WHAT DO TRANSCRIPTOME MODELS LEARN? FROM EXPRESSION PROBING TO BIOLOGICAL PREDICTION AND RNA QUESTION ANSWERING

Anonymous authors
Paper under double-blind review

ABSTRACT

Transcriptome foundation models are increasingly used for biological prediction and connected to language models to answer questions about transcriptomic data. However, differences in both pretraining methods and training data make it difficult to determine which approach works best for a given use case. To address this, we train six single-cell and four bulk models from scratch on one shared corpus per modality, following literature-inspired pretraining recipes. Our evaluation addresses three questions. *How does pretraining affect expression probing?* The recipe with the highest score after pretraining is not always the one that improves the most, and for some recipes linear probes already achieve high scores on representations from untrained models. *How do the recipes perform on biological tasks?* Pretraining improves classification over untrained models for every recipe, but its effect on perturbation prediction is inconsistent. Moreover, recipes that score higher on expression probes do not necessarily perform better on biological tasks. Early in pretraining, LogME transferability scores already correlate positively with recipe performance on all five biological tasks, without training any downstream predictor. *Do the differences observed in probing and biological prediction extend to RNA question answering?* We connect the frozen pretrained models to Qwen and evaluate the probability assigned to reference answers for released CellWhisperer questions. Among single-cell models, cell-type classification rankings agree most closely with reference-answer likelihood rankings on cell-identity questions, whereas expression-value probe rankings show no such correspondence. Cell-type LogME rankings also agree between RNA and Qwen representations with an untrained adapter, suggesting that, as in vision–language models, encoder strengths carry over into the language model.

1 INTRODUCTION

Transcriptome foundation models are pretrained on large collections of RNA-seq profiles to learn representations that transfer to downstream tasks such as cell-type classification, disease classification, and perturbation prediction (Theodoris et al., 2023, Gélard et al., 2025). More recently, these models have been coupled with large language models (LLMs), while other approaches feed gene expression directly into LLMs, enabling natural-language interaction with transcriptomic data (Zhao et al., 2024, Levine et al., 2024, Schaefer et al., 2025). Existing transcriptome models, however, differ in how expression is encoded as input, for example as expression-ranked gene sequences, discretized expression bins, or continuous values, and in their pretraining objectives, including masked gene-identity prediction, expression-presence prediction, and expression-value prediction (Theodoris et al., 2023, Rosen et al., 2023, Cui et al., 2024, Zeng et al., 2025, Kang et al., 2026). These choices may influence what information the learned representation retains and how well it supports different downstream tasks.

Identifying which design suits a given application is difficult with existing models. Released models differ not only in their pretraining design but also in training data and model scale, so performance differences observed in benchmarks (Kedzierska et al., 2025, Bendidi et al., 2024) cannot be attributed to the design alone. To address this, recent work has compared self-supervised methods and model configurations for single-cell data under controlled training, mainly in terms of downstream

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.

performance (Ovcharenko et al., 2025, De Waele et al., 2026). However, it remains unclear how well probes predict gene presence, expression order, and expression values from each model's representations, how their scores relate to downstream performance, and whether such findings extend to bulk RNA-seq and to language-based applications.

In this work, we compare six single-cell and four bulk pretraining recipes, each specifying preprocessing, input encoding, masking, the pretraining objective, and training settings (Section 3). All models are trained from scratch on one shared corpus per modality, with matched depth and hidden size. Untrained models serve as baselines. We fit linear probes on frozen representations to predict *presence* (whether a gene has positive measured expression), *order* (the relative ordering of expressed genes within a sample), and *value* (the expression value of an expressed gene on a fixed normalized scale), regardless of the pretraining target. We evaluate biological performance on cell-type and cancer-type classification and genetic and chemical perturbation prediction, and relate these results to expression probe scores. We also examine whether downstream performance can be estimated without training the downstream predictor, using transferability scores computed with LogME. Finally, we connect the same frozen models to an LLM for RNA question answering (RNA QA). Using CellWhisperer conversations, we ask whether the differences found through expression probing and biological prediction also appear in answers about expression, cell/sample identity, and biological function or state. These evaluations address the following research questions:

- **RQ1.** How does pretraining change probe performance on gene presence, order, and expression value across recipes?
- **RQ2.** Does pretraining improve biological prediction, how does performance relate to expression probe scores, and can downstream performance be estimated without training the downstream predictor?
- **RQ3.** Do stronger expression probing or biological prediction results translate into better RNA QA performance, and how does this relationship vary across question types?

Our main findings are as follows. **RQ1:** Strong expression-probe scores can arise from untrained models, and the highest pretrained score need not reflect the largest pretraining gain (Section 4). **RQ2:** Expression-probe rankings do not consistently agree with biological performance rankings. Simple expression baselines outperform all pretrained models on both bulk tasks, while early LogME scores correlate positively with performance on all five biological tasks (Section 5). **RQ3:** Among single-cell models, cell-type classification rankings agree most closely with reference-answer likelihood rankings on cell-identity questions, but do not consistently agree across other question types. Expression-value probing and bulk classification do not show a consistent positive correspondence with RNA QA performance (Section 6).

2 Related Work

Transcriptome pretraining recipes. Transcriptome foundation models differ in how they encode gene expression and what they learn to predict. Geneformer predicts masked gene identities from expression-ranked sequences, whereas UCE focuses on expression presence (Theodoris et al., 2023, Rosen et al., 2023). Other models learn from discretized or continuous expression values, including scBERT, scGPT, and CellFM, while tGPT uses autoregressive next-gene prediction over expression-ranked genes (Yang et al., 2022, Cui et al., 2024, Zeng et al., 2025, Shen et al., 2023). Similar variation exists in bulk RNA-seq, where BulkRNABert, GeneRAIN, and BulkFormer use expression-bin, gene-identity, autoregressive, or continuous-value prediction objectives (Gélard et al., 2025, Su et al., 2025, Kang et al., 2026). These models differ not only in their pretraining objectives but also in their input encodings and training settings. We therefore compare these choices together as pretraining recipes.

Evaluating transcriptome representations. The benefits of transcriptome pretraining have been studied through biological tasks such as cell-type classification and perturbation prediction (Kedzierska et al., 2025, Bendidi et al., 2024, Liu et al., 2026). Controlled comparisons further examine how self-supervised methods, model configurations, and pretraining data affect downstream performance (Ovcharenko et al., 2025, DenAdel et al., 2026, De Waele et al., 2026). Probing tests what information can be predicted from frozen representations by training a separate predictor (Alain & Bengio,

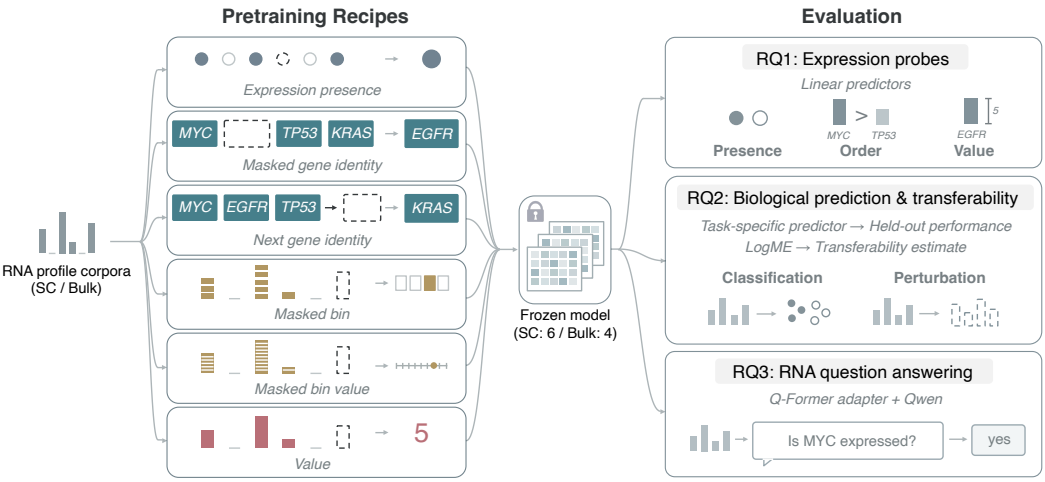

Figure 1: Overview of our study. Models are pretrained on one shared corpus per modality and kept frozen for three evaluations: (1) linear probing of gene presence, order, and expression value, (2) biological prediction, covering cell-type and cancer-type classification and genetic and chemical perturbation prediction, with performance compared with task-aware transferability scores (LogME), and (3) RNA QA, where a Q-Former connects the RNA encoder to Qwen3-1.7B to evaluate reference-answer likelihood for expression interpretation, cell/sample identity, and biological function or state. SC and Bulk denote single-cell RNA-seq and bulk RNA-seq, respectively.

2016, Belinkov, 2022). We combine these evaluations for single-cell and bulk models pretrained on shared data, examining how expression probe scores relate to biological performance.

Transcriptome–language models. Recent transcriptome–language models have enabled an increasingly broad range of tasks, including cell-type annotation, gene expression profile generation, transcriptome interpretation, and biological question answering (Zhao et al., 2024, Levine et al., 2024, Schaefer et al., 2025). By incorporating language, these models can draw on textual biological knowledge and extend transcriptome analysis beyond fixed-label prediction to free-text generation and question answering. However, it remains unclear how different RNA representations affect performance when both pretraining data and the language-model setup are held fixed. We therefore compare multiple transcriptome encoders pretrained on shared corpora using the same language-model backbone and adapter architecture to examine how encoder choice affects RNA QA.

3 Experimental Design

Pretraining data and recipes. We pretrain single-cell and bulk models separately on 20 million human cells from the CZ CELLxGENE Census (Chan Zuckerberg Initiative, 2025) and approximately 575,000 bulk RNA-seq profiles from PreBULK (Kang et al., 2026). Within each modality, all recipes use the same data and a common vocabulary of 20,053 genes. We compare six single-cell and four bulk recipes inspired by existing transcriptome models and name each by its modality (SC or Bulk) and prediction target: expression presence, meaning nonzero measured expression (Presence), masked gene identity (Identity), next-gene identity (GPT), expression-bin categories (Bin-Class), numerical expression-bin values (Bin-Value), or continuous expression values (Value). Table 1 lists the recipes. SC Value and Bulk Value use $\ln(1 + \mathrm{CP10K})$ and $\ln(1 + \mathrm{TPM})$, respectively, for continuous expression inputs and targets.[1] Bulk Presence is omitted because bulk RNA-seq is less sparse, which makes presence prediction less informative.

