interested in potentiating certain attributes of designed sequences, such as thermostability. Formally, we introduce a steering potential $S(\mathbf{a})$, with the aim of generating sequences that minimize S. Importantly, $S(\mathbf{a})$ is not a generative score; sequences with low S but high E may not be valid, biologically active protein sequences. Conversely, functional sequences with low E may have high S. It is therefore important to consider both scores simultaneously.

We seek a sampling strategy $Q(\mathbf{a})$ satisfying **two constraints**: **i)** An expected steering score, $\mathbb{E}_Q[S(\mathbf{a})] = C_{\mathrm{S}}$; and **ii)** An expected generative quality (or cross-entropy) $\mathbb{E}_Q[E(\mathbf{a})] = C_{\mathrm{E}}$. Among all such sampling strategies, we prefer the one that remains as diverse as possible, which leads naturally to a maximum-entropy formulation:

Proposition 2.1. *Among all distributions $Q(\mathbf{a})$ satisfying these two constraints, the one with the highest entropy $H[Q] = -\sum_{\mathbf{a}} Q(\mathbf{a}) \log Q(\mathbf{a})$ has the form:*

$$Q^*(\mathbf{a}) \propto \exp\left(-\beta E(\mathbf{a}) - \lambda S(\mathbf{a})\right) \tag{1}$$

where β and λ are Lagrange multipliers chosen to satisfy the target expectations C_{E} and C_{S}.

While Energy-Based Models (EBMs) are naturally suited to facilitate sampling from such distributions via Monte Carlo (MC) techniques (like Gibbs sampling or MH), applying similar methods to autoregressive models poses significant challenges (Karan & Du, 2025). An important distinction is that the inverse temperature parameter β can be misinterpreted in the context of the autoregressive LLMs literature, where "low-temperature sampling" often refers to the local sharpening of the next-token distribution. In that case, logits over the vocabulary of the next token are multiplied by β. We refer to this variant as *greedy* autoregressive low-temperature sampling. See Appendix C for further discussion on why these two approaches are fundamentally different and the computational obstacles to correctly sampling from (1). In fact, any other sampling strategy (top-p, beam search, etc.) (Radford et al., 2019; Holtzman et al., 2019) necessarily produces less diverse samples than Q^*.

Given the difficulty of sampling directly from (1), it is useful to characterize how other sampling strategies Q relate to this optimal distribution.

Proposition 2.2. *For any other distributions $Q_1(\mathbf{a}), Q_2(\mathbf{a})$ satisfying the same constraints:*

$$H[Q_1] > H[Q_2] \Leftrightarrow D_{\mathrm{KL}}(Q_1 \| Q^*) < D_{\mathrm{KL}}(Q_2 \| Q^*) \tag{2}$$

where D_{KL} is the Kullback-Leibler (KL) divergence.

See Appendices B.2 and B.1 for the proofs of 2.1 and 2.2.

In the next section we introduce ILMC, an algorithm to approximate sampling from (1). Beyond enabling attribute-targeted generation (a challenge in standard autoregressive pLMs), 2.2 ensures that approximating Q^* (in the sense of minimizing $D_{\mathrm{KL}}(Q \| Q^*)$) automatically preserves sample diversity (in the sense of maximizing $H[Q]$).

3. Iterative Lookback Monte Carlo (ILMC)

Sampling from the base model $P(\mathbf{a})$, corresponding to setting $\beta = 1$ and $\lambda = 0$ in (1), is computationally cheap in standard pLMs (Madani et al., 2023; Bhatnagar et al., 2025), which can be sampled efficiently via autoregressive decoding. ILMC leverages this base sampler to draw sequences from the tilted target Q^* at $\beta > 1$ or $\lambda > 0$. We extend a recently proposed method (Karan & Du, 2025) to approximate sampling from distributions of the form $\propto P(\mathbf{a})^{\beta}$, a particular case of (1) with $\lambda = 0$, *i.e.*, without steering. Here we extend this framework to incorporate a steering potential $S(\mathbf{a})$.

To describe our sampling algorithm, we consider an intermediate iteration where a sequence prefix $\mathbf{a}_{\leq t} = (a_1, \ldots, a_t)$ has already been generated. We introduce a partial generative energy and steering potential defined on the prefix:

- **Partial Generative Energy** $E_t(\mathbf{a}_{\leq t})$, defined as the cumulative negative log-likelihood of the prefix under

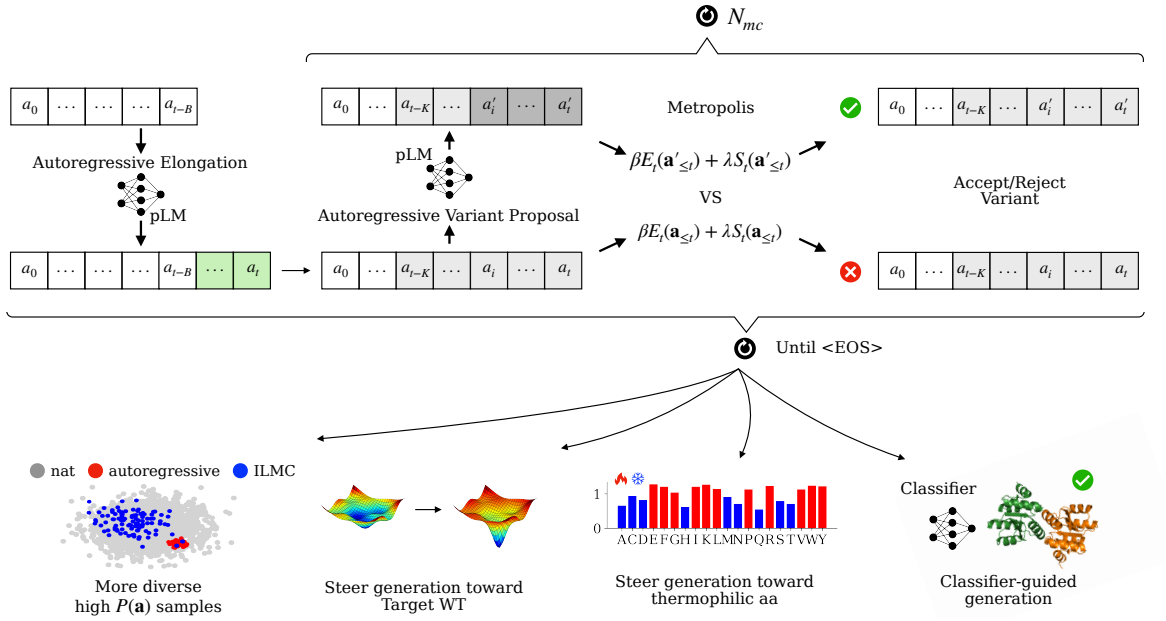

Figure 1. **Iterative Lookback Monte Carlo (ILMC)** sampling framework. The algorithm approximates sampling from the target $Q^*(\mathbf{a})$, (1), by interleaving two phases: Autoregressive Elongation and MH Refinement. In the elongation phase, the pLM extends the sequence prefix by a block of tokens. In the refinement phase, a variant is proposed by resampling a segment of the generated history using the pLM; this variant is accepted or rejected based on a global potential that combines the generative energy E of the model and a user-defined steering potential S. This procedure promotes higher sampling diversity and enables the steering of the generation toward regions of the sequence space that satisfy specific properties, such as thermostability or vicinity to a target wild-type, without requiring gradient updates or model retraining.

the base generative model:

$$E_t(\mathbf{a}_{\leq t}) = -\sum_{i=1}^{t} \log P(a_i|\mathbf{a}_{<i}) \tag{3}$$

- **Partial Steering Potential** $S_t(\mathbf{a}_{\leq t})$, which depends on the attribute one aims to optimize. It must satisfy two formal requirements: i) must not depend on future tokens, $\mathbf{a}_{>t}$; and ii) must be compatible with the steering score at end-of-sequence, *i.e.*, $S_L(\mathbf{a}_L) = S(\mathbf{a})$.

With these quantities defined, the procedure to sample the rest of the sequence consists of alternating Elongation and Refinement steps. The algorithm depends on three input parameter settings: block-size B, lookback window K, and number of MC steps N_{MC}.

Step 1: Autoregressive Elongation. First, we extend the current prefix by sampling a block of B new tokens. For each position $i \in \{t+1, \ldots, t+B\}$, draw

$$a_i \sim Q_i^{\text{AR}}(a_i|\mathbf{a}_{<i}) = \frac{P(a_i|\mathbf{a}_{<i})^\beta}{\sum_{a'_i} P(a'_i|\mathbf{a}_{<i})^\beta} \tag{4}$$

as in greedy "low-temperature" autoregressive sampling. This process is repeated B times to extend the sequence, after which the position index is updated to $t \leftarrow t + B$.

Step 2: Lookback Refinement and Steering. To correct the distribution towards the target potential defined by $\beta E_t(\mathbf{a}_{\leq t}) + \lambda S_t(\mathbf{a}_{\leq t})$, we perform N_{MC} steps of a Metropolis-Hastings refinement on the current sequence segment. In each refinement step:

1. **Selection**: A pivot position j is chosen at random from the lookback window $\{t-K+1, \ldots, t\}$.
2. **Resampling**: A candidate variant $\mathbf{a}'_{\leq t}$ is generated. The prefix $\mathbf{a}'_{<j}$ remains identical to the current sequence $\mathbf{a}_{<j}$. The suffix from j to t is resampled autoregressively using the proposal distribution q defined in Eq. (4), until the length t is reached.
3. **Acceptance**: The candidate $\mathbf{a}'_{\leq t}$ is accepted with a MH probability:

$$\min\left(1, \frac{e^{-\beta E_t(\mathbf{a}'_{\leq t}) - \lambda S_t(\mathbf{a}'_{\leq t})}}{e^{-\beta E_t(\mathbf{a}_{\leq t}) - \lambda S_t(\mathbf{a}_{\leq t})}} \frac{Q_t^{\text{AR}}(\mathbf{a}_{\leq t})}{Q_t^{\text{AR}}(\mathbf{a}'_{\leq t})}\right) \tag{5}$$

The proposal distribution $Q_t^{\text{AR}}(\mathbf{a}_{\leq t})$ is the product of the autoregressive steps used to generate the sequence up to t:

$$Q_t^{\text{AR}}(\mathbf{a}_{\leq t}) = \prod_{i=1}^{t} Q_i^{\text{AR}}(a_i|\mathbf{a}_{<i}) \tag{6}$$

Note that in the ratio $Q_t^{\text{AR}}(\mathbf{a}_{\leq t})/Q_t^{\text{AR}}(\mathbf{a}'_{\leq t})$, terms $i < j$ cancel because of the shared prefix up to j.

This procedure repeats until the end-of-sequence token ($\langle$EOS$\rangle$) is reached. While the complete sequence energy $E(\mathbf{a})$ and steering potential $S(\mathbf{a})$ are only fully realized at the final iteration, the intermediate $S_t(\mathbf{a}_{\leq t})$ and $E_t(\mathbf{a}_{\leq t})$ steer the chain towards high-probability regions early in the generation process.

When $K = 1$, ILMC mathematically reduces to greedy autoregressive low-temperature sampling. In the idealized limit $N_{\mathrm{MC}} \to \infty$ and full lookback $K = L$, the procedure converges to the target distribution $Q^*(\mathbf{a})$.

In practice, a restricted lookback window $K < L$ and moderate N_{MC} are sufficient to reach high-quality samples in realistic pLMs, as we show in Sec. 4. Moreover, a smaller K results in higher acceptance rate of the proposals made by Q^{AR} (4), which lacks steering. If K were large, the proposals sample long segments without guidance from $S_t(\mathbf{a}_{\leq t})$. The probability that such non-steered segments have good steering scores can be low. Consequently, we observe a decrease in acceptance probability as a function of K. See Appendix L for more discussion of this trade-off and Appendix P for guidelines on choosing these hyperparameters.

Effective steering potentials. ILMC depends critically on the choice of the steering potential $S_t(\mathbf{a}_{\leq t})$. While any sequence-level score can in principle define a target distribution Q^*, the autoregressive nature of our sampler requires that the corresponding partial potential provides a meaningful signal early in the generation process.

Importantly, a full-sequence steering objective $S(\mathbf{a})$ does not uniquely determine its prefix decomposition $S_t(\mathbf{a}_{\leq t})$. In addition to choosing the objective itself, one must choose how to distribute it along the generation process. Effective decompositions make the dependence on the current token as explicit as possible while minimizing residual dependence on earlier tokens. As discussed in Appendix F, strong residual dependence on the past leads to a larger mismatch between the practical sampler and the ideal target Q^*, although larger lookback windows K can partially compensate for this effect.

In this work, we consider two kinds of steering signal constructs. First, we use objectives with natural prefix decompositions: an additive composition-based score for thermostability and a prefix-compatible distance function for wild-type proximity. Second, for the more challenging non-decomposable objective of homodimerization structural class, we show that ILMC can be naturally extended to classifier-guided steering. In both settings, the steering signal biases sampling toward the desired region of sequence space, while the base pLM preserves generative quality and suppresses unrealistic or non-functional sequences.

Additional examples, mathematical details, and a broader discussion of effective and ineffective steering potentials are provided in Appendices E and F.

Computational complexity and scalability. ILMC sampling introduces a refinement phase consisting of N_{MC} MH steps after each elongation block of size B. As derived in Appendix G, the total computational cost scales approximately as $\frac{N_{\mathrm{MC}}K}{2B}L^2$, assuming that the main computational bottleneck is the evaluation of the generative score $E_t(\mathbf{a}_{\leq t})$ rather than the steering score $S_t(\mathbf{a}_{\leq t})$, whose cost depends on its specific form and is not the limiting factor in the settings studied here. This relation highlights the dependence on the lookback window K. When the lookback window is non-extensive (fixed K, independent of L), ILMC remains in the same asymptotic complexity class as standard autoregressive sampling with key-value caching, namely $O(L^2)$, albeit with a larger constant factor. By contrast, when the lookback window is extensive ($K \propto L$), the complexity scales as $O(L^3)$.

4. Experiments

To validate ILMC, we consider three protein families: **Chorismate Mutase (CM)**, **Phage Lysozyme (PL)**, and **Response Regulator (RR)**. CM and PL are well-established benchmarks where generative models have successfully produced biologically functional artificial variants (Russ et al., 2020; Madani et al., 2023). RR is used to evaluate classifier-guided steering on a harder structural objective, and connects our setting to recent pLM steering literature (Caredda et al., 2026). We evaluate ILMC across two architectures of different scales: 1) **Small Decoder-Only (sDO)**, a compact autoregressive Transformer with approximately 0.8M parameters, trained from scratch on the natural sequences of each family; 2) **ProGen3**, a 112M-parameter version of ProGen3 (Madani et al., 2023; Bhatnagar et al., 2025), the large-scale pLM pre-trained on UniProt (The UniProt Consortium, 2025) and subsequently fine-tuned on the specific family datasets. See Appendix H for details on the architectures, training data, optimization protocols, and model hyperparameters.

4.1. Generating Diverse High-Probability Samples

We first investigate the case where no steering potential is applied, i.e., $S(\mathbf{a}) = 0$. From Proposition 2.1, sampling from the exponentiated distribution $Q^*(\mathbf{a}) \propto e^{-\beta E(\mathbf{a})}$ is the unique strategy that maximizes entropy for a fixed expected generative quality C_{E}. In an Energy-Entropy plane, this defines a Pareto efficiency frontier; for any given mean energy $\mathbb{E}_Q[E]$, no sampling strategy can achieve higher diversity than Q^*. Conversely, Q^* reaches the lowest possible average energy for a fixed entropy budget.

Empirically, the lookback window K is a critical determi-

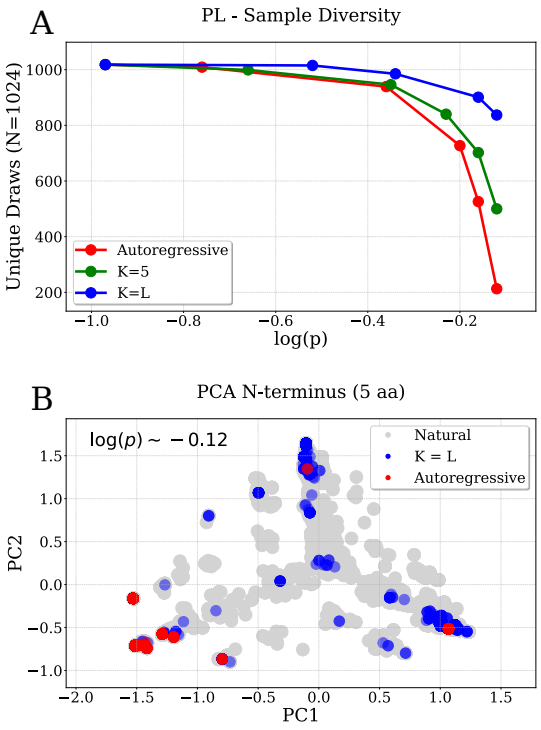

Figure 2. **Diversity vs. generative quality** (sDO on PL, 1024 sequences per sample). (A) Unique sequences ($N = 1024$) vs. $\log P$. Standard autoregressive sampling (red) suffers from rapid diversity collapse, while ILMC ($K = 5$ green, $K = L$ blue; $B = 2, N_{\mathrm{MC}} = 10$) draws more unique variants. (B) PCA of N-termini (first 5 residues) at high probability ($\log P \approx -0.12$). Standard sampling (red) collapses into narrow modes, while global lookback ($K = L$, blue) covers the natural sequence space (grey).

nant of the quality-diversity trade-off. To achieve a specific target mean energy, standard autoregressive sampling ($K = 1$) requires a significantly higher inverse temperature β than ILMC with $K > 1$, and the required β decreases as K increases (see Table 3). This high local sharpening in the autoregressive case collapses the distribution's support, leading to a marked reduction in diversity. Using the number of unique draws over a sample of 1024 sequences as a proxy for entropy, we find that increasing K lets ILMC reach high-probability sequences while maintaining a significantly higher count of unique variants, as illustrated in Figure 2 (sDO model, PL family). While a higher K intuitively provides a better approximation of the target Q^*, and thus higher entropy per Proposition 2.2, this is not always mathematically guaranteed, as it is possible to find pathological cases that defy this intuition; see Appendix D. Furthermore, because we set a fixed number of MH steps ($N_{\mathrm{MC}} = 10$) and do not reach strict equilibrium, even when $K = L$ the sampling only approximates Q^*.

Various library diversity metrics across sDO and ProGen3 models and the two families (PL and CM), using sampling parameters $B = 2$, $N_{\mathrm{MC}} = 10$ and varying K and β, are reported in Table 3. Specifically, we track the total number of unique draws (out of 1024), the number of unique draws within an edit distance radius of 5 (meaning that once a sequence is selected, all subsequent draws with a distance ≤ 5 are considered duplicates), the mean pairwise distance within the library (Dist Gen-Gen), and the mean edit distance to the closest neighbor (Min Dist Gen-Gen). These metrics generally improve as K increases, in agreement with Proposition 2.2. The results consistently show increased diversity for both sDO and ProGen3 across both families, indicating that the method scales effectively. In contrast, autoregressive samples exhibit a pronounced mode collapse at high $\log P$, evident in the histogram of pairwise distances (Dist Gen-Gen), shown in Appendix Fig. 6.

