Effective Biological Representation Learning by Masking Gene Expression

Kian Kenyon-Dean[1] **Alina Selega**[1⋆] **Ihab Bendidi**[1,2,3⋆†] **Jordan M. Sorokin**[1†]
Luca Bertinetto[1] **David Errington**[1] **Hayley Donnella**[1†] **Oren Kraus**[1†]
[1]Recursion [2]Valence Labs [3]École Normale Supérieure PSL
kian.kd@recursion.com info@rxrx.ai

Abstract

RNA sequencing produces rich and diverse datasets of gene expression, offering compelling insights into cellular state and function that have many applications in drug discovery. Modeling such data is challenging due to inherent technical noise and experimental batch effects, as evidenced by many existing transcriptomic foundation models (FMs) underperforming relative to linear baselines. Such results raise the question of whether deep representation learning provides a distinct advantage over the direct use of raw transcript counts. Our work explores this by developing a new self-supervised model, TxFM, with a focus on inductive representation learning evaluations. TxFM employs a masked autoencoding approach tailored to diverse RNA-seq count data, and our ablation study empirically identifies crucial architecture configurations required for strong transfer performance. Additionally, we curate a public training corpus, DiverseRNA-1.4M, and find that TxFM trained on this curated dataset yields high-fidelity gene representations that outperform FMs trained on atlas-scale corpora over $100\times$ larger. Overall, our results indicate that inductive self-supervised learning is a viable modeling approach for transcriptomics representation, provided a careful synthesis of model architecture and training data curation.[1]

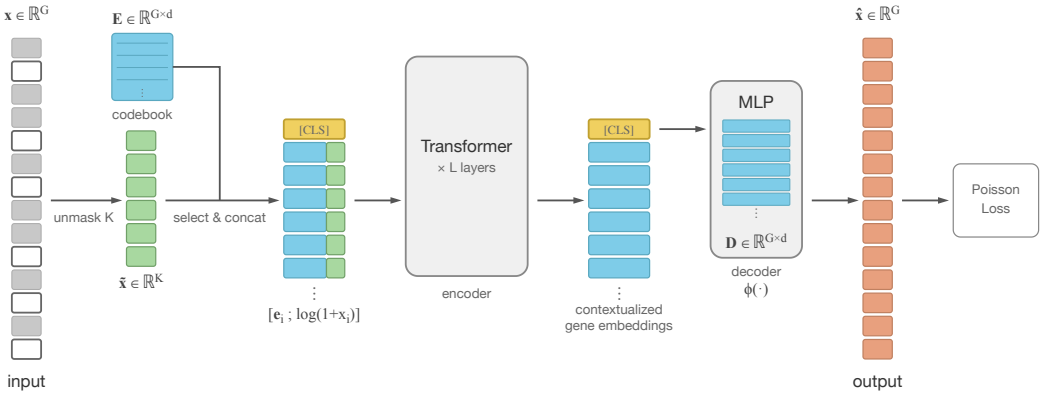


Figure 1: **Overview of our approach.** TxFM is a self-supervised learning (SSL) model trained for masked reconstruction of partially observed gene expression count data $\mathbf{x}$, from bulk and single-cell RNA-seq transcriptomic assays. We evaluate inductive representation quality of genetically perturbed input cells (CLS token), and the functional relationships recalled by the transformer encoder's gene codebook $\mathbf{E} \in \mathbb{R}^{G\times d}$, and the MLP decoder's gene reconstruction parameters $\mathbf{D} \in \mathbb{R}^{G\times d}$.

[1]⋆: equal contribution. †: work done while still employed at Recursion.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

1 Introduction

Deep representation learning is now standard in vision and language, yet its application to modeling gene expression has encountered various difficulties. Several recent benchmarking studies find that transcriptomic foundation models (FMs) fail to outperform simple linear baselines such as PCA on held-out tasks (Szałata et al., 2024; Boiarsky et al., 2024; Bendidi et al., 2024; Ahlmann-Eltze et al., 2025; Kedzierska et al., 2025; Csendes et al., 2025; Wenkel et al., 2026). These results admit two readings: either gene expression counts already carry most recoverable biological signal in their raw form, or current pretraining approaches are inadequate for transfer learning to new data.

This work explores the latter hypothesis. We develop **TxFM**, a masked autoencoder (MAE) for transcriptomics, and systematically investigate the recipe needed to produce representations that can surpass count-based baselines. Our investigation is organized around three axes.

