BIOSTITCH: GATED CROSS-ATTENTION FOR BIOLOGICAL–LANGUAGE MODELING

Anonymous authors
Paper under double-blind review

ABSTRACT

Integrating biological foundation models with large language models (LLMs) is crucial for a broad range of biological tasks, from sequence understanding to text-guided biological generation. However, existing approaches typically connect these models through input-space fusion, without fully exploiting the rich representations distributed across their internal layers. To understand how pretrained representations can be better connected beyond input-space fusion, we systematically probe the layer-wise interaction between biological models and LLMs. We find that, when transferring DNA representations into LLMs, intermediate layers provide more effective features on average than conventionally used final-layer representations. Guided by this finding, we propose BioStitch, a versatile framework that stitches biological foundation models and LLMs through multi-level gated cross-attention, allowing hidden states in one model to selectively query internal representations from the other. On biological-sequence-to-text tasks, BioStitch delivers strong improvements on DNA variant-effect benchmarks, protein question answering, and zero-shot single-cell annotation, without multimodal pre-training. On text-conditioned protein generation, BioStitch substantially improves text–sequence alignment over conventional input-space conditioning. BioStitch is also highly efficient in the measured long-DNA setting, reducing peak training memory by up to $4.7\times$ and LLM prompt KV-cache usage by $71.7\times$. All these results indicate that BioStitch is an effective and efficient framework for connecting specialized biological and language representations.

1 INTRODUCTION

Biological tasks increasingly sit at the interface between biological sequences and natural language. In one direction, sequence-to-text problems such as DNA/protein question answering require language models to reason over specialized biological signals (Fallahpour et al., 2025; Jararweh et al., 2025). In the other, text-conditioned design asks biological sequence models to generate scientific entities such as proteins or mRNA that satisfy natural-language specifications (Fang et al., 2024; Dai et al., 2024; Liu et al., 2025). These settings bring together two complementary model families: biological foundation models, which learn domain-specific representations over biological entities (Dalla-Torre et al., 2025; Brixi et al., 2026; Lin et al., 2023; Nijkamp et al., 2023), and large language models (LLMs), which provide a flexible interface for natural-language instructions and reasoning. Effectively connecting these two representation spaces is therefore central to both biological understanding and language-guided biological design.

A common strategy is input-space fusion: representations from a source model are projected or resampled into tokens that are presented to the receiver model at its input (de Almeida et al., 2025; Fallahpour et al., 2025; Jararweh et al., 2025; Li et al., 2023). This design is simple, but interaction between the two models is introduced only at the input boundary and typically relies on a single representation from the source model, despite the different information encoded across its layers. Internal cross-attention provides a more direct alternative, allowing the receiver to query source representations from selected internal layers (Alayrac et al., 2022; Ye et al., 2025).

We first examine how the choice of source and receiver layers affects this interaction on DNA-to-text tasks. We find that source extraction depth matters substantially more than injection depth, with intermediate biological representations providing the strongest overall features, whereas no single

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.

injection depth is consistently best. Based on these observations, we propose BioStitch, which connects pretrained biological and language models through multi-level gated cross-attention. A selected intermediate source representation provides keys and values, while hidden states of the receiver query it at shallow, middle, and deep layers rather than at a single injection point. The same interface supports both directions: biological representations can condition an LLM for sequence-to-text tasks, while language representations can condition a protein decoder for text-to-protein generation.

We evaluate BioStitch across four settings spanning biological-sequence understanding and text-conditioned generation. On strict held-out DNA variant-effect benchmarks, BioStitch consistently outperforms BioReason, a matched input-space fusion baseline, with an average F1 gain of 15.3 points across four model–dataset configurations. Beyond DNA, the same interface transfers to protein question answering and zero-shot single-cell annotation, where it compares favorably with specialized published systems despite substantially smaller or no prior cross-modal pre-training. In the reverse text-to-protein direction, BioStitch improves text–sequence alignment over matched input-prefix conditioning with frozen backbones, showing that the same interface formulation also applies to biological generation. BioStitch is also substantially more efficient for long biological inputs, reducing peak training memory by $4.7\times$ and LLM prompt KV-cache usage by $71.7\times$ in the measured long-DNA setting.

Our contributions can be summarized as follows:

- **We identify source depth as an important design choice for biological–language fusion.** In DNA-to-text layer-pair probes, source extraction depth explains substantially more performance variation than injection depth. Broader ablations further show that intermediate biological representations provide the strongest aggregate performance across the tested DNA configurations.
- **We introduce BioStitch, a unified framework connecting biological and language models.** BioStitch uses multi-level gated cross-attention to expose internal source representations directly to a receiver model. By exchanging the source and receiver roles, the same interface supports both biological-sequence understanding and text-guided protein generation.
- **We demonstrate the effectiveness and efficiency of this interface across multiple settings.** BioStitch consistently improves strict held-out DNA variant-effect prediction, transfers to protein question answering and single-cell annotation without prior cross-modal pre-training, and strengthens text–sequence alignment in protein generation. It also substantially reduces training memory and prompt-state cost for long biological inputs.

2 Related Work

Biological foundation models. Biological foundation models learn representations of DNA, protein, and transcriptome data. DNA models range from masked Transformer encoders over k-mers or subwords (Ji et al., 2021; Zhou et al., 2024; Dalla-Torre et al., 2025) to long-context architectures built on Hyena operators, structured state spaces, and StripedHyena (Poli et al., 2023; Gu & Dao, 2024; Nguyen et al., 2023; Schiff et al., 2024; Nguyen et al., 2024; Brixi et al., 2026). For proteins, masked encoders such as ESM2 return residue-level representations that transfer to structure and function prediction (Lin et al., 2023), whereas autoregressive models such as ProGen2 and NatureLM model sequence likelihood directly and generate new sequences (Nijkamp et al., 2023; Xia et al., 2025). For transcriptomes, models such as Geneformer, scBERT, and scGPT treat a cell's rank-ordered expression profile as a token sequence and return a cell-level representation (Theodoris et al., 2023; Yang et al., 2022; Cui et al., 2024). How to combine these representations with language models, however, remains an open question.

Connecting modalities to language models. Vision–language systems realize input-space fusion either by projecting modality features directly into the input sequence (Liu et al., 2023; Zhu et al., 2024) or by first compressing them into a small set of query tokens (Li et al., 2023; Dai et al., 2023; Bai et al., 2023; Zhang et al., 2025); Flamingo and mPLUG-Owl3 instead place cross-attention inside the language model (Alayrac et al., 2022; Ye et al., 2025). Biology–language systems follow the same taxonomy: ChatNT and BioReason project genomic features into the prompt (de Almeida et al., 2025; Fallahpour et al., 2025), Protein2Text resamples protein features (Jararweh et al., 2025), ProtT3 uses

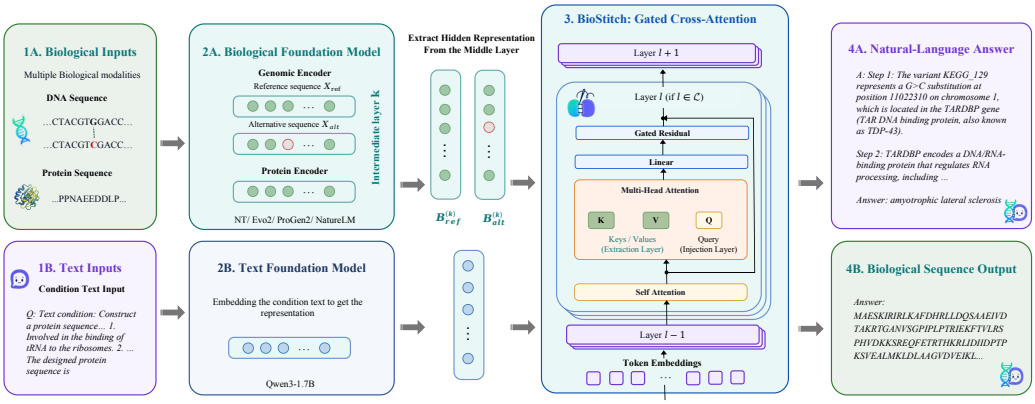


Figure 1: **Overview of BioStitch.** BioStitch connects a source model and a receiver through a gated cross-attention interface. The connector uses the representation from a selected source layer as keys and values, while hidden states of the receiver provide queries at multiple depths; the attended features are normalized, projected, and injected through layer-specific gates. Top row (1A, 2A, 4A), biological sequence to text: the biological model acts as the source and the LLM as the receiver, which generates a natural-language output. Bottom row (1B, 2B, 4B), text to biological sequence: the source and receiver roles are exchanged, with the language model providing the conditioning representations and the biological sequence model generating the output sequence.

query tokens (Liu et al., 2024b), and BioMedGPT connects biological inputs to an LLM (Luo et al., 2026). In the generation direction, text-conditioned protein systems are built as instruction-tuned sequence models or as dedicated design pipelines (Fang et al., 2024; Dai et al., 2024; Liu et al., 2025). Most of these systems exchange representations once, at the input of the receiver model. Closest to our design, Evolla inserts tanh-gated cross-attention layers into a frozen LLM to read compressed protein representations (Zhou et al., 2025). BioStitch builds on this family and focuses on where the connector reads and writes: it selects the source depth from a layer-wise analysis, injects the same source representation at several receiver depths, and uses the same formulation in either direction. Appendix A gives a fuller account.

3 METHOD

BioStitch is a general framework for connecting biological foundation models and language models through gated cross-attention. It connects a source model, which provides internal representations, with a receiver model whose hidden states attend to these representations. We describe the formulation primarily through the biological-sequence-to-text setting, where the biological model serves as the source and the LLM as the receiver (Figure 1, top row). The text-to-biological-sequence setting follows the same formulation with the source and receiver roles exchanged (Figure 1, bottom row).

Source-to-receiver cross-attention. For the biological-sequence-to-text setting, given a biological sequence x and a natural-language question q, the receiver autoregressively generates an answer a. Let $B^{(k)}$ denote the hidden representation extracted from the k-th layer of the source, and let $H^{(l)}$ denote the hidden states at the l-th layer of the receiver, computed from q and the previously generated answer tokens.

Let $\mathcal{I}$ denote the set of receiver layers at which cross-attention is applied. For each $l \in \mathcal{I}$, the receiver hidden states query the source representation through multi-head attention (denoted as MHA) (Vaswani et al., 2017):

$$\begin{aligned} \widetilde{C}^{(l)} &= \mathrm{MHA}(H^{(l)}, B^{(k)}, B^{(k)}), \quad C^{(l)} = \mathrm{LayerNorm}(\widetilde{C}^{(l)}), \\ D^{(l)} &= \mathrm{Linear}(C^{(l)}), \quad H^{(l)} \leftarrow H^{(l)} + \tanh(\gamma^{(l)}) D^{(l)}. \end{aligned} \tag{1}$$

Here, $H^{(l)}$ provides the queries, while $B^{(k)}$ provides the keys and values. The linear projection maps the attended source features to the hidden dimension of the receiver. Each receiver layer $l \in \mathcal{I}$

has its own multi-head attention, normalization, projection, and learnable scalar gate $\gamma^{(l)}$, and all of them read the same source representation $B^{(k)}$. We refer to the cross-attention, normalization and projection layers, together with the layer-specific gates, as the *connector*. The parameters of the connector are trainable. In our default configuration, the same source representation $B^{(k)}$ is provided to three receiver layers at shallow, middle, and deep positions. Throughout the paper, we define shallow, middle, and deep as the layers closest to 20%, 50%, and 80% of the backbone depth, respectively, for both source extraction and receiver injection. For Evo2-1B, we use its attention blocks 3, 10, and 17.

Paired biological sequences. Some biological tasks involve paired sequence inputs whose differences are informative for prediction. For example, variant-effect tasks provide a reference sequence x_{ref} and an alternative sequence x_{alt}. We encode the two sequences separately, attend to their representations independently, and combine the resulting contexts together with their difference:

$$\widetilde{C}^{(l)}_{\mathrm{ref}} = \mathrm{MHA}(H^{(l)}, B^{(k)}_{\mathrm{ref}}, B^{(k)}_{\mathrm{ref}}), \quad \widetilde{C}^{(l)}_{\mathrm{alt}} = \mathrm{MHA}(H^{(l)}, B^{(k)}_{\mathrm{alt}}, B^{(k)}_{\mathrm{alt}}),$$
$$\widetilde{C}^{(l)} = [\widetilde{C}^{(l)}_{\mathrm{ref}} \| \widetilde{C}^{(l)}_{\mathrm{alt}} \| (\widetilde{C}^{(l)}_{\mathrm{alt}} - \widetilde{C}^{(l)}_{\mathrm{ref}})].$$

Here, $\|$ denotes concatenation along the feature dimension. The combined representation $\widetilde{C}^{(l)}$ is then normalized, projected, and injected into the receiver following Eqn. (1).