This high-$\log P$ regime, where ILMC yields its largest diversity gains, is particularly relevant in practical protein design. In downstream experimental pipelines, practitioners often prioritize the highest-scoring sequences through highly selective strategies such as greedy decoding ($\beta \to \infty$ limit) (Faizi et al., 2025) or post-hoc filtering (Lambert et al., 2025), because high-likelihood proteins are more likely to remain functional in wet-lab validation (Russ et al., 2020; Fields et al., 2025).

Visual inspection of the sampled libraries reveals that increasing the lookback window K leads to a significantly wider distribution of sequence variety at the N-terminus in the high $\log P$ samples. We show this in Figure 2 for the sDO model applied to the PL family, which presents a Principal Component Analysis (PCA) projection of the first five amino acids, comparing standard autoregressive sampling ($K = 1$) with the global lookback case ($K = L$). Additional examples are provided in Appendix Fig. 7. This phenomenon can be intuitively attributed to the inverse temperature β. In the $K = 1$ regime, achieving a given low-energy target requires very high β (see Table 3), which effectively collapses the initial generation to few sequences as the local logit profile becomes very sharp. Diversity is only introduced when a lower-probability token, or "stochastic error," is sampled, but as β increases, the waiting time for this "error" event grows, causing a collapse in the diversity of the N-terminus. In contrast, ILMC allows the model to explore a broader set of initial states, that are later refined into high-quality sequences. This also suggests that the high-probability samples at higher K settings better cover the sequence space around the natural sequences. To quantify this, Table 3 reports the mean distance of natural sequences to a closest sequence of the generated library (Min Dist Gen-Nat). Lower values indicate that the generated samples better explore the region of sequence space occupied by natural proteins. This metric tends to decrease with K at a given generative quality (mean value of $\log P$), confirming that the samples explore a broader region of the sequence space. A comparison with top-p

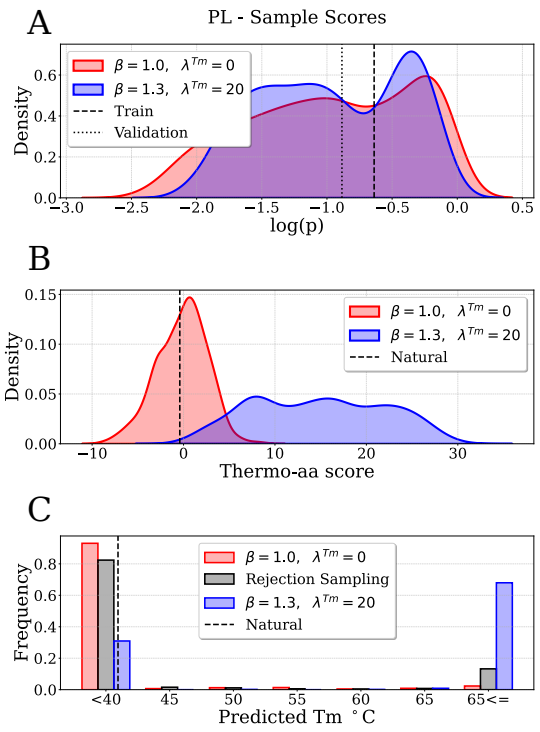

Figure 3. **Steering generation toward thermophilic adaptation** (sDO on PL, 1024 sequences per sample). (A) Distribution of $\log P$ comparing the unsteered baseline (red) with the steered ILMC sampling strategy (blue, $\lambda^{\mathrm{Tm}} = 20, \beta = 1.3, B = 2, K = 5, N_{\mathrm{MC}} = 10$). The parameters of the ILMC were explicitly tuned to match the $\log P$ of the baseline. (B) Distribution of the thermophilic amino acid score $-S^{\mathrm{Tm}}(\mathbf{a})$. The steered samples exhibit a significant shift toward higher scores, far exceeding the range observed in the unsteered baseline and the natural sequences (dashed line). (C) Predicted melting temperatures (T_m) estimated using TemStaPro.

sample diversity is provided in the Table 4.

4.2. Steering generation toward enhanced thermostability

Comparative proteomic analyses have shown that amino acid composition is a primary determinant of thermal stability. Specifically, thermophilic organisms exhibit a distinct enrichment of a subset of residues, often identified by the mnemonic IVYWREL, which facilitate the formation of robust salt bridges, improved hydrophobic packing, and increased hydrogen bonding (Fukuchi & Nishikawa, 2001; Zeldovich et al., 2007). In this experiment, we define a steering potential $S^{\mathrm{Tm}}(\mathbf{a})$ to bias generation toward these stabilizing amino-acids. Following established findings that surface residues are less constrained by evolutionary conservation and more responsive to environmental adaptation, we use propensity scores derived specifically from surface amino acid compositions (Fukuchi & Nishikawa, 2001).

For an amino acid $a \in \mathcal{A}$, let $w(a)$ denote the ratio of its frequency in thermophilic surfaces to its frequency in mesophilic surfaces. The table of propensities $w(a)$ is provided in Appendix J. We define the global steering potential as the negative sum of these ratios across the sequence:

$$S^{\mathrm{Tm}}(\mathbf{a}) = -\sum_{i=1}^{L} w(a_i) \tag{7}$$

Optimizing this score in isolation would not yield meaningful protein designs: one could trivially increase it by enriching sequences in favorable residues without preserving global plausibility or function. ILMC avoids this failure mode by balancing the steering score against the generative energy $E(\mathbf{a})$. Steering therefore operates only within regions of sequence space that remain plausible under the pLM. In this sense, ILMC can turn simple property-correlated proxy scores into practically useful design objectives.

The partial steering potential for an incomplete prefix $\mathbf{a}_{\leq t}$ is naturally defined as the cumulative sum over the existing residues:

$$S_t^{\mathrm{Tm}}(\mathbf{a}_{\leq t}) = -\sum_{i=1}^{t} w(a_i) \tag{8}$$

This additive formulation ensures that the criteria for effective steering discussed in Section 3 are satisfied, providing a steering signal from the initiation of the sequence.
The global target distribution is then (*c.f.* Eq. (1)):

$$Q^*(\mathbf{a}) \propto \exp(-\beta E(\mathbf{a}) - \lambda^{\mathrm{Tm}} S^{\mathrm{Tm}}(\mathbf{a})) \tag{9}$$

where λ^{Tm} is the Lagrange multiplier associated with the steering potential $S^{\mathrm{Tm}}(\mathbf{a})$. Interestingly, we observe that increasing λ^{Tm} at constant β, decreases the average S^{Tm} but *increases* the average generative E. To maintain a target generative quality, we must also increase β. This indicates an inherent trade-off between the generative energy and thermostability for the designed sequences. These results are reported in Table 1. Further analysis regarding the Pareto frontier between generative energy and the steering score are provided in Appendix M.

To assess whether this sequence-level steering translates into thermophilic adaptation, we predicted melting temperatures (T_m) of generated sequences using TemStaPro (Pudžiuvelytė et al., 2024) (implementation details in Appendix I). At matched generative quality, ILMC increases the predicted T_m by an average of 18.3°C relative to the standard autoregressive baseline and 7.1°C against a runtime-matched rejection-sampling obtained by generating standard autoregressive samples with the same total compute budget and filtering them by $S^{\mathrm{Tm}}(\mathbf{a})$. To verify that steering does not degrade structural plausibility, we also computed pLDDT scores with ESMFold (Lin

Table 1. **Performance of Thermostability Steering.** We compare standard autoregressive sampling (Base: $\lambda^{\mathrm{Tm}} = 0$), a compute-matched rejection-sampling baseline (Rejection), and ILMC (Steered: $\lambda^{\mathrm{Tm}} = 20, B = 2, K = 5, N_{\mathrm{MC}} = 10$) across two architectures (sDO, ProGen3) and protein families (CM, PL). Statistics are computed over 1024 sequences per sample. Metrics include: $\log P$ (mean per-token $\log P$); pLDDT (predicted local structural confidence via ESMFold); T_m (predicted melting temperature via TemStaPro); Dist Gen-Gen (mean pairwise edit distance within the library); Min Dist Gen-Gen (mean edit distance to the nearest generated neighbor); and Min Dist Gen-Nat (mean edit distance to the nearest natural sequence).

Family	Model	Strategy	β	$\log P$	pLDDT	Tm (°C)	Length	Dist (Gen-Gen)	Min Dist (Gen-Gen)	Min Dist (Gen-Nat)
CM	sDO	Base	1.00	-0.80 ± 0.46	78 ± 4	41.1 ± 6.5	89.8 ± 1.6	64.5 ± 9.8	29.3 ± 16.9	25.7 ± 16.8
		Rejection	1.50	-0.82 ± 0.28	78 ± 4	56.1 ± 11.0	89.4 ± 1.7	59.8 ± 8.5	30.6 ± 12.1	33.0 ± 13.4
		Steered	1.80	-0.78 ± 0.31	78 ± 4	**61.5 ± 10.8**	90.0 ± 1.8	53.1 ± 14.0	20.8 ± 12.4	35.0 ± 17.1
	ProGen3	Base	1.10	-0.62 ± 0.45	79 ± 3	40.8 ± 6.4	88.6 ± 5.8	63.8 ± 9.9	25.9 ± 15.0	22.8 ± 15.8
		Rejection	1.80	-0.63 ± 0.27	79 ± 2	55.6 ± 11.6	89.6 ± 2.2	60.0 ± 10.0	27.4 ± 14.1	30.2 ± 14.7
		Steered	2.00	-0.60 ± 0.25	79 ± 2	**60.8 ± 11.0**	89.3 ± 1.8	56.4 ± 10.2	21.5 ± 11.5	29.7 ± 14.9
PL	sDO	Base	1.00	-0.97 ± 0.63	60 ± 11	39.2 ± 5.6	128.1 ± 5.9	91.2 ± 13.0	48.7 ± 28.6	45.4 ± 28.7
		Rejection	1.10	-1.06 ± 0.63	57 ± 10	45.5 ± 11.3	127.6 ± 6.7	92.5 ± 13.9	52.5 ± 28.4	51.5 ± 28.1
		Steered	1.30	-0.96 ± 0.54	59 ± 11	**57.3 ± 13.7**	132.4 ± 5.3	89.4 ± 16.9	45.0 ± 27.1	51.7 ± 29.5
	ProGen3	Base	1.20	-0.53 ± 0.40	67 ± 7	39.7 ± 6.6	128.2 ± 6.9	85.6 ± 15.0	36.1 ± 26.0	34.0 ± 25.7
		Rejection	1.60	-0.53 ± 0.30	64 ± 8	48.4 ± 12.8	128.1 ± 7.3	81.1 ± 17.4	33.9 ± 22.7	37.2 ± 23.3
		Steered	1.50	-0.52 ± 0.32	67 ± 7	**54.4 ± 13.9**	130.6 ± 13.4	85.5 ± 24.4	30.8 ± 24.0	38.7 ± 30.2

et al., 2023), finding that the optimization of $S(\mathbf{a})$ does not measurably reduce predicted structural confidence.

Results for the sDO model on the PL family are illustrated in Figure 3, and quantitative results for both sDO and ProGen3 are reported in Table 1. Sampling parameters were set to $B = 2$, $K = 5$, and $N_{\mathrm{MC}} = 10$, with varying β and λ^{Tm}. ILMC thus steers generation toward thermostable regions of sequence space that are inaccessible to standard sampling while avoiding diversity collapse, as shown by the library-distance metrics reported in Table 1 (Dist Gen-Gen, Min Dist Gen-Gen, and Min Dist Gen-Nat).

4.3. Steering generation towards the vicinity of a target wild-type

Many applications in biological sequence engineering require mutating an existing wild-type (WT) sequence rather than generating *de novo* proteins (Calvanese et al., 2025; Lambert et al., 2025; Hie et al., 2024; Blalock et al., 2025). Identifying low-energy states in the local vicinity of natural variants is of significant biological and engineering relevance. In this section, we show that by introducing an appropriate steering potential, we are able to bias the presented sampling process toward the mutational neighborhood of a specific target.

While the edit distance is a natural metric for string divergence, it turns out to be a poor steering potential in terms of the criteria discussed in Section 3. For example, the raw edit distance between a low-length prefix and a full-length target is mostly determined by the length difference and is almost independent of the prefix compatibility with the beginning of the target. To address this, we define a modified edit distance. Let $\mathbf{a}_{\leq t}$ be the sequence prefix generated so far, and $\mathbf{w}$ the target wild-type of length L_w. We define $S^{\mathrm{WT}}(\mathbf{a}_{\leq t})$ as the minimum number of edits required to transform the prefix $\mathbf{a}_{\leq t}$ into any prefix of the target $\mathbf{w}$:

$$S_t^{\mathrm{WT}}(\mathbf{a}_{\leq t}) = \min_{1 \leq k \leq L_w} \mathrm{EditDistance}(\mathbf{a}_{\leq t}, \mathbf{w}_{\leq k}) \tag{10}$$

The complete target distribution for the steered sampling process is:

$$Q^*(\mathbf{a}) \propto \exp\left(-\beta E(\mathbf{a}) - \lambda^{\mathrm{WT}} S^{\mathrm{WT}}(\mathbf{a}_{\leq L})\right) \tag{11}$$

where λ^{WT} is the Lagrange multiplier tuning the distance towards $\mathbf{w}$. This formulation ensures that $S_t^{\mathrm{WT}}(\mathbf{a}_{\leq t})$ provides a meaningful steering signal from the beginning of the generated sequence. Details on the implementation of this distance, along with illustrative examples and more discussion justifying its steering effectiveness, are provided in Appendix K .

Our results demonstrate that by tuning the steering strength λ^{WT}, we can sample variants closer to the target wild-type (see Figure 4A). The generative energies of these proximal designs are distributed around the energy of the original wild-type and are significantly lower than those of random mutations at the same distance, as shown in Figure 4B (illustrated for sDO on PL). Table 5 reports the average $\log P$ and edit distance to the target wild-type of steered samples for both protein families and model architectures. The chosen target wild-type was the first in the provided training dataset (index 0). Sampling parameters were set to $B = 2$, $K = 5$, and $N_{\mathrm{MC}} = 300$. Runtime-matched rejection sampling systematically fails to reach the desired mutational neighborhood: as seen in Figure 4A, the tail of the standard autoregressive distribution remains far from the target WT, so sampling close variants by post-hoc filtering would require prohibitively large sample sizes.

Another interesting application of this framework is multi-objective design, achieved by introducing two steering potentials simultaneously. For example, we could aim to enhance thermostability (T_m) starting from a given wild-type

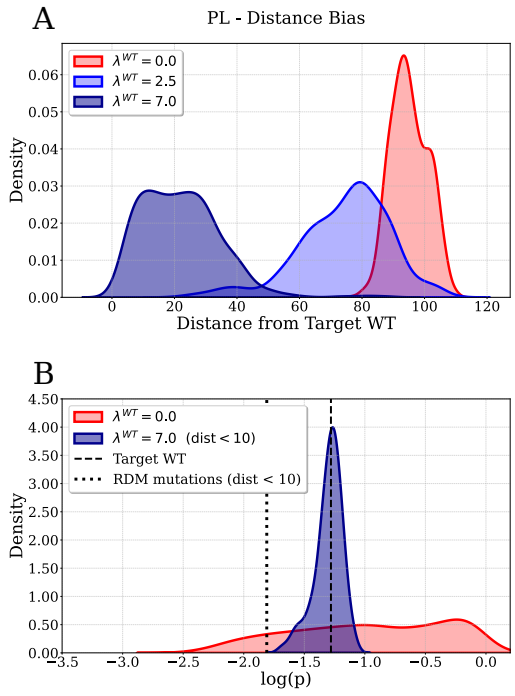

Figure 4. **Steering toward a target wild-type** (sDO on PL, 1024 sequences per sample). (A) Edit distance distributions for increasing steering strength λ^{WT} ($B = 2, K = 5, N_{\mathrm{MC}} = 300$). Higher λ^{WT} concentrates sampling in the immediate local neighborhood (dark blue). (B) $\log P$ of variants within 10 mutations. Steered samples cluster around the WT score (dashed) and significantly outperform random mutations (dotted) compared to the unsteered baseline (red).

sequence, while restricting the number of modifications on the original sequence. This task is naturally addressed with ILMC, by combining the steering potentials for thermostability and wild-type vicinity, Equations (8) and (10). ILMC will then search for thermostable variants within a small mutational radius of a wild-type target. Our results (Appendix N) confirm that ILMC is able to identify high-likelihood sequences in the close vicinity of the given wild-type with significantly improved predicted thermostability.

4.4. Classifier-guided steering for non-decomposable properties

The steering potentials considered so far admit natural prefix decompositions, making them particularly suitable for ILMC. However, many important protein-design objectives are not easily expressible as additive or otherwise decomposable sequence scores. Examples include global structural classes and complex functional labels that arise from long-range sequence constraints. To extend ILMC to this setting, we combine it with *classifier-guided steering*. This also enables a direct comparison with classifier-guided autoregressive decoding methods such as FUDGE (Future Discriminators for Generation) (Yang & Klein, 2021), of which ILMC can be viewed as a non-myopic extension.

FUDGE trains an any-prefix classifier, i.e., a predictor of the target label from an incomplete prefix, $p_\phi(y \mid \mathbf{a}_{\le t})$, and uses it to bias next-token generation during autoregressive decoding. Intuitively, prefixes that remain compatible with multiple classes receive only weak guidance, whereas continuations that make the target class unlikely are down-weighted immediately. Concretely, if y denotes the target label, FUDGE samples each token from

$$Q_t^{\mathrm{FUDGE}}(a_t \mid \mathbf{a}_{<t}, y) \propto P(a_t \mid \mathbf{a}_{<t})^{\beta}\, p_\phi(y \mid \mathbf{a}_{\le t})^{\lambda}, \tag{12}$$

and the probability of a full sequence factorizes as

$$Q^{\mathrm{FUDGE}}(\mathbf{a} \mid y) = \prod_{t=1}^{L} Q_t^{\mathrm{FUDGE}}(a_t \mid \mathbf{a}_{<t}, y). \tag{13}$$

As with greedy low-temperature sampling, this construction is purely local and therefore myopic.

Within ILMC, the same prefix classifier is used differently. Rather than sampling directly from the locally guided autoregressive product in Eq. (13), we use the classifier output to define a global target distribution and approximate sampling from it with lookback Monte Carlo refinement. Specifically, we define the prefix steering potential

$$S_t^{\mathrm{clf}}(\mathbf{a}_{\le t}) = -\log p_\phi(y \mid \mathbf{a}_{\le t}), \tag{14}$$

which yields the target distribution

$$Q^*(\mathbf{a}) \propto e^{-\beta E(\mathbf{a}) - \lambda^{\mathrm{clf}} S_L^{\mathrm{clf}}(\mathbf{a})} \propto P(\mathbf{a})^{\beta} p_\phi(y \mid \mathbf{a})^{\lambda^{\mathrm{clf}}}. \tag{15}$$

The Response Regulator (RR) receiver-domain family (Pfam PF00072 (Mistry et al., 2021)) consists of different subclasses that are associated with distinct homodimerization geometries (Gao & Stock, 2009). We train an autoregressive sDO model on the RR multiple-sequence alignment ($L = 111$). We then use a lightweight any-prefix

Table 2. **Classifier-guided steering on Response Regulators.** Comparison of unsteered autoregressive sampling, FUDGE, and ILMC on the RR benchmark, reported at two matched mean per-token $\log P$ operating points. Metrics are Unique (Radius 5), Dist Gen-Gen, and iRMSD (AlphaFold3) to the target structure (GerE, PDB 4ZMS). For reference, natural sequences in this structural class have an iRMSD of 11.3 ± 3.4, vs. 15.1 ± 4.4 for sequences from other structural classes.