Architecture and training. We match model depth and width across recipes at eight layers with hidden size 512, and vary the components that define a recipe: its preprocessing, masking scheme,

[1] CP10K denotes counts scaled to sum to 10,000 per cell, and TPM denotes transcripts per million.

Table 1: Input encodings, prediction targets, losses, and reference models for the ten recipes. Every recipe embeds gene identities, and the term after the plus sign is the extra embedding each gene token carries, so SC PRESENCE sees expression only through which genes are in its input. Rank is a gene's position after sorting by expression, value is its continuous expression, and numbers in parentheses count expression bins including one for zero. BIN-CLASS predicts the bin as a category, whereas BIN-VALUE regresses its numeric value. CE, BCE, and MSE denote cross-entropy, binary cross-entropy, and mean squared error. Implementation details are provided in Appendix A.

Recipe	Input encoding	Prediction target	Loss	Reference model
Single-cell RNA-seq				
SC PRESENCE	Gene ID only	Expression presence	BCE	UCE (Rosen et al., 2023)
SC IDENTITY	Gene ID + rank	Masked gene identity	CE	Geneformer (Theodoris et al., 2023)
SC GPT	Gene ID + rank	Next gene identity	CE	tGPT (Shen et al., 2023)
SC BIN-CLASS	Gene ID + bin (6)	Masked bin	CE	scBERT (Yang et al., 2022)
SC BIN-VALUE	Gene ID + bin (51)	Masked bin value	MSE	scGPT (Cui et al., 2024)
SC VALUE	Gene ID + value	Masked continuous value	MSE	CellFM (Zeng et al., 2025)
Bulk RNA-seq				
BULK IDENTITY	Gene ID + rank	Masked gene identity	CE	GeneRAIN (Su et al., 2025)
BULK GPT	Gene ID + rank	Next gene identity	CE	GeneRAIN (Su et al., 2025)
BULK BIN-CLASS	Gene ID + bin (65)	Masked bin	CE	BulkRNABert (Gélard et al., 2025)
BULK VALUE	Gene ID + value	Masked continuous value	MSE	BulkFormer (Kang et al., 2026)

input gene selection, and prediction head, each set to stay close to the source model (Table 1). Single-cell recipes read up to 2,048 genes with Transformers (Vaswani et al., 2017). In bulk RNA-seq, BULK IDENTITY and BULK GPT likewise read the top 2,048 genes, as in GeneRAIN (Su et al., 2025), whereas BULK BIN-CLASS and BULK VALUE read the full measured gene panel, as BulkRNABert (Gélard et al., 2025) and BulkFormer (Kang et al., 2026) do. For these two recipes, we use Performers (Choromanski et al., 2021), as in BulkFormer, to keep attention over about 20,000 genes tractable. GPT recipes use causal attention and the others use bidirectional attention. Each recipe is pretrained for two epochs on single-cell data (312,500 optimizer steps per epoch) or twenty epochs on bulk data (8,976 steps per epoch), with an effective batch size of 64 RNA profiles and seeds 7, 17, and 27. Pretraining runs were executed on two nodes equipped with NVIDIA H100 GPUs. Appendix A gives full implementation details.

Evaluation. As summarized in Figure 1, we evaluate each pretrained model through linear probing, biological prediction, and RNA QA, keeping the transcriptome model frozen throughout. We select the checkpoint with the lowest validation loss from each training run. Probes predict gene presence, order, and expression values, biological tasks test downstream prediction, and RNA QA tests whether language-model answers reflect the queried RNA profile. For probing and biological prediction, we report both pretrained scores and gains over untrained models. Unless stated otherwise, results are aggregated within each task before reporting the mean and sample standard deviation across three pretraining seeds (7, 17, and 27). Gains are computed within each seed before averaging. Probe scores, classification macro-F1, and their gains are multiplied by 100, while perturbation correlations are unscaled.

4 PROBING EXPRESSION INFORMATION (RQ1)

Setup. We fit probes on 20,000 pretraining-corpus profiles per modality and evaluate on held-out profiles, keeping the models frozen. Mean-pooled probes fit a separate linear predictor for each gene. For order and value, we also evaluate token-level representations using one predictor shared across genes. Presence uses logistic regression and accuracy on measured-zero and positive labels balanced within each evaluation profile. Order uses ridge regression to predict within-profile ranks and is evaluated by pairwise ordering accuracy. Value uses ridge regression to predict log-normalized expression and reports the median per-gene R^2, relative to each gene's mean positive expression in the probe training data. Order and value are evaluated on expressed genes included in

Table 2: Expression probing of mean-pooled representations. Score is probe performance with the pretrained model. Gain is the pretrained score minus the corresponding untrained-model score. Bold indicates the highest Score and Gain separately for presence, order, and value within each modality.

	Presence		Order		Value	
Recipe	Score	Gain	Score	Gain	Score	Gain
Single-cell RNA-seq						
SC PRESENCE	$\mathbf{71.7 \pm 0.1}$	-0.6 ± 0.0	82.4 ± 0.0	$+1.2 \pm 0.0$	53.0 ± 0.3	$\mathbf{+20.1 \pm 0.5}$
SC IDENTITY	71.2 ± 0.2	-0.8 ± 0.2	82.6 ± 0.1	$+1.4 \pm 0.1$	63.5 ± 1.4	$+8.1 \pm 2.3$
SC GPT	71.6 ± 0.1	-0.4 ± 0.1	$\mathbf{82.8 \pm 0.0}$	$+0.7 \pm 0.0$	59.8 ± 1.4	$+0.6 \pm 2.3$
SC BIN-CLASS	69.7 ± 0.1	$\mathbf{+7.4 \pm 0.3}$	82.5 ± 0.0	$+1.1 \pm 0.0$	67.5 ± 0.3	$+3.2 \pm 0.3$
SC BIN-VALUE	70.4 ± 0.1	$+5.2 \pm 0.3$	82.7 ± 0.1	$\mathbf{+1.8 \pm 0.1}$	60.1 ± 4.8	$+2.5 \pm 4.7$
SC VALUE	70.5 ± 0.1	-0.3 ± 0.2	$\mathbf{82.8 \pm 0.0}$	$+1.4 \pm 0.0$	$\mathbf{67.7 \pm 1.0}$	$+15.2 \pm 1.2$
Bulk RNA-seq						
BULK IDENTITY	84.2 ± 0.6	$+1.9 \pm 0.7$	91.1 ± 0.2	$+1.4 \pm 0.2$	59.2 ± 1.9	$+15.7 \pm 1.7$
BULK GPT	$\mathbf{86.1 \pm 0.0}$	$+4.5 \pm 0.4$	$\mathbf{91.8 \pm 0.0}$	$+1.9 \pm 0.0$	65.1 ± 0.1	$+20.2 \pm 0.5$
BULK BIN-CLASS	83.0 ± 0.2	$+16.1 \pm 0.3$	90.9 ± 0.0	$\mathbf{+2.9 \pm 0.0}$	$\mathbf{69.5 \pm 0.1}$	$\mathbf{+22.5 \pm 0.2}$
BULK VALUE	83.9 ± 0.1	$\mathbf{+16.2 \pm 0.7}$	91.0 ± 0.0	$+0.8 \pm 0.0$	69.2 ± 0.2	$+3.9 \pm 0.2$

Table 3: Mean-pooled versus token-level representations: top recipe by pretrained probe score and by gain. Value is reported as the median per-gene R^2 among genes expressed in at least 50 test profiles. Presence is omitted because some recipes produce no tokens for measured-zero genes.

Modality	Target	Criterion	Mean-pooled	Token-level
Single-cell	Order	Highest score	SC GPT (82.8)	SC GPT (98.8)
		Largest gain	SC BIN-VALUE (+1.8)	SC VALUE (+35.0)
	Value	Highest score	SC VALUE (67.7)	SC VALUE (90.6)
		Largest gain	SC PRESENCE (+20.1)	SC VALUE (+87.8)
Bulk	Order	Highest score	BULK GPT (91.8)	BULK BIN-CLASS (89.6)
		Largest gain	BULK BIN-CLASS (+2.9)	BULK IDENTITY (+11.8)
	Value	Highest score	BULK BIN-CLASS (69.5)	BULK GPT (76.4)
		Largest gain	BULK BIN-CLASS (+22.5)	BULK GPT (+64.0)

every recipe's input. Tables 2 and 3 use the same genes and scoring. Data splits, probe fitting, and scoring are detailed in Appendix B.1.

Results. *The highest probe scores and largest pretraining gains often belong to different recipes.* The top-scoring and largest-gain recipes differ in five of the six modality–target comparisons (Table 2). Single-cell value probing illustrates this distinction: SC VALUE achieves the highest score (67.7) with a gain of 15.2, whereas SC PRESENCE achieves the largest gain (20.1) with a lower score (53.0). Bulk value is the only exception: BULK BIN-CLASS leads in both score (69.5) and gain (22.5). Although SC PRESENCE does not embed numeric expression values, its input genes are selected using expression values. Its value-probe gain therefore cannot be attributed to the presence objective alone. *Mean-pooled and token-level probes can yield different recipe rankings.* In bulk, the highest-scoring recipe switches from BULK GPT to BULK BIN-CLASS for order, and from BULK BIN-CLASS to BULK GPT for value, when moving from mean-pooled to token-level probes (Table 3). In single-cell data, the highest-scoring recipes remain unchanged, but token-level scores are higher: SC GPT increases from 82.8 to 98.8 for order, and SC VALUE from 67.7 to 90.6 for value. Because pooled probes are gene-specific whereas the token-level probe is shared across genes, these comparisons change both the representation and parameter sharing. Thus, the score differences should not be interpreted as the effect of mean pooling alone.