Text to biological sequence generation. The same operations, with a separately trained connector, apply with the source and receiver roles exchanged: a frozen language model serves as the source and supplies keys and values, and a frozen biological sequence model, such as a protein language model, serves as the receiver and supplies queries from its preceding tokens. Figure 1 illustrates this direction in its bottom row.

Training. BioStitch is trained with a conditional autoregressive objective. Given a source input and its conditioning context, the receiver generates the target sequence token by token. For biological-sequence-to-text tasks, the target is the natural-language answer a, and we optimize

$$\mathcal{L} = -\frac{1}{T_a} \sum_{t=1}^{T_a} \log p_\theta \left(a_t \mid a_{<t}, q, x \right), \tag{2}$$

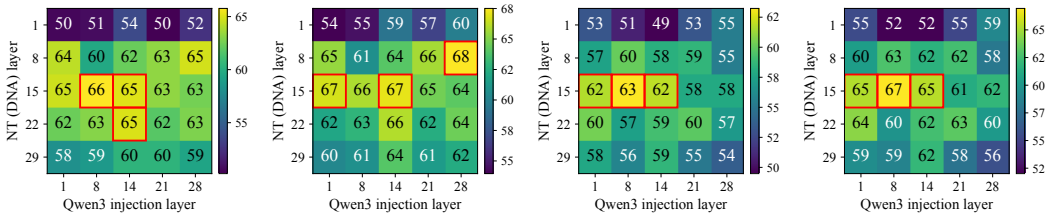
where T_a is the number of answer tokens. The exact set of trainable parameters, together with task-specific optimization procedures and any auxiliary warm-up stage, is specified in Section 4.1 and Appendix B.

4 Experiments

We analyze where the interface should extract and inject (Section 4.2), compare the resulting design with input-space fusion (Section 4.3), test its transfer to other modalities and to the reverse direction (Sections 4.4–4.6), and measure its cost (Section 4.7).

4.1 Setup

We evaluate BioStitch in four settings spanning biological understanding and text-conditioned generation. **DNA-to-text** requires reasoning over genomic sequences for variant-effect prediction (VEC and VE-NonSNV) and disease-pathway questions (KEGG). Following the setup of BioReason (Fallahpour et al., 2025), we use NT-v2-500M (Dalla-Torre et al., 2025) or Evo2-1B (Brixi et al., 2026) as the source model, paired with Qwen3-1.7B or Qwen3-4B (Yang et al., 2025) as the receiver. **Protein-to-text** evaluates natural-language question answering conditioned on protein sequences using the Protein2Text-QA benchmark. We use ESM2-650M (Lin et al., 2023) as the source model and Llama-2-7B (Touvron et al., 2023) as the receiver. **Single-cell-to-text** predicts cell-type labels from single-cell gene-expression profiles. Following the external evaluation protocol of CellWhisperer (Schaefer et al., 2025), we use Geneformer (Theodoris et al., 2023) as the source model and Qwen3-1.7B as the receiver. **Text-to-protein** studies the reverse direction, generating protein sequences from natural-language functional descriptions. Here, we focus on how the proposed interface behaves



(a) VEC, Acc (b) VEC, F1 (c) VE-NonSNV, Acc (d) VE-NonSNV, F1

Figure 2: **Single-point layer-pair probe** (5×5). Rows vary the source extraction layer; columns vary the receiver injection layer. NT-v2-500M is frozen and Qwen3-1.7B is LoRA-tuned. These diagnostic runs use the legacy protocol and one run per cell.

across different protein-generation backbones rather than on a single specialized design system. We use Qwen3-1.7B as the source model and consider NatureLM-1B-Instruct (Xia et al., 2025) and ProGen2-base (Nijkamp et al., 2023) as receivers. The source extraction layers and receiver injection layers used in each setting are summarized in Table A2(c).

Trainable parameters. The source model is frozen in all four settings, while the connector is always trained. For the three sequence-to-text settings, we additionally fine-tune the receiver (i.e., the LLM) with LoRA (Hu et al., 2022), following the BioReason setup in the DNA experiments. For text-to-protein generation, both the source and receiver remain frozen, and only the connector is trained. For tasks with a closed label set, we first warm up the connector with direct label supervision before generative fine-tuning. Specifically, we mean-pool the gated cross-attention features at each receiver injection layer and train an auxiliary classifier with cross-entropy loss; these classifiers are discarded after warm-up. We apply this stage to DNA-to-text and single-cell-to-text, but not to protein-to-text or text-to-protein, whose targets are free-form text and variable-length protein sequences, respectively. Appendix C provides the full procedure and ablation; the reproduced BioReason baseline follows its original training recipe and does not use this stage. Appendix B provides the remaining optimization settings, gate initializations, and data budgets.

4.2 Layer-wise analysis of the interface

We first study whether performance is more sensitive to where representations are extracted from the source or where they are injected into the receiver. We connect a single NT-v2-500M layer to a single Qwen3-1.7B layer and independently sweep five source extraction depths and five receiver injection depths on VEC and VE-NonSNV (Figure 2).

Source extraction depth matters more than receiver injection depth. The heatmaps in Figure 2 form clear horizontal bands: scores remain relatively stable across receiver injection layers but shift substantially across source extraction layers. To quantify this effect, we decompose the variation in each 5×5 grid into source-depth and injection-depth components. The source-depth component accounts for 70–93% of the total variation across the four grids, whereas the injection-depth component accounts for only 1.6–10.7%. Middle source layers also perform consistently better than the shallowest and deepest layers in these probes.

Multi-level injection outperforms single-point injection. Figure 2 also shows that no single receiver injection depth is consistently optimal across tasks. We therefore inject the same source representation at shallow, middle, and deep receiver layers. For Qwen3-1.7B, these correspond to layers 6, 14, and 22. Compared with the best single-point configuration at the middle source layer (NT layer 15; Figure 2), the full multi-level configuration improves F1 from 67.02 to 70.84 on VEC and from 66.94 to 79.96 on VE-NonSNV, corresponding to gains of 3.82 and 13.02 points. This motivates multi-level injection as the default design, avoiding the need to select a task-specific receiver layer.

Intermediate source layers provide the strongest overall default. With multi-level injection fixed, we further compare shallow, middle, and deep source extraction across three DNA tasks and

Table 1: **Source extraction depth under multi-level injection (F1).** SHALLOW, MIDDLE, and DEEP source layers across three DNA→text benchmarks and four source–receiver pairs; accuracy is in Table A15. All runs use seed 23 (one run per cell) and the legacy protocol (validation reuses test). Bold marks the best depth within each dataset–model block.

	VEC			**VE-NonSNV**			**KEGG** (weighted)		
Encoder + LLM	Shallow	Middle	Deep	Shallow	Middle	Deep	Shallow	Middle	Deep
NT + Qwen3-1.7B	69.61	**70.84**	68.51	72.69	**79.96**	77.53	88.85	**90.98**	90.61
NT + Qwen3-4B	67.48	**71.91**	66.52	74.57	78.92	**79.39**	93.81	**94.73**	94.19
Evo2 + Qwen3-1.7B	71.99	68.43	**73.93**	80.45	**89.56**	88.35	90.15	91.15	**91.91**
Evo2 + Qwen3-4B	68.01	**68.94**	68.47	84.31	**91.46**	88.85	92.97	**94.43**	93.82
mean over 12 blocks		SHALLOW 79.6		$<$	DEEP 81.8	$<$	**MIDDLE 82.6**		

Table 2: **Strict held-out DNA-to-text results.** NT-v2-500M is frozen; both systems are locally evaluated with seed 23 and final-epoch checkpoints. Acc. and positive-class F1 are percentages; Δ gives percentage-point gains. VEC prompts are rewritten for both systems.

Dataset	Encoder + LLM	**BioStitch**		BioReason		Δ Acc	Δ F1
		Acc	F1	Acc	F1		
VEC	NT + Qwen3-1.7B	**66.59**	**66.17**	58.88	56.56	+7.71	**+9.61**
VEC	NT + Qwen3-4B	**64.96**	**64.06**	58.15	51.41	+6.81	**+12.65**
VE-NonSNV	NT + Qwen3-1.7B	**72.97**	**77.39**	55.21	61.40	+17.76	**+15.99**
VE-NonSNV	NT + Qwen3-4B	**74.34**	**78.12**	55.90	55.28	+18.44	**+22.84**
Mean		**69.72**	**71.44**	57.04	56.16	+12.68	**+15.27**

four source–receiver pairs. Middle extraction achieves the highest aggregate F1 (82.6, compared with 81.8 for deep and 79.6 for shallow) and accuracy (81.0, compared with 79.8 and 77.2; Table A15). However, it is not optimal in every configuration: deep extraction performs best in three of the twelve F1 settings (Table 1). We therefore use intermediate source layers as the default.

4.3 DNA-TO-TEXT

We now compare the resulting design with input-space fusion, as implemented in BioReason (Fallahpour et al., 2025), under a strict held-out protocol that carves a validation split out of the training data and leaves the test split unchanged (Table A1). Both systems are trained and evaluated locally with NT-v2-500M, identical test sets, the same seed, and rewritten VEC prompts that remove label-correlated gene-name cues (Appendix D). VEC shares sequence pairs between its training and validation splits, but neither overlaps with test.

Consistent gains under strict held-out evaluation. BioStitch improves both accuracy and F1 in all four dataset–receiver configurations (Table 2). F1 gains range from 9.61 to 22.84 points, with a mean of 15.27, and accuracy gains average 12.68 points. On these imbalanced test sets, BioStitch also leads in balanced accuracy and the Matthews correlation coefficient (MCC) in all four configurations, and unpaired 95% intervals for its accuracy gains exclude zero (Appendix E). Our BioReason reproduction scores below its published VE-NonSNV accuracy (55.21 versus 72.82 with Qwen3-1.7B). The BioReason paper describes a held-out validation split, but its released training script validates and selects checkpoints on the test split, and our run of that script also stays below the published value (52.46). BioStitch matches or exceeds even the published values (72.97 and 74.34 versus 72.82 and 65.59).

The gain is concentrated in recall. BioStitch mainly recovers pathogenic variants that BioReason misses: recall gains exceed precision gains in all four configurations (Appendix E). For VE-NonSNV with Qwen3-4B, true positives rise from 238 to 400 while false positives move only from 100 to 101, so nearly all of the net additional positive predictions are correct.

Table 3: **Protein2Text-QA results on the full test split.** Baseline scores are published system results with different training corpora and LLMs. Bold marks the best score per row.

Metric	**BioStitch**	Protein2Text	Mol-Inst-7B	BioMedGPT	ProtT3	Llama-3.1	GPT-4o-mini
BLEU-2	**26.64**	14.4	6.5	7.4	∼0.00	1.37	2.02
BLEU-4	**17.31**	8.3	3.6	3.5	∼0.00	0.67	0.88
ROUGE-1	**37.93**	32.2	18.7	16.0	6.2	4.22	6.98
ROUGE-2	**21.79**	18.0	9.2	5.6	0.1	1.86	2.79
ROUGE-L	**35.26**	28.8	16.8	14.4	6.1	3.87	5.89
METEOR	35.95	**37.7**	27.3	14.0	1.74	11.0	15.6
[CLS] cosine similarity (not BERTScore)							
BERT	**91.02**	89.1	74.3	75.0	76.8	61.3	67.0
BiomedBERT	**95.27**	94.3	87.8	90.5	84.3	80.14	88.0

Robustness of the gain. Removing the connector warm-up, which BioReason does not use, lowers F1 by only 0.8 points on average across twelve legacy-protocol configurations, and on VEC BioStitch without the warm-up still leads BioReason by 7.8–18.8 F1 (Appendix C). The gain holds for both Qwen3 receivers, and the larger receiver brings no consistent benefit: BioStitch F1 changes by -2.11 on VEC and $+0.73$ on VE-NonSNV from 1.7B to 4B. Replacing Qwen3-1.7B with Llama-3.2-1B-Instruct (Grattafiori et al., 2024), a different model family, changes F1 by less than one point under the legacy protocol (Appendix H). The gain therefore does not appear to depend on a particular receiver size or family.

