GGBOUND: GENOME-GROUNDED REASONING FOR LIFE-BOUNDARY PHENOTYPE PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Predicting microbial life-boundary phenotypes from genomic sequence offers a scalable alternative to costly experimental screening, but remains challenging under sparse genomic observations and diverse phenotype types. Existing approaches either rely on predefined genomic features or perform direct genome-to-language prediction, leaving complementary predictive signals and intermediate phenotype proposals underused. In this paper, we formulate the task as *proposal-conditioned phenotype refinement*, where genomic evidence is used to refine the proposals. We characterize proposal-conditioned genomic utility, showing that effective genomic representations should retain phenotype-relevant information while suppressing irrelevant variation. Guided by this analysis, we introduce GGBound, which learns phenotype-specific genomic representations and grounds them in a language decoder through phenotype-oriented reasoning supervision, yielding interpretable rationales alongside predictions. We additionally introduce Sufficiency-to-Decision GRPO to strengthen the use of genomic evidence beyond auxiliary proposals during refinement. We further construct a strain-centric benchmark spanning 17 microbial life-boundary phenotypes from IJSEM strains and BacDive annotations. Experiments show that GGBound consistently outperforms genome-based baselines across phenotype categories and even surpasses frontier LLMs using strain names to leverage organism-specific knowledge on most phenotype groups.

1 INTRODUCTION

Characterizing the physiological life boundaries of microbial strains, such as temperature, pH, and salinity conditions permitting growth, is fundamental to biotechnology and microbial ecology (Rothschild & Mancinelli, 2001; Harrison et al., 2013; Merino et al., 2019). However, such phenotypes are traditionally determined through extensive in vitro screening (Nichols et al., 2010; Lagier et al., 2018). The growing availability of microbial genomic data therefore offers a promising opportunity to infer physiological traits directly from sequence (Sayers et al., 2021), motivating increasing efforts toward genome-based phenotype prediction.

Early genome-based approaches typically rely on predefined genomic features and phenotype-specific predictors, using signals such as protein-family profiles (Weimann et al., 2016) or phenotype-associated k-mers (Aun et al., 2018) to infer microbial traits. Biological foundation models offer a more general alternative by learning transferable representations directly from sequence (Dalla-Torre et al., 2025; Nguyen et al., 2023; 2024; Brixi et al., 2026). Recent systems further connect these representations with LLMs for unified biological prediction and reasoning. ChatNT formulates diverse genomics tasks within a sequence-to-language framework (de Almeida et al., 2025), while BioReason enables multi-step reasoning over genomic representations (Fallahpour et al., 2025).

As LLMs continue to advance, their ability to synthesize diverse evidence enables increasingly flexible reasoning across knowledge-intensive tasks (Yadav et al., 2026). However, existing genome-to-language frameworks largely retain a direct genome-to-output formulation (Fallahpour et al., 2025), while microbial phenotype predictors typically rely on genome-wide features. As both paradigms primarily focus on producing phenotype predictions, how genomic evidence can support intermediate phenotype proposals and enable their verification and refinement remains largely unexplored.

In this paper, we formulate microbial life-boundary prediction as *proposal-conditioned phenotype refinement* to leverage the integrative capabilities of LLMs. We theoretically characterize proposal-

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

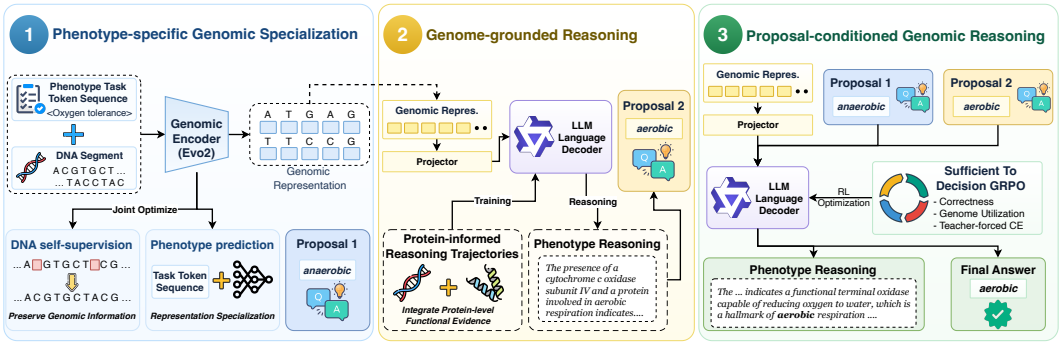

Figure 1: Overview of GGBound. The framework comprises phenotype-specific genomic specialization, genome-grounded reasoning with protein-informed supervision, and proposal-conditioned refinement using S2D-GRPO to produce the final life-boundary phenotype prediction.

conditioned genomic utility, showing that effective representations should retain phenotype-relevant information while suppressing irrelevant variation and support refinement beyond auxiliary proposals. Guided by this analysis, we introduce GGBound, a unified framework for genome-grounded microbial life-boundary phenotype prediction. As shown in Figure 1, GGBound first learns phenotype-specific genomic representations and grounds them in a language decoder through protein-informed reasoning supervision, enabling natural-language interpretation of genomic evidence. Following our theoretical analysis, we further introduce *Sufficiency-to-Decision GRPO* (S2D-GRPO) to strengthen the predictive use of genomic evidence beyond auxiliary proposals during phenotype refinement.

To support systematic evaluation, we further construct a strain-centric benchmark spanning 17 microbial life-boundary phenotypes from IJSEM strains and BacDive annotations, together with paired training data for model development. Extensive experiments show consistent gains over genome-based baselines and stronger performance than frontier LLMs given strain names on most phenotype groups. Ablations further validate the contributions of phenotype-specific specialization, genome-to-language grounding, and S2D-GRPO. Our contributions are summarized as follows:

- We formulate microbial phenotype prediction as proposal-conditioned refinement and theoretically characterize the predictive utility of genomic representations, showing that effective representations should retain phenotype-relevant information while suppressing irrelevant variation. (Sec. 4.1)
- We propose *GGBound*, a unified framework that combines phenotype-specific specialization with genome-grounded proposal refinement. We further introduce *S2D-GRPO* to enhance genomic evidence utilization during prediction. (Sec. 4.2)
- We construct a strain-centric benchmark spanning 17 microbial life-boundary phenotypes under sparse genomic observations. Extensive experiments demonstrate the consistent improvements and validate the proposed components of GGBound. (Sec. 5)

2 Related Work

Microbial genome-to-phenotype modeling. Predicting microbial phenotypes from genomic information has long been a central problem in computational biology. Existing approaches typically formulate this task as supervised inference from genomic features to predefined traits. Early methods transformed genomes into structured representations such as gene repertoires, protein-family profiles, and phyletic patterns, and used these features for supervised phenotype prediction (Feldbauer et al., 2015; Brbić et al., 2016; Weimann et al., 2016). Sequence-centric approaches instead derived predictive features directly from genome-wide k-mer patterns (Aun et al., 2018; Drouin et al., 2019). Subsequent work extended this paradigm to quantitative physiological properties such as optimal growth temperature (Sauer & Wang, 2019). Recent studies further expanded genome-based prediction to broader strain-level traits using curated BacDive annotations (Karlsen et al., 2023; Koblitz et al., 2025). Collectively, these studies establish a strong predictive link between microbial genomes and phenotype. However, existing inference is still predominantly formulated through task-specific classification or regression models (Karlsen et al., 2023), offering limited support for integrating heterogeneous evidence within a unified reasoning process. We instead investigate whether genomic information can support language-based reasoning across diverse life-boundary phenotypes.

Biological Foundation Models and Genome-Grounded Reasoning. Biological foundation models have enabled transferable representation learning across both proteins and genomic sequences. Protein language models such as ESM (Rives et al., 2021) and ProtTrans (Elnaggar et al., 2022) operate on compact and functionally constrained amino-acid sequences, showing that large-scale self-supervision can recover rich structural and functional signals. Extending this paradigm to genomes is more challenging because genomic sequences are substantially longer and exhibit complex long-range dependencies across diverse functional regions (Nguyen et al., 2023). DNABERT (Ji et al., 2021), GROVER (Sanabria et al., 2024), and Nucleotide Transformer (Dalla-Torre et al., 2025) explore complementary directions in DNA representation learning, including transferable pretraining and long-context modeling, while Evo and Evo 2 further extend sequence modeling toward genome scale (Nguyen et al., 2024; Brixi et al., 2026). Building on these representations, ChatNT connects biological sequence encoders with language models (de Almeida et al., 2025), while BioReason introduces supervised and reinforcement learning for multi-step biological reasoning (Fallahpour et al., 2025). Despite this progress, genome-grounded reasoning for microbial life-boundary phenotypes remains underexplored. GGBound addresses this setting through phenotype-specific genomic specialization and proposal-conditioned reasoning.

3 PRELIMINARIES

Phenotype prediction from sparse genomic observations. Let $X \in \mathcal{X}$ denote a contiguous DNA segment sampled from a strain genome, and let $Y \in \mathcal{Y}$ denote the corresponding strain-level life-boundary phenotype, such as optimal growth temperature. We consider an encoder–decoder model consisting of a genomic representation encoder $f_{\theta_{\text{enc}}}$ and a LLM-based language decoder $\pi_{\theta_{\text{dec}}}$. Given the DNA segment X, the encoder extracts the phenotype-specific genomic representation $Z = f_{\theta_{\text{enc}}}(X)$, while the decoder interprets Z through natural-language phenotype reasoning. To complement the sparse genomic evidence, the language decoder additionally receives auxiliary phenotype proposals A, which provide informative but imperfect candidate predictions. It then reasons over Z together with A to produce the final phenotype prediction $\hat{Y}$.

Phenotype-specific genomic representation learning. Genomic foundation models capture broad sequence-level information across the genome, encoding diverse biological patterns (Brixi et al., 2026; James et al., 2025), whereas a specific phenotype is typically determined by only a subset of these genomic factors. Consequently, directly transferring such broad representations may obscure phenotype-relevant signals (Koblitz et al., 2025), while phenotype-specific adaptation may overwrite potentially useful sequence information (Veiner & Supek, 2026). We therefore seek a representation Z that concentrates evidence relevant to the target phenotype while preserving useful genomic information and suppressing phenotype-irrelevant variation.

Proposal-conditioned phenotype refinement. The auxiliary proposals provide informative but imperfect phenotype hypotheses, while the genomic representation supplies complementary evidence unavailable from the proposals alone. Thus, Z should retain genomic information predictive of Y beyond A while suppressing phenotype-irrelevant variation, leading to

$$\max_{\theta_{enc},\theta_{dec}} I(Z;Y \mid A) - \alpha I(X;Z \mid Y,A), \tag{1}$$

where $\alpha > 0$ controls the prediction–compression trade-off. The first term captures predictive genomic evidence beyond the proposals, whereas the second penalizes residual genomic information unnecessary for phenotype prediction.

4 METHODOLOGY

4.1 FROM GENOMIC RETENTION TO PREDICTIVE UTILITY

The formulation in Eq. 1 specifies the desired phenotype-specific bottleneck, which we study from two complementary aspects: preserving predictive genomic information during specialization and ensuring that the language decoder exploits the retained evidence beyond the auxiliary proposals.

Phenotype-specific specialization requires selective genomic retention. Genomic foundation models encode broad sequence-level information, whereas phenotype-specific learning seeks to

concentrate the genomic factors most relevant to the target phenotype. Such specialization creates an inherent trade-off, as it should suppress irrelevant genomic variation without discarding information that remains predictive of Y. Since $Z = f_{\theta_{\text{enc}}}(X)$, we have $I(Z;Y \mid A) = I(X;Z \mid A) - I(X;Z \mid Y,A)$, showing that phenotype-relevant information depends jointly on genomic retention and the suppression of phenotype-irrelevant variation. Phenotype-specific specialization should therefore selectively balance these two factors.

Genomic retention should contribute to predictive use. Preserving phenotype-relevant information in Z does not by itself ensure that the decoder uses it when producing the final prediction. At the language-decoding stage, the genomic contribution admits a similar decomposition, $I(Z;\hat{Y} \mid A) = I(Y;\hat{Y} \mid A) + I(Z;\hat{Y} \mid Y,A)$, where the second term captures genomic dependence that remains unexplained by the true phenotype. Since $I(Z;\hat{Y} \mid Y,A) \le I(X;Z \mid Y,A)$, suppressing phenotype-irrelevant information in Z also limits such dependence on the final decision. We refer to the additional predictive contribution of the genomic representation as proposal-conditioned genomic utility. The following theorem connects this utility to predictive information in the final decision.

Theorem 1 (Proposal-Conditioned Genomic Utility)**.** *For a fixed proposal condition A, let $X_1,\dots,X_K$ be independent proposal-matched genomic segment alternatives, with $Z_k = f_{\theta_{\text{enc}}}(X_k)$. Define genomic utility as $\mathcal{U}_K = \mathbb{E}\Big[\log \pi_{\theta_{\text{dec}}}(\hat{Y} \mid Z,A) - \log\Big(\frac{1}{K}\sum_{k=1}^{K} \pi_{\theta_{\text{dec}}}(\hat{Y} \mid Z_k,A)\Big)\Big]$. Under the Assumption 1, if the predictive likelihoods are bounded below by $\delta > 0$, then*

$$\mathcal{U}_K - I(X;Z \mid Y,A) - \frac{(1-\delta)^2}{8\delta^2 K} \le I(Y;\hat{Y} \mid A) \le \mathcal{U}_K.$$

Remark 1. *The proof is provided in Appendix A. Theorem 1 shows that suppressing phenotype-irrelevant information and increasing proposal-conditioned references tighten the correspondence between $\mathcal{U}_K$ and predictive information, motivating stronger use of genomic evidence in refinement.*

From predictive utility to optimization. The representation analysis and Theorem 1 yield two coupled requirements for optimization. At the representation level, genomic specialization should balance information retention with the suppression of phenotype-irrelevant variation. At the decoder level, the retained evidence should further contribute predictive information beyond the auxiliary proposals. We next develop a unified optimization framework that addresses both requirements.

4.2 GGBound: Genome-Grounded Reasoning for Life-Boundary Phenotypes

To realize the representation- and decoder-level requirements in Section 4.1, we develop GGBound, a unified framework for genome-grounded life-boundary reasoning. GGBound first specializes pretrained genomic representations toward the target phenotype while preserving useful genomic information, and then grounds them in the language decoder for proposal-conditioned prediction.

Task tokens specialize genomic representations. The representation objective in Eq. 1 requires Z to emphasize genomic information predictive of the target phenotype rather than preserve all sequence variation indiscriminately. We realize this requirement by prepending a phenotype-specific task token sequence to each genomic input and jointly optimizing the genomic encoder with a lightweight prediction MLP. The task tokens condition the encoder on the target phenotype, promoting task-relevant genomic representations. Phenotype supervision from the MLP further reshapes the pretrained general-purpose representation toward genomic evidence predictive of Y. This joint optimization produces phenotype-specific genomic representations with a shared genomic encoder.

DNA self-supervision preserves genomic information. Phenotype-driven specialization may discard sequence information useful for downstream reasoning. We therefore regularize the encoder with a local DNA reconstruction objective that preserves genomic information. Specifically, we randomly perturb a small fraction of nucleotides and train the encoder to recover the original nucleotides at these positions. Combining phenotype supervision with DNA self-supervision, we jointly optimize $\mathcal{L}_{\text{rep}} = \mathcal{L}_{\text{task}} + \lambda_{\text{SSL}}\mathcal{L}_{\text{SSL}}$, where $\mathcal{L}_{\text{task}}$ is the phenotype prediction loss and $\mathcal{L}_{\text{SSL}}$ is the reconstruction loss. Together, the two objectives promote phenotype-specific specialization while preserving genomic information during adaptation. The jointly trained prediction MLP further provides an auxiliary phenotype proposal for subsequent reasoning.

Protein-informed reasoning trajectories enable phenotype-oriented reasoning. After phenotype-specific specialization, Z concentrates phenotype-relevant genomic evidence. However, this represen-

tation alone does not explicitly organize such evidence into phenotype-oriented functional reasoning. We therefore project Z into the LLM decoder embedding space and use it as genomic context. Although Z is derived from a local genomic segment, the prediction target is defined at the strain level. Accordingly, we introduce phenotype-relevant protein information from the corresponding strain as training supervision. Specifically, we filter and integrate protein descriptions into reasoning trajectories that connect strain-level functional evidence to phenotypic outcomes. These trajectories supervise the decoder to associate segment-derived Z with phenotype-relevant functional semantics, thereby enabling it to derive an independent phenotype proposal.

Proposal-conditioned reasoning refines predictions. To further improve predictive accuracy, we treat the two phenotype proposals as auxiliary predictions A and retain the genomic representation Z as direct evidence for refinement. The key challenge is to ensure that the decoder uses Z to revise inaccurate proposals rather than simply reproduce the information in A. We address this with Sufficiency-to-Decision GRPO (S2D-GRPO), which optimizes phenotype prediction correctness while promoting the predictive use of genomic evidence beyond the auxiliary proposals.

Token-level genomic utility. Theorem 1 motivates strengthening the predictive contribution of genomic evidence beyond the auxiliary proposals. We therefore encourage such contribution throughout the reasoning process. As the decoder forms its prediction autoregressively, we realize this principle through a token-level surrogate using the same proposal-matched contrast. Specifically, given a rollout $O = (o_1, \ldots, o_{|O|}) \sim \pi_\theta(\cdot \mid Z, A)$ and K proposal-matched genomic alternatives $\{Z_k\}_{k=1}^K$, we define the local genomic utility at step t as $u_t = \log \pi_\theta(o_t \mid Z, A, o_{<t}) - \log\Big[\frac{1}{K}\sum_{k=1}^K \pi_\theta(o_t \mid Z_k, A, o_{<t})\Big]$. A larger u_t indicates greater dependence of the token likelihood on the observed genomic representation.

Policy optimization. In addition to the token-level genomic utility, we introduce phenotype-specific correctness rewards r_i to score each rollout. Following GRPO (Shao et al., 2024), we normalize the rewards within each rollout group as $\hat{r}_i = \frac{r_i - \mathrm{mean}(r)}{\mathrm{std}(r)}$, yielding a stabilized sequence-level advantage for policy optimization. We then incorporate genomic utility by weighting the advantage as $\hat{r}_i(1 + w_1 u_{i,t})$. To further stabilize optimization, we include a teacher-forced cross-entropy loss $\mathcal{L}_{\mathrm{CE}}$ that provides direct supervision toward the correct phenotype prediction. Concretely, the language decoder π_θ is iteratively updated by sampling a group of outputs $\{O_1, \ldots, O_G\}$ from the previous policy $\pi_{\theta_{\mathrm{old}}}$ and maximizing the following objective:

$$\mathcal{J}(\theta) \approx \mathbb{E}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|O_i|}\sum_{t=1}^{|O_i|}\frac{\pi_\theta(o_{i,t} \mid Z, A, o_{i,<t})}{\pi_{\theta_{\mathrm{old}}}(o_{i,t} \mid Z, A, o_{i,<t})}\left(\hat{r}_i(1 + w_1 u_{i,t})\right)\right] - \beta D_{\mathrm{KL}}\big[\pi_\theta \,\|\, \pi_{\mathrm{ref}}\big] - w_2\mathcal{L}_{\mathrm{CE}}, \quad (2)$$

where π_{ref} is the reference model, w_1 and w_2 weight genomic utility and cross-entropy supervision, and the KL term regularizes the policy toward the reference model at the token level.

5 Experiments

5.1 Dataset and Benchmark for Genome-Grounded Life-Boundary Prediction

To support the training and evaluation of GGBound, we construct a unified dataset covering **17** microbial phenotypes across five categories. The data are split into training, validation, and benchmark sets, with expert-curated labels to ensure biological consistency and relevance.

Phenotype scope and task coverage. We consider five categories of strain-level microbial phenotypes from sparse genomic segments, encompassing 17 phenotypes in total. These comprise *physiological boundaries* defined by growth temperature, pH, and salinity ranges, together with their corresponding *physiological optimum*; *metabolism*, including carbon source, electron donor, electron acceptor, and nitrogen source utilization; *categorical physiology*, including oxygen tolerance and energy type; and *morphology*, including cell shape, motility, flagella, spore formation, and Gram stain. Collectively, these phenotypes characterize complementary dimensions of microbial life boundaries.