Inductive representation learning. We focus on *inductive* SSL: the goal is to train a model whose representations transfer to cellular contexts unseen during training, rather than to precisely model the training data distribution. We benchmark various count-based baselines and 16 different FMs on three held-out genetic perturbation datasets featuring cell lines not seen during training (§ 4.1). Furthermore, we evaluate the intrinsic gene representations learned directly within FM parameters as a byproduct of SSL (§ 4.2), testing their ability to group functionally related genes (e.g., those encoding subunits of a protein complex). Unlike work leveraging pretrained protein embeddings (Lin et al., 2023; Adduri et al., 2025), TxFM does not incorporate external biological priors, allowing us to isolate what can be learned strictly from expression data.

Training data composition. We construct **DiverseRNA-1.4M**, a curated public corpus of 1.4 million bulk and single-cell RNA-seq samples with broad genetic coverage (Table 1). We find that TxFM and baselines trained on this data produce higher-signal gene representations than models trained on atlas-scale datasets over $100\times$ larger. This advantage holds true both at inference-time for unseen perturbed cells and for the intrinsic gene features learned in model parameters, suggesting that data curation is a fruitful lever for learning rich representations of perturbations and genes.

Architecture and reconstruction objective. TxFM is an asymmetric MAE. Unmasked input counts are mapped by a gene transformer encoder to a bottlenecked CLS-token representation, which an MLP decoder uses to reconstruct the full expression vector (Figure 1, § 3). Following standard efficiency conventions (He et al., 2022), only unmasked tokens pass through the encoder. We introduce a domain-specific reconstruction loss based on the Poisson likelihood, stabilized by a novel *rectified tanh* activation that constrains decoder outputs to the data support while preserving gradient flow.

Our ablation study (§ 4.4) empirically identifies each impactful design choice in TxFM, offering practical guidance for building transferable gene expression models. Beyond inference performance on unseen data, TxFM's learned decoder gene weights achieve state-of-the-art recovery of known biological relationships without direct supervision. Interestingly, we find that these relationships concentrate on lower-dimensional manifolds in gene parameter matrices—a phenomenon consistent across pretrained transformers and with the field's understanding of functional gene programs (Heimberg et al., 2016). Altogether, these results motivate future research into mining pretrained models to discover new mechanisms, biomarkers, and drug targets (Schenone et al., 2013).

2 Related Work

Existing FMs for transcriptomics use SSL but address the high-dimensional, non-sequential nature of gene expression data in different ways. scBERT (Yang et al., 2022), scGPT (Cui et al., 2024), and GeneFormer (Theodoris et al., 2023) employ masked language modeling. scFoundation (Hao et al., 2024) embeds raw counts instead of binned or ranked values. UCE (Rosen et al., 2023) uses ESM-2 (Lin et al., 2022) embeddings to represent genes. scTab (Fischer et al., 2024) employs an architecture tailored to tabular data. CellPLM (Wen et al., 2023) and scCello (Yuan et al., 2024) jointly model other modalities, and Cell2Sentence (Rizvi et al., 2025) casts gene expression as linguistic sentences with genes ordered by decreasing expression. AIDO.Cell (Ho et al., 2024), scPrint (Kalfon et al., 2025), and TranscriptFormer (Pearce et al., 2025) scaled pre-training datasets to 50–100 million cell atlases. Tahoe-100M (Gandhi et al., 2025) is a large small-molecule perturbational dataset with multiple recent models pre-trained on it, including the dataset's companion FM Tahoe-x1, GeneJEPA (Litman et al., 2025), and STATE (Adduri et al., 2025).

Table 1: Composition of the default curated public TxFM SSL training dataset. sc: single-cell.

dataset summary	mode	# samples	# genes
Glioblastoma (Ruiz-Moreno et al., 2022)	sc	504,929	21,310
K562 CRISPRi *phenoprints* (Replogle et al., 2022)	sc	502,080	8,248
MixSeq cancer cell lines (McFarland et al., 2020)	sc	102,205	15,438
sci-Plex compounds (Srivatsan et al., 2020)	sc	99,300	18,486
Gastric metaplasia (Nowicki-Osuch et al., 2023)	sc	88,399	16,445
Tumor microenvironment (Guimarães et al., 2024)	sc	71,585	17,619
Breast cancer (Wu et al., 2021)	sc	31,542	24,712
TCGA (Weinstein et al., 2013)	bulk	23,733	19,594
GTEX (Consortium, 2020)	bulk	10,526	36,695
Total composite dataset: **DiverseRNA-1.4M**	mixed	1,434,299	44,349

However, benchmarking studies have found many FMs unable to generalize or outperform simple baselines across a range of tasks despite their scale (Boiarsky et al., 2024; Bendidi et al., 2024; Kedzierska et al., 2025; Wong et al., 2025; Ahlmann-Eltze et al., 2025). Indeed, increasing data size does not necessarily translate to improved performance of SSL models (DenAdel et al., 2024).