Answer commitment on KEGG. On KEGG disease-pathway questions (Kanehisa & Goto, 2000), BioStitch almost always commits to a single disease, whereas BioReason often names several. With NT + Qwen3-1.7B, BioReason names multiple diseases in 136 of 290 answers, and requiring one committed disease lowers its accuracy from 87.93 to 47.24, versus 91.72 to 91.38 for BioStitch (legacy protocol; Appendix F). An anonymized, order-balanced LLM judge also prefers the BioStitch reasoning chain on 141 of these items versus 50 (Appendix J).

4.4 PROTEIN-TO-TEXT

We next test whether the interface transfers to other source modalities, starting with protein question answering on Protein2Text-QA, where we compare BioStitch with published systems (Table 3). BioStitch is trained on $3{,}898$ Protein2Text-QA pairs with no prior cross-modal alignment and evaluated on the full $38{,}585$-question test split (Jararweh et al., 2025). The published Protein2Text reference instead uses $393{,}849$ alignment pairs and $209{,}847$ QA fine-tuning pairs with Llama-3.1-8B-Instruct, so this is a reference comparison on a shared test split rather than a controlled one. Appendix B describes the training subsets and the overlap audit.

Despite training on over 150 times fewer pairs with an older LLM (Llama-2-7B), BioStitch exceeds the Protein2Text reference on seven of eight metrics. BLEU-2 and BLEU-4 increase by 12.24 and 9.01 points, ROUGE-L increases from 28.8 to 35.26, and both `[CLS]` cosine similarities improve; METEOR is the exception (35.95 versus 37.7).

4.5 SINGLE-CELL-TO-TEXT

Single-cell annotation extends this test from biological sequences to gene-expression profiles. With a frozen Geneformer source and a Qwen3-1.7B receiver, BioStitch trains the connector together with LoRA adapters that add 17.4M parameters (1.0% of the receiver). We evaluate zero-shot cell-type annotation on TS-20 from Tabula Sapiens (The Tabula Sapiens Consortium, 2022) and on AIDA-9 (Kock et al., 2025), following the external benchmark protocol of CellWhisperer (Schaefer et al., 2025): the model is given a gene-expression profile and scores a closed set of candidate cell types that it was never trained to predict.

Without any change to its formulation, the interface reaches the level of a dedicated contrastive retrieval model on this benchmark. CellWhisperer is a contrastive transcriptome–text model that

Table 4: **Zero-shot single-cell cell-type annotation.** A frozen Geneformer source is connected to a LoRA-adapted Qwen3-1.7B receiver. CellWhisperer scores are the published values.

Method	TS-20			AIDA-9		
	Acc	macro-F1	**AUROC**	Acc	macro-F1	**AUROC**
CellWhisperer	55.0	45.0	94.0	79.0	74.0	**99.0**
BioStitch	**57.9**	**53.9**	**95.3**	**84.5**	**83.2**	98.4

Table 5: **Text-to-protein generation on PDFBench.** Only the BioStitch and ProGen2 prefix columns use frozen Qwen3 conditioning. NatureLM LoRA adapts the decoder directly; zero-shot and the authors' released Mol-Inst-7B are reference systems. Training corpora differ between decoder groups; bold compares BioStitch with LoRA (NatureLM) and with the prefix (ProGen2).

Metric	Mol-Inst-7B (released)	NatureLM-1B decoder			ProGen2-base decoder	
		zero-shot‡	+ LoRA	+ BioStitch	+ prefix	+ BioStitch
ProTrek ↑	6.608 ± 0.073	0.680 ± 0.067	6.444 ± 0.126	$\mathbf{9.024 \pm 0.053}$	4.74	**8.22**
Ret. Acc.@10 ↑	24.46	11.51	24.66	**33.56**	23.02	**30.69**
pLDDT ↑	50.844 ± 0.636	54.857 ± 0.610	$\mathbf{62.458 \pm 0.384}$	57.332 ± 0.310	63.77	**65.92**
PAE ↓	18.868 ± 0.049	19.595 ± 0.283	$\mathbf{16.290 \pm 0.154}$	18.421 ± 0.148	**14.82**	14.83
Repeat ↓	19.161 ± 0.308	26.805 ± 0.313	$\mathbf{19.480 \pm 0.205}$	21.221 ± 0.146	**7.62**	16.93
Invalid-prompt drop† ↓	0%	0.02%	0.02%	**0**%	8.37%	**3.01**%

For columns with ±, error bars denote sample standard deviation over four candidate generations, not training seeds. Ret. Acc.@10 is PDFBench's retrieval accuracy against 9 random negative descriptions (NORMAL setting). ‡ Off-the-shelf NatureLM-1B-Instruct evaluated without task adaptation or fine-tuning.
† Rate of prompts discarded due to invalid outputs (counts: 0, 1, 1, 0, 492, and 177); validity checks and metric denominators differ across decoder groups and between the two ProGen2 columns.

ranks candidate labels by cosine similarity, whereas BioStitch scores the same label set by teacher-forced generative likelihood, with a smaller data budget and no separate cross-modal pre-training corpus. BioStitch has higher accuracy and macro-F1 on both datasets (Table 4), and AUROC is 95.3 versus 94.0 on TS-20 and 98.4 versus 99.0 on AIDA-9. As the systems differ in training data, objective, and scoring rule, this is a reference comparison; on AUROC, the metric least sensitive to how label scores are normalized, the two are on par. Appendix G gives the scoring protocol and caveats.

4.6 TEXT-TO-PROTEIN

We then exchange the source and receiver roles to test whether the same formulation can also condition biological generation: a frozen Qwen3-1.7B serves as the source, and a frozen protein decoder serves as the receiver. We evaluate on 5,876 held-out prompts with four generated candidates per prompt using PDFBench (Kuang et al., 2026). Appendices K and B give the full evaluation.

The ProGen2 group is the controlled comparison, with the backbones, corpus, and training budget held fixed: the prefix control projects text representations into tokens prepended to the frozen decoder, as in input-space fusion, whereas BioStitch supplies the same condition through internal cross-attention. The two conditioning paths differ in capacity: BioStitch trains 40.16M parameters and the prefix projection 3.15M (Appendix L). The NatureLM group answers a different question, namely how connector-only training compares with adapting the decoder itself with LoRA. The off-the-shelf NatureLM instruction model serves as a zero-shot reference, and Mol-Inst-7B (Fang et al., 2024) as an external one. ProTrek (Su et al., 2026) and retrieval accuracy measure text–sequence alignment, pLDDT and PAE measure predicted structural confidence, and repetition and invalid-prompt drops capture sequence-level failures.

Alignment improves, with decoder-dependent trade-offs. In the controlled ProGen2 comparison (Table 5), BioStitch improves ProTrek ($4.74 \rightarrow 8.22$), retrieval accuracy ($23.02\% \rightarrow 30.69\%$), and pLDDT ($63.77 \rightarrow 65.92$) over the prefix control, with paired bootstrap 95% intervals excluding zero

Table 6: **Measured cost in the long-DNA setting.** BioReason concatenates biological tokens; BioStitch uses internal cross-attention. Both use Evo2-1B and Qwen3-1.7B with the same batch size and LoRA setup. Prompt KV cache excludes separately retained conditioning states.

Measurement	BioReason *(concatenation)*	**BioStitch** *(cross-attention)*	Ratio
Training			
Peak device memory (`nvidia-smi`)	79.9 GB	**16.8 GB**	$\mathbf{4.7}\times$
Peak tensor allocation (`torch`)	59.8 GB	**12.7 GB**	$\mathbf{4.7}\times$
Time per sample (bs 2, no accum.)	0.393 s	**0.151 s**	$\mathbf{2.6}\times$
Inference			
LLM prompt length	4,158 tok	**58 tok**	$\mathbf{71.7}\times$
LLM KV cache at prompt	476.9 MB	**6.65 MB**	$\mathbf{71.7}\times$
Prefill latency per sample	205.2 ms	**96.4 ms**	$\mathbf{2.13}\times$
Peak memory (torch reserved, bs 1)	11.3 GB	10.8 GB	$\approx$ equal
Decode latency per token	**34.9 ms**	36.1 ms	BioStitch 3% slower
End-to-end latency, 64 tokens	2,404 ms	2,368 ms	$\approx$ equal

for all three gains. Invalid-prompt drops also fall from 8.37% to 3.01%, while PAE is unchanged and repetition rises from 7.62 to 16.93. With NatureLM, whose zero-shot retrieval accuracy of 11.51% is near the 10% random baseline, BioStitch raises ProTrek from 6.44 to 9.02 and retrieval accuracy from 24.66% to 33.56% relative to direct LoRA, while pLDDT falls from 62.46 to 57.33 and PAE and repetition increase. Internal cross-attention therefore strengthens the measured conditioning signal under frozen backbones without uniformly improving the other generation criteria. Dedicated design systems such as Pinal and ProDVa, which differ in training data, model scale, and pipeline, remain stronger on several absolute metrics (Appendix K).

4.7 Efficiency

Internal cross-attention also keeps long biological representations out of the receiver's self-attention. For text length L and biological-feature length M, direct concatenation of biological tokens, as in BioReason, incurs self-attention over $L + M$ tokens, whereas BioStitch retains $O(L^2)$ text self-attention and adds $O(LM)$ cross-attention at the selected layers. Encoding and storing the conditioning states still incur costs.

In the measured long-DNA setting, the savings concentrate in training and prompt state (Table 6). Peak device memory falls from 79.9 to 16.8 GB and training time per sample from 0.393 to 0.151 s. The LLM prompt contracts from $4{,}158$ to 58 tokens, reducing its prompt KV cache from 476.9 to 6.65 MB; this is a cache-specific saving, as peak reserved memory is 11.3 versus 10.8 GB. Prefill is $2.13\times$ faster, whereas per-token decoding is about 3% slower and end-to-end latency for 64 tokens is nearly unchanged. In the reverse direction, BioStitch still lowers peak training memory by $1.55\times$ and time per sample by $1.62\times$, but with short conditioning text, prefill differs by only $1.04\times$ (Appendix L); training savings therefore hold in both directions.

5 Conclusions

We found, in DNA-to-text probes, that source extraction depth matters more than injection depth, with intermediate layers the best default on average. Building on this, BioStitch connects pretrained models through multi-level gated cross-attention and, by exchanging the source and receiver roles, supports both biological-sequence understanding and text-guided protein generation. It improves strict held-out DNA variant-effect prediction, compares favorably with published systems on protein question answering and single-cell annotation, and strengthens text–sequence alignment in protein generation, while reducing training memory and prompt-state cost for long biological inputs. In future work, we will test whether the source-depth finding holds for other source modalities.

REPRODUCIBILITY STATEMENT

Appendix B reports data splits, model configurations, optimization settings, hardware, and checkpoint selection; Appendix D gives the VEC prompt rewrites and Appendix E the checkpoint and classification details. Appendix J.1 specifies the automated judging protocol and Appendix G the single-cell scoring protocol. Captions distinguish local reruns from published baselines, and legacy from strict held-out results. Tables A1 and A10 report the VEC training–validation overlap and the available training seeds. Code, processed data splits, rewritten prompts, and evaluation scripts will be released in a public repository.

ETHICS STATEMENT

This work uses public research datasets to study biological–language interfaces. Variant-effect predictions are research outputs and should not be used for clinical decisions without independent validation. Generated protein sequences have not been experimentally validated; text-conditioned design can also create dual-use risks. Any subsequent synthesis or deployment requires appropriate biological safety review. Generated explanations and automated evaluation scores should not be treated as verified biological evidence.

AI USE STATEMENT

In this work, we used generative AI tools as the automated judge for the reasoning-chain evaluation in Appendix J. Additionally, we used generative AI tools to polish the language of the manuscript. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikołaj Bińkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. Flamingo: a Visual Language Model for Few-Shot Learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 23716–23736. Curran Associates, Inc., 2022. doi: 10.52202/068431-1723. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/960a172bc7fbf0177ccccbb411a7d800-Paper-Conference.pdf.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-VL: A Versatile Vision-Language Model for Understanding, Localization, Text Reading, and Beyond. *arXiv preprint arXiv:2308.12966*, 2023. URL https://arxiv.org/abs/2308.12966.

Gonzalo Benegas, Carlos Albors, Alan J. Aw, Chengzhong Ye, and Yun S. Song. A DNA language model based on multispecies alignment predicts the effects of genome-wide variants. *Nature Biotechnology*, 43(12):1960–1965, 2025. doi: 10.1038/s41587-024-02511-w.