Data collection and expert curation. We begin by collecting microbial strains from taxonomic articles published in the *International Journal of Systematic and Evolutionary Microbiology*, providing a broad pool of experimentally characterized organisms. The extracted strain identities are normalized

and matched to NCBI (Sayers et al., 2021) to retrieve the corresponding genome assemblies. We then cross-reference each matched strain with BacDive (Schober et al., 2025) to obtain the initial structured strain-level phenotype annotations used for both training and evaluation. However, as the retrieved annotations are often fragmented, heterogeneous, and inconsistently formatted, human experts further clean and harmonize them into biologically meaningful task-specific label spaces. Detailed collection procedures and expert-defined label mappings are provided in Appendix B.1.

Dataset partition and benchmark composition. We partition the curated data at the strain level to ensure that training, validation, and test strains are mutually exclusive. As prokaryotic genes span approximately 1,000 bp on average (Xu et al., 2006), we use 2,048-bp genomic segments to preserve gene-scale local context while maintaining a standardized input length. For training, we sample multiple segments from each strain–phenotype record with phenotype-aware balancing, increasing the number of optimization instances without introducing additional strains. In contrast, the benchmark is neither expanded nor rebalanced; genomic segments are sampled once from held-out records and kept fixed throughout evaluation. Given the limited and uneven availability of experimentally annotated strains across phenotypes, the number of held-out examples varies across tasks, with several metabolic phenotypes containing relatively few test samples. We retain these tasks to preserve the breadth of microbial life-boundary evaluation. Table 1 summarizes the dataset composition, with detailed sampling and partitioning procedures provided in Appendix B.2.

Table 1: Dataset composition.

Category	Ground.	S2D	Bench.
Boundary	108,000	1,536	505
Optimum	81,200	1,536	419
Metabolism	27,726	769	148
Categorical	52,800	661	219
Morphology	124,800	2,355	789
Total	**394,526**	**6,857**	**2,080**

Protein-informed reasoning trajectory construction. To incorporate protein-level functional evidence into phenotype reasoning, we further construct protein-informed reasoning trajectories for the training data. For each strain, we retrieve its NCBI protein sequences and obtain eggNOG-based functional annotations (Cantalapiedra et al., 2021). Given the target phenotype, we identify mechanistically relevant proteins from the resulting annotations and use them to construct concise rationales linking protein functions to the observed phenotype. The rationales are further refined and quality-controlled to remove unsupported or inconsistent claims, and are then aligned with the corresponding genomic-segment samples for decoder training. Details are provided in Appendix B.3.

5.2 Experimental Setup

We summarize the experimental setup below; full experimental details are provided in Appendix C.

Architecture. We use Evo2-7B (Brixi et al., 2026) as the genomic encoder and Qwen3.5-4B (Qwen Team, 2026) as the LLM decoder. Each 2,048-bp genomic segment is encoded into phenotype-specific representations. To preserve coarse sequence structure, we partition the encoder outputs into 32 consecutive bins and average within each bin, yielding 32 ordered genomic representations. These representations are then projected into the decoder embedding space as soft gene evidence.

Implementation Details. We train GGBound sequentially, with each stage initialized from the preceding checkpoint and optimized for one epoch. During representation specialization, we prepend 100 phenotype-specific task tokens to each genomic input. Evo2 remains trainable through the subsequent genome-to-language grounding stage and is frozen for S2D-GRPO optimization. In the final S2D-GRPO training, we sample $G = 8$ rollouts per input and use $K = 4$ proposal-matched genomic alternatives to estimate token-level genomic utility.

Baselines. We compare GGBound with three controlled baselines: *Evo2-MLP* for direct phenotype prediction from specialized genomic representations, *G-Qwen* for genome-grounded language prediction, and *P-Qwen* for proposal-conditioned prediction without S2D-GRPO. We additionally include *Evo2-Prob*, a frozen-Evo2 baseline with a trainable MLP head (Dalla-Torre et al., 2025). Following recent evaluations of general-purpose LLMs on raw genomic sequences (Long et al., 2026), we further evaluate GPT-5.6 Sol, Claude Opus 5, GLM-5.3 (GLM-5-Team et al., 2026), Kimi K3 (Team et al., 2026), and DeepSeek-V4 Pro (DeepSeek-AI et al., 2026) using the genomic segments as input.

Metrics. We evaluate all methods on the same strain-disjoint benchmark with fixed genomic crops. We report RMSE and normalized RMSE (NRMSE) for physiological boundaries and optimum, F1 score for metabolic preferences, and accuracy for categorical physiology and morphology.

Table 2: Main results across 17 microbial life-boundary phenotypes. Best results are shown in **bold** and second-best results are underlined. D.S., E2-P., E2-M., G-Qw., P-Qw., and GGB. denote DeepSeek, Evo2-Prob, Evo2-MLP, G-Qwen, P-Qwen, and GGBound, respectively.

Group	Phenotype	GPT	Opus	D.S.	GLM	Kimi	E2-P.	E2-M.	G-Qw.	P-Qw.	GGB.
Boundary RMSE ↓	Temperature	12.7	12.8	12.9	12.4	12.9	10.2	7.92	9.45	9.59	**7.72**
	pH	1.33	1.32	1.37	1.30	1.37	1.29	1.27	1.42	1.35	**1.23**
	Salinity	4.92	4.50	4.48	4.52	4.48	4.43	3.34	4.09	3.87	**3.29**
Optimum RMSE ↓	Temperature	8.17	8.44	11.3	8.63	9.54	8.20	5.45	6.59	6.37	**5.00**
	pH	1.11	1.11	1.15	1.15	1.15	1.09	0.87	1.16	1.13	**0.83**
	Salinity	3.52	3.70	3.91	3.91	3.91	3.32	2.66	3.25	3.03	**2.57**
Metabolism F1 ↑	Carbon source	0.46	0.48	0.43	0.43	0.43	0.62	0.62	0.58	0.46	**0.63**
	Electron acceptor	0.33	0.11	0.33	0.11	0.11	0.11	0.52	0.65	0.63	**0.87**
	Electron donor	0.43	0.40	0.40	0.35	0.38	0.21	0.54	0.82	0.82	**0.89**
	Nitrogen source	0.52	0.53	0.57	0.57	0.47	0.48	0.60	0.49	0.54	**0.83**
Categorical Acc.(%) ↑	Oxygen tolerance	31.6	25.9	30.1	25.9	29.5	35.8	66.8	59.6	60.1	**67.8**
	Energy type	69.2	76.9	76.9	76.9	76.9	34.6	76.9	80.8	80.8	**84.6**
Morphology Acc.(%) ↑	Cell shape	27.2	24.3	24.3	24.3	24.3	14.6	45.6	42.7	47.6	**54.4**
	Motility	55.5	55.0	50.0	50.0	57.5	58.5	76.0	68.0	50.5	**77.5**
	Flagella	37.9	44.8	21.8	21.8	21.8	51.8	81.6	81.6	85.1	**86.2**
	Spore formation	65.8	67.8	50.3	59.8	53.8	64.8	91.5	90.5	49.8	**92.5**
	Gram stain	57.5	59.5	58.0	57.5	57.5	52.0	96.0	95.5	69.0	**96.5**

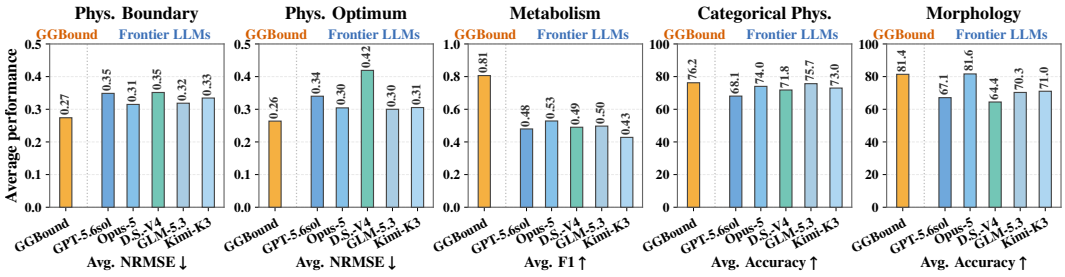

Figure 2: Group-level comparison with strain-name-conditioned frontier LLMs. Results are averaged across phenotypes within each group.

5.3 Main Results

General-purpose LLMs struggle with sparse genomic inputs. As shown in Table 2, frontier LLMs consistently underperform genome-specific predictors across phenotype groups. The gap is particularly clear for physiological and morphological traits. For example, temperature-boundary RMSE decreases from 12.4 with GLM-5.3 to 7.92 with Evo2-MLP, while flagella accuracy increases from 44.8% to 81.6%. These results indicate that directly applying general-purpose LLMs to sparse genomic segments is insufficient for phenotype prediction, motivating genome-specific modeling.

GGBound consistently achieves the strongest performance across phenotype groups. Table 2 reveals complementary strengths of the genome-specific baselines. Specifically, Evo2-MLP performs strongly on physiological and morphological phenotypes, while G-Qwen is more competitive on several metabolic and categorical traits. P-Qwen shows mixed results compared with G-Qwen, indicating that simply adding auxiliary proposals does not consistently improve phenotype prediction. By integrating the proposals from Evo2-MLP and G-Qwen, GGBound achieves the best performance on all 17 phenotypes. The improvements are substantial, with electron-acceptor F1 increasing from 0.65 to 0.87 and temperature-optimum RMSE decreasing from 5.45 to 5.00. These results demonstrate the benefit of refining complementary phenotype proposals using genomic evidence.

Sparse genomic evidence rivals frontier LLMs with strain-name access. We further query frontier LLMs using only the strain name, which can provide taxonomic and phenotype-related lexical cues while also eliciting parametric knowledge about known microorganisms. In contrast, GGBound receives no strain identity and predicts phenotypes from a single 2,048-bp genomic segment. As

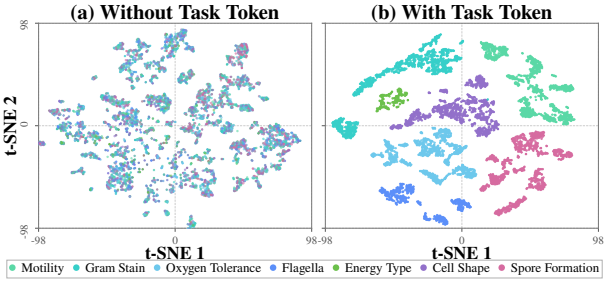

Figure 3: Effect of task-token conditioning.

Table 3: Representation ablation.

	Metric	Adapt	+Token	+SSL	Full
Task token		×	✓	×	✓
DNA SSL		×	×	✓	✓
Bound.	NRM.↓	0.28	0.29	0.28	0.28
Optim.	NRM.↓	0.28	0.29	0.26	0.28
Metab.	F1 ↑	0.51	0.54	0.55	0.57
Categ.	Acc. ↑	62.7	61.0	16.8	71.9
Morph.	Acc. ↑	73.9	74.5	25.9	78.1

shown in Figure 2, GGBound achieves the strongest performance on four of the five phenotype groups. Compared with the best frontier LLM, GGBound reduces average normalized RMSE (NRMSE) from 0.31 to 0.27 on physiological boundaries and increases metabolic F1 score from 0.53 to 0.81, while remaining comparable on morphology (81.4% vs. 81.6%). These results show that phenotype-relevant signals contained in a short genomic segment can rival, and often exceed, the knowledge accessible to frontier LLMs from strain identity. Further analyses are provided in Appendix D.1 and D.2.

5.4 Empirical Analysis of GGBound's Mechanism

Task-token conditioning induces phenotype-specific specialization. We first study task-token conditioning at the representation level using Evo2-MLP, before genome-to-language grounding. To examine its effect on genomic representations, we collect genome representations generated with and without task tokens and visualize them using t-SNE. As shown in Figure 3, representations without task tokens are largely intermixed across phenotypes, whereas task conditioning produces substantially clearer phenotype-dependent separation. We further examine whether this representation shift directly improves prediction. As shown in Table 3, adding task tokens alone does not uniformly improve prediction performance over purely phenotype-supervised Evo2 adaptation. The NRMSE on Boundary and Optimum increases from 0.28 to 0.29, while Categorical accuracy decreases from 62.7% to 61.0%. These results suggest that task-token conditioning induces clearer phenotype-specific organization in the genomic representation space, but this specialization may come at the expense of genomic information that remains useful for prediction.

Table 4: Task-token length sensitivity.

Group	Metric	50	100	200
Boundary	NRMSE↓	0.28	0.28	0.28
Optimum	NRMSE↓	0.27	0.28	0.28
Metabolic	F1 score↑	0.57	0.57	0.52
Categorical	Acc.(%) ↑	69.1	71.9	69.4
Morphology	Acc.(%) ↑	77.6	78.1	77.8

DNA self-supervision complements phenotype-specific specialization. We thus incorporate DNA self-supervision to preserve useful genomic information during genomic representation learning. As shown in Table 3, DNA SSL alone maintains or improves Evo2-MLP performance on physiological and metabolic phenotypes, but substantially degrades classification performance, with Categorical accuracy dropping from 62.7% to 16.8%. In contrast, combining DNA SSL with task-token conditioning improves all five phenotype groups over task conditioning alone, increasing Categorical accuracy from 61.0% to 71.9%. We further evaluate the effect of task-token length under DNA self-supervision. As shown in Table 4, performance remains broadly stable from 50 to 200 tokens, with 100 tokens achieving the strongest overall results on the categorical and morphological groups. These results indicate that DNA self-supervision complements task conditioning while remaining relatively insensitive to the precise task-token length. Additional detailed results are provided in Appendix D.3.

Genomic specialization improves downstream grounding. We next examine whether phenotype-oriented genomic adaptation benefits subsequent genome-to-language grounding. As shown in Figure 4, using purely phenotype-supervised Evo2 adaptation (*Adapt*) as the reference, freezing the Evo2 encoder degrades performance on four of the five phenotype groups, with relative drops of 15% on Metabolism and 16.1% on Morphology. Task-token conditioning improves all five groups, with particularly large gains on Metabolism, Categorical physiology, and Morphology. DNA self-supervision provides similarly consistent improvements, including a 10.2% gain on Physiological optimum. Combining both components leverages their complementary strengths across all groups, showing that phenotype adaptation supports downstream grounding through both task conditioning and DNA self-supervision. Details are provided in Appendix D.4 and D.5.

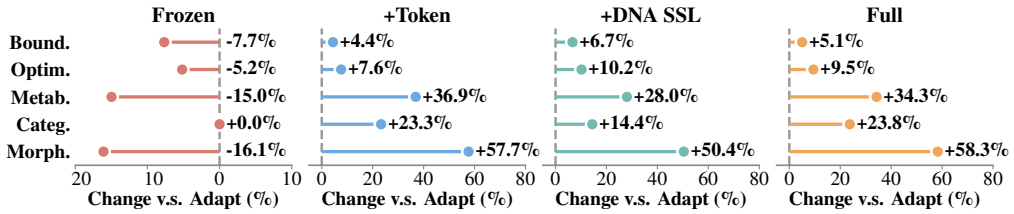

Figure 4: Effect of genomic specialization on downstream grounding.

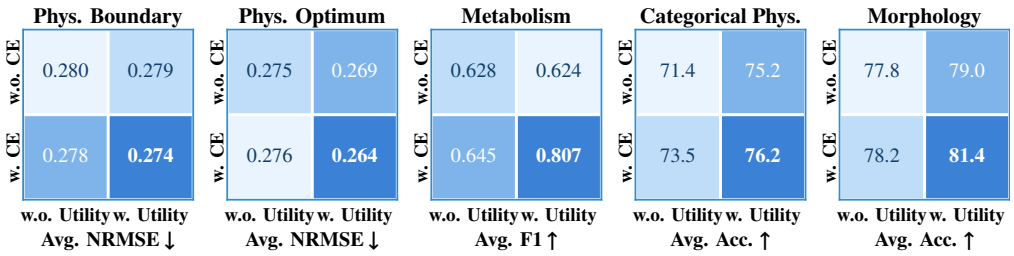

Figure 5: Component ablation of S2D-GRPO. Rows correspond to CE anchoring and columns to genomic utility. Darker shades indicate stronger relative performance within each phenotype group.

Genomic utility and CE anchoring strengthen proposal-conditioned refinement. We further examine the contributions of genomic utility and CE anchoring in S2D-GRPO by removing each component independently. As shown in Figure 5, the full objective performs best across all five phenotype groups, while each component alone improves four of the five groups over standard GRPO. We observe the largest improvement on metabolism, where average F1 reaches 0.81, compared with 0.65 without genomic utility, 0.62 without CE anchoring, and 0.63 with standard GRPO. These results suggest the complementary roles of two components. Genomic utility emphasizes genome-sensitive token credit, whereas CE anchoring provides direct supervision when rollout rewards are uninformative. Additional results are provided in Appendix D.6.

S2D-GRPO improves proposal conflict resolution. We analyze predictions according to the correctness of the two auxiliary proposals. As shown in Table 5, when both proposals are correct, all RL variants largely preserve the correct prediction, reaching nearly 100% accuracy. The difference is substantially larger when exactly one proposal is correct: standard GRPO achieves 63.4% accuracy, while genomic utility and CE anchoring individually improve it to 72.3% and 68.3%, respectively. Combining both signals further raises accuracy to 84.2%. These results show that S2D-GRPO better resolves conflicting auxiliary predictions while preserving cases in which the proposals already agree with the target. We provide further results in Appendix D.7.

Table 5: Proposal resolution.

Method	Both	Single
P-Qwen	73.2	44.6
GRPO	99.9	63.4
+ Utility	99.9	72.3
+ CE	98.9	68.3
GGBound	**100.0**	**84.2**

6 Conclusion and Broader Impact

Conclusion. In this paper, we formulate microbial life-boundary prediction as proposal-conditioned phenotype refinement and introduce GGBound, which integrates phenotype-specific genomic specialization, protein-derived genome-to-language grounding, and S2D-GRPO for genomic evidence utilization. We further construct a strain-centric benchmark spanning 17 life-boundary phenotypes. Across diverse phenotype groups, GGBound consistently outperforms baselines and shows that informative phenotype predictions can be derived from sparse genomic observations. Additional analyses and limitations are discussed in the Appendix D.8 and E.

Broader Impact. We observe that phenotype-specific adaptation of genomic foundation models is important for genome-grounded reasoning under sparse sequence observations. Meanwhile, phenotype signals from short genomic segments may help prioritize strains and experimental conditions for downstream wet-lab validation. We hope these findings can inform future development of genome-grounded models and experimental studies of microbial phenotypes.

AI USE STATEMENT

We used generative AI tools to assist with GGBound code development and debugging, result analysis, literature review, and manuscript language polishing. We also used language models (Qwen3.5-9B and Qwen3.5-27B) to construct the protein-informed reasoning trajectories used for genome-to-language grounding, as described in Appendix B.3. These trajectories were derived from indexed protein annotations and subjected to evidence-grounding and consistency checks, while all phenotype labels used for training and evaluation were obtained from curated database annotations. All AI-assisted code, analyses, and manuscript text were reviewed and verified by the authors. We take responsibility for the final content of this work, including all claims, results, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work uses publicly available microbial genomic and phenotype data and does not involve human subjects or personally identifiable information. The proposed benchmark focuses on microbial life-boundary phenotypes and does not include pathogenicity, virulence, or experimental design objectives. Nevertheless, genome-based phenotype prediction may have broader biological applications, and predictions from sparse genomic segments should be treated as computational hypotheses awaiting experimental validation. We report the provenance and processing of the genomic and phenotype data to support responsible use and interpretation of the benchmark.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of data collection, phenotype harmonization, genomic sampling, and benchmark construction in Appendix B.1 and B.2, and describe the construction and quality control of protein-informed reasoning trajectories in Appendix B.3. Appendix C specifies the model architectures, training objectives, optimization settings, inference procedures, and hyperparameters for all training stages. Appendix A provides the assumptions and complete proofs for the theoretical results. All methods are evaluated on the same fixed genomic crops, with the benchmark kept strain-disjoint from all model-development data. Additional ablations and evaluation results are reported in Appendix D.1. The benchmark data are included in the supplementary material.

REFERENCES

Erki Aun, Age Brauer, Veljo Kisand, Tanel Tenson, and Maido Remm. A k-mer-based method for the identification of phenotype-associated genomic biomarkers and predicting phenotypes of sequenced bacteria. *PLoS computational biology*, 14(10):e1006434, 2018.

Maria Brbić, Matija Piškorec, Vedrana Vidulin, Anita Kriško, Tomislav Šmuc, and Fran Supek. The landscape of microbial phenotypic traits and associated genes. *Nucleic Acids Research*, 44(21):10074–10090, 12 2016. ISSN 0305-1048. doi: 10.1093/nar/gkw964. URL `https://doi.org/10.1093/nar/gkw964`.