Table 4: Single-cell biological prediction. Tabula reports macro-F1 averaged over logistic regression, k-nearest neighbors, and random forest, with equal weighting across tissues and folds. Perturbation scores average top-20-DE correlations across conditions. For the Tabula expression baselines, SD is over five evaluation seeds. GEARS-only uses no transcriptome-model features.

(a) Cell-type classification

Model / input	Tabula Macro-F1 (%)
SC PRESENCE	$\mathbf{83.3 \pm 0.2}$
SC IDENTITY	77.4 ± 0.1
SC GPT	82.4 ± 0.0
SC BIN-CLASS	71.1 ± 1.4
SC BIN-VALUE	73.0 ± 0.8
SC VALUE	74.9 ± 2.3
Expression	57.9 ± 0.1
Expression PCA50	81.3 ± 0.1

(b) Genetic perturbation

Model	Adamson Pearson r	Norman Pearson r
SC PRESENCE	0.712 ± 0.018	0.659 ± 0.056
SC IDENTITY	0.612 ± 0.128	0.654 ± 0.044
SC GPT	0.593 ± 0.246	0.656 ± 0.010
SC BIN-CLASS	0.719 ± 0.040	0.671 ± 0.004
SC BIN-VALUE	0.716 ± 0.016	0.679 ± 0.021
SC VALUE	$\mathbf{0.738 \pm 0.010}$	$\mathbf{0.683 \pm 0.012}$
GEARS-only	0.480	0.558

Table 5: Bulk biological prediction. TCGA reports macro-F1 averaged over five patient-grouped folds. PANACEA reports correlations averaged across conditions in held-out cell lines.

(a) Cancer-type classification

Model / input	TCGA Macro-F1 (%)
BULK IDENTITY	83.5 ± 0.8
BULK GPT	82.8 ± 0.6
BULK BIN-CLASS	84.5 ± 0.6
BULK VALUE	$\mathbf{86.2 \pm 0.4}$
Expression PCA50–RF	88.8

(b) Chemical perturbation

Model / predictor	PANACEA Pearson r
BULK IDENTITY	$\mathbf{0.126 \pm 0.003}$
BULK GPT	0.102 ± 0.007
BULK BIN-CLASS	0.078 ± 0.009
BULK VALUE	0.075 ± 0.006
Raw control expression	0.201
Train-fold drug mean	0.175

5 BIOLOGICAL PREDICTION (RQ2)

Setup. For single-cell RNA-seq, we evaluate cell-type classification on Tabula Sapiens v2 (The Tabula Sapiens Consortium & Quake, 2025) and genetic perturbation prediction on Adamson (Adamson et al., 2016) and Norman (Norman et al., 2019). Classification uses mean-pooled representations, five-fold cell-level splits, and macro-F1. Perturbation prediction feeds token-level representations to GEARS, a graph neural network for perturbation prediction (Roohani et al., 2024); we hold out interventions and predict expression changes over 5,000 genes selected from the training data. Following the GEARS protocol, we report Pearson correlation over the 20 most differentially expressed (DE) genes per condition, averaged across conditions, which tests whether the response pattern of the most affected genes is recovered. For bulk RNA-seq, we evaluate cancer-type classification on TCGA (Weinstein et al., 2013) and chemical perturbation prediction on PANACEA (Douglass et al., 2022), both from mean-pooled representations. Classification uses five patient-grouped folds and macro-F1. For perturbation, we hold out cell lines and fit a PCA–ridge predictor on control-RNA representations, drug identity, and dose to predict expression changes over 5,000 genes selected from the training data. We report Pearson correlation across all 5,000 genes, averaged across conditions, which tests whether the response pattern across the full panel is recovered.

Results. *The leading recipe changes across tasks and does not consistently follow value-probe performance.* In single-cell data, SC PRESENCE leads Tabula macro-F1 (83.3) and beats the expression PCA50 baseline (81.3) despite the lowest pooled value-probe score, whereas SC VALUE leads Adamson (0.738) and Norman (0.683) (Tables 2 and 4). All six pretrained models improve on GEARS-only, but untrained-model features already close 56% (Adamson) and 93% (Norman) of this gap on average (Table 13). In bulk, BULK VALUE leads TCGA (86.2) but ranks last on PANACEA (0.075), where BULK IDENTITY, the weakest value-probe recipe, ranks first (0.126). No bulk recipe beats simple expression baselines: PCA50 reaches 88.8 on TCGA, and raw con-

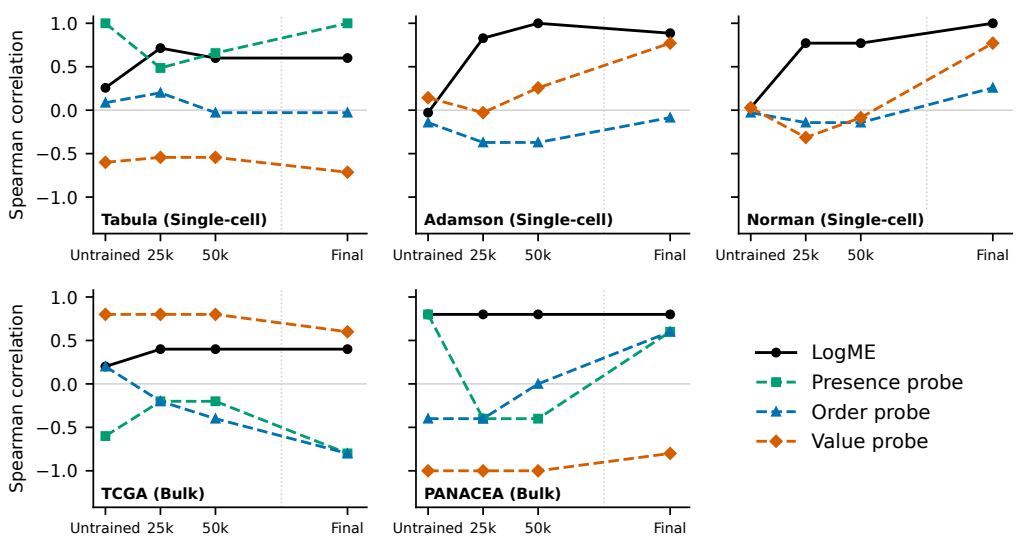

Figure 2: Spearman correlation, across the six single-cell or four bulk recipes, between LogME or expression-probe scores measured at each pretraining stage and biological performance. Probes use mean-pooled representations for classification and PANACEA, and gene tokens for Adamson and Norman. Checkpoint categories are not equally spaced training times: 25k steps correspond to 0.08 epoch for single-cell and 2.78 epochs for bulk.

trol expression and the drug mean reach 0.201 and 0.175 on PANACEA (Table 5). Recipes should therefore be compared on the target task against both untrained models and expression baselines.

Early signals of biological performance. We test how well LogME (You et al., 2021) and expression-probe scores at 0, 25k, 50k, and final checkpoints rank recipes by downstream performance, using Spearman correlations (Figure 2). LogME scores frozen representations against task training targets with a Bayesian linear model. Some signals exist before pretraining: at 0 steps, presence probes match Tabula rankings ($\rho = 1.00$), value probes track TCGA (0.80), and LogME tracks PANACEA (0.80, unchanged throughout), whereas LogME is near zero on Adamson and Norman. *By 25k steps, LogME correlates positively with performance on all five tasks* ($\rho = 0.40$–0.83). On Adamson and Norman, it reaches 0.83 and 0.77 while value-probe correlations are still negative, and rises to 1.00 and 0.77 at 50k. Probes can still outperform LogME: TCGA value probes do so at both early checkpoints, and Tabula presence and TCGA value probes at the final one. Thus, for genetic perturbation, LogME gives an earlier signal than value probes.

6 RNA Question Answering (RQ3)

Setup. We use CellWhisperer conversations (Schaefer et al., 2025) from Census-derived single-cell profiles and pseudobulk aggregates for single-cell models, and ARCHS4 GEO profiles for bulk models. Questions cover expression interpretation, cell/sample identity, and function/state, with 400 separate Tabula QA cell-type questions. A Q-Former connects each frozen RNA encoder to Qwen3-1.7B, with adaptation on modality-specific conversations and seed 7 for RNA pretraining and adaptation. We compare recipe rankings on reference-answer likelihood with expression-value probing and biological classification separately for each question group (Table 6), and report per-recipe perplexities (PPL, lower is better) in Table 7. We also replace RNA while keeping the text fixed to assess answer dependence on RNA. Training details and question counts are provided in Appendix B.3.

Results. *Biological performance aligns with RNA QA when the content is related.* Across single-cell encoders, cell-type classification correlates with reference-answer likelihood most strongly for main-source identity ($\rho = 0.94$), then function/state (0.77), Tabula QA (0.66), and expression inter-

Table 6: How expression probing and biological classification relate to RNA QA. Spearman correlations are computed across recipes. QA performance is negative mean reference-answer NLL, so positive correlations indicate agreement with better QA performance. Classification uses Tabula macro-F1 in SC and TCGA macro-F1 in Bulk. The value probes evaluate pooled and token representations, respectively.

Question group	Classification	Pooled value probe	Token value probe
Single-cell (six recipes)			
Expression interpretation	0.49	−0.54	−0.54
Cell/sample identity	0.94	−0.83	−0.60
Function/state	0.77	−0.66	−0.37
Tabula QA identity	0.66	−0.60	−0.43
Bulk (four recipes)			
Expression interpretation	−0.80	−0.40	0.20
Cell/sample identity	−0.40	−0.80	−0.60
Function/state	0.20	−0.40	−0.80

Table 7: Reference-answer perplexity for each RNA encoder connected to Qwen. Lower is better. Compare recipes within a question group and modality, since reference wording and length differ across groups. Parentheses give question counts. Tabula QA evaluates cell identity on a separate question set.