Garyk Brixi, Matthew G. Durrant, Jerome Ku, Mohsen Naghipourfar, Michael Poli, Gwanggyu Sun, Greg Brockman, Daniel Chang, Alison Fanton, Gabriel A. Gonzalez, Samuel H. King, David B. Li, Aditi T. Merchant, Eric Nguyen, Chiara Ricci-Tam, David W. Romero, Jonathan C. Schmok, Ali Taghibakhshi, Anton Vorontsov, Brandon Yang, Myra Deng, Liv Gorton, Nam Nguyen, Nicholas K. Wang, Michael T. Pearce, Elana Simon, Etowah Adams, Zachary J. Amador, Euan A. Ashley, Stephen A. Baccus, Haoyu Dai, Steven Dillmann, Stefano Ermon, Daniel Guo, Michael H. Herschl, Rajesh Ilango, Ken Janik, Amy X. Lu, Reshma Mehta, Mohammad R. K. Mofrad, Madelena Y. Ng, Jaspreet Pannu, Christopher Ré, John St. John, Jeremy Sullivan, Joseph Tey, Ben Viggiano, Kevin Zhu, Greg Zynda, Daniel Balsam, Patrick Collison, Anthony B. Costa, Tina Hernandez-Boussard, Eric Ho, Ming-Yu Liu, Thomas McGrath, Kimberly Powell, Sudarshan Pinglay, Dave P. Burke,

Hani Goodarzi, Patrick D. Hsu, and Brian L. Hie. Genome modelling and design across all domains of life with Evo 2. *Nature*, 652(8112):1349–1361, 2026. doi: 10.1038/s41586-026-10176-5.

Siwei Chen, Laurent C. Francioli, Julia K. Goodrich, Ryan L. Collins, Masahiro Kanai, Qingbo Wang, Jessica Alföldi, Nicholas A. Watts, Christopher Vittal, Laura D. Gauthier, et al. A genomic mutational constraint map using variation in 76,156 human genomes. *Nature*, 625(7993):92–100, 2024. doi: 10.1038/s41586-023-06045-0.

Haotian Cui, Chloe Wang, Hassaan Maan, Kuan Pang, Fengning Luo, Nan Duan, and Bo Wang. scGPT: toward building a foundation model for single-cell multi-omics using generative AI. *Nature Methods*, 21(8):1470–1480, 2024. doi: 10.1038/s41592-024-02201-0.

Fengyuan Dai, Yuliang Fan, Jin Su, Chentong Wang, Chenchen Han, Xibin Zhou, Jianming Liu, Hui Qian, Shunzhi Wang, Anping Zeng, Yajie Wang, and Fajie Yuan. Toward De Novo Protein Design from Natural Language. *bioRxiv*, 2024. URL https://www.biorxiv.org/content/10.1101/2024.08.01.606258v1.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. InstructBLIP: Towards General-purpose Vision-Language Models with Instruction Tuning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 49250–49267. Curran Associates, Inc., 2023. doi: 10.52202/075280-2142. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/9a6a435e75419a836fe47ab6793623e6-Paper-Conference.pdf.

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, Guillaume Richard, Marcin Skwark, Karim Beguir, Marie Lopez, and Thomas Pierrot. Nucleotide Transformer: building and evaluating robust foundation models for human genomics. *Nature Methods*, 22(2):287–297, 2025. doi: 10.1038/s41592-024-02523-z.

Bernardo P. de Almeida, Guillaume Richard, Hugo Dalla-Torre, Christopher Blum, Lorenz Hexemer, Priyanka Pandey, Stefan Laurent, Chandana Rajesh, Marie Lopez, Alexandre Laterre, Maren Lang, Uğur Şahin, Karim Beguir, and Thomas Pierrot. A multimodal conversational agent for DNA, RNA and protein tasks. *Nature Machine Intelligence*, 7(6):928–941, 2025. doi: 10.1038/s42256-025-01047-1.

Adibvafa Fallahpour, Andrew Magnuson, Purav Gupta, Shihao Ma, Jack Naimer, Arnav Shah, Haonan Duan, Omar Ibrahim, Hani Goodarzi, Chris J. Maddison, and Bo Wang. BioReason: Incentivizing Multimodal Biological Reasoning within a DNA-LLM Model. In *Advances in Neural Information Processing Systems*, volume 38, pp. 123695–123722, 2025. doi: 10.52202/085713-3728. URL https://openreview.net/forum?id=mDjEKAwJOF.

Yin Fang, Xiaozhuan Liang, Ningyu Zhang, Kangwei Liu, Rui Huang, Zhuo Chen, Xiaohui Fan, and Huajun Chen. Mol-Instructions: A Large-Scale Biomolecular Instruction Dataset for Large Language Models. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Tlsdsb6l9n.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, et al. The Llama 3 Herd of Models. *arXiv preprint arXiv:2407.21783*, 2024. URL https://arxiv.org/abs/2407.21783.

Albert Gu and Tri Dao. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. In *Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=tEYskw1VY2.

Minsheng Hao, Jing Gong, Xin Zeng, Chiming Liu, Yucheng Guo, Xingyi Cheng, Taifeng Wang, Jianzhu Ma, Xuegong Zhang, and Le Song. Large-scale foundation model on single-cell transcriptomics. *Nature Methods*, 21(8):1481–1491, 2024. doi: 10.1038/s41592-024-02305-7.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*, 2022. URL https://iclr.cc/virtual/2022/poster/6319.

Ala Jararweh, Oladimeji Macaulay, David Arredondo, Yue Hu, Luis E Tafoya, Kushal Virupakshappa, and Avinash Sahu. Protein2Text: Resampling Mechanism to Translate Protein Sequences into Human-Interpretable Text. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: Industry Track)*, pp. 918–937. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.naacl-industry.68. URL https://aclanthology.org/2025.naacl-industry.68/.

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. doi: 10.1093/bioinformatics/btab083.

Qiao Jin, Yifan Yang, Qingyu Chen, and Zhiyong Lu. GeneGPT: augmenting large language models with domain tools for improved access to biomedical information. *Bioinformatics*, 40(2):btae075, 2024. doi: 10.1093/bioinformatics/btae075.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, 2021. doi: 10.1038/s41586-021-03819-2.

Minoru Kanehisa and Susumu Goto. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Research*, 28(1):27–30, 2000. doi: 10.1093/nar/28.1.27.

Kian Hong Kock, Le Min Tan, Kyung Yeon Han, Yoshinari Ando, Damita Jevapatarakul, Ankita Chatterjee, Quy Xiao Xuan Lin, Eliora Violain Buyamin, Radhika Sonthalia, Deepa Rajagopalan, Yoshihiko Tomofuji, Shvetha Sankaran, Mi-So Park, Mai Abe, Juthamard Chantaraamporn, Seiko Furukawa, Supratim Ghosh, Gyo Inoue, Miki Kojima, Tsukasa Kouno, Jinyeong Lim, Keiko Myouzen, Sarintip Nguantad, Jin-Mi Oh, Nirmala Arul Rayan, Sumanta Sarkar, Akari Suzuki, Narita Thungsatianpun, Prasanna Nori Venkatesh, Jonathan Moody, Masahiro Nakano, Ziyue Chen, Chi Tian, Yuntian Zhang, Yihan Tong, Crystal T.Y. Tan, Anteneh Mehari Tizazu, Marie Loh, You Yi Hwang, Roger C. Ho, Anis Larbi, Tze Pin Ng, Hong-Hee Won, Fred A. Wright, Alexandra-Chloé Villani, Jong-Eun Park, Murim Choi, Boxiang Liu, Arindam Maitra, Manop Pithukpakorn, Bhoom Suktitipat, Kazuyoshi Ishigaki, Yukinori Okada, Kazuhiko Yamamoto, Piero Carninci, John C. Chambers, Chung-Chau Hon, Ponpan Matangkasombut, Varodom Charoensawan, Partha P. Majumder, Jay W. Shin, Woong-Yang Park, and Shyam Prabhakar. Asian diversity in human immune cells. *Cell*, 188(8):2288–2306.e24, 2025. doi: 10.1016/j.cell.2025.02.017.

Jiahao Kuang, Nuowei Liu, Changzhi Sun, Jie Wang, Tao Ji, and Yuanbin Wu. PDFBench: A Benchmark for De Novo Protein Design from Function. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=Z6dCaMpiqV.

Melissa J Landrum, Jennifer M Lee, Mark Benson, Garth R Brown, Chen Chao, Shanmuga Chitipiralla, Baoshan Gu, Jennifer Hart, Douglas Hoffman, Wonhee Jang, Karen Karapetyan, Kenneth Katz, Chunlei Liu, Zenith Maddipatla, Adriana Malheiro, Kurt McDaniel, Michael Ovetsky, George Riley, George Zhou, J Bradley Holmes, Brandi L Kattman, and Donna R Maglott. ClinVar: improving access to variant interpretations and supporting evidence. *Nucleic Acids Research*, 46 (D1):D1062–D1067, 2018. doi: 10.1093/nar/gkx1153.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 12888–12900. PMLR, 2022. URL https://proceedings.mlr.press/v162/li22n.html.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 19730–19742. PMLR, 2023. URL https://proceedings.mlr.press/v202/li23q.html.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Salvatore Candido, and Alexander Rives. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023. doi: 10.1126/science.ade2574.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual Instruction Tuning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 34892–34916. Curran Associates, Inc., 2023. doi: 10.52202/075280-1516. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/6dcf277ea32ce3288914faf369fe6de0-Paper-Conference.pdf.

Nuowei Liu, Jiahao Kuang, Yanting Liu, Tao Ji, Changzhi Sun, Man Lan, and Yuanbin Wu. Protein Design with Dynamic Protein Vocabulary. In *Advances in Neural Information Processing Systems*, volume 38, pp. 22638–22669, 2025. doi: 10.52202/085713-0672.

Tianyu Liu, Yijia Xiao, Xiao Luo, Hua Xu, Wenjin Zheng, and Hongyu Zhao. Geneverse: A Collection of Open-source Multimodal Large Language Models for Genomic and Proteomic Research. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 4819–4836. Association for Computational Linguistics, 2024a. doi: 10.18653/v1/2024.findings-emnlp.277. URL https://aclanthology.org/2024.findings-emnlp.277/.

Zhiyuan Liu, An Zhang, Hao Fei, Enzhi Zhang, Xiang Wang, Kenji Kawaguchi, and Tat-Seng Chua. ProtT3: Protein-to-Text Generation for Text-based Protein Understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5949–5966. Association for Computational Linguistics, 2024b. doi: 10.18653/v1/2024.acl-long.324. URL https://aclanthology.org/2024.acl-long.324/.

Renqian Luo, Liai Sun, Yingce Xia, Tao Qin, Sheng Zhang, Hoifung Poon, and Tie-Yan Liu. BioGPT: generative pre-trained transformer for biomedical text generation and mining. *Briefings in Bioinformatics*, 23(6):bbac409, 2022. doi: 10.1093/bib/bbac409.

Yizhen Luo, Jiahuan Zhang, Siqi Fan, Kai Yang, Massimo Hong, Yushuai Wu, Mu Qiao, and Zaiqing Nie. BioMedGPT: An Open Multimodal Large Language Model for BioMedicine. *IEEE Journal of Biomedical and Health Informatics*, 30(2):981–992, 2026. doi: 10.1109/JBHI.2024.3505955.

Eric Nguyen, Michael Poli, Marjan Faizi, Armin Thomas, Michael Wornow, Callum Birch-Sykes, Stefano Massaroli, Aman Patel, Clayton Rabideau, Yoshua Bengio, Stefano Ermon, Christopher Ré, and Stephen Baccus. HyenaDNA: Long-Range Genomic Sequence Modeling at Single Nucleotide Resolution. In *Advances in Neural Information Processing Systems*, volume 36, pp. 43177–43201. Curran Associates, Inc., 2023. doi: 10.52202/075280-1872. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/86ab6927ee4ae9bde4247793c46797c7-Paper-Conference.pdf.

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, Jeremy Sullivan, Madelena Y. Ng, Ashley Lewis, Aaron Lou, Stefano Ermon, Stephen A. Baccus, Tina Hernandez-Boussard, Christopher Ré, Patrick D. Hsu, and Brian L. Hie. Sequence modeling and design from molecular to genome scale with Evo. *Science*, 386(6723):eado9336, 2024. doi: 10.1126/science.ado9336.

Erik Nijkamp, Jeffrey A. Ruffolo, Eli N. Weinstein, Nikhil Naik, and Ali Madani. ProGen2: Exploring the boundaries of protein language models. *Cell Systems*, 14(11):968–978.e3, 2023. doi: 10.1016/j.cels.2023.10.002.

Michael Poli, Stefano Massaroli, Eric Nguyen, Daniel Y Fu, Tri Dao, Stephen Baccus, Yoshua Bengio, Stefano Ermon, and Christopher Ré. Hyena Hierarchy: Towards Larger Convolutional

Language Models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 28043–28078. PMLR, 2023. URL https://proceedings.mlr.press/v202/poli23a.html.

