

# AMix-1: A Pathway to Test-Time Scalable Protein Foundation Model

<sup>1</sup>Shanghai Artificial Intelligence Laboratory

<sup>2</sup>Generative Symbolic Intelligence Lab (GenSI), Tsinghua University

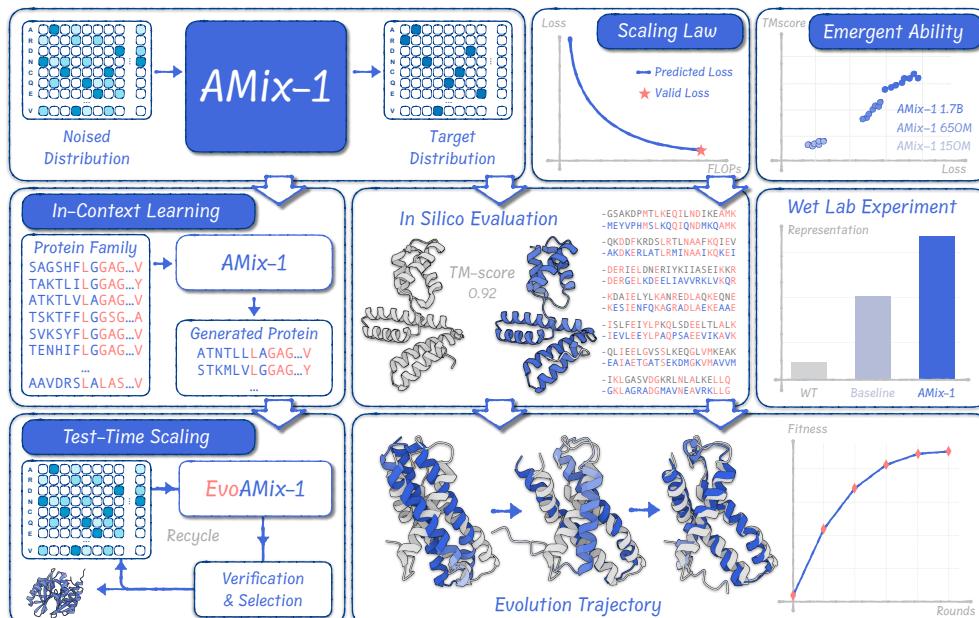
<sup>3</sup>Institute for AI Industry Research (AIR), Tsinghua University

## Abstract

We introduce AMIX-1, a powerful protein foundation model built on Bayesian Flow Networks and empowered by a systematic training methodology, encompassing pretraining scaling laws, emergent capability analysis, in-context learning mechanism, and test-time scaling algorithm. To guarantee robust scalability, we establish a predictive scaling law and reveal the progressive emergence of structural understanding via loss perspective, culminating in a strong 1.7-billion model. Building on this foundation, we devise a multiple sequence alignment (MSA)-based in-context learning strategy to unify protein design into a general framework, where AMIX-1 recognizes deep evolutionary signals among MSAs and consistently generates structurally and functionally coherent proteins. This framework enables the successful design of a dramatically improved AmeR variant with an up to 50 $\times$  activity increase over its wild type. Pushing the boundaries of protein engineering, we further empower AMIX-1 with an evolutionary test-time scaling algorithm for *in silico* directed evolution that delivers substantial, scalable performance gains as verification budgets are intensified, laying the groundwork for next-generation lab-in-the-loop protein design.

**Correspondence:** zhouchao@air.tsinghua.edu.cn

**Project Page:** <https://gensi-thuair.github.io/AMix-1/>



**Figure 1** An overview of AMIX-1. AMIX-1 is built upon a pathway methodology of scaling law, emergent ability, in-context learning, and test-time scaling, and is validated on both *in silico* evaluation and wet-lab assay.

## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Protein Modeling with Bayesian Flow Network</b>	<b>3</b>
<b>3</b>	<b>Scaling Law</b>	<b>5</b>
3.1	Pretraining	5
3.2	Predictive Scaling Behavior	5
3.3	Optimal Estimation under Fixed FLOPs	7
<b>4</b>	<b>Emergent Ability</b>	<b>9</b>
4.1	Experimental Setup	9
4.2	Findings	10
<b>5</b>	<b>In-Context Learning</b>	<b>11</b>
5.1	AMIX-1 is a Unified Protein Designer	12
5.2	In Silico: AMIX-1 Excels at In-Context Learning	13
5.3	Wet Lab: AMIX-1 Designs High-Activity Proteins	15
<b>6</b>	<b>Test-Time Scaling</b>	<b>15</b>
6.1	Scaling AMIX-1 with Evolutionary Algorithm	16
6.2	A Proposal Distribution Perspective	16
6.3	AMIX-1 Exhibits Robust Test-Time Scaling across Diverse Tasks	17
<b>7</b>	<b>Conclusion</b>	<b>19</b>
<b>A</b>	<b>Pretrainig Algorithm</b>	<b>28</b>
A.1	Training	28
A.2	Inference	28
<b>B</b>	<b>Performance of Prediction Tasks</b>	<b>29</b>
<b>C</b>	<b>Details of CAMEO-20</b>	<b>29</b>
<b>D</b>	<b>Further Discussion on Scaling Laws</b>	<b>29</b>
D.1	Unscalability when Extreme Noise Levels	29
D.2	Mean Relative Error	29
D.3	Parametric Scaling Laws under Several Noise Levels	30
<b>E</b>	<b>Experimental Details of EvoAMix-1</b>	<b>30</b>
<b>F</b>	<b>EvoAMix-1 Ablation Study</b>	<b>31</b>
F.1	Assessing the Role of the Generative Model in Test-Time Scaling	31
F.2	Assessing the Impact of Iterative Condition Refinement	33
<b>G</b>	<b>Verification in wet-lab</b>	<b>33</b>

## 1 Introduction

Advanced artificial intelligence models, such as AlphaFold [1–3] and ESM [4–6], have significantly facilitated a wide range of protein research, including structure prediction [1–3, 7–9], inverse folding [10], functional property prediction [11, 12], mutational effect estimation [13–15], and *de novo* protein design [6, 16].

However, these models lack a unified, scalable, and systematic methodology akin to that of cutting-edge large language models (LLMs) [17–23], whose capabilities consistently improve with increased scale in data, model size, and compute. Consequently, the systematic trading of compute for performance has revealed unprecedented emergent intelligence, as models continue to improve with larger computational investments, both during pretraining [24–26] and at inference [27–29]. This observation motivates a fundamental question:

As large language models begin to exhibit traits of artificial general intelligence, how can we craft protein foundation models under a converging methodology grounded in **scaling law** characterization, **emergent capability** analysis, **in-context learning** mechanism, and **test-time scaling** algorithms, to unlock scalable and universal protein design of high fidelity?

In this study, we propose AMIX-1, a pathway methodology for scalable protein design (Figure 1). We first characterize a predictable scaling law for Bayesian Flow Networks [30] and reveal that structural understanding emerges naturally as training progresses, with which we pretrained a powerful 1.7B AMIX-1. Then we leverage its in-context learning capability with multiple sequence alignments to unify structure- and function-guided protein design into a general framework. With AMIX-1, we successfully design a novel AmeR variant validated through wet-lab experiments, achieving a 50× activity increase over the wild type. To push the limits further, we fledge AMIX-1 with an evolutionary test-time scaling (TTS) algorithm for *in silico* directed evolution, allowing plug-and-play metrics and delivering strong performance gains with increasing verification budgets.

We summarize the main contributions of our study as follows:

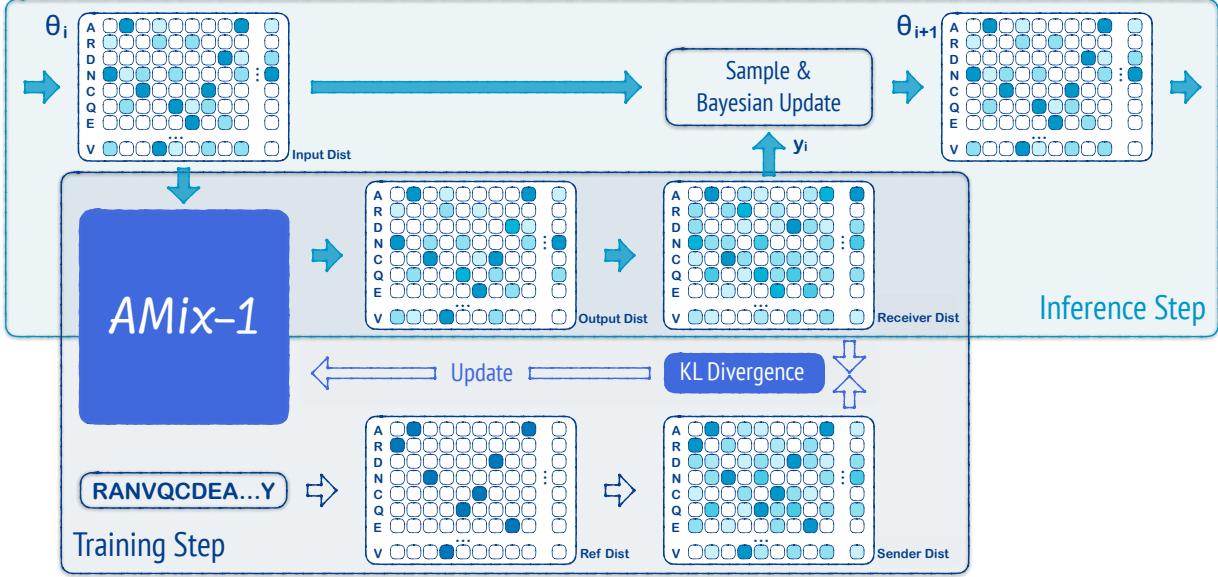
- We propose a systematic paradigm, including scaling laws, structural emergent ability, in-context learning, and test-time scaling, for crafting protein foundation models, emphasizing a road map towards scalable protein design;
- We are the first to characterize scaling law and emergent ability for an advanced generative model, i.e., Bayesian Flow Network, and pretrain a family of powerful models, AMIX-1, scaling from 8 million to 1.7 billion parameters;
- We introduce a novel evolutionary test-time scaling algorithm by leveraging AMIX-1’s in-context learning for unified protein design, amplifying its effectiveness in *in silico* directed evolution, delivering rapid gains as verification budgets grow;
- We design a high-activity AmeR variant with a 50× improvement over the wild type, showcasing AMIX-1’s real-world impact on functional protein engineering.

## 2 Protein Modeling with Bayesian Flow Network

Protein modeling with bayesian flow networks (BFNs) [30] introduces a generative framework that learns the continuous parameters of protein sequence distributions through iterative Bayesian updates, rather than directly modeling discrete amino acid sequences. This process iteratively refines a set of noisy samples initiated from an uninformative prior  $\theta_0$ , producing successive posteriors  $\theta_i$  that gradually incorporate more information and increase certainty.

The generative process is modeled as a message-passing protocol between **sender distribution** and **receiver distribution**, where the sender’s visibility is confined to the sample space, while the receiver infers messages based on its interpretation of samples and parameters.

During each communication round, the **sender distribution**  $p_s(y | x; \alpha)$  perturbs protein sequences  $x$  with a noise scheduler  $\alpha$  and sends the noisy protein sequences  $y$  to the receiver distribution, which mirrors a



**Figure 2** Illustration of Bayesian Flow Networks (BFNs). The network  $\phi$  takes as input the distribution  $p_{in}$  and generates the output distribution  $p_{out}$ . The training objective is to minimize the Kullback-Leibler (KL) divergence between the sender distribution  $p_s$  and the receiver distribution  $p_r$ , both of which are perturbed by additive noise  $\alpha_i$ .

forward diffusion mechanism. Specifically, for the protein sequence distribution, the sender distribution can be formulated as follows:

$$p_s(\mathbf{y} \mid \mathbf{x}; \alpha) = \mathcal{N}(\mathbf{y} \mid \alpha(K\mathbf{e}(\mathbf{x}) - \mathbf{1}), \alpha K \mathbf{I}) \quad (1)$$

where  $\mathbf{e}(\mathbf{x})$  is the one-hot encoding of the protein sequence  $\mathbf{x}$ , flattened into a vector, with each position encoded as a  $K$ -dimensional vector, and  $K$  is the number of amino acid types. Here  $\alpha$  is derived from the schedule  $\beta(t)$ , where  $\beta_1 \in R^+$  is the final noise level at  $t = 1$ :

$$\alpha = \beta(t) = \beta_1 t^2 \quad (2)$$

On the other hand, the **receiver distribution** employs a neural network  $\phi$  to approximate the sender distribution with its current belief  $\theta_{i-1}$ . Concretely, the network  $\phi$  takes the belief state  $\theta_{i-1}$  as input, and the corresponding distribution  $p_{in}(\cdot|\theta)$  is termed the **input distribution**. The network output  $\phi(\theta_{i-1})$  is an updated distribution  $p_{out}(\cdot|\phi(\theta_{i-1}))$  over the sample space, termed the **output distribution**. Integrating the output distribution with the noise scheduler  $\alpha$  yields the receiver distribution as follows:

$$p_r(\mathbf{y}_i \mid \theta_{i-1}, \phi, \alpha_i) = \mathbb{E}_{p_{out}(\mathbf{x}'|\phi(\theta_{i-1}))} p_s(\mathbf{y}_i \mid \mathbf{x}'; \alpha_i) \quad (3)$$

Therefore, the final training objective integrates the KL divergence between sender and receiver distributions across timesteps and trajectories, yielding the following loss:

$$\mathcal{L}(\phi) = \mathbb{E}_{\mathbf{x} \sim p_{data}} \mathbb{E}_{\prod_{i=1}^n p_s(\mathbf{y}_i \mid \mathbf{x}, \alpha_i)} D_{KL}(p_s(\mathbf{y}_i \mid \mathbf{x}, \alpha_i) \parallel p_r(\mathbf{y}_i \mid \theta_{i-1}, \phi, \alpha_i)), \quad (4)$$

Leveraging the Bayesian update function to model the information gathering process, which iteratively refines priors into data-informed posteriors as follows:

$$\theta_i = h(\theta_{i-1}, \mathbf{y}_i, \alpha_i) = \frac{e^{\mathbf{y}_i} \theta_{i-1}}{\sum_{k=1}^K e^{\mathbf{y}_k} (\theta_{i-1})_k} \quad (5)$$

In practice, we use the continuous categorical values of output distribution  $\phi(\theta_{i-1})$  without sampling to directly update  $\theta$  bypassing the sampling of noisy data needed for Bayesian update, which is similar to the noise reduced sampling method used in MolCRAFT [31]. The effective sampling method is described in appendix Algorithm 2. We also provide Figure 2 to help the readers visually grasp the key concepts of BFNs.

### 3 Scaling Law

This section begins by detailing the pretraining setup, followed by an investigation of the scaling laws that govern AMIX-1 models. These laws characterize the relationships among key factors, including cross-entropy loss, model size, the number of training tokens, and computational cost measured in FLOPs.

**Key Result ①:** AMIX-1’s cross entropy is precisely predictable against compute, data quantity and model size.

#### 3.1 Pretraining

**Datasets** The scaling law exploration in the pre-training stage is conducted using the **UniRef50** dataset, derived from UniProtKB [2] and selected UniParc sequences through iterative clustering (UniProtKB+UniParc → UniRef100 → UniRef90 → UniRef50) [32, 33]. This process ensures that UniRef50’s representative sequences are high-quality, non-redundant, and diverse, providing broad coverage of the protein sequence space for protein language models. Specifically, we utilized the pre-processed UniRef50 dataset provided by EvoDiff [34], comprising 41,546,293 sequences for training and 82,929 sequences for validation. Sequences longer than 1,024 residues were randomly cropped to 1,024 to reduce computational costs and generate diverse subsequences.

**Architecture** AMIX-1 model series are built upon the encoder-only Transformer architecture [35]. To encode positional information while preserving sequence invariance in attention patterns, we incorporate Rotary Position Embeddings (RoPE) [36], which introduce relative position dependencies through a rotation-based mechanism applied to query and key vectors. To explore the impact of model capacity on performance, we scale the architecture by systematically modifying the number of encoder layers, the number of attention heads, and the dimensionality of the hidden representations. These architectural configurations and corresponding parameter counts are summarized in [Table 1](#).