$\log P$	Method	β	Unique (Radius 5)	Length	Dist (Gen-Gen)	iRMSD (Å)
-0.56	Base	2.0	1006	111	67.7 ± 10.9	17.3 ± 3.5
	FUDGE	1.8	1000	111	58.35 ± 13.7	9.8 ± 4.3
	ILMC	1.4	1004	111	63.9 ± 13.3	10.6 ± 6.5
-0.37	Base	3.5	694	111	63.9 ± 15.3	15.0 ± 5.0
	FUDGE	2.8	709	111	48.8 ± 16.4	9.0 ± 3.8
	ILMC	1.8	927	111	63.4 ± 15.1	10.7 ± 3.8

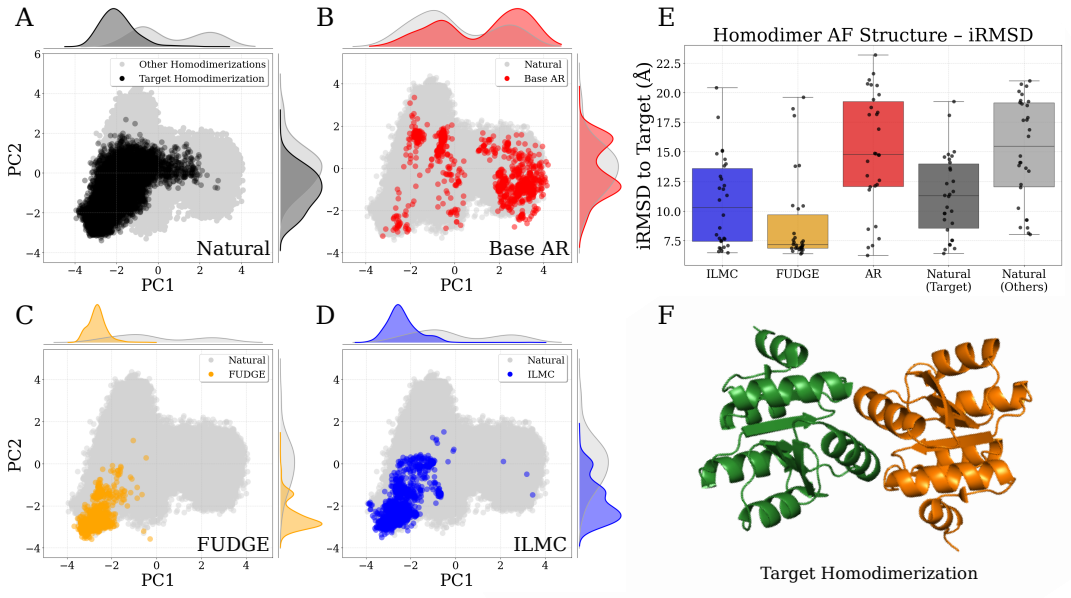

Figure 5. **Classifier-guided steering on the Response Regulator family.** All generated samples are shown at matched mean per-token $\log P = -0.37$. (A) PCA projection of natural sequences ($N = 855{,}395$), highlighting sequences with the target GerE homodimerization class in black. (B-D) 1024 samples from unsteered autoregressive, FUDGE, and ILMC. (E) iRMSD to target GerE class ($N = 30$). (F) GerE target homodimerization structure (PDB 4ZMS).

classifier (see Appendix O) to steer generation toward a specific GerE homodimerization target (Caredda et al., 2026). Since the generator and prefix classifier are identical across the FUDGE and ILMC conditions, the comparison isolates the effect of the *sampling strategy*, rather than differences in the models.

Empirically, both FUDGE and ILMC ($B = 2$, $K = 10$, $N_{MC} = 10$) enrich for the target RR structural class, as assessed by AlphaFold3 (Abramson et al., 2024) based validation against the reference dimerization geometry (Table 2; Fig. 5E). At moderate sharpening ($\log P = -0.56$), the two methods produce a similarly large number of unique sequences within edit-distance radius 5 (1000 for FUDGE and 1004 for ILMC), but ILMC already preserves higher intra-library diversity, with mean pairwise edit distance 63.9 ± 13.3 compared with 58.35 ± 13.7 for FUDGE (Table 2). Under stronger sharpening ($\log P = -0.37$), the diversity advantage increases: ILMC produces 927 unique sequences within radius 5, versus 709 for FUDGE, and maintains substantially broader intra-library diversity (63.4 ± 15.1 versus 48.8 ± 16.4; Table 2). This broader exploration is also visible in sequence space (Fig. 5C and D). Both methods strongly improve over the unsteered baseline in terms of iRMSD, but ILMC more closely spans the iRMSD range observed for natural sequences from the correct structural class (Table 2; Fig. 5E), with values consistent with the existing literature (Caredda et al., 2026).

Increasing the lookback window yields diminishing diversity gains. For example, $K = 20$ yields modest improvements (Unique within radius 5: 914; mean intra-library distance: 65.8 ± 14.4).

These results show that ILMC is not restricted to simple decomposable steering scores. When no such proxy is available, a lightweight prefix classifier can be used as a modular steering potential, extending ILMC to non-decomposable properties while retaining a clear advantage over purely autoregressive guidance in library diversity.

5. Concluding remarks

ILMC is a *drop-in* sampler for steering pLM generation, mitigating the diversity collapse often seen with probability sharpening and reward-based decoding. By targeting a maximum-entropy distribution that balances generative likelihood with steering constraints, it produces diverse, high-quality libraries. ILMC can turn simple property-correlated proxies into effective design objectives, and can also be applied to classifier-guided steering, acting as a non-myopic correction to purely autoregressive methods such as FUDGE. From family-specific models to the state-of-the-art ProGen3, ILMC generates diverse, high-likelihood sequences and supports practical steering signals such as thermostability, wild-type proximity, and homodimerization structural class. ILMC can be broadly useful for protein engineering, with natural extensions to richer multi-objective steering and experimental validation.

Impact statement. This paper presents work whose goal is to advance the field of machine learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Abramson, J., Adler, J., Dunger, J., Evans, R., Green, T., Pritzel, A., Ronneberger, O., Willmore, L., Ballard, A. J., Bambrick, J., Bodenstein, S. W., Evans, D. A., Hung, C.-W., O'Neill, M., Reiman, D., Tunyasuvunakool, K., Wu, Z., Žemgulytė, A., Arvaniti, E., Beattie, C., Bertolli, O., Bridgland, A., Cherepanov, A., Congreve, M., Cowen-Rivers, A. I., Cowie, A., Figurnov, M., Fuchs, F. B., Gladman, H., Jain, R., Khan, Y. A., Low, C. M. R., Perlin, K., Potapenko, A., Savy, P., Singh, S., Stecula, A., Thillaisundaram, A., Tong, C., Yakneen, S., Zhong, E. D., Zielinski, M., Žídek, A., Bapst, V., Kohli, P., Jaderberg, M., Hassabis, D., and Jumper, J. M. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630(8016):493–500, jun 2024. doi: 10.1038/s41586-024-07487-w.

Bhatnagar, A., Jain, S., Beazer, J., Curran, S. C., Hoffnagle, A. M., Ching, K., Martyn, M., Nayfach, S., Ruffolo, J. A., and Madani, A. Scaling unlocks broader generation and deeper functional understanding of proteins. *bioRxiv*, 2025.

Blalock, N., Seshadri, S., Babbar, A., Fahlberg, S. A., Kulkarni, A., and Romero, P. A. Functional alignment of protein language models via reinforcement learning. *bioRxiv*, 2025.

Calvanese, F., Peinetti, G., Pavlinova, P., Nghe, P., and Weigt, M. Integrating experimental feedback improves generative models for biological sequences. *Nucleic Acids Research*, 53(16):gkaf832, 2025.

Caredda, F., Gennai, L., De Los Rios, P., and Pagnani, A. Controllable protein design via autoregressive direct coupling analysis conditioned on principal components. *PLOS Computational Biology*, 22(2):e1013996, feb 2026.

Elnaggar, A., Heinzinger, M., Dallago, C., Rhabert, G., Wang, Y., Wan, L., and Rost, B. Prottrans: Toward understanding the language of life through self-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):7112–7127, 2022.

Faizi, M., Sakharova, H., and Lareau, L. F. A generative language model decodes contextual constraints on codon choice for mrna design. *bioRxiv*, may 2025. doi: 10.1101/2025.05.13.653614.

Fernandez-de Cossio-Diaz, J., Hardouin, P., Lyonnet du Moutier, F.-X., Di Gioacchino, A., Marchand, B., Ponty, Y., Sargueil, B., Monasson, R., and Cocco, S. Designing molecular rna switches with restricted boltzmann machines. *Nature Communications*, 16(1):11223, 2025.

Fields, P. W., Ngampruetikorn, V., Schwab, D. J., and Palmer, S. E. Understanding temperature tuning in energy-based models. *arXiv preprint arXiv:2512.09152*, 2025.

Fukuchi, S. and Nishikawa, K. Protein surface amino acid compositions distinctively differ between thermophilic and mesophilic bacteria. *Journal of Molecular Biology*, 309(4):835–843, 2001.

Gao, R. and Stock, A. M. Biological insights from structures of two-component proteins. *Annual Review of Microbiology*, 63:133–154, 2009. doi: 10.1146/annurev.micro.091208.073214.

He, A., Fried, D., and Welleck, S. Rewarding the unlikely: Lifting grpo beyond distribution sharpening. *arXiv preprint arXiv:2506.02355*, 2025.

Hie, B. L., Shanker, V. R., Xu, D., Bruun, T. U. J., Weidenbacher, P. A., Tang, S., Wu, W., Pak, J. E., and Kim, P. S. Efficient evolution of human antibodies from general protein language models. *Nature Biotechnology*, 42(2):275–283, 2024.

Hill, M. O. Diversity and evenness: a unifying notation and its consequences. *Ecology*, 54(2):427–432, 1973.

Holtzman, A., Buys, J., Du, L., Forbes, M., and Choi, Y. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*, 2019.

Karan, A. and Du, Y. Reasoning with sampling: Your base model is smarter than you think. *arXiv preprint arXiv:2510.14901*, 2025.

Kim, K., Lee, S., and Lee, S. Koconovel: Annotated dataset of character coreference in korean novels. *arXiv preprint arXiv:2404.01140*, 2024.

Lambert, C. N., Opuu, V., Calvanese, F., Pavlinova, P., Zamponi, F., Hayden, E. J., Weigt, M., Smerlak, M., and Nghe, P. Exploring the space of self-reproducing ribozymes using generative models. *Nature communications*, 16(1):7836, 2025.

Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., Verkuil, R., Kabeli, O., Shmueli, Y., et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

Madani, A., Krause, B., Greene, E. R., Subramanian, S., Mohr, B. P., Holton, J. M., Olmos Jr, J. L., Xiong, C., Sun, Z. Z., Socher, R., et al. Large language models generate functional protein sequences across diverse families. *Nature biotechnology*, 41(8):1099–1106, 2023.

Mistry, J., Chuguransky, S., Williams, L., Qureshi, M., Salazar, G. A., Sonnhammer, E. L. L., Tosatto, S. C. E.,

Paladin, L., Raj, S., Richardson, L. J., et al. Pfam: The protein families database in 2021. *Nucleic Acids Research*, 49(D1):D412–D419, 2021.

Nawrocki, E. P. and Eddy, S. R. Infernal 1.1: 100-fold faster rna homology searches. *Bioinformatics*, 29(22): 2933–2935, 2013.

Nijkamp, E., Ruffolo, J. A., Weinstein, E. N., Naik, N., and Madani, A. Progen2: exploring the boundaries of protein language models. *Cell systems*, 14(11):968–978, 2023.

Pudžiuvelytė, I., Olechnovič, K., Godliauskaitė, E., Sermokas, K., Urbaitis, T., Gasiūnas, G., and Kazlauskas, D. Temstapro: protein thermostability prediction using sequence representations from protein language models. *Bioinformatics*, 40(4):btae157, 2024.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. *OpenAI*, 2019. Accessed: 2024-11-15.

Russ, W. P., Figliuzzi, M., Stocker, C., Barrat-Charlaix, P., Socolich, M., Kast, P., Hilvert, D., Monasson, R., Cocco, S., Weigt, M., et al. An evolution-based model for designing chorismate mutase enzymes. *Science*, 369 (6502):440–445, 2020.

Song, Y., Kempe, J., and Munos, R. Outcome-based exploration for llm reasoning. In *NeurIPS 2025 Workshop: Second Workshop on Aligning Reinforcement Learning Experimentalists and Theorists*, 2025.

Stocco, F., Garibbo, M., and Ferruz, N. Guiding generative models for protein design: Prompting, steering and aligning. *arXiv preprint arXiv:2511.21476*, 2025.

The UniProt Consortium. Uniprot: the universal protein knowledgebase in 2025. *Nucleic Acids Research*, 53(D1): D609–D617, 2025.

Yang, K. and Klein, D. Fudge: Controlled text generation with future discriminators. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3511–3535, Online, jun 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.276.

Zeldovich, K. B., Berezovsky, I. N., and Shakhnovich, E. I. Protein and dna sequence determinants of thermophilic adaptation. *PLoS Computational Biology*, 3(1):e5, 2007.

A. Supplementary results

Table 3. **Diversity metrics for unsteered sampling strategies.** We compare standard autoregressive sampling (Base, $K = 1$) with ILMC ($B = 2, N_{\mathrm{MC}} = 10, K = 5, L$). All sample sets consist of 1024 sequences. Reference metrics for Natural Datasets: for CM, mean pairwise dist. 65.2 ± 9.1, min dist. 11 ± 12; for PL, mean pairwise dist. 91.3 ± 12.2, min dist. 17.4 ± 20.0.

Family	Model	Strategy	β	$\log P$	Unique (on 1024)	Unique (radius 5)	Length	Dist (Gen-Gen)	Min Dist (Gen-Gen)	Min Dist (Gen-Nat)	Min Dist (Nat-Gen)
CM	ProGen3	Base	1.30	-0.49 ± 0.29	1021	963	90.0 ± 1.50	63.7 ± 10.8	22.1 ± 14.8	19.1 ± 14.5	26.9 ± 16.2
		$K = 5$	1.20	-0.48 ± 0.29	1022	964	89.4 ± 2.0	63.4 ± 11.0	21.2 ± 14.5	18.2 ± 14.4	26.9 ± 16.2
		$K = L$	1.04	-0.47 ± 0.27	1022	973	88.2 ± 1.7	60.7 ± 12.8	18.9 ± 13.5	16.2 ± 12.8	28.7 ± 16.3
		Base	2.00	-0.31 ± 0.19	961	793	90.0 ± 1.5	63.1 ± 12.9	13.1 ± 13.0	12.2 ± 11.8	26.6 ± 16.2
		$K = 5$	1.50	-0.30 ± 0.18	982	808	89.8 ± 1.7	63.2 ± 12.0	12.7 ± 12.2	10.8 ± 11.3	26.0 ± 16.7
		$K = L$	1.08	-0.31 ± 0.16	1009	862	88.8 ± 1.4	58.2 ± 15.7	11.2 ± 9.6	9.0 ± 7.8	28.6 ± 17.7
		Base	8.00	-0.20 ± 0.07	205	93	90.3 ± 1.5	49.6 ± 27.6	1.2 ± 5.7	5.6 ± 6.1	38.5 ± 15.1
		$K = 5$	1.80	-0.20 ± 0.12	905	643	90.0 ± 1.7	62.7 ± 13.6	8.3 ± 10.7	7.0 ± 8.5	26.5 ± 17.0
		$K = L$	1.20	-0.19 ± 0.07	935	523	89.5 ± 1.4	58.0 ± 17.2	4.9 ± 6.1	3.5 ± 3.1	29.3 ± 18.8
CM	sDO	Base	1.30	-0.52 ± 0.34	1018	937	90.2 ± 1.5	63.5 ± 11.5	20.3 ± 15.1	17.9 ± 14.7	26.2 ± 15.8
		$K = 5$	1.15	-0.53 ± 0.35	1020	953	89.9 ± 1.5	63.4 ± 11.5	20.4 ± 14.9	17.4 ± 14.2	26.6 ± 16.0
		$K = L$	1.02	-0.52 ± 0.35	1018	951	90.2 ± 1.6	63.4 ± 12.8	18.4 ± 14.1	15.6 ± 13.2	28.0 ± 16.7
		Base	2.86	-0.23 ± 0.15	757	423	90.0 ± 1.7	58.1 ± 20.5	5.7 ± 9.8	6.8 ± 7.9	28.0 ± 16.6
		$K = 5$	1.55	-0.24 ± 0.16	932	609	90.0 ± 1.5	60.6 ± 16.2	7.5 ± 10.2	6.4 ± 7.9	26.6 ± 17.0
		$K = L$	1.10	-0.23 ± 0.11	988	653	90.2 ± 1.6	59.5 ± 16.8	6.3 ± 7.0	4.7 ± 4.3	28.5 ± 18.3
		Base	6.50	-0.17 ± 0.07	154	82	90.2 ± 2.0	36.7 ± 32.9	0.9 ± 4.2	3.8 ± 3.9	37.3 ± 16.6
		$K = 5$	2.00	-0.16 ± 0.07	740	296	90.1 ± 1.5	55.2 ± 21.2	3.1 ± 6.9	3.3 ± 3.8	29.8 ± 18.0
		$K = L$	1.20	-0.17 ± 0.06	847	396	90.5 ± 1.6	58.8 ± 18.2	3.4 ± 5.8	2.6 ± 2.5	31.1 ± 19.2
PL	ProGen3	Base	1.18	-0.55 ± 0.40	1008	910	128.0 ± 7.5	86.4 ± 15.2	37.4 ± 26.7	34.2 ± 26.6	42.3 ± 25.9
		$K = 5$	1.10	-0.56 ± 0.40	1010	940	126.0 ± 7.2	85.8 ± 13.9	36.2 ± 24.6	33.3 ± 25.0	42.2 ± 25.9
		$K = L$	1.02	-0.56 ± 0.41	1015	898	115.9 ± 7.5	84.6 ± 13.0	32.1 ± 24.6	30.7 ± 24.7	48.5 ± 24.4
		Base	1.50	-0.32 ± 0.25	947	787	128.2 ± 5.5	80.4 ± 16.9	22.4 ± 21.2	21.2 ± 20.6	41.4 ± 26.2
		$K = 5$	1.30	-0.32 ± 0.25	975	788	127.2 ± 6.2	82.8 ± 15.9	22.1 ± 21.3	20.2 ± 20.6	40.7 ± 26.5
		$K = L$	1.04	-0.32 ± 0.24	997	800	117.7 ± 8.3	82.6 ± 16.5	18.3 ± 19.0	17.4 ± 18.4	46.4 ± 25.4
		Base	8.00	-0.12 ± 0.04	62	27	126.4 ± 3.6	32.6 ± 25.8	0.4 ± 4.3	4.0 ± 3.8	68.0 ± 20.5
		$K = 5$	2.20	-0.12 ± 0.08	625	312	128.5 ± 5.2	75.0 ± 24.5	4.8 ± 11.3	5.3 ± 9.5	47.1 ± 28.9
		$K = L$	1.15	-0.13 ± 0.05	863	376	121.1 ± 6.2	77.9 ± 23.0	4.2 ± 9.0	3.4 ± 4.5	50.3 ± 29.2
PL	sDO	Base	1.50	-0.37 ± 0.29	939	749	127.5 ± 5.4	79.1 ± 18.0	19.5 ± 18.9	19.3 ± 18.8	39.9 ± 26.2
		$K = 5$	1.30	-0.35 ± 0.30	946	750	127.6 ± 5.5	80.9 ± 17.3	19.4 ± 19.5	18.2 ± 18.7	40.4 ± 26.1
		$K = L$	1.02	-0.34 ± 0.34	985	720	129.9 ± 6.9	81.9 ± 24.0	16.5 ± 20.2	14.9 ± 18.2	44.1 ± 27.3
		Base	3.00	-0.16 ± 0.12	526	297	127.9 ± 4.5	61.2 ± 27.9	4.9 ± 11.1	7.1 ± 10.6	48.9 ± 25.3
		$K = 5$	1.80	-0.16 ± 0.15	702	421	128.8 ± 5.5	77.7 ± 23.2	7.0 ± 12.3	7.3 ± 11.1	41.9 ± 27.0
		$K = L$	1.05	-0.16 ± 0.13	901	465	129.4 ± 6.1	80.9 ± 24.1	6.2 ± 10.6	5.2 ± 7.7	46.4 ± 29.4
		Base	5.00	-0.12 ± 0.61	213	96	126.8 ± 3.7	41.4 ± 29.3	1.1 ± 5.0	3.9 ± 5.7	63.9 ± 23.1
		$K = 5$	2.30	-0.12 ± 0.10	500	248	129.6 ± 5.7	74.5 ± 28.7	4.1 ± 11.0	4.4 ± 8.0	44.8 ± 27.2
		$K = L$	1.10	-0.12 ± 0.07	837	308	129.6 ± 6.3	79.5 ± 26.7	3.4 ± 7.6	2.9 ± 3.9	48.6 ± 29.5

Table 4. **Top-p sampling analysis.** We compare top-p decoding settings for the **sDO** architecture across the CM and PL families. Statistics are computed from 1024 generated sequences for each setting. Reported metrics include the target $\log P$ used for sampling, enabling comparison with the values in Table 3. At matched generative quality, these metrics are very similar to those obtained with standard autoregressive low-temperature decoding.