Moritz Schaefer, Peter Peneder, Daniel Malzl, Salvo Danilo Lombardo, Mihaela Peycheva, Jake Burton, Anna Hakobyan, Varun Sharma, Thomas Krausgruber, Celine Sin, Jörg Menche, Eleni M. Tomazou, and Christoph Bock. Multimodal learning enables chat-based exploration of single-cell data. *Nature Biotechnology*, 2025. doi: 10.1038/s41587-025-02857-9.

Yair Schiff, Chia Hsiang Kao, Aaron Gokaslan, Tri Dao, Albert Gu, and Volodymyr Kuleshov. Caduceus: Bi-Directional Equivariant Long-Range DNA Sequence Modeling. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 43632–43648. PMLR, 2024. URL https://proceedings.mlr.press/v235/schiff24a.html.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S. Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, Perry Payne, Martin Seneviratne, Paul Gamble, Chris Kelly, Abubakr Babiker, Nathanael Schärli, Aakanksha Chowdhery, Philip Mansfield, Dina Demner-Fushman, Blaise Agüera y Arcas, Dale Webster, Greg S. Corrado, Yossi Matias, Katherine Chou, Juraj Gottweis, Nenad Tomasev, Yun Liu, Alvin Rajkomar, Joelle Barral, Christopher Semturs, Alan Karthikesalingam, and Vivek Natarajan. Large language models encode clinical knowledge. *Nature*, 620(7972):172–180, 2023. doi: 10.1038/s41586-023-06291-2.

Jin Su, Yan He, Shiyang You, Shiyu Jiang, Xibin Zhou, Xuting Zhang, Yuxuan Wang, Xining Su, Igor Tolstoy, Xing Chang, Hongyuan Lu, and Fajie Yuan. A trimodal protein language model enables advanced protein searches. *Nature Biotechnology*, 44(8):1310–1316, 2026. doi: 10.1038/s41587-025-02836-0.

Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. Galactica: A Large Language Model for Science. *arXiv preprint arXiv:2211.09085*, 2022. URL https://arxiv.org/abs/2211.09085.

The Tabula Sapiens Consortium. The Tabula Sapiens: A multiple-organ, single-cell transcriptomic atlas of humans. *Science*, 376(6594):eabl4896, 2022. doi: 10.1126/science.abl4896.

Christina V. Theodoris, Ling Xiao, Anant Chopra, Mark D. Chaffin, Zeina R. Al Sayed, Matthew C. Hill, Helene Mantineo, Elizabeth M. Brydon, Zexian Zeng, X. Shirley Liu, and Patrick T. Ellinor. Transfer learning enables predictions in network biology. *Nature*, 618(7965):616–624, 2023. doi: 10.1038/s41586-023-06139-9.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open Foundation and Fine-Tuned Chat Models. *arXiv preprint arXiv:2307.09288*, 2023. URL https://arxiv.org/abs/2307.09288.

Maria Tsimpoukelli, Jacob L Menick, Serkan Cabi, S. M. Ali Eslami, Oriol Vinyals, and Felix Hill. Multimodal Few-Shot Learning with Frozen Language Models. In *Advances in Neural Information Processing Systems*, volume 34, pp. 200–212. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/01b7575c38dac42f3cfb7d500438b875-Paper.pdf.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://papers.nips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Yingce Xia, Peiran Jin, Shufang Xie, Liang He, Chuan Cao, Renqian Luo, Guoqing Liu, Yue Wang, Zequn Liu, Yuan-Jyue Chen, Zekun Guo, Yeqi Bai, Pan Deng, Yaosen Min, Ziheng Lu, Hongxia Hao, Han Yang, Jielan Li, Chang Liu, Jia Zhang, Jianwei Zhu, Ran Bi, Kehan Wu, Wei Zhang, Kaiyuan Gao, Qizhi Pei, Qian Wang, Xixian Liu, Yanting Li, Houtian Zhu, Yeqing Lu, Mingqian Ma, Zun Wang, Tian Xie, Krzysztof Maziarz, Marwin Segler, Zhao Yang, Zilong Chen, Yu Shi, Shuxin Zheng, Lijun Wu, Chen Hu, Peggy Dai, Tie-Yan Liu, Haiguang Liu, and Tao Qin. Nature Language Model: Deciphering the Language of Nature for Scientific Discovery. *arXiv preprint arXiv:2502.07527*, 2025. URL https://arxiv.org/abs/2502.07527.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 Technical Report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Fan Yang, Wenchuan Wang, Fang Wang, Yuan Fang, Duyu Tang, Junzhou Huang, Hui Lu, and Jianhua Yao. scBERT as a large-scale pretrained deep language model for cell type annotation of single-cell RNA-seq data. *Nature Machine Intelligence*, 4(10):852–866, 2022. doi: 10.1038/s42256-022-00534-z.

Jiabo Ye, Haiyang Xu, Haowei Liu, Anwen Hu, Ming Yan, Qi Qian, Ji Zhang, Fei Huang, and Jingren Zhou. mPLUG-Owl3: Towards Long Image-Sequence Understanding in Multi-Modal Large Language Models. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=pr37sbuhVa.

Shaolei Zhang, Qingkai Fang, Zhe Yang, and Yang Feng. LLaVA-Mini: Efficient Image and Video Large Multimodal Models with One Vision Token. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=UQJ7CDW8nb.

Xibin Zhou, Chenchen Han, Yingqi Zhang, Huan Du, Jiayuan Tian, Jin Su, Renju Liu, Kai Zhuang, Shiyu Jiang, Anthony Gitter, Li Liu, Huayu Li, Meiqi Wu, Shiyang You, Zichen Yuan, Feng Ju, Huilin Zhang, Wei Zheng, Fengyuan Dai, Yuyang Zhou, Yuyang Tao, Dan Wu, Zongze Shao, Yang Liu, Hongyuan Lu, and Fajie Yuan. Decoding the Molecular Language of Proteins with Evolla. *bioRxiv*, 2025. doi: 10.1101/2025.01.05.630192. URL https://www.biorxiv.org/content/10.1101/2025.01.05.630192.

Zhihan Zhou, Yanrong Ji, Weijian Li, Pratik Dutta, Ramana V Davuluri, and Han Liu. DNABERT-2: Efficient Foundation Model and Benchmark For Multi-Species Genomes. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=oMLQB4EZE1.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. MiniGPT-4: Enhancing Vision-Language Understanding with Advanced Large Language Models. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=1tZbq88f27.

APPENDIX

A EXTENDED RELATED WORK

Biological foundation models. Genomic models use different architectures to handle long sequences. DNABERT (Ji et al., 2021; Zhou et al., 2024) and the Nucleotide Transformer (Dalla-Torre et al., 2025) apply Transformer self-attention to tokenized DNA. HyenaDNA (Nguyen et al., 2023) and Caduceus (Schiff et al., 2024) use Hyena operators (Poli et al., 2023) and structured state spaces (Gu & Dao, 2024), respectively, while the Evo models (Nguyen et al., 2024; Brixi et al., 2026) use StripedHyena architectures for long genomic sequences. Protein models also differ in their training objectives: masked encoders such as ESM2 (Lin et al., 2023) learn residue representations for downstream prediction, whereas autoregressive models such as ProGen2 (Nijkamp et al., 2023) generate sequences. Structure prediction is a related but distinct objective, exemplified by AlphaFold (Jumper et al., 2021). NatureLM (Xia et al., 2025) extends autoregressive modeling across scientific token domains and supports instruction tuning. For BioStitch, the relevant distinction is the role of the biological backbone: an encoder supplies keys and values in sequence-to-text tasks, while a generator supplies queries in text-to-protein generation. Single-cell models include Geneformer (Theodoris et al., 2023), scBERT (Yang et al., 2022), scGPT (Cui et al., 2024), and scFoundation (Hao et al., 2024); CellWhisperer (Schaefer et al., 2025) aligns transcriptomes with text. Section 4.5 evaluates this modality.

Where modalities enter a language model. Vision–language systems illustrate several ways to connect pretrained representations. *Input-space fusion* projects modality features into the language model's embedding space and appends them to the text, as in Frozen (Tsimpoukelli et al., 2021), LLaVA (Liu et al., 2023), and MiniGPT-4 (Zhu et al., 2024). The projected tokens extend the sequence processed by the decoder's self-attention layers. *Compression* maps modality features to a smaller token budget, as in BLIP-2 (Li et al., 2023), InstructBLIP (Dai et al., 2023), and Qwen-VL (Bai et al., 2023); LLaVA-Mini (Zhang et al., 2025) combines compression with visual information fusion into text tokens. This reduces decoder context length, with a tradeoff between the token budget and retained information. *Cross-attention* lets hidden states query modality representations without appending them to the decoder's sequence. Flamingo (Alayrac et al., 2022) introduces gated cross-attention into a frozen language model, and mPLUG-Owl3 (Ye et al., 2025) studies efficient cross-modal attention integration. BLIP (Li et al., 2022) also uses cross-attention for multimodal encoding and decoding. Connector structure and freezing schedule vary across this family.

Biology–language systems. Several systems connect biological encoders to language models through projected or resampled input features, including ChatNT (de Almeida et al., 2025), BioReason (Fallahpour et al., 2025), Protein2Text (Jararweh et al., 2025), ProtT3 (Liu et al., 2024b), and BioMedGPT (Luo et al., 2026). Other approaches acquire domain knowledge from text and instruction data, including BioGPT (Luo et al., 2022), Galactica (Taylor et al., 2022), Med-PaLM (Singhal et al., 2023), and Geneverse (Liu et al., 2024a); GeneGPT (Jin et al., 2024) instead accesses external genomic tools. For text-to-protein generation, Mol-Instructions (Fang et al., 2024) provides biological instruction data, while Pinal (Dai et al., 2024) and ProDVa (Liu et al., 2025) develop dedicated generation pipelines. Evolla (Zhou et al., 2025) is the closest biology–language design: a resampler compresses protein representations to a fixed length, and tanh-gated cross-attention layers inserted every four decoder layers of a frozen LLM read them. Protein2Text reports that adding a gated cross-attention layer after its resampler, in which projected protein features attend to the instruction before both enter the LLM as input tokens, lowered its QA scores (Jararweh et al., 2025); in BioStitch, the receiver's hidden states query the source representation inside the network. Our focus is the interface between existing backbones: BioStitch exposes internal representations through gated cross-attention and uses the same connector formulation in both directions by exchanging the query and key/value roles. The depth experiments in Section 4.2 examine where this interface should read biological features and where it should inject them into the language model.

B TRAINING DATA, SPLITS AND HYPERPARAMETERS

Table A1 summarizes the DNA data and split conventions; Table A2 lists optimization and model settings.

Table A1: **Data and split conventions for the DNA experiments.** The upper block is the legacy protocol inherited from BioReason, under which the test split is also used for validation; the lower block is the strict held-out convention used for Table 2, which carves a 10% validation split out of training and leaves the test split byte-identical.

Dataset	Train	Validation	Test	Notes
legacy protocol				
VEC	48,850	1,233	1,233	Coding-variant split, prompts rewritten as in Appendix D; validation reuses test. Test labels 676 pathogenic / 557 benign.
VE-NonSNV	35,215	873	873	Validation reuses test. Test labels 523 / 350. The training target is the full answer string; evaluation extracts the binary label.
KEGG	1,159	290	290	The original split is validation 144 + test 146; both validation and final testing use their union. 33 disease labels.
strict held-out				
VEC	43,964	4,886	1,233	Validation carved out of the original 48,850; test unchanged.
VE-NonSNV	31,694	3,521	873	Validation carved out of the original 35,215; test unchanged.

In the strict VEC split, training and validation can contain differently worded questions about the same variant; neither overlaps with the test split.

We use the variant-effect datasets released with BioReason (Fallahpour et al., 2025). VEC, called VEP-Coding in that work, originates from the GPN-MSA study (Benegas et al., 2025): pathogenic variants and disease phenotypes come from ClinVar (Landrum et al., 2018), while benign variants come from gnomAD (Chen et al., 2024). Its chromosome split holds out chromosome 8 for testing. VE-NonSNV, called VEP-Non-SNV in BioReason, contains coding non-SNVs in the nuclear genome from ClinVar, restricted to variants of at most 64 base pairs and a review status of at least two stars; disease representation is balanced across training and test data.

Under the *legacy protocol*, validation reuses the test split. Our *strict held-out protocol* instead reserves 10% of the original training set for validation and retains the original test set. The VEC training and validation sets can still contain differently worded questions about the same variant, as disclosed below Table A1.