**Training Details** All models were trained using the Adamw [37] optimizer with a learning rate of  $4 \times 10^{-4}$ ,  $\alpha$  values of (0.9, 0.98), and a weight decay of 0.01. A polynomial learning rate schedule was adopted, spanning a total of 1 million training steps with a warmup phase of 2,000 steps. During warmup, the learning rate increased linearly from  $1 \times 10^{-7}$  to  $4 \times 10^{-4}$ , followed by a polynomial decay to  $4 \times 10^{-5}$  with a power factor of 1. Training was performed using mixed-precision (BF16) and a distributed data parallel (DDP) setup. To mitigate exploding gradients, gradient norms were clipped to a maximum of 1.0. Early stopping was applied based on validation loss, with a patience threshold of 1,000 steps. For AMIX-1-1.7B and AMIX-1-650M, training was conducted with a batch size of 4 million across 16 nodes, each equipped with 8 A800 GPUs. Smaller models were trained with a batch size of 1 million.

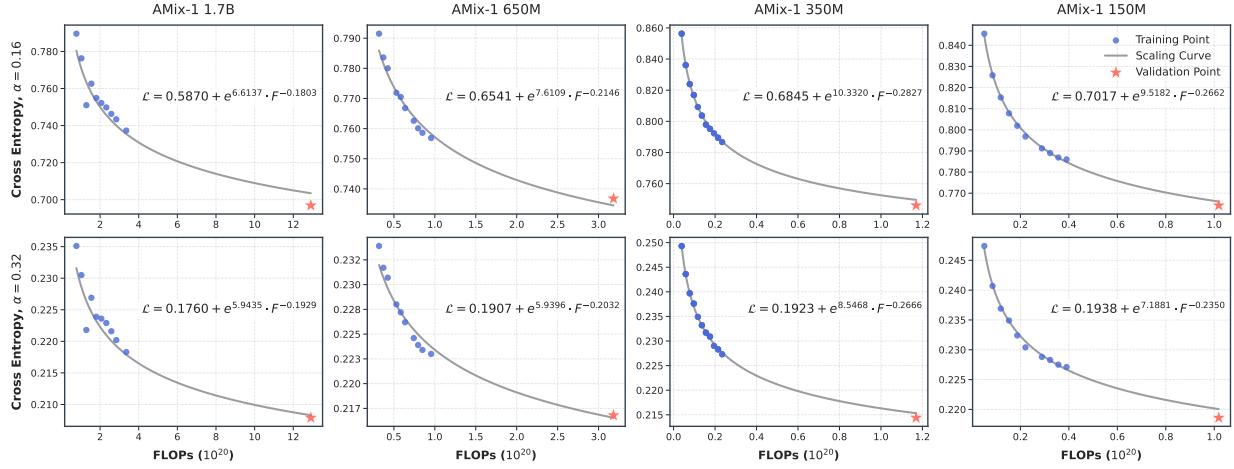
**Noise Level** In BFN models, the noise level  $\alpha$  is defined as:  $\alpha = \beta(t) = \beta_1 \cdot t^2$ , ( $1 \geq t \geq 0$ ), where  $\beta_1$  controls the initial noise magnitude. Note that a smaller  $\alpha$  corresponds to a higher noise intensity. Unless otherwise specified, all AMIX-1 models were trained with  $\beta_1 = 1.0$ .

#### 3.2 Predictive Scaling Behavior

**Motivations** First, predicting the loss trajectory of a model [38, 39] has gained significant attention in the LLMs community. The primary motivation lies in the substantial computational and financial costs of

**Table 1** Architecture of AMIX-1 at various scales.

Model Size	8M	35M	150M	350M	650M	1.7B
Number of layers	6	12	30	33	33	48
Hidden dimensions	320	480	640	960	1280	1680
Attention heads	20	20	20	20	20	40



**Figure 3** Scaling laws of cross-entropy and training FLOPs across various model sizes and the noise level  $\alpha$ . We show the equation between  $\mathcal{L}$  and FLOPs  $F$ , indicating that the AMIX-1 models have predictive scaling laws.

training LLMs. Accurately forecasting final performance early in the training process enables practitioners to make informed decisions about whether to continue, halt, or adjust the training strategy. This predictable capability mitigates the risk of wasted resources and enables more efficient exploration of architectural and training choices. This motivation is particularly relevant in the context of BFN models, which also operate at scale and under varying noise levels.

Second, predictable loss modeling is closely linked to the phenomenon of emergence (see [Section 4](#)), in which specific downstream capabilities arise abruptly once the training loss falls below a critical threshold. Consequently, forecasting the loss trajectory not only enhances computational efficiency but also offers a principled approach to anticipating the onset of emergent behaviors.

**Experiment Settings and Metrics** Prior studies [40, 41] often adopt an averaged loss across different noise levels to characterize the scaling behavior of diffusion-style models. In contrast, for AMIX-1 models, we investigate fine-grained scaling behavior under varying noise levels, rather than aggregating results over a single average noise configuration. We make this decision for two main reasons. First, BFNs naturally involve multi-objective learning across noise levels, making simple averaging inappropriate as it masks the distinct learning dynamics of each subtask. Second, under extreme noise settings, BFN models often fail to converge, rendering them unscalable; averaging in such cases introduces excessive variance and distorts the scaling law. Therefore, we compute the scaling relationship separately for each noise level  $\alpha$  to capture a more faithful and fine-grained behavior of BFN models.

As for the metric, although we employ KL divergence (i.e., [Equation \(4\)](#)) as the training objective, we report the cross-entropy between the output distribution  $p_{\text{out}}$  and the ground-truth sequence,  $\mathcal{L} = -\log p_{\text{out}}$ , to evaluate scaling behavior. This choice is motivated by two factors. First, from an information-theoretic perspective, the cross-entropy measures the inefficiency of encoding the ground-truth sequence using the model’s predicted distribution. Since the ground truth is inherently one-hot, minimizing cross-entropy directly reflects the model’s ability to recover precise, token-level information. Second, prior works on scaling laws [25, 26, 42] have consistently adopted cross-entropy, allowing for direct and meaningful comparisons.

**Scaling Law of Compute** We begin by varying the number of training steps for our AMIX-1 models, covering model sizes from 8M to 1.7B parameters. Since the total training FLOPs  $F$  can be approximated as  $F \approx 6ND$ , where  $N$  denotes the number of model parameters and  $D$  is the number of training tokens, fixing the model size allows us to use FLOPs as a proxy for training compute. Specifically, for a given noise level  $\alpha$ ,

we examine the power-law relationship between cross-entropy loss  $\mathcal{L}$  and training FLOPs  $F$ :

$$\mathcal{L}(F) = E_0 + e^a \cdot F^b, \quad (6)$$

where  $E_0$  denotes the irreducible loss, representing the asymptotic lower bound that cannot be reduced through additional compute. As shown in [Figure 3](#), our AMIX-1 models exhibit clear scaling behavior under two noise levels  $\alpha = 0.16$  and  $0.32$ : the validated data points (red stars) align closely with the extrapolated power-law curves, consistent with the scaling laws observed in large language models [19, 42]. This consistency enables reliable performance forecasting and informs efficient compute allocation for larger training regimes.

In addition to moderate noise levels ( $\alpha = 0.16$  and  $0.32$ ), we evaluate  $\mathcal{L}$  under extreme noise conditions ( $\alpha = 0.08$  and  $0.64$ ), as shown in [Figure 13](#). Under these settings, the BFN models fail to capture the training objective defined in [Equation \(4\)](#), leading to erratic and unpredictable cross-entropy values. What's more, although both  $\alpha = 0.16$  and  $\alpha = 0.32$  yield similar scaling trends, the subtle differences in their exponents indicate that careful scheduling of  $\alpha$  can significantly influence convergence behavior and generalization. Since  $\alpha$  directly modulates the injected noise, we point out that **an appropriate schedule of  $\alpha$  when training may be essential for stabilizing training and achieving optimal performance in BFN-based models.**

### 3.3 Optimal Estimation under Fixed FLOPs

In this section, we address the following question: Given a fixed training FLOPs budget, how should it be optimally allocated between model size and the number of training tokens for AMIX-1 models? Following the methodology of Chinchilla [26], we examine this trade-off from two complementary perspectives.

**Parameter-aware Scaling Dynamics** Under a fixed compute budget, it is important to determine an optimal balance between model size and data scale. To this end, we adopt a parametric scaling law that models the cross-entropy  $\mathcal{L}$  as a joint function of the number of model parameters  $N$  and the number of training tokens  $D$ . Specifically, under a fixed noise level, we fit the cross-entropy across multiple configurations of  $(N, D)$  using the following formulation:

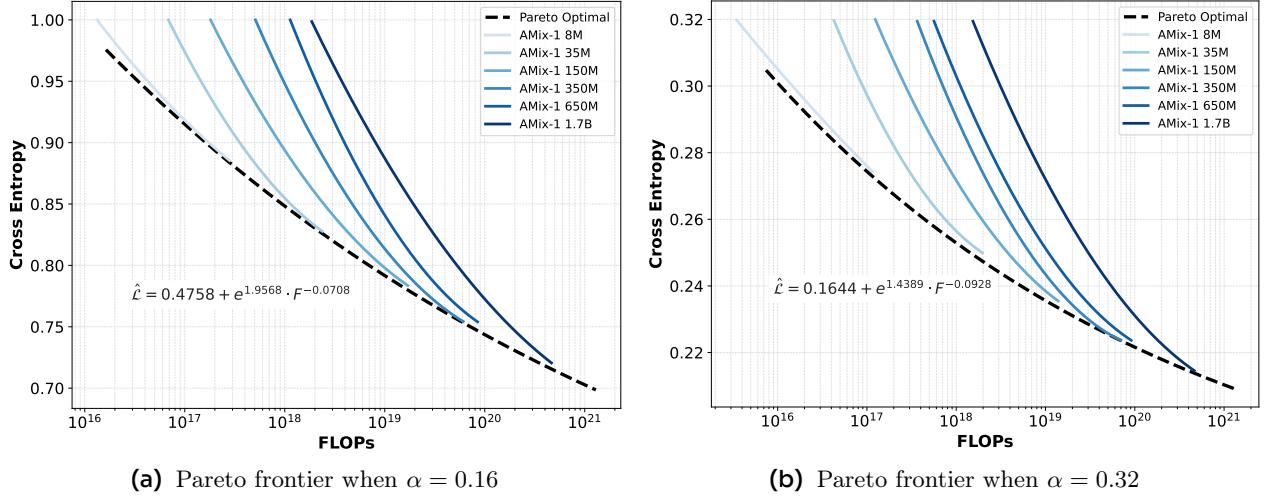
$$\mathcal{L}(N, D) = E + \frac{A}{N^n} + \frac{B}{D^d}, \quad (7)$$

where  $E$  represents the irreducible loss floor, and the two additive terms quantify the diminishing returns from increasing model size and training data, respectively.

As discussed in [Section 3.2](#), we investigate the parametric scaling law of AMIX-1 models across different noise levels  $\alpha$ , with detailed results summarized in [Table 2](#). Additional fitting results for other noise settings are provided in [Appendix D.3](#). To ensure robustness and assess generalization, we split the data by training tokens, using the former 30% for fitting and the latter 30% for validation. Following Hoffmann et al. [26], we minimize the Huber loss [43] between the predicted and observed log loss using the L-BFGS algorithm [44] to estimate [Equation \(7\)](#). We quantify the goodness of fit using two metrics: the Fitting Mean Relative Error (MRE) and the Validation MRE, defined in [Appendix D.2](#). These metrics reflect the relative prediction accuracy within and beyond the fitting region, respectively.

**Table 2** Different parametric scaling laws under various noise levels. Note that both  $N$  and  $D$  are measured in millions.

Noise Level	Parameters					MRE (%)		Scalable	
	$\alpha$	$E$	$A$	$B$	$n$	$d$	Fitting	Valid	
0.02	2.5941	0.0000	0.7461	0.2595	0.4891		0.56	0.55	✗
0.16	0.4638	0.2570	4.6740	0.0380	0.3332		0.73	1.66	✓
0.32	0.1703	0.0122	0.5813	0.4850	0.2047		1.06	1.57	✓
0.64	0.0278	0.0000	0.5758	0.3088	0.5887		2.22	5.92	✗



**Figure 4** Pareto optimal curves comparing validation loss versus FLOPs for six different model scales. The dashed black lines denote the Pareto frontiers, capturing the best achievable trade-offs between compute and validation loss under different noise levels.

Similar to the results in the scaling law of compute, the parametric scaling law is also  $\alpha$ -aware. For example,  $\alpha = 0.02$  and  $0.64$  exhibit unscalable behaviors, as evidenced by the parameter  $A$  being effectively estimated as  $0.000$ . This implies that under these settings, increasing the number of model parameters  $N$  brings no improvement to the cross entropy  $\mathcal{L}$ . The vanishing  $A$  values at both ends of the  $\alpha$  spectrum suggest that only within a moderate noise regime can model size effectively trade-off against training tokens to reduce the cross-entropy. This is precisely what we observe at  $\alpha = 0.16$  and  $0.32$ , where  $A$  is significantly non-zero, and both Fitting and Valid MREs are small, confirming the presence of a predictive, scalable relationship.

**Compute-Optimal Frontier** Summarizing the above analysis, we conclude that as the compute budget increases, both model size and the number of training tokens should scale approximately proportionally to maintain optimal performance. Following this principle, we adopt the approach of Hoffmann et al. [26] to estimate a **compute-optimal frontier**, which serves as a guideline for jointly managing the model size and data scale under a fixed FLOPs budget.

For each noise level  $\alpha$  under which BFN models exhibit scalable behavior, we analyze the compute-optimal Pareto frontier to identify the most efficient combination of model size and training token count that minimizes validation loss under a fixed computational budget. This frontier delineates the best attainable trade-off between parameter count and data usage, offering a principled framework for resource allocation across varying noise regimes. Under a given noise level  $\alpha$ , our objective is to characterize the compute-optimal frontier that relates the minimum achievable cross-entropy  $\hat{\mathcal{L}}$  to the training FLOPs  $F$ , similar to Equation (6):

$$\hat{\mathcal{L}} = E_0 + e^a \cdot F^{-b}. \quad (8)$$

To accurately estimate the scaling parameters  $a$  and  $b$ , we analyze the cross-entropy loss as a function of FLOPs for six AMIX-1 model variants ranging from 8M to 1.7B parameters. In Figure 4, we present two compute-optimal frontiers for  $\alpha = 0.16$  and  $\alpha = 0.32$ . Each curve corresponds to a fixed model size, with dots denoting empirical measurements under different training regimes and solid lines representing smoothed scaling trends. Larger models such as AMIX-1-650M and AMIX-1-1.7B display steeper loss reductions, suggesting greater efficiency in leveraging additional compute. In contrast, smaller models, such as AMIX-1-8M and AMIX-1-35M, saturate earlier, indicating a limited capacity to benefit from extended training. The dashed black line shows the global Pareto frontier, fit across all configurations. Notably, AMIX-1-650M and AMIX-1-1.7B closely align with this frontier, demonstrating near-optimal compute efficiency. Conversely, the pronounced deviation of smaller models from the frontier highlights their sub-optimality in the compute–performance trade-off.

## 4 Emergent Ability

Emergent ability [45] is a critical topic in large language modeling, as it seeks to understand how high-level capabilities arise, such as reasoning, as foundation models become more powerful with increasing size and computational investment. In the context of protein foundation models, studying emergent abilities is equally important for understanding how the capacity to represent and generate protein sequences, structures, and even functions evolves as the training objective is progressively optimized. Such investigations help determine whether a specific modeling goal is achievable by continually scaling data quantity, model size, and compute within a given foundation model framework.

Building on the predictive scaling law established in Section 3, we further investigate the less predictive aspects of AMIX-1’s performance in protein generation. Our analysis of emergent behavior adopts a loss-centered perspective, following Du et al. [46]. We adopt the predictive cross-entropy loss in Section 3 as an anchor to empirically map training loss to protein generation performance. This empirical mapping provides a practical framework for interpreting how emergent capabilities arise during training, offering insight into the relationship between optimization dynamics and functional outcomes in protein modeling.

**Key Result ②:** During the training process of AMIX-1, its structural understanding progressively emerges, with a consistent empirical alignment to its cross-entropy across model capacity.