FAMILY	p	β	TARGET $\log P$	UNIQUE (ON 1024)	UNIQUE (RADIUS 5)	LENGTH	DIST (GEN–GEN)	MIN DIST (GEN–GEN)	MIN DIST (GEN–NAT)
CM	1.00	2.60	-0.23	744	420	89.9 ± 1.7	56.7 ± 21.7	5.4 ± 9.1	6.8 ± 7.4
	0.95	2.60		727	412	89.9 ± 1.7	58.3 ± 20.5	5.2 ± 8.9	6.9 ± 7.5
	0.90	2.30		707	411	89.9 ± 1.6	58.9 ± 19.8	5.0 ± 8.8	7.0 ± 7.7
	0.85	2.05		735	419	89.9 ± 1.6	59.6 ± 18.8	5.0 ± 8.3	7.1 ± 7.6
	0.80	1.85		735	405	89.9 ± 1.5	59.7 ± 18.9	4.6 ± 7.9	7.0 ± 7.4
PL	1.00	3.00	-0.16	527	294	123.5 ± 4.3	59.6 ± 25.9	4.2 ± 9.5	11.2 ± 9.2
	0.95	2.50		557	297	123.9 ± 4.4	62.1 ± 26.2	3.8 ± 8.5	11.2 ± 8.9
	0.90	2.20		563	309	124.1 ± 4.6	64.0 ± 26.1	3.7 ± 8.3	11.2 ± 8.7
	0.85	1.95		540	303	124.2 ± 4.8	65.2 ± 26.9	3.9 ± 8.2	11.3 ± 9.4
	0.80	1.75		530	307	124.3 ± 4.8	66.4 ± 26.6	3.8 ± 8.4	11.1 ± 9.0

Table 5. **Steering generation toward a Target wild-type (WT).** We report the mean Edit Distance to the target sequence and the generative quality ($\log P$). Statistics are computed over 1024 sequences per sample. Base refers to standard autoregressive sampling, while steered samples are generated using ILMC with parameters $B = 2$, $K = 5$, and $N_{\mathrm{MC}} = 10$.

FAMILY	MODEL	STRATEGY	DIST (TO WT)	$\log P$
CM	sDO	TARGET WT	0	-1.19
		BASE ($\beta = 1$)	64.2 ± 4.0	-0.83 ± 0.47
		$\beta = 1, \lambda^{\mathrm{WT}} = 2.5$	35.5 ± 15.9	-1.05 ± 0.34
		$\beta = 1, \lambda^{\mathrm{WT}} = 7.5$	4.3 ± 3.7	-1.16 ± 0.04
	ProGen3	TARGET WT	0	-1.08
		BASE ($\beta = 1.1$)	64.2 ± 4.0	-0.66 ± 0.37
		$\beta = 1.1, \lambda^{\mathrm{WT}} = 2.5$	45.2 ± 14.0	-0.89 ± 0.37
		$\beta = 1.1, \lambda^{\mathrm{WT}} = 7.5$	7.7 ± 5.6	-1.14 ± 0.10
PL	sDO	TARGET WT	0	-1.28
		BASE ($\beta = 1$)	95.1 ± 5.9	-0.97 ± 0.63
		$\beta = 1, \lambda^{\mathrm{WT}} = 2.5$	74.5 ± 14.5	-1.22 ± 0.51
		$\beta = 1, \lambda^{\mathrm{WT}} = 7.0$	21.9 ± 12.5	-1.63 ± 0.29
	ProGen3	TARGET WT	0	-1.03
		BASE ($\beta = 1.1$)	94.5 ± 7.1	-0.65 ± 0.44
		$\beta = 1.1, \lambda^{\mathrm{WT}} = 4.5$	52.9 ± 21.9	-1.47 ± 0.35
		$\beta = 1.1, \lambda^{\mathrm{WT}} = 10.0$	35.5 ± 21.3	-1.67 ± 0.30

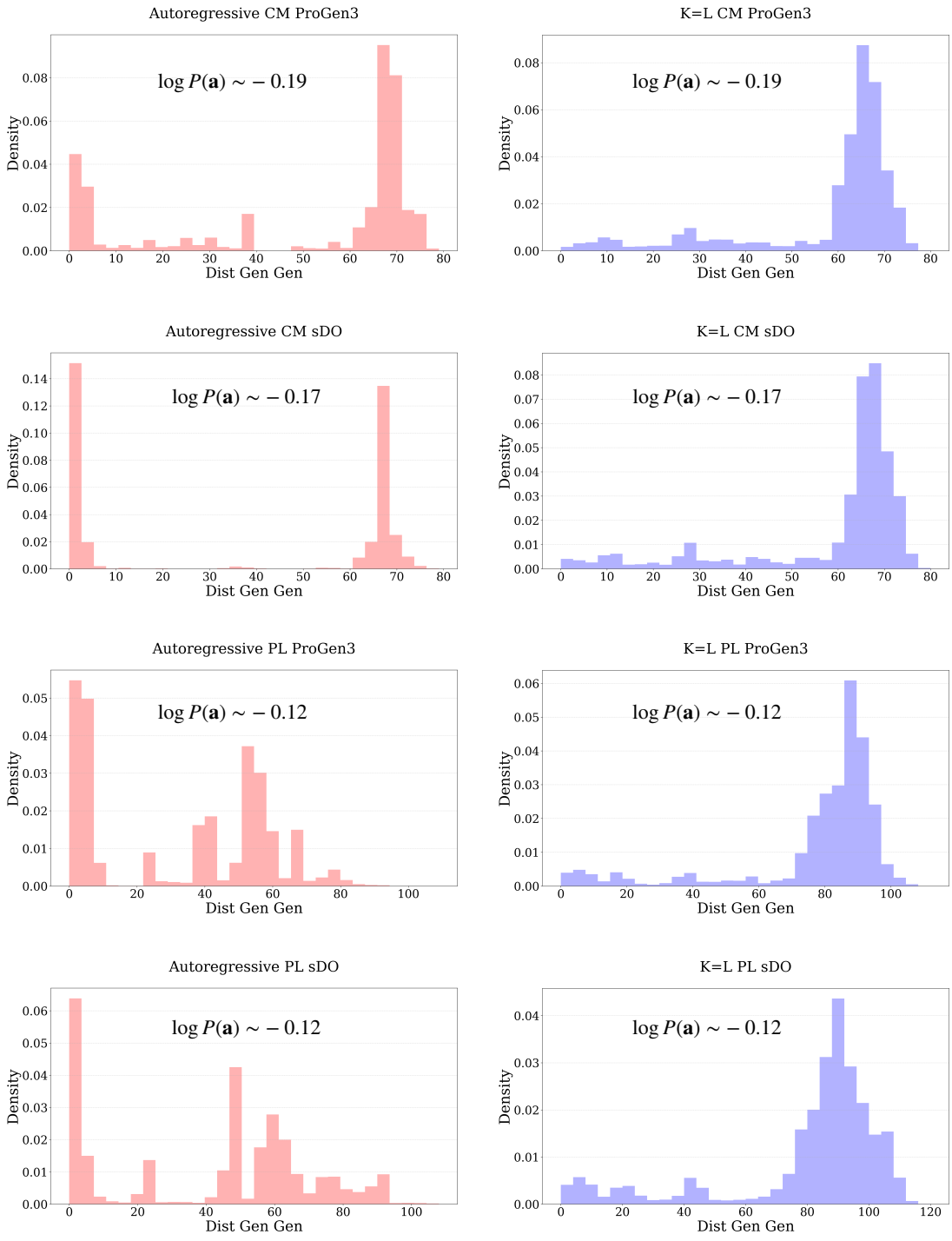

Figure 6. **Library intra-distance (Dist Gen-Gen) of the highest** $\log P$ **energy samples from Table 3.** Each row compares standard autoregressive sampling (left, red) and ILMC with $K = L$ ($B = 2, N_{\mathrm{MC}} = 10$) (right, blue), matched for the same mean $\log P$. The rows correspond to: (1) ProGen3 on CM, (2) sDO on CM, (3) ProGen3 on PL, and (4) sDO on PL. All samples consist of 1024 sequences. It is evident that the autoregressive sampling (left, red) is collapsing in diversity, as indicated by the consistent peak near 0 pairwise edit distance. ILMC samples (right column) do not exhibit this peak, consistently sampling libraries with typical pairwise distances close to the average distance between natural sequences (91 for PL and 64 for CM).

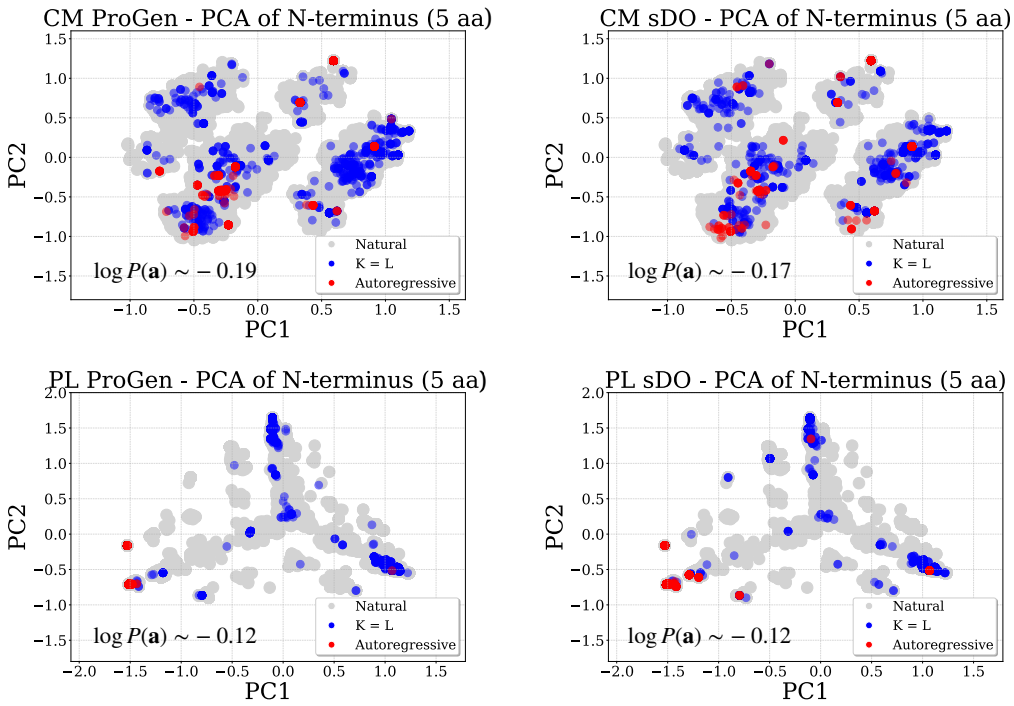

Figure 7. **PCA projection of the first 5 amino acids.** The panels correspond to the two families (CM top, PL bottom) and the two models (ProGen3 top, sDO bottom). The projection of the natural data is shown in gray. High $\log P$ samples generated via standard autoregressive sampling are shown in red, while samples generated via ILMC ($B = 2, K = L, N_{\mathrm{MC}} = 10$) are shown in blue. All samples consist of 1024 sequences. These distributions correspond to the highest $\log P$ samples reported in Table 3.

B. Proofs

B.1. Proof of Proposition 2.1

We seek to maximize the entropy $H[Q] = -\sum_{\mathbf{a}} Q(\mathbf{a}) \log Q(\mathbf{a})$ subject to the constraints $\mathbb{E}_Q[E(\mathbf{a})] = C_{\mathrm{E}}$, $\mathbb{E}_Q[S(\mathbf{a})] = C_{\mathrm{S}}$, and the normalization constraint $\sum_{\mathbf{a}} Q(\mathbf{a}) = 1$. The Lagrangian for this optimization problem is:

$$\mathcal{L} = -\sum_{\mathbf{a}} Q(\mathbf{a}) \log Q(\mathbf{a}) - \beta \left(\sum_{\mathbf{a}} Q(\mathbf{a}) E(\mathbf{a}) - C_{\mathrm{E}} \right) - \lambda \left(\sum_{\mathbf{a}} Q(\mathbf{a}) S(\mathbf{a}) - C_{\mathrm{S}} \right) - \gamma \left(\sum_{\mathbf{a}} Q(\mathbf{a}) - 1 \right) \tag{16}$$

Taking the functional derivative with respect to $Q(\mathbf{a})$ and setting it to zero:

$$\frac{\partial \mathcal{L}}{\partial Q(\mathbf{a})} = -\log Q(\mathbf{a}) - 1 - \beta E(\mathbf{a}) - \lambda S(\mathbf{a}) - \gamma = 0 \tag{17}$$

Solving for $Q(\mathbf{a})$ yields:

$$Q^*(\mathbf{a}) = \exp(-1 - \gamma) \exp(-\beta E(\mathbf{a}) - \lambda S(\mathbf{a})) \tag{18}$$

By defining the partition function $Z^*(\beta, \lambda) = \exp(1 + \gamma)$, so that (18) is a normalized distribution, we recover the form of the canonical Gibbs distribution:

$$Q^*(\mathbf{a}) = \frac{1}{Z^*(\beta, \lambda)} \exp(-\beta E(\mathbf{a}) - \lambda S(\mathbf{a})) \tag{19}$$

B.2. Proof of Proposition 2.2

Consider any distribution $Q(\mathbf{a})$ satisfying the constraints $\mathbb{E}_Q[E(\mathbf{a})] = C_{\mathrm{E}}$ and $\mathbb{E}_Q[S(\mathbf{a})] = C_{\mathrm{S}}$. The KL divergence from Q to the optimal Q^* is:

$$\begin{aligned} D_{\mathrm{KL}}(Q \parallel Q^*) &= \sum_{\mathbf{a}} Q(\mathbf{a}) \log \frac{Q(\mathbf{a})}{Q^*(\mathbf{a})} \\ &= -H[Q] - \sum_{\mathbf{a}} Q(\mathbf{a}) \log Q^*(\mathbf{a}) \end{aligned} \tag{20}$$

Substituting the form $\log Q^*(\mathbf{a}) = -\beta E(\mathbf{a}) - \lambda S(\mathbf{a}) - \log Z^*(\beta, \lambda)$:

$$\begin{aligned} D_{\mathrm{KL}}(Q \parallel Q^*) &= -H[Q] - \sum_{\mathbf{a}} Q(\mathbf{a}) \left(-\beta E(\mathbf{a}) - \lambda S(\mathbf{a}) - \log Z^*(\beta, \lambda)\right) \\ &= -H[Q] + \beta \left(\sum_{\mathbf{a}} Q(\mathbf{a}) E(\mathbf{a}) \right) + \lambda \left(\sum_{\mathbf{a}} Q(\mathbf{a}) S(\mathbf{a}) \right) + \log Z^*(\beta, \lambda) \\ &= -H[Q] + \beta C_{\mathrm{E}} + \lambda C_{\mathrm{S}} + \log Z^*(\beta, \lambda) \end{aligned} \tag{21}$$

Since the terms $\beta C_{\mathrm{E}} + \lambda C_{\mathrm{S}} + \log Z^*(\beta, \lambda)$ are independent of Q, we have that $D_{\mathrm{KL}}(Q \parallel Q^*)$ is minimized as a function of Q if and only if $H[Q]$ is maximized.

C. Greedy low-temperature sampling in autoregressive models

A common strategy employed to sample high probability sequences from autoregressive models is *greedy* "low-temperature" autoregressive sampling. Despite the name, it differs from low-temperature sampling in the statistical physics sense (Karan & Du, 2025), which corresponds to sampling from a exponentiated distribution $Q^*(\mathbf{a}) \propto P(\mathbf{a})^{\beta}$. Here, we analyze the discrepancy between these two strategies and explain why direct autoregressive sampling from the optimal entropy-constrained distribution Q^* is computationally intractable. In this section we focus on the non-steered case (*i.e.*, $\lambda = 0$ in Eq. (1)) for simplicity. The generalization of the analysis discussion to the steered case is very similar and is given in Appendix F.