Protein question answering. For protein→text, we train on the released Protein2Text-QA (Jararweh et al., 2025) `intro` and `discussion` subsets: $1{,}269 + 2{,}629 = 3{,}898$ QA pairs over 191 proteins. We evaluate on the full official `test` split of 38,585 QA pairs over 1,044 proteins. The dataset authors designated the two smaller subsets for cross-domain evaluation; using them for training is our adaptation of the released data. The reference system instead uses 393,849 pre-alignment pairs and 209,847 QA fine-tuning pairs, so this is a common test-set comparison at very different training budgets. Our overlap audit found no shared proteins between the 191 training proteins and the 1,044 test proteins under exact amino-acid sequence identity or protein name, affecting 0 of the 38,585 test QA pairs.

Text-conditioned protein generation. The NatureLM runs use the Mol-Instructions (Fang et al., 2024) `protein_design` split, with 180,594 training, 9,505 validation, and 5,876 test pairs. The ProGen2 runs use the ProDVa (Liu et al., 2025) Molinst-SwissProtCLAP corpus, with 712,248 training pairs. Both decoder groups are evaluated using PDFBench (Kuang et al., 2026) on the same

Table A2: **Hyperparameters for the four settings.** (a) Task settings; a standalone "–" repeats the DNA→text entry. (b) DNA→text generation settings in Stage 2 (Appendix C). (c) Source and injection indices, listed in shallow/middle/deep order where three layers are shown.

(a) By task

Hyperparameter	DNA → text	Protein → text	Single-cell → text	Text → protein
Source model	NT-v2-500M / Evo2-1B	ESM2-650M	Geneformer-12L-30M	Qwen3-1.7B
Receiver model	Qwen3-1.7B / 4B	Llama-2-7B	Qwen3-1.7B	NatureLM-1B-Instruct / ProGen2-base
Receiver adaptation	LoRA 32 / 64 / 0.05	LoRA 16	LoRA 16 / 32 / 0.05	None (frozen)
Trained components	Connector + LoRA	Connector + LoRA	–	Connector only
Seed	23	–	–	–
Precision	bf16-mixed	–	–	–
Optimizer	AdamW	–	–	AdamW ($\beta = 0.9/0.95$)
Base learning rate	5×10^{-5}	–	–	5×10^{-4}
Weight decay	0.01	–	–	–
Warmup / scheduler	0.05 / cosine	–	–	–
Effective batch	16 (8 on KEGG)	128	128	128 (NatureLM) / 192 (ProGen2)
Gate init / clamp	0.1 / 3.0	–	–	0.03 / –
Max condition length	2048 (1024 bp per side)	1022 residues	2048 rank tokens	384
Max target length	1024	640	256 / 128 (Stage 1 / 2)	1024
Checkpoint choice	Final epoch	Validation loss	Best val F1	Final epoch
Hardware	1× H100	8× A100	4× A100	8× A100

In single-cell → text, the LoRA adapters add $17{,}432{,}576$ trainable parameters (1.003% of the receiver), and the connector is trained in addition. Stage 1 runs 2 epochs over a 50-class subset, and Stage 2 runs 6 epochs over the full 482-class label set.

(b) DNA→text, Stage 2

Dataset	Text model	Batch	Accum.	Max gen.
VEC	Qwen3-1.7B	2	8	64
VEC	Qwen3-4B	1	16	64
VE-NonSNV	Qwen3-1.7B	2	8	512
VE-NonSNV	Qwen3-4B	1	16	512
KEGG	Both	1	8	800

(c) Source and injection layers

Model (role)	Shallow / middle / deep
Qwen3-1.7B (injection)	$\{6, 14, 22\}$
Qwen3-4B (injection)	$\{7, 18, 29\}$
NT-v2-500M (source)	$\{6, 15, 23\}$
Evo2-1B (source)	$\{3, 10, 17\}$
ESM2-650M (source)	$\{7, 17, 26\}$
Text → protein	
Qwen3-1.7B (source)	14
NatureLM-1B (injection)	$\{3, 8, 13\}$
ProGen2-base (injection)	$\{5, 14, 22\}$

$5{,}876$ test prompts in the same order. The groups differ in their backbones and training data, and evaluator-specific filtering produces the invalid-prompt counts reported in Table 5.

Training configurations. DNA→text runs on NVIDIA H100 GPUs; protein→text, single-cell, and text→protein run on NVIDIA A100 GPUs. DNA→text and single-cell use the two-stage schedule of Appendix C; protein→text and text→protein train with the conditional objective alone. Table A2 lists task settings, DNA generation settings, and source/injection layers; in Table A2(a), a standalone "–" repeats the DNA→text entry. The text→protein runs freeze both backbones and use a learning rate of 5×10^{-4} with a gate initialization of 0.03, against 5×10^{-5} and 0.1 in DNA→text.

C CONNECTOR WARM-UP

Two of the four settings train the connector in two stages. This appendix states the warm-up objective, the condition a task must satisfy for it to be defined, and the settings in which that condition does not hold.

Stage 1: warm-up. The biological source is frozen and the receiver is given only the question, not the answer. At each injection site $l \in \mathcal{I}$ we mean-pool the gated context over the question tokens, $d^{(l)} = \mathrm{MeanPool}(\tanh(\gamma^{(l)})D^{(l)})$, attach a linear classifier ψ_l, and minimize

$$\mathcal{L}_{\mathrm{warm}} = \frac{1}{|\mathcal{I}|} \sum_{l \in \mathcal{I}} \mathrm{CE}\left(\psi_l(d^{(l)}),\, y\right), \tag{3}$$

where y is the class label of the answer, with any reasoning steps excluded. Every injection site therefore receives direct supervision before the autoregressive objective is applied, and the classifiers ψ_l are discarded afterwards.

Stage 2: generative fine-tuning. The connector and the receiver's LoRA parameters are then trained with the conditional objective of Section 3 on the full answer string, including any reasoning steps in the training response. The two stages separate learning the interface from adapting the language model's wording. Table A2(b) lists the Stage 2 generation settings for DNA→text.

When the warm-up is defined. Eqn. (3) requires a closed and finite label set, one label per example, and a label that is predictable from the question and the conditioning input alone, because the receiver does not see the answer at this stage. Table A3 records which settings satisfy this.

Table A3: **The warm-up is used exactly where the target is a closed label set.**

Setting	Target of the task	Closed label set	Warm-up
DNA: VEC, VE-NonSNV	pathogenic / benign	yes (2)	yes
DNA: KEGG	disease label	yes (33)	yes
Single-cell	cell type	yes (482; 50 in Stage 1)	yes
Protein → text	free-text answer	no	no
Text → protein	amino-acid sequence	no	no

The two generation settings have no class label y: their targets are a free-text answer and a variable-length amino-acid sequence. A contrastive alignment stage, which scores pooled contexts against paired sequence or text embeddings, could play the same role there, and we leave it to future work.

Warm-up ablation. Table A4 removes Stage 1 from the middle-extraction configuration of Table 1 and trains with the generative objective alone; the legacy protocol, seed 23, and all other settings are unchanged. Removing the warm-up lowers F1 in 8 of the 12 configurations and by 0.8 points on average across all 12. The drop is consistent on VE-NonSNV (1.5–6.7 points), whereas on VEC with Evo2 the model trained without the warm-up scores higher. The warm-up therefore helps on average but does not explain the gap to BioReason, which is trained without it: on VEC, where legacy-protocol BioReason runs are available (Table A10), BioStitch without the warm-up still leads by 7.8–18.8 F1.

D VEC PROMPT REWRITING

Removing label-correlated gene-name cues. The original VEC prompts contain gene names for pathogenic examples, such as CLN8, GDAP1, and RECQL4, but omit them for benign examples. The presence of a gene name therefore provides a shortcut to the label without requiring the DNA sequence. We rewrite the prompts to remove these names while retaining chromosome-level context, and train, validate, and evaluate both systems on the rewritten prompts. Table A5 gives examples; the DNA sequences, answer labels, and split membership are unchanged. VE-NonSNV prompts name

Table A4: **Warm-up ablation** (legacy protocol, seed 23, middle source layer). "Full" is the two-stage schedule of Table 1; "w/o warm-up" trains with the generative objective only. F1 is positive-class F1 for VEC and VE-NonSNV and weighted F1 for KEGG, as in Table 1.

		F1			Accuracy		
Dataset	Encoder + LLM	Full	w/o warm-up	Δ	Full	w/o warm-up	Δ
VEC	NT + Qwen3-1.7B	70.84	70.50	−0.34	67.48	66.26	−1.22
VEC	NT + Qwen3-4B	71.91	68.82	−3.09	67.56	67.15	−0.41
VEC	Evo2 + Qwen3-1.7B	68.43	74.38	+5.95	67.15	71.29	+4.14
VEC	Evo2 + Qwen3-4B	68.94	71.06	+2.12	68.29	68.29	0.00
VE-NonSNV	NT + Qwen3-1.7B	79.96	75.78	−4.18	76.17	71.59	−4.58
VE-NonSNV	NT + Qwen3-4B	78.92	77.09	−1.83	75.03	73.31	−1.72
VE-NonSNV	Evo2 + Qwen3-1.7B	89.56	88.04	−1.52	87.40	86.37	−1.03
VE-NonSNV	Evo2 + Qwen3-4B	91.46	84.77	−6.69	89.92	83.05	−6.87
KEGG	NT + Qwen3-1.7B	90.98	90.38	−0.60	91.72	91.03	−0.69
KEGG	NT + Qwen3-4B	94.73	94.90	+0.17	95.17	95.86	+0.69
KEGG	Evo2 + Qwen3-1.7B	91.15	90.99	−0.16	91.38	90.34	−1.04
KEGG	Evo2 + Qwen3-4B	94.43	95.24	+0.81	95.17	95.52	+0.35

Table A5: **Representative VEC prompt rewrites**, drawn from the VEC test split and aligned by sample ID (IDs abbreviated from `Task1_test_*`). Only the natural-language prompt changes: the reference and variant DNA sequences (4096 bp each), answer label, and split membership are unchanged.

Sample ID	Label / cue	Original prompt	Processed prompt
`test_0`	Pathogenic; CLN8	Mutation context: Chromosome 8, Gene CLN8 (CLN8 transmembrane ER and ERGIC protein). Determine if this variant is likely to be benign or pathogenic. Mention the disease if applicable.	This sequence variant lies on Chromosome 8. Is it clinically significant, and what condition might it cause if any?
`test_344`	Pathogenic; GDAP1	The following genetic variant occurs in GDAP1 (ganglioside induced differentiation associated protein 1) on Chromosome 8. Classify its clinical effect—pathogenic or benign—and list any associated condition.	This alteration on Chromosome 8 may affect genome function. Does it lead to a disease or is it benign?
`test_675`	Pathogenic; RECQL4	Here is a genetic alteration in RECQL4 (RecQ like helicase 4) on Chromosome 8. Based on the data, is it a benign variant or a cause of disease?	This variant is present on Chromosome 8. Is the change likely to result in a pathogenic outcome?
`test_676`	Benign; no cue	This variant is located on Chromosome 8. Evaluate its biological effect and specify any disease association.	Chromosome 8 is altered by this variant. Does this mutation result in a disease or is it benign?
`test_953`	Benign; no cue	This mutation is located on Chromosome 8. Is it associated with a disease or is it a benign polymorphism?	This mutation on Chromosome 8 may be significant. Is it associated with any clinical condition, or is it benign?
`test_1232`	Benign; no cue	This mutation occurs on Chromosome 8. Does this change lead to a known medical condition, or is it benign?	With a mutation on Chromosome 8, classify this variant as benign or pathogenic. Include the disease if it is pathogenic.

the gene for both classes, so the presence of a gene name is not label-correlated there and we keep them unchanged; gene identity can still carry prior information, which is equally available to both systems.

E DETAILED VARIANT-EFFECT CLASSIFICATION RESULTS

Table A6: **Detailed metrics and confusion counts for Table 2.** Results use the strict held-out protocol, NT-v2-500M, and seed 23. Bold identifies the final-epoch BioStitch metrics used in the main table, rather than the highest value across checkpoints.