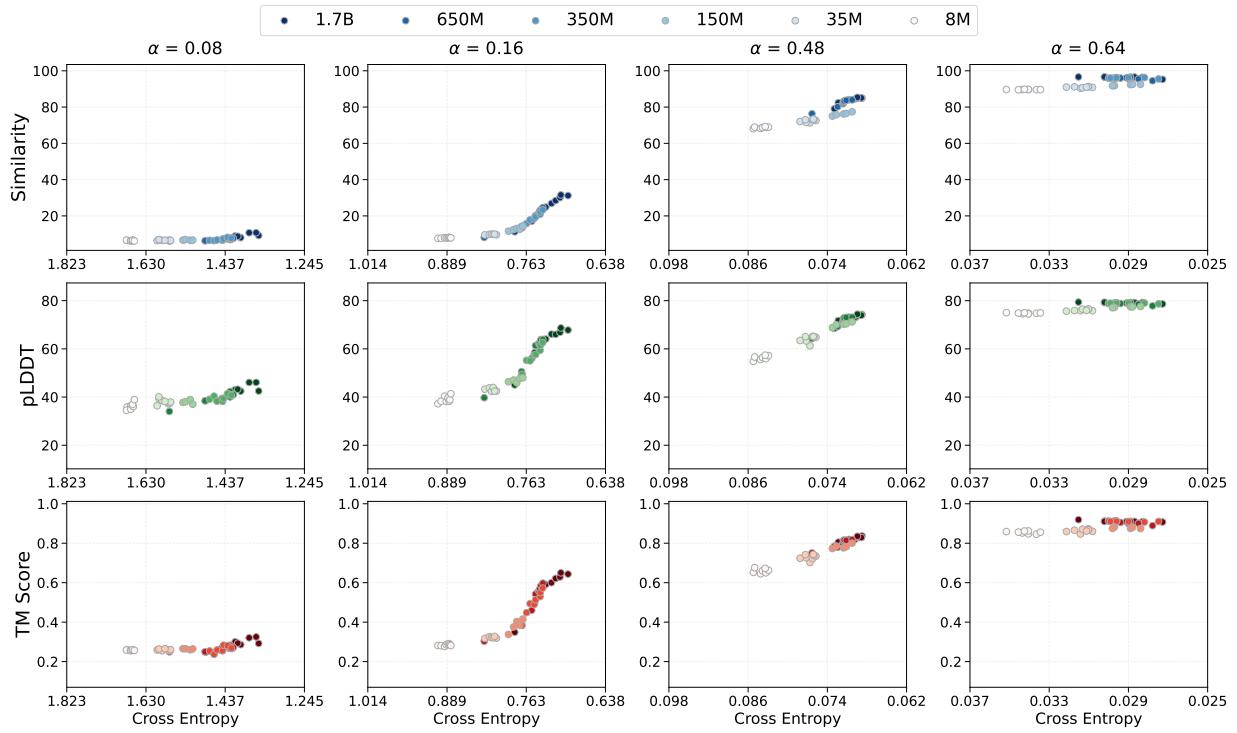
### 4.1 Experimental Setup

In protein sequence learning, a central question is how structural information emerges as sequences are progressively learned, an inquiry grounded in the widely accepted “sequence-structure-function” paradigm [47]. To investigate this emergent process, we focus on three key capabilities: **sequence consistency**, reflecting sequence-level recovery from corrupted sequence distribution; **foldability**, representing the transition from sequence understanding to structural feasibility; and **structural consistency**, indicating the model’s ability to preserve structural features. Notably, we demonstrate that structural consistency plays a critical role, as it underpins the emergence of few-shot learning capabilities discussed in Section 5.

**Sequence Consistency** We evaluate the sequence consistency of the generated protein sequences  $\mathbf{y} = \{y_1, \dots, y_m\}$  by measuring their similarity to the corresponding input sequences  $\mathbf{x} = \{x_1, \dots, x_m\}$ , using the metric as  $\frac{1}{m} \sum_j \mathbf{I}[x_j = y_j]$ . This metric quantifies the model’s ability to recover a corrupted distribution at the sequence level, aligning with the self-supervised pretraining objective. A higher sequence consistency indicates that the generated sequence closely follows the corrupted input, suggesting stronger adherence to the original distribution, while a lower SC score implies greater novelty in the generated output.

**Foldability** Foldability serves as a critical assessment bridging the gap between protein sequence and structure. Following recent practices in the community, we evaluate the foldability of generated sequences using ESMFold [5]. For each generated sequence, we predict its three-dimensional structure and extract the corresponding predicted Local Distance Difference Test (pLDDT) score [48]. The pLDDT score acts as a proxy for local structural confidence, with higher values indicating more reliable and well-formed structural predictions. From a foldability standpoint, elevated pLDDT scores suggest that the generated sequences are more likely to adopt stable, well-defined conformations, rather than collapsing into disordered or non-folding structures. Thus, this metric offers a valuable indication of the biophysical plausibility of novel proteins generated by the model.

**Structural Consistency** To assess how well AMIX-1, trained with a sequence-level objective, preserves structural consistency between generated sequences and their corresponding inputs, we employ the TM-score—a widely used metric for quantifying structural similarity between protein conformations. Specifically, we compare the three-dimensional structures predicted by ESMFold [5] for both the input and generated sequences. The TM-score ranges from 0 to 1, with higher values indicating greater structural alignment; scores above 0.5 typically suggest that the two structures share the same overall fold. From the perspective of



**Figure 5** Emergent abilities of AMIX-1 models under varying noise levels  $\alpha$ . Noise levels of  $\alpha = 0.16$  and  $0.48$  exhibit clear emergence. Darker colors indicate models with more parameters. Lower cross-entropy, and consequently stronger downstream performance, emerges as the model size increases.

sequence-conditioned generation, a high TM-score indicates that the model successfully retains the global structural scaffold of the input protein while introducing meaningful sequence-level diversity. This consistency metric is especially important in applications where structural conservation is essential due to functional or evolutionary constraints.

For the above metrics, we take Automated Model Evaluation (CAMEO) [49] as our evaluation benchmark. It is a continuously updated benchmark dataset that provides newly released, high-quality protein structures from the Protein Data Bank (PDB) for real-time assessment of protein structure prediction methods. To assess the scalability of our models, we randomly selected 20 proteins from CAMEO. Details of these sequences are provided in [Appendix C](#).

In this emergent ability analysis, AMIX-1 adopts a setup consistent with the scaling law experiments described in [Section 3](#). We do not apply additional decoding strategies, unlike previous foundation models [50, 51], in order to stabilize the emergent ability analysis and avoid introducing confounding effects. We structure the analysis along the axis of corruption noise, denoted as  $\alpha$ . Specifically, we investigate four representative values:  $\{0.08, 0.16, 0.48, 0.64\}$ , where  $\alpha = 0.08$  and  $\alpha = 0.64$  correspond to extremely high and low noise levels, respectively, providing a comprehensive view of emergent behavior across the noise spectrum. These same  $\alpha$  values are also used as the initial noise levels in AMIX-1’s generation process. Our goal is to align the cross-entropy dynamics at these specific noise levels with the corresponding generation behavior, enabling a direct mapping between training metrics and generation performance.

## 4.2 Findings

We conduct emergent ability analysis using sequence consistency, foldability, and structural consistency as evaluation axes, and use the starting noise levels  $\alpha \in \{0.08, 0.16, 0.48, 0.64\}$  as the noise dimension. The results are presented in [Figure 5](#), from which we draw several key findings summarized below.

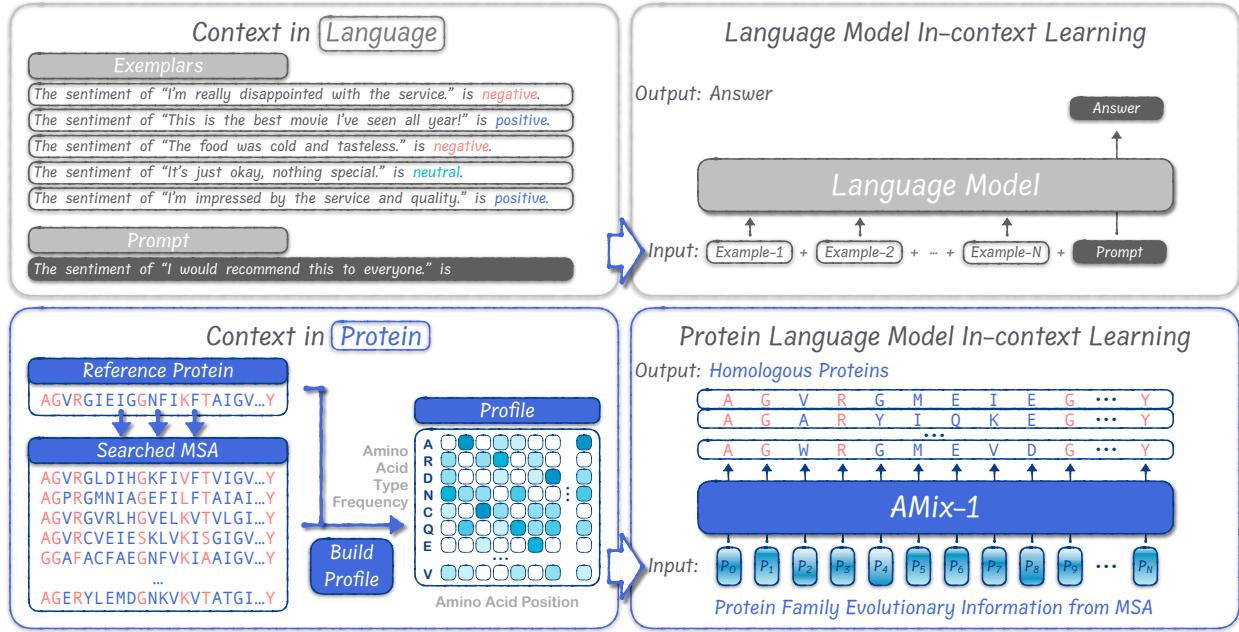
**Emergent capabilities are governed by the training objective.** Model capabilities emerge as a direct consequence of optimizing the underlying training objective, underscoring the central role of loss minimization in the development of foundation models. Notably, we observe that, regardless of parameter count or training stage, all evaluated metrics are consistently aligned with the cross-entropy loss at each corruption level  $\alpha$ . These findings suggest that emergent abilities in BFN-based models are not solely a function of increasing model size; rather, they critically depend on the degree of epistemic uncertainty introduced via the noise parameter  $\alpha$ . This alignment between loss and capability offers a principled metric, as a ruler, for assessing the expected performance of newly trained models, as predicted by the established scaling law. Leveraging this ruler, we successfully train AMIX-1-1.7B, ensuring that it achieves the desired emergent behaviors with predictive outcomes.

**Structural understanding emerges as the sequence objective is optimized.** Despite being trained purely with sequence-level objectives, AMIX-1 exhibits progressively stronger structural awareness as training advances, demonstrating that structural capabilities can indeed emerge without explicit structural supervision. We observe that as the sequence-level metric, namely cross-entropy, continues to decrease, both structural metrics (pLDDT and TM-score), along with sequence consistency, begin to improve rapidly once the cross-entropy drops below a certain threshold. Focusing on a representative case with median noise level  $\alpha = 0.16$ , we find that performance initially plateaus: structural metrics show limited improvement even as the model continues to optimize its loss. However, upon further investment of compute, and as cross-entropy reaches a sufficiently low value, the model suddenly exhibits a sharp increase in structural performance—an effect commonly referred to as emergent ability. This observation is encouraging, as it suggests that structural understanding can emerge reliably from purely sequence-based, self-supervised learning, provided that sufficient optimization and compute are applied.

**Emergent behavior is suppressed at large and small noise levels.** Models trained at extreme noise levels, either too low or too high, fail to exhibit emergent capabilities, suggesting that only intermediate noise regimes provide sufficient signal for scalable generalization. Specifically, we observe that improvements in structural prediction metrics, including pLDDT, TM-score, and structural similarity, emerge under intermediate noise levels, particularly at  $\alpha = 0.16$  and  $\alpha = 0.48$ . In these regimes, as cross-entropy loss decreases, AMIX-1 consistently demonstrates substantial performance gains across all three metrics, regardless of model size. These improvements only materialize when the model is sufficiently trained to reduce cross-entropy below a critical threshold and when the injected noise balances exploration and convergence during training. Biologically, this shift reflects a transition from residue-level prediction to coherent, fold-level modeling, an indicator of structurally grounded intelligence. In contrast, no emergent behavior is observed at extreme values of  $\alpha$ , such as  $\alpha = 0.08$  (high noise) or  $\alpha = 0.64$  (low noise). At high noise levels, training dynamics become dominated by excessive stochasticity, which impairs the model's ability to extract stable structural patterns. Despite continued optimization of the loss function, all structural performance metrics stagnate at low levels, showing no meaningful improvement with increased model size. Conversely, at low noise levels, the near-deterministic training dynamics lead to premature convergence and overfitting. The model fails to explore diverse structural configurations, limiting its capacity to generalize beyond the training data. In both cases, model scaling alone is insufficient to yield qualitative improvements in structure prediction. These findings underscore that emergent capabilities require not just increased scale but also a well-calibrated training noise level within an appropriate operating range.

## 5 In-Context Learning

In-context learning (ICL) emerges as a central capability that enables large language models to learn from a few contextual exemplars to perform unlimited tasks without task-specific fine-tuning [17, 52]. This intriguing ability motivated us to develop a training-free strategy **to unify diverse protein design tasks by leveraging the in-context learning ability of AMix-1**, which also serves as the foundation of our test-time scaling algorithm described in [Section 6](#). In this section, we propose a novel approach that conditions AMIX-1 on evolutionary profiles as prompts, enabling the generation of proteins that preserve both structural integrity and functional relevance. We validated the effectiveness and generalizability of this approach through extensive *in-silico*



**Figure 6** Comparison of standard LLM in-context learning (top) with protein family conditioning via MSA profiles in AMix-1 (bottom). In LLM ICL, few-shot exemplars are token-level demonstrations; in AMix-1, the MSA profile serves as a dense, position-wise family prompt.

evaluations in structural and functional consistency. Wet-lab experiments further confirmed its potential, successfully engineering the AmeR transcriptional repressor with 50-fold increase in activity over the wild type.

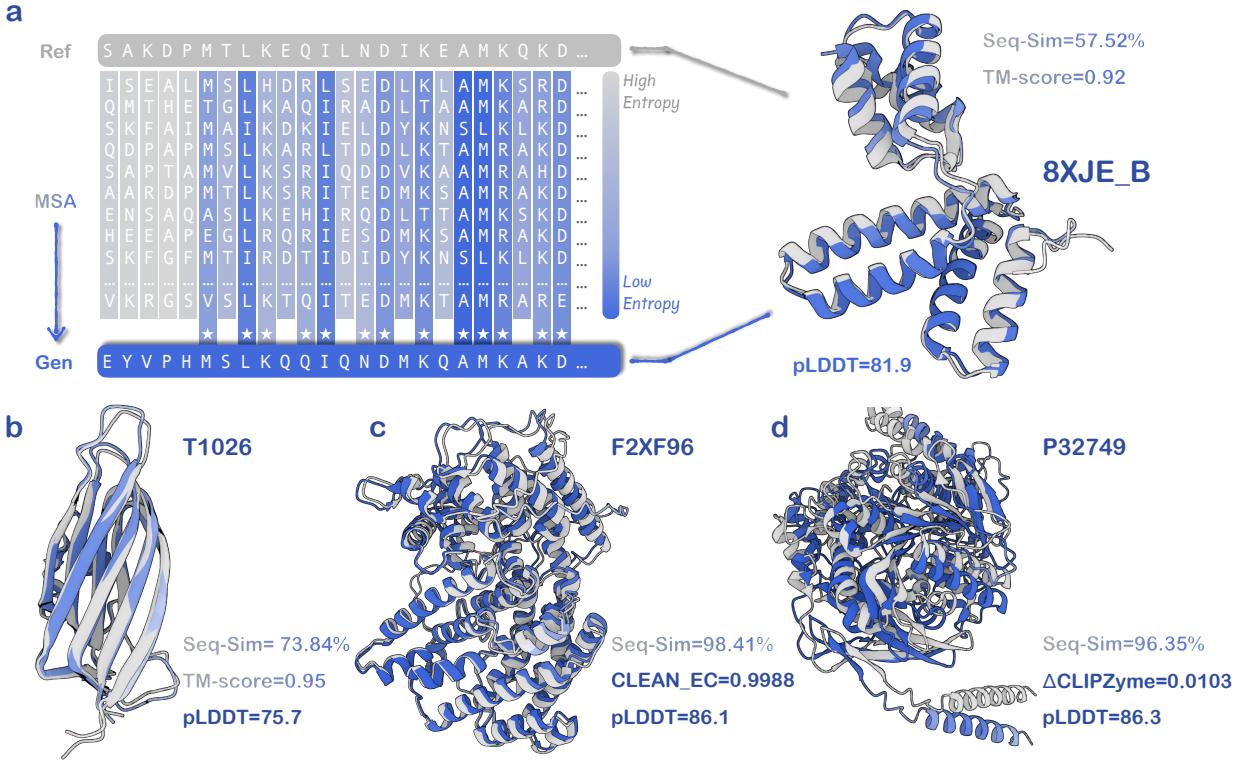
**Key Result ③:** AMix-1 unifies diverse protein design tasks by leveraging its in-context learning ability to generate structure- and function-preserved proteins with a conditional protein profile.