Let a protein sequence be denoted by $\mathbf{a} = (a_1, \ldots, a_L)$. The pre-trained autoregressive model provides a probability distribution $P(\mathbf{a}) = \prod_{t=1}^{L} P(a_t | \mathbf{a}_{<t})$. Standard low-temperature sampling sharpens the next-token probabilities at each

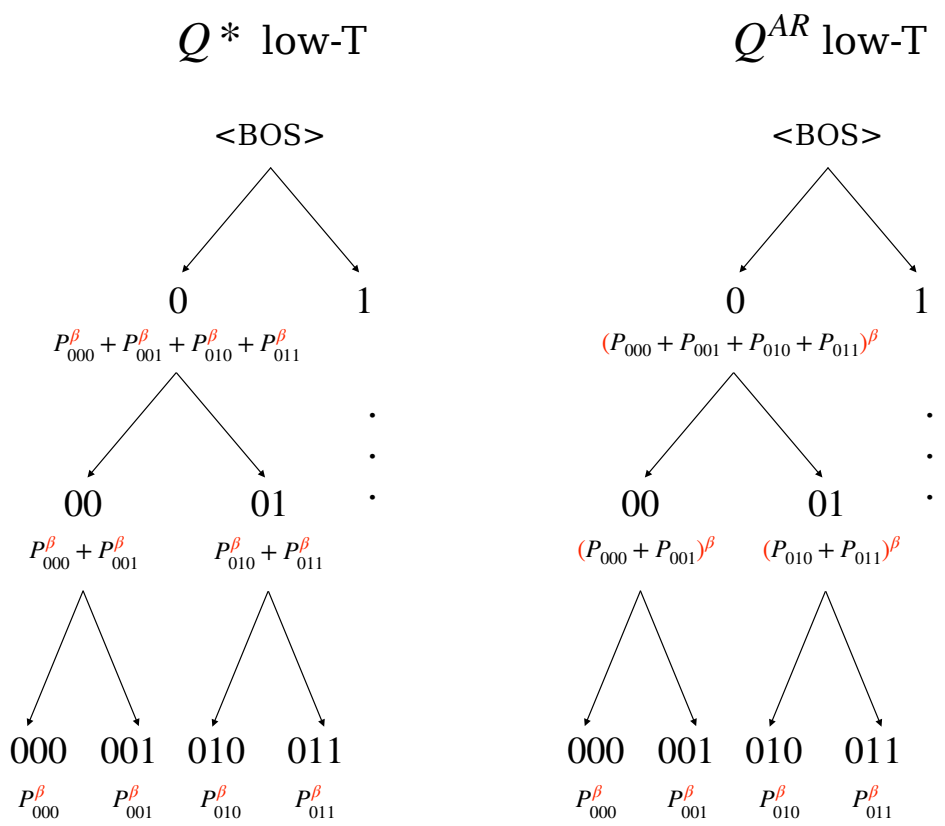

Figure 8. Illustrated example of the sampling process as a branching tree decomposition for a binary sequence of length 3. The values at each node represent the relative weight of that branch in relation to the concurrent one. **Left:** The optimal global distribution Q^*, derived by exponentiating the probabilities of the leaves. **Right:** The standard autoregressive Q^{AR}, derived by exponentiating local transitions. At any given node, the autoregressive model only provides access to the standard probability mass of its children (the sum in black, corresponding to $\beta = 1$). For the standard strategy (Right), the sampling weights are trivially obtained by exponentiating these local values. However, for the optimal strategy (Left), the correct node weight corresponds to the sum of the exponentiated leaves. Since the summation of probabilities does not commute with exponentiation (i.e., $(\sum p)^\beta \neq \sum p^\beta$), determining the weights for Q^* requires an intractable summation over all future trajectories rather than a simple local operation.

step t toward high-probability tokens by exponentiating their distribution with an inverse temperature parameter β. Given a prefix $\mathbf{a}_{<t}$, the next token is drawn from a locally re-normalized distribution:

$$Q^{\mathrm{AR}}(a_t|\mathbf{a}_{<t}) = \frac{P(a_t|\mathbf{a}_{<t})^\beta}{Z_t(\mathbf{a}_{<t};\beta)} \tag{22}$$

where $Z_t(\mathbf{a}_{<t};\beta) = \sum_{v\in\mathcal{A}} P(v|\mathbf{a}_{<t})^\beta$ sums only over the vocabulary $\mathcal{A}$. Consequently, the probability of a full sequence generated via this strategy is simply the multiplication of its token draw probabilities:

$$Q^{\mathrm{AR}}(\mathbf{a}) = \prod_{t=1}^{L} \frac{P(a_t|\mathbf{a}_{<t})^\beta}{Z_t(\mathbf{a}_{<t};\beta)} = \frac{P(\mathbf{a})^\beta}{\prod_{t=1}^{L} Z_t(\mathbf{a}_{<t};\beta)} \tag{23}$$

On the other hand, as derived in Proposition 2.1, the unique distribution that maximizes entropy for a fixed mean energy is the Boltzmann distribution $Q^*(\mathbf{a}) = P(\mathbf{a})^\beta / Z^*(\beta)$, where $Z^*(\beta)$ is the global partition function summing over all possible sequences in the space $\mathcal{A}^L$.

To understand why $Q^{\mathrm{AR}} \neq Q^*$, we must examine the conditional probability of the next token a_t from the optimal distribution Q^*. By definition, $Q^*(a_t|\mathbf{a}_{<t}) = Q^*(\mathbf{a}_{\leq t})/Q^*(\mathbf{a}_{<t})$. Here, the marginal probability of a prefix is obtained by

summing the global probabilities of all completed sequences (the "leaves") rooted at that prefix. We derive the expression for the optimal conditional probability starting from this definition:

$$Q^*(a_t|\mathbf{a}_{<t}) = \frac{Q^*(\mathbf{a}_{\leq t})}{Q^*(\mathbf{a}_{<t})} \tag{24}$$

$$= \frac{\sum_{\mathbf{a}'_{>t}} Q^*(\mathbf{a}_{\leq t}, \mathbf{a}'_{>t})}{\sum_{a'_t} \sum_{\mathbf{a}'_{>t}} Q^*(\mathbf{a}_{<t}, a'_t, \mathbf{a}'_{>t})} \tag{25}$$

Substituting the definition of the global Boltzmann distribution $Q^*(\mathbf{a}) = \frac{1}{Z^*(\beta)} P(\mathbf{a})^\beta$:

$$Q^*(a_t|\mathbf{a}_{<t}) = \frac{\frac{1}{Z^*(\beta)} \sum_{\mathbf{a}'_{>t}} P(\mathbf{a}_{\leq t}, \mathbf{a}'_{>t})^\beta}{\frac{1}{Z^*(\beta)} \sum_{a'_t} \sum_{\mathbf{a}'_{>t}} P(\mathbf{a}_{<t}, a'_t, \mathbf{a}'_{>t})^\beta} \tag{26}$$

$$= \frac{\sum_{\mathbf{a}'_{>t}} (P(\mathbf{a}_{\leq t}) P(\mathbf{a}'_{>t}|\mathbf{a}_{\leq t}))^\beta}{\sum_{a'_t} \sum_{\mathbf{a}'_{>t}} (P(\mathbf{a}_{<t}, a'_t) P(\mathbf{a}'_{>t}|\mathbf{a}_{<t}, a'_t))^\beta} \tag{27}$$

We factor out the prefix probabilities which are constant with respect to the summation over future trajectories $\mathbf{a}'_{>t}$:

$$Q^*(a_t|\mathbf{a}_{<t}) = \frac{P(\mathbf{a}_{\leq t})^\beta \sum_{\mathbf{a}'_{>t}} P(\mathbf{a}'_{>t}|\mathbf{a}_{\leq t})^\beta}{\sum_{a'_t} \left[P(\mathbf{a}_{<t}, a'_t)^\beta \sum_{\mathbf{a}'_{>t}} P(\mathbf{a}'_{>t}|\mathbf{a}_{<t}, a'_t)^\beta \right]} \tag{28}$$

The denominator is a normalization constant involving a summation over all possible next tokens a'_t. To find the proportionality relation for a specific a_t, we focus on the numerator. Using the decomposition $P(\mathbf{a}_{\leq t}) = P(\mathbf{a}_{<t}) P(a_t|\mathbf{a}_{<t})$:

$$Q^*(a_t|\mathbf{a}_{<t}) \propto P(\mathbf{a}_{\leq t})^\beta \sum_{\mathbf{a}'_{>t}} P(\mathbf{a}'_{>t}|\mathbf{a}_{\leq t})^\beta \tag{29}$$

$$\propto (P(\mathbf{a}_{<t}) P(a_t|\mathbf{a}_{<t}))^\beta \sum_{\mathbf{a}'_{>t}} P(\mathbf{a}'_{>t}|\mathbf{a}_{\leq t})^\beta \tag{30}$$

Since $P(\mathbf{a}_{<t})^\beta$ depends only on the fixed history and is constant with respect to the choice of a_t, it can be absorbed into the proportionality constant. Defining the lookahead term,

$$\mathcal{Z}_{\text{future}}(a_t, \mathbf{a}_{<t}) \equiv \sum_{\mathbf{a}'_{>t}} P(\mathbf{a}'_{>t}|\mathbf{a}_{\leq t})^\beta, \tag{31}$$

we arrive at the final relation:

$$Q^*(a_t|\mathbf{a}_{<t}) \propto \underbrace{P(a_t|\mathbf{a}_{<t})^\beta}_{\text{Autoregressive Term}} \times \underbrace{\mathcal{Z}_t^{\text{future}}(a_t, \mathbf{a}_{<t})}_{\text{Lookahead Term}} \tag{32}$$

Equation (32) highlights the fundamental flaw in greedy autoregressive low-temperature sampling. Q^{AR} selects the next token based solely on the Autoregressive Term, effectively assuming that $\mathcal{Z}_{\text{future}}(a_t, \mathbf{a}_{<t})$ does not depend on a_t. However, $\mathcal{Z}_t^{\text{future}}(a_t, \mathbf{a}_{<t})$ may vary significantly: it measures the "volume" of high-probability paths accessible from the current state, which in some cases may depend critically on the choice of the current token a_t. Autoregressive sampling is therefore myopic; it sharpens the next token distribution based only on the immediate edge in the probability tree (Figure 8, right) and may select a token that has high local probability but leads to a "dead end" in the energy landscape. In contrast, global sampling is prescient, re-weighting the immediate choice a_t by the exponentiated weight of the reachable ending sequences (leaves).

To illustrate this effect, consider a simple scenario where, conditioned on a specific prefix, there are m equally probable future completions (paths). The future weight term behaves as:

$$\mathcal{Z}^{\text{future}} \approx \sum_{i=1}^{m} \left(\frac{1}{m} \right)^\beta = m \cdot m^{-\beta} = m^{1-\beta} \tag{33}$$

Since $\beta > 1$, the exponent $1 - \beta$ is negative. This implies that as m increases (representing a more diffuse or uncertain future), the term $\mathcal{Z}_{\text{future}}$ decreases. Consequently, the optimal distribution Q^* penalizes paths that lead to high-entropy futures, favoring sequences that converge toward specific, high-likelihood outcomes. The greedy autoregressive sampler Q^{AR} is blind to this downstream uncertainty.

This creates an apparent paradox: Q^* is the unique maximum entropy distribution for a fixed mean generative quality, yet it penalizes diffuse (high-entropy) futures more aggressively than Q^{AR}. The resolution lies in the comparison of the sampling parameters. To achieve an identical mean generative energy using the myopic autoregressive strategy Q^{AR}, one must employ a significantly higher inverse temperature $\beta_{AR} \gg \beta$. This excessive local sharpening in Q^{AR} collapses the distribution's support early in the generation process, severely restricting diversity.

This analysis also explains why sampling from Q^* in a single autoregressive pass is impossible. Evaluating $\mathcal{Z}_t^{\text{future}}(\mathbf{a}_{\leq t})$ requires summing over $|\mathcal{A}|^{L-t}$ future trajectories (see Figure 8), which is computationally intractable for typical protein sequence lengths. This justifies the need for the ILMC method, which approximates this global distribution.

We also note that $\frac{1}{1-\beta} \log \mathcal{Z}_t^{\text{future}}$ is mathematically equivalent to the Rènyi entropy of the distribution $P(\mathbf{a}_{>t}|\mathbf{a}_{\leq t})$ of continuations $\mathbf{a}_{>t}$ conditioned on a given prefix $\mathbf{a}_{\leq t}$. Rènyi entropies are often used as diversity measures (Hill, 1973), in part because of the following well-known properties:

- For $\beta > 1$, the Rènyi entropy is larger for prefixes $\mathbf{a}_{\leq t}$ for which the conditional distribution $P(\mathbf{a}_{>t}|\mathbf{a}_{\leq t})$ has few but high-probability continuations $\mathbf{a}_{>t}$.
- For $\beta < 1$, the Rènyi entropy is larger for prefixes $\mathbf{a}_{\leq t}$ for which the conditional distribution $P(\mathbf{a}_{>t}|\mathbf{a}_{\leq t})$ has many but low-probability continuations $\mathbf{a}_{>t}$.
- As $\beta \to 1$, the Rènyi entropy approaches the Shannon entropy, which is maximized when the conditional distribution $P(\mathbf{a}_{>t}|\mathbf{a}_{\leq t})$ becomes a flat uniform measure over the possible continuations $\mathbf{a}_{>t}$.

The behavior of $\mathcal{Z}_t^{\text{future}}$ at $\beta > 1$ drives Q^* towards high-probability sequences.

D. Is $K > 1$ Always Closer to the Optimal Distribution?

Based on Proposition 2.2, minimizing the KL divergence (D_{KL}) between a sampling strategy and the optimal distribution Q^* is equivalent to maximizing the entropy of the sampled library for a fixed generative energy. Since the ILMC method samples from Q^* in the limit where the lookback window K equals the sequence length L and the number of MC steps $N_{\text{MC}} \to \infty$, it is intuitive to hypothesize that increasing K monotonically improves the approximation of Q^*. Specifically, one might expect that a strategy with a larger lookback window K would always yield higher entropy for a given target energy compared to a more myopic strategy.

While this intuition holds true in our computational experiments, as shown in Section 4, it is not universally correct. We demonstrate this with a counter-example where $K = 2$ and $B = 2$ (effectively corresponding to an autoregressive strategy that samples token pairs rather than single tokens) yields lower entropy with respect to the standard autoregressive strategy for a given level of mean energy. Consider a binary sequence space of length $L = 4$ governed by the following probability distribution $P(\mathbf{a})$:

$$P(\mathbf{a}) = \begin{cases} 0100 & \text{with prob } 0.44 \quad \text{(Ground State A)} \\ 0110 & \text{with prob } 0.06 \\ 1010 & \text{with prob } 0.44 \quad \text{(Ground State B)} \\ 1100 & \text{with prob } 0.06 \\ \text{others} & \text{with prob } 0.00 \end{cases} \tag{34}$$

Let's consider the limit where the inverse temperature goes to infinity $\beta \to \infty$. In this situation the sampler is forced to strictly follow the path of highest total probability mass. We observe a critical divergence in behavior: the single-token sampler preserves the symmetry between the two ground states, whereas the block sampler breaks this symmetry, leading to mode collapse.

Case 1: Single-Token Autoregressive Sampling. The decision process for the single-token sampler ($B = 1$) starts on the first token. The marginal probability of starting with `0` is the sum of probabilities for sequences `0100` and `0110` ($0.44 + 0.06 = 0.50$). Similarly, the marginal probability of starting with `1` is the sum for `1010` and `1100` ($0.44 + 0.06 = 0.50$). Since the initial masses are identical (0.50 vs 0.50), the sampler selects the initial token with equal probability. Consequently, the sampler can settle into either Ground State A or Ground State B with equal likelihood ($P = 0.5$). The resulting entropy is $\log 2$. See Figure 9.

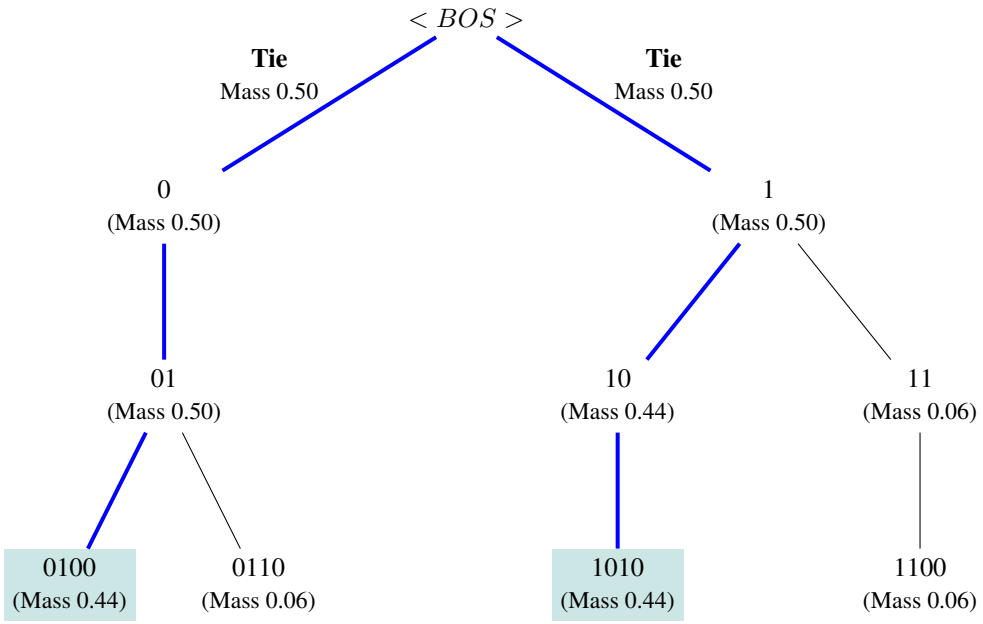

Figure 9. Case 1: Single-Token Autoregressive Sampling.

Case 2: $K = 2$, $B = 2$ Sampling. Now, consider a block size of $B = 2$ tokens. The sampler compares the joint probabilities of the prefixes (x_1, x_2) directly. The valid prefixes are `01`, `10`, and `11`.

Here, the symmetry is broken. The block `01` aggregates the mass of the dominant state A and the minor state ($0.44 + 0.06 = 0.50$). However, due to the sequence structure, the path to Ground State B requires the block `10`, which has a mass of only 0.44.

Because $0.50 > 0.44$, the block sampler in the limit of high β will exclusively select the `01` branch. It essentially "blindly" discards Ground State B because its associated prefix mass is lower, even though the final state probability is identical to Ground State A. The entropy for this strategy drops to 0. See Figure 10.

This phenomenon is not limited to the ground state limit but persists across intermediate energy regimes. Figure 11 demonstrates that the single-token strategy yields strictly higher entropy for a given energy level compared to the block-wise approach.

E. Good and Bad Steering Potentials: Illustrative Examples

Here we illustrate two example steering potentials, a *bad* one that results in a poor sampling strategy, and a *good* one that guides the sampling effectively.