Garyk Brixi, Matthew G Durrant, Jerome Ku, Mohsen Naghipourfar, Michael Poli, Gwanggyu Sun, Greg Brockman, Daniel Chang, Alison Fanton, Gabriel A Gonzalez, et al. Genome modelling and design across all domains of life with evo 2. *Nature*, 652(8112):1349–1361, 2026.

Carlos P Cantalapiedra, Ana Hernández-Plaza, Ivica Letunic, Peer Bork, and Jaime Huerta-Cepas. eggnog-mapper v2: functional annotation, orthology assignments, and domain prediction at the metagenomic scale. *Molecular biology and evolution*, 38(12):5825–5829, 2021.

Hugo Dalla-Torre, Liam Gonzalez, Javier Mendoza-Revilla, Nicolas Lopez Carranza, Adam Henryk Grzywaczewski, Francesco Oteri, Christian Dallago, Evan Trop, Bernardo P De Almeida, Hassan Sirelkhatim, et al. Nucleotide transformer: building and evaluating robust foundation models for human genomics. *Nature methods*, 22(2):287–297, 2025.

Bernardo P de Almeida, Guillaume Richard, Hugo Dalla-Torre, Christopher Blum, Lorenz Hexemer, Priyanka Pandey, Stefan Laurent, Chandana Rajesh, Marie Lopez, Alexandre Laterre, et al. A

multimodal conversational agent for dna, rna and protein tasks. *Nature Machine Intelligence*, 7(6): 928–941, 2025.

DeepSeek-AI, Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chengyu Hou, Chenhao Xu, Chenze Shao, Chong Ruan, Conner Sun, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Donghao Li, Dongjie Ji, Erhang Li, Fang Wei, Fangyun Lin, Fangzhou Yuan, Feiyu Xia, Fucong Dai, Guangbo Hao, Guanting Chen, Guoai Cao, Guolai Meng, Guowei Li, Han Yu, Han Zhang, Hanwei Xu, Hao Li, Haofen Liang, Haoling Zhang, Haoming Luo, Haoran Wei, Haotian Yuan, Haowei Zhang, Haowen Luo, Haoyu Chen, Haozhe Ji, Hengqing Zhang, Honghui Ding, Hongxuan Tang, Huanqi Cao, Huazuo Gao, Hui Qu, Hui Zeng, J Yang, JQ Zhu, Jia Luo, Jia Song, Jia Yu, Jialiang Huang, Jialu Cai, Jian Liang, Jiangting Zhou, Jiasheng Ye, Jiashi Li, Jiaxin Xu, Jiewen Hu, Jieyu Yang, Jin Chen, Jin Yan, Jingchang Chen, Jingli Zhou, Jingting Xiang, Jingyang Yuan, Jingyuan Cheng, Jingzi Zhou, Jinhua Zhu, Jiping Yu, Joseph Sun, Jun Ran, Junguang Jiang, Junjie Qiu, Junlong Li, Junmin Zheng, Junxiao Song, Kai Dong, Kaige Gao, Kang Guan, Kexing Zhou, Kezhao Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Wang, Leyi Xia, Li Zhang, Liang Zhao, Lihua Guo, Lingxiao Luo, Linwang Ma, Linyan Zhu, Litong Wang, Liyu Cai, Liyue Zhang, Longhao Chen, MS Di, MY Xu, Max Mei, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Mingxu Zhou, Minmin Han, Ning Wang, Panpan Huang, Panpan Wang, Peixin Cong, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qingyang Li, Qinyu Chen, Qiushi Du, Qiwei Jiang, Rui Tian, Ruifan Xu, Ruijie Lu, Ruiling Xu, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runqian Chen, Runqiu Yin, Runxin Xu, Ruomeng Shen, Ruoyu Zhang, Ruyi Chen, SH Liu, Shanghao Lu, Shangmian Sun, Shangyan Zhou, Shanhuang Chen, Shaofei Cai, Shaoheng Nie, Shaoqing Wu, Shaoyuan Chen, Shengding Hu, Shengyu Liu, Shiqiang Hu, Shirong Ma, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, Shuying Yu, Songyang Zhou, Tao Ni, Tao Yun, Tian Jin, Tian Pei, Tian Ye, Tianle Lin, Tianran Ji, Tianyi Cui, Tianyuan Yue, Tingting Yu, Tun Wang, W Zhang, WL Xiao, Wangding Zeng, Wei An, Weilin Zhao, Wen Liu, Wenfeng Liang, Wenjie Pang, Wenjing Luo, Wenjing Yao, Wenjun Gao, Wenkai Yang, Wenlve Huang, Wenqing Hou, Wentao Zhang, Wenting Ma, Xi Gao, Xiang He, Xiangwen Wang, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaokang Zhang, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingchen Liu, Xingkai Yu, Xingyou Li, Xinyu Yang, Xinyu Zhang, Xu Chen, Xuanyu Wang, Xuecheng Su, Xueyin Chen, Xuheng Lin, Xuwei Fu, YC Yan, YQ Wang, YW Ma, Yanfeng Luo, Yang Zhang, Yanhong Xu, Yanru Ma, Yanwen Huang, Yao Li, Yao Li, Yao Xu, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Qian, Yi Shao, Yi Yu, Yichao Zhang, Yifan Ding, Yifan Shi, Yijia Wu, Yiliang Xiong, Yiling Ma, Ying He, Ying Tang, Ying Zhou, Yingjia Luo, Yinmin Zhong, Yishi Piao, Yisong Wang, Yixiang Zhang, Yixiao Chen, Yixuan Tan, Yixuan Wei, Yiyang Ma, Yiyuan Liu, Yonglun Yang, Yongqiang Guo, Yongtong Wu, Yu Wu, YuKun Li, Yuan Cheng, Yuan Ou, Yuanfan Xu, Yuanhao Li, Yuduan Wang, Yuehan Yang, Yuer Xu, Yuhan Wu, Yuhao Meng, Yuheng Zou, Yukun Zha, Yunfan Xiong, Yupeng Chen, Yuping Lin, Yuqian Cao, Yuqian Wang, Yushun Zhang, Yuting Yan, Yutong Lin, Yuxian Gu, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuxuan Zhou, Yuyang Zhou, Yuzhen Huang, ZF Wu, Zehao Wang, Zehua Zhao, Zehui Ren, Zekai Zhang, Zhangli Sha, Zhe Fu, Zhe Ju, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zheren Gao, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhixian Huang, Zhixuan Chen, Zhiyu Wu, Zhizhou Ren, Zhongyu Wu, Zhuoshu Li, Zhuping Zhang, Zian Xu, Zihao Wang, Zihua Qu, Zihui Gu, Zijia Zhu, Zilin Li, Zipeng Zhang, Ziwei Xie, Ziyi Gao, Ziyi Wan, Zizheng Pan, and Zongqing Yao. Deepseek-v4: Towards highly efficient million-token context intelligence, 2026. URL https://arxiv.org/abs/2606.19348.

Alexandre Drouin, Gaël Letarte, Frédéric Raymond, Mario Marchand, Jacques Corbeil, and François Laviolette. Interpretable genotype-to-phenotype classifiers with performance guarantees. *Scientific reports*, 9(1):4071, 2019.

Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, et al. Prottrans: toward understanding the language of life through self-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 44(10):7112–7127, 2022.

Adibvafa Fallahpour, Andrew Magnuson, Purav Gupta, Shihao Ma, Jack Naimer, Arnav Shah, Haonan Duan, Omar Ibrahim, Hani Goodarzi, Chris J. Maddison, and BO WANG. Biorea-

son: Incentivizing multimodal biological reasoning within a DNA-LLM model. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=mDjEKAwJOF.

Roman Feldbauer, Frederik Schulz, Matthias Horn, and Thomas Rattei. Prediction of microbial phenotypes based on comparative genomics. *BMC bioinformatics*, 16(Suppl 14):S1, 2015.

GLM-5-Team, :, Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, Chenzheng Zhu, Congfeng Yin, Cunxiang Wang, Gengzheng Pan, Hao Zeng, Haoke Zhang, Haoran Wang, Huilong Chen, Jiajie Zhang, Jian Jiao, Jiaqi Guo, Jingsen Wang, Jingzhao Du, Jinzhu Wu, Kedong Wang, Lei Li, Lin Fan, Lucen Zhong, Mingdao Liu, Mingming Zhao, Pengfan Du, Qian Dong, Rui Lu, Shuang-Li, Shulin Cao, Song Liu, Ting Jiang, Xiaodong Chen, Xiaohan Zhang, Xuancheng Huang, Xuezhen Dong, Yabo Xu, Yao Wei, Yifan An, Yilin Niu, Yitong Zhu, Yuanhao Wen, Yukuo Cen, Yushi Bai, Zhongpei Qiao, Zihan Wang, Zikang Wang, Zilin Zhu, Ziqiang Liu, Zixuan Li, Bojie Wang, Bosi Wen, Can Huang, Changpeng Cai, Chao Yu, Chen Li, Chengwei Hu, Chenhui Zhang, Dan Zhang, Daoyan Lin, Dayong Yang, Di Wang, Ding Ai, Erle Zhu, Fangzhou Yi, Feiyu Chen, Guohong Wen, Hailong Sun, Haisha Zhao, Haiyi Hu, Hanchen Zhang, Hanrui Liu, Hanyu Zhang, Hao Peng, Hao Tai, Haobo Zhang, He Liu, Hongwei Wang, Hongxi Yan, Hongyu Ge, Huan Liu, Huanpeng Chu, Jia'ni Zhao, Jiachen Wang, Jiajing Zhao, Jiamin Ren, Jiapeng Wang, Jiaxin Zhang, Jiayi Gui, Jiayue Zhao, Jijie Li, Jing An, Jing Li, Jingwei Yuan, Jinhua Du, Jinxin Liu, Junkai Zhi, Junwen Duan, Kaiyue Zhou, Kangjian Wei, Ke Wang, Keyun Luo, Laiqiang Zhang, Leigang Sha, Liang Xu, Lindong Wu, Lintao Ding, Lu Chen, Minghao Li, Nianyi Lin, Pan Ta, Qiang Zou, Rongjun Song, Ruiqi Yang, Shangqing Tu, Shangtong Yang, Shaoxiang Wu, Shengyan Zhang, Shijie Li, Shuang Li, Shuyi Fan, Wei Qin, Wei Tian, Weining Zhang, Wenbo Yu, Wenjie Liang, Xiang Kuang, Xiangmeng Cheng, Xiangyang Li, Xiaoquan Yan, Xiaowei Hu, Xiaoying Ling, Xing Fan, Xingye Xia, Xinyuan Zhang, Xinze Zhang, Xirui Pan, Xu Zou, Xunkai Zhang, Yadi Liu, Yandong Wu, Yanfu Li, Yidong Wang, Yifan Zhu, Yijun Tan, Yilin Zhou, Yiming Pan, Ying Zhang, Yinpei Su, Yipeng Geng, Yong Yan, Yonglin Tan, Yuean Bi, Yuhan Shen, Yuhao Yang, Yujiang Li, Yunan Liu, Yunqing Wang, Yuntao Li, Yurong Wu, Yutao Zhang, Yuxi Duan, Yuxuan Zhang, Zezhen Liu, Zhengtao Jiang, Zhenhe Yan, Zheyu Zhang, Zhixiang Wei, Zhuo Chen, Zhuoer Feng, Zijun Yao, Ziwei Chai, Ziyuan Wang, Zuzhou Zhang, Bin Xu, Minlie Huang, Hongning Wang, Juanzi Li, Yuxiao Dong, and Jie Tang. Glm-5: from vibe coding to agentic engineering, 2026. URL https://arxiv.org/abs/2602.15763.

Jesse P Harrison, Nicolas Gheeraert, Dmitry Tsigelnitskiy, and Charles S Cockell. The limits for life under multiple extremes. *Trends in microbiology*, 21(4):204–212, 2013.

Tamsin James, Ben Williamson, Peter Tino, and Nicole Wheeler. Whole-genome phenotype prediction with machine learning: open problems in bacterial genomics. *Bioinformatics*, 41(7):btaf206, 2025.

Yanrong Ji, Zhihan Zhou, Han Liu, and Ramana V Davuluri. Dnabert: pre-trained bidirectional encoder representations from transformers model for dna-language in genome. *Bioinformatics*, 37 (15):2112–2120, 2021.

Signe T Karlsen, Martin H Rau, Benjamín J Sánchez, Kristian Jensen, and Ahmad A Zeidan. From genotype to phenotype: computational approaches for inferring microbial traits relevant to the food industry. *FEMS Microbiology Reviews*, 47(4):fuad030, 2023.

Julia Koblitz, Lorenz Christian Reimer, Rüdiger Pukall, and Jörg Overmann. Predicting bacterial phenotypic traits through improved machine learning using high-quality, curated datasets. *Communications Biology*, 8(1):897, 2025.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

Jean-Christophe Lagier, Grégory Dubourg, Matthieu Million, Frédéric Cadoret, Melhem Bilen, Florence Fenollar, Anthony Levasseur, Jean-Marc Rolain, Pierre-Edouard Fournier, and Didier Raoult. Culturing the human microbiota and culturomics. *Nature Reviews Microbiology*, 16(9): 540–550, 2018.

Weicai Long, Yusen Hou, Junning Feng, Houcheng su, Shuo Yang, Donglin Xie, and Yanlin Zhang. GenomeQA: Benchmarking general large language models for genome sequence understanding. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 35771–35792, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1655. URL https://aclanthology.org/2026.acl-long.1655/.

Nancy Merino, Heidi S Aronson, Diana P Bojanova, Jayme Feyhl-Buska, Michael L Wong, Shu Zhang, and Donato Giovannelli. Living at the extremes: extremophiles and the limits of life in a planetary context. *Frontiers in microbiology*, 10:780, 2019.

Eric Nguyen, Michael Poli, Marjan Faizi, Armin Thomas, Michael Wornow, Callum Birch-Sykes, Stefano Massaroli, Aman Patel, Clayton Rabideau, Yoshua Bengio, et al. Hyenadna: Long-range genomic sequence modeling at single nucleotide resolution. *Advances in neural information processing systems*, 36:43177–43201, 2023.

Eric Nguyen, Michael Poli, Matthew G Durrant, Brian Kang, Dhruva Katrekar, David B Li, Liam J Bartie, Armin W Thomas, Samuel H King, Garyk Brixi, et al. Sequence modeling and design from molecular to genome scale with evo. *Science*, 386(6723):eado9336, 2024.

D Nichols, N Cahoon, EM Trakhtenberg, L Pham, A Mehta, A Belanger, Tanya Kanigan, Kim Lewis, and Epstein. Use of ichip for high-throughput in situ cultivation of "uncultivable" microbial species. *Applied and environmental microbiology*, 76(8):2445–2450, 2010.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

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.

Lynn J Rothschild and Rocco L Mancinelli. Life in extreme environments. *Nature*, 409(6823): 1092–1101, 2001.

Melissa Sanabria, Jonas Hirsch, Pierre M Joubert, and Anna R Poetsch. Dna language model grover learns sequence context in the human genome. *Nature Machine Intelligence*, 6(8):911–923, 2024.

David B Sauer and Da-Neng Wang. Predicting the optimal growth temperatures of prokaryotes using only genome derived features. *Bioinformatics*, 35(18):3224–3231, 2019.

Eric W Sayers, Jeffrey Beck, Evan E Bolton, Devon Bourexis, James R Brister, Kathi Canese, Donald C Comeau, Kathryn Funk, Sunghwan Kim, William Klimke, et al. Database resources of the national center for biotechnology information. *Nucleic acids research*, 49(D1):D10–D17, 2021.

Isabel Schober, Julia Koblitz, Joaquim Sardà Carbasse, Christian Ebeling, Marvin Leon Schmidt, Adam Podstawka, Rohit Gupta, Vinodh Ilangovan, Javad Chamanara, Jörg Overmann, et al. Bac dive in 2025: the core database for prokaryotic strain data. *Nucleic acids research*, 53(D1): D748–D756, 2025.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, Jianfeng Cai, Xinyuan Cai, Peizhou Cao, Yuxuan Cao, Ziwei Chai, Y Charles, et al. Kimi k3: Open frontier intelligence. *arXiv preprint arXiv:2607.24653*, 2026.

Marcell Veiner and Fran Supek. The dna dialect: a comprehensive guide to pretrained genomic language models. *Molecular Systems Biology*, 22:309 – 332, 2026. URL https://api.semanticscholar.org/CorpusID:284882728.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. TRL: Transformers Reinforcement Learning, 2020. URL `https://github.com/huggingface/trl`.

Aaron Weimann, Kyra Mooren, Jeremy Frank, Phillip B Pope, Andreas Bremges, and Alice C McHardy. From genomes to phenotypes: Traitar, the microbial trait analyzer. *MSystems*, 1(6), 2016.

Lin Xu, Hong Chen, Xiaohua Hu, Rongmei Zhang, Ze Zhang, and ZW Luo. Average gene length is highly conserved in prokaryotes and eukaryotes and diverges only between the two kingdoms. *Molecular biology and evolution*, 23(6):1107–1108, 2006.

Itisha Yadav, Sirko Schindler, Diana Peters, and Roman Klinger. External knowledge integration in large language models: A survey on methods, challenges, and future directions. *Semantic Web*, 17 (4):22104968261453132, 2026.

A Theoretical Justification and Proofs

This section provides the theoretical justification for proposal-conditioned genomic utility. We first state the reference-sampling assumption and prove Theorem 1, then characterize the effect of additional reference-selection constraints. Throughout, the phenotype task and model parameters are fixed. We use natural logarithms, interpret predictive likelihoods as probability masses over decoder outputs, and assume all expectations are finite.

Assumption 1 (Proposal-conditioned reference sampling)**.** *Conditional on A, the genomic alternatives $X_1, \dots, X_K$ are independent copies of X, and $X_{1:K}$ is independent of $(X, Y, \hat{Y})$.*

Theorem 1 (Proposal-Conditioned Genomic Utility)**.** *For a fixed proposal condition A, let $X_1, \dots, X_K$ be independent proposal-matched genomic segment alternatives, with $Z_k = f_{\theta_{\mathrm{enc}}}(X_k)$. Define genomic utility as $\mathcal{U}_K = \mathbb{E}\Big[\log \pi_{\theta_{\mathrm{dec}}}(\hat{Y} \mid Z, A) - \log\Big(\frac{1}{K}\sum_{k=1}^{K} \pi_{\theta_{\mathrm{dec}}}(\hat{Y} \mid Z_k, A)\Big)\Big]$. Under Assumption 1, if the predictive likelihoods are uniformly bounded below by $0 < \delta \le 1$ over the relevant factual and reference evaluations, then*

$$\mathcal{U}_K - I(X; Z \mid Y, A) - \frac{(1-\delta)^2}{8\delta^2 K} \le I(Y; \hat{Y} \mid A) \le \mathcal{U}_K.$$

Proof. We first relate $\mathcal{U}_K$ to the genomic information used by the decoder. Since $\hat{Y}$ is generated only from (Z, A), its proposal-conditioned marginal is $p_\theta(\hat{y} \mid A) = \mathbb{E}[\pi_{\theta_{\mathrm{dec}}}(\hat{y} \mid Z, A) \mid A]$. Hence, $I(Z; \hat{Y} \mid A) = \mathbb{E}[\log \pi_{\theta_{\mathrm{dec}}}(\hat{Y} \mid Z, A) - \log p_\theta(\hat{Y} \mid A)]$.

Let $\widehat{p}_K(\hat{y} \mid A) = K^{-1}\sum_{k=1}^{K} \pi_{\theta_{\mathrm{dec}}}(\hat{y} \mid Z_k, A)$. Under Assumption 1, the representations $Z_1, \dots, Z_K$ have the same conditional distribution as Z and are independent of $\hat{Y}$ given A. Therefore, $\mathbb{E}[\widehat{p}_K(\hat{Y} \mid A) \mid \hat{Y}, A] = p_\theta(\hat{Y} \mid A)$. By concavity of the logarithm, $\mathbb{E}[\log \widehat{p}_K(\hat{Y} \mid A) \mid \hat{Y}, A] \le \log p_\theta(\hat{Y} \mid A)$, which implies $I(Z; \hat{Y} \mid A) \le \mathcal{U}_K$.