## 5.1 AMix-1 is a Unified Protein Designer

We now introduce how evolutionary prompts are constructed to enable in-context learning in AMIX-1. In the ICL framework of AMIX-1, we treat MSAs as prompts that guide generation toward a target protein family. MSAs encode evolutionary constraints—such as conserved motifs and co-evolving positions—which are critical for generating structurally and functionally valid sequences. Rather than conditioning directly on raw alignments, we transform the MSA into a position-wise frequency profile  $\mathbf{P}$ , which serves as a dense, compressed representation of the family prompt.

Formally, given an alignment matrix  $\mathbf{X} \in \{0, \dots, K\}^{n \times m}$  of  $n$  homologous sequences aligned to length  $m$ , we define the profile  $\mathbf{P} = \{\mathbf{P}_i\}_{i=1}^m$ , where each  $\mathbf{P}_i \in \Delta^{K-1}$  is a categorical distribution over amino acids at position  $i$ . Each component is computed as  $\mathbf{P}_{i,k} = \frac{1}{n} \sum_{j=1}^n \mathbb{I}[\mathbf{X}_{ji} = k]$ , where  $k$  indexes the amino acid alphabet of size  $K$ ,  $\mathbf{P}_{i,k}$  is the frequency of amino acid  $k$  at position  $i$  in the MSA, and  $\mathbb{I}[\cdot]$  denotes the indicator function.

This transformation serves two purposes. First, it introduces a mild form of noise by averaging over observed residues—naturally aligning with AMIX-1’s denoising training objective. Second, by compressing MSAs of arbitrary depth into a fixed-size matrix, the profile representation makes AMIX-1 invariant to the number of input sequences. This not only enables consistent inference efficiency regardless of MSA depth, but also simplifies the conditioning interface to the model [53].



**Figure 7 Case studies validating in-context learning in AMix-1.** AMix-1 preferentially conserves evolutionarily important positions—i.e., positions with low entropy (shown in **a** left, colored in blue)—where the generated proteins more frequently match the wild-type sequence, compared to high-entropy (variable) positions (gray). The right side of **a**, as well as **b-d**, shows structural alignments and functional assessments between the generated and wild-type proteins. (**a, b**): Structural similarity to the wild type, quantified by TM-score. (**c**): Functional consistency based on EC Number, measured by CLEAN confidence. (**d**): Catalytic activity preservation on a given reaction, measured by the change in CLIPZyme score ( $\Delta$ ). In all panels, blue structures denote generated proteins, and gray structures denote the input wild type.

Given a profile prompt  $\mathbf{P}$ , AMIX-1 models the conditional probability over sequences in the family as:

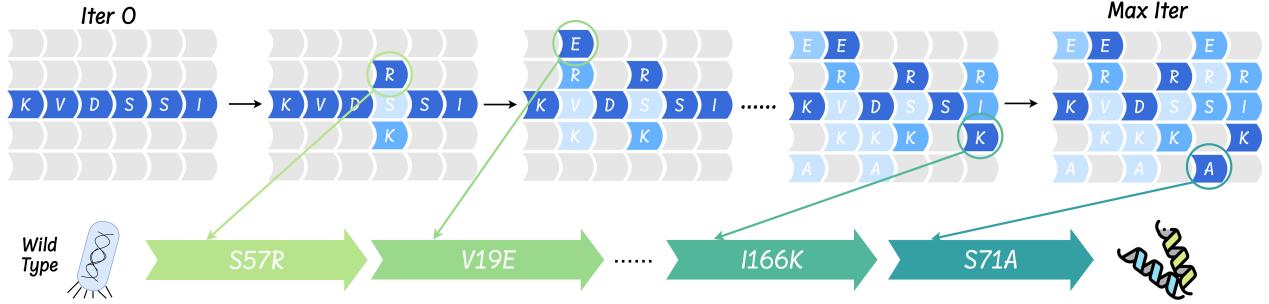
$$p_{\theta}(\mathbf{x} \mid \mathbf{P}) = \prod_{i=1}^m p_{\theta}(x_i \mid \mathbf{P}) \quad (9)$$

where  $\mathbf{x} = (x_1, \dots, x_m)$  is the generated sequence and  $\theta$  denotes the frozen model parameters. During decoding, we select the most probable residue at each position via  $\arg \max_k p_{\theta}(x_{i,k} \mid \mathbf{P})$ .

In this way, AMIX-1 acts as a conditional generator over the protein manifold, using the MSA-derived profile as a soft constraint that guides generation toward the desired family. This enables general-purpose, structure- and function-preserving design without any parameter updates.

## 5.2 In Silico: AMix-1 Excels at In-Context Learning

We validate the in-context learning capability of AMIX-1 through a set of in silico case studies. Each case examines whether the model can accurately extract and generalize structural or functional constraints from input exemplars—encoded as profiles—with access to explicit labels or structural supervision. We consider four representative case studies, each conditioned on a distinct profile, and generate novel proteins using AMIX-1. Structural properties of the output sequences are evaluated using ESMFold [5], while functional predictions are made using CLEAN [54] and CLIPZyme [55]. Seq-Sim of a generated protein  $x'$  is computed as  $\max_{x \in \mathbf{X}} \frac{1}{m} \sum_{j=1}^m \mathbb{I}(x_j = x'_j)$ , where  $\mathbf{X}$  is the input MSA and  $m$  is the sequence length.



**Figure 8** Generation Iterations and corresponding mutation process in wet-lab

**Structural Generalization from Random Homolog.** The input consists of a randomly sampled protein family from CAMEO. As shown in Figure 7(a), AMIX-1 generates novel proteins whose predicted 3D structures remain highly consistent with the input, as measured by structural alignment metric TM-score. The average predicted pLDDT exceeds 85, indicating confident and foldable outputs. Despite sequence-level novelty (low Seq-Sim to any input MSA sequence), the generated structures preserve the canonical fold, highlighting the model’s capacity to generalize conserved scaffolds.

**Generalization from Orphan Protein.** In Figure 7(b), we examine a challenging scenario where the input is an orphan protein—i.e., one with only a few known homologs and minimal representation in existing protein families. Such proteins are underrepresented in training data, making them difficult for models to generalize to. Despite this, AMIX-1 successfully generates foldable variants that exhibit coherent tertiary structures with high pLDDT scores. This demonstrates that even sparse sequence profiles can serve as effective prompts, and that AMIX-1 is capable of extrapolating structural patterns purely from weak but positionally localized evolutionary signals.

**Function Retention—Enzyme with Original EC.** We next assess functional generalization. Figure 7(c) presents results from prompting with an MSA containing enzymes annotated with EC 4.2.3.120 (beta-pinene synthase). The generated proteins are evaluated using CLEAN, a state-of-the-art model for EC number prediction. Over 80% of the generations are assigned the correct EC number with high confidence, and the predicted functional entropy remains low, indicating strong specificity. The average predicted CLEAN confidence for generated sequences closely matches the input enzyme, demonstrating functional conservation through in-context prompting.

**Catalytic Activity Conserved on Specific Reaction.** Finally, we test whether AMIX-1 can preserve catalytic activity for a chemically complex reaction. As shown in Figure 7(d), the input MSA consists of enzymes known to catalyze the transformation:  $[\text{CH}_3][\text{N}^+](\text{CH}_3)(\text{CH}_3)\text{CH}_2\text{CHO} + \text{H}_2\text{O} + \text{O}_2 \longrightarrow [\text{CH}_3][\text{N}^+](\text{CH}_3)(\text{CH}_3)\text{CH}_2\text{COOH} + \text{H}_2\text{O}_2$ . Using CLIPZyme, we quantify the catalytic plausibility of each generated sequence by computing the change in predicted activity scores relative to the input enzyme. We report a positive mean shift  $\Delta$  in catalytic confidence across generations, demonstrating that the model does not merely preserve function—it can even improve functional alignment under certain scoring metrics.

Across these diverse scenarios, AMIX-1 consistently demonstrates strong in-context learning capabilities. It synthesizes sequences that maintain both **structural fidelity** (as evidenced by high pLDDT and TM-score to prompt structure) and **functional specificity** (as validated by CLEAN and CLIPZyme predictions). Notably, this is achieved without any model fine-tuning or task-specific supervision, using only profile-based conditioning at inference time. These results underscore the utility of MSA profiles as compact, expressive prompts that unlock biologically meaningful generalization in protein generation.

### 5.3 Wet Lab: AMix-1 Designs High-Activity Proteins

To validate the practical efficacy of AMIX-1’s in-context learning, we experimentally engineered the AmeR transcriptional repressor [56]. The results demonstrated that model-generated mutants enhanced the protein’s activity by up to 50 fold compared to the wild type. This represents an approximate 77% improvement over the best previously reported results.

**Setup.** We targeted the AmeR protein, a TetR-family transcriptional repressor. In synthetic biology, AmeR is a widely used orthogonal molecular switch—a fundamental component for building predictable genetic circuits [56]. High-performance switches are critical for sophisticated applications in metabolic engineering, biosensors and cell therapies. Our objective was to enhance AmeR’s DNA binding affinity, thereby creating a more sensitive and reliable regulator.

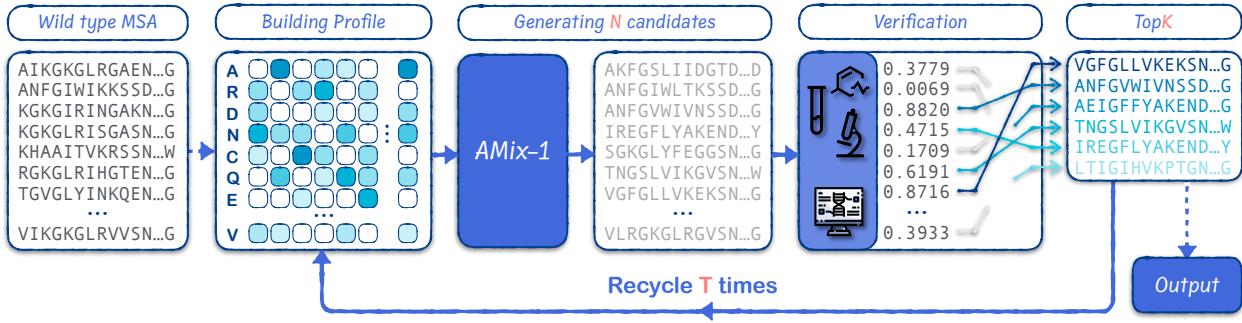
Using our AMIX-1-650M model, we generated 40 candidate sequences, each with ten or fewer mutations. The fitness of these variants was quantified using a fluorescent reporter assay, where higher DNA binding affinity results in stronger repression of a fluorescent protein, measured as “fold repression”. A higher fold repression value indicates superior function. The performance of AMIX-1 was benchmarked against three baselines: (1) Wild Type (WT); (2) a Single-Mutant Library; and (3) EvoAI [57], a state-of-the-art directed evolution method. Illustration of wet lab verification process guided iteratively by AMIX-1 is shown in Figure 8 while detailed protocols are provided in Appendix G.

**Results.** The full experimental results are summarized in Figure 9. The superior performance of the AMIX-1-designed variants highlights our model’s key strengths. It demonstrates a powerful ability to navigate the vast sequence-function landscape, identifying not just individual high impact mutations but also beneficial combinatorial effects. This capability allows AMIX-1 to discover highly active sequences inaccessible to methods that rely on local exploration or the recombination of known variants.

## 6 Test-Time Scaling

Test-time scaling (TTS) enhances model performance by allocating additional computational resources during inference time [27, 58, 59]. In this section, we introduce EvoAMIX-1, an evolutionary test-time scaling algorithm that leverages AMIX-1 as a proposer and an external verifier to iteratively generate and select candidate protein sequences under evolutionary constraints. By integrating either task-specific *in silico* reward functions or experimental feedback from assays, EvoAMIX-1 enables efficient directed protein evolution without model fine-tuning. Extensive experiments show that EvoAMIX-1 consistently outperforms AMIX-1’s in-context learning and strong baseline methods across all six design tasks, highlighting its potential for real-world applications in protein engineering.

**Key Result ④:** The protein design ability of AMIX-1 is scalable in test time with increasing verification budgets, and is universal to apply to a variety of directed evolution tasks with plug-in verification.



**Figure 10** Schematic of the AMix-1 test-time scaling loop. Starting from an initial MSA of the wild-type sequence, a position-wise frequency profile  $\mathbf{P}^{(0)}$  is computed and serves as input to the **proposer** (AMix-1), which generates a batch of  $N$  candidate sequences. An external **verifier**  $R(\cdot)$  scores each candidate and selects the top- $k$  variants, from which a refined profile  $\mathbf{P}^{(t+1)}$  is constructed. This propose-verify-update cycle repeats for  $T$  rounds, yielding progressively optimized protein designs without any parameter updates.

## 6.1 Scaling AMix-1 with Evolutionary Algorithm

EvoAMIX-1 can be viewed as a form of evolutionary algorithm. Evolutionary algorithms are inspired by the principles of natural selection, in which biomolecules—such as proteins and genes—are iteratively mutated and screened to identify variants with enhanced or targeted functional properties [60]. We illustrate the test-time scaling workflow of AMIX-1 in Figure 10. Specifically, at the beginning of the evolutionary process, a MSA constructed from the wild-type sequence is provided as input to AMIX-1. Acting as a **proposer**, AMIX-1 generates a batch of candidate sequences. An external **verifier** then evaluates these candidates and selects the top- $k$  based on the desired criteria. These top-ranked sequences are used to construct a new MSA, which is fed back into AMIX-1 to prompt the next round of evolution.

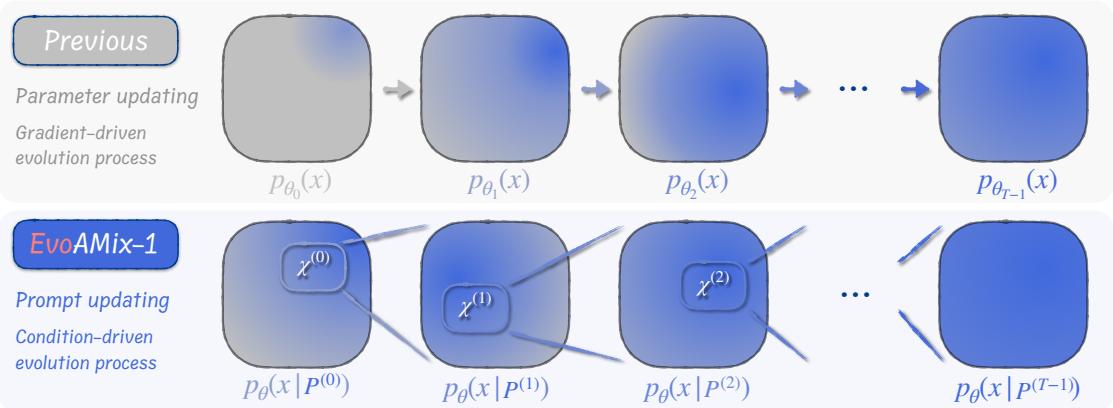
We further provide a formal description of EvoAMIX-1 in Algorithm 3. Let  $\mathbf{X}$  denote an input MSA, from which we construct the initial profile  $\mathbf{P}^{(0)} = \text{BuildProfile}(\mathbf{X})$ . The function  $\text{BuildProfile}(\cdot)$  constructs a position-wise amino acid frequency profile as defined in Section 5.1. At each round  $t$ , we sample  $N$  candidate sequences from AMIX-1 conditioned on  $\mathbf{P}^{(t)}$ , score them with an external verifier  $R(\cdot)$ , and maintain a history of all generated sequences and their scores. We then select the top- $k$  sequences from this accumulated history and construct an updated profile  $\mathbf{P}^{(t+1)}$  by computing their empirical residue frequencies. The procedure is iterated for  $T$  rounds.