If the steering potential only depends on the terminal portion of the sequence, the early stages of generation will lack guidance, potentially driving the chain into regions of the sequence space where the desired characteristic is no longer reachable. To illustrate this necessity, consider a situation in which we want to steer the generation toward sequences that exhibit a specific target motif, defined as a fixed reference sub-sequence $\mathbf{s} = (s_m, \ldots, s_l)$ residing within the position

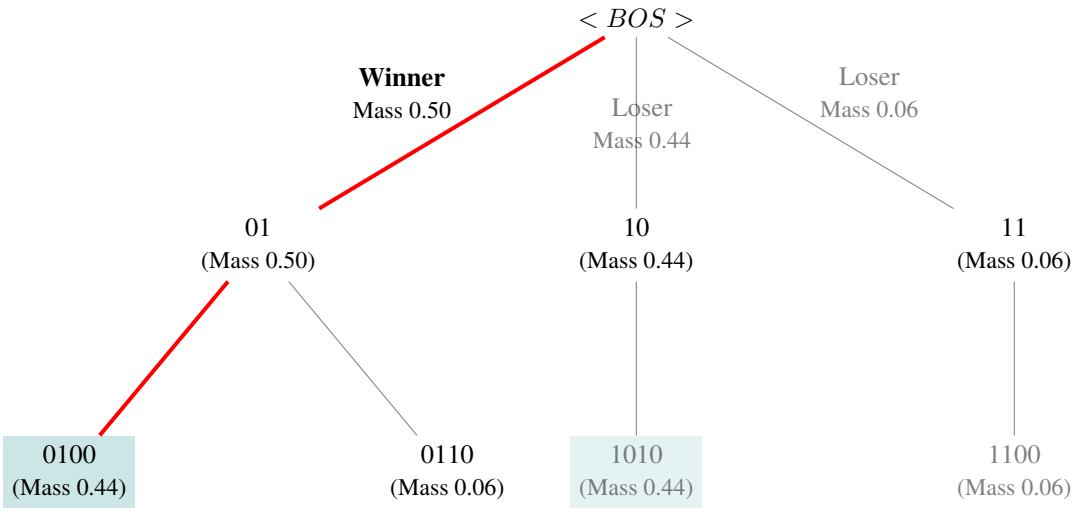

Figure 10. Case 2: $K = 2$, $B = 2$ sampling.

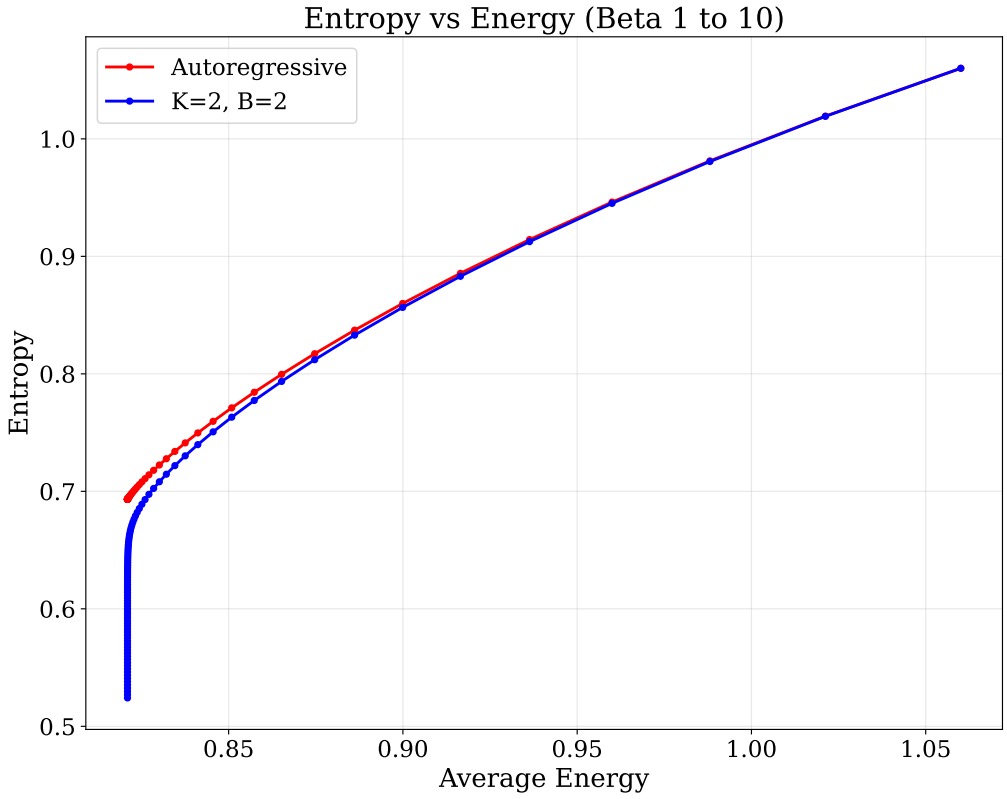

Figure 11. Comparison of the Energy-Entropy frontier between the standard single-token autoregressive sampler and the block-wise Iterative Lookback sampler ($K = 2, B = 2$) using as baseline the example distribution $P(a)$. As the inverse temperature β increases from 1 to 10 in increments of 0.1, the autoregressive sampler maintains higher entropy at given average energy values compared to the $K = 2, B = 2$ ILMC strategy.

window $[m, l]$. A natural steering potential to quantify the deviation from this target is the Hamming distance:

$$S^{\text{bad}}(\mathbf{a}) = \sum_{i=m}^{l} \mathbb{I}(a_i \neq s_i) \tag{35}$$

The corresponding partial steering potential $S_t^{\text{bad}}(\mathbf{a}_{\leq t})$ can be defined as the Hamming distance to the motif evaluated only on the residues currently generated. For any $t < m$, the partial potential $S_t^{\text{bad}}(\mathbf{a}_{\leq t})$ is trivially constant. Consequently, the refinement steps during the first m tokens provide no steering. The model generates a prefix guided solely by the base generative energy $E_t(\mathbf{a}_{\leq t})$. By the time the sampler reaches index m, the prefix may be structurally incompatible with the required motif, leading to a poor generative quality.

To illustrate an example of an effective steering potential, consider the case where the objective is to modify the amino acid composition of the generated proteins, for instance, by increasing the frequency of a desired amino acid A. The global score $S^{\text{good}}(\mathbf{a})$ is the negative total count of A, while the partial score is the cumulative count:

$$S_t^{\text{good}}(\mathbf{a}_{\leq t}) = -\sum_{i=1}^{t} \mathbb{I}(a_i = A) \tag{36}$$

In this case, the steering is effective from the very first token. The refinement steps immediately bias the sequence toward a higher frequency of A, ensuring that the early prefix correlates with the desired final outcome.

F. What makes a steering potential effective?

Practically, steering potentials are most effective when their prefix form provides an informative signal throughout generation, so that early token choices do not create large, hard-to-correct differences in the remaining achievable score. In this section we attempt to make this intuition more precise. In doing so we generalize some of the calculations from Section C, which did not consider steering, to the steered case.

We begin by recalling the target distribution, which we write as:

$$Q^*(\mathbf{a}) = \frac{1}{Z^*} e^{-U(\mathbf{a})}, \quad \text{with} \quad U(\mathbf{a}) = \beta E(\mathbf{a}) + \lambda S(\mathbf{a}), \tag{37}$$

as introduced in (1), where

$$Z^* = \sum_{\mathbf{a}} e^{-U(\mathbf{a})} \tag{38}$$

is a normalization constant. We suppose that we can decompose these potentials as follows:

$$E(\mathbf{a}) = \sum_i e_i(\mathbf{a}_{\leq i}), \quad S(\mathbf{a}) = \sum_i s_i(\mathbf{a}_{\leq i}), \quad U(\mathbf{a}) = \sum_i u_i(\mathbf{a}_{\leq i}) \tag{39}$$

where the sum considers the contribution of each token, conditioned on the preceding tokens. Note that such a decomposition is formally always possible. For example, given an arbitrary potential $S(\mathbf{a})$, it suffices to define

$$s_i(\mathbf{a}_{\leq i}) = \begin{cases} 0 & i < L \\ S(\mathbf{a}) & i = L \end{cases} \tag{40}$$

Then Eq. (39) is trivially satisfied. But although this definition is mathematically compatible with (39), it would be terrible from the point of view of the ILMC algorithm we introduce in this work, as we will see.

On the other hand, there are several ways to achieve a decomposition like (39). For example, given arbitrary functions $\phi_0, \ldots, \phi_L$, we have that

$$u_i'(\mathbf{a}_{\leq i}) = u_i(\mathbf{a}_{\leq i}) + \phi_i(\mathbf{a}_{\leq i}) - \phi_{i-1}(\mathbf{a}_{\leq i-1}) \tag{41}$$

is also compatible with the global potential when summed across the full sequence,

$$\sum_i u_i'(\mathbf{a}_{\leq i}) = \sum_i u_i(\mathbf{a}_{\leq i}) = U(\mathbf{a}) \tag{42}$$

provided that $\phi_0 \equiv 0$ and $\phi_L \equiv 0$. This telescoping gauge invariance may allow us to reallocate lookahead information into prefix terms, which in turn makes ILMC more effective than a naive decomposition such as Eq. (40).

As in the main-text, we consider a generated prefix $\mathbf{a}_{\leq t}$, and define the partial quantities:

$$E_t(\mathbf{a}_{\leq t}) = \sum_{i \leq t} e_i(\mathbf{a}_{\leq i}), \quad S_t(\mathbf{a}_{\leq t}) = \sum_{i \leq t} s_i(\mathbf{a}_{\leq i}), \quad U_t(\mathbf{a}_{\leq t}) = \sum_{i \leq t} u_i(\mathbf{a}_{\leq i}) \tag{43}$$

Now, substituting the decomposition (39), the target distribution Q^* can be written:

$$Q^*(\mathbf{a}) = \frac{1}{Z^*} \exp\left\{ -\sum_j u_j(\mathbf{a}_{\le j}) \right\} \tag{44}$$

We focus on the generation of a token at site i. To emphasize the dependence on a_i, we split the exponent as follows:

$$Q^*(\mathbf{a}) = \frac{1}{Z^*} \exp\left\{ -\sum_{j<i} u_j(\mathbf{a}_{\le j}) - u_i(a_i, \mathbf{a}_{<i}) - \sum_{j>i} u_j(\mathbf{a}_{i+1:j}, a_i, \mathbf{a}_{<i}) \right\} \tag{45}$$

Consider the next-token distribution, $Q_i^*(a_i|\mathbf{a}_{<i})$. Following calculations very similar to those already given in Sec. C,

$$Q_i^*(a_i|\mathbf{a}_{<i}) = \frac{\sum_{\mathbf{a}'_{>i}} Q^*(\mathbf{a}'_{>i}, a_i, \mathbf{a}_{<i})}{\sum_{\mathbf{a}'_{\ge i}} Q^*(\mathbf{a}'_{>i}, a'_i, \mathbf{a}_{<i})} \tag{46}$$

$$= \frac{\sum_{\mathbf{a}'_{>i}} \exp\left\{ -\sum_{j<i} u_j(\mathbf{a}_{\le j}) - u_i(a_i, \mathbf{a}_{<i}) - \sum_{j>i} u_j(\mathbf{a}'_{i+1:j}, a_i, \mathbf{a}_{<i}) \right\}}{\sum_{\mathbf{a}'_{\ge i}} \exp\left\{ -\sum_{j<i} u_j(\mathbf{a}_{\le j}) - u_i(a'_i, \mathbf{a}_{<i}) - \sum_{j>i} u_j(\mathbf{a}'_{i+1:j}, a'_i, \mathbf{a}_{<i}) \right\}} \tag{47}$$

$$= \exp\{-u_i(a_i, \mathbf{a}_{<i})\} \frac{\sum_{\mathbf{a}'_{>i}} \exp\left\{ -\sum_{j>i} u_j(\mathbf{a}'_{i+1:j}, a_i, \mathbf{a}_{<i}) \right\}}{\sum_{\mathbf{a}'_{\ge i}} \exp\left\{ -u_i(a'_i, \mathbf{a}_{<i}) - \sum_{j>i} u_j(\mathbf{a}'_{i+1:j}, a'_i, \mathbf{a}_{<i}) \right\}} \tag{48}$$

$$= Q_i^{\mathrm{AR}}(a_i|\mathbf{a}_{<i}) \frac{\mathcal{Z}_i^{\mathrm{future}}(a_i, \mathbf{a}_{<i})}{\mathbb{E}_{a'_i \sim Q_i(a'_i|\mathbf{a}_{<i})} \mathcal{Z}_i^{\mathrm{future}}(a'_i, \mathbf{a}_{<i})} \tag{49}$$

where

$$Q_i^{\mathrm{AR}}(a_i|\mathbf{a}_{<i}) = \frac{\exp\{-u_i(a_i, \mathbf{a}_{<i})\}}{\sum_{a'_i} \exp\{-u_i(a'_i, \mathbf{a}_{<i})\}} \tag{50}$$

is the greedy autoregressive sampler, and

$$\mathcal{Z}_i^{\mathrm{future}}(\mathbf{a}_{\le i}) = \sum_{\mathbf{a}_{>i}} \exp\left\{ -\sum_{j>i} u_j(\mathbf{a}_{\le j}) \right\} \tag{51}$$

Note that in contrast to the previously defined (23) and (31) from Sec. C, we now incorporate the prefix steering potential as part of Q^{AR}.

To keep the calculation simple, we have effectively assumed that autoregressive sampling is able to incorporate the prefix steering potential exactly in the next-token proposal. Note that this can always be achieved by making N_{MC} sufficiently large, regardless of the block-size B or the lookback window K parameters of ILMC. Even if the next-token proposal incorporates steering exactly, the sampling strategy may still deviate significantly from the target Q^*, due to the finiteness of K or B. In the calculation that follows, we can either assume that $B = K = 1$, or that the tokens actually represent blocks of size $B = K$.

Thus, now the goal is to compare the autoregressive sampler:

$$Q^{\mathrm{AR}}(\mathbf{a}) = \prod_i Q_i^{\mathrm{AR}}(a_i|\mathbf{a}_{<i}) \tag{52}$$

to the target distribution Q^*. To do so, we compute their KL divergence. Since we also have $Q^*(\mathbf{a}) = \prod_i Q_i^*(a_i|\mathbf{a}_{<i})$, we can write:

$$D_{\mathrm{KL}}(Q^{\mathrm{AR}} \| Q^*) = \sum_{\mathbf{a}} Q^{\mathrm{AR}}(\mathbf{a}) \log \frac{Q^{\mathrm{AR}}(\mathbf{a})}{Q^*(\mathbf{a})} = \sum_i \sum_{\mathbf{a}} Q^{\mathrm{AR}}(\mathbf{a}_{<i}) Q_i^{\mathrm{AR}}(a_i|\mathbf{a}_{<i}) \log \frac{Q_i^{\mathrm{AR}}(a_i|\mathbf{a}_{<i})}{Q_i^*(a_i|\mathbf{a}_{<i})} \tag{53}$$

$$= \sum_i \mathbb{E}_{\mathbf{a}_{<i} \sim Q^{\mathrm{AR}}} D_{\mathrm{KL}}\left(Q_i^{\mathrm{AR}}(\cdot|\mathbf{a}_{<i}) \| Q_i^*(\cdot|\mathbf{a}_{<i}) \right) \tag{54}$$

with

$$D_{\mathrm{KL}}\left(Q_i^{\mathrm{AR}}(\cdot|\mathbf{a}_{<i})\|Q_i^*(\cdot|\mathbf{a}_{<i})\right) = \sum_{a_i} Q_i^{\mathrm{AR}}(a_i|\mathbf{a}_{<i}) \log \frac{Q_i^{\mathrm{AR}}(a_i|\mathbf{a}_{<i})}{Q_i^*(a_i|\mathbf{a}_{<i})} \tag{55}$$

$$= -\sum_{a_i} Q_i^{\mathrm{AR}}(a_i|\mathbf{a}_{<i}) \log \frac{\mathcal{Z}_i^{\mathrm{future}}(a_i, \mathbf{a}_{<i})}{\mathbb{E}_{a_i' \sim Q_i^{\mathrm{AR}}(a_i'|\mathbf{a}_{<i})} \mathcal{Z}_i^{\mathrm{future}}(a_i', \mathbf{a}_{<i})} \tag{56}$$

$$= \log \mathbb{E}_{a_i \sim Q_i^{\mathrm{AR}}(a_i|\mathbf{a}_{<i})} \mathcal{Z}_i^{\mathrm{future}}(a_i, \mathbf{a}_{<i}) - \mathbb{E}_{a_i \sim Q_i^{\mathrm{AR}}(a_i|\mathbf{a}_{<i})} \log \mathcal{Z}_i^{\mathrm{future}}(a_i, \mathbf{a}_{<i}) \tag{57}$$

Therefore, $Q_i^{\mathrm{AR}}(a_i|\mathbf{a}_{<i})$ equals $Q_i^*(a_i|\mathbf{a}_{<i})$ if and only if $\mathcal{Z}_i^{\mathrm{future}}(a_i, \mathbf{a}_{<i})$ does not depend on a_i within the support of $Q_i^{\mathrm{AR}}(a_i|\mathbf{a}_{<i})$. Otherwise, any dependence of the lookahead term $\mathcal{Z}_i^{\mathrm{future}}(a_i, \mathbf{a}_{<i})$ on the current token a_i, manifests itself in a larger KL divergence between the autoregressive proposal $Q_i^{\mathrm{AR}}(a_i|\mathbf{a}_{<i})$ and the target $Q_i^*(a_i|\mathbf{a}_{<i})$, which in turn results in global errors in (53).

An easy bound on the error induced by the non-flatness of the lookahead term is obtained by defining:

$$r_i(\mathbf{a}_{<i}) = \log \frac{\max_{a_i} \mathcal{Z}_i^{\mathrm{future}}(a_i, \mathbf{a}_{<i})}{\min_{a_i} \mathcal{Z}_i^{\mathrm{future}}(a_i, \mathbf{a}_{<i})} \tag{58}$$

Then it follows from (55) that:

$$D_{\mathrm{KL}}\left(Q_i^{\mathrm{AR}}(\cdot|\mathbf{a}_{<i})\|Q_i^*(\cdot|\mathbf{a}_{<i})\right) \leq \log \max_{a_i} \mathcal{Z}_i^{\mathrm{future}}(a_i, \mathbf{a}_{<i}) - \log \min_{a_i} \mathcal{Z}_i^{\mathrm{future}}(a_i, \mathbf{a}_{<i}) = r_i(\mathbf{a}_{<i}) \tag{59}$$

If we furthermore have a uniform bound over all prefixes:

$$R_i = \max_{\mathbf{a}_{<i}} r_i(\mathbf{a}_{<i}) \tag{60}$$

then an immediate consequence of (53) is that:

$$D_{\mathrm{KL}}(Q^{\mathrm{AR}}\|Q^*) \leq \sum_i R_i \tag{61}$$

which bounds the global error on the approximation of the target distribution.

Although computing this error bound may be difficult in practice, this calculation proves that making $\mathcal{Z}_i^{\mathrm{future}}(a_i, \mathbf{a}_{<i})$ as flat as possible as a function of a_i, or more precisely minimizing (60), brings the distribution Q^{AR} closer to the target.