We next bound the finite-K bias. Conditional on $(\hat{Y}, A)$, write $V = \widehat{p}_K(\hat{Y} \mid A)$ and $\mu = p_\theta(\hat{Y} \mid A)$. The reference likelihoods are independent random variables in $[\delta, 1]$, so $\mathbb{E}[V \mid \hat{Y}, A] = \mu$ and $\mathrm{Var}(V \mid \hat{Y}, A) \le (1-\delta)^2/(4K)$. For $g(v) = -\log v$, the bound $g''(v) \le \delta^{-2}$ on this interval gives $g(v) \le g(\mu) + g'(\mu)(v-\mu) + (v-\mu)^2/(2\delta^2)$. Taking conditional expectations eliminates the linear term and yields $0 \le \mathbb{E}[\log \mu - \log V \mid \hat{Y}, A] \le (1-\delta)^2/(8\delta^2 K)$. Averaging over $(\hat{Y}, A)$ therefore establishes

$$0 \le \mathcal{U}_K - I(Z; \hat{Y} \mid A) \le \frac{(1-\delta)^2}{8\delta^2 K}. \tag{3}$$

We now connect decoder-level genomic information to predictive information about Y. Because the decoder receives only (Z, A), we have $Y \perp\!\!\!\perp \hat{Y} \mid Z, A$. The chain rule consequently gives $I(Z; \hat{Y} \mid A) = I(Y; \hat{Y} \mid A) + I(Z; \hat{Y} \mid Y, A)$. Since Z is a deterministic function of X, conditional data processing further yields $I(Z; \hat{Y} \mid Y, A) = I(X; \hat{Y} \mid Y, A) \le I(X; Z \mid Y, A)$. Thus, $I(Y; \hat{Y} \mid A) \le I(Z; \hat{Y} \mid A)$ and $I(Y; \hat{Y} \mid A) \ge I(Z; \hat{Y} \mid A) - I(X; Z \mid Y, A)$. Combining these inequalities with Eq. 3 establishes the stated bounds. □

Effect of additional reference-selection constraints. The theorem assumes that reference alternatives follow the same proposal-conditioned distribution as the factual input. Additional restrictions on phenotype labels or source identities can change this distribution. We characterize this discrepancy separately from the finite-K bias.

Let q denote a reference-selection rule that may depend on (X, Y, A), but is independent of decoder sampling conditional on these variables. Define its empirical predictive mixture as $\widehat{m}_{q,K}(\hat{y}) = K^{-1}\sum_{k=1}^{K} \pi_{\theta_{\mathrm{dec}}}(\hat{y} \mid Z_k, A)$ and its conditional mean as $m_q(\hat{y} \mid X, Y, A) = \mathbb{E}_q[\widehat{m}_{q,K}(\hat{y}) \mid X, Y, A]$. Let $\mathcal{U}_K^{(q)}$ denote genomic utility evaluated under this selection rule.

We distinguish the reference-distribution discrepancy $\mathcal{B}_q = \mathbb{E}[\log p_\theta(\hat{Y} \mid A) - \log m_q(\hat{Y} \mid X, Y, A)]$ from the finite-reference bias $e_{q,K} = \mathbb{E}[\log m_q(\hat{Y} \mid X, Y, A) - \log \widehat{m}_{q,K}(\hat{Y})]$. Conditional independence of reference selection and decoder sampling gives $\mathbb{E}_q[\widehat{m}_{q,K}(\hat{Y}) \mid X, Y, A, \hat{Y}] = m_q(\hat{Y} \mid$

X, Y, A). Jensen's inequality therefore implies $e_{q,K} \geq 0$, without requiring mutually independent alternatives. Adding and subtracting the corresponding reference log-likelihoods yields

$$\mathcal{U}_K^{(q)} = I(Y; \hat{Y} \mid A) + I(Z; \hat{Y} \mid Y, A) + \mathcal{B}_q + e_{q,K}. \tag{4}$$

Using $0 \leq I(Z; \hat{Y} \mid Y, A) \leq I(X; Z \mid Y, A)$, we obtain

$$I(Y; \hat{Y} \mid A) \geq \mathcal{U}_K^{(q)} - \mathcal{B}_q - I(X; Z \mid Y, A) - e_{q,K},$$
$$I(Y; \hat{Y} \mid A) \leq \mathcal{U}_K^{(q)} - \mathcal{B}_q.$$

Under Assumption 1, $m_q(\hat{y} \mid X, Y, A) = p_\theta(\hat{y} \mid A)$, so $\mathcal{B}_q = 0$. Additional label and identity constraints need not preserve this equality, and $\mathcal{B}_q$ has no guaranteed sign. Increasing K can reduce finite-reference bias under appropriate sampling assumptions, but does not itself remove the reference-distribution discrepancy. Accordingly, the implemented token-level contrast serves as a theory-motivated credit signal rather than an exact estimator of predictive mutual information.

This reference analysis applies to examples with valid alternatives. When restricted to such examples, all distributions and information quantities are evaluated under the same conditional data distribution. Assigning zero genomic credit to unmatched examples is an optimization convention rather than an evaluation of the theoretical contrast.

B Data Collection and Benchmark Construction

B.1 Raw Data Collection and Phenotype Harmonization

Strain collection from IJSEM. We construct the raw strain pool from taxonomic descriptions published in the *International Journal of Systematic and Evolutionary Microbiology* [1] (IJSEM). We process 18,498 IJSEM articles published before January 16, 2026. Given the article text and extracted tables, an LLM-assisted pipeline first retrieves strain-relevant evidence and then identifies the primary strains under study. The retrieved evidence is preserved with its attribution context, while the original scientific name, strain identifier, type-strain notation, and reported collection aliases are retained for each strain. IJSEM is used only to establish strain identities and cross-database correspondence; its physiological descriptions are not used as prediction targets in the present dataset.

Strain normalization and cross-database alignment. The article-level strain strings are normalized only for database lookup. The scientific name is reduced to a valid binomial form, while the primary strain identifier is retained as a separate modifier; nomenclatural qualifiers, type-strain suffixes, and collection-code aliases are removed from the lookup key but preserved in the original record. We first query NCBI with the normalized scientific name and strain modifier and retain only matches for which the species- and strain-level identifiers consistently refer to the same organism. Ambiguous matches, species-only matches, and records without sufficient strain-level support are excluded. The matched NCBI assembly is then fixed as the genomic source for the strain.

Phenotype collection from BacDive. Using the same normalized strain identities, the NCBI-matched strains are cross-referenced with BacDive. All phenotype targets used for training and evaluation are taken from structured BacDive annotations. The final phenotype scope contains 17 tasks: growth temperature, pH, and salinity ranges; optimuml temperature, pH, and salinity; carbon source, electron donor, electron acceptor, and nitrogen source; oxygen tolerance and energy type; and cell shape, motility, flagella, spore formation, and Gram stain. The original BacDive values are retained alongside their normalized targets so that every benchmark label can be traced back to its source annotation.

Expert phenotype harmonization. BacDive annotations vary substantially in terminology and granularity, particularly for categorical and metabolic phenotypes. Continuous measurements are standardized to task-specific numeric representations. For categorical and multilabel tasks, domain experts manually define deterministic mappings from raw BacDive values to controlled phenotype vocabularies. Synonymous or biologically equivalent surface forms are merged, naturally co-occurring labels are preserved for multilabel tasks, and annotations that are ambiguous or incompatible with the target definition are excluded. These mappings are applied before support-based label filtering and are not produced by a language model. Table 6 summarizes the mapping spaces.

[1] https://www.microbiologyresearch.org/content/journal/ijsem

Table 6: Summary of expert harmonization for discrete BacDive phenotypes. Raw values are mapped to expert-defined canonical labels, with ambiguous entries excluded.

Phenotype	Raw values	Canonical labels	Excluded
Oxygen tolerance	9	4	0
Energy type	22	4	2
Cell shape	18	8	0
Motility	2	2	0
Flagella	10	5	0
Spore formation	2	2	0
Gram stain	3	2	1
Carbon source	516	14	34
Electron donor	84	12	3
Electron acceptor	56	9	10
Nitrogen source	171	8	56

Complete Expert Mappings To make the expert curation directly auditable, we group all raw BacDive annotations by the canonical label to which they are mapped. Each box corresponds to one final expert-defined label before support-based filtering; the number in parentheses gives the number of distinct raw annotation strings assigned to that label. Entries in the *Excluded* boxes are removed during expert curation. Annotation frequencies are omitted here for readability and are retained in the released mapping files.

Oxygen tolerance.

`aerobic` — 2 raw values

aerobe; obligate aerobe.

`anaerobic` — 4 raw values

anaerobe; obligate anaerobe; aerotolerant; microaerotolerant.

`facultative` — 2 raw values

facultative anaerobe; facultative aerobe.

`microaerophilic` — 1 raw values

microaerophile.

Energy type.

`chemoorganoheterotroph` — 9 raw values

chemoorganotroph; chemoheterotroph; heterotroph; chemoorganoheterotroph; organotroph; methylotroph; organoheterotroph; chemotroph; methanotroph.

`chemolithoautotroph` — 6 raw values

chemolithoautotroph; autotroph; chemolithotroph; chemoautolithotroph; chemoautotroph; lithotroph.

`phototroph` — 3 raw values

phototroph; photoheterotroph; photolithotroph.

`chemolithoheterotroph` — 2 raw values

lithoheterotroph; chemolithoheterotroph.

Excluded	2 raw values
diazotroph; oligotroph.	

Cell shape.

`rod`	1 raw values
rod-shaped.	

`coccoid`	3 raw values
coccus-shaped; sphere-shaped; diplococcus-shaped.	

`ovoid`	2 raw values
ovoid-shaped; oval-shaped.	

`filamentous`	1 raw values
filament-shaped.	

`spiral`	2 raw values
spiral-shaped; helical-shaped.	

`pleomorphic`	1 raw values
pleomorphic-shaped.	

`vibrio`	3 raw values
vibrio-shaped; curved-shaped; crescent-shaped.	

`other`	5 raw values
other; spore-shaped; star-shaped; dumbbell-shaped; flask-shaped.	

Motility.

`true`	1 raw values
true.	

`false`	1 raw values
false.	

Flagella.

`gliding`	1 raw values
gliding.	

`monotrichous`	5 raw values
polar; monotrichous, polar; monotrichous; subpolar; monopolar.	

`peritrichous`	1 raw values
peritrichous.	

lophotrichous	2 raw values
lophotrichous; polytrichous, monopolar.	

amphitrichous	1 raw values
amphitrichous.	

Spore formation.

true	1 raw values
true.	

false	1 raw values
false.	

Gram stain.

positive	1 raw values
positive.	

negative	1 raw values
negative.	

Excluded	1 raw values
variable.	

Carbon source.

monosaccharides	44 raw values
glucose; mannose; fructose; arabinose; galactose; xylose; rhamnose; ribose; D-fructose; D-mannose; D-galactose; D-glucose; L-arabinose; L-rhamnose; D-xylose; D-ribose; fucose; L-fucose; alpha-D-glucose; D-fucose; D-arabinose; L-sorbose; D-psicose; D-tagatose; D-lyxose; L-xylose; sorbose; D-rhamnose; D-sorbose; lyxose; tagatose; (-)-D-fructose; L-lyxose; psicose; 6-deoxy-d-galactose; beta-D-fructofuranose; erythrose; 2-deoxy-d-ribose; alpha-D-rhamnose; alpha-L-fucose; alpha-L-rhamnose; beta-d-glucose; D-glyceraldehyde; L-fructose.	

organic_acids	88 raw values
citrate; acetate; succinate; lactate; malate; pyruvate; propionate; methyl pyruvate; 3-hydroxybutyrate; adipate; malonate; 2-oxoglutarate; L-malate; bromosuccinate; valerate; acetoacetate; 2-oxobutanoate; fumarate; sodium citrate; 2-hydroxybutyrate; itaconate; 2-oxopentanoate; (S)-lactate; acetic acid; D-malate; suberic acid; methyl (R)-lactate; cis-aconitate; sodium acetate; monomethyl succinate; beta-hydroxybutyrate; succinamate; citric acid; trans-aconitate; azelaate; butyrate; tartrate; 4-hydroxybutyrate; (R)-malic acid; glutaric acid; (S)-malic acid; mesaconate; oxalate; sodium pyruvate; sodium butyrate; sodium malonate; sebacic acid; 1 % sodium lactate; sodium succinate; alpha-hydroxybutyrate; glutarate; sodium propionate; glycolate; L-lactate; L-tartrate; 4-oxopentanoate; D-tartrate; disodium malate; tricarballylate; (R)-lactate; glycerate; lactamide; oxaloacetate; citraconate; disodium fumarate; sodium lactate; D-lactic acid methyl ester; ferric citrate; meso-tartrate; pimelate; sodium tartrate; ascorbate; glyoxylate; isocitrate; maleic acid; aconitate; crotonate; D-lactate; sodium malate; alpha-hydroxyglutarate-gamma-lactone; D-glycerate; isobutyrate; isovalerate; lactone; methylmalonic acid; potassium sodium L-tartrate; sorbic acid; suberate.	

disaccharides	17 raw values
maltose; sucrose; cellobiose; trehalose; lactose; melibiose; turanose; gentiobiose; D-trehalose; alpha-lactose; lactulose; palatinose; maltose hydrate; leucrose; (2)-D-lactose; beta lactose; beta-lactose.	

amino_acids 91 raw values

glutamate; alanine; proline; serine; aspartate; L-glutamate; L-serine; L-alanine; asparagine; L-aspartate; L-histidine; histidine; threonine; arginine; leucine; gamma-aminobutyric acid; L-arginine; L-alaninamide; ornithine; glycine; phenylalanine; L-proline; D-serine; L-ornithine; L-leucine; L-phenylalanine; D-aspartate; glycine-proline; L-asparagine; L-pyroglutamic acid; L-threonine; urocanic acid; L-lysine; 4-hydroxy-L-proline; D-tryptophan; D-alanine; L-tryptophan; L-alanylglycine; beta-alanine; L-pyroglutamate; L-glutamine; glycyl-l-glutamate; L-cysteine; L-tyrosine; glycyl L-aspartic acid; L-alanine 4-nitroanilide; L-methionine; L-valine; glycyl-L-proline; hydroxyproline; L-isoleucine; sarcosine; methionine; glutamine; L-citrulline; cysteine; N-acetyl-L-glutamate; creatine; dl-alanine; lysine; valine; tyrosine; 4-aminobutyrate; glycyl-L-glutamate; isoleucine; 2-aminobutyrate; hippurate; tryptophan; 3-aminobutyrate; 5-aminovaleric acid; 5-oxoproline; citrulline; L-serine 2-naphthylamide; kynurenine; L-cystine; L-glutamate-gamma-3-carboxy-4-nitroanilide; sodium glutamate; sodium L-glutamate; 2-aminopentanoic acid; D-alloisoleucine zwitterion; D-methionine; D-ornithine; DL-2-gamma-aminobutyrate; glycyl-L-bromosuccinic glutamic acid; L-arginine-2-naphthylamide; L-cysteine 2-naphthylamide; L-glycine; L-histidine 2-naphthylamide; L-norleucine; L-pyroglutamic acid 2-naphthylamide; norleucine.

polyols 18 raw values

glycerol; mannitol; myo-inositol; sorbitol; D-mannitol; D-sorbitol; ribitol; xylitol; L-arabitol; D-arabitol; galactitol; erythritol; maltitol; glycerol 1-phosphate; glycerol 3-phosphate; arabinitol; (+)-L-lyxitol; L-inositol.

sugar_derivatives 61 raw values

N-acetylglucosamine; salicin; gluconate; glucuronamide; D-gluconate; methyl D-glucoside; D-glucuronate; N-acetylgalactosamine; D-galacturonic acid; D-glucose 6-phosphate; 5-dehydro-D-gluconate; D-fructose 6-phosphate; methyl beta-D-glucopyranoside; 3-O-methyl-D-glucose; galacturonate; 2-dehydro-D-gluconate; D-glucarate; galactarate; n-acetylneuraminate; L-galactonic acid gamma-lactone; potassium gluconate; arbutin; amygdalin; glucosamine; N-acetyl-D-mannosamine; esculin; methyl alpha-D-glucopyranoside; methyl alpha-D-mannoside; N-acetyl-beta-D-mannosamine; glucose 1-phosphate; D-saccharate; D-galactonic acid lactone; methyl beta-D-xylopyranoside; galactonate; D-glucosaminic acid; D-salicin; esculin ferric citrate; N-acetyl-D-glucosamine; potassium 5-dehydro-D-gluconate; potassium 2-dehydro-D-gluconate; 2-oxogluconate; n-acetyl-D-galactosamine; sodium gluconate; alpha-D-glucose 6-phosphate; glucuronate; methyl alpha-D-galactoside; 3-methylglucose; methyl beta-D-galactoside; beta-D-glucuronic acid; D-galactonate; D-glucosamine; 4-nitrophenyl beta-D-galactopyranoside; n-acetyl-neuraminic acid; (R)-amygdalin; L-galactonate; 3-O-methyl alpha-D-glucopyranoside; D-glucuronolactone; methyl alpha-D-xylopyranoside; methyl beta-D-mannoside; O-nitrophenyl-beta-D-galactopyranosid; uronic acid.

oligo_polysaccharides 39 raw values

raffinose; dextrin; glycogen; alpha-cyclodextrin; pectin; melezitose; stachyose; inulin; starch; cellulose; dextran; gamma-cyclodextrin; maltotriose; xylan; beta-cyclodextrin; cyclodextrin; mannan; sedoheptulosan; carboxymethylcellulose; chitin; agar; alginate; amylose; pullulan; sodium alginate; agarose; amylopectin; arabinan; arabinogalactan; carrageenan; chitosan; galactan; gellan gum; iota-carrageenan; kappa-carrageenan; laminarin; My-carrageenan; Ny-carrageenan; xanthan.

fatty_acids_lipids 11 raw values

tween 80; tween 40; tween 20; decanoate; tween 60; hexanoate; octanoate; nonanoate; heptanoate; hexadecanoate; octadecanoate.

aromatics_hydrocarbons 45 raw values

phenylacetate; quinate; 4-hydroxyphenylacetic acid; 4-hydroxybenzoate; benzoate; 3-hydroxybenzoate; (-)-quinic acid; gentisate; indole-3-acetic acid; trigonelline; salicylate; coumarate; phenol; trimethoxybenzoate; (R)-mandelate; 2-aminobenzoate; 3-aminobenzoate; 3-phenylpropionate; 4-aminobenzoate; catechol; hexadecane; isophthalate; phenanthrene; phthalate; protocatechuate; terephthalate; (S)-mandelic acid; 4-coumarate; 4-hydroxybenzaldehyde; acetylsalicylate; anthracene;

benzoin; chloridazon; diphenylamine; fluorene; indole; mandelate; melanin; nicotinate; pyrene; sodium benzoate; testosterone; tetraphene; vanillate; xylene.

nucleobases 13 raw values

inosine; uridine; thymidine; adenosine; uridine 5'-monophosphate; 2-deoxyadenosine; thymidine 5'-monophosphate; amp; adenine; guanine; cytosine; hypoxanthine; nucleoside.

c1_compounds 10 raw values

formate; methanol; methylamine; sodium formate; carbon dioxide; methane; dimethylamine; formaldehyde; trimethylamine; hydrogencarbonate.

amines_amides 17 raw values

putrescine; urea; DL-carnitine; phenylethylamine; ethanolamine; betaine; histamine; tryptamine; acetamide; ethylamine; benzylamine; beta-phenylethylamine; butan-1-amine; carnitine; pentan-1-amine; spermine; diethylamine.

complex_media 13 raw values

gelatin; yeast extract; casamino acids; peptone; tryptone; casein; polypeptone; soyton; yeast extract (0.01 %, w/v); casamino acids (0.01 %, w/v); casein hydrolysate; dna; Peptidoglycan(N-acetyl-D-glucosamine)p.

alcohols 15 raw values

ethanol; 2,3-butanediol; butanol; propanol; 2-propanol; acetone; ethylene glycol; 1,2-propandiol; 1,4-propandiol; 2,4-butanediol; 2-butanol; acetoin; butane-1,4-diol; pentanol; propane-1,3-diol.

Excluded 34 raw values

fusidate; nalidixic acid; lithium chloride; potassium tellurite; sodium bromate; aztreonam; guanidinium chloride; tetrazolium violet; tetrazolium blue; lincomycin; niaproof; rifamycin sv; thiosulfate; troleandomycin; ammonium chloride; ammonium nitrate; ammonium sulfate; antipyrine; minocycline; nitrate; potassium nitrate; sodium nitrate; sodium sulfate; spectinomycin; sulfide; thiamine; thioglycolate; vancomycin; ammonium; dihydrogen; dithionitrobenzoic acid; polyethylene glycol; sulfate; thioacetamide.