## 6.2 A Proposal Distribution Perspective

We now turn to a more formal discussion of the mechanism underlying EvoAMIX-1, framing it through the lens of **proposal distributions** as shown in Figure 11. This perspective not only clarifies the optimization dynamics of our method, but also unifies traditional evolutionary algorithms in computational biology under a common framework, highlighting key distinctions between those approaches and EvoAMIX-1.

As proposed in Snell et al. [27], test-time scaling (TTS) algorithms can be interpreted through a proposer-verifier framework, where the proposer samples candidate solutions from a distribution, and the verifier evaluates them, selecting those with the highest fitness. Formally, this can be modeled as a sequence of proposal distributions, denoted as  $p^{(0)}(\mathbf{x}), p^{(1)}(\mathbf{x}), \dots, p^{(T-1)}(\mathbf{x})$ , where  $p^{(t)}(\mathbf{x})$  represents the proposal distribution at the  $t$ -th optimization round, and  $T$  is the maximum number of iterations. At each step, the algorithm samples candidate proteins from  $p^{(t)}(\mathbf{x})$  and updates the proposal distribution to  $p^{(t+1)}(\mathbf{x})$  by a combined signal of the last distribution, the generated candidates, and feedback from the verifier. This iterative process reflects the dynamic adaptation of the proposal distribution in response to fitness signals.

Many conventional methods, such as ALDE [61] and EVOLVEpro [62], can also be interpreted within this framework. These methods adopt a parametric predictor to shape a proposal distribution and generate new protein variants with importance sampling. The sampled protein are evaluated via *in silico* or wet-lab assays



**Figure 11** Proposal distribution shifting during the test-time scaling process. (1) Top: Previous methods typically adopt a gradient-driven evolution process, in which the model parameters  $\theta$  are iteratively updated. During optimization, the proposal distribution shifts as the model becomes increasingly tailored to generate samples that align with feedback from the verifier in the previous round. This approach modifies the underlying model to improve precision on newly verified data points. (2) Bottom: In contrast, EvoAMIX-1 utilizes a condition-driven evolution strategy through prompt updating rather than parameter tuning. At each iteration, EvoAMIX-1 identifies the most promising sequences and updates the in-context examples (e.g., MSAs or profiles) that serve as prompts to the foundation model. This iterative prompt refinement effectively reshapes the proposal distribution without altering the model itself, progressively narrowing the generation focus toward regions of sequence space favored by the verifier.

and the evaluation results are then used to retrain the predictor, producing a new proposal distribution for the next round. This yields a sequence of parameterized distributions  $p_{\theta^{(0)}}(\mathbf{x}), p_{\theta^{(1)}}(\mathbf{x}), \dots, p_{\theta^{(T-1)}}(\mathbf{x})$ , where  $\theta^{(t)}$  represents the model parameters at iteration  $t$ . The optimization process is thus driven by continual updates to the model parameters to better fit the observed feedback.

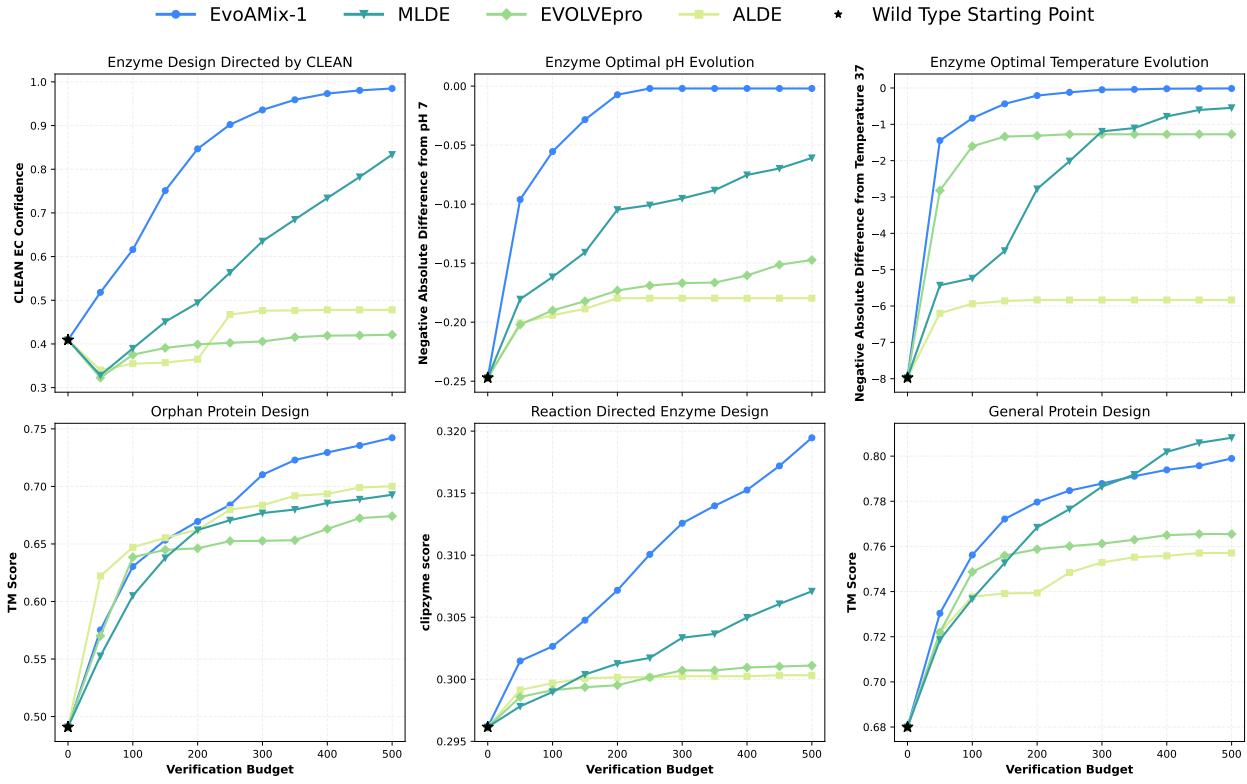
In contrast, EvoAMIX-1 adopts a fundamentally different philosophy: rather than updating model parameters, it constructs proposal distributions through in-context exemplars. At each round, AMIX-1 takes a set of MSAs or their profile as a prompt, treating it as the input condition to the protein foundation model. The foundation model then samples neighboring sequences, effectively defining a new conditional proposal distribution. This yields a sequence of context-dependent distributions:  $p(\mathbf{x}|\mathbf{P}^{(0)}), p(\mathbf{x}|\mathbf{P}^{(1)}), \dots, p(\mathbf{x}|\mathbf{P}^{(T-1)})$ , where  $\mathbf{P}^{(t)}$  is the prompt at round  $t$ , and  $p(\mathbf{x})$  is the frozen base distribution defined by the pretrained protein foundation model. This approach assumes that the model has already learned a rich, generalizable manifold of natural proteins and contains comprehensive structural and functional knowledge.

Rather than fine-tuning model weights, EvoAMIX-1 embraces a prompting-based methodology, akin to in-context learning [17, 52] and chain-of-thought prompting in large language models [63]. This allows for adaptive behavior during optimization without the need for costly retraining, aligning with the broader trend in foundation models of extracting knowledge through flexible conditioning rather than parameter updates.

### 6.3 AMix-1 Exhibits Robust Test-Time Scaling across Diverse Tasks

We applied EvoAMIX-1 to a series of protein design tasks. The results in Figure 12 show that it consistently exhibits strong test-time scaling behavior across all tasks—namely, as the number of verifier calls increases, the model continues to produce progressively higher-quality designs. Specifically, we evaluate EvoAMIX-1 across three representative classes of protein design tasks:

- **Identifying structurally consistent family members.** This task requires the model to generate protein sequences whose structures are more similar to a given reference. To evaluate this, we use ESMFold to predict the structures of generated sequences and compute their TM-scores against the reference structure. The TM-score serves as the output of the verifier in this task. This task covers both orphan proteins (with limited homologous information) and general proteins (with dense homology), allowing



**Figure 12** Test-time scaling performance of AMIX-1 across six in silico directed-evolution benchmarks. Each curve shows mean top-5 task score as a function of cumulative verifier calls (proxy for experimental throughput) for structural, biophysical, and functional design objectives. AMIX-1 exhibits monotonic performance gains with increasing inference compute, outperforms or matches state-of-the-art baselines in five out of six tasks, and avoids the premature plateaus observed in methods with restricted mutation schemes.

us to assess the robustness of EvoAMIX-1 under varying levels of evolutionary context.

- **Optimization of biophysical properties.** This task requires the model to iteratively generate variants starting from a wild-type sequence, guiding the design toward desired target properties. In this work, we focus on two representative biophysical attributes of enzymes: optimal temperature and pH. We use Seq2Topt [64] and Seq2pHopt [65] as task-specific verifier models to evaluate these properties.
- **Functional reprogramming.** This task aims to modify or enhance the function of a wild-type protein. Specifically, we focus on reprogramming enzymatic functions by targeting either EC number annotations or catalytic activity toward specific chemical reactions. We use CLEAN [54] and CLIPZyme [55] as functional verifiers for these two respective objectives.

In addition, we selected a set of strong directed evolution algorithms as baselines for comparison, including:

- **ALDE** [61] is an iterative protein engineering strategy that combines machine learning with Bayesian optimization to efficiently explore the epistatic sequence space and identify high-fitness protein variants. However, it requires manual specification of mutation sites, which significantly constrains its exploration space.
- **EVOLVEpro** [62] is an active learning framework that integrates a pretrained protein language model with a regression model to guide in silico directed evolution and discover high-activity protein variants. Notably, EVOLVEpro limits its search space to single-point mutations, reducing its capacity to explore more diverse or combinatorial sequence variations.

- **MLDE** [66] utilizes ESM language models and combines importance-based masking with random masking strategies to progressively introduce mutations into the wild-type sequence. It automatically generates and optimizes protein variants aimed at enhancing target functionality.

We report performance with respect to the verification budget, i.e., the number of verifier calls, as this directly reflects wet-lab throughput in real-world protein design scenarios. This perspective offers practical insight for lab-in-the-loop frameworks. More implementation details of these experiments can be found in [Appendix E](#). We now proceed to a more detailed discussion of the experimental results.

**EvoAMIX-1 is a universal protein optimizer.** Across all directed evolution tasks, EvoAMIX-1 demonstrates strong potential as a general-purpose optimizer for protein design under arbitrary objective functions. In contrast to our baseline approaches that rely on specialized training with task-specific environmental feedback, EvoAMIX-1 operates without requiring retraining for each protein design dataset. Instead, it treats environmental feedback as a modular component of the optimization loop. A particularly appealing feature of EvoAMIX-1 is that the feedback mechanism, whether derived from *in silico* metrics or experimental assays, can be seamlessly integrated as a plug-in. The success across diverse tasks highlights EvoAMIX-1’s flexibility and effectiveness, establishing it as a universal protein optimizer.

**EvoAMIX-1 consistently produces strong final designs.** In 5 out of 6 tasks, EvoAMIX-1 achieved the best design performance. It was slightly outperformed by MLDE only in the general protein design task, yet it still significantly outperformed all other baseline methods. This highlights the generality of AMIX-1, demonstrating its ability to quickly adapt to diverse types of protein design tasks.

**EvoAMIX-1 exhibits fast scalability.** Across all evaluated tasks, EvoAMIX-1 demonstrates rapid optimization performance, achieving the fastest improvement rate in 5 out of 6 benchmarks. This scalability advantage stems from its flexible design: unlike traditional approaches that constrain each evolutionary round to a predefined set of mutation sites, EvoAMIX-1 enables unrestricted exploration over the entire sequence space. This broader search strategy facilitates more efficient discovery of high-performing protein variants, accelerating the overall optimization process.

**EvoAMIX-1 escapes local optima.** In all tasks, EvoAMIX-1 continued to improve without observable performance plateaus, as long as the metric had not yet reached its theoretical upper bound. In comparison, ALDE and EVOLVEpro showed noticeable performance plateaus in several tasks, indicating limited capacity for continued improvement. This limitation is primarily due to the manual constraints imposed by both methods on mutation types and allowable mutation sites. Unlike these methods, EvoAMIX-1 does not rely on manually defined constraints, enabling broader and more flexible exploration of the sequence space.

**Ablation studies confirm key design choices.** To better understand the contribution of each component in EvoAMIX-1, we conduct targeted ablation experiments presented in [Appendix F](#). These studies reveal that removing either the pretrained generative model or the iterative profile refinement mechanism leads to substantial drops in both reward optimization and biological plausibility. Notably, substituting AMIX-1 with random mutation causes the model to quickly drift away from structurally and functionally coherent solutions, while disabling prompt updates prevents convergence to high-fitness regions. These findings validate the importance of valid generation and adaptive conditioning in realizing effective test-time scaling.

## 7 Conclusion

In this work, we propose a pathway methodology for training a scalable and emergent protein language model based on the Bayesian Flow Network (BFN) for scalable protein design. This methodology addresses key challenges in developing protein foundation models, inspired by advances in large language models, namely, scaling laws, emergent capabilities, in-context learning, and test-time scaling. Demonstrating scalability across data, model size, and computational resources, we train a 1.7 billion AMIX-1, in which sequence and structural understanding progressively emerge during training. Leveraging multiple sequence alignments, we

introduce an in-context learning mechanism that enables AMIX-1 to generate novel proteins conditioned on a small number of examples, while preserving structural similarity to the input proteins. Wet-lab experiments further validate that AMIX-1 significantly improves the design of high-activity AmeR variants. In addition, we present an evolutionary test-time scaling algorithm that enhances AMIX-1’s applicability to protein-directed evolution, and validate its performance using a variety of *in silico* target metrics, simulating a lab-in-the-loop environment to facilitate future real-world integration.

**Limitation and Future Work** Although AMIX-1 demonstrates promising results, we recognize several limitations in its current form. As the field moves toward protein foundation models that jointly learn from both sequence and structure, AMIX-1 remains a purely sequence-based learning paradigm, leaving a large amount of structural data unutilized, despite growing evidence that structural information significantly enhances protein understanding. Additionally, the test-time scaling algorithm for AMIX-1 has so far only been validated on simulated tasks using *in silico* metrics, lacking experimental assays for real-world verification. As part of future work, we are actively extending AMIX-1 by integrating structural components and applying it to a broader range of real-world experimental assays.

## Contributions

### Project Lead

Jiangtao Feng<sup>1,2,3</sup>

### Scaling Law & Emergent Ability

Changze Lv<sup>1,2,5</sup>, Lihao Wang<sup>1,2</sup>, Dongyu Xue<sup>1,2</sup>

### In-Context Learning & Test-time Scaling

Jiang Zhou<sup>1,2,6,7</sup>, Siyu Long<sup>2,3</sup>, Dongyu Xue<sup>1,2</sup>

### Evaluation

Zherui Zhang<sup>1,8</sup>, Yuchen Cai<sup>1,5</sup>, Zhiqiang Gao<sup>1</sup>

### Wet Experiment

Yu Pei<sup>2,3</sup>, Hao Wang<sup>2,3</sup>, Ziyuan Ma<sup>4</sup>, Shuyi Zhang<sup>4</sup>

### Infrastructure

Lihao Wang<sup>1,2</sup>, Jiakai Hu<sup>1</sup>

### Other Contributors

Chaochen Gao<sup>9</sup>, Jingjing Gong<sup>2,3</sup>, Yuxuan Song<sup>2,3</sup>, Xiaoqing Zheng<sup>5</sup>, Deyi Xiong<sup>6</sup>, Lei Bai<sup>1</sup>, Ya-Qin Zhang<sup>3</sup>, Wei-Ying Ma<sup>3</sup>, Bowen Zhou<sup>1,4</sup>

### Correspondence

Hao Zhou<sup>1,2,3</sup>

### Affiliation

<sup>1</sup>Shanghai Artificial Intelligence Laboratory

<sup>2</sup>Generative Symbolic Intelligence Lab (GenSI), Tsinghua University

<sup>3</sup>Institute for AI Industry Research (AIR), Tsinghua University

<sup>4</sup>Tsinghua University

<sup>5</sup>Fudan University

<sup>6</sup>Tianjin University

<sup>7</sup>Georgia Institute of Technology

<sup>8</sup>Beijing University of Posts and Telecommunications

<sup>9</sup>University of Chinese Academy of Sciences

### Acknowledgments

We thank DeepLink Team for their infrastructure support. This work is supported by Shanghai Artificial Intelligence Laboratory.