We can also obtain a bound that relates more directly to the variation of the potential terms $u_j(\mathbf{a}_{\leq j})$. Defining:

$$u_{i\to j}^{\max}(\mathbf{a}_{<i}, \mathbf{a}_{i+1:j}) = \max_{a_i} u_j(\mathbf{a}_{<i}, a_i, \mathbf{a}_{i+1:j}), \quad u_{i\to j}^{\min}(\mathbf{a}_{<i}, \mathbf{a}_{i+1:j}) = \min_{a_i} u_j(\mathbf{a}_{<i}, a_i, \mathbf{a}_{i+1:j}) \tag{62}$$

for $i < j$, it is clear that:

$$\sum_{\mathbf{a}_{>i}} \exp\left\{-\sum_{j>i} u_{i\to j}^{\max}(\mathbf{a}_{<i}, \mathbf{a}_{i+1:j})\right\} \leq \mathcal{Z}_i^{\mathrm{future}}(\mathbf{a}_{\leq i}) \leq \sum_{\mathbf{a}_{>i}} \exp\left\{-\sum_{j>i} u_{i\to j}^{\min}(\mathbf{a}_{<i}, \mathbf{a}_{i+1:j})\right\} \tag{63}$$

Hence,

$$r_i(\mathbf{a}_{<i}) \leq \log \frac{\sum_{\mathbf{a}_{i+1:j}} \exp\left\{-\sum_{j>i} u_{i\to j}^{\min}(\mathbf{a}_{<i}, \mathbf{a}_{i+1:j})\right\}}{\sum_{\mathbf{a}_{i+1:j}} \exp\left\{-\sum_{j>i} u_{i\to j}^{\max}(\mathbf{a}_{<i}, \mathbf{a}_{i+1:j})\right\}}. \tag{64}$$

Next, defining:

$$\Delta_{i\to j}(\mathbf{a}_{<i}) = \max_{\mathbf{a}_{i+1:j}} \left\{u_{i\to j}^{\max}(\mathbf{a}_{<i}, \mathbf{a}_{i+1:j}) - u_{i\to j}^{\min}(\mathbf{a}_{<i}, \mathbf{a}_{i+1:j})\right\} \tag{65}$$

$$= \max_{\mathbf{a}_{i+1:j}} \max_{a_i, a_i'} |u_j(\mathbf{a}_{<i}, a_i, \mathbf{a}_{i+1:j}) - u_j(\mathbf{a}_{<i}, a_i', \mathbf{a}_{i+1:j})| \tag{66}$$

we have that $u^{\max}_{i\to j}(\mathbf{a}_{<i}, \mathbf{a}_{i+1:j}) \leq u^{\min}_{i\to j}(\mathbf{a}_{<i}, \mathbf{a}_{i+1:j}) + \Delta_{i\to j}(\mathbf{a}_{<i})$, and therefore,

$$r_i(\mathbf{a}_{<i}) \leq \sum_{j>i} \Delta_{i\to j}(\mathbf{a}_{<i}) \tag{67}$$

which gives a bound directly in terms of the "lookahead" variations $\Delta_{i\to j}(\mathbf{a}_{<i})$ in the potentials. In turn, these can be used in Equations (60) and (61) to bound the global KL divergence directly.

As an example, consider a potential of the simple form:

$$U(\mathbf{a}) = \sum_i u_i(a_i) \tag{68}$$

a particular case of Eq. (39) where the contribution to the potential of each token can be made independent of previous tokens. By substituting (68) into (51), one sees immediately that $\mathcal{Z}_i^{\text{future}}(a_i, \mathbf{a}_{<i})$ becomes independent of a_i. Indeed, $\Delta_{i\to j}(\mathbf{a}_{<i}) \equiv 0$ in this simple example. The autoregressive strategy then matches the target Q^* perfectly.

However, one could also reorganize the potential decomposition (68) using the telescopic gauge invariance (41), bringing it to the form (40). In this case, all the lookahead terms exhibit dependences on the current token a_i, which weakens the bounds (60) and results in a much larger global divergence (53). This example shows that, besides the definition of the potential itself, the way in which the decomposition (39) is constructed also plays a crucial role in the effectiveness of the resulting sampling strategy.

In the framework of ILMC that we have introduced here, the generative potential $E(\mathbf{a})$ and its prefix decomposition are fixed by the autoregressive form of the base generative model. However, we can still reduce dependences of the lookahead term on the next token through our choice of steering potential. The main lesson from this analysis is that the best choice, whenever possible, is one where the dependence of the potential S on the token a_i is concentrated on the term $s_i(\mathbf{a}_{\leq i})$ of the decomposition, while future contributions $s_j(\mathbf{a}_{\leq j})$ for $j > i$ have as small a dependence as possible on previous tokens a_i. This has been the guiding principle behind the steering potentials we have employed in the computational experiments of this work.

G. Computational complexity of ILMC

Here we derive the asymptotic computational complexity of the ILMC algorithm with respect to the total sequence length L. We assume that the primary computational bottleneck is the evaluation of the generative energy $E_t(\mathbf{a}_{\leq t})$, which requires a forward pass of the Transformer model. The cost of computing the steering potential $S_t(\mathbf{a}_{\leq t})$ is assumed to be negligible in comparison.

In a standard Transformer-based autoregressive model with key-value caching, generating the t-th token requires computing attention over the previous $t-1$ tokens. The computational cost for a single forward pass at step t is therefore proportional to t. The total cost to generate a sequence of length L is:

$$\mathcal{C}_{\text{gen}} \propto \sum_{t=1}^{L} t \approx \frac{1}{2}L^2 \implies O(L^2) \tag{69}$$

In our proposed method, the sequence grows in blocks of size B. Let $M = L/B$ be the total number of elongation steps. We index these steps by $n = 1, \ldots, M$. At step n, the current sequence length is $t_n = n \cdot B$.

The computational cost at step n, denoted $C(n)$, consists of two components: the elongation phase, where B new tokens are generated, and the refinement phase, which involves performing N_{MC} resampling steps over a window of size K.

For the refinement step at length t_n, we perform N_{MC} iterations. In each iteration, we resample a suffix of length roughly proportional to K (on average $K/2$). Since the current sequence length is t_n, the cost to regenerate these tokens is proportional to the context length multiplied by the number of tokens generated: $K \cdot t_n$. Thus, the cost of one full refinement phase at step n is:

$$\mathcal{C}_{\text{ref}}(n) \propto N_{\text{MC}} \cdot K \cdot t_n \tag{70}$$

The total cost $\mathcal{C}_{\text{total}}$ is the sum over all elongation steps n:

$$\begin{aligned}\mathcal{C}_{\text{total}} &\approx \sum_{n=1}^{L/B} (\mathcal{C}_{\text{elong}}(n) + \mathcal{C}_{\text{ref}}(n)) \\ &\approx \sum_{n=1}^{L/B} N_{\text{MC}} K(nB) \quad \text{(Dominant term)}\end{aligned} \tag{71}$$

Factoring out constants and substituting the sum of integers $\sum_{n=1}^{M} n \approx \frac{1}{2}M^2$:

$$\mathcal{C}_{\text{total}} \propto N_{\text{MC}} K B \sum_{n=1}^{L/B} n \approx N_{\text{MC}} K B \frac{1}{2} \left(\frac{L}{B}\right)^2 \tag{72}$$

Simplifying this expression yields:

$$\mathcal{C}_{total} \propto \frac{N_{\text{MC}} K}{2B} L^2 \tag{73}$$

From the derived expression, we identify two distinct scaling regimes based on the choice of the lookback window K.

- First, in the case of non-extensive lookback ($K = \text{const}$), if K is a fixed hyperparameter independent of the sequence length, then K acts as a constant prefactor. The scaling becomes:

$$\mathcal{C}_{\text{total}} \propto O(L^2) \tag{74}$$

  This matches the complexity class of standard autoregressive sampling.

- Second, in the case of extensive lookback ($K \propto L$), if the lookback window grows with the sequence length (e.g., $K = L$ to ensure global optimality as per Proposition 1), we substitute $K \propto L$ into the equation:

$$\mathcal{C}_{\text{total}} \propto L \cdot L^2 \implies O(L^3) \tag{75}$$

  In this regime, the algorithm incurs a cubic computational cost.

H. Model Architecture and Training

To validate ILMC, we used two autoregressive Transformer architectures on the Chorismate Mutase (CM) and Phage Lysozyme (PL) families. In addition, for the classifier-guided steering experiments on the Response Regulator (RR) family, we trained an additional sDO.

The first architecture is ProGen3 (Bhatnagar et al., 2025), a family of eight autoregressive pLMs that employ a sparse mixture of experts (MoE) transformer decoder architecture. These models range in scale from 112M to 46B parameters and were pre-trained on a massive corpus of 3.4 billion protein sequences. For our experiments, we fine-tuned the smallest ProGen3 version (ProGen3-112M) on the Chorismate Mutase and Phage Lysozyme families, as we found it to be already complex enough to correctly model the related sequence landscapes. This version of the model features 10 MoE layers with a latent dimension of 384, 6 attention heads, 8 experts per layer, RMSNorm for layer normalization, SiLU activation function, and rotary positional embedding with a context length up to 65,536 tokens. For both families, we reserved a random subset of 1000 sequences as a validation set, keeping 17,094 training sequences for Chorismate Mutase and 10,859 training sequences for Phage Lysozyme, and selected the same set of parameters for fine-tuning, without further optimization. In particular, we set the learning rate to $5 \cdot 10^{-4}$, the batch size to 512, and trained the model until minimization of the validation loss, evaluated at the end of each epoch, with an early stopping patience of 3 epochs. We minimized the same pre-training loss function, obtained as the sum of the autoregressive loss (with weight 1.00) and the MoE load balancing loss (with weight 0.05). The procedure resulted in 5 fine-tuning epochs (170 steps) for Chorismate Mutase and 6 (132 steps) for Phage Lysozyme.

The second architecture is a custom, lightweight Decoder-only Transformer (sDO) trained from scratch using the exact same training and validation splits described above (17,094 training sequences for CM, 10,859 for PL and 855,395 aligned

sequences for RR). This model comprises approximately 0.8 million parameters and features a context size sufficient to cover the full length of the sequences, which were short enough to avoid truncation. The architecture consists of 16 attention blocks with an embedding dimension of 64. Each block employs a Multi-Head Attention mechanism with 16 heads ($d_{head} = 4$). The sDO utilizes Rotary Positional Embeddings (RoPE), applies Layer Normalization before the attention and MLP blocks, and uses the GeLU activation function. For both families, the sDO was trained with a batch size of 512 and a learning rate of 10^{-3} using the AdamW optimizer. We employed a validation loss minimization strategy, which resulted in convergence at iteration 7,500 for Chorismate Mutase, iteration 4,500 for Phage Lysozyme and iteration 12'500 for Response Regulator.

For both architectures, the training sequences were curated by using Hidden Markov Models (HMM) to scan the UniProt database (The UniProt Consortium, 2025). For the Chorismate Mutase family, we constructed an HMM based on natural data from (Russ et al., 2020), while for Phage Lysozyme and Response Regulator, we utilized the standard Pfam HMM profile (Mistry et al., 2021). All entries containing more than 15% gaps in the resulting Infernal (Nawrocki & Eddy, 2013) alignment were removed, and the resulting datasets were used in their unaligned form.

I. Melting Temperature Prediction

Predicted melting temperatures (T_m) were estimated using TemStaPro (Pudžiuvelytė et al., 2024), a tool based on ProtT5-XL (Elnaggar et al., 2022) embeddings that classifies sequence stability across thresholds from $40°\mathrm{C}$ to $65°\mathrm{C}$. To compute the statistics reported in Table 1, we mapped the predicted stability bins to scalar values using the bin centers. Specifically, sequences predicted as unstable at $40°\mathrm{C}$ were assigned a value of $37.5°\mathrm{C}$, those stable at $65°\mathrm{C}$ were assigned $67.5°\mathrm{C}$, and all intermediate bins were assigned their arithmetic midpoint (e.g., stability in the $[40°\mathrm{C}, 45°\mathrm{C})$ range was assigned $42.5°\mathrm{C}$).

J. Steering potential for thermostability enhancement

The amino acid composition of proteins in thermophilic organisms differs systematically from that of mesophilic organisms. Specifically, thermophilic proteomes show a marked enrichment in charged residues (such as Glu, Arg, and Lys) and a corresponding depletion of polar uncharged residues (such as Gln and Asn). These compositional adaptations facilitate the formation of stabilizing interactions, such as salt bridges, and reduce the presence of residues prone to thermal deamidation (Zeldovich et al., 2007; Fukuchi & Nishikawa, 2001).

To construct our steering potential, we utilized the ratio of amino acid frequencies observed on the surfaces of thermophilic proteins relative to mesophilic proteins (Fukuchi & Nishikawa, 2001). We relied specifically on surface composition rather than whole-chain metrics because the protein interior is strictly constrained by structural packing requirements, whereas surface residues possess greater evolutionary freedom to adapt to environmental stressors. To center the potential such that a score of zero represents a neutral mesophilic background, we define the weight $w(a)$ for amino acid k as:

$$w(a) = \frac{f_{\text{thermo}}(a)}{f_{\text{meso}}(a)} - 1 \tag{76}$$

Positive values indicate residues associated with thermostability, while negative values indicate residues associated with mesophilic environments. The steering function $S(\mathbf{a})$ for a sequence $\mathbf{a} = (a_1, \ldots, a_L)$ is defined as the negative sum of these weights:

$$S^{\mathrm{Tm}}(\mathbf{a}) = -\sum_{i=1}^{L} w(a_i) \tag{77}$$

The complete set of weights used in our experiments is provided in Table 6.

Figure 12 illustrates the scatter plot of predicted melting temperatures against the defined thermophilic steering score for the natural sequences. While the correlation is robust across the generated designs, it appears less pronounced in the natural sequences. This is primarily because the natural sequences for Chorismate Mutase and Phage Lysozyme are predominantly mesophilic, clustering within the low melting temperature regime ($T_m < 40°\mathrm{C}$). Also, the natural data exhibits limited variability in the thermophilic amino acid scores relative to the broader landscape explored by our sampling method.

Table 6. Thermophilic propensity weights derived from surface amino acid ratios. Weights are calculated as $(\text{Ratio} - 1)$, centered at 0.

CATEGORY	AMINO ACID	RATIO	WEIGHT $w(a)$
STRONGLY ENRICHED (RATIO > 1.10)	GLU (E)	1.28	+0.28
	LYS (K)	1.27	+0.27
	TRP (W)	1.24	+0.24
	ARG (R)	1.23	+0.23
	TYR (Y)	1.22	+0.22
	ILE (I)	1.21	+0.21
	PHE (F)	1.21	+0.21
	LEU (L)	1.14	+0.14
	VAL (V)	1.13	+0.13
	PRO (P)	1.13	+0.13
NEUTRAL (0.90 − 1.10)	GLY (G)	1.04	+0.04
	CYS (C)	0.94	-0.06
	MET (M)	0.91	-0.09
STRONGLY DEPLETED (RATIO < 0.90)	ASP (D)	0.82	-0.18
	SER (S)	0.79	-0.21
	ASN (N)	0.71	-0.29
	THR (T)	0.71	-0.29
	ALA (A)	0.66	-0.34
	HIS (H)	0.62	-0.38
	GLN (Q)	0.55	-0.45

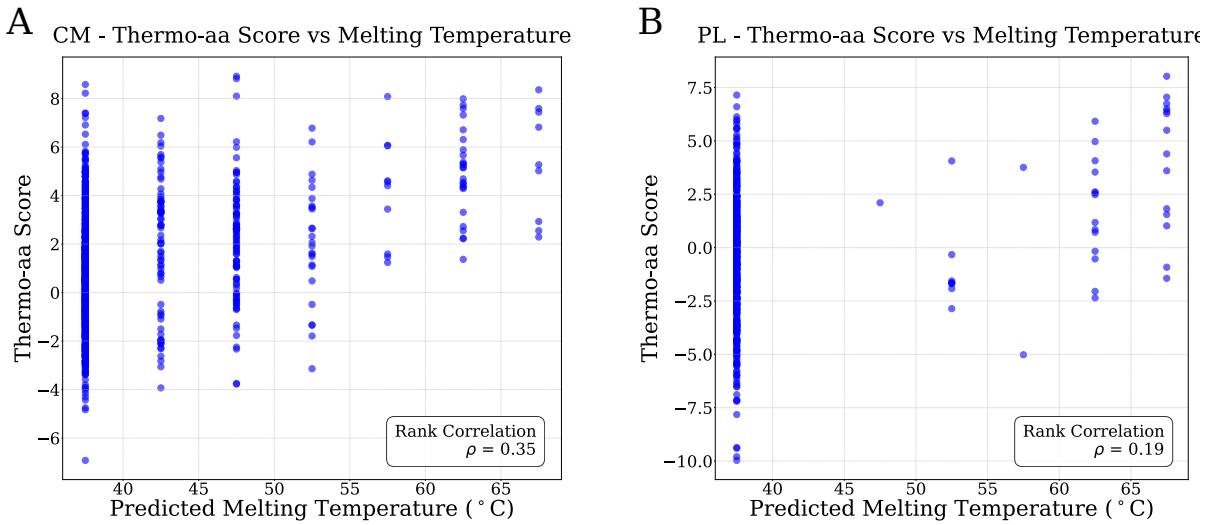

Figure 12. **Correlation between thermophilic amino acid score $-S^{\text{Tm}}(a)$ and predicted melting temperature.** Scatter plots showing the relationship between the steering score (Thermo-aa Score) and predicted melting temperature (T_m) for 1024 natural sequences. **(A)** Chorismate Mutase (CM) family (Rank Correlation $\rho = 0.35$). **(B)** Phage Lysozyme (PL) family (Rank Correlation $\rho = 0.19$). The observed correlations are limited primarily because the natural sequences are predominantly mesophilic, clustering within the lowest temperature bin ($T_m < 40°$C).

K. Steering potential for sampling in the vicinity of a wild-type sequence

The necessity of the modified definition for the steering potential $S_t^{\mathrm{WT}}(\mathbf{a}_{\leq t})$ arises from the inability of the standard edit distance to provide a meaningful steering signal during the early stages of autoregressive generation.

Consider the standard edit distance $\mathrm{EditDistance}(\mathbf{x}, \mathbf{y})$. If we were to naively use this as our partial steering potential $S_t^{\mathrm{WT}}(\mathbf{a}_{\leq t}) = \mathrm{EditDistance}(\mathbf{a}_{\leq t}, \mathbf{w})$ against a target wild-type $\mathbf{w}$, the steering signal $S_t^{\mathrm{WT}}(\mathbf{a}_{\leq t})$ would remain practically constant for any prefix length $t \ll L_w$. This occurs because, in most cases, the edit distance is entirely determined by the length difference between the target $\mathbf{w}$ and the partial generation $\mathbf{a}_{\leq t}$, rendering the specific identity of the generated tokens irrelevant.

This failure is best illustrated through an example. Let the target protein be:

$$\mathbf{w} = \mathrm{KKKKKKDNRCQA} \tag{78}$$

Assume the model has correctly generated the prefix $\mathbf{a}_{<t} = \mathrm{KKK}$. We now consider the emission of the next token a_t:

- Scenario A (Correct Continuation): We emit $a_t = \mathrm{K}$, forming the prefix KKKK.
- Scenario B (Incorrect Continuation): We emit $a_t = \mathrm{A}$, forming the prefix KKKA.