Masked autoencoders (MAEs). Originally designed for images, MAE (He et al., 2022) offers a self-supervised learning regime with reliable performance and transfer to downstream biological tasks (Kraus et al., 2024; Kenyon-Dean et al., 2025; Kim et al., 2025; Richter et al., 2025). TxFM belongs to the MAE model family but incorporates several adaptations for count-based gene expression data, in particular an MLP decoder using only the final encoder CLS token (rather than a token-wise transformer decoder), and other design choices described below.

3 Approach

Let $\mathbf{x} \in \mathbb{N}^G$ be a gene expression count vector for a single-cell (sc) or bulk RNA-seq sample with G profiled genes. We seek low-dimensional embeddings $\mathbf{e} \in \mathbb{R}^d$ ($d \ll G$) that isolate biological signal from high-dimensional technical noise, enabling downstream analyses such as comparing perturbations and inferring gene relationships.

We adopt an SSL framework based on masked reconstruction. TxFM consists of two main components: the **encoder**, a transformer that takes masked gene expression $\mathbf{x}$ as input to produce a sample-level representation $\mathbf{e}$; and the **decoder**, a lightweight MLP that takes $\mathbf{e}$ as input and predicts a full expression vector $\hat{\mathbf{x}} \in \mathbb{R}^G$. Unlike language modeling where word order is critical, a gene expression profile is an unordered set of measurements of the abundance of individual RNA molecules taken simultaneously, akin to a bag-of-words. For this reason, we do not model gene expression as a sequence.

Training dataset. We train our default model on a curated collection of public transcriptomic datasets with both single-cell and bulk RNA-seq samples, selected with a focus on oncology and perturbational biology. We call this dataset containing $\sim$1.4 million samples *DiverseRNA-1.4M* (Table 1). For the large-scale perturbational dataset of CRISPRi-edited K562 cells (Replogle et al., 2022), we apply a *phenoprint* curation strategy that filters for perturbations with distinct transcriptional profiles (see § B.1) to enrich for biologically meaningful variation; a strategy shown by Kenyon-Dean et al. (2025) to work effectively when training MAEs for microscopy. For the other single-cell datasets, we retain only genes expressed in at least 1,000 cells and cells expressing at least 2,000 genes, yielding an aggregate union of 44,349 unique genes in the training data.

On dataset heterogeneity. As different transcriptomic datasets contain different sets of sequenced genes, our DiverseRNA-1.4M has different degrees of gene overlap between its composite datasets. To handle these feature differences, and, importantly, to enable cross-dataset integration, we adopt the following strategy instead of zero-padding: TxFM learns representations for every gene in the vocabulary, but only unmasks and applies reconstruction loss on the genes that are actually measured in an individual sample, as described below.

Preprocessing. Each input gene expression vector $\mathbf{x}$ is normalized to have a total count of $\sum_{i=1}^{G} \mathbf{x}_i = L = 10^5$ (library size normalization), then log-transformed after adding a pseudocount to accommodate zero counts: $\mathbf{x}_i = \log(\mathbf{x}_i + 1)$, or log1p. This is a standard normalization procedure for RNA-seq analysis that enables comparisons between samples with varying numbers of total mapped counts (Conesa et al., 2016).

Masking strategy. To enable self-supervision, we uniformly at random sample a subset $\mathbf{k} \subseteq \{1, \ldots, G\}$ of size K and only reveal the values $\{\mathbf{x}_i : i \in \mathbf{k}\}$ to the encoder. The remaining gene expression values are not seen by the encoder. Sampling is uniform across all *observable* genes within each sample; e.g. if $K = 2048$, a cell from the K562 dataset (Replogle et al., 2022) has an effective mask ratio of $\sim$75% since there are 8,248 possible genes to sample, while a sample from GTEX (Table 1) would have an implicit mask ratio of $\sim$94% ($1 - \frac{2048}{36{,}695}$).

Model. Each unmasked gene indexed with $i \in \mathbf{k}$ is embedded by *concatenating* a learnable gene embedding $\mathbf{E}_i \in \mathbb{R}^{d-1}$ with its preprocessed expression value $\mathbf{x}_i$, forming: $\text{token}_i = [\mathbf{E}_i; \mathbf{x}_i] \in \mathbb{R}^d$.

A learnable CLS token is appended to this set and all $K + 1$ tokens are passed through a transformer encoder, relating every unmasked gene to each other through dense self-attention. The transformer's output embedding of the CLS token $\mathbf{e}$ is then passed through an MLP (count) decoder: $\hat{\mathbf{x}} = \phi(\text{MLP}(\mathbf{e}))$, $\hat{\mathbf{x}} \in \mathbb{R}^G$, where ϕ is a novel *rectified tanh* activation function we introduce and define on the model's predicted gene expression logits z as:

$$\phi(z) = \alpha \text{ReLU}\left(\tanh\left(\frac{z}{\alpha}\right)\right), \quad \alpha = \log(L + 1). \tag{1}$$

We designed this activation function to respect the non-negative, count-based nature of the data. Unlike pure ReLU, this activation is bounded and asymptotically approaches the normalized upper bound of the data, setting $\alpha = \log(L + 1)$, preventing divergence by naturally dampening gradients as the predicted count increases (§ C.1, Figure 10).