SC model + Qwen	Expression interpretation (60)	Cell/sample identity (115)	Function/state (96)	Cell identity Tabula QA (400)
SC PRESENCE	3.28	2.59	2.97	34.29
SC IDENTITY	3.30	2.70	2.97	32.30
SC GPT	3.31	2.62	2.94	31.19
SC BIN-CLASS	3.31	2.72	3.00	34.40
SC BIN-VALUE	3.32	2.80	3.00	37.59
SC VALUE	3.34	2.70	2.98	36.43

Bulk model + Qwen	Expression interpretation (35)	Cell/sample identity (76)	Function/state (67)
BULK IDENTITY	4.30	2.41	3.42
BULK GPT	4.34	2.42	3.43
BULK BIN-CLASS	4.32	2.63	3.46
BULK VALUE	4.37	2.41	3.39

pretation (0.49) (Tables 6 and 8). *Expression-value probing does not.* Pooled and token-level value probes both correlate negatively with expression questions ($\rho = -0.54$), so better expression reconstruction need not yield better RNA QA; bulk classification also shows no consistent alignment. The leading single-cell encoder varies by group: SC PRESENCE leads expression interpretation and main-source identity, and SC GPT leads function/state and Tabula QA (Table 7). Replacing the RNA profile lowers reference-answer likelihood for every encoder in every core group, and single-cell encoders more sensitive to this replacement tend to perform better, especially on function/state ($\rho = 1.00$) and main-source identity (0.83), although sensitivity does not fully determine the ranking (Appendix C.3).

Cell-type LogME across RNA and Qwen representations. On the 400 Tabula QA cells, we compare LogME of RNA cell embeddings and Qwen's last prompt-token representations using the same 20 cell-type labels, without answer tokens or conversation history. Across the six single-cell recipes, the two LogME rankings correlate at 0.94 with a randomly initialized Q-Former, 0.89 after alignment, and 0.89 after instruction tuning. This agreement predates alignment and cannot be attributed to adaptation alone. This analysis uses QA evaluation labels and does not establish prospective encoder-selection performance (Appendix D.4).

Table 8: Exploratory subtype correlations across six single-cell recipes. ρ is the Spearman correlation between a model score and reference-answer likelihood; classification is cell-type macro-F1. Rows were selected from 592 comparisons, and none survives Benjamini–Hochberg correction. LOO gives the leave-one-recipe-out range; intervals are 95% profile-bootstrap intervals, unadjusted for selection.

Model score	QA subtype	Questions	ρ	LOO	Profile interval
Classification	Cell type/subtype	100	0.94	0.90–1.00	$[0.60, 0.94]$
Classification	Tissue origin	15	-0.26	-0.70–0.30	$[-0.77, 0.37]$
Classification	Activation/state	23	0.89	0.80–0.90	$[0.14, 0.94]$
Adamson	Activation/state	23	0.83	0.70–1.00	$[0.03, 0.89]$
Norman	Activation/state	23	-0.31	-0.60–-0.10	$[-0.77, 0.20]$
Token value probe	Expression patterns	21	-0.09	-0.30–0.30	$[-0.77, 0.37]$

7 Discussion and Conclusion

What does pretraining add beyond the input and architecture? A representation can contain task-relevant information without learning it through pretraining. For example, SC Presence does not embed numeric expression values, but uses them to select input genes. High probe scores show that information is decodable, not that pretraining made it so; untrained encoders in NLP likewise score well on probes (Zhang & Bowman, 2018). The same holds for downstream prediction: untrained-model features already recover most of the genetic-perturbation gain over GEARS-only, especially on Norman (93%). Reporting pretrained scores together with pretrained-minus-untrained gains separates useful representations from improvements associated with pretraining, without equating those gains with an amount of learned information.

Why do pretrained models not always win? Our study reproduces a pattern seen in prior benchmarks (Kedzierska et al., 2025, Bendidi et al., 2024, Ahlmann-Eltze et al., 2025): pretraining does not consistently outperform methods without it. Pretraining improves classification for every recipe, but its perturbation gains are inconsistent, and no recipe leads across tasks. In bulk, expression PCA50 exceeds all recipes on TCGA, as do control expression and the drug mean on PANACEA. We did not isolate the cause, but the biology offers plausible explanations. Cancer types differ mainly through tissue-of-origin programs that PCA captures directly (Hoadley et al., 2018). On PANACEA, the drug mean alone reaches 0.175, suggesting that much of the response is a drug-specific signature shared across cell lines rather than a property of the baseline state encoded in a pooled representation. The benefit of pretraining therefore needs to be established for the intended task.

Which biological strengths carry over to RNA QA? In vision, encoders once judged mainly by image classification now serve as the front end of multimodal LLMs (Tong et al., 2024a), and their properties carry over: CLIP's failure patterns recur in MLLMs (Tong et al., 2024b), and class information persists in VLM hidden states (Zhang et al., 2024). Our results suggest the same for transcriptome encoders. Single-cell classification aligns most closely with identity QA, and cell-type LogME rankings agree between RNA and Qwen representations ($\rho = 0.94$) even with an untrained adapter. As transcriptome analysis moves toward language interfaces (Schaefer et al., 2025), encoders must be judged by what they deliver through language.

Can early evaluation guide recipe selection? Without fitting the downstream predictor, LogME tracks final genetic-perturbation performance by 25k steps ($\rho = 0.83$ and 0.77), while value probes remain negatively correlated. In other tasks, however, rankings can already exist at initialization, as on PANACEA, and selection quality depends on which recipes are compared (Appendix D.2). Similar low-cost signals guide data selection in language-model pretraining, where their reliability likewise varies by task (Magnusson et al., 2025, Bhagia et al., 2025). Prospective tests on new tasks are therefore needed. Overall, we recommend three steps: compare each pretrained model with the same model before pretraining, report how much pretraining improves its score, and check whether it beats simple expression baselines on the task of interest.

Ethics Statement

This study uses only publicly available transcriptomic data, and we did not collect new data from human participants. The human data were released in de-identified form by the original studies, and sample identifiers are excluded from model inputs and QA prompts. The pretrained models and transcriptome–language models are research artifacts for comparing pretraining recipes. They have not been validated for clinical use, and the QA models can produce fluent but incorrect answers.

Reproducibility Statement

Section 3 defines the pretraining comparison. Appendices A, B.1, B.2, and B.3 provide the data splits, preprocessing, training settings, and evaluation protocols. All datasets are public.

AI Use Statement

We used OpenAI ChatGPT/Codex and Claude Code to provide feedback on research methodology and experiments, assist with method implementation and edit the manuscript. The authors reviewed all AI-assisted work.

References

Britt Adamson, Thomas M. Norman, Marco Jost, Min Y. Cho, et al. A Multiplexed Single-Cell CRISPR Screening Platform Enables Systematic Dissection of the Unfolded Protein Response. *Cell*, 167(7):1867–1882.e21, 2016. doi: 10.1016/j.cell.2016.11.048.

Constantin Ahlmann-Eltze, Wolfgang Huber, and Simon Anders. Deep-learning-based gene perturbation effect prediction does not yet outperform simple linear baselines. *Nature Methods*, 22: 1657–1661, 2025. doi: 10.1038/s41592-025-02772-6.

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016. URL https://arxiv.org/abs/1610.01644.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022. URL https://aclanthology.org/2022.cl-1.7/.

Ihab Bendidi, Shawn Whitfield, Kian Kenyon-Dean, Hanene Ben Yedder, Yassir El Mesbahi, Emmanuel Noutahi, and Alisandra K. Denton. Benchmarking transcriptomics foundation models for perturbation analysis: one PCA still rules them all. *arXiv preprint arXiv:2410.13956*, 2024. URL https://arxiv.org/abs/2410.13956.

Akshita Bhagia, Jiacheng Liu, Alexander Wettig, David Heineman, Oyvind Tafjord, Ananya Harsh Jha, Luca Soldaini, Noah A. Smith, Dirk Groeneveld, Pang Wei Koh, Jesse Dodge, and Hannaneh Hajishirzi. Establishing task scaling laws via compute-efficient model ladders. In *Conference on Language Modeling*, 2025. URL https://arxiv.org/abs/2412.04403.

Chan Zuckerberg Initiative. CELLxGENE Census: Data releases, 2025. URL https://chanzuckerberg.github.io/cellxgene-census/cellxgene_census_docsite_data_release_info.html. Release 2025-11-08.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamás Sarlós, et al. Rethinking attention with performers. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2009.14794.

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.

Gaetan De Waele, Gerben Menschaert, and Willem Waegeman. A systematic assessment of single-cell language model configurations. *NAR Genomics and Bioinformatics*, 8(3):lqag095, 2026. doi: 10.1093/nargab/lqag095.

Alan DenAdel, Madeline Hughes, Akshaya Thoutam, Anay Gupta, Andrew W. Navia, Nicolo Fusi, et al. Evaluating the role of pretraining dataset size and diversity on single-cell foundation model performance. *Nature Methods*, 23(7):1447–1457, 2026. doi: 10.1038/s41592-026-03120-y.

Eugene F. Douglass, Robert J. Allaway, Bence Szalai, Wenyu Wang, Tingzhong Tian, Adrià Fernández-Torras, et al. A community challenge for a pancancer drug mechanism of action inference from perturbational profile data. *Cell Reports Medicine*, 3(1):100492, 2022. doi: 10.1016/j.xcrm.2021.100492.

Mary J. Goldman, Brian Craft, Mim Hastie, Kristupas Repečka, Fran McDade, Akhil Kamath, Ayan Banerjee, Yunhai Luo, Dave Rogers, Angela N. Brooks, Jingchun Zhu, and David Haussler. Visualizing and interpreting cancer genomics data via the Xena platform. *Nature Biotechnology*, 38(6):675–678, 2020. doi: 10.1038/s41587-020-0546-8.

Maxence Gélard, Guillaume Richard, Thomas Pierrot, and Paul-Henry Cournède. BulkRNABert: Cancer prognosis from bulk RNA-seq based language models. In *Proceedings of Machine Learning Research*, volume 259, pp. 384–400, 2025. URL https://proceedings.mlr.press/v259/gelard25a.html.