Intuitively, Scenario A should be highly preferred as it matches the local sequence context. However, under the standard edit distance against the full target $\mathbf{w}$, both scenarios yield an identical distance score:

$$\begin{aligned} &\mathrm{EditDistance}(\mathrm{KKKK}, \mathbf{w}) = 9 \quad \text{(9 deletions to match)} \\ &\mathrm{EditDistance}(\mathrm{KKKA}, \mathbf{w}) = 9 \quad \text{(Match final 'A', delete intermediate)} \end{aligned} \tag{79}$$

In Scenario B, the algorithm matches the newly emitted 'A' to the final 'A' of the target string $\mathbf{w}$, reaching the exact same distance because the cost of deleting the ending portion of the target is identical to the cost of deleting internal segments. Practically, for a prefix $\mathbf{a}_{\leq t}$ where $t \ll L_w$, the distance is determined by the number of deletions required to account for the length difference, since almost any amino acid generated at step t is likely to appear somewhere later in a long target chain. Consequently, the sampler receives no signal indicating whether the generated token helps in steering the generation toward the target sequence.

Our proposed definition, $S_t^{\mathrm{WT}}(\mathbf{a}_{\leq t}) = \min_k \mathrm{EditDistance}(\mathbf{a}_{\leq t}, \mathbf{w}_{\leq k})$, resolves this problem by effectively eliminating the penalty for deleting the ending part of the target:

- For KKKK, we compare against all prefixes of $\mathbf{w}$. The best match is $\mathbf{w}_{\leq 4}$ ("KKKK"), yielding a distance of 0.
- For KKKA, we compare against all prefixes. The best match (not unique) is against $\mathbf{w}_{\leq 4}$ ("KKKK"), where 'A' results in a mismatch, yielding a distance of 1.

This metric successfully creates a steering signal that can accurately inform the sampling regarding the similarity with the target sequence.

L. Acceptance Rate Decay

The original sampling framework proposed by (Karan & Du, 2025) for text-based architectures is designed to sample from the exponentiated distribution of the model itself, $P(\mathbf{a})^\beta$. In that unsteered scenario, the proposal distribution—standard autoregressive elongation at inverse temperature β—is naturally aligned with the target distribution. Consequently, the acceptance rates remain relatively high even for larger lookback windows, as the proposal provides a good approximation of the target.

The introduction of an external steering potential $S(\mathbf{a})$ creates a discrepancy between the proposal distribution and the target distribution. In ILMC framework, the proposal steps (Eq. 4) are generated via the base model, which is agnostic to the steering potential $S(\mathbf{a})$. The steering constraint is enforced solely through the acceptance step of the Metropolis-Hastings transition:

$$A(\mathbf{a} \to \mathbf{a}') = \min\left(1, e^{-\beta \Delta E - \lambda \Delta S} \frac{q(\mathbf{a})}{q(\mathbf{a}')}\right) \tag{80}$$

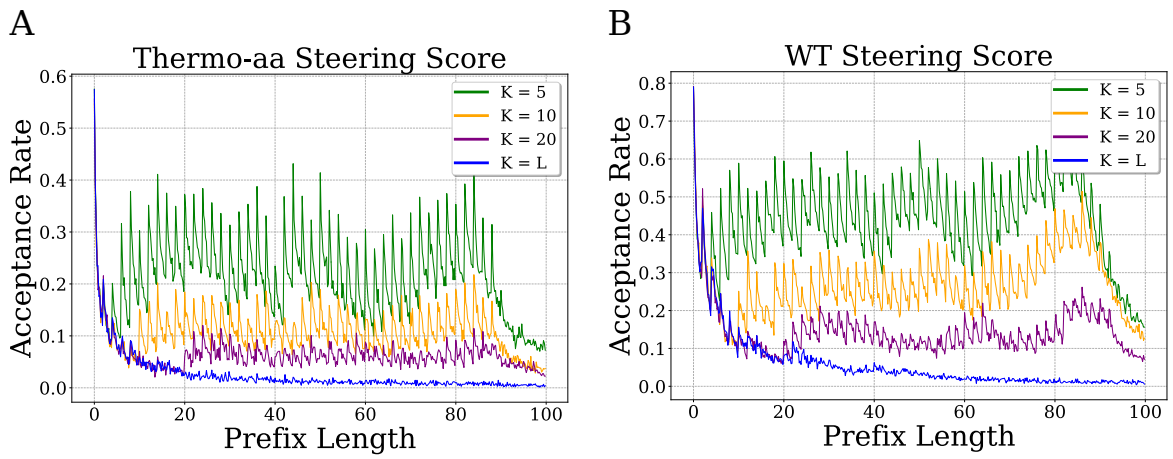

Figure 13. **Metropolis acceptance rate as a function of the generated prefix length** during ILMC for the sDO architecture on the CM family ($B = 2$, $N_{\mathrm{MC}} = 10$, averaged over 1024 sequences). For both steering with S^{Tm} (A) and S^{WT} (B), we observe a sharp decline in acceptance rate as the lookback window K increases. This indicates that while larger K brings the distribution closer to sampling from Q^*, it renders the sampling procedure significantly less efficient.

where $\Delta E = E_t(\mathbf{a}'_{\leq t}) - E_t(\mathbf{a}_{\leq t})$ and $\Delta S = S_t(\mathbf{a}'_{\leq t}) - S_t(\mathbf{a}_{\leq t})$. When the lookback window K is large, the proposal mechanism effectively generates a long sequence segment "blindly," without conditioning on the steering objective. The probability that such an unguided trajectory spontaneously satisfies the target steering constraints decays with the length of the segment. Consequently, newly proposed states frequently result in unfavorable steering shifts (large ΔS), causing the term $e^{-\lambda \Delta S}$ to go to 0, leading to high rejection rates.

To mitigate this issue and ensure computational efficiency, it is practically necessary to restrict the size of the lookback window K. By using a shorter window, the proposal trajectory is forced to remain in the vicinity of the existing prefix, which has already been steered into a favorable region of the landscape, thereby increasing the probability of acceptance. We observe empirically that the acceptance rate is inversely related to K (Figure 13). To conclude, while $K = L$ is theoretically required to sample exactly from Q^*, a restricted K is practically required to maintain a feasible acceptance rate when the steering strength $\lambda > 0$.

M. Pareto Frontier of Generative Quality and Steering Potential

When sampling from the target distribution $Q^*(\mathbf{a}) \propto \exp(-\beta E(\mathbf{a}) - \lambda^{\mathrm{Tm}} S^{\mathrm{Tm}}(\mathbf{a}))$, the objective is to identify sequences that simultaneously minimize the generative energy $E(\mathbf{a})$, ensuring biological viability, and the steering potential $S^{\mathrm{Tm}}(\mathbf{a})$, reflecting the desired trait. However, because the steering potential is not perfectly aligned with the natural sequence distribution learned by the base model, these two objectives are likely to compete. Optimizing for the steering potential beyond a certain threshold requires the sampler to explore regions of the sequence space that the model considers less probable, thereby increasing the generative energy. This competition defines a Pareto frontier in the (E, S) plane, representing the set of optimal trade-offs where one metric cannot be improved without degrading the other. To empirically illustrate this relationship, we performed a systematic analysis with the sDO base model, using the thermophilic amino acid score Eq. (8) as the steering potential by varying the inverse temperature β and the steering coefficient λ^{Tm}. As shown in Figure 14, the resulting samples clearly delineate a frontier; it is not possible to achieve high steering scores without sacrificing generative quality, and vice versa.

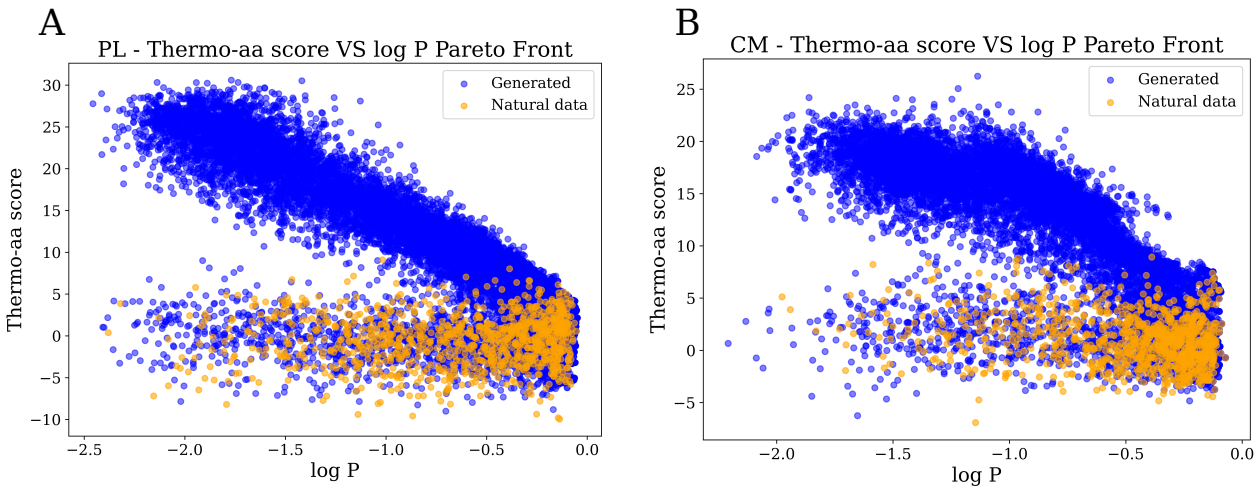

Figure 14. The Pareto frontier between the mean per-token log-probability ($\log P$) and the thermophilic amino acid score $-S$ for the sDO models trained on Phage Lysozyme (PL) and Chorismate Mutase (CM). For each protein family, libraries of 1024 sequences were sampled utilizing ILMC with fixed parameters $B = 2, K = 5, N_{\mathrm{MC}} = 10$, across a dense grid of inverse temperature β and steering strength λ^{Tm}. For CM, the grid parameters were $\beta \in \{1, 1.8, 2.6\}$ and $\lambda^{\mathrm{Tm}} \in \{0, 5, 10, 15, 20\}$; for PL, $\beta \in \{1, 1.3, 2\}$ and $\lambda^{\mathrm{Tm}} \in \{0, 5, 10, 15, 20\}$. Natural sequences are indicated in orange for reference. The plots clearly demonstrate the Pareto frontier in the upper-right corner, illustrating the intrinsic trade-off between maximizing generative likelihood and enforcing the targeted steering potential.

N. Double Steering

Another interesting application of ILMC is the simultaneous imposition of multiple steering potentials. For instance, we could aim to enhance thermostability while strictly maintaining the generation within the mutational vicinity of a known wild-type. Mathematically, the extension of our method to two objectives is straightforward. The target distribution becomes:

$$Q^*(\mathbf{a}) \propto \exp\left(-\beta E(\mathbf{a}) - \lambda^{\mathrm{WT}} S^{\mathrm{WT}}(\mathbf{a}_{\leq L}) - \lambda^{\mathrm{Tm}} S^{\mathrm{Tm}}(\mathbf{a})\right) \tag{81}$$

To demonstrate the efficacy of this approach, we targeted a specific CM wild-type sequence (Index 2 in the dataset), chosen for its low predicted melting temperature ($T_m < 40$). Our objective is to generate variants possessing between 14 and 16 mutations relative to this wild-type, while simultaneously pushing for higher thermostability.

We compared two generation strategies:

1. Distance-Only Steering: We sampled using parameters tuned to target the 14-16 mutation range without any thermo-aa composition bias ($\beta = 1.0, \lambda^{\mathrm{WT}} = 7.5, \lambda^{\mathrm{Tm}} = 0$).
2. Double Steering: We sampled using parameters tuned to target the same mutational range but with thermo-aa composition bias ($\beta = 1.3, \lambda^{\mathrm{WT}} = 6.5, \lambda^{\mathrm{Tm}} = 6.5$).

From the resulting 2048 generated sequences, we selected the ones that met two criteria: (1) exactly 14 to 16 edit distance from the wild-type, and (2) a generative log-likelihood greater than the average likelihood of unsteered autoregressive samples $\log P(\mathbf{a}) > -0.8$. This ensures that we compare only high-quality, viable protein candidates.

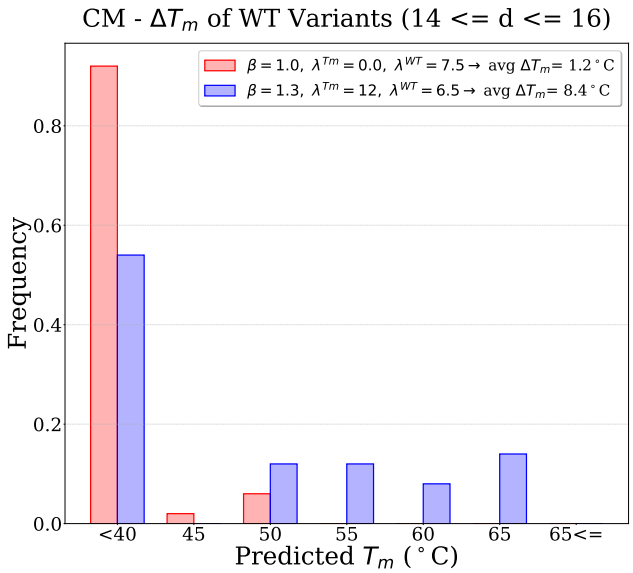

Figure 15. Distribution of predicted melting temperatures (T_m) for Chorismate Mutase (CM) variants generated within a 14–16 mutation shell relative to the target wild-type (WT $T_m = 37.5°$C). Sequences were sampled using the sDO architecture via ILMC ($B = 2, K = 10, N_{\mathrm{MC}} = 100$). We compare two strategies: "Distance Only" ($\beta = 1.0, \lambda^{\mathrm{WT}} = 7.5, \lambda^{\mathrm{Tm}} = 0$) and "Double Steering" ($\beta = 1.3, \lambda^{\mathrm{WT}} = 6.5, \lambda^{\mathrm{Tm}} = 6.5$), with all candidates filtered for high likelihood ($\log P > -0.8$). The figure displays the top 50 sequences ranked by thermophilic amino acid score $-S^{\mathrm{Tm}}(\mathbf{a})$ from total populations of 185 (Distance Only) and 155 (Double Steering) variants, demonstrating that double steering successfully shifts the distribution toward higher T_m while maintaining the target edit distance.

As illustrated in Figure 15, the multi-objective approach proves effective: the average ΔT_m increases from $1.2°$C in the distance-only baseline to $8.4°$C) in the double-steered case, demonstrating that ILMC can effectively navigate the intersection of multiple constraints to design stable, high-likelihood mutations.

The practical challenge lies in the hyperparameter search. Balancing the inverse temperature β, the distance penalty λ^{WT}, and the thermo-bias λ^{Tm} requires a non-trivial search to ensure the sampler explores the correct intersection of the energy landscape (high likelihood, specific distance, high stability).

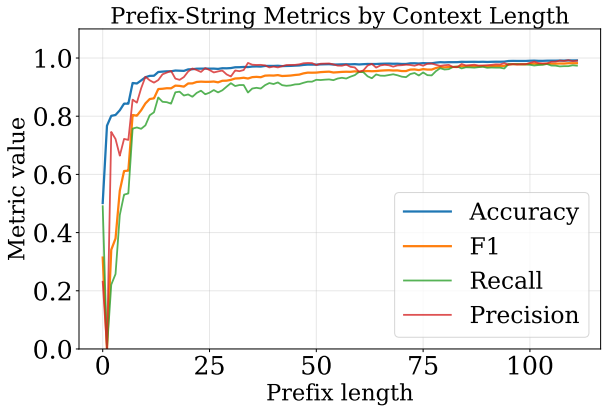

Figure 16. **Any-prefix classifier metrics versus prefix length** for the Response Regulator benchmark. Test-time accuracy, precision, recall, and F1 are reported as a function of the available prefix length.

O. Any-prefix classifier for Response Regulators

For classifier-guided steering on the Response Regulator (RR) family, we trained a lightweight any-prefix classifier to predict the target GerE homodimerization class from incomplete sequence prefixes. The classifier was trained on the same aligned RR dataset used for the generation experiments, comprising 855,395 sequences of length 111, of which 23.26% belong to the positive class.

The model receives as input a prefix $\mathbf{a}_{\leq t}$, padded to length 111, and outputs a binary logit. Architecturally, it consists of a token embedding layer with vocabulary size 21 and embedding dimension 16, followed by flattening of the full embedded prefix, a fully connected layer from 1776 to 16 units, a ReLU activation, and a final linear layer from 16 to 1. The total number of trainable parameters is 28,785.

We trained the classifier with weighted binary cross-entropy to account for class imbalance, sampling prefix lengths uniformly during training so that the model learned to score arbitrary contexts rather than only full sequences. Training required approximately 5 minutes on a single NVIDIA RTX 4090 GPU.

Classifier performance was then evaluated as a function of prefix length. As expected, very short prefixes are only weakly informative, whereas predictive performance improves steadily as longer prefixes are observed. The complete dependence of the evaluation metrics on prefix length is reported in Fig. 16.

P. Hyperparameter Selection Guide

A few practical heuristics can be used to initialize the ILMC hyperparameters $(\beta, \lambda, K, N_{\mathrm{MC}})$.

Selection of β: In practice, we first tune β in the unsteered setting ($\lambda = 0$) so that the generated sequences match a desired mean per-token $\log P$. A useful reference point can be obtained from the distribution of $\log P$ on natural sequences; in particular, the empirical mean over the natural data provides a convenient starting target.

Selection of λ: A useful rule of thumb for choosing λ is to set it such that, for typical local proposals, the steering contribution is of the same order as the generative contribution $\beta\Delta E \sim \lambda\Delta S$. This prevents either term from dominating the Metropolis acceptance ratio.

Selection of K: In our experiments, moderate lookback windows were usually sufficient, and values of K between 5 and 10 already yielded strong improvements over purely autoregressive sampling. Larger values of K can be useful when maximizing diversity is especially important, but in practice they often reduce the acceptance probability substantially and therefore typically require larger N_{MC}, especially in the steered setting.

Selection of N_{MC}: The number of refinement steps N_{MC} should then be increased until the chain performs a non-trivial number of accepted corrections during generation. Empirically, we found it useful to tune N_{MC} until acceptance rates during elongation were on the order of ~ 0.3. It is also useful to monitor the acceptance rate as a function of the refinement

step: if it still fluctuates strongly across the N_{MC} updates, this suggests that a larger value may be needed to approach a local equilibrium over the current prefix. Different steering potentials require different values of N_{MC} depending on their ruggedness. For the thermostability steering potential, relatively small values of N_{MC} were sufficient, because the score is additive and each token swap produces a small graded variation in S_t^{Tm}. By contrast, the wild-type proximity objective required substantially larger N_{MC}, since most local proposals only weakly affect the distance-based score and many accepted moves are needed to accumulate a significant shift toward the target sequence.

Overall, we found the following initialization strategy to work well in practice: tune β to match the target mean $\log P$, set λ so that $\lambda \Delta S \sim \beta \Delta E$ for typical mutations, use $K \in [5, 10]$, and increase N_{MC} until acceptance rates during generation are around 0.3.

Because ILMC operates sequentially, hyperparameter search can be accelerated by working on partial sequence prefixes of about 20–30 tokens. In practice, one can generate only the first 20–30 residues and monitor acceptance rate, mean $\log P$, and steering score; this already provides a useful proxy for the behavior of the full sampler while significantly reducing computational cost.