Electron donor.

c1_compounds 3 raw values

formate; methanol; methane.

organic_acids 20 raw values

fumarate; pyruvate; succinate; acetate; lactate; malate; propionate; butyrate; citrate; decanoate; valerate; tartrate; 2-oxoglutarate; ascorbate; glycolate; crotonate; hexadecanoate; maleic acid; oxalate; thioglycolate.

alcohols 3 raw values

ethanol; propanol; butanol.

hydrogen 2 raw values

dihydrogen; diydrogen.

polyols 4 raw values

glycerol; mannitol; sorbitol; D-sorbitol.

reduced_sulfur	8 raw values
thiosulfate; sulfide; elemental sulfur; tetrathionate; sulfite; thiocyanate; trithionate; hydrogen sulfide.	

aromatics	6 raw values
benzoate; benzene; hexane; phenol; toluene; xylene.	

sugars	16 raw values
glucose; fructose; sucrose; lactose; maltose; starch; D-glucose; xylose; D-fructose; D-galactose; D-mannose; D-ribose; galactose; gluconate; mannose; melibiose.	

reduced_metals	2 raw values
iron; manganese.	

complex_media	3 raw values
yeast extract; casamino acids; peptone.	

amino_acids	12 raw values
aspartate; cysteine; serine; alanine; glutamate; glycine; L-glutamate; methionine; proline; choline; isoleucine; L-serine.	

inorganic_nitrogen	2 raw values
ammonium; nitrite.	

Excluded	3 raw values
ferric iron; iodide; sulfate.	

Electron acceptor.

nitrogen_oxides	5 raw values
nitrate; nitrite; potassium nitrate; sodium nitrate; sodium nitrite.	

sulfur_oxides	7 raw values
thiosulfate; sulfate; sulfite; tetrathionate; sodium sulfate; sodium sulfite; sodium thiosulfate.	

elemental_sulfur	1 raw values
elemental sulfur.	

organic_acids	8 raw values
fumarate; acetate; malate; propionate; benzoate; butyrate; succinate; citrate.	

oxygen	1 raw values
dioxygen.	

iron_manganese	10 raw values
ferric iron; ferric oxide; ferrihydrite; manganese dioxide; amorphous iron (iii) oxide; ferric citrate; amorphous fe(iii) oxyhydroxid; fe(iii) citrate; iron trichloride; manganese.	

organosulfur	2 raw values
dimethyl sulfoxide; dimethyl sulfide.	

metalloids_heavy_metals	7 raw values
arsenate; selenate; chromate; selenite; antimonate; molybdate; vanadate.	

methylated	5 raw values
methanol; trimethylamine n-oxide; betaine; choline; methylamine.	

Excluded	10 raw values
aspartate; alanine; glycine; serine; glutamate; hydrogen sulfide; L-arginine; L-cystine; mannitol; sorbitol.	

Nitrogen source.

amino_acids	78 raw values
L-serine; L-arginine; glycine; L-alanine; L-proline; L-tyrosine; L-asparagine; L-threonine; L-glutamate; L-aspartate; L-histidine; L-valine; L-glutamine; L-cysteine; L-methionine; L-lysine; L-phenylalanine; creatine; L-tryptophan; glutamate; hydroxyproline; L-isoleucine; aspartate; glutamine; L-leucine; isoleucine; tryptophan; alanine; arginine; L-alanine 4-nitroanilide; L-ornithine; D-serine; tyrosine; valine; asparagine; L-cystine; gamma-aminobutyric acid; methionine; proline; phenylalanine; threonine; cysteine; glycine-proline; lysine; serine; 2-aminobutyrate; L-pyroglutamic acid; leucine; 4-hydroxy-L-proline; DL-2-gamma-aminobutyrate; dl-alanine; histidine; hydroxy-L-proline; N-glutaryl-L-phenylalanine 2-naphthylamide; ornithine; arginine hydrochloride; D-aspartate; D-proline; D-valine; glycyl L-aspartic acid; glycyl-l-glutamate; L-cysteine 2-naphthylamide; sarcosine; sodium L-glutamate; 2-aminopentanoic acid; cystine; D-alanine; D-leucine; D-threonine; D-tryptophan; glycyl-L-proline; L-arginine-2-naphthylamide; L-glycine; L-phenylalanine 2-naphthylamide; L-proline-4-nitroanilide; L-serine 2-naphthylamide; L-valine 2-naphthylamide; norleucine.	

ammonium	6 raw values
ammonium; ammonium chloride; ammonium sulfate; ammonia; ammonium nitrate; diammonium hydrogen phosphate.	

nitrate_nitrite	5 raw values
nitrate; nitrite; potassium nitrate; sodium nitrate; sodium nitrite.	

nucleobases	5 raw values
hypoxanthine; adenine; xanthine; inosine; thymine.	

complex_media	8 raw values
peptone; casamino acids; yeast extract; casein; casein hydrolysate; gelatin; tryptone; yeast extract (0.01 %, w/v).	

urea_other_n_organics	6 raw values
urea; 4-aminobenzoate; betaine; succinimide; guanidinium chloride; propionamide.	

amino_sugars	5 raw values
N-acetylglucosamine; glucuronamide; N-acetylgalactosamine; N-acetyl-beta-D-mannosamine; n-acetyl-neuraminic acid.	

dinitrogen	2 raw values
dinitrogen; dinitrogen oxide.	

Excluded	56 raw values
myo-inositol; sucrose; arabinose; glucose; mannose; rhamnose; raffinose; amygdalin; melezitose; sorbose; trehalose; cellobiose; D-galactose; glycerol; maltose; melibiose; D-mannitol; D-mannose; D-sorbitol; fructose; galactose; mannitol; ribitol; sodium acetate; sodium citrate; sodium propionate; sulfate; xylitol; xylose; (S)-lactate; 2-hydroxybutyrate; acetate; acetoacetate; alpha-D-glucose; citrate; D-arabitol; D-fructose; D-fructose 6-phosphate; D-gluconate; D-glucose; D-ribose; D-xylose; dextrin; gentiobiose; L-arabinose; L-fucose; L-rhamnose; lactose; methyl beta-D-glucopyranoside; pectin; salicin; stachyose; thiocyanate; thiosulfate; turanose; tween 40.	

B.2 Dataset Construction and Benchmark Composition

Genome segmentation. For each aligned NCBI assembly, we sample contiguous 2,048-bp genomic segments as model inputs. Each segment is associated with the strain-level annotation of the queried phenotype, while the assembly accession, strain identity, and genomic coordinates are retained for reproducibility. A strain may contribute annotations to multiple prediction tasks, and a strain–task record may contribute multiple genomic segments during training or evaluation. We therefore report strain–task records, unique strains, and genomic segments separately throughout the dataset statistics.

Strain-level data separation. All data separation is performed at the strain level before genomic segmentation, proposal generation, or reasoning-trajectory construction. Records associated with the same strain remain within the same data source, ensuring that genomic segments from one biological strain cannot appear across the training data for phenotype-specific genomic specialization and genome-to-language grounding, the S2D-GRPO training corpus, and the final benchmark. Assembly identities, phenotype records, genomic segments, cached genomic representations, phenotype proposals, and reasoning trajectories inherit the assignment of their source strain. The training strains used for genomic specialization and grounding, the S2D-GRPO training strains, and the final benchmark strains are therefore mutually disjoint. The final benchmark is maintained as a single held-out evaluation set and is not further partitioned.

Balanced training for genomic specialization and grounding. As shown in Table 7, the training partition contains 37,722 strain-task records from 7,465 unique strains. To mitigate phenotype imbalance, we vary the number of continuous 2,048-bp genomic segments sampled from each record. Segment allocation approximately balances target classes for binary and multiclass phenotypes, prioritizes underrepresented positive labels for multilabel phenotypes, and balances empirical target-midpoint quantiles for continuous phenotypes. This produces 397,595 training segments for phenotype-specific genomic specialization without introducing additional strains or phenotype annotations. Genome-to-language grounding inherits the same strain partition and sampling scheme, retaining samples with valid reasoning trajectories. The validation and internal-test sets contain 2,290 and 4,576 records from 424 and 861 strains, respectively, and are not rebalanced.

Genome-to-language grounding supervision. We construct protein-informed reasoning trajectories to ground the specialized genomic representations in the LLM decoder. The trajectories are aligned with genomic segments only after the strain partitions and genomic sampling have been fixed, and only those validated and rewritten by Qwen3.5-27B are retained. As detailed in Appendix B.3, the validation removes unsupported or inconsistent biological claims. This results in 394,526 training examples, 2,274 validation examples, and 4,545 internal-test examples for genome-to-language grounding. These trajectories serve as auxiliary supervision without changing the underlying strains, phenotype annotations, or genomic inputs.

S2D-GRPO corpus construction. The S2D-GRPO corpus is constructed from a separate set of training strains and retains at most one source record for each strain–task pair. For tasks with at least 512 eligible records, we retain 512 unique records, whereas lower-resource tasks retain all available records. Selection within each task is phenotype-aware. Binary and multiclass records are selected to improve class coverage, multilabel records are selected to improve the marginal coverage of positive labels, and continuous records are selected across empirical quantiles of the target midpoint and interval width. The resulting corpus contains 6,857 strain–task records from 2,511 unique strains and

Table 7: Composition of the balanced training corpus for genomic specialization and grounding.

Group	Phenotype	Type	Records	Strains	Segments
Phys. boundaries	Temperature	Regression	2,426	2,426	36,000
	pH	Regression	1,970	1,970	32,000
	Salinity	Regression	2,012	2,012	40,000
Phys. optimum	Temperature	Regression	2,031	2,031	32,000
	pH	Regression	1,947	1,947	24,000
	Salinity	Regression	1,851	1,851	25,200
Metabolism	Carbon source	Multilabel	2,218	2,218	14,997
	Electron acceptor	Multilabel	55	55	4,567
	Electron donor	Multilabel	54	54	5,738
	Nitrogen source	Multilabel	112	112	5,493
Categorical phys.	Oxygen tolerance	Multiclass	5,189	5,189	48,000
	Energy type	Multiclass	336	336	4,800
Morphology	Cell shape	Multiclass	4,691	4,691	35,200
	Motility	Binary	4,623	4,623	24,000
	Flagella	Multiclass	732	732	17,600
	Spore formation	Binary	2,631	2,631	24,000
	Gram stain	Multiclass	4,844	4,844	24,000
Total	**17 phenotypes**	–	**37,722**	**7,465**	**397,595**

covers all 17 prediction tasks. Each record is associated with one cached 2,048-bp genomic segment, yielding 6,857 genomic inputs before runtime sampling.

Dual phenotype proposals. Each S2D-GRPO record is augmented with the two auxiliary phenotype proposals used for proposal-conditioned refinement. The first is produced by the phenotype-specialized Evo2–MLP predictor, while the second is produced by the genome-grounded LLM decoder after genome-to-language grounding. Both proposals are generated from the same genomic segment and remain fixed during S2D-GRPO optimization. The phenotype target is used only offline to determine proposal correctness and is never included in the policy input. Among the 6,857 records, 2,841 have both proposals correct, 637 have only the Evo2–MLP proposal correct, 554 have only the genome-grounded LLM proposal correct, and 2,825 have neither proposal correct. Exact agreement is used for binary and multiclass tasks, while multilabel and continuous tasks use the corresponding task-level score and a fixed acceptance threshold. These proposal relations are used only for training-time sampling and are not presented to the LLM decoder.

Balanced S2D-GRPO sampling. Balancing during S2D-GRPO is implemented by the runtime sampler rather than by storing multiple genomic segments for the same record. The immutable corpus contains 6,857 unique records and is expanded to 13,672 prompt groups per nominal epoch through deterministic sampling with replacement. Each generation batch contains eight prompt groups, with two sampled from each of the four proposal-relation groups. One epoch therefore contains 1,709 balanced batches and 3,418 prompt groups from each relation. Every unique S2D-GRPO record is presented at least once.

Reasoning supervision for S2D-GRPO. Each S2D-GRPO record is paired with one Qwen3.5-27B-curated reasoning trajectory after the strain identity, genomic segment, and frozen phenotype proposals have been fixed. Only the final evidence-grounded trajectory is retained, while rejected candidates and intermediate generations are excluded. The resulting S2D-GRPO corpus therefore contains 6,857 genomic records and 6,857 aligned reasoning trajectories. Detailed protein-evidence selection, rationale refinement, and quality-control procedures are provided in Appendix B.3.

Final benchmark. As shown in Table 8, the final benchmark is a single fixed evaluation set containing 2,080 strain–task records from 745 unique strains across all 17 prediction tasks. It is not further partitioned, expanded, or rebalanced for evaluation. Because one strain may be annotated for multiple phenotypes, the number of strain–task records exceeds the number of unique strains. The genomic

Table 8: Task-wise composition of the S2D-GRPO training corpus and final benchmark. Total strain counts are deduplicated across phenotypes.

			S2D-GRPO training corpus			Final benchmark	
Group	Phenotype	Type	Records	Strains	Segments	Records	Strains
Phys. boundaries	Temperature	Regression	512	512	512	173	173
	pH	Regression	512	512	512	152	152
	Salinity	Regression	512	512	512	180	180
Phys. optimum	Temperature	Regression	512	512	512	135	135
	pH	Regression	512	512	512	110	110
	Salinity	Regression	512	512	512	174	174
Metabolism	Carbon source	Multilabel	512	512	512	123	123
	Electron acceptor	Multilabel	66	66	66	9	9
	Electron donor	Multilabel	52	52	52	5	5
	Nitrogen source	Multilabel	139	139	139	11	11
Categorical phys.	Oxygen tolerance	Multiclass	512	512	512	193	193
	Energy type	Multiclass	149	149	149	26	26
Morphology	Cell shape	Multiclass	512	512	512	103	103
	Motility	Binary	512	512	512	200	200
	Flagella	Multiclass	307	307	307	87	87
	Spore formation	Binary	512	512	512	199	199
	Gram stain	Multiclass	512	512	512	200	200
Total	**17 phenotypes**	–	**6,857**	**2,511**	**6,857**	**2,080**	**745**

input associated with each benchmark record is sampled once and kept fixed across all evaluated models, ensuring consistent comparison on the same held-out records.

B.3 Protein-Informed Reasoning Trajectory Construction

We construct protein-informed reasoning trajectories as additional training supervision for connecting genomic evidence with microbial phenotypes. For each strain, we retrieve the protein sequences associated with its matched NCBI genome assembly and obtain functional annotations using eggNOG-mapper (Cantalapiedra et al., 2021). We primarily use the eggNOG functional descriptions as natural-language protein evidence, complemented by KEGG and BiGG reaction annotations when available. Since these annotations are computationally inferred from sequence homology and orthology, we treat them as functional evidence rather than experimentally verified facts.

Phenotype-conditioned protein evidence selection. Most proteins encoded by a strain are unrelated to a specific phenotype. We therefore first use Qwen3.5-9B to identify phenotype-relevant protein annotations for each strain–task pair. The model is provided with the phenotype task, its BacDive label, and candidate protein annotations, and is asked to return only proteins with a plausible mechanistic connection to the target phenotype. The phenotype label is used only to condition evidence retrieval and is not itself regarded as biological evidence.

Prompt for phenotype-conditioned protein evidence selection

System prompt.
You are a careful bacterial functional genomics curator. Your job is to identify which candidate protein annotations may be relevant to one BacDive phenotype/task.
Select a candidate only if its description or reaction has a plausible mechanistic link to the task. If none are relevant, return an empty list. Only return indices appearing in the provided candidate proteins, and do not invent or modify protein information. Do not provide chain-of-thought or additional prose.
Input.
Strain, phenotype task, task type, BacDive phenotype value, and candidate protein annotations containing functional descriptions and available KEGG/BiGG reactions.

Output.
`{"task": "...", "strain_name": "...", "relevant_indices": [0, 3, 7]}`

The returned indices are mapped back to the original protein annotations and validated before subsequent processing. This ensures that the selection model can only choose among existing annotations and cannot modify protein descriptions or introduce proteins absent from the corresponding strain.

Initial phenotype rationale generation. The selected protein annotations are then used to construct an initial phenotype-oriented rationale with Qwen3.5-9B. The model receives the phenotype task, the target phenotype as a private generation constraint, and the selected protein descriptions as supporting biological evidence. Its role is to explain how the available protein functions may mechanistically support the target phenotype, while avoiding unsupported biological claims. The final phenotype answer is constructed directly from the verified dataset label.

Prompt for initial phenotype rationale generation

System prompt.
You are a bacterial phenotype rationale writer.
You will receive only a phenotype task, a desired phenotype output to explain, and protein description text. Write a concise, biologically plausible rationale that makes the desired phenotype output a reasonable prediction from the supplied protein descriptions.
The phenotype value is a private generation constraint and must not be treated as evidence. Use only the provided protein descriptions, do not invent proteins, pathways, measurements, or strain properties, and use cautious language such as "may support", "is consistent with", or "suggests capacity for" when the evidence is indirect. For numerical environmental phenotypes, do not claim that protein annotations directly measure temperature, pH, or salinity.
Output.
`{"reasoning_trace": "2--4 concise biological sentences"}`

This stage provides a concise phenotype-oriented draft, but label-conditioned generation can still encourage overly strong biological explanations when the available evidence is weak. We therefore apply a second, stricter evidence-grounding stage.

Evidence-grounded rationale refinement. We use Qwen3.5-27B to reconstruct each initial rationale from the underlying protein evidence. Each protein description is assigned an explicit evidence index, and the 9B rationale is provided only as an editable draft. The refinement model is required to rebuild the connection from protein function to cellular or metabolic mechanism and ultimately to phenotype, while associating every retained claim with specific protein evidence. Claims may be labeled as `direct` or `indirect`; phenotype components without sufficient support should be omitted rather than forcefully explained.

Prompt for evidence-grounded rationale refinement

Input.
Phenotype task, phenotype to explain privately, evidence-indexed protein descriptions, and the initial rationale draft.
Instruction.
Rebuild the rationale from the provided protein descriptions. Use the draft only as an editable starting point. Remove unsupported claims and distinguish direct evidence from indirect support. Do not force an explanation for phenotype components that lack evidence.
Each claim must contain one complete biological sentence, one or more valid `evidence_indices`, and a `support_level` of either `direct` or `indirect`. General biological knowledge may be used to explain the consequence of an observed protein function, but must not introduce unobserved proteins, missing pathway components, or strain-specific facts. Absence of a protein from the provided evidence must not be interpreted as evidence of biological absence.
For metabolic substrate tasks, generic transport, central metabolism, biosynthesis, or salvage alone is insufficient to establish utilization of a specific substrate. For negative phenotypes such as `motility=false` or `spore_formation=false`, do not infer loss of the corresponding biological system solely from missing annotations.

Output.
{"claims": [{"text": "...", "evidence_indices": [0, 4], "support_level": "direct"}]}

Quality control and trajectory alignment. The refined claims are subjected to deterministic validation before being included in the training set. We remove claims with invalid or missing evidence references, unsupported biological assertions, supervision leakage, or conclusions inferred solely from missing annotations, ensuring that each retained statement remains traceable to the provided protein evidence. For metabolic phenotypes, we further reject overly strong substrate-utilization claims supported only by generic transporters, central metabolic reactions, or biosynthetic functions. Indirect evidence is also required to use appropriately calibrated language rather than being presented as definitive support.

After validation, the evidence indices are removed and the retained claims are concatenated into the final natural-language rationale. Each rationale is then aligned with its corresponding genomic-segment sample and verified phenotype answer through stable sample identifiers. The resulting trajectory therefore provides an auditable training target that connects protein-level functional evidence to phenotype reasoning, while the protein annotations themselves are never supplied during inference. The trajectories provide strain-level functional supervision and are not intended as locus-level annotations of the sampled genomic segment.

C Experimental Details

C.1 Baselines Implementation.

We provide implementation details for all baselines below, following the same task definitions and evaluation protocol as GGBound.