Katherine A. Hoadley, Christina Yau, Toshinori Hinoue, Denise M. Wolf, Alexander J. Lazar, et al. Cell-of-Origin Patterns Dominate the Molecular Classification of 10,000 Tumors from 33 Types of Cancer. *Cell*, 173(2):291–304.e6, 2018. doi: 10.1016/j.cell.2018.03.022.

Boming Kang, Rui Fan, Meizheng Yi, Chunmei Cui, and Qinghua Cui. BulkFormer: A large-scale foundation model for bulk transcriptomes. *Cell Systems*, 17(7):101657, 2026. doi: 10.1016/j.cels.2026.101657.

Kasia Z. Kedzierska, Lorin Crawford, Ava P. Amini, and Alex X. Lu. Zero-shot evaluation reveals limitations of single-cell foundation models. *Genome Biology*, 26(1), 2025. doi: 10.1186/s13059-025-03574-x.

Daniel Levine, Syed A. Rizvi, Sacha Lévy, Nazreen Pallikkavaliyaveetil, David Zhang, Xingyu Chen, et al. Cell2Sentence: Teaching large language models the language of biology. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 27299–27325, 2024. URL https://proceedings.mlr.press/v235/levine24a.html.

Tianyu Liu, Kexing Li, Yuge Wang, Hongyu Li, and Hongyu Zhao. Evaluating the Utilities of Foundation Models in Single-Cell Data Analysis. *Advanced Science*, 2026. doi: 10.1002/advs.202514490.

Ian Magnusson, Nguyen Tai, Ben Bogin, David Heineman, Jena D. Hwang, Luca Soldaini, Akshita Bhagia, Jiacheng Liu, Dirk Groeneveld, Oyvind Tafjord, Noah A. Smith, Pang Wei Koh, and Jesse Dodge. DataDecide: How to predict best pretraining data with small experiments. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, 2025. URL https://proceedings.mlr.press/v267/magnusson25a.html.

Thomas M. Norman, Max A. Horlbeck, Joseph M. Replogle, Alex Y. Ge, et al. Exploring genetic interaction manifolds constructed from rich single-cell phenotypes. *Science*, 365(6455):786–793, 2019. doi: 10.1126/science.aax4438.

Olga Ovcharenko, Florian Barkmann, Philip Toma, Imant Daunhawer, Julia E. Vogt, Sebastian Schelter, and Valentina Boeva. scSSL-Bench: Benchmarking self-supervised learning for single-cell data. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 47416–47442, 2025. URL https://proceedings.mlr.press/v267/ovcharenko25a.html.

Yusuf Roohani, Kexin Huang, and Jure Leskovec. Predicting transcriptional outcomes of novel multigene perturbations with GEARS. *Nature Biotechnology*, 42(6):927–935, 2024. doi: 10.1038/s41587-023-01905-6.

Yanay Rosen, Yusuf Roohani, Ayush Agrawal, Leon Samotorcan, Tabula Sapiens Consortium, Stephen R. Quake, and Jure Leskovec. Universal Cell Embeddings: A Foundation Model for Cell Biology. *bioRxiv*, 2023. doi: 10.1101/2023.11.28.568918.

Moritz Schaefer, Peter Peneder, Daniel Malzl, Salvo Danilo Lombardo, Mihaela Peycheva, Jake Burton, et al. Multimodal learning enables chat-based exploration of single-cell data. *Nature Biotechnology*, 2025. doi: 10.1038/s41587-025-02857-9.

Hongru Shen, Jilei Liu, Jiani Hu, Xilin Shen, Chao Zhang, Dan Wu, Mengyao Feng, Meng Yang, Yang Li, Yichen Yang, Wei Wang, Qiang Zhang, Jilong Yang, Kexin Chen, and Xiangchun Li. Generative pretraining from large-scale transcriptomes for single-cell deciphering. *iScience*, 26 (5):106536, 2023. doi: 10.1016/j.isci.2023.106536.

Zheng Su, Mingyan Fang, Andrei Smolnikov, Marcel E. Dinger, Emily C. Oates, and Fatemeh Vafaee. GeneRAIN: multifaceted representation of genes via deep learning of gene expression networks. *Genome Biology*, 26:288, 2025. doi: 10.1186/s13059-025-03749-6.

The Tabula Sapiens Consortium and Stephen R. Quake. Tabula Sapiens reveals transcription factor expression, senescence effects, and sex-specific features in cell types from 28 human organs and tissues. *bioRxiv*, 2025. doi: 10.1101/2024.12.03.626516. URL https://www.biorxiv.org/content/10.1101/2024.12.03.626516v2.

Christina V. Theodoris, Ling Xiao, Anant Chopra, Mark D. Chaffin, Zeina R. Al Sayed, Matthew C. Hill, et al. Transfer learning enables predictions in network biology. *Nature*, 618(7965):616–624, 2023. doi: 10.1038/s41586-023-06139-9.

Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, Ziteng Wang, Rob Fergus, Yann LeCun, and Saining Xie. Cambrian-1: A fully open, vision-centric exploration of multimodal LLMs. In *Advances in Neural Information Processing Systems*, 2024a. URL https://arxiv.org/abs/2406.16860.

Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal LLMs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9568–9578, 2024b.

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*, 2017. URL https://arxiv.org/abs/1706.03762.

John N. Weinstein, Eric A. Collisson, Gordon B. Mills, Kenna R. Mills Shaw, Brad A. Ozenberger, Kyle Ellrott, Ilya Shmulevich, Chris Sander, Joshua M. Stuart, and The Cancer Genome Atlas Research Network. The Cancer Genome Atlas Pan-Cancer analysis project. *Nature Genetics*, 45 (10):1113–1120, 2013. doi: 10.1038/ng.2764.

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.

Kaichao You, Yong Liu, Jianmin Wang, and Mingsheng Long. LogME: Practical assessment of pre-trained models for transfer learning. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 12133–12143. PMLR, 2021. URL https://proceedings.mlr.press/v139/you21b.html.

Yuansong Zeng, Jiancong Xie, Ningyuan Shangguan, Zhuoyi Wei, Wenbing Li, Yun Su, et al. CellFM: a large-scale foundation model pre-trained on transcriptomics of 100 million human cells. *Nature Communications*, 16:4679, 2025. doi: 10.1038/s41467-025-59926-5.

Kelly Zhang and Samuel R. Bowman. Language modeling teaches you more than translation does: Lessons learned through auxiliary syntactic task analysis. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, 2018. URL https://aclanthology.org/W18-5448/.

Yuhui Zhang, Alyssa Unell, Xiaohan Wang, Dhruba Ghosh, Yuchang Su, Ludwig Schmidt, and Serena Yeung-Levy. Why are visually-grounded language models bad at image classification? In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2405.18415.

Suyuan Zhao, Jiahuan Zhang, Yushuai Wu, Yizhen Luo, and Zaiqing Nie. LangCell: Language-cell pre-training for cell identity understanding. In *Proceedings of the 41st International Conference on Machine Learning*, 2024. URL https://openreview.net/pdf?id=GcZjpKA37R.

A PRETRAINING RECIPES AND IMPLEMENTATION

A.1 PRETRAINING RESOURCES AND SCOPE

From CELLxGENE Census release 2025-11-08, we select human primary-cell 10x profiles with at least 15,000 measured genes, 200 detected genes, and positive total counts. We partition the selected studies into disjoint training, validation, and test sets and sample 20,000,000/500,000/1,000,000 cells from 139/8/16 studies, respectively. The training set spans 245 datasets, and the three sets together span 280 datasets from 163 studies. Measurement masks distinguish measured zeros from unmeasured genes on a common 20,053-gene panel.

We retain the released PreBULK train/test split of 580,315/1,188 profiles and reserve 5,803 profiles from its training split for validation, leaving 574,512 for training. Because study and batch annotations are unavailable, these partitions are sample-disjoint rather than study-disjoint.

A.2 ARCHITECTURE AND MODEL SIZE

All models use eight layers, hidden size 512, eight attention heads, feed-forward size 2,048, and dropout 0.1. Transformer-based recipes read at most 2,048 genes. The full-panel BULK BIN-CLASS and BULK VALUE recipes use Performers with 256 random features. GPT recipes use causal attention, whereas all other recipes use bidirectional attention. Pooling averages valid gene states, with measured-zero entries excluded for BULK IDENTITY and BULK GPT. Table 9 reports total parameter counts, which are not matched despite the shared encoder depth and width.

A.3 SINGLE-CELL INPUT SELECTION, ZERO EXPRESSION, AND PADDING

Gene selection and measurement states. SC IDENTITY and SC GPT include positive genes only. SC VALUE, SC BIN-CLASS, and SC BIN-VALUE prioritize positive genes by expression and fill remaining positions up to 2,048 with measured-zero genes using a deterministic ordering. Unmeasured genes are excluded, and any remaining positions are padding. Measured-zero entries participate in attention and pooling for these three recipes, although only positive entries are masked and supervised. SC PRESENCE removes held-out positive genes before constructing its visible positive-gene context and predicts presence through separate gene queries. Consequently, absence from the contextual sequence does not distinguish an unmeasured gene, a measured zero, and a positive gene omitted by truncation.

SC PRESENCE. We hold out 20% of positive genes and select up to 2,048 remaining positive genes according to expression. Each contextual token combines a learned gene-identity embedding with an embedding of the constant value one, without positional embeddings. A small MLP predicts presence from the pooled representation together with a query gene embedding. Up to 512 held-out positive genes and an equal number of measured-zero genes are used as queries under binary cross-entropy. This adapts the UCE presence objective without its protein-derived embeddings or repeated gene sampling (Rosen et al., 2023). Although numeric expression values are not embedded directly, expression-dependent input selection can convey magnitude through gene inclusion. The same selection rule is applied to untrained models. The pretrained–untrained difference therefore measures the effect of pretraining conditional on this rule, rather than the contribution of the selection rule itself.