## References

- [1] Andrew W Senior, Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin Žídek, Alexander WR Nelson, Alex Bridgland, et al. Improved protein structure prediction using potentials from deep learning. **Nature**, 577(7792):706–710, 2020.
- [2] Mihaly Varadi, Stephen Anyango, Mandar Deshpande, Sreenath Nair, Cindy Natassia, Galabina Yordanova, David Yuan, Oana Stroe, Gemma Wood, Agata Laydon, et al. AlphaFold protein structure database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. **Nucleic acids research**, 50(D1):D439–D444, 2022.
- [3] Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J Ballard, Joshua Bambrick, et al. Accurate structure prediction of biomolecular interactions with alphaFold 3. **Nature**, pages 1–3, 2024.
- [4] Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C Lawrence Zitnick, Jerry Ma, et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. **Proceedings of the National Academy of Sciences**, 118(15):e2016239118, 2021.
- [5] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. **Science**, 379(6637):1123–1130, 2023.
- [6] Tomas Hayes, Roshan Rao, Halil Akin, Nicholas J Sofroniew, Deniz Oktay, Zeming Lin, Robert Verkuil, Vincent Q Tran, Jonathan Deaton, Marius Wiggert, et al. Simulating 500 million years of evolution with a language model. **bioRxiv**, pages 2024–07, 2024.
- [7] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Sal Candido, et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. **bioRxiv**, 2022.
- [8] Ruidong Wu, Fan Ding, Rui Wang, Rui Shen, Xiwen Zhang, Shitong Luo, Chenpeng Su, Zuofan Wu, Qi Xie, Bonnie Berger, et al. High-resolution de novo structure prediction from primary sequence. **BioRxiv**, pages 2022–07, 2022.
- [9] Xiaomin Fang, Fan Wang, Lihang Liu, Jingzhou He, Dayong Lin, Yingfei Xiang, Kunrui Zhu, Xiaonan Zhang, Hua Wu, Hui Li, and Le Song. A method for multiple-sequence-alignment-free protein structure prediction using a protein language model. **Nature Machine Intelligence**, 5(10):1087–1096, October 2023. ISSN 2522-5839. doi: 10.1038/s42256-023-00721-6. URL <http://dx.doi.org/10.1038/s42256-023-00721-6>.
- [10] Chloe Hsu, Robert Verkuil, Jason Liu, Zeming Lin, Brian Hie, Tom Sercu, Adam Lerer, and Alexander Rives. Learning inverse folding from millions of predicted structures. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, **Proceedings of the 39th International Conference on Machine Learning**, volume 162 of **Proceedings of Machine Learning Research**, pages 8946–8970. PMLR, 17–23 Jul 2022. URL <https://proceedings.mlr.press/v162/hsu22a.html>.
- [11] Yaan J Jang, Qi-Qi Qin, Si-Yu Huang, Arun T John Peter, Xue-Ming Ding, and Benoit Kornmann. Accurate prediction of protein function using statistics-informed graph networks. **Nature Communications**, 15(1):6601, 2024.
- [12] Qianmu Yuan, Chong Tian, and Yuedong Yang. Genome-scale annotation of protein binding sites via language model and geometric deep learning. **Elife**, 13:RP93695, 2024.
- [13] Pascal Notin, Mafalda Dias, Jonathan Frazer, Javier Marchena-Hurtado, Aidan N Gomez, Debora Marks, and Yarin Gal. Tranception: protein fitness prediction with autoregressive transformers and inference-time retrieval. In **International Conference on Machine Learning**, pages 16990–17017. PMLR, 2022.
- [14] Jin Su, Chenchen Han, Yuyang Zhou, Junjie Shan, Xibin Zhou, and Fajie Yuan. Saprot: Protein language modeling with structure-aware vocabulary. **bioRxiv**, pages 2023–10, 2023.
- [15] Moritz Glaser and Johannes Brägelmann. ESM-effect: An effective and efficient fine-tuning framework towards accurate prediction of mutation’s functional effect. In **ICLR 2025 Workshop on Machine Learning for Genomics Explorations**, 2025. URL <https://openreview.net/forum?id=kY3z0TMtjU>.

- [16] Bo Ni, David L Kaplan, and Markus J Buehler. Forcegen: End-to-end de novo protein generation based on nonlinear mechanical unfolding responses using a language diffusion model. *Science Advances*, 10(6):eadl4000, 2024.
- [17] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. URL <https://arxiv.org/abs/2005.14165>.
- [18] Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, Weng Lam Tam, Zixuan Ma, Yufei Xue, Jidong Zhai, Wenguang Chen, Peng Zhang, Yuxiao Dong, and Jie Tang. Glm-130b: An open bilingual pre-trained model, 2023. URL <https://arxiv.org/abs/2210.02414>.
- [19] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiro, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emry Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Toootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024. URL <https://arxiv.org/abs/2303.08774>.
- [20] DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei

Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhua Chen, Shaoqing Wu, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wanjia Zhao, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaokang Zhang, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song, Xinxia Shan, Xinyi Zhou, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Yang Zhang, Yanhong Xu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Ying Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu, Yuan Ou, Yuchen Zhu, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yukun Zha, Yunfan Xiong, Yunxian Ma, Yuting Yan, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu Li, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Ziyi Gao, and Zizheng Pan. Deepseek-v3 technical report, 2025. URL <https://arxiv.org/abs/2412.19437>.

- [21] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhusu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhua Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL <https://arxiv.org/abs/2501.12948>.
- [22] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL <https://arxiv.org/abs/2505.09388>.
- [23] Gemini Team and et.al. Gemini: A family of highly capable multimodal models, 2025. URL <https://arxiv.org/abs/2312.11805>.

- [24] Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md. Mostofa Ali Patwary, Yang Yang, and Yanqi Zhou. Deep learning scaling is predictable, empirically, 2017. URL <https://arxiv.org/abs/1712.00409>.
- [25] Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, Eliza Rutherford, Tom Hennigan, Jacob Menick, Albin Cassirer, Richard Powell, George van den Driessche, Lisa Anne Hendricks, Maribeth Rauh, Po-Sen Huang, Amelia Glaese, Johannes Welbl, Sumanth Dathathri, Saffron Huang, Jonathan Uesato, John Mellor, Irina Higgins, Antonia Creswell, Nat McAleese, Amy Wu, Erich Elsen, Siddhant Jayakumar, Elena Buchatskaya, David Budden, Esme Sutherland, Karen Simonyan, Michela Paganini, Laurent Sifre, Lena Martens, Xiang Lorraine Li, Adhiguna Kuncoro, Aida Nematzadeh, Elena Gribovskaya, Domenic Donato, Angeliki Lazaridou, Arthur Mensch, Jean-Baptiste Lespiau, Maria Tsimpoukelli, Nikolai Grigorev, Doug Fritz, Thibault Sottiaux, Mantas Pajarskas, Toby Pohlen, Zhitao Gong, Daniel Toyama, Cyprien de Masson d'Autume, Yujia Li, Tayfun Terzi, Vladimir Mikulik, Igor Babuschkin, Aidan Clark, Diego de Las Casas, Aurelia Guy, Chris Jones, James Bradbury, Matthew Johnson, Blake Hechtman, Laura Weidinger, Jason Gabriel, William Isaac, Ed Lockhart, Simon Osindero, Laura Rimell, Chris Dyer, Oriol Vinyals, Kareem Ayoub, Jeff Stanway, Lorrynne Bennett, Demis Hassabis, Koray Kavukcuoglu, and Geoffrey Irving. Scaling language models: Methods, analysis & insights from training gopher, 2022. URL <https://arxiv.org/abs/2112.11446>.
- [26] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models, 2022. URL <https://arxiv.org/abs/2203.15556>.
- [27] Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters, 2024. URL <https://arxiv.org/abs/2408.03314>.
- [28] Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling, 2024. URL <https://arxiv.org/abs/2407.21787>.
- [29] Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. Inference scaling laws: An empirical analysis of compute-optimal inference for problem-solving with language models, 2025. URL <https://arxiv.org/abs/2408.00724>.
- [30] Alex Graves, Rupesh Kumar Srivastava, Timothy Atkinson, and Faustino Gomez. Bayesian flow networks. **arXiv preprint arXiv:2308.07037**, 2023.
- [31] Yanru Qu, Keyue Qiu, Yuxuan Song, Jingjing Gong, Jiawei Han, Mingyue Zheng, Hao Zhou, and Wei-Ying Ma. Molcraft: Structure-based drug design in continuous parameter space. In **International conference on machine learning**, 2024.
- [32] Baris E Suzek, Hongzhan Huang, Peter McGarvey, Raja Mazumder, and Cathy H Wu. Uniref: comprehensive and non-redundant uniprot reference clusters. **Bioinformatics**, 23(10):1282–1288, 2007.
- [33] Baris E Suzek, Yuqi Wang, Hongzhan Huang, Peter B McGarvey, Cathy H Wu, and UniProt Consortium. Uniref clusters: a comprehensive and scalable alternative for improving sequence similarity searches. **Bioinformatics**, 31(6):926–932, 2015.
- [34] Sarah Alabdari, Nitya Thakkar, Rianne van den Berg, Alex X Lu, Nicolo Fusi, Ava P Amini, and Kevin K Yang. Protein generation with evolutionary diffusion: sequence is all you need. **BioRxiv**, pages 2023–09, 2023.
- [35] A Vaswani. Attention is all you need. **Advances in Neural Information Processing Systems**, 2017.
- [36] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. **Neurocomputing**, 568:127063, 2024.
- [37] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In **International Conference on Learning Representations**, 2018.
- [38] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. **Journal of Machine Learning Research**, 24(240):1–113, 2023.

- [39] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. **Journal of Machine Learning Research**, 25(70):1–53, 2024.
- [40] Zhengyang Liang, Hao He, Ceyuan Yang, and Bo Dai. Scaling laws for diffusion transformers. **arXiv preprint arXiv:2410.08184**, 2024.
- [41] Rahul Ravishankar, Zeeshan Patel, Jathushan Rajasegaran, and Jitendra Malik. Scaling properties of diffusion models for perceptual tasks. In **Proceedings of the Computer Vision and Pattern Recognition Conference**, pages 12945–12954, 2025.
- [42] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. **arXiv preprint arXiv:2303.08774**, 2023.
- [43] Peter J Huber. Robust estimation of a location parameter. In **Breakthroughs in statistics: Methodology and distribution**, pages 492–518. Springer, 1992.
- [44] Jorge Nocedal. Updating quasi-newton matrices with limited storage. **Mathematics of computation**, 35(151):773–782, 1980.
- [45] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. **arXiv preprint arXiv:2206.07682**, 2022.
- [46] Zhengxiao Du, Aohan Zeng, Yuxiao Dong, and Jie Tang. Understanding emergent abilities of language models from the loss perspective. **arXiv preprint arXiv:2403.15796**, 2024.
- [47] Christian B Anfinsen. Principles that govern the folding of protein chains. **Science**, 181(4096):223–230, 1973.
- [48] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. **nature**, 596(7873):583–589, 2021.
- [49] Xavier Robin, Juergen Haas, Rafal Gumienny, Anna Smolinski, Gerardo Tauriello, and Torsten Schwede. Continuous automated model evaluation (cameo)—perspectives on the future of fully automated evaluation of structure prediction methods. **Proteins: Structure, Function, and Bioinformatics**, 89(12):1977–1986, 2021.
- [50] Xinyou Wang, Zaixiang Zheng, Fei Ye, Dongyu Xue, Shujian Huang, and Quanquan Gu. Diffusion language models are versatile protein learners. **arXiv preprint arXiv:2402.18567**, 2024.
- [51] Timothy Atkinson, Thomas D. Barrett, Scott Cameron, Bora Guloglu, Matthew Greenig, Louis Robinson, Alex Graves, Liviu Copoiu, and Alexandre Laterre. Protein sequence modelling with bayesian flow networks. **bioRxiv**, 2024. doi: 10.1101/2024.09.24.614734. URL <https://www.biorxiv.org/content/early/2024/09/26/2024.09.24.614734>.
- [52] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Tianyu Liu, et al. A survey on in-context learning. **arXiv preprint arXiv:2301.00234**, 2022.
- [53] Jingjing Gong, Yu Pei, Siyu Long, Yuxuan Song, Zhe Zhang, Wenhao Huang, Ziyao Cao, Shuyi Zhang, Hao Zhou, and Wei-Ying Ma. Steering protein family design through profile bayesian flow. In **The Thirteenth International Conference on Learning Representations, ICLR**. OpenReview.net, 2025.
- [54] Tianhao Yu, Haiyang Cui, Jianan Canal Li, Yunan Luo, Guangde Jiang, and Huimin Zhao. Enzyme function prediction using contrastive learning. **Science**, 379(6639):1358–1363, 2023.
- [55] Peter Mikhael, Itamar Chinn, and Regina Barzilay. Clipzyme: Reaction-conditioned virtual screening of enzymes. In **Forty-first International Conference on Machine Learning**, 2024.
- [56] Brynne C Stanton, Alec AK Nielsen, Alvin Tamsir, Kevin Clancy, Todd Peterson, and Christopher A Voigt. Genomic mining of prokaryotic repressors for orthogonal logic gates. **Nature chemical biology**, 10(2):99–105, 2014.
- [57] Ziyuan Ma, Wenjie Li, Yunhao Shen, Yunxin Xu, Gengjiang Liu, Jiamin Chang, Zeju Li, Hong Qin, Boxue Tian, Haipeng Gong, et al. Evoai enables extreme compression and reconstruction of the protein sequence space. **Nature Methods**, 22(1):102–112, 2025.

- [58] OpenAI and et.al. Openai o1 system card, 2024. URL <https://arxiv.org/abs/2412.16720>.
- [59] Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling, 2025. URL <https://arxiv.org/abs/2501.19393>.
- [60] Yajie Wang, Pu Xue, Mingfeng Cao, Tianhao Yu, Stephan T Lane, and Huimin Zhao. Directed evolution: methodologies and applications. **Chemical reviews**, 121(20):12384–12444, 2021.
- [61] Jason Yang, Ravi G Lal, James C Bowden, Raul Astudillo, Mikhail A Hameedi, Sukhvinder Kaur, Matthew Hill, Yisong Yue, and Frances H Arnold. Active learning-assisted directed evolution. **Nature Communications**, 16(1):714, 2025.
- [62] Kaiyi Jiang, Zhaoqing Yan, Matteo Di Bernardo, Samantha R Sgrizzi, Lukas Villiger, Alisan Kayabolen, BJ Kim, Josephine K Carscadden, Masahiro Hiraizumi, Hiroshi Nishimasu, et al. Rapid in silico directed evolution by a protein language model with evolvepro. **Science**, page eadr6006, 2024.
- [63] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. **Advances in neural information processing systems**, 35:24824–24837, 2022.
- [64] Sizhe Qiu, Bozhen Hu, Jing Zhao, Weiren Xu, and Aidong Yang. Seq2topt: a sequence-based deep learning predictor of enzyme optimal temperature. **Briefings in Bioinformatics**, 26(2):bbaf114, 2025.
- [65] Sizhe Qiu. Seq2topt. <https://github.com/SizheQiu/Seq2Topt>, 2023. commit ab12cd3, accessed 2025-07-04.
- [66] Thanh VT Tran and Truong Son Hy. Protein design by directed evolution guided by large language models. **IEEE Transactions on Evolutionary Computation**, 2024.