Loss function. TxFM can be trained to minimize any reconstruction loss. Our default architecture uses a Poisson-based reconstruction loss function on the model's prediction $\hat{\mathbf{x}}$ and the gene expression in the log-normalized training data $\mathbf{x}$ for each sample (derivation in § A.3):

$$\mathcal{L}_{\text{Poisson}}(\mathbf{x}, \hat{\mathbf{x}}) = \frac{1}{G} \sum_{i=1}^{G} (e^{\hat{\mathbf{x}}_i} - \hat{\mathbf{x}}_i e^{\mathbf{x}_i}). \tag{2}$$

By combining this loss with the rectified tanh activation, we constrain the hypothesis class to a library-bounded link function, where the Poisson rate $\lambda \in [1, L + 1)$ matches the empirical support of the library-normalized target data with pseudocounts. The rectification in $\phi(z)$ clamps negative gene logits to the minimum rate (λ=1) without an explicit mixture model of zero-inflation.

On the match between input data, activation, and loss function. Library size normalization removes sequencing depth variation while preserving gene rankings. While Poisson is a discrete distribution defined on natural numbers as support, the Poisson likelihood remains a valid, convex, and differentiable objective for continuous targets. Thus, the optimal reconstruction under the Poisson loss corresponds to the same gene expression patterns regardless of whether targets are raw or normalized. Unlike MSE or SmoothL1, the Poisson loss generates gradients proportional to the relative prediction error, which prioritizes reconstruction of genes with low-to-moderate expression (which comprise the majority of scRNA-seq data).

While RNA-seq is often overdispersed (Love et al., 2014), Poisson may be a better SSL training objective than negative binomal (NB) and zero-inflated NB (ZINB), as these display vanishing gradients for overestimation errors on high-dispersion genes, while Poisson retains gradient signal (§ A.4). We hypothesize that *not* modeling gene dispersion explicitly can force the encoder to absorb biological variation into the CLS token embedding, rather than reducing the loss by increasing the dispersion parameters of a NB model. We interrogated this hypothesis by regressing gene mean and variance against TxFM's encoder parameters, and we found that the model encoded both moments into its latent space (§ A.5). Further, as the gradient of the loss passes through our rectified tanh activation function during backpropagation, the sech^2 factor attenuates the gradient for highly expressed genes, focusing the model's capacity on low-to-moderate expression levels where the Poisson approximation is more appropriate (§ A.3).

Table 2: **Inductive SSL.** *Genetic perturbation representation learning results from model inference.* Benchmarks on RPE1, HEPG2, and Jurkat cell lines, measuring linear separability of perturbations in cellular replicates across batches in an inductive SSL setting (data is unseen during pretraining). We report linear probing performance (**lin**; mean of top-1 and top-5 accuracy of linear classifier predicting perturbation identity, evaluated on held-out batches) (Bendidi et al., 2024), mean ± std across 5 random seeds. ***Overall*** is the mean of all six downstream tasks (batch effect correction, linear probing, latent space organization, perturbation consistency, biological relationship recall, and latent space interpretability), averaged across the three cell lines. Best is in **bold**. See § B.1.1 for task descriptions, § D for score breakdown by metric.