SC IDENTITY and SC GPT. SC IDENTITY normalizes counts to CP10K using the full source count sum and divides each gene by its training-fitted median before ranking positive genes. Gene-identity and learned rank-position embeddings are then provided to the Transformer. Fifteen percent of selected identities are prediction targets, using 80% MASK replacement, 10% random replacement, and 10% unchanged inputs. The objective is cross-entropy over the original gene identities. SC GPT instead ranks genes by CP10K without gene-median correction and predicts the next gene identity under causal attention. It uses no random masking, and padding positions or positions without a valid successor do not contribute to the objective. These recipes adapt the masked-identity objective of Geneformer and the next-identity objective of tGPT, respectively (Theodoris et al., 2023, Shen et al., 2023).

SC BIN-CLASS and SC BIN-VALUE. SC BIN-CLASS applies $\log_2(1+\mathrm{CP10K})$, floors positive values, clips them to levels 1–5, and reserves category 0 for measured zeros. SC BIN-VALUE applies $\log(1+\mathrm{CP10K})$ and defines bins 1–50 from within-context positive-expression quantiles, again reserving 0 for measured zeros. Both recipes add learned gene and bin embeddings without positional embeddings, mask 20% of positive entries, and replace masked values with a MASK embedding. SC BIN-CLASS predicts categorical bins using cross-entropy, whereas SC BIN-VALUE regresses numerical bin values using MSE. The two recipes therefore differ in their bin definitions, target scales, and loss functions. They adapt scBERT and scGPT inputs without reproducing all aspects of the source models' architectures or training procedures (Yang et al., 2022, Cui et al., 2024).

SC VALUE. Inputs and targets use $\log(1+\mathrm{CP10K})$. A CellFM-inspired nonlinear scalar-value embedding is added to each gene embedding. We mask 20% of positive values and predict their original values using a linear head with MSE. The shared Transformer omits CellFM's separate read-depth projection, CLS auxiliary objective, and full-gene decoder (Zeng et al., 2025).

A.4 BULK INPUT ORDERING, ZERO EXPRESSION, AND SOURCE-MODEL DIFFERENCES

BULK IDENTITY and BULK GPT. Both recipes compute CP10M from TPM-derived pseudo-counts across the full measured panel, apply $\log_{10}(1+x)$, and compare each value with 1,998 training-fitted boundaries for that gene. Measured zeros receive score 1, and positive values receive scores 2–2,000. The 2,048 genes with the highest scores are retained, with deterministic tie breaking. Only gene identities and rank positions enter the Transformer; the scores determine gene selection and ordering but are not embedded as additional expression features. Selected measured-zero genes retain identities and positions but are excluded as attention keys and values and from pooled representations.

BULK IDENTITY independently selects context positions with probability 0.15, excludes measured zeros from the target set, replaces each selected positive identity with MASK, and minimizes identity cross-entropy. BULK GPT instead uses causal attention and next-identity cross-entropy without random masking. Its labels include selected measured-zero identities and exclude structural padding. Query states at measured-zero positions remain active. These choices adapt GeneRAIN's Binning-By-Gene preprocessing and released BERT/GPT input paths (Su et al., 2025). Our pseudo-count source, eight-layer backbone, vocabulary, training-fitted boundaries, and pretraining corpus differ from those of the source model.

BULK BIN-CLASS and BULK VALUE. Both recipes retain the full measured gene panel in fixed gene-ID order and use Performers. Unmeasured genes and padding are excluded, whereas measured zeros remain valid entries and may be supervised. BULK BIN-CLASS maps $\log_{10}(1+\mathrm{TPM})$ to positive bins 1–64 using a training-fitted global maximum, with measured zero assigned to category 0. Learned gene and bin embeddings are added, and 15% of measured entries are selected using 80/10/10 MASK/random/unchanged replacement before categorical prediction with cross-entropy (Gélard et al., 2025).

BULK VALUE adds learned gene embeddings to sinusoidal embeddings of $\log(1+\mathrm{TPM})$. Fifteen percent of measured entries have their value embeddings replaced by zero vectors, which remain distinct from the sinusoidal representation of measured-zero expression. A linear head predicts the selected continuous values using MSE. This adapts BulkFormer's value transformation without its graph- and protein-based components (Kang et al., 2026).

Gene-specific scaling in SC IDENTITY and BULK IDENTITY/BULK GPT can alter within-sample gene order. Expression-dependent gene selection can also encode expression information even when numeric expression values are not embedded directly.

A.5 OPTIMIZATION AND CHECKPOINT SELECTION

Pretraining uses seeds 7, 17, and 27, with two epochs for single-cell RNA-seq and twenty epochs for bulk RNA-seq. AdamW uses weight decay 0.01, gradient-norm clipping at 1.0, 500 warmup steps, and a constant learning rate thereafter. The effective batch size is 64 RNA profiles. Validation is performed every 5,000 optimizer steps, and the checkpoint with the lowest validation loss is selected

separately for each recipe and seed. Table 9 reports recipe-specific learning rates and masking fractions. Validation loss is used only for checkpoint selection within a recipe and is not compared across recipes.

Table 9: Recipe configurations. Shared encoder blocks contain 25.22M parameters. Total counts include prediction heads and 256 fixed frequencies for BULK VALUE; Performer projection buffers are excluded. Epochs denote the allotted pretraining budget.

Recipe	Parameters	Learning rate	Masking	Epochs
SC PRESENCE	36,276,737	10^{-4}	20%	2
SC IDENTITY	46,823,509	5×10^{-5}	15%	2
SC GPT	46,823,509	5×10^{-5}	0%	2
SC BIN-CLASS	35,494,406	5×10^{-5}	20%	2
SC BIN-VALUE	35,515,393	5×10^{-5}	20%	2
SC VALUE	35,685,634	5×10^{-5}	20%	2
BULK IDENTITY	46,823,509	10^{-4}	15%	20
BULK GPT	46,823,509	10^{-4}	0%	20
BULK BIN-CLASS	35,554,881	10^{-4}	15%	20
BULK VALUE	35,488,513	10^{-4}	15%	20

B EVALUATION PROTOCOLS

B.1 EXPRESSION PROBING

Data and targets. Training/validation/test partitions contain 20,000/5,000/3,558 single-cell profiles and 20,000/4,997/1,186 bulk profiles from the corresponding pretraining corpora, after removing two bulk test profiles duplicated in the training data. Presence training uses measured-gene labels and excludes unmeasured entries. For presence evaluation, each validation and test profile contributes equal numbers of measured-zero genes and positive genes that appear in every recipe's input. Labels are pooled across profiles rather than weighted equally across genes. Order and value use all positive profile–gene pairs in the shared input intersection, restricted to genes with eligible positive support in the probe-training data. Value targets are $\log(1 + \mathrm{CP10K})$ for single-cell RNA-seq and $\log(1 + \mathrm{TPM})$ for bulk RNA-seq.

Fitting and scoring. Probes use unmasked final-layer representations, with features and gene-specific value targets standardized on probe-training data. Logistic probes use binary cross-entropy on measured entries with per-gene positive-class weights. Regularization is selected by validation performance: accuracy for presence and order, and median per-gene R^2 for value. Candidate strengths range from 10^{-6} to 10^{-3} for logistic probes and from 10^{-3} to 10^{8} in powers of ten for ridge probes. Order targets are within-profile average-tie ranks divided by the number of expressed genes. Pairwise accuracy excludes equal target values and gives half credit to tied predictions. For value,

$$R_g^2 = 1 - \frac{\sum_{i \in E_g} (\hat{y}_{ig} - y_{ig})^2}{\sum_{i \in E_g} (y_{ig} - \mu_g)^2},$$

where E_g contains evaluation profiles in which gene g is expressed and included in every recipe's input. The baseline μ_g is its mean positive expression under the same input restriction in probe-training data. We report the median over genes with at least 50 such evaluation profiles and a nonzero denominator, retaining negative scores. Untrained models use the same probe-fitting and selection procedures.

Paired pooled/token comparison. Table 3 evaluates pooled and token-level representations on the same positive profile–gene pairs using the same target definitions and scoring procedures. Pooled ridge probes are gene-specific, whereas the token-level ridge probe is shared across genes. Regularization is selected separately for each representation and target. The comparison therefore changes both the representation and the degree of parameter sharing and should not be interpreted as an isolated effect of pooling.

B.2 BIOLOGICAL EVALUATION

Cell-type classification. Tabula Sapiens v2 includes 65,970 cells from blood, bone marrow, lung, mammary tissue, and thymus after excluding cell types represented by fewer than ten cells within a tissue. Splits are stratified by cell type. We evaluate nearest neighbors, logistic regression, and random forests using the settings in Table 10. KNN and logistic regression scale features by training-set standard deviations without centering, whereas random forests use unscaled features. Within each tissue, macro-F1 is averaged over classifiers, folds, and evaluation seeds 42–46; tissue-level means are then averaged equally.

Genetic perturbation prediction. A learned $512 \rightarrow 64$ projection maps frozen control-cell token representations to GEARS gene embeddings. Genes without RNA-model tokens retain learned GEARS embeddings. Target genes are selected according to variance among training cells without altering the native RNA-model inputs or renormalizing prediction targets. The projection and GEARS are trained jointly using the GEARS expression and direction losses with the settings in Table 10. Validation selects the checkpoint with the lowest condition-averaged top-DE MSE. Predictions are averaged within each perturbation condition before subtracting mean control expression. All recipes and the GEARS-only baseline use the same intervention-disjoint split, predictor seed 1, gene panels, control pairings, and training settings.

Cancer-type classification. TCGA expression is obtained from UCSC Xena (Goldman et al., 2020). Within each of five patient-grouped folds, PCA fitted on the training fold reduces the input to 50 dimensions before classification with a random forest using bootstrap-balanced class weights. Splitting and prediction use seed 42, and macro-F1 is averaged equally across folds.

Chemical perturbation prediction. PANACEA targets are defined as treated $\log(1 + \mathrm{CP10K})$ expression minus mean transformed DMSO expression. Profiles are required to contain at least 100,000 counts, including both test replicates. Each of eleven folds holds out one cell line for testing and one for validation. Gene selection is based on variance across treated profiles and unique controls from the remaining nine training cell lines; this gene panel is then kept fixed when refitting on all ten non-test lines. All test drugs are represented in the training data.