# Appendix

## A Pretrainig Algorithm

### A.1 Training

The continuous-time discrete Bayesian flow loss, following the ProfileBFN framework [53], is adopted for model training.

$$\mathcal{L}(\phi) = \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}} \mathbb{E}_{\prod_{i=1}^n p_s(\mathbf{y}_i | \mathbf{x}, \alpha_i)} D_{KL}(p_s(\mathbf{y}_i | \mathbf{x}, \alpha_i) || p_r(\mathbf{y}_i | \boldsymbol{\theta}_{i-1}, \phi, \alpha_i)) \quad (10)$$

$$= \frac{1}{2} \beta'(t) K \|\phi(\boldsymbol{\theta}_{i-1}) - \mathbf{e}(\mathbf{x})\|^2 \quad (11)$$

$$= \beta_1 t K \|\phi(\boldsymbol{\theta}_{i-1}) - \mathbf{e}(\mathbf{x})\|^2 \quad (12)$$

---

#### Algorithm 1 Discrete Bayesian Flow Loss

---

**Require:** data points  $\mathbf{x}$ , output distribution neural network  $\phi$ ,

**Ensure:**  $\mathcal{L}(\phi)$

- 1:  $t \sim U(0, 1)$
  - 2:  $\mathbf{y}_t \sim \mathcal{N}(\beta_1 t^2 (\mathbf{K}\mathbf{e}(\mathbf{x}) - \mathbf{1}), \beta_1 t^2 K \mathbf{I})$
  - 3:  $\boldsymbol{\theta}_t \leftarrow \text{softmax}(\mathbf{y}_t)$
  - 4:  $\mathcal{L}(\phi) = \beta_1 t K \|\phi(\boldsymbol{\theta}_t) - \mathbf{e}(\mathbf{x})\|^2$
  - 5: **return**  $\mathcal{L}(\phi)$
- 

### A.2 Inference

---

#### Algorithm 2 Noise Reduced Sampling

---

**Require:**  $N$  steps, output distribution neural network  $\phi$

**Ensure:** inference sample  $\mathbf{x}$

- 1:  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  - 2:  $\mathbf{y}^{(0)} \leftarrow \mathbf{0}$
  - 3: **for**  $i = 0 \dots N - 1$  **do**
  - 4:      $t \leftarrow \frac{i}{N}$   $j \leftarrow \frac{i+1}{N}$
  - 5:      $\boldsymbol{\theta}_t \leftarrow \text{softmax}(\mathbf{y}_t)$
  - 6:      $\beta(t) \leftarrow \beta_1 t^2$
  - 7:      $\mathbf{y}_j \leftarrow \beta(j) (K\phi(\boldsymbol{\theta}_t) - \mathbf{1}) + \sqrt{K\beta(j)} \mathbf{z}$
  - 8: **end for**
  - 9:  $\boldsymbol{\theta}_1 \leftarrow \text{softmax}(\mathbf{y}_1)$
  - 10:  $\mathbf{x} = \text{argmax}(\phi(\boldsymbol{\theta}^{(1)}))$
  - 11: **return**  $\mathbf{x}$
-

## B Performance of Prediction Tasks

**Table 3** AMIX-1 demonstrates robust capabilities across diverse protein prediction tasks, indicating a strong comprehension of protein. To ensure a fair comparison, we evaluated various models with 650 million parameters, all initialized from ESM2 weights. \*: protein structure is provided. †: results are quoted from SaProt [14]. The best scores are highlighted in **bold** and the second-best scores are underlined.

Model(650M)	Thermostability	HumanPPI	Metal Ion Binding	EC	GO			DeepLoc	
					Spearman’s $\rho$	ACC(%)	ACC(%)	Fmax	Subcellular
								Fmax	Binary
ESM-2 †	0.680	76.67	71.56	0.877	0.668	0.345	0.411	82.09	91.96
SaProt*	<b>0.724</b>	86.41	<b>75.75</b>	<b>0.884</b>	0.678	0.356	0.414	<b>85.57</b>	<u>93.55</u>
DPLM	0.695	86.41	<u>75.15</u>	0.875	<u>0.680</u>	0.357	0.409	84.56	93.09
DPLM-2	0.714	84.44	74.28	0.878	<u>0.680</u>	<u>0.359</u>	0.411	82.98	<b>93.64</b>
AMIX-1	<u>0.718</u>	<b>89.13</b>	71.13	0.875	<b>0.682</b>	<b>0.495</b>	<b>0.482</b>	<u>84.62</u>	93.31

## C Details of CAMEO-20

In this section, we will show detailed information about CAMEO-20, which we applied to evaluate the performance of AMIX-1 models on downstream tasks.

**Table 4** Detailed information of each protein for the CAMEO-20 dataset, partially from ProfileBFN [53]

PDB	Chain	Length	Title
8F9R	A	501	Rabbit sialic acid esterase
8JDH	A	166	Crystal structure of anti-CRISPR AcrIF25
8JGO	A	535	Crystal structure of Deinococcus radiodurans exopolyphosphatase
8JRB	A	597	Structure of DNA polymerase I from Aquifex pyrophilus
8QL0	A	693	Structure of human PAD6 Phosphomimic mutant V10E/S446E, apo
8QVC	B	100	Deinococcus aerius TR0125 C-glycosyl deglycosidase (CGD), wild type crystal cryoprotected with glycerol
8S4S	A	145	PrgE from plasmid pCF10
8SUF	A	1007	The complex of TOL-1 ectodomain bound to LAT-1 Lectin domain
8SUF	E	114	The complex of TOL-1 ectodomain bound to LAT-1 Lectin domain
8SW5	C	47	Protein Phosphatase 1 in complex with PPI-specific Phosphatase targeting peptide (PhosTAP) version 1
8UAI	B	494	Crystal structure of hetero hexameric hazelnut allergen Cor a 9
8V8P	A	231	Sorghum Chalcone Isomerase
8WEX	A	468	Crystal structure of N-acetyl sugar amidotransferase from Legionella pneumophila
8WG0	D	100	Crystal structure of GH97 glucodextranase from Flavobacterium johnsoniae in complex with glucose
8WOP	A	100	Crystal structure of Arabidopsis thaliana UDP-glucose 4-epimerase 2 (AtUGE2) complexed with UDP, wild-type
8WTB	B	187	Crystal structure of McsA/McsB complex truncated by chymotrypsin
8XJE	B	153	Crystal structure of the YqeY protein from Campylobacter jejuni
8XJG	A	153	Crystal structure of the YqeY protein from Vibrio parahaemolyticus
9B1R	A	562	Functional implication of the homotrimeric multidomain vacuolar sorting receptor 1 from Arabidopsis thaliana
9BCZ	A	644	Chicken 1-phosphatidylinositol 4,5-bisphosphate phosphodiesterase zeta-1 (PLCZ1) in complex with calcium and phosphorylated threonine
9F63	A	572	Crystal structure of Saccharomyces cerevisiae pH nine-sensitive protein 1 (PNS1)

## D Further Discussion on Scaling Laws

### D.1 Unscalability when Extreme Noise Levels

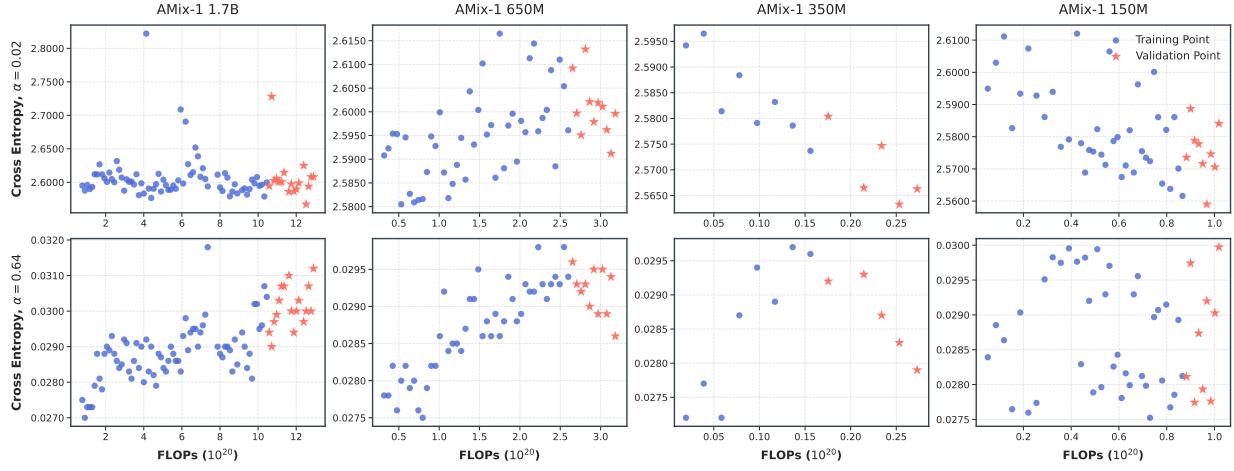
We show the variation of cross-entropy  $\mathcal{L}_{CE}$  under noise level  $\alpha = 0.02$  and  $0.64$  in Figure 13, and we find that there is no scaling behavior at all on these settings.

### D.2 Mean Relative Error

We utilize the mean relative error (MRE) to assess the goodness of scaling laws, defined as:

$$\text{Fitting MRE} = \frac{1}{|\mathcal{D}_{\text{fit}}|} \sum_{(x,y) \in \mathcal{D}_{\text{fit}}} \left| \frac{f_{\theta}(x) - y}{y} \right|, \quad \text{Valid MRE} = \frac{1}{|\mathcal{D}_{\text{val}}|} \sum_{(x,y) \in \mathcal{D}_{\text{val}}} \left| \frac{f_{\theta}(x) - y}{y} \right|. \quad (13)$$

Here,  $f_{\theta}(x)$  is the predicted loss given model size  $N$  and training tokens  $D$ , and  $y$  is the actual cross-entropy.



**Figure 13** Cross entropy and FLOPs may be unpredictable under several noise levels.

### D.3 Parametric Scaling Laws under Several Noise Levels

In addition to the noise levels reported in [Table 2](#), we evaluate a broader range of noise settings, as summarized in [Table 5](#). These additional results further reinforce our conclusion that AMIX-1 models exhibit consistent scalability when trained under appropriately chosen noise levels.

**Table 5** Parametric scaling laws under noise level  $\alpha \in \{0.06, 0.08, 0.48, \text{ and } 0.64\}$ . Note that both  $N$  and  $D$  are measured in millions.

Noise Level	Parameters					MRE (%)		Scalable
	$E$	$A$	$B$	$n$	$d$	Fitting	Valid	
0.06	1.4764	0.5914	5.4300	0.5687	0.2865	0.55	1.84	✓
0.08	1.1630	0.0187	3.1436	0.0163	0.2093	0.66	1.28	✓
0.48	0.0583	0.0231	0.3050	0.2006	0.2563	1.10	1.82	✓
0.64	0.0281	0.0000	0.5247	0.8930	2.6603	1.98	5.41	✗

## E Experimental Details of EvoAMix-1

**Experimental Setup** The formal description of EvoAMIX-1 is provided in [Algorithm 3](#). Building upon this framework, we evaluate EVOAMIX-1 on six diverse in silico protein design tasks, encompassing structure-guided, biophysical, and functional objectives. Unless otherwise noted, all experiments share a unified optimization configuration: 10 total iterations ( $T = 10$ ), 50 candidates(batch size) generated per round ( $N = 50$ ), and top- $k$  selection size  $k = 5$ .

**Task-Specific Settings** Each task is implemented with a distinct evaluation metric and verifier model. [Table 6](#) summarizes the datasets used, verifier models, and key hyperparameters. For each task, we report mean top- $k$  from cumulative generated proteins.

**Reward Function Definitions** The reward function  $R(\cdot)$  is defined separately for each task to reflect task-specific design objectives. For the Orphan Design and General Design tasks,  $R$  is the TM-score between the ESMFold-predicted structure of a generated sequence and the known structure of the input wild-type protein. For the Optimal Temperature and Optimal pH tasks, the design objective is to match target physiological conditions—specifically 37°C and pH 7, respectively. In these cases,  $R(x) = -|f(x) - y^*|$ , where

**Table 6** Summary of task configurations and hyperparameters.

Task	Input Samples	Verifier Model	Inference Steps	Initial Noise Level
Orphan Design	T1026, T1033, T1039, T1043, T1064, T1074	TM-score (ESMFold)	50	0.99
General Design	8JGO_A, 8SW5_C, 8WOP_A, 8XJE_B, 9B1R_A	TM-score (ESMFold)	50	0.90
Optimal Temperature	an in-house protein	Seq2Topt [64]	250	0.99
Optimal pH	an in-house protein	Seq2pHopt [65]	250	0.99
Enzyme EC Matching	F2XF95, F2XF96, F2XFA8, R9QMQ9, R9QMR0, R9QMR3, R9QMW1, R9QMW5, R9QMY8, R9QMY9	CLEAN [54]	50	0.99
Reaction Activity	A0A0A7E972, B7X755, O42275, P18142, P32749	CLIPZyme [55]	100	0.99

---

### Algorithm 3 EVOAMIX-1

---

**Require:** Initial MSA  $\mathbf{X}$ , batch size  $N$ , number of rounds  $T$ , selection size  $k$ , external verifier  $R(\cdot)$

**Ensure:** Top- $k$  sequences from full generation history, ranked by verifier

```

1:  $\mathbf{P}^{(0)} \leftarrow \text{BuildProfile}(\mathbf{X})$                                  $\triangleright$  Convert MSA to profile
2:  $\mathcal{H} \leftarrow \emptyset$                                                   $\triangleright$  Initialize history of generated sequences
3: for  $t = 0$  to  $T - 1$  do
4:    $\{\mathbf{x}_i^{(t)}\}_{i=1}^N \leftarrow \text{AMIX-1}(\mathbf{P}^{(t)})$                  $\triangleright$  Generate  $N$  sequences conditioned on profile
5:   for  $i = 1$  to  $N$  do
6:      $s_i^{(t)} \leftarrow R(\mathbf{x}_i^{(t)})$                                           $\triangleright$  Compute score
7:      $\mathcal{H} \leftarrow \mathcal{H} \cup (\mathbf{x}_i^{(t)}, s_i^{(t)})$                           $\triangleright$  Add to generation history
8:   end for
9:    $\mathcal{X}^{(t)} \leftarrow \text{Top-}k(\mathcal{H})$                                       $\triangleright$  Select top- $k$  sequences so far
10:   $\mathbf{P}^{(t+1)} \leftarrow \text{BuildProfile}(\mathcal{X}^{(t)})$                           $\triangleright$  Update profile from selected sequences
11: end for
12: return  $\text{Top-}k(\mathcal{H})$                                                $\triangleright$  Final output: best sequences across all rounds

```

---

$f(x)$  is the output of the predictor (Seq2Topt or Seq2pHopt) and  $y^*$  is the target value. For the Enzyme EC Matching task, the initial inputs are proteins predicted by CLEAN to belong to EC class 4.2.3.120, and the goal is to evolve sequences toward EC class 4.2.3.113. Here,  $R(x)$  is the classification confidence assigned by CLEAN to the target EC class. Finally, for the Reaction Activity task, the goal is to maximize the predicted catalytic competence toward the target reaction  $[\text{CH}_3][\text{N}^+](\text{CH}_3)(\text{CH}_3)\text{CH}_2\text{CHO} + \text{H}_2\text{O} + \text{O}_2 \longrightarrow [\text{CH}_3][\text{N}^+](\text{CH}_3)(\text{CH}_3)\text{CH}_2\text{COOH} + \text{H}_2\text{O}_2$  as evaluated by CLIPZyme, and  $R(x)$  is defined as the model’s confidence score for this specific transformation.

**Baseline Adaptation** To ensure a fair comparison with baseline methods (ALDE, EVOLVEpro, MLDE), we modify their original frameworks by simply replacing their internal fitness predictors or experimental feedback modules with a shared external verifier oracle  $R(\cdot)$ . This harmonization ensures that all methods receive identical feedback signals across evolution process, isolating the effect of optimization strategies from feedback quality.

For each baseline, we simulate directed evolution over total 500 variants. After each round (50 proteins are considered as a round), we compute the cumulative top- $k$  mean score using the shared verifier, mirroring the evaluation protocol used for EVOAMIX-1. This allows us to compare the evolution trajectories and final design quality under a consistent verification budget and objective.