		Linear probe (lin)			
model	train data, # samples	**RPE1**	**HEPG2**	**Jurkat**	***Overall***
Input counts baselines					
random label shuffle	n/a	0.9 ± .01	0.7 ± .01	0.5 ± .00	*19.43*
(Lib+Log)Norm+1k HVG	evaluation data	23.7 ± .02	15.0 ± .04	9.7 ± .01	*34.46*
(Lib+Log)Norm+5k HVG	evaluation data	26.7 ± .03	19.2 ± .05	13.1 ± .01	*35.08*
Raw data	n/a	36.7 ± .06	23.2 ± .74	20.5 ± .05	*35.27*
(Lib+Log)Norm	n/a	30.1 ± .03	21.5 ± .05	16.7 ± .02	*36.29*
Atlas-scale SSL (different data)					
Geneformer-v2	Genecorpus, 104M	4.1 ± .01	2.2 ± .00	3.0 ± .00	*19.89*
GeneJEPA	Tahoe, 100M	4.7 ± .00	2.6 ± .00	3.1 ± .01	*23.01*
scTab	Atlases, 22M	6.7 ± .01	4.6 ± .02	3.7 ± .01	*24.80*
CellPLM	Tumor atlases, 11M	7.3 ± .00	5.1 ± .02	4.3 ± .00	*24.84*
UCE	Atlases, 36M	7.4 ± .00	5.2 ± .01	5.0 ± .00	*26.44*
scCello	Atlases, 22M	14.0 ± .01	8.5 ± .03	7.3 ± .01	*27.04*
scPrint-L	Atlases, 50M	16.3 ± .02	3.0 ± .00	8.4 ± .01	*27.49*
scGPT	Atlases, 33M	12.6 ± .01	7.2 ± .01	7.7 ± .01	*28.96*
Tahoe-x1-70M	Tahoe+atlases, 266M	17.2 ± .00	9.5 ± .02	9.6 ± .01	*30.99*
AIDO.Cell-100M	Atlases, 50M	27.7 ± .02	17.0 ± .03	15.1 ± .00	*34.50*
Cell2Sentence	Web+atlases, 50M	26.7 ± .00	16.1 ± .03	13.9 ± .00	*35.44*
STATE-SE	Tahoe+atlases, 170M	29.1 ± .00	17.2 ± .00	17.2 ± .00	*36.43*
Atlas-scale SSL (same data)					
TranscriptFormer-Sapiens	TF-Sapiens, 57M	18.7 ± .01	12.2 ± .02	10.8 ± .01	*32.12*
TxFM-B (ours)	TF-Sapiens, 57M	27.5 ± .00	17.8 ± .04	16.5 ± .01	*35.62*
Curated data SSL (same data)					
scVI-L (VAE)	DiverseRNA, 1.4M	22.1 ± .01	14.2 ± .03	13.0 ± .01	*32.85*
PCA	DiverseRNA, 1.4M	16.1 ± .01	11.1 ± .05	7.6 ± .00	*33.54*
Linear autoencoder	DiverseRNA, 1.4M	24.2 ± .03	16.6 ± .07	14.1 ± .05	*35.38*
Ridge PCA	DiverseRNA, 1.4M	28.2 ± .07	20.2 ± .09	16.5 ± .09	*36.74*
TxFM-B (ours)	DiverseRNA, 1.4M	**37.6** ± .02	**25.9** ± .07	**23.8** ± .01	***39.11***

4 Evaluation

We focus evaluation on two primary TxFMs, each using a Base transformer backbone and 768 model dimensions: (1) a TxFM-B trained on our curated DiverseRNA-1.4M (along with several baselines trained on the same data, including a 1024-dimensional scVI (Lopez et al., 2018), § B.1.2); (2) a TxFM-B trained with 4× more compute on the 40× larger atlas-scale TF-Sapiens 57 million single-cell corpus used to train the Transcriptformer model (Pearce et al., 2025). Hyperparameters in § A.1.

4.1 Inference-time cell perturbation representation learning

Inductive SSL results. Table 2 benchmarks representations of genetically perturbed cells across 16 FMs and strong baselines on three held-out datasets (each with hundreds of thousands of input cells spanning 2,392 CRISPRi genetic perturbations) according to the methodology proposed by Bendidi et al. (2024). We focus on the linear probing benchmark, and report performance on the remaining five tasks in § D, where relative performance trends are similar. The linear prob-

ing benchmark measures how accurately batch-corrected cell embeddings can be linearly classified into their corresponding perturbations in batches held-out from the linear probe fit. The *overall* aggregates performance across all six tasks (described in § B.1.1); a higher overall score indicates more biologically meaningful organization of samples and perturbations into discriminative clusters while preserving structural interpretability of underlying gene expression changes. Additional cell representation benchmarks on non-perturbational datasets are reported in § B.2.4.

Models versus strong baselines. We first note that standard count-data preprocessing (library size normalization with log transform, or highly variable gene selection) establishes a competitive baseline that many atlas-scale FMs fail to exceed. Against this backdrop, TxFM-B trained on DiverseRNA-1.4M achieves the highest linear probing accuracy and overall score on every held-out dataset, outperforming all competing FMs including STATE-SE (Adduri et al., 2025). Notably, it achieves this despite having nearly $4\times$ fewer parameters, training on $100\times$ less data, and using no auxiliary biological priors such as pretrained ESM2 protein embeddings.

Factors driving TxFM's performance. Two controlled comparisons shed light on factors contributing to improved performance. First, TxFM-B trained on the curated DiverseRNA-1.4M substantially outperforms the same architecture trained on the TF-Sapiens 57M cell atlas (35.62 vs 39.11), suggesting that atlas-scale volume alone is not sufficient for high-signal representation of perturbational data inputs. Second, when controlling for training data, TxFM outperforms comparison models on both DiverseRNA-1.4M and TF-Sapiens 57M, indicating that architectural choices are impactful when holding the training data constant.