Evo2-MLP. We use Evo2-MLP to directly predict phenotypes from the specialized Evo2-7B encoder. Each example contains a 2,048-bp genomic crop and its phenotype label, with crops balanced across tasks and phenotypes. A phenotype-specific task prefix is prepended to the genomic sequence, and the 4,096-dimensional hidden state at the final position is used as the genomic representation. This representation is fed to a shared MLP with two 2,048-dimensional hidden layers and a 68-dimensional output layer, using input layer normalization, GELU activations, and 0.1 dropout. We use cross-entropy for categorical tasks, binary cross-entropy for multi-label tasks, and mean squared error on standardized regression targets. Encoder specialization further includes DNA reconstruction with a masking ratio of 0.05, label smoothing of 0.02, and loss weight 0.2. The encoder and MLP are jointly optimized for one epoch with AdamW using learning rates of 2×10^{-5} and 2×10^{-4}, respectively. Training uses a per-GPU batch size of 16 across eight GPUs, weight decay 10^{-3}, Adam coefficients $(0.9, 0.95)$, gradient clipping at 1.0, and cosine decay with a 0.1 warmup ratio.

G-Qwen. We connect the specialized Evo2-7B encoder to Qwen3.5-4B through a trainable genomic projector. For each 2,048-bp crop, the Evo2 hidden states are divided into 32 contiguous bins of 64 positions and averaged within each bin. The resulting representations are mapped to 32 continuous tokens in the Qwen embedding space using a projector with layer normalization, two linear layers, and GELU. These tokens are directly inserted into the language-model input without additional cross-attention. Detailed training hyperparameters are provided in Appendix C.3.

P-Qwen. P-Qwen uses the same frozen Evo2 encoder, genomic projector, Qwen3.5-4B checkpoint, and SFT configuration as G-Qwen. At inference time, we additionally provide phenotype proposals from the genome-only SFT model and Evo2-MLP. No further parameter updates are performed, and S2D-GRPO is not applied. This baseline therefore isolates proposal-conditioned prediction using the pre-RL language model.

Evo2-Prob. For Evo2-Prob, we freeze the original pretrained Evo2-7B encoder and train an MLP readout from scratch using the same crops and balanced training manifest as Evo2-MLP. The MLP takes the 4,096-dimensional final-position representation and follows the same architecture as Evo2-MLP. Task identity is used only to select the supervised output dimensions and is not provided to the encoder or concatenated with the genomic representation. We use cross-entropy, binary cross-entropy, and mean squared error on standardized targets for categorical, multilabel, and regression tasks,

Table 9: Phenotype-specific task codes. Each pattern is repeated $10\times$.

Group	Phenotype	Base pattern
Phys. boundary	Temperature range	`@#~@@~#@~#`
	pH range	`~@@#@##@~~`
	Salinity range	`~###@~~####@`
Phys. optimum	Temperature optimum	`~~@#~@@##~`
	pH optimum	`##~~#~@@#@`
	Salinity optimum	`@~#@~@~@##`
Metabolism	Carbon source	`~~#~#@#@~@`
	Electron donor	`#@##~~@@~#`
	Electron acceptor	`@#@@#~~##~`
	Nitrogen source	`~@~@##@~~#`
Categorical phys.	Oxygen tolerance	`@~@#@~#~#@`
	Energy type	`~~@~@#~#@#`
Morphology	Cell shape	`##~#@#@~@~`
	Motility	`#@#~~@~#@~`
	Flagella	`#@~@~@#~#@`
	Spore formation	`#~~#@@~#~@`
	Gram stain	`@##~##~~@@`

respectively. Only the MLP is optimized for one epoch with AdamW at a learning rate of 2×10^{-4}, while the remaining optimization settings follow Evo2-MLP. The Evo2 encoder remains frozen in evaluation mode throughout training, and no DNA reconstruction objective is used.

Frontier LLMs. GPT-5.6 Sol, Claude Opus 5, GLM-5.3, Kimi K3, and DeepSeek-V4 Pro are evaluated through their respective APIs without task-specific parameter updates. Each model is queried directly for phenotype prediction under the corresponding evaluation setting, with no additional fine-tuning or adaptation on the benchmark data.

C.2 Evo2 Backbone and Phenotype-Specific Specialization

Evo2 representation backbone. We initialize the genomic encoder from Evo2-7B. For each phenotype, the 100-character task-token sequence is prepended to the 2,048-bp genomic segment and processed jointly by the encoder. We extract the 4,096-dimensional activation from `blocks.28.mlp.l3` following Brixi et al. (2026). During phenotype-specific specialization, the representation at the final valid DNA position is used for phenotype prediction. The shared prediction head consists of LayerNorm followed by three linear layers with hidden width 2,048, GELU activations, and dropout 0.1:

$$4096 \to 2048 \to 2048 \to 68.$$

The 68 output dimensions are shared across all phenotypes. The first 25 dimensions form a global label space for the seven exact-prediction tasks, 31 dimensions are assigned to the four multilabel tasks, and the remaining 12 dimensions represent the two endpoints of the six continuous phenotypes. Exact-prediction tasks use cross-entropy, multilabel tasks use binary cross-entropy over their task-specific output slices, and continuous targets are standardized using training-set statistics and optimized with mean-squared error.

Phenotype-specific task-token construction. We assign each of the 17 phenotype tasks a fixed task code constructed from the non-nucleotide symbols `#`, `@`, and `~`, all of which belong to the native Evo2 vocabulary. Each code consists of a task-specific 10-character pattern repeated ten times, yielding the 100-character task-token sequence used throughout training and evaluation. The codes contain no phenotype labels, target values, strain identities, or other semantic information and serve solely as task identifiers. Since all symbols reuse the original Evo2 vocabulary, no vocabulary expansion or task-specific embedding parameters are introduced. Table 9 lists the exact base patterns used for all phenotype tasks.

DNA self-supervision. The auxiliary DNA objective uses a second forward pass over a corrupted copy of the same genomic input. We randomly select 5% of valid DNA positions and replace each selected nucleotide by a uniformly sampled token from $\{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}, \mathrm{N}\}$. The block-28 representation at each perturbed position is mapped through a linear $4096 \to 5$ reconstruction head to predict the original nucleotide. We use label-smoothed cross-entropy with smoothing coefficient 0.02 and normalize the reconstruction loss by the random-guess entropy $\log 5$. The resulting objective is

$$\mathcal{L}_{\mathrm{rep}} = \mathcal{L}_{\mathrm{task}} + 0.2\,\mathcal{L}_{\mathrm{SSL}}.$$

Both the Evo2 backbone and the phenotype and reconstruction heads are optimized during this stage.

Optimization details. We use AdamW with learning rates 2×10^{-5} for Evo2 and 2×10^{-4} for the prediction and reconstruction heads. The Adam coefficients are $(0.9, 0.95)$, weight decay is 0.001, and the maximum gradient norm is 1.0. Training uses linear warmup followed by cosine decay, BF16 precision, a per-GPU batch size of 16 on eight GPUs, and FSDP full sharding.

C.3 Genome-to-Language Grounding

Ordered genomic memory. For genome-to-language grounding, we retain representations over the complete genomic segment rather than using the single-position feature employed by the phenotype head. The 2,048 DNA positions are divided into 32 consecutive regions of 64 bp, and the representations within each region are mean-pooled:

$$u_k = \frac{1}{64} \sum_{i=64k}^{64(k+1)-1} H_i, \qquad k = 0, \ldots, 31.$$

This produces 32 ordered 4,096-dimensional genomic representations, preserving coarse spatial organization across the input segment.

Genomic projector and continuous-token fusion. Each genomic representation is independently mapped into the Qwen3.5-4B embedding space using a shared projector,

$$P(u) = W_2\,\mathrm{GELU}\,(W_1\,\mathrm{LN}(u) + b_1) + b_2,$$

where the two linear mappings have dimensions $4096 \to 2560$ and $2560 \to 2560$. The projected representations therefore form 32 continuous tokens with the same hidden dimension as the LLM decoder. The textual input contains exactly one pair of `<gene_evidence>` markers, and the continuous genomic tokens are inserted immediately after the opening marker. These tokens participate directly in the decoder's standard causal self-attention. No Q-Former, cross-attention module, routing layer, or additional modality-specific fusion mechanism is used.

Prompt for genome-to-language grounding

User prompt.
`<gene_evidence></gene_evidence>`
Gene evidence tokens are provided inside the tags above.
Predict the BacDive phenotype for task: {`task_display_name`}.
Use the gene evidence to determine the phenotype, then answer in exactly this format:
`<think>`
evidence-based biological reasoning
`</think>`
`{"prediction": ...}`
Do not add markdown or extra commentary.
Assistant target.
`<think>`
{protein-informed reasoning trajectory}
`</think>`
{"prediction":{gold prediction}}

Grounding supervision. The protein annotations described in Appendix B.3 are used only to construct the supervised reasoning target and are never supplied to the decoder as input. Training therefore requires the decoder to associate the continuous genomic context with the corresponding phenotype-oriented reasoning trajectory. We use the standard causal language-model cross-entropy

loss. Prompt tokens, genomic continuous tokens, and other input positions are masked from the loss; only assistant reasoning tokens, the structured phenotype prediction, and the end-of-sequence token are supervised. No separate reasoning loss, answer reweighting, auxiliary phenotype classifier, or additional genomic-contrastive objective is used.

The genomic encoder remains trainable during this grounding stage together with the projector and LLM decoder, and is frozen only before S2D-GRPO optimization. We set the learning rates of Evo2 backbone, Qwen decoder, and genomic projector as 1×10^{-6}, 1×10^{-5}, and 1×10^{-4}, respectively. We use AdamW with coefficients $(0.9, 0.95)$, weight decay 0.001, gradient clipping at 1.0, and 5% linear warmup followed by cosine decay. The maximum language-model sequence length is 1,024 tokens. Training uses BF16 precision, gradient checkpointing, a per-GPU batch size of 16 on eight GPUs, and FSDP full sharding.

C.4 S2D-GRPO Optimization

Policy input and frozen components. S2D-GRPO starts from the genome-grounded decoder. During policy optimization, the genomic representations and projector are kept fixed, and only Qwen3.5-4B is updated. Each policy input contains the genomic representation together with two frozen phenotype proposals: the Evo2–MLP proposal and the genome-grounded Qwen proposal. Both proposals are generated before reinforcement learning and remain fixed throughout optimization.

Prompt for S2D-GRPO optimization

User prompt.
`<gene_evidence></gene_evidence>`
Gene evidence tokens are provided inside the tags above.
Task: {`task`}
Current Qwen genome-only proposal: {`qwen proposal`}
Evo2 MLP proposal: {`mlp proposal`}
Independently evaluate both proposals using the gene evidence. Either proposal may be wrong. Keep a proposal only when supported by the genome; otherwise revise both.
Answer exactly:
`<think>` evidence-based reasoning `</think>`
`{"task": "...", "prediction": ...}`
Do not add markdown or extra text.

Phenotype reward and group-relative advantage. The rollout reward is defined according to the phenotype output type. For exact-prediction tasks, we use binary exact-match correctness. For multilabel tasks, we use the average of set F_1 and Jaccard similarity. For continuous phenotypes, the predicted and reference endpoints are first sorted. We normalize the error of each endpoint using the central 90% range of the corresponding training targets. Specifically, for endpoint j, the normalization scale is $s_j = P_{95}(y_j) - P_{05}(y_j)$, and the normalized endpoint error is $\tilde{e}_j = \frac{\hat{y}_j - y_j}{s_j}$. The normalized endpoint RMSE is then computed as

$$d = \sqrt{\frac{1}{2}\sum_{j=1}^{2} \tilde{e}_j^2} = \sqrt{\frac{1}{2}\sum_{j=1}^{2} \left(\frac{\hat{y}_j - y_j}{s_j}\right)^2},$$

from which the continuous reward is defined as $r = \exp\left(-\frac{d}{0.1}\right)$. Malformed or incompatible predictions receive zero reward. Following GRPO, rewards within each rollout group are normalized as $\hat{A}_i = \frac{r_i - \mathrm{mean}(r)}{\mathrm{std}(r) + 10^{-4}}$. Groups with zero reward variance are excluded from the policy-gradient term, while remaining active for KL regularization and teacher-forced supervision.

Proposal-matched genomic alternatives. The token-level genomic utility is computed using proposal-matched genomic alternatives. Each alternative must belong to the same phenotype task and preserve both frozen phenotype proposals, while differing from the factual example in phenotype label, strain identity, and genome identity. The alternatives therefore change the underlying genomic evidence without changing the proposal context available to the decoder. When the required number of valid matches is unavailable, the genomic-credit term for that example is set to zero rather than relaxing the proposal-matching constraint.

For each rollout token, we compute the genomic utility using the same proposal-conditioned contrast:

$$u_{i,t} = \log \pi_{\theta_{\text{old}}}(o_{i,t} \mid Z, A, o_{i,<t}) - \log \left[\frac{1}{K} \sum_{k=1}^{K} \pi_{\theta_{\text{old}}}(o_{i,t} \mid Z_k, A, o_{i,<t}) \right].$$

In implementation, genomic credit is assigned only to tokens inside the generated reasoning region and the phenotype prediction value. Negative utility is removed and the remaining value is clipped to $[0, 1]$. The resulting utility is detached from the counterfactual forward passes, such that the alternative genomes determine credit allocation without introducing an additional counterfactual loss.

S2D-GRPO policy optimization. For each rollout group, the phenotype reward is normalized to obtain the group-relative advantage $\hat{r}_i$. The token-level genomic utility $u_{i,t}$ then reweights this advantage to emphasize reasoning and prediction tokens whose likelihood depends more strongly on the observed genomic representation. Specifically, the language decoder is optimized with

$$\mathcal{J}(\theta) \approx \mathbb{E}\left[\frac{1}{G} \sum_{i=1}^{G} \frac{1}{|O_i|} \sum_{t=1}^{|O_i|} \frac{\pi_\theta(o_{i,t} \mid Z, A, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} \mid Z, A, o_{i,<t})} \left(\hat{r}_i(1 + w_1 u_{i,t})\right) \right] - \beta D_{\text{KL}}\left[\pi_\theta \,\|\, \pi_{\text{ref}}\right] - w_2 \mathcal{L}_{\text{CE}}, \quad (5)$$

We set $w_1 = 2$, $w_2 = 0.05$, and $\beta = 0.04$. The genomic-utility term is applied only to the reasoning and phenotype-prediction tokens identified above, while the remaining completion tokens retain the standard group-relative advantage. The cross-entropy term uses the corresponding supervised reasoning trajectory and verified phenotype target, and the KL term regularizes the policy toward the frozen genome-grounded reference model.

Regularization and teacher-forced supervision. We maintain a frozen copy of the genome-grounded decoder as the reference policy and apply token-level KL regularization over the complete sampled response. In addition, each underlying training prompt is paired with one supervised reasoning-and-answer target. This target is optimized using ordinary causal cross-entropy and is evaluated once per prompt rather than once for each sampled rollout. The complete objective is

$$\mathcal{L} = \mathcal{L}_{\text{S2D-GRPO}} + 0.04\, \mathcal{L}_{\text{KL}} + 0.05\, \mathcal{L}_{\text{CE}}.$$

No proposal-shaped reward, explicit proposal-selection loss, decision-label supervision, or additional genome-dependency loss is active.

Optimization details. Only the LLM decoder is optimized, using AdamW with learning rate 1×10^{-6}, coefficients $(0.9, 0.999)$, weight decay 0.01, and maximum gradient norm 1.0. We use a cosine learning-rate schedule with 5% warmup, BF16 precision, and gradient checkpointing. Training is distributed over eight GPUs with DeepSpeed ZeRO-2. Rollouts are sampled with temperature 1.0, top-$p = 0.95$, a maximum textual prompt length of 512 tokens, and a maximum completion length of 1,024 tokens. Generation is performed with vLLM (Kwon et al., 2023) during policy optimization, and the framework is built on TRL (von Werra et al., 2020).

C.5 Evaluation Prompts and Inference Settings

We provide the inference prompts used for genome-grounded models and frontier LLMs. For all genome-based evaluations, the same fixed 2,048-bp genomic segments from the benchmark are used across methods. Predictions are evaluated only from the required structured output fields.

G-Qwen inference prompt. G-Qwen receives 32 projected genomic tokens together with the phenotype task, without auxiliary phenotype proposals. The genomic tokens are inserted as continuous embeddings within the `<gene_evidence>` markers. The decoder generates a biological rationale followed by a structured phenotype prediction, and evaluation uses only the `prediction` field.

G-Qwen inference prompt

User prompt.

`<gene_evidence></gene_evidence>`
Gene evidence tokens are provided inside the tags above.
Predict the BacDive phenotype for task: `[task display name]`.
Use the gene evidence to determine the phenotype, then answer in exactly this format:

<think>
evidence-based biological reasoning
</think>
{"prediction": ...}
Do not add markdown or extra commentary.

Proposal-conditioned inference prompt. P-Qwen and GGBound receive the same genomic context together with two independently generated phenotype proposals from G-Qwen and Evo2-MLP. The decoder is instructed to evaluate both proposals against the genomic evidence and retain or revise them when unsupported. P-Qwen performs proposal-conditioned prediction without S2D-GRPO, whereas GGBound uses the decoder optimized by S2D-GRPO. The proposal values are fixed during inference and supplied in the same prompt.

Proposal-conditioned inference prompt

User prompt.

<gene_evidence></gene_evidence>
Gene evidence tokens are provided inside the tags above.
Task: [task]
G-Qwen proposal: [G-Qwen prediction]
Evo2-MLP proposal: [MLP prediction]
Independently evaluate both proposals using the gene evidence. Either proposal may be wrong. Keep a proposal only when supported by the genome; otherwise revise it.
Answer exactly:
<think>
evidence-based biological reasoning
</think>
{"task": "...", "prediction": ...}
Do not add markdown or extra text.

Frontier LLM evaluation. We evaluate frontier LLMs under two complementary input conditions: genomic sequence and strain name. Both settings cover the same 2,080 strain–task records and use the same task definitions, allowed labels, and structured prediction formats, without labeled demonstrations. The genomic-sequence condition provides the fixed 2,048-bp benchmark segment without strain identity, whereas the strain-name condition provides only the strain identity without genomic sequence. The two settings are evaluated independently under closed-book instructions without external retrieval.

Predictions are returned as a JSON object indexed by the supplied anonymous record identifiers. Binary and multiclass tasks require one allowed-label string, multilabel tasks require a nonempty array of unique allowed labels, and regression tasks require two finite values representing the lower and upper endpoints. Explanations, confidence scores, and additional metadata are excluded from the submitted predictions. In the templates below, [task schema] denotes the complete task definitions and allowed labels included in the evaluation prompt.

Genomic-sequence prompt. Each record contains a phenotype task and one fixed 2,048-bp genomic segment. The same segment is used for the corresponding record across all genome-based methods, and no strain identity is provided.

Frontier LLM genomic-sequence evaluation prompt

Instruction.

You are a structured microbial-trait prediction engine. Predict every item in the supplied input JSON or batch JSON.

Evidence available.

Each value contains only task and one 2,048-bp genomic DNA segment in sequence. Do not assume that a strain name is available.

Mandatory isolation rules.

- This is a closed-book test. Do not use web search, a browser, network access, external databases, RAG/retrieval, plugins, MCP/apps, or another agent.

- Do not inspect any file outside the supplied blind input or batch and this prompt.
- Do not attempt to reconstruct hidden labels from IDs, filenames, repository history, caches, logs, prior predictions, or other evaluation artifacts.
- Use only the supplied record and knowledge already contained in the model.

Exact output contract.

Return exactly one JSON object whose top-level keys are the supplied anonymous IDs. Every input ID must occur exactly once, and no additional ID is allowed. Each value is the prediction itself, without explanation or confidence fields:

- binary or multiclass: one exact allowed-label string;
- multilabel: a nonempty array of unique exact allowed-label strings;
- regression: exactly two finite JSON numbers `[lower, upper]`, with `lower <= upper`.

Do not emit Markdown, prose, citations, comments, null, NaN, or additional metadata.
Complete task definitions and allowed labels.

`[task schema]`

Strain-name prompt. Each record contains only the phenotype task and strain name. This condition allows the model to use biological and taxonomic knowledge already encoded in its parameters while prohibiting retrieval of additional information. It uses the same task definitions and structured output contract as the genomic-sequence condition.

Frontier LLM strain-name evaluation prompt

Instruction.

You are a structured microbial-trait prediction engine. Predict every item in the supplied input JSON or batch JSON.
Evidence available.

Each value contains only `task` and `strain_name`. Use biological and taxonomic knowledge already present in the model; do not retrieve additional facts.
Mandatory isolation rules.