## F EVOAMIX-1 Ablation Study

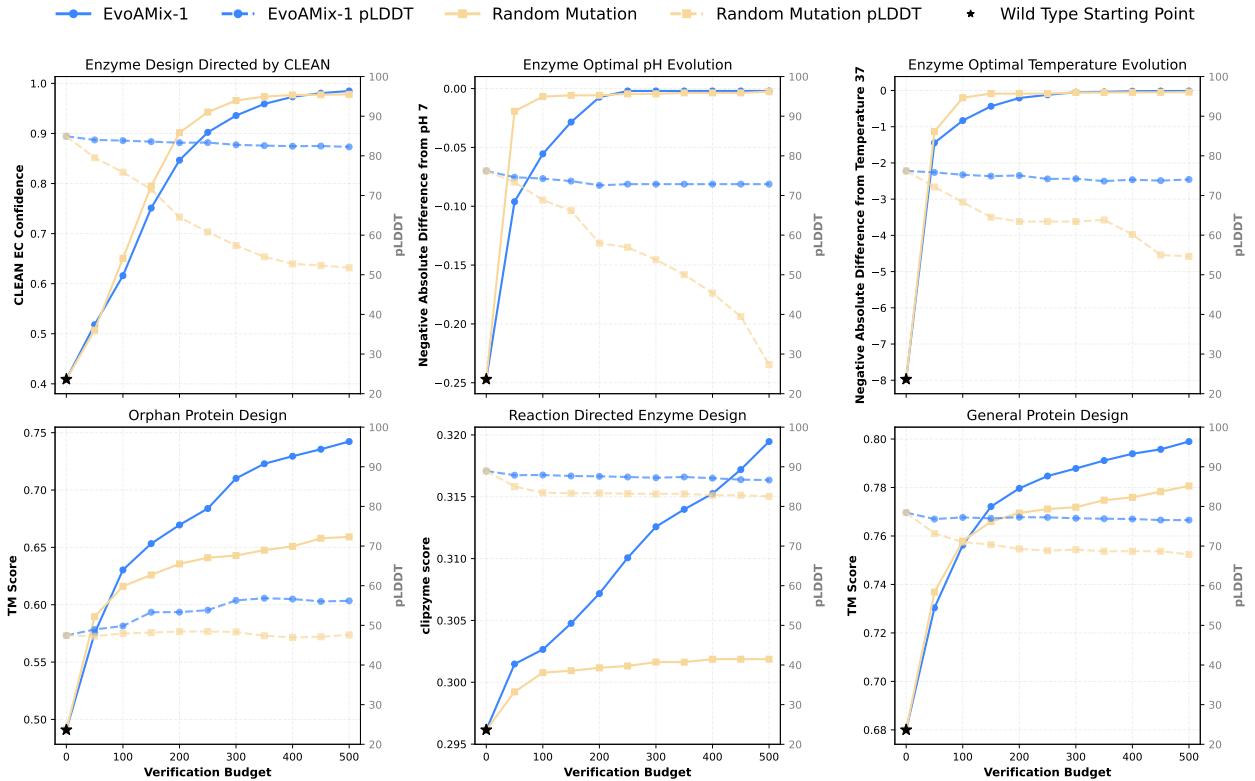
### F.1 Assessing the Role of the Generative Model in Test-Time Scaling

To assess the necessity of using a pretrained protein generative model in EVOAMIX-1, we conduct an ablation study in which we replace AMIX-1 with a simple stochastic generator based on random mutation. Specifically, at each round of test-time scaling, we randomly perturb 10% of the residues in the top-1 sequences, uniformly sampling new amino acids at mutated positions. This generates a new candidate pool without relying on a trained protein foundation model. All other components of the TTS loop—including the verifier, profile update mechanism, and sampling procedure—remain unchanged.

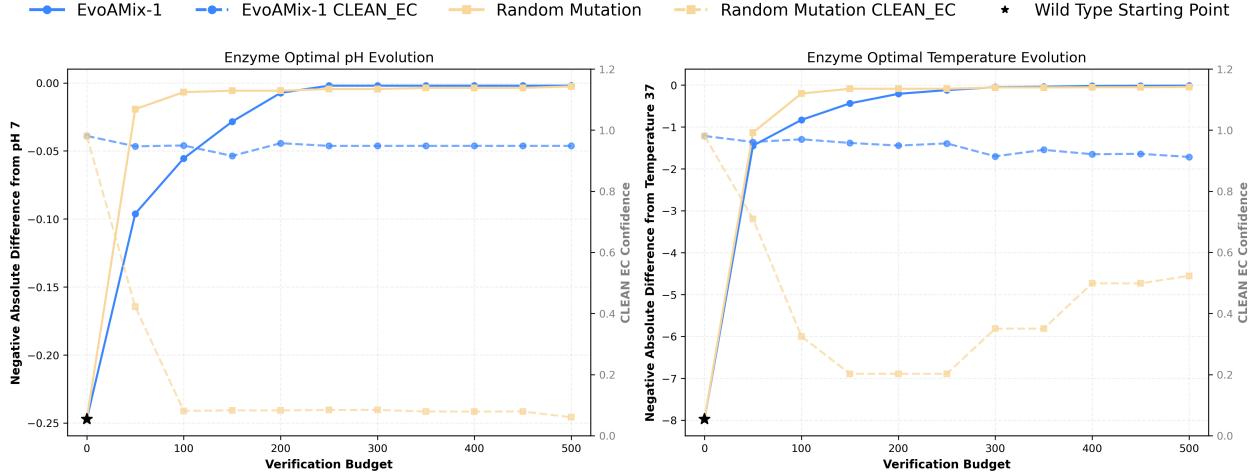
**Figure 14** compares the performance of this random mutation variant against the full EvoAMix-1 across six directed evolution benchmarks. On certain objectives—such as optimal temperature and pH—random mutation initially shows comparable or even slightly better performance than EvoAMix-1 in terms of the target reward metric. However, this superficial gain comes at a critical cost: the predicted structural quality (measured by pLDDT) of generated proteins degrades sharply over time. By contrast, EvoAMix-1 maintains consistently high pLDDT scores throughout the optimization process, despite not receiving any explicit structural feedback during evolution.

To further probe the functional integrity of generated proteins, we evaluated whether the optimized sequences retained the original enzymatic function class, as measured by CLEAN confidence scores with respect to the input EC number. **Figure 15** shows the CLEAN prediction scores during pH and temperature optimization. While EvoAMix-1 preserves near-perfect EC predictions across all iterations, random mutation causes CLEAN confidence to deteriorate progressively—indicating a loss of functional specificity. This difference is particularly notable given that no CLEAN-based supervision was used in either setting, further underscoring the inductive bias encoded in the AMIX-1 generative prior.

These results highlight that naïve mutation-based exploration—even when embedded in a TTS framework—can rapidly drift away from biologically plausible regions of sequence space. In contrast, the structured sampling behavior of AMIX-1 ensures that sequence modifications remain compatible with both structure and function constraints. We conclude that a pretrained foundation model is essential for ensuring biological fidelity during test-time scaling.



**Figure 14** Ablation study: structural integrity of sequences generated by EvoAMix-1 versus random mutation. Each curve shows the trajectory of the task reward (solid line, left y-axis) and mean pLDDT (dashed line, right y-axis) over increasing verifier calls. While random mutation achieves comparable or even higher reward scores in some tasks, it causes a rapid decline in structural confidence. In contrast, EvoAMix-1 maintains consistently high pLDDT values throughout optimization, despite the absence of any explicit structure-based supervision.



**Figure 15** Ablation study: functional consistency of generated sequences under pH and temperature optimization. We measure CLEAN-predicted EC confidence scores (y-axis) over increasing verifier calls. EvoAMIX-1 consistently retains the enzymatic identity of the starting protein (i.e., original EC number), even without using CLEAN as a part of reward signal. In contrast, random mutation degrades functional consistency over time, indicating that it drifts away from the original functional class.

## F.2 Assessing the Impact of Iterative Condition Refinement

To isolate the impact of iterative condition adaptation in EvoAMIX-1, we conduct an ablation in which the profile prompt  $\mathbf{P}^{(t)}$  is fixed throughout the entire optimization process. Specifically, after computing the initial profile  $\mathbf{P}^{(0)}$  from the wild-type MSA, all subsequent generations are conditioned on this static prompt, regardless of the quality of sequences discovered during evolution. In other words, we remove the feedback loop between the verifier and the generative model, breaking the propose–verify–refine cycle.

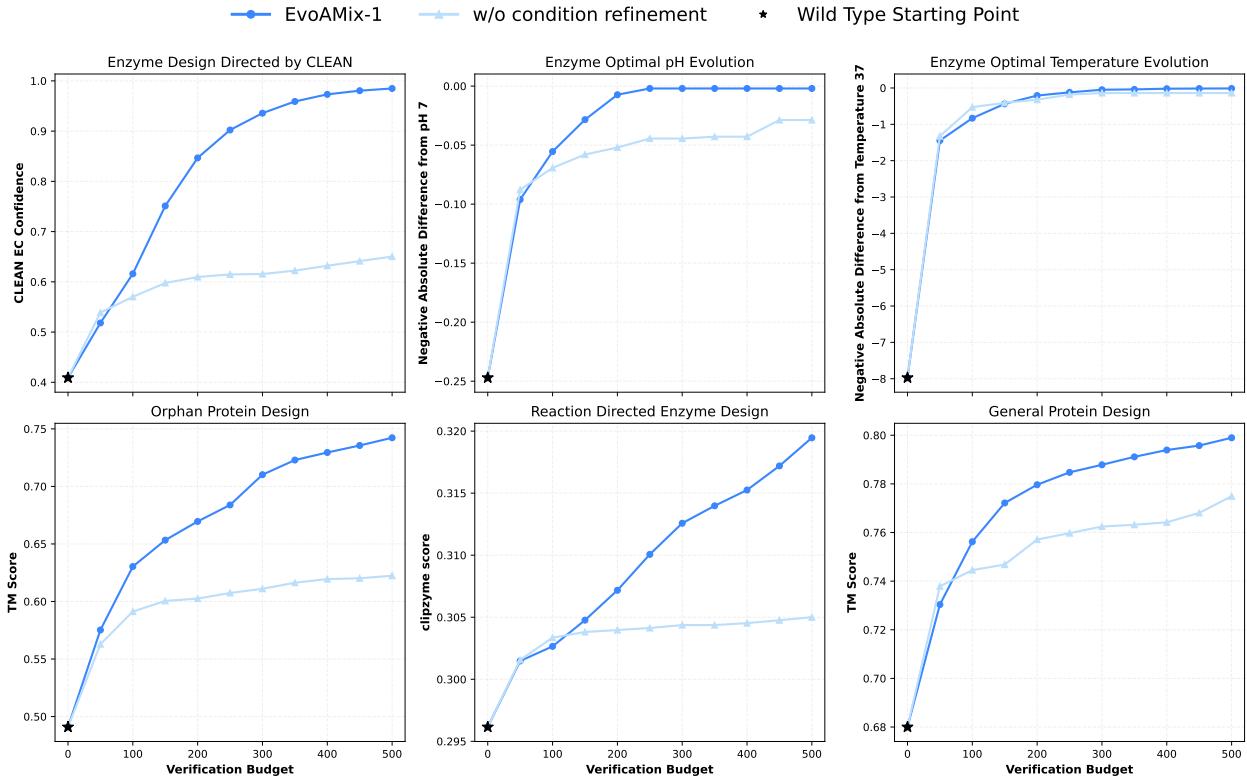
Figure 16 summarizes the result across all six design tasks. Compared to full EvoAMIX-1, the fixed-profile variant consistently underperforms, both in terms of final reward and convergence speed. These results demonstrate that iterative prompt refinement is a key driver of performance gains in TTS: by updating the proposal distribution to reflect high-quality sequences discovered so far, the model is progressively steered toward regions of sequence space favored by the verifier. Without this adaptive mechanism, the optimization stagnates, highlighting the importance of dynamic conditioning even in the absence of parameter updates.

## G Verification in wet-lab

As silico experiments has shown AMIX-1’s effectiveness in protein-direct evolution and test-time scaling. In this section, wet-lab experiments are included to validate such a generative model to produce functional proteins. We selected the AmeR protein as a target and aimed to optimize its DNA-binding affinity. We employed the model to generate a library of AmeR mutants, whose fitness was subsequently verified through wet-lab experiments.

The results demonstrated that model-generated mutants enhanced the protein’s activity by up to 50-fold compared to the wild type. This represents an approximate 77% improvement over the best previously reported results. These findings confirm the model’s ability to reliably generate novel, functional protein sequences. This capability allows for the simulation of directed evolution, which can significantly accelerate the experimental process while reducing the associated costs and labor.

**Wet-lab Backgrounds.** The AmeR protein, a member of the TetR family of transcriptional regulators, is central to this study. In synthetic biology, AmeR is widely utilized as an orthogonal biological logic gate, where it represses gene expression by binding to its corresponding promoter[56]. A key challenge in



**Figure 16** Ablation study: removing iterative profile refinement during test-time scaling. In this variant, EvoAMix-1 is executed without updating the profile  $\mathbf{P}^{(t)}$  across rounds—i.e., all generations are conditioned on the initial profile  $\mathbf{P}^{(0)}$  derived from the wild-type sequence. Across all tasks, this non-adaptive variant leads to significantly worse final performance and slower convergence, highlighting the essential role of iterative prompt refinement in guiding the generation process toward regions of higher fitness.

engineering AmeR is the scarcity of its homologs, which results in a vast and underexplored mutational landscape, complicating targeted functional optimization.

This study aims to employ our generative model to produce AmeR variants with enhanced DNA-binding affinity. These improved variants are envisioned as more reliable molecular switches for constructing complex and predictable synthetic biology systems, with potential applications in metabolic engineering, biosensors, and cell therapy.

**Wet-lab Methods.** To comprehensively explore the sequence-function landscape of AmeR and identify mutants with enhanced fitness, we employed our generative model, AMIX-1. The model was conditioned on a profile constructed from the wild-type AmeR sequence, its multiple sequence alignment (MSA), and a set of previously characterized variants. Using the 650M-parameter version of our model (AMIX-1-650M), we performed conditional sampling. The generated sequences were then subjected to similarity filtering based on specific hyper-parameter settings, yielding a final set of 40 candidate sequences, each containing ten or fewer mutations relative to the wild-type.

To quantify the DNA-binding affinity of AmeR mutants, we established a fluorescent reporter assay adapted from EvoAI[57] in the wet lab. In this system, the DNA-binding activity of AmeR is inversely coupled to the expression of an enhanced yellow fluorescent protein (eYFP). Specifically, high-affinity binding of an AmeR variant to its corresponding operator (pAmeR) represses eYFP transcription, leading to a low fluorescence signal. Conversely, weak or no binding results in high eYFP expression and a strong signal. The expression of the AmeR variants was placed under the control of an IPTG-inducible pTac promoter. We quantified the

repressive strength using the metric “fold repression” calculated as follows:

$$\text{Fold Repression} = \frac{\text{Fluorescence}_{[\text{uninduced}]} - \text{Fluorescence}_{[\text{background}]}}{\text{Fluorescence}_{[\text{induced}]} - \text{Fluorescence}_{[\text{background}]}} \quad (14)$$

Here, background fluorescence was measured from a control strain carrying an empty plasmid. This metric is positively correlated with the DNA-binding affinity of AmeR, representing its efficacy as a transcriptional “off-switch”.

To acquire the data, engineered strains harboring each mutant were cultured, and the fluorescence signal was measured via flow cytometry to determine the fold repression for each variant and the wild type.

**Baselines.** The performance of our model-generated mutants was evaluated against three key baselines:

- **Wild-Type (WT):** The original, unmodified native AmeR sequence.
- **Single-Mutant Library:** A collection of previously characterized AmeR variants, each containing a single amino-acid substitution relative to the wild-type.
- **EvoAI Variants:** Mutants generated by EvoAI[57], a previously reported evolutionary method. This method identifies beneficial single mutations (anchors) from local fitness peaks and then combinatorially introduces them to create new sequences with seven or fewer substitutions.

**Wet-lab results.** Our results indicate that AMIX-1 explores the sequence space more effectively than existing methods. As illustrated by our experimental pipeline in [Figure 8](#), AMIX-1 enables the controlled introduction of both single and multiple mutations. A summary of the peak fold repression values achieved by all methods is presented in [Figure 9](#). Notably, AMIX-1-designed variants demonstrated a significant improvement over all baselines. The top-performing AMIX-1 variant, which contained 10 mutations, exhibited a fold repression nearly double that of the best EvoAI variant (7 mutations).

These findings underscore AMIX-1’s superior ability to navigate the sequence-function landscape to identify high-impact mutations and discover highly active variants.