(a) Classification metrics

Dataset	LLM	Method	Checkpoint	Acc	P	R	F1	BAcc	MCC
VEC	1.7B	BioStitch	Best = final	**66.59**	**74.35**	**59.62**	**66.17**	**67.33**	**0.35**
VEC	1.7B	BioReason	Best = final	58.88	67.21	48.82	56.56	59.96	0.20
VEC	4B	BioStitch	Best = final	**64.96**	**73.19**	**56.95**	**64.06**	**65.82**	**0.32**
VEC	4B	BioReason	Best = final	58.15	70.73	40.38	51.41	60.05	0.22
VE-NonSNV	1.7B	BioStitch	Best	69.87	70.77	84.70	77.11	66.21	0.35
VE-NonSNV	1.7B	BioStitch	Final	**72.97**	**77.54**	**77.25**	**77.39**	**71.91**	**0.44**
VE-NonSNV	1.7B	BioReason	Best = final	55.21	63.47	59.46	61.40	54.16	0.08
VE-NonSNV	4B	BioStitch	Best	73.54	78.97	76.10	77.51	72.91	0.45
VE-NonSNV	4B	BioStitch	Final	**74.34**	**79.84**	**76.48**	**78.12**	**73.81**	**0.47**
VE-NonSNV	4B	BioReason	Best = final	55.90	70.41	45.51	55.28	58.47	0.17

(b) Confusion counts

Dataset	LLM	Method	Checkpoint	TP	TN	FP	FN
VEC	1.7B	BioStitch	Best = final	403	418	139	273
VEC	1.7B	BioReason	Best = final	330	396	161	346
VEC	4B	BioStitch	Best = final	385	416	141	291
VEC	4B	BioReason	Best = final	273	444	113	403
VE-NonSNV	1.7B	BioStitch	Best	443	167	183	80
VE-NonSNV	1.7B	BioStitch	Final	404	233	117	119
VE-NonSNV	1.7B	BioReason	Best = final	311	171	179	212
VE-NonSNV	4B	BioStitch	Best	398	244	106	125
VE-NonSNV	4B	BioStitch	Final	400	249	101	123
VE-NonSNV	4B	BioReason	Best = final	238	250	100	285

Best denotes validation-selected checkpoints; final denotes the last epoch. The main table reports final-epoch results throughout. For VEC, training and validation overlap at the variant level as described in Table A1. BAcc is balanced accuracy and MCC the Matthews correlation coefficient, both computed from the counts in panel (b).

Table A6 resolves the headline Accuracy and F1 of Table 2 into precision, recall and the four confusion cells, for both systems under the same strict held-out protocol.

The largest gains are in recall. Relative to BioReason, precision increases by $+7.14$, $+2.46$, $+14.07$, and $+9.43$ points across the four configurations, while recall increases by $+10.80$, $+16.57$, $+17.79$, and $+30.97$ points. For VEC with the 4B language model, BioReason has precision 70.73 versus 73.19 for BioStitch, but recall is 40.38%: it identifies 273 pathogenic variants and misses 403. On VE-NonSNV, its precision is 70.41 and recall is 45.51, with 285 of 523 positive cases missed. BioStitch increases true positives from 238 to 400 while false positives change from 100 to 101.

Imbalance-robust metrics. Both test sets contain more pathogenic than benign variants (55% on VEC and 60% on VE-NonSNV), which inflates positive-class F1: a predictor that always answers "pathogenic" already reaches F1 of 70.8 and 74.9. We keep positive-class F1 for comparability with BioReason and add two metrics that this imbalance does not inflate, balanced accuracy and the Matthews correlation coefficient (MCC), computed from the counts in Table A6(b). The always-pathogenic predictor scores 50 and 0 on them, whereas BioStitch leads BioReason on both in all four

configurations, by 5.8–17.8 points of balanced accuracy and 0.10–0.36 in MCC. For the accuracy gains, unpaired binomial 95% intervals are $[3.9, 11.5]$, $[3.0, 10.6]$, $[13.3, 22.2]$, and $[14.1, 22.8]$ points.

F KEGG DISEASE-PATHWAY REASONING

Table A7: **KEGG disease-pathway reasoning under the legacy protocol.** Superscripts give the change in percentage points relative to BioReason (↑ / ↓, = tie).

Encoder + LLM	Method	Acc	Macro-F1	Weighted-F1	Micro-F1
NT + Qwen3-1.7B	BioReason	87.93	71.19	83.21	79.44
	BioStitch	**91.72**$^{\uparrow 3.8}$	**78.92**$^{\uparrow 7.7}$	**90.98**$^{\uparrow 7.8}$	**90.78**$^{\uparrow 11.3}$
NT + Qwen3-4B	BioReason	92.76	73.15	89.20	86.36
	BioStitch	**95.17**$^{\uparrow 2.4}$	**88.55**$^{\uparrow 15.4}$	**94.73**$^{\uparrow 5.5}$	**94.68**$^{\uparrow 8.3}$
Evo2 + Qwen3-1.7B	BioReason	85.86	64.52	83.54	81.91
	BioStitch	**91.38**$^{\uparrow 5.5}$	**81.97**$^{\uparrow 17.5}$	**91.15**$^{\uparrow 7.6}$	**91.38**$^{\uparrow 9.5}$
Evo2 + Qwen3-4B	BioReason	95.17	69.63	87.85	83.01
	BioStitch	95.17$^{=}$	**82.71**$^{\uparrow 13.1}$	**94.43**$^{\uparrow 6.6}$	**94.68**$^{\uparrow 11.7}$

Table A7 reports disease classification on KEGG under the legacy protocol, where the union of the original validation and test sets is used for both validation and final evaluation. We rescore both systems' generated outputs with the same one-vs-rest substring evaluator and report accuracy, macro-F1, weighted-F1, and micro-F1. BioStitch improves every F1 entry: macro-F1 increases by $+7.7$ to $+17.5$ points, weighted-F1 by $+5.5$ to $+7.8$, and micro-F1 by $+8.3$ to $+11.7$. Accuracy improves in three configurations and ties for Evo2 + Qwen3-4B. The published BioReason macro-F1 for NT + Qwen3-1.7B is 81.46, whereas rescoring its outputs with this evaluator yields 71.19. We report the common-evaluator result for comparability.

Table A8: **Answer commitment on the KEGG test set** ($n = 290$, NT + Qwen3-1.7B). Loose substring scoring credits an output if it contains the gold disease. The stricter alternative requires a single committed disease.

	BioStitch	BioReason
Answers naming more than one disease	**1 / 290**	136 / 290 (47%)
of which credited by substring match	1	118
Accuracy (loose substring; convention of Table A7)	91.72	87.93
Accuracy (single committed disease required)	**91.38**	**47.24**

Answer commitment. We also evaluate whether a generated answer commits to a single disease (Table A8). BioStitch names multiple diseases in 1 of 290 answers, compared with 136 (47%) for BioReason; the substring evaluator credits 118 of the latter answers. Requiring a single committed disease changes BioStitch accuracy from 91.72 to 91.38, and BioReason accuracy from 87.93 to 47.24. The accuracy gap therefore depends on the scoring rule: 3.79 points under substring matching and 44.14 points when a single committed disease is required.

G SINGLE-CELL ANNOTATION PROTOCOL

Section 4.5 reports zero-shot cell-type annotation on TS-20 (The Tabula Sapiens Consortium, 2022) and AIDA-9 (Kock et al., 2025) under the external benchmark protocol of CellWhisperer (Schaefer et al., 2025). Both label sets are closed: for each cell we score every candidate cell-type description under teacher forcing and take the highest-scoring label as the prediction, so the reported accuracy and macro-F1 are closed-set scores rather than free generation. AUROC is read from the same

per-label scores. A frozen Geneformer (Theodoris et al., 2023) supplies keys and values, and the Qwen3-1.7B receiver is adapted with LoRA. The adapters add $17{,}432{,}576$ trainable parameters (1.003% of the receiver), and the connector is trained in addition. Training follows the two-stage schedule of Appendix C, with a 50-class warm-up subset followed by generative fine-tuning over the full 482-class label set.

Table 4 compares a generative connector with a discriminative retrieval model: the two systems differ in training data, training objective, and how a label is scored at test time. Closed-set scoring also admits several normalizations, and accuracy and macro-F1 depend more on this choice than AUROC does.

H NON-QWEN TEXT BACKBONE

We replace Qwen3-1.7B with Llama-3.2-1B-Instruct to test whether the connector also works with another language-model family (Table A9). The change affects the tokenizer, hidden width, parameter count, and pretrained model simultaneously. F1 remains within one point on both datasets, while accuracy changes by $+2.92$ points on VEC and -2.63 on VE-NonSNV. Both versions exceed the listed BioReason reference on these metrics under the legacy protocol. The connector therefore transfers to a non-Qwen receiver.

Table A9: **Replacing the text backbone with a different, smaller model family.** Replacing Qwen3-1.7B with Llama-3.2-1B-Instruct changes the language-model family, tokenizer, width, and size together; encoder, fusion mode, legacy protocol, schedule, and test sets are unchanged. Bold marks the highest value in each row. BioReason values are local legacy-protocol runs on VEC and the published values (Fallahpour et al., 2025) on VE-NonSNV.

Task	Metric	BioReason	**BioStitch**		Δ
		Qwen3-1.7B	Qwen3-1.7B	Llama-3.2-1B	
VEC (n=1,233)	F1	51.69	**70.84**	70.00	-0.84
	Acc	53.77	67.48	**70.40**	$+2.92$
VE-NonSNV (n=873)	F1	76.93	**79.96**	79.75	-0.21
	Acc	72.82	**76.17**	73.54	-2.63

I SEED ROBUSTNESS

Table A10: **Seed robustness on VEC** (*legacy protocol*; F1 is binary positive-class throughout). BioReason values are single local runs.

Configuration		Seed 23	Seed 42	Seed 7	Mean $\pm$ std	BioReason	Δ mean
NT + Qwen3-1.7B	Acc	67.48	65.37	63.67	65.51 ± 1.56	53.77	$+11.74$
	F1	70.84	68.21	67.82	68.96 ± 1.34	51.69	$+17.27$
NT + Qwen3-4B	Acc	67.56	67.96	66.67	67.40 ± 0.54	52.47	$+14.93$
	F1	71.91	71.36	72.02	71.76 ± 0.29	54.15	$+17.61$
Evo2 + Qwen3-1.7B	Acc	67.15	68.37	—	67.76 ± 0.61	67.96	-0.20
	F1	68.43	71.15	—	69.79 ± 1.36	66.27	$+3.52$
Evo2 + Qwen3-4B	Acc	68.29	—	69.02	68.66 ± 0.37	67.07	$+1.59$
	F1	68.94	—	74.33	71.63 ± 2.70	63.29	$+8.34$

"—" marks a seed that was not run. The seed runs use 4-GPU DDP, whereas the main-table run used a single GPU; the effective batch size (16) is matched. Standard deviations are population standard deviations over the available runs.

Table A10 reports the available VEC runs for seeds 23, 42, and 7 under the legacy protocol. Both NT configurations have three completed seeds. Their mean F1 exceeds the listed BioReason reference by

$+17.27$ and $+17.61$ points, with standard deviations of 1.34 and 0.29, respectively; the corresponding accuracy differences are $+11.74$ and $+14.93$. The gains persist across these NT runs. The BioReason references are single runs.

Each Evo2 configuration has two completed seeds. The two-seed mean F1 exceeds BioReason by $+3.52$ points for Evo2 + Qwen3-1.7B and by $+8.34$ for Evo2 + Qwen3-4B, with standard deviations of 1.36 and 2.70. The corresponding mean accuracy differences are -0.20 and $+1.59$ points. With two seeds, these variability estimates are less precise than those for NT.

J REASONING-CHAIN EVALUATION

We evaluate generated explanations in addition to disease labels using an anonymized, order-balanced pairwise LLM judge. The recorded judge is Claude Opus 4.8, queried at its default reasoning effort. For each of the 290 KEGG evaluation items, it receives the question, gold answer, and two anonymized traces, with no system names. Two separate calls swap the traces between slots A and B, giving 580 calls. Tables A11 and A12 report the aggregate preferences and failure annotations. The results are judge preferences under the stated rubric, not expert biological assessments.

J.1 JUDGE PROTOCOL AND PROMPT

The judge is Claude Opus 4.8. It receives one JSON object per call containing the question, the gold answer and the two anonymized traces, and returns a per-trace `process_score`, a list of failure-mode tags with verbatim quoted evidence, and a verdict in $\{$A, B, tie$\}$. Figure A1 summarizes the scoring rubric and anti-bias instructions. The rubric includes dataset-specific guidance, so the preferences it produces are specific to this prompt. The anti-bias rules are meant to counter the known tendency of LLM judges to prefer longer and more fluent answers. Because trace length also differs between the two systems, the judge's preferences only indicate roughly which explanations are better; they are not an exact or unbiased score.