Training-fitted scaling and PCA50 transform the control-RNA representations, which are concatenated with drug indicators and standardized $\log(1 + \mathrm{dose})$ values in micromolar units. Ridge regression ($\alpha = 10$) predicts PCA50 target coordinates, which are subsequently mapped back to gene space. Predictions and targets are averaged over the two replicates within each condition. Raw-expression and training-fold drug-mean baselines use the same folds.

Table 10: Biological predictor settings. Preprocessing, split construction, and selection criteria are defined in the task-specific protocols. Settings are shared across RNA recipes and their corresponding controls.

Predictor	Settings
Tabula classifiers	5-NN; logistic regression with $C = 1$ and 100 iterations; random forest with 100 trees.
GEARS	20 epochs, batch size 32, Adam learning rate 10^{-3}, weight decay 5×10^{-4}, learning rate halved after each epoch.
TCGA random forest	500 trees, bootstrap-balanced class weights, predictor seed 42.
PANACEA ridge	Input and target PCA50, $\alpha = 10$, predictor seed 42.

Pretraining exposure. The Tabula Sapiens study used for biological cell-type classification contributes no cells to single-cell pretraining. This exposure check does not include the separate Tabula QA set. TCGA and PANACEA evaluation samples are excluded from bulk pretraining.

B.3 RNA QA DATA, ADAPTATION, AND SCORING

Data and adaptation. We use the released CellWhisperer descriptions and instruction conversations (Schaefer et al., 2025), retaining their original question wording, reference answers, and

conversation histories. Single-cell systems use Census-derived single-cell profiles and pseudobulk aggregates, including Smart-seq2 profiles assigned to this source. Bulk systems use ARCHS4 GEO bulk profiles. Training does not mix these sources. Each RNA encoder retains its native gene panel, measurement mask, expression normalization, and token-selection procedure. The Q-Former reads the complete valid RNA-token sequence together with its corresponding mask.

Table 11: RNA QA data and evaluation overview. Training counts denote conversations. Evaluation counts denote selected final answers and may include multiple questions about the same RNA profile. Tabula QA is a separate cell-identity source with short cell-type references. Core question groups contain 849 questions in total; 67 additional questions belong to auxiliary strata.

Data or evaluation stratum	SC	Bulk
Alignment training conversations	376,881	705,330
Instruction training conversations	52,878	53,731
Expression interpretation (E)	60	35
Cell/sample identity, main source (I)	115	76
Biological function or state (B)	96	67
Multi-topic descriptions, auxiliary (S)	15	37
Experimental metadata, auxiliary (M)	1	14
Tabula QA cell-type questions	400	—

Alignment trains the Q-Former for one epoch while keeping the RNA encoder and Qwen frozen. Instruction tuning then updates the aligned Q-Former together with Qwen LoRA for one epoch, while the RNA encoder remains frozen. We evaluate the resulting instruction-tuned model without validation-based checkpoint selection or early stopping.

The Q-Former has hidden size 768, 12 layers, 12 attention heads, feed-forward size 3,072, and cross-attention every second layer. Eight learned queries attend to 512-dimensional RNA tokens and are projected to eight 2,048-dimensional vectors for Qwen3-1.7B. LoRA uses rank 16, alpha 32, zero dropout, and the q/k/v/o and gate/up/down projections. AdamW uses learning rate 10^{-4} during alignment and 2×10^{-5} during instruction tuning, with zero weight decay, cosine scheduling, and a warmup fraction of 0.03. Both stages use BF16 and an effective global batch size of 64. Only assistant-answer tokens, including the end token, contribute to the training cross-entropy. All retained examples fit within the 2,048-token context window without truncation.

RNA QA evaluates a different target from expression probing and perturbation prediction. Questions about treatment or genetic-intervention effects ask the model to interpret an observed RNA profile rather than predict a counterfactual expression response. They therefore do not directly evaluate the perturbation-prediction tasks in Section 5. QA performance is measured by reference-answer likelihood rather than generated-answer accuracy.

Question content and sources. The main evaluation contains 516 questions from 200 CellWhisperer conversations, comprising 287 single-cell and 229 bulk questions. Our inventory includes follow-up turns beyond the 200 initial question–answer pairs used in the published CellWhisperer evaluation (Schaefer et al., 2025). Questions are assigned to three primary content groups without reference to model scores: expression interpretation, cell or sample identity, and biological function or state. Multi-topic descriptions and experimental metadata are retained as auxiliary strata but are excluded from the primary E/I/B comparison.

Tabula QA contributes an additional 400 first-turn cell-identity questions, with 20 cells per reference cell type and short annotation-based answers. It is a separate evaluation source and does not reproduce the published CellWhisperer evaluation over 15,158 transcriptomes. Original reference answers and conversation histories are retained throughout. All 95 expression questions are follow-up turns, and earlier answers may already reveal cell identity; comparisons across content groups therefore also reflect differences in conversation history and reference construction.

We do not exclude questions on the basis of annotation uncertainty. Among the retained questions, 94 category assignments were marked uncertain and 21 were flagged for potential reference or context issues. These annotations were assigned without inspecting model scores and were not independently validated; inter-annotator agreement was not measured.

Likelihood scoring. Using teacher forcing, we compute mean negative log-likelihood (NLL) over the final reference answer, including its end token, conditioned on the original conversation history and RNA profile. Each question receives equal weight, and reported perplexity is the exponential of the mean question-level NLL. Earlier conversation turns condition the model but are not included in the score.

C Controls and Robustness

C.1 Untrained models

Untrained models retain the architecture, initialization seed, and preprocessing of the corresponding recipe. Downstream predictors are refitted using the same splits and evaluation settings as for pretrained models. Tables report three-seed means, and gains are paired by recipe and seed.

Table 12: Cell-type (Tabula) and cancer-type (TCGA) classification macro-F1 (%) for untrained and pretrained models, averaged across three pretraining seeds.

SC recipe	Untrained → pretrained	Bulk recipe	Untrained → pretrained
Presence	$43.6 \rightarrow 83.3$	Identity	$77.6 \rightarrow 83.5$
Identity	$46.1 \rightarrow 77.4$	GPT	$74.7 \rightarrow 82.8$
GPT	$67.0 \rightarrow 82.4$	Bin-Class	$31.7 \rightarrow 84.5$
Bin-Class	$34.3 \rightarrow 71.1$	Value	$78.5 \rightarrow 86.2$
Bin-Value	$27.3 \rightarrow 73.0$		
Value	$46.8 \rightarrow 74.9$		

Table 13: Genetic perturbation prediction for untrained and pretrained model–GEARS systems. Entries report top-20-DE Pearson correlation on the 5,000-gene panel, averaged across three pretraining seeds. GEARS-only reaches 0.480 on Adamson and 0.558 on Norman.

Recipe	Adamson	Norman
SC Presence	$0.734 \rightarrow 0.712$	$0.656 \rightarrow 0.659$
SC Identity	$0.482 \rightarrow 0.612$	$0.693 \rightarrow 0.654$
SC GPT	$0.285 \rightarrow 0.593$	$0.663 \rightarrow 0.656$
SC Bin-Class	$0.588 \rightarrow 0.719$	$0.650 \rightarrow 0.671$
SC Bin-Value	$0.737 \rightarrow 0.716$	$0.667 \rightarrow 0.679$
SC Value	$0.730 \rightarrow 0.738$	$0.626 \rightarrow 0.683$

Pretraining improves top-20-DE correlation in 7/18 Adamson and 11/18 Norman recipe–seed pairs.

C.2 Full-input expression baselines

Tabula expression baselines use all 20,053 panel genes together with measurement masks and $\log(1 + \mathrm{CP10K})$, either directly or after training-fold scaling and PCA50. They use the same cells, splits, and classifiers as the representation-based comparison. Mean macro-F1 is 57.9 ± 0.1 for normalized expression and 81.3 ± 0.1 for PCA50, with the standard deviation computed over evaluation seeds 42–46. Logistic regression reaches the 100-iteration limit in 17 of 250 fits. These baselines use the complete expression panel, whereas RNA-model inputs differ across recipes.

TCGA uses the same 10,429 samples and patient-grouped folds as the representation-based comparison. The original 20,010-gene expression matrix receives no additional normalization. Training-fold PCA50 followed by the same 500-tree random forest yields a mean macro-F1 of 88.8, compared with 86.2 for the best pretrained recipe. This is a single shared baseline rather than a three-pretraining-seed measurement.

C.3 Dependence of RNA QA on the RNA input

To test whether RNA QA predictions depend on the RNA input, we replace each question's RNA profile with 30 other profiles drawn from the same source. We use the same replacement profiles

Table 14: Mean increase in reference-answer NLL after replacing the RNA input (ΔNLL). Positive values indicate higher likelihood for the reference answer with the original RNA profile. Parentheses give the number of questions.

SC model + Qwen	Expression interpretation (60)	Cell/sample identity (115)	Function/state (96)	Cell identity Tabula QA (400)
SC PRESENCE	0.106	1.062	0.123	0.827
SC IDENTITY	0.095	0.982	0.124	0.816
SC GPT	0.081	1.126	0.130	0.879
SC BIN-CLASS	0.062	0.914	0.105	0.687
SC BIN-VALUE	0.067	0.916	0.102	0.567
SC VALUE	0.066	1.045	0.113	0.834

Bulk model + Qwen	Expression interpretation (35)	Cell/sample identity (76)	Function/state (67)
BULK IDENTITY	0.017	0.982	0.091
BULK GPT	0.015	1.006	0.094
BULK BIN-CLASS	0.013	0.712	0.068
BULK VALUE	0.008	1.032	0.097

for all encoders while keeping the question, conversation history, and reference answer fixed. For each question, we compute the replacement-minus-original answer NLL (ΔNLL), average over the 30 replacements, and then average equally across questions. Positive ΔNLL indicates that replacing the RNA makes the reference answer less likely. Every encoder shows positive mean ΔNLL in every core question group (Table 14), indicating that the original RNA improves mean reference-answer likelihood. This control measures dependence on the RNA input, rather than the effect of RNA pretraining.