Ablating impact of dataset composition on performance. Our curation of DiverseRNA-1.4M includes cells from another cell type (K562) sequenced using a similar assay protocol as the evaluation data (Replogle et al., 2022), raising a potential concern about data leakage. We address this with a controlled ablation when training TxFM-B (§4.4): removing all K562 cells yields an overall score of 36.73; including only the 72,000 K562 control cells (no perturbed cells) improves this to 38.22; the full DiverseRNA-1.4M (including phenoprint-curated perturbed K562 cells) achieves 39.11. All three configurations outperform every competing FM, demonstrating that TxFM's advantage is not attributable to perturbational overlap with the evaluation data, but rather reflects the broader curation of DiverseRNA and the model architecture. We also note that STATE and Tahoe-x1 may benefit from HEPG2 cells in Tahoe training data.

Transductive SSL results. In Table 3, we evaluate a transductive setting where TxFM is SSL fine-tuned directly on the evaluation dataset, using the same masked reconstruction objective (§A.1.1). This reflects a common practitioner workflow where generalization beyond the target data is not the primary goal, instead opting for a bespoke model that learns dataset-specific structure.

Table 3: **Transductive SSL.** Linear probing and overall performance for PCA, scVI, and SSL-fine-tuned (f.t.) TxFM-B models trained separately on RPE1, HEPG2, Jurkat evaluation datasets.

	Linear probe (lin)			
model	**RPE1**	**HEPG2**	**Jurkat**	***Overall***
scVI	19.7 $\pm$ .02	12.6 $\pm$ .04	10.0 $\pm$ .01	*35.30*
PCA	32.0 $\pm$ .05	23.9 $\pm$ .09	15.6 $\pm$ .02	*38.72*
TxFM-B f.t.	**42.1** $\pm$ .01	**29.3** $\pm$ .04	**25.6** $\pm$ .03	***41.21***

SSL fine-tuning the default TxFM-B (pre-trained on DiverseRNA-1.4M) outperforms both PCA and scVI fitted to the same target data, yielding a 5–19% relative improvement to an overall score of 41.21. Notably, the *inductive* TxFM-B (overall 39.11, Table 2) outperforms transductive PCA (overall 38.72), meaning that a forward pass through the pre-trained encoder produces cell representations competitive with a traditional model fit on the evaluation dataset. In Figure 3 (§A.1.1), we visualize reconstruction loss epoch-wise when training a TxFM-B from scratch on RPE1 compared to SSL finetuning. Validation loss is already near optimal performance at the start of SSL fine-tuning, thus demonstrating the efficacy of pre-training.

4.2 Gene representation learning in model parameters

Many transcriptomics models learn static non-perturbational gene representations as a byproduct of training. Transformers often include a learnable codebook of gene token features, while models with full expression decoders (like TxFM and scVI (Lopez et al., 2018)) learn an additional matrix

Table 4: **Whole-genome known gene-gene relationship recall from learned gene parameters** for 155,390 relationships combined over 5 databases (CORUM, hu.MAP, Signor, Reactome-PPI, StringDB). We report the % of relationships recalled at 95th percentile cosine similarity threshold, ± recall std from 10 runs with different null distributions sampled from random pairs of protein-coding genes (Celik et al., 2024). Each gene matrix is benchmarked before (raw centered) and after post-processing to its best-performing PCA-reduced n components. Best is in **bold**.

model	layer	train data, # samples	**recall** (raw)	**recall** (PCA)	n
random vectors	n/a	n/a	4.9 ± .02	4.9 ± .02	n/a
Geneformer-v1	codebook	Genecorpus, 30M	18.2 ± .16	20.2 ± .11	512
Geneformer-v2	codebook	Genecorpus, 104M	19.4 ± .19	20.2 ± .11	256
STATE-SE	projection	Tahoe+atlases, 170M	22.0 ± .12	31.3 ± .19	256
scGPT	codebook	Atlases, 33M	31.4 ± .13	31.8 ± .16	256
Tahoe-x1 70M	codebook	Tahoe+atlases, 266M	29.2 ± .10	29.6 ± .08	512
Tahoe-x1 1B	codebook	Tahoe+atlases, 266M	24.0 ± .18	25.7 ± .15	256
Tahoe-x1 3B	codebook	Tahoe+atlases, 266M	32.5 ± .09	35.9 ± .15	256
TranscriptFormer	codebook	TF-Sapiens, 57M	23.4 ± .12	32.9 ± .21	256
TxFM-B (ours)	codebook	TF-Sapiens, 57M	36.1 ± .09	38.3 ± .10	128
TxFM-B (ours)	decoder	TF-Sapiens, 57M	40.1 ± .13	40.6 ± .09	512
PCA (transpose fit)	n/a	DiverseRNA, 1.4M	29.2 ± .15	29.4 ± .16	1024
scVI-L	codebook	DiverseRNA, 1.4M	23.6 ± .15	22.3 ± .12	1024
TxFM-B (ours)	codebook	DiverseRNA, 1.4M	26.3 ± .06	32.6 ± .23	64
scVI-L	decoder	DiverseRNA, 1.4M	40.4 ± .24	40.9 ± .24	1024
TxFM-B (ours)	decoder	DiverseRNA, 1.4M	**42.7** ± .15	**43.9** ± .20	256