Aggregation. A judge call returns a verdict in $\{$A, B, tie$\}$ for one slot order, so each item carries two verdicts. The outcome labels in Table A11 are not returned by the judge; we assign them afterwards by combining an item's two verdicts, which admit exactly four cases: both orders name the same system (a *decisive win* for that system), one order names a system and the other is a tie (a *weak win* for that system), both orders are ties (a *tie*), or the two orders name different systems (an *A/B order split*). A split means that the judge's preference for that item depends on presentation order rather than on the traces. Because the four cases are exhaustive, the counts in Table A11 partition all 290 items.

Table A11: **Anonymized, order-balanced pairwise judging of generated reasoning chains** on all 290 KEGG test items (NT + Qwen3-1.7B). Each item is judged twice with the two traces swapped between the A and B slots (580 calls). The judge sees only the question, the gold answer and two anonymized traces.

Outcome	Count	Share
BioStitch decisive win	**141**	**48.6%**
BioReason decisive win	50	17.2%
BioStitch weak win	32	11.0%
BioReason weak win	20	6.9%
Tie	36	12.4%
A/B order split	11	3.8%

J.2 RESULTS

BioStitch receives 141 decisive wins versus 50 for BioReason (48.6% versus 17.2%). The judge exhibits a slot-A preference ($p = 0.0075$); both systems benefit from it, and BioStitch has more wins in either slot separately.

```
# ROLE
You are NOT a helpful assistant here. You are an adversarial grader. Your
default posture toward a fluent, confident, well-formatted trace is
suspicion, not approval. You do not know which model produced which trace.

# THE STRUCTURAL TRAP OF THIS DATASET
Many distinct variants share the same pathway, and therefore the same
mechanistic chain, but map to DIFFERENT diseases. It follows that a
correct-sounding mechanistic chain carries ZERO information about which
disease is right.

    THE SUBSTITUTION TEST - If I swapped the gold disease for another
    disease reachable through this same pathway, would this trace's
    reasoning read just as convincingly? If YES, the mechanism is
    boilerplate and the disease call is unearned, regardless of whether
    the trace happened to land on the right label.

A trace that reaches the gold answer via an unearned leap is *lucky*,
not *correct*. Say so.

# NAMED FAILURE MODES  (tag each trace with every one that applies)
  FABRICATED_PREMISE      Invents a variant identity AND makes the final
                          disease call depend on it.
  TOPOLOGY_CONTRADICTION  A step contradicts the network expression as
                          given (wrong direction, regulator, substrate).
  BIOLOGICAL_ERROR        A step is factually wrong independent of prompt.
  VACUOUS_HEDGE           A step commits to nothing.
  INCOHERENT_STEP         A step is nonsensical or a non sequitur.
  SELF_CONTRADICTION      Two steps assert incompatible things.

# SCORING   process_score, per trace, independent of the other trace:
  0  Contradicts the topology, or truncated, or names the wrong gene.
  1  Multiple biological errors; the chain does not hold together.
  2  Mechanistically plausible but generic; fails the Substitution Test.
  3  Sound chain, generic, but one non-boilerplate observation.
  4  Sound and at least partly variant-specific.
  5  Sound, specific, and survives the Substitution Test.
Score the PROCESS. A trace with a wrong final answer can still score 3;
a trace with the right final answer can still score 1.

# ANTI-BIAS RULES - these override your instincts
  - Length is not quality. In this dataset the longer trace is the less
    accurate model's roughly 70% of the time.
  - Fluency and confidence are not quality. The most polished trace in a
    pair is frequently the confidently-wrong one.
  - Specificity is not grounding.
  - Never break a genuine tie by picking the longer, the first, or the
    prettier one. Output `tie`.
  - Slot A and slot B are randomized. Position carries no information.
```

Figure A1: **Scoring-relevant excerpt of the judge system prompt.** Wording is condensed from the full prompt, and some rules and four failure modes are omitted.

The failure annotations in Table A12 show fewer incoherent steps (-21.9 points), vacuous hedges (-12.1), and topology contradictions (-3.5) for BioStitch. However, biological errors remain at 38.3%, and fabricated premises are *more* frequent for BioStitch: 15.2% versus 13.3% ($+1.9$ points). The judge therefore prefers BioStitch overall under this rubric.

K Full text-to-protein evaluation panel

Table A13 gives the full PDFBench metric panel for the NatureLM group of Table 5. The panel includes text–sequence alignment and retrieval scores, sequence likelihood and repetition, diversity, and predicted structural confidence. The three left columns use a common evaluation procedure and

Table A12: **Failure types annotated by the judge** over the same 580 calls as Table A11. Categories co-occur: one trace can carry several labels, so columns do not sum to 100%.

Failure type	BioStitch	BioReason	Absolute change
INCOHERENT_STEP	**15.2%**	37.1%	**−21.9** pp
VACUOUS_HEDGE	**24.3%**	36.4%	**−12.1** pp
BIOLOGICAL_ERROR	**38.3%**	51.7%	**−13.4** pp
FABRICATED_PREMISE	15.2%	**13.3%**	+1.9 pp
TOPOLOGY_CONTRADICTION	**5.5%**	9.0%	**−3.5** pp
SELF_CONTRADICTION	**15.0%**	21.2%	**−6.2** pp

common per-metric denominators. Pinal and ProDVa are external references from the PDFBench leaderboard, evaluated with their own inference pipelines on all 5,876 prompts.

Comparison with dedicated generation systems. Pinal and ProDVa achieve higher values on several alignment and predicted structural-confidence metrics. Pinal has pLDDT 75.25 and PAE 10.96, compared with 57.33 and 18.42 for BioStitch. Its reported main configuration has roughly 15.5B parameters and is pretrained on 1.7B function–sequence pairs. ProDVa uses a dynamic vocabulary of protein fragments; in a separate instrumented rerun, 90.4% of generated residues came from fragment-generation steps. Its leaderboard scores include pLDDT 76.84, PAE 8.67, repetition 1.92, and reference identity 21.48.

The NatureLM configuration of BioStitch trains 63M connector parameters on 180,594 pairs for three epochs with both backbones frozen. It has ProTrek 9.02, compared with 17.50 and 17.40 for Pinal and ProDVa, and Retrieval Normal@4 46.18, compared with 71.69 and 77.52. The systems differ in architecture, backbone size, training data, and inference procedure.

Tradeoffs among the three common-evaluation columns. BioStitch has the highest score on six of the twelve metrics in these columns, including ProTrek (9.02 versus 6.44 and 6.61), Retrieval Normal@4 (46.18 versus 38.77 and 39.97), and GT-Identity (14.15 versus 9.37 and 7.93), as well as ESM2-F1, EvoLlama, and GT-TMScore. It does not lead on perplexity, repetition, sequence diversity, Retrieval Hard@4, pLDDT, or PAE, so the gains in conditioning coexist with tradeoffs in sequence and predicted structural confidence. Retrieval Hard@4 stays near the 25% chance level across the whole panel (24.26 for BioStitch, 27.8–29.5 for the dedicated systems), which shows that distinguishing the closest negative descriptions remains difficult in this evaluation.

Table A13: **Full PDFBench metric panel for the NatureLM group of Table 5.** The three left columns use common evaluation settings and per-metric denominators; bold marks their best value per metric. The two right columns are external references from the PDFBench leaderboard, using their own inference pipelines and all 5,876 prompts.

Metric	**BioStitch**	Direct + LoRA	Mol-Inst-7B	*Pinal*	*ProDVa*
ProGen2 PPL ↓	6.134	**5.506**	9.394	*5.81*	*7.63*
Repeat ↓	21.221	19.480	**19.161**	*12.83*	*1.92*
ESM2-F1 ↑	**74.986**	72.218	72.217	*76.63*	*75.23*
GT-Identity ↑	**14.145**	9.374	7.926	*18.65*	*21.48*
Sequence diversity ↑	85.523	91.768	**93.637**	*82.96*	*83.29*
ProTrek ↑	**9.024**	6.444	6.608	*17.50*	*17.40*
EvoLlama ↑	**53.871**	50.440	50.423	*53.40*	*51.19*
Retrieval Normal@4 ↑	**46.183**	38.766	39.966	*71.69*	*77.52*
Retrieval Hard@4 ↑	24.255	**24.345**	24.332	*29.51*	*27.84*
pLDDT ↑	57.332	**62.458**	50.844	*75.25*	*76.84*
PAE ↓	18.421	**16.290**	18.868	*10.96*	*8.67*
GT-TMScore ↑	**20.407**	18.645	19.333	*23.75*	*20.03*

L COST IN THE TEXT-TO-PROTEIN DIRECTION

Table A14: **Text-to-protein efficiency.** Both implementations use frozen Qwen3-1.7B and ProGen2-base backbones on one A100-80GB. Ratios divide the prefix measurement by the cross-attention measurement unless stated otherwise.

Measurement	Prefix *(K=384, unpooled)*	**BioStitch** *(cross-attention)*	Ratio
Training			
Peak device memory (`nvidia-smi`)	25,683 MiB	**16,581 MiB**	$\mathbf{1.55}\times$
Peak tensor allocation (`torch`)	22,508 MiB	**14,442 MiB**	$1.56\times$
Time per sample	0.154 s	**0.095 s**	$\mathbf{1.62}\times$
Trainable parameters	**3.15 M**	40.16 M	BioStitch $12.8\times$ more
Inference			
Decoder prompt length	386 tok	**2 tok**	$193\times$
Decoder KV cache at prompt	61.07 MB	**0.32 MB**	$193\times$
Cached condition residency	**0**	1.50 MB	—
Prefill latency per sample	117.3 ms	**112.6 ms**	$1.04\times$
Peak memory (torch reserved)	11,980 MiB	**7,432 MiB**	$1.61\times$
Decode latency per token	**36.55 ms**	37.22 ms	BioStitch 1.8% slower
End-to-end latency, 64 tokens	**2,420 ms**	2,457 ms	$\approx$ equal

Table A14 measures the text→protein direction with both implementations on one A100-80GB. It complements the DNA→text measurements in Table 6 by exchanging the conditioning and generation roles.

BioStitch uses $16{,}581$ MiB of peak training memory versus $25{,}683$ MiB for unpooled prefix conditioning ($1.55\times$ lower), and 0.095 s per sample versus 0.154 s ($1.62\times$ faster). It nevertheless trains more parameters: 40.16 M versus 3.15 M, a $12.8\times$ increase. The lower memory use is therefore consistent with savings in sequence-dependent activations rather than a smaller trainable parameter count.

At inference, prefix conditioning produces a 386-token decoder prompt and 61.07 MB of prompt KV cache, while cross-attention leaves 2 decoder tokens and 0.32 MB of decoder prompt KV cache. BioStitch also retains 384×2048 conditioning states, which add 1.50 MB. Including these states gives 1.82 MB versus 61.07 MB, a $33.6\times$ reduction in the reported prompt-state total; the $193\times$ figure applies only to decoder prompt KV cache.

Prefill latency is similar (112.6 versus 117.3 ms, $1.04\times$), rather than showing the $2.13\times$ difference measured in the DNA direction. The prompt lengths differ substantially across tasks: the DNA setting has $L_{\text{text}} \ll L_{\text{bio}}$, whereas the text-conditioned setting places an instruction on the conditioning side. Per-token decoding is 1.8% slower for BioStitch, and end-to-end latency is similar ($2{,}457$ versus $2{,}420$ ms for 64 tokens). Training memory and training time therefore improve in both evaluated directions, while the inference-latency benefit depends on where the long sequence sits.

M SOURCE-DEPTH ABLATION IN ACCURACY

Table A15 gives the accuracy counterpart of Table 1, moved here for space; runs, protocol, and layer indices are identical.

Table A15: **Source-depth ablation in accuracy.** Same runs as Table 1: seed 23, one run per cell, legacy protocol (validation reuses test). Bold marks the best depth within each dataset–model block.

	VEC			VE-NonSNV			KEGG		
Encoder + LLM	Shallow	Middle	Deep	Shallow	Middle	Deep	Shallow	Middle	Deep
NT + Qwen3-1.7B	64.31	**67.48**	65.86	67.47	**76.17**	70.79	88.97	**91.72**	90.69
NT + Qwen3-4B	63.42	**67.56**	63.50	69.76	75.03	**75.37**	94.14	**95.17**	94.83
Evo2 + Qwen3-1.7B	67.88	67.15	**69.51**	78.01	**87.40**	86.25	90.69	91.38	**92.07**
Evo2 + Qwen3-4B	66.50	**68.29**	67.64	82.13	**89.92**	86.94	93.45	**95.17**	94.14
mean over 12 blocks		SHALLOW 77.2		<	DEEP 79.8	<		**MIDDLE 81.0**	