D TRANSFERABILITY ANALYSES

D.1 PROBE–BIOLOGY CORRESPONDENCE

Spearman correlations use one three-seed mean per recipe, giving six single-cell or four bulk observations per comparison. Probes use pooled representations for classification and PANACEA and token-level representations for Adamson and Norman, matching the corresponding biological predictors. Gain correlations use pretrained-minus-untrained differences paired by recipe and seed, with both probes and downstream predictors refitted. Seeds and evaluation folds are not treated as independent recipes.

Biological protocols are defined in Appendix B.2. The positive token-value score correlations on Adamson and Norman are substantially weaker when pretrained-minus-untrained gains are compared instead. Agreement in pretrained scores therefore does not imply agreement in the benefit attributable to pretraining.

D.2 TASK-AWARE LOGME

LogME estimates training-target evidence under a Bayesian linear model applied to frozen representations (You et al., 2021), without retraining the transcriptome encoder or the full downstream predictor. This is a retrospective analysis designed after the biological results were available. Comparisons are made only within a task, using six single-cell or four bulk recipes and averaging task-level quantities across the three pretraining seeds.

Features, targets, and aggregation. Tabula and TCGA use unscaled 512-dimensional mean-pooled training representations with one-vs-rest class labels. Their aggregation follows the biological evaluation: Tabula averages folds and evaluation seeds within tissue before weighting tissues equally, whereas TCGA weights its five folds equally.

Table 15: Exploratory Spearman correlations across recipes. Score correlation compares probe and biological scores at the final checkpoints, whereas gain correlation compares their paired pretrained-minus-untrained differences. Each comparison contains six single-cell or four bulk recipes.

Task	Representation	Probe target	Score correlation	Gain correlation
SC Tabula	Mean-pooled	Presence	1.00	0.26
		Order	−0.03	0.49
		Value	−0.71	0.20
SC Adamson	Token-level	Order	−0.09	0.20
		Value	0.77	0.14
SC Norman	Token-level	Order	0.26	−0.09
		Value	0.77	−0.09
Bulk TCGA	Mean-pooled	Presence	−0.80	0.40
		Order	−0.80	0.80
		Value	0.60	0.80
Bulk PANACEA	Mean-pooled	Presence	0.60	−1.00
		Order	0.60	0.20
		Value	−0.80	0.20

PANACEA uses the ten-cell-line refit partition together with the same training-fitted input and target transformations used by the PCA–ridge predictor (Appendix B.2). Fold-level evidence is weighted by the number of evaluated treatment conditions, matching the biological-score aggregation.

For Adamson and Norman, each feature row is the mean 512-dimensional token representation of one gene over 1,024 training control cells sampled with seed 42, using only cells in which the corresponding token is present. The resulting common gene sets contain 5,000 and 4,998 genes, respectively. Targets are each gene's mean expression change across 52 Adamson or 102 Norman training interventions relative to the corresponding training controls, with intervention conditions weighted equally. These targets summarize training response patterns and do not reproduce GEARS prediction of unseen interventions. The contributing control cells may differ across recipes because token availability depends on the native input construction.

Estimator and selection scope. The final aggregate analysis uses the official fixed-point LogME estimator for classification and PANACEA and the ICML 2021 estimator for the genetic perturbation tasks, where we observed numerical-rank instability with the former. Recomputing classification and PANACEA with the ICML 2021 estimator preserves the mean and seed-level correlations and selected recipes. Early-checkpoint and fold-specific analyses use the ICML 2021 estimator throughout.

Averaging LogME across folds before selecting a recipe allows a sample's label to contribute to selection through another fold in which that sample belongs to the training set. We therefore additionally evaluate fold-specific selection, in which each test fold is scored using only LogME computed from its corresponding training partition.

Recipe selection within each training fold. For each fold and pretraining seed, we select the recipe with the highest training-fold LogME and evaluate that recipe at its final checkpoint on the corresponding test fold. For Tabula, selection is additionally performed separately within tissue and evaluation seed. When multiple recipes tie in LogME, their test performance is averaged. Table 16 compares this policy with expected performance under uniform random recipe selection and with a test-score oracle, using the original task-specific aggregation weights.

On PANACEA, LogME computed before pretraining selects recipes with higher held-out performance than LogME at 25k steps (0.111 versus 0.102). On TCGA, removing BULK VALUE reduces the selected macro-F1 to 83.50, below expected random selection among the remaining recipes (83.58). Expression baselines are not included among the selection candidates. For Tabula, only five-fold mean performance is stored; because the same recipe is selected across folds within each group, the LogME policy remains evaluable, but a fold-specific oracle cannot be recovered. Adam-

Table 16: Held-out performance after recipe selection using training-fold LogME. Columns 0-step through Final indicate the pretraining stage at which LogME is computed; the selected recipe is always evaluated at its final checkpoint. Random is expected performance under uniform random recipe selection. Oracle selects using test scores and is not available to a practical selection procedure. Classification scores are macro-F1 (%); PANACEA reports condition-wise Pearson correlation.

Task	0-step	25k	50k	Final	Random	Oracle
Tabula	82.42	83.29	83.29	83.29	77.03	—
TCGA	86.17	86.17	86.17	86.17	84.23	86.20
PANACEA	0.111	0.102	0.111	0.126	0.095	0.149

son and Norman do not provide the corresponding outer-fold structure and are excluded from this selection analysis.

Early-checkpoint comparison. All ten recipes and three pretraining seeds are evaluated at 25k and 50k optimizer steps, corresponding to 0.08/0.16 epochs for single-cell RNA-seq and 2.78/5.57 epochs for bulk RNA-seq. Expression probes use the same data, fitting procedure, and scoring protocol at every stage. Table 17 compares these stage-specific LogME and probe scores with biological performance measured at the final pretrained checkpoints. Downstream biological predictors are not refitted on the early checkpoints, and these correlations are not used to select checkpoints.

Table 17: Spearman correlation across recipes between stage-specific evaluation scores and biological performance at the final checkpoints. Presence, order, and value denote expression probes, using the representations specified in Figure 2.

Task	Measure	0-step	25k	50k	Final
Tabula	LogME	0.257	0.714	0.600	0.600
Tabula	Presence	1.000	0.486	0.657	1.000
Tabula	Order	0.086	0.200	−0.029	−0.029
Tabula	Value	−0.600	−0.543	−0.543	−0.714
TCGA	LogME	0.200	0.400	0.400	0.400
TCGA	Presence	−0.600	−0.200	−0.200	−0.800
TCGA	Order	0.200	−0.200	−0.400	−0.800
TCGA	Value	0.800	0.800	0.800	0.600
PANACEA	LogME	0.800	0.800	0.800	0.800
PANACEA	Presence	0.800	−0.400	−0.400	0.600
PANACEA	Order	−0.400	−0.400	0.000	0.600
PANACEA	Value	−1.000	−1.000	−1.000	−0.800
Adamson	LogME	−0.029	0.829	1.000	0.886
Adamson	Order	−0.143	−0.371	−0.371	−0.086
Adamson	Value	0.143	−0.029	0.257	0.771
Norman	LogME	0.029	0.771	0.771	1.000
Norman	Order	−0.029	−0.143	−0.143	0.257
Norman	Value	0.029	−0.314	−0.086	0.771

D.3 RNA QA CORRESPONDENCE WITH PROBE AND BIOLOGICAL RANKINGS

Table 6 correlates model scores with negative mean reference-answer NLL across recipes. For the single-cell models, cell-type classification and pooled presence probing induce identical recipe rankings, so their corresponding QA correlations should not be interpreted as independent evidence. For main-source identity questions, the classification correlation is 0.94, with a conditional 95% profile-bootstrap interval of $[0.54, 0.94]$ and a leave-one-recipe-out range of 0.90–1.00. For expression-interpretation questions, the corresponding correlation is 0.49, with interval $[-0.09, 0.83]$.

Intervals use 4,000 paired RNA-profile bootstrap draws that retain all questions from each sampled profile and use the same profile multiplicities across recipes. Tabula QA resampling is stratified by reference cell-type label. These intervals condition on the observed recipes and training seed and

Table 18: Cell-type-label LogME on the same 400 Tabula QA cells. Qwen columns use representations extracted before any answer token. RNA and Qwen features differ in dimensionality, so their raw LogME values should not be interpreted as information gained or lost between representation spaces.

Recipe	RNA	Qwen Random adapter	Qwen Aligned	Qwen Instruction-tuned
SC PRESENCE	0.8961	0.6019	0.7543	0.7881
SC IDENTITY	0.5993	0.3604	0.6923	0.7211
SC GPT	0.7556	0.3847	0.7423	0.7689
SC VALUE	0.6344	0.3502	0.6190	0.6511
SC BIN-CLASS	0.5105	0.1750	0.5499	0.6068
SC BIN-VALUE	0.5447	0.3192	0.4620	0.5003

therefore do not describe variability across retraining. The broader multiple-comparison analysis is summarized in Table 8.

D.4 CELL-TYPE-LABEL LOGME IN RNA AND QWEN REPRESENTATIONS

We score native 512-dimensional RNA cell representations and 2,048-dimensional Qwen representations at the final prompt token against the same 20 cell-type labels for 400 Tabula QA cells. Each label contributes 20 cells, and each prompt asks "Which cell type is this cell?" without conversation history or answer tokens. The cell-type labels are used only as LogME targets.

We compute LogME using the ICML 2021 estimator (You et al., 2021) from Qwen representations obtained under three adapter states: a reconstructed random-initialization Q-Former control, the aligned Q-Former, and the instruction-tuned Q-Former with Qwen LoRA. The RNA encoder remains frozen, so the native RNA representations are identical across these stages.

Spearman correlations between the RNA and Qwen LogME rankings are 0.943, 0.886, and 0.886 for the random-adapter, aligned, and instruction-tuned stages, respectively. The corresponding leave-one-recipe-out ranges are 0.90–1.00, 0.80–0.90, and 0.80–0.90. All six Qwen LogME scores increase after alignment and again after instruction tuning, but the strong cross-recipe rank agreement is already present under the random-adapter control. These comparisons use labels from the QA evaluation set and therefore do not establish prospective encoder-selection performance.