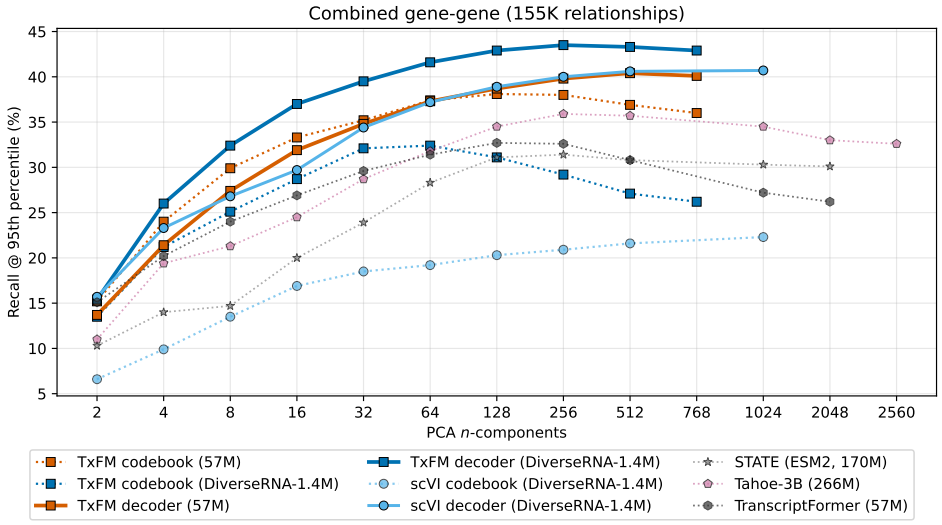


Figure 2: Gene relationship recall as a function of PCA-postprocessing a model's *learned gene embedding parameters* to n-components (codebook tokens and/or decoder reconstruction weights).

of gene reconstruction weights. Other models, such as UCE (Rosen et al., 2023) and STATE-SE (Adduri et al., 2025), bypass this by leveraging frozen ESM2 protein structure embeddings. As visualized in Figure 1, TxFM learns both gene codebook features and decoder gene weights from scratch in matrices **E** and **D**, respectively.

Table 4 compares 15 different sources of gene representations from the learned parameters of different models. We report gene-gene relationship recall (Celik et al., 2024; Kraus et al., 2024; Kenyon-Dean et al., 2025) to determine how well different models intrinsically capture protein-protein interactions, functional connections, and other known relationships annotated in public databases. We

compute recall on the model's immediate centered gene matrix, before and after post-processing it with PCA to the optimal performing lower dimensionality (Figure 2).

Recall performance comparison. Our default model, TxFM-B trained on DiverseRNA-1.4M, obtains the highest recall with its decoder's gene parameters both before and after PCA post-processing (42.7% / 43.9%); scVI's decoder trained on the same data performs second best (40.4% / 40.9%). The PCA baseline (fit on the transpose of the same training data) underperforms compared to scVI and TxFM, indicating that non-linear models can capture more relationships from gene expression data. Codebooks and decoders from various models trained on atlas-scale data underperform compared to training on DiverseRNA-1.4M, while TxFM trained on TF-Sapiens 57M has higher recall both from its codebook and decoder compared to each other atlas-scale model.

On the impact of PCA-postprocessing. Interestingly, we find that PCA-postprocessing the gene embeddings to a lower dimensionality often provides significant improvements, especially for transformer codebook representations (Figure 2). For example, PCA-reducing STATE-SE's 2048-dimensional projection of frozen ESM2 embeddings down to 256 dimensions improves recall by 42% (22.0→31.3), and Transcriptformer's gene embeddings similarly improve by 41% (23.4→32.9). TxFM's gene representations also benefit from such post-processing, while scVI (a VAE MLP) performs best with full dimensionality. In § A.2 (Figure 5) we evaluate how relationship recall develops across all 5 databases separately, epoch-wise during training. The TxFM decoder learns gene-gene relationships quickly, within the first 50 epochs, while the codebook features gradually improve up to epoch 150, underperforming the decoder on every database except Signor (Perfetto et al., 2016), similar to scVI.