- This is a closed-book test. Do not use web search, a browser, network access, external databases, RAG/retrieval, plugins, MCP/apps, or another agent.
- Do not inspect any file outside the supplied blind input or batch and this prompt.
- Do not attempt to reconstruct hidden labels from IDs, filenames, repository history, caches, logs, prior predictions, or other evaluation artifacts.
- Use only the supplied record and knowledge already contained in the model.

Exact output contract.

Return exactly one JSON object whose top-level keys are the supplied anonymous IDs. Every input ID must occur exactly once, and no additional ID is allowed. Each value is the prediction itself, without explanation or confidence fields:

- binary or multiclass: one exact allowed-label string;
- multilabel: a nonempty array of unique exact allowed-label strings;
- regression: exactly two finite JSON numbers `[lower, upper]`, with `lower <= upper`.

Do not emit Markdown, prose, citations, comments, null, NaN, or additional metadata.
Complete task definitions and allowed labels.

`[task schema]`

D Additional Experimental Results and Analyses

D.1 Extended Main Results

Evaluation metrics. We further evaluate the benchmark predictions using complementary metrics beyond those reported in the main text. For a predicted interval $\hat{Y}_i = [\hat{l}_i, \hat{u}_i]$ and reference interval $Y_i = [l_i, u_i]$, we report endpoint mean absolute error, $\mathrm{MAE} = \frac{1}{2N}\sum_i(|\hat{l}_i - l_i| + |\hat{u}_i - u_i|)$, interval-

width mean absolute error, $\text{W-MAE} = \frac{1}{N}\sum_i |(\hat{u}_i - \hat{l}_i) - (u_i - l_i)|$, and interval intersection-over-union, $\text{IoU} = \frac{1}{N}\sum_i |\hat{Y}_i \cap Y_i|/|\hat{Y}_i \cup Y_i|$. These metrics measure endpoint deviation, interval-width deviation, and interval overlap, respectively. For multilabel metabolic phenotypes, we report sample-level precision, $\text{Precision} = \frac{1}{N}\sum_i |\hat{Y}_i \cap Y_i|/|\hat{Y}_i|$, sample-level recall, $\text{Recall} = \frac{1}{N}\sum_i |\hat{Y}_i \cap Y_i|/|Y_i|$, and exact match, $\text{ExactMatch} = \frac{1}{N}\sum_i \mathbb{I}[\hat{Y}_i = Y_i]$. Precision and recall are computed directly over the predicted and reference label sets and therefore do not use a cutoff k. For categorical physiology and morphology, we report balanced accuracy, defined as the mean recall across classes. We first report complementary metrics for the sequence-only benchmark and then evaluate frontier LLMs conditioned on strain names.

Table 10: Additional sequence-only results on physiological boundaries.

	Temperature			pH			Salinity		
Model	MAE↓	W-MAE↓	IoU↑	MAE↓	W-MAE↓	IoU↑	MAE↓	W-MAE↓	IoU↑
GPT-5.6	9.780	15.341	0.414	1.010	1.453	0.496	3.129	5.565	0.378
Opus-5	9.849	15.584	0.412	1.014	1.358	0.523	2.669	4.428	0.391
DeepSeek	10.173	16.740	0.414	1.076	1.510	0.540	2.538	4.235	0.414
GLM-5.3	9.217	13.763	0.413	1.017	1.348	0.514	2.404	3.885	0.423
Kimi-K3	10.173	16.740	0.414	1.076	1.510	0.540	2.538	4.235	0.414
Evo2-Prob	8.026	12.053	0.350	1.007	1.351	0.527	2.663	3.886	0.354
Evo2-MLP	6.385	10.986	0.398	0.993	1.356	0.529	2.029	3.033	0.415
G-Qwen	7.205	11.740	0.389	1.072	1.571	0.493	2.084	3.372	0.407
P-Qwen	7.201	11.596	0.396	1.028	1.481	0.499	2.025	3.229	**0.428**
GGBound	**6.103**	**10.423**	**0.415**	**0.957**	**1.331**	**0.542**	**1.952**	**2.924**	0.427

Complementary metrics confirm the gains on physiological boundaries. As shown in Table 10, GGBound achieves the lowest MAE and W-MAE on all three boundary phenotypes. It also obtains the highest IoU on temperature and pH, while nearly matching the best salinity IoU (0.427 versus 0.428). The lower RMSE reported in the main text is therefore accompanied by more accurate interval endpoints and widths across physiological boundaries.

Table 11: Additional sequence-only results on physiological optimum.

	Temperature			pH			Salinity		
Model	MAE↓	W-MAE↓	IoU↑	MAE↓	W-MAE↓	IoU↑	MAE↓	W-MAE↓	IoU↑
GPT-5.6	5.193	2.407	0.056	0.809	0.552	0.078	2.144	1.814	0.090
Opus-5	5.643	3.199	0.044	0.798	0.521	0.076	2.235	1.260	0.054
DeepSeek	9.011	15.711	0.067	0.871	0.775	0.094	2.382	1.711	0.064
GLM-5.3	5.567	4.185	0.095	0.871	0.775	0.094	2.382	1.711	0.064
Kimi-K3	6.730	7.844	0.064	0.871	0.775	0.094	2.382	1.711	0.064
Evo2-Prob	5.328	1.997	0.024	0.809	0.454	0.035	2.223	0.757	0.044
Evo2-MLP	4.005	2.078	0.032	0.650	0.445	0.051	1.635	0.738	0.055
G-Qwen	4.493	2.104	**0.150**	0.913	0.368	0.079	1.891	**0.449**	0.109
P-Qwen	4.284	1.969	0.148	0.897	**0.357**	0.079	1.811	0.580	**0.110**
GGBound	**3.468**	**1.489**	0.144	**0.616**	0.408	**0.115**	**1.572**	0.692	0.073

GGBound retains strong endpoint accuracy on physiological optimum. As shown in Table 11, GGBound achieves the lowest MAE on all three optimum phenotypes, including a reduction in temperature MAE from 4.005 with Evo2-MLP to 3.468. It also achieves the lowest W-MAE on temperature and the highest IoU on pH, while the remaining interval metrics are more mixed across genome-specific baselines. The improvement on physiological optimum is therefore most consistent in endpoint localization.

Metabolic gains reflect a stronger precision–recall balance. As shown in Table 12, GGBound achieves precision/recall of 0.907/0.880 on electron acceptor, 0.855/0.950 on electron donor, and 0.848/0.894 on nitrogen source, together with the highest or tied-highest exact-match rates on these phenotypes. Carbon source shows a different pattern, where Evo2-Prob achieves the highest recall

Table 12: Additional sequence-only results on metabolic phenotypes.

	Carbon Source			Electron Acceptor			Electron Donor			Nitrogen Source		
Model	Prec.↑	Recall↑	Exact↑	Prec.↑	Recall↑	Exact↑	Prec.↑	Recall↑	Exact↑	Prec.↑	Recall↑	Exact↑
GPT-5.6	0.575	0.495	0.000	0.333	0.389	0.000	0.600	0.367	0.000	0.470	0.773	0.000
Opus-5	**0.610**	0.517	0.000	0.111	0.111	0.111	0.600	0.317	0.000	0.462	0.795	0.000
DeepSeek	0.539	0.484	0.000	0.333	0.389	0.000	0.600	0.317	0.000	0.545	0.720	0.000
GLM-5.3	0.539	0.484	0.000	0.111	0.111	0.111	0.800	0.233	0.000	0.545	0.720	0.000
Kimi-K3	0.539	0.484	0.000	0.111	0.111	0.111	0.467	0.350	0.000	0.386	0.833	0.000
Evo2-Prob	0.526	**1.000**	**0.203**	0.111	0.111	0.111	0.500	0.133	0.000	0.500	0.561	0.091
Evo2-MLP	0.528	0.963	0.187	0.574	0.509	0.222	0.633	0.533	0.200	0.636	0.720	0.182
G-Qwen	0.547	0.768	0.098	0.526	**0.889**	0.111	0.827	**0.967**	0.200	0.477	0.561	0.273
P-Qwen	0.499	0.629	0.106	0.574	0.750	0.111	0.833	0.867	0.200	0.536	0.667	0.273
GGBound	0.533	0.995	0.179	**0.907**	0.880	**0.556**	**0.855**	0.950	**0.400**	**0.848**	**0.894**	**0.455**

and exact match, while GGBound retains near-complete recall of 0.995. These results show that the F1 improvements on most metabolic phenotypes arise from maintaining high recall together with substantially stronger precision.

Table 13: Additional sequence-only balanced accuracy on categorical physiology and morphology.

Model	Oxygen Tol.	Energy Type	Cell Shape	Motility	Flagella	Spore Form.	Gram Stain
GPT-5.6	30.9	30.0	21.1	55.5	29.0	65.8	57.5
Opus-5	25.0	33.3	12.5	55.0	32.6	67.9	59.5
DeepSeek	29.0	33.3	12.5	50.0	25.0	50.0	58.0
GLM-5.3	25.0	33.3	12.5	50.0	25.0	59.7	57.5
Kimi-K3	28.5	33.3	12.5	57.5	25.0	53.5	57.5
Evo2-Prob	35.2	61.7	25.6	58.5	61.7	64.7	52.0
Evo2-MLP	66.6	43.3	47.9	76.0	77.7	91.5	96.0
G-Qwen	59.6	76.7	57.0	68.0	79.8	90.5	95.5
P-Qwen	59.8	76.7	59.4	50.5	80.2	50.0	69.0
GGBound	**67.8**	**83.3**	**62.9**	**77.5**	**88.5**	**92.5**	**96.5**

Categorical gains persist under class-balanced evaluation. As shown in Table 13, GGBound achieves the highest balanced accuracy on all seven categorical and morphological phenotypes. The gains are particularly clear on energy type (83.3% versus 76.7%), cell shape (62.9% versus 59.4%), and flagella (88.5% versus 80.2%). Thus, the improvements in standard accuracy reported in the main text remain consistent when individual classes receive equal weight.

Comparison with strain-name-conditioned frontier LLMs. We further examine how sparse genomic evidence compares with phenotype priors available from strain identity. In this setting, frontier LLMs receive the strain name without genomic sequence, whereas GGBound receives a single 2,048-bp genomic segment without strain identity. Table 14 reports the primary metrics used for the benchmark.

Sparse genomic evidence remains competitive with strain-name access. As shown in Table 14, GGBound achieves lower NRMSE than all strain-name-conditioned frontier LLMs on five of the six continuous phenotypes and higher F1 on all four metabolic phenotypes. The differences are substantial on several tasks, with temperature-boundary NRMSE decreasing from 0.336 to 0.234 and electron-acceptor F1 increasing from 0.460 to 0.873 relative to the strongest strain-name-conditioned result. Strain-name access is more competitive on categorical and morphological traits, outperforming GGBound on energy type, cell shape, spore formation, and Gram stain. These results show that strain identity provides useful phenotype priors, while sparse genomic evidence remains highly informative without access to strain identity.

Genomic evidence yields more accurate boundary endpoints and widths. As shown in Table 15, GGBound achieves the lowest W-MAE on all three physiological boundaries and the lowest MAE on temperature. On pH and salinity, its MAE remains close to the strongest strain-name-conditioned

Table 14: Per-phenotype comparison with strain-name-conditioned frontier LLMs.

Group	Phenotype	GPT-5.6	Opus-5	DeepSeek	GLM-5.3	Kimi-K3	GGBound
Phys. Boundary RMSE ↓	Temperature	11.899	11.099	12.902	11.401	11.496	**7.723**
	pH	1.310	1.249	1.364	**1.225**	1.328	1.235
	Salinity	4.233	3.524	3.781	3.707	3.799	**3.287**
Phys. Boundary NRMSE ↓	Temperature	0.361	0.336	0.391	0.345	0.348	**0.234**
	pH	0.342	0.321	0.358	**0.310**	0.350	0.315
	Salinity	0.344	0.287	0.305	0.300	0.305	**0.273**
Phys. Optimum RMSE ↓	Temperature	6.432	5.960	10.782	5.972	5.901	**4.999**
	pH	1.050	0.883	1.150	0.871	0.953	**0.831**
	Salinity	3.365	3.070	3.260	3.016	2.973	**2.573**
Phys. Optimum NRMSE ↓	Temperature	0.324	0.300	0.543	0.301	0.297	**0.252**
	pH	0.315	0.265	0.344	0.260	0.284	**0.249**
	Salinity	0.381	0.348	0.370	0.338	0.335	**0.291**
Metabolism F1 ↑	Carbon Source	0.426	0.532	0.433	0.433	0.437	**0.631**
	Electron Acceptor	0.411	0.448	0.460	0.385	0.285	**0.873**
	Electron Donor	0.547	0.614	0.500	0.628	0.514	**0.895**
	Nitrogen Source	0.534	0.519	0.567	0.542	0.475	**0.827**
Categorical Phys. Acc. ↑	Oxygen Tolerance	47.7	59.6	51.3	59.1	57.5	**67.9**
	Energy Type	88.5	88.5	**92.3**	**92.3**	88.5	84.6
Morphology Acc. ↑	Cell Shape	42.7	**55.3**	38.8	49.5	44.7	54.4
	Motility	61.0	75.5	62.0	76.5	73.0	**77.5**
	Flagella	69.0	83.9	46.0	58.6	50.6	**86.2**
	Spore Formation	80.4	**95.5**	86.9	73.9	92.0	92.5
	Gram Stain	82.5	**98.0**	88.5	93.0	95.0	96.5

Table 15: Comparison with strain-name-conditioned frontier LLMs on physiological boundaries.

	Temperature			pH			Salinity		
Model	MAE↓	W-MAE↓	IoU↑	MAE↓	W-MAE↓	IoU↑	MAE↓	W-MAE↓	IoU↑
GPT-5.6	9.312	15.688	0.426	0.992	1.470	0.504	2.358	4.106	0.430
Opus-5	8.685	15.092	**0.434**	**0.944**	1.351	**0.550**	**1.902**	3.286	**0.476**
DeepSeek	10.173	17.006	0.412	1.063	1.523	0.545	2.245	4.059	0.440
GLM-5.3	8.702	14.561	0.425	0.948	1.381	0.536	2.026	3.276	0.437
Kimi-K3	9.046	15.225	0.431	1.040	1.503	0.549	2.441	4.572	0.431
GGBound	**6.103**	**10.423**	0.415	0.957	**1.331**	0.542	1.952	**2.924**	0.427

results, while Opus-5 obtains higher interval IoU. Together with the NRMSE results in Table 14, the advantage of genomic evidence is most consistent in endpoint and interval-width estimation.

Table 16: Comparison with strain-name-conditioned frontier LLMs on physiological optimum.

	Temperature			pH			Salinity		
Model	MAE↓	W-MAE↓	IoU↑	MAE↓	W-MAE↓	IoU↑	MAE↓	W-MAE↓	IoU↑
GPT-5.6	4.519	2.763	0.053	0.745	0.506	0.090	2.039	1.399	0.084
Opus-5	3.950	3.648	0.080	0.671	0.518	0.085	1.778	1.078	0.112
DeepSeek	8.919	15.600	0.067	0.867	0.788	0.094	2.024	2.100	0.080
GLM-5.3	4.563	5.156	0.110	0.673	0.788	**0.134**	1.943	2.062	0.096
Kimi-K3	4.437	5.667	0.086	0.760	0.788	0.100	1.773	2.060	**0.113**
GGBound	**3.468**	**1.489**	**0.144**	**0.616**	**0.408**	0.115	**1.572**	**0.692**	0.073

The advantage is more consistent on physiological optimum. As shown in Table 16, GGBound achieves the lowest MAE and W-MAE on all three optimum phenotypes. It also obtains the highest

temperature IoU, while strain-name-conditioned models achieve higher IoU on pH and salinity. Together with Table 14, these results show that sparse genomic evidence provides consistently stronger endpoint and interval-width estimates, although interval overlap remains task dependent.

Table 17: Comparison with strain-name-conditioned frontier LLMs on metabolic phenotypes.

	Carbon Source			Electron Acceptor			Electron Donor			Nitrogen Source		
Model	Prec.↑	Recall↑	Exact↑	Prec.↑	Recall↑	Exact↑	Prec.↑	Recall↑	Exact↑	Prec.↑	Recall↑	Exact↑
GPT-5.6	0.539	0.470	0.000	0.556	0.417	0.000	0.800	0.467	0.200	0.470	0.788	0.000
Opus-5	**0.594**	0.612	0.016	0.500	0.472	0.000	0.733	0.550	0.200	0.455	0.788	0.000
DeepSeek	0.542	0.483	0.000	0.611	0.481	0.000	0.700	0.417	0.200	0.545	0.720	0.000
GLM-5.3	0.549	0.478	0.033	0.667	0.296	0.111	0.833	0.550	0.200	0.530	0.720	0.091
Kimi-K3	0.545	0.490	0.000	0.444	0.231	0.111	0.633	0.450	0.200	0.386	0.833	0.000
GGBound	0.533	**0.995**	**0.179**	**0.907**	**0.880**	**0.556**	**0.855**	**0.950**	**0.400**	**0.848**	**0.894**	**0.455**

Genome-grounded predictions show a clear advantage on metabolism. As shown in Table 17, GGBound achieves the highest recall and exact-match rate on all four metabolic phenotypes and the highest precision on three of the four. The difference is particularly pronounced on electron acceptor, where precision and recall reach 0.907 and 0.880, compared with at most 0.667 and 0.481 among strain-name-conditioned frontier LLMs. Exact match also increases from 0.111 to 0.556. Together with the F1 results in Table 14, these results show that the metabolic gains arise from recovering more complete label sets while maintaining high precision.

Table 18: Comparison with strain-name-conditioned frontier LLMs using balanced accuracy.

Model	Oxygen Tol.	Energy Type	Cell Shape	Motility	Flagella	Spore Form.	Gram Stain
GPT-5.6	46.3	90.0	33.6	61.0	57.0	80.3	82.5
Opus-5	57.9	85.0	48.2	75.5	65.0	**95.5**	**98.0**
DeepSeek	50.0	**91.7**	24.5	62.0	51.6	86.9	88.5
GLM-5.3	57.4	86.7	43.6	76.5	52.2	73.7	93.0
Kimi-K3	56.0	80.0	36.5	73.0	56.5	91.9	95.0
GGBound	**67.8**	83.3	**62.9**	**77.5**	**88.5**	92.5	96.5

The categorical comparison remains task-dependent under balanced accuracy. As shown in Table 18, GGBound achieves the highest balanced accuracy on oxygen tolerance, cell shape, motility, and flagella, while strain-name-conditioned frontier LLMs remain stronger on energy type, spore formation, and Gram stain. The difference is largest on flagella, where GGBound reaches 88.5% compared with 65.0% for the strongest strain-name-conditioned model. This pattern closely follows the standard-accuracy comparison in Table 14, showing that the observed differences are not explained by class imbalance alone.

D.2 Robustness to Genomic Crop Selection

As shown in Table 8, Electron Acceptor, Electron Donor, and Nitrogen Source contain relatively few independent held-out strains, reflecting the limited availability of reliable strain-level annotations for these phenotypes. To examine whether the reported performance is sensitive to genomic crop selection, we additionally sample ten fixed 2,048-bp genomic segments from each held-out strain for these three tasks. For each crop index, all methods are evaluated on the same genomic segments, yielding ten crop-specific F1 scores over the same set of held-out strains. We report the mean F1 across these ten evaluations, without treating the additional crops as independent biological samples.

As shown in Table 19, GGBound achieves the highest mean F1 across all three phenotypes. It reaches 0.824 on Electron Acceptor and 0.803 on Nitrogen Source, exceeding the strongest baseline by 0.208 and 0.221, respectively. These consistent gains across different genomic crops indicate that the improvements are not specific to the segments used in the main benchmark.

Table 19: Mean F1 over ten additional genomic crop evaluations for three low-resource metabolic phenotypes.

Task	GGBound	Evo2-MLP	G-Qwen	P-Qwen	Evo2-Prob
Electron Acceptor	**0.824**	0.581	0.574	0.616	0.111
Electron Donor	**0.811**	0.533	0.655	0.755	0.208
Nitrogen Source	**0.803**	0.582	0.509	0.560	0.477

D.3 Analysis of Task-Token Conditioning and DNA Self-Supervision

Task-token conditioning reshapes the genomic representation. We further examine the representation shift visualized in Figure 3 under a controlled comparison. The setting without task tokens uses the +SSL specialization variant, which combines phenotype supervision and DNA self-supervision but does not introduce task tokens. The task-conditioned setting uses the full specialization encoder of GGBound, which differs by additionally prepending the 100 task tokens used throughout GGBound. For each 2,048-bp genomic segment, we collect the contextualized Evo2 hidden states at the 2,048 DNA positions and mean-pool them into a single representation for t-SNE visualization; the task-token positions themselves are excluded from pooling. The clearer phenotype-dependent separation in Figure 3 therefore reflects how task conditioning changes the encoded DNA representations, rather than directly including the task-token representations in the visualization.

Table 20: Per-phenotype ablation of task-token conditioning and DNA self-supervision.

Group	**Phenotype**	**Adapt**	**+Token**	**+SSL**	**Full**
Phys. Boundary RMSE ↓	Temperature	8.120	8.507	8.090	**7.920**
	pH	1.265	1.264	**1.236**	1.274
	Salinity	3.351	3.555	**3.287**	3.340
Phys. Boundary NRMSE ↓	Temperature	0.246	0.258	0.245	**0.240**
	pH	0.324	0.323	**0.316**	0.324
	Salinity	0.281	0.287	**0.273**	0.276
Phys. Optimum RMSE ↓	Temperature	5.376	5.692	**5.292**	5.448
	pH	0.874	0.908	**0.843**	0.866
	Salinity	2.662	2.788	**2.612**	2.657
Phys. Optimum NRMSE ↓	Temperature	0.271	0.287	**0.267**	0.274
	pH	0.262	0.272	**0.253**	0.259
	Salinity	0.302	0.316	**0.294**	0.301
Metabolism F1 ↑	Carbon Source	0.615	0.625	**0.628**	0.623
	Electron Acceptor	0.321	0.295	0.387	**0.525**
	Electron Donor	0.536	**0.638**	0.518	0.543
	Nitrogen Source	0.568	0.614	**0.667**	0.602
Categorical Phys. Acc. ↑	Oxygen Tolerance	63.9	64.2	33.7	**66.8**
	Energy Type	61.5	57.7	0.0	**76.9**
Morphology Acc. ↑	Cell Shape	42.5	35.0	19.4	**45.6**
	Motility	71.0	73.5	20.0	**76.0**
	Flagella	74.6	78.2	18.4	**81.6**
	Spore Formation	87.7	91.0	40.2	**91.5**
	Gram Stain	94.0	95.0	31.5	**96.0**

Task tokens and DNA self-supervision provide complementary specialization signals. As shown in Table 20, task-token conditioning primarily benefits discrete phenotypes, improving electron-donor F1 from 0.536 to 0.638 and flagella accuracy from 74.6% to 78.2%, while providing little benefit on continuous phenotypes. In contrast, DNA self-supervision consistently improves all six continuous phenotypes under both RMSE and NRMSE, and also raises electron-acceptor F1 from 0.321 to 0.387, but substantially degrades categorical and morphological performance when used alone. Combining the two signals restores and further strengthens discrete prediction, increasing energy-type accuracy to 76.9% and improving all five morphological phenotypes over task-token conditioning alone,

while retaining competitive performance on continuous traits. These results indicate complementary roles: task conditioning promotes phenotype-specific discrimination, whereas DNA self-supervision preserves quantitative genomic information during specialization.

Table 21: Per-phenotype sensitivity to task-token length under DNA self-supervision.

Group	Phenotype	50	100	200
Phys. Boundary RMSE ↓	Temperature	**7.845**	7.920	8.060
	pH	**1.269**	1.274	1.277
	Salinity	3.256	3.340	**3.242**
Phys. Boundary NRMSE ↓	Temperature	**0.238**	0.240	0.244
	pH	0.325	**0.324**	0.325
	Salinity	**0.268**	0.276	0.270
Phys. Optimum RMSE ↓	Temperature	**5.372**	5.448	5.403
	pH	**0.851**	0.866	0.858
	Salinity	**2.603**	2.657	2.666
Phys. Optimum NRMSE ↓	Temperature	**0.271**	0.274	0.272
	pH	**0.255**	0.259	0.257
	Salinity	**0.295**	0.301	0.301
Metabolism F1 ↑	Carbon Source	**0.626**	0.623	0.623
	Electron Acceptor	0.448	**0.525**	0.359
	Electron Donor	**0.577**	0.543	0.494
	Nitrogen Source	**0.633**	0.602	0.614
Categorical Phys. Acc. ↑	Oxygen Tolerance	65.3	**66.8**	65.8
	Energy Type	73.1	**76.9**	73.1
Morphology Acc. ↑	Cell Shape	42.7	**45.6**	44.7
	Motility	73.5	**76.0**	72.5
	Flagella	**83.9**	81.6	**83.9**
	Spore Formation	92.0	91.5	**92.5**
	Gram Stain	**96.0**	**96.0**	95.5

Performance remains stable across task-token lengths. As shown in Table 21, no single task-token length consistently dominates across phenotypes. For the continuous tasks, the differences in both RMSE and NRMSE are generally small, with 50 tokens performing slightly better on several phenotypes. In contrast, 100 tokens achieves the strongest accuracy on both categorical-physiology tasks and on cell shape and motility, while performance on the remaining morphological tasks is comparable across lengths. Increasing the prefix to 200 tokens provides no consistent improvement. We therefore use 100 task tokens as the fixed configuration of GGBound across all phenotypes.

D.4 Genomic Specialization for Downstream Grounding

Controlled comparison of genomic specialization. We provide the per-phenotype results underlying the group-level comparison in Figure 4. To isolate the effect of genomic specialization on subsequent genome-to-language grounding, all variants use the same grounding architecture and training procedure, differing only in the genomic encoder configuration. *Frozen* uses the pretrained Evo2 encoder without phenotype-specific adaptation and keeps it frozen during grounding; *Adapt* applies phenotype supervision without task tokens or DNA self-supervision; *+Token* and *+DNA SSL* add task-token conditioning and DNA self-supervision individually; and *Full* combines both components. The *Full* configuration corresponds to the specialized genomic encoder used by GGBound before proposal-conditioned refinement.

Phenotype-specific specialization substantially improves downstream grounding. As shown in Table 22, the Full configuration improves 14 of the 17 phenotypes over phenotype-only adaptation and ties on energy type. The gains are particularly pronounced on discrete phenotypes: electron-acceptor F1 increases from 0.396 to 0.645, while cell-shape and flagella accuracy increase from 12.6% to 42.7% and from 39.1% to 81.6%, respectively. These per-phenotype results support the group-level improvements in Figure 4, showing that genomic specialization remains beneficial after the representations are grounded in the language decoder.

Table 22: Per-phenotype effect of genomic specialization on downstream grounding.

Group	Phenotype	Frozen	Adapt	+Token	+DNA SSL	Full
Phys. Boundary RMSE ↓	Temperature	11.755	10.761	9.639	**9.403**	9.450
	pH	1.404	**1.379**	1.449	1.414	1.421
	Salinity	5.025	4.507	4.174	**4.072**	4.092
Phys. Boundary NRMSE ↓	Temperature	0.356	0.326	0.292	**0.285**	0.286
	pH	0.360	**0.356**	0.374	0.363	0.363
	Salinity	0.407	0.361	0.331	**0.325**	0.341
Phys. Optimum RMSE ↓	Temperature	8.243	7.568	6.720	**6.534**	6.588
	pH	1.164	1.226	1.181	**1.151**	1.158
	Salinity	3.987	3.591	3.311	**3.198**	3.247
Phys. Optimum NRMSE ↓	Temperature	0.415	0.381	0.338	**0.329**	0.332
	pH	0.351	0.368	0.355	**0.345**	0.347
	Salinity	0.448	0.404	0.372	**0.361**	0.365
Metabolism F1 ↑	Carbon Source	0.577	0.572	0.578	0.577	**0.578**
	Electron Acceptor	0.489	0.396	0.642	0.634	**0.645**
	Electron Donor	0.317	0.317	**0.878**	0.837	0.824
	Nitrogen Source	0.223	**0.604**	0.487	0.370	0.488
Categorical Phys. Acc. ↑	Oxygen Tolerance	32.6	32.6	59.1	52.8	**59.6**
	Energy Type	**80.8**	**80.8**	**80.8**	76.9	**80.8**
Morphology Acc. ↑	Cell Shape	6.8	12.6	**42.7**	38.8	**42.7**
	Motility	48.0	55.0	67.5	62.0	**68.0**
	Flagella	42.5	39.1	**81.6**	73.6	**81.6**
	Spore Formation	41.2	56.8	89.9	89.4	**90.5**
	Gram Stain	62.0	75.5	95.0	**95.5**	**95.5**

Task conditioning and DNA self-supervision contribute complementary downstream gains. Table 22 further shows different strengths of the two specialization components. DNA self-supervision achieves the lowest RMSE and NRMSE on five of the six continuous phenotypes, whereas task-token conditioning produces larger gains on several metabolic, categorical, and morphological phenotypes. Combining both components retains competitive continuous prediction while achieving the best or tied-best performance on 10 of the 11 discrete phenotypes. This complementarity explains why the full specialization provides the strongest overall performance across phenotype groups.

D.5 Genomic Soft-Token Sensitivity

Genomic soft-token construction. We further examine the sensitivity of genome-to-language grounding to the number of genomic soft tokens. For each 2,048-bp genomic segment, the contextualized Evo2 representations over DNA positions are uniformly partitioned into M contiguous regions and mean-pooled within each region, yielding M ordered genomic representations before projection into the Qwen embedding space. We vary $M \in \{16, 32, 64, 128\}$ while keeping the remaining grounding architecture and optimization settings fixed. The 32-token setting corresponds to the default grounding configuration used by GGBound.

Grounding remains stable across genomic soft-token resolutions. As shown in Table 23, no single soft-token length consistently dominates across the 17 phenotypes. For example, 32 tokens achieve the lowest RMSE and NRMSE on temperature and pH boundaries and the highest electron-donor F1, whereas 16, 64, and 128 tokens perform better on different physiological, metabolic, and morphological phenotypes. Increasing the number of soft tokens therefore does not produce a monotonic improvement in downstream prediction.

Table 24: Genomic token sensitivity.

Group	Metric	16	32	64	128
Boundary	NRM.↓	0.34	0.33	0.34	0.33
Optimum	NRM.↓	0.35	0.35	0.35	0.37
Metabolic	F1 ↑	0.63	0.63	0.60	0.63
Categorical	Acc.↑	70.2	70.2	69.4	69.4
Morphology	Acc.↑	75.6	75.7	75.8	75.4

Table 23: Per-phenotype sensitivity to the number of genomic soft tokens.

Group	Phenotype	16	32	64	128
Phys. Boundary RMSE ↓	Temperature	9.661	**9.450**	9.566	9.545
	pH	1.480	**1.421**	1.605	1.486
	Salinity	4.242	4.092	4.218	**3.942**
Phys. Boundary NRMSE ↓	Temperature	0.293	**0.286**	0.290	0.289
	pH	0.377	**0.363**	0.401	0.379
	Salinity	0.341	0.341	0.339	**0.316**
Phys. Optimum RMSE ↓	Temperature	**6.211**	6.588	6.402	7.305
	pH	1.209	1.158	**1.140**	1.210
	Salinity	3.291	**3.247**	3.343	3.351
Phys. Optimum NRMSE ↓	Temperature	**0.313**	0.332	0.322	0.368
	pH	0.363	0.347	**0.343**	0.364
	Salinity	0.369	**0.365**	0.374	0.376
Metabolism F1 ↑	Carbon Source	0.571	0.578	0.560	**0.581**
	Electron Acceptor	0.668	0.645	0.634	**0.683**
	Electron Donor	0.785	**0.824**	0.705	0.817
	Nitrogen Source	**0.515**	0.488	**0.515**	0.439
Categorical Phys. Acc. ↑	Oxygen Tolerance	**59.6**	**59.6**	58.0	58.0
	Energy Type	**80.8**	**80.8**	**80.8**	**80.8**
Morphology Acc. ↑	Cell Shape	39.8	42.7	**43.7**	36.9
	Motility	70.5	68.0	67.0	**72.5**
	Flagella	82.8	81.6	**83.9**	80.5
	Spore Formation	89.4	90.5	88.9	**91.0**
	Gram Stain	95.5	95.5	95.5	**96.0**

The 32-token configuration provides a robust default across phenotype groups. Despite task-level variation, the group-level differences remain small: with 32 tokens, average NRMSE is 0.330 on physiological boundaries and 0.348 on physiological optimum, metabolic F1 is 0.634, categorical accuracy is 70.2%, and morphology accuracy is 75.7%. These results are competitive with the strongest setting in each group without requiring a finer genomic partition. We therefore use 32 genomic soft tokens as the fixed grounding configuration throughout GGBound.

D.6 S2D-GRPO Component Ablation

Controlled component ablation. We provide the per-phenotype results underlying the group-level ablation in Figure 5. All variants start from the same genome-grounded decoder and use identical genomic representations and frozen phenotype proposals, differing only in the policy objective. *GRPO* removes both genomic utility and CE anchoring, *+Utility* adds genomic utility without CE anchoring, and *+CE* adds CE anchoring without genomic utility. GGBound combines both components in the complete S2D-GRPO objective.

As shown in Table 25, the full objective achieves the best or tied-best performance across all 17 phenotypes. The two components provide task-dependent improvements when used independently. Genomic utility yields notable gains on several physiological, categorical, and morphological phenotypes, while CE anchoring is particularly beneficial on several metabolism tasks. Their combination produces the most consistent improvements, including substantial gains on electron acceptor, electron donor, nitrogen source, and cell shape.

D.7 Proposal Conflict Resolution

We further analyze proposal-conditioned predictions on categorical tasks, where proposal correctness can be determined by exact match. We partition examples according to whether both auxiliary proposals are correct or exactly one proposal is correct. The single-correct subset is further separated according to which proposal matches the target.

Table 25: Per-phenotype ablation of genomic utility and CE anchoring in S2D-GRPO.

Group	Phenotype	GRPO	+Utility	+CE	GGBound
Phys. Boundary RMSE ↓	Temperature	7.975	7.860	7.868	**7.723**
	pH	1.266	1.259	1.257	**1.235**
	Salinity	3.326	3.381	3.345	**3.287**
Phys. Boundary NRMSE ↓	Temperature	0.242	0.238	0.238	**0.234**
	pH	0.322	0.321	0.320	**0.315**
	Salinity	0.275	0.279	0.276	**0.273**
Phys. Optimum RMSE ↓	Temperature	5.425	5.099	5.247	**4.999**
	pH	0.840	0.839	0.869	**0.831**
	Salinity	2.654	2.653	2.664	**2.570**
Phys. Optimum NRMSE ↓	Temperature	0.273	0.257	0.264	**0.252**
	pH	0.251	0.251	0.261	**0.249**
	Salinity	0.301	0.300	0.303	**0.291**
Metabolism F1 ↑	Carbon Source	0.619	0.626	0.624	**0.631**
	Electron Acceptor	0.521	0.528	0.540	**0.873**
	Electron Donor	0.807	0.822	0.807	**0.895**
	Nitrogen Source	0.564	0.519	0.609	**0.827**
Categorical Phys. Acc. ↑	Oxygen Tolerance	65.8	65.8	66.3	**67.9**
	Energy Type	76.9	**84.6**	80.8	**84.6**
Morphology Acc. ↑	Cell Shape	40.8	46.6	41.7	**54.4**
	Motility	76.5	76.0	**77.5**	**77.5**
	Flagella	85.1	85.1	85.1	**86.2**
	Spore Formation	90.5	91.5	91.0	**92.5**
	Gram Stain	96.0	96.0	95.5	**96.5**

As shown in Table 26, standard GRPO exhibits substantially different behavior across the two single-correct subsets, achieving 8.0% accuracy when only G-Qwen is correct and 81.6% when only Evo2-MLP is correct. Genomic utility increases the latter to 92.1%, while CE anchoring substantially improves the former to 40.0%. Combining both signals yields the strongest performance in both directions, reaching 52.0% and 94.7%, respectively. These results further show that the two components provide complementary signals for resolving conflicting auxiliary predictions.

Table 26: Proposal resolution breakdown.

Method	Both Correct	Qwen Only	MLP Only	Single Correct
P-Qwen	73.2	60.0	39.5	44.6
GRPO	99.9	8.0	81.6	63.4
+ Utility	99.9	12.0	92.1	72.3
+ CE	98.9	40.0	77.6	68.3
GGBound	**100.0**	**52.0**	**94.7**	**84.2**

D.8 Analysis by Taxonomic Overlap

Although the benchmark is strain-disjoint from all model-development data, related organisms may still occur across partitions. We therefore perform a post-hoc analysis based on taxonomic overlap with the union of model-development strains. Species identity is determined using the NCBI species TaxID associated with each local assembly, while genus identity is obtained from the matched BacDive taxonomy. We cross-check the taxonomy mapping using the corresponding assembly, BacDive ID, and culture-collection strain identifiers to ensure consistent strain alignment.

Each benchmark record is then assigned to one of three mutually exclusive groups: *Seen species*, where another strain from the same species occurs during model development; *Genus only*, where the species is unseen but the genus has been observed; and *Unseen genus*, where neither the species nor genus occurs during model development. Among the 2,080 benchmark records, 361 belong

Table 27: Performance across different levels of taxonomic overlap with model-development strains. The records with unresolved taxonomy are excluded from this analysis.

Phenotype	Overlap	n	Evo2-MLP	Evo2-Prob	G-Qwen	P-Qwen	GGBound
Temperature range RMSE (↓)	Seen species	18	9.998	11.541	12.973	12.651	**9.949**
	Genus only	128	7.479	9.219	8.801	8.571	**7.315**
	Unseen genus	21	8.886	10.495	8.742	9.237	**8.354**
Temperature optimum RMSE (↓)	Seen species	22	5.113	6.205	5.205	4.959	**4.421**
	Genus only	97	5.407	8.142	6.414	6.205	**4.977**
	Unseen genus	14	6.501	10.780	9.508	9.235	**6.183**
Carbon source F1 (↑)	Seen species	22	0.544	0.564	0.470	0.325	**0.565**
	Genus only	87	0.641	0.643	0.602	0.494	**0.647**
	Unseen genus	13	0.664	0.659	0.647	0.506	**0.669**
Oxygen tolerance Acc.(%) (↑)	Seen species	43	**72.1**	44.2	60.5	67.4	**72.1**
	Genus only	123	62.6	33.3	60.2	56.9	**64.2**
	Unseen genus	24	**75.0**	29.2	50.0	58.3	**75.0**
Motility Acc.(%) (↑)	Seen species	34	**76.5**	73.5	70.6	61.8	**76.5**
	Genus only	137	76.6	56.9	68.6	50.4	**77.4**
	Unseen genus	24	75.0	41.7	62.5	41.7	**83.3**

to seen species, 1,418 to genus-only overlap, and 255 to unseen genera0. Taxonomy could not be unambiguously resolved under this mapping procedure for the remaining 46 records, which are excluded from the stratified analysis.

Table 27 reports representative phenotypes with sufficient samples across the three overlap levels. GGBound remains best or tied-best across all reported subsets, and its improvements are not concentrated on benchmark strains whose species were observed during model development. For example, on temperature optimum, GGBound reduces RMSE from 6.501 to 6.183 relative to Evo2-MLP for unseen genera, while on motility it improves accuracy from 75.0% to 83.3%. Similar relative gains are retained for temperature range and carbon source. These results provide evidence that the improvements of GGBound persist under reduced taxonomic overlap, although this post-hoc analysis does not replace a benchmark explicitly constructed for taxonomic extrapolation.

E LIMITATIONS

Limited phenotype coverage. Our benchmark is constrained by the availability of experimentally characterized strain-level phenotypes that can be reliably matched to genome assemblies. Although the benchmark spans 17 life-boundary phenotypes, annotation coverage remains uneven across tasks, with several metabolic phenotypes containing relatively few held-out strains. These low-resource tasks are retained to preserve broad phenotype coverage. Expanding experimentally characterized strain collections, particularly for underrepresented metabolic phenotypes, remains an important direction for future work.

Taxonomic separation. Our benchmark evaluates phenotype prediction for previously unseen strains and therefore uses strain-disjoint partitions, while allowing related organisms to occur across model-development and evaluation data. This setting reflects practical scenarios in which newly sequenced strains may belong to previously observed taxonomic groups. To characterize the effect of such overlap, we additionally stratify benchmark records into seen-species, genus-only, and unseen-genus subsets. Across representative phenotypes, GGBound retains its relative improvements across these overlap levels, including on unseen-genus subsets (Appendix D.8). Nevertheless, the benchmark is not explicitly constructed to enforce species- or genus-level separation, and evaluation under stricter taxonomic extrapolation remains an important direction for future work.