Taken together, these results show that many fundamental biological relationships emerge naturally in lower-rank manifolds of high-dimensional transformer gene representation spaces, recoverable through simple dot products, as a byproduct of SSL training on transcriptomics data.

4.3 Comparing TxFM's learned gene parameters to sequence-based representations

Table 5 evaluates gene embeddings made by three sequence-based models, compared to the default TxFM-B decoder gene matrix (§ 4.2). EVO2 (Brixi et al., 2026) is a nucleotide-based sequence model that can create 4096-dimensional gene embeddings by passing each gene's corresponding DNA nucleotide sequence through the model; we used the coding region for each gene. Orthrus (Fradkin et al., 2026) is a nucleotide-based sequence model trained on mature RNA transcript-level sequences, creating 512-dimensional embeddings. ESM2 (Lin et al., 2023) operates on amino-acid sequences to learn evolutionary biology, and can similarly be evaluated by passing each gene's corresponding amino acid sequence to produce 2560-dimensional gene embeddings.

The gene-gene relationship benchmarking in Table 5 shows that these structure-based gene representations *also* benefit substantially from PCA-post processing to a lower dimensionality. TxFM's learned gene parameters substantially outperform each of these models' inference-time gene embeddings on this task. In general, sequence-based models appear to capture more structural relationships such as paralog families, whereas TxFM's decoder matrix captures more pathway biology annotated in the benchmark databases.

model	**rec.** (raw)	**rec.** (PCA)	n
EVO2	12.4 $\pm$ .06	21.3 $\pm$ .14	1024
Orthrus	14.3 $\pm$ .07	21.9 $\pm$ .13	64
ESM2	14.3 $\pm$ .07	29.7 $\pm$ .15	128
TxFM-B	**42.7** $\pm$ .15	**43.9** $\pm$ .20	256

Table 5: Gene relationship recall and optimal PCA n-components using sequence-based models to create gene representations via inference, compared to the fixed TxFM-B decoder.

4.4 Ablation of the main properties of TxFM

Table 6 presents our ablation study, inspired by He et al. (2022), on TxFM's architecture and training data to understand the main drivers contributing to performance improvements. We evaluate TxFM as an encoder by measuring perturbation consistency across the three evaluation datasets unseen during training (one of the six benchmarks used for § 4.1), and relationship recall of the post-PCA non-perturbational gene features captured in TxFM's learned parameters (as per § 4.2).[2]

[2]If not otherwise specified, TxFM's settings are: library size normalization followed by log1p count preprocessing, $K = 2048$ unmasked genes, uniform random mask sampling, a 4-layer MLP decoder, rectified tanh count activation function

Table 6: **TxFM architecture and data ablation experiments**. We report total perturbation consistency on RPE1, Jurkat, and HEPG2 combined (perts) and whole-genome gene-gene relationship recall captured in a TxFM's learned parameters: (enc)oder gene tokens and (dec)oder gene weights. Default TxFM architecture is marked in gray. Performance is significantly different (if $p_{adj} < 0.05$, § C) from Default when indicated with $\star$. Best performance in an ablation is in **bold**.

loss	perts	enc	dec
MSE	27.2*	31.1*	41.8*
SmoothL1	23.4*	23.2*	38.1*
Cross-entropy	31.2*	30.9*	26.9*
Poisson	**37.3**	32.6	**43.9**
NB	32.1*	29.1*	41.7*
ZINB	31.1*	**34.2***	32.1*

(a) **Reconstruction loss.** Our Poisson-based loss function significantly improves performance.

preprocessing	perts	enc	dec
log1p	27.9*	25.7*	42.2*
LibNorm, log1p	**37.3**	**32.6**	**43.9**

(b) **Count preprocessing.** Normalizing count data for library size helps to learn high-quality perturbational and non-perturbational gene representations.

# unmasked	perts	enc	dec
512	31.9*	26.8*	40.2*
1024	35.2*	31.9*	**44.3**
2048	**37.3**	**32.6**	43.9
4096	35.6*	25.3*	42.8*

(c) **Mask ratio.** Training with 2048 unmasked gene tokens ($\sim$90% mask ratio) yields strong performance, despite 4096 using much more training compute.

temperature τ	perts	enc	dec
-1	31.3*	32.5	42.7*
-0.5	34.4*	**34.3***	43.6
0 (uniform)	37.3	32.6	**43.9**
0.5	**37.6**	28.9*	42.0*
1.0	36.4	25.7*	40.3*

(d) **Masking strategy.** Masking genes uniformly at random offers a good performance trade-off vs frequency-weighted masking.

decoder	perts	enc	dec
linear	36.5	32.3	43.6
1-layer MLP	36.0	32.3	43.8
4-layer MLP	**37.3**	**32.6**	**43.9**
8-layer MLP	36.5	32.0*	43.6

(e) **Decoder depth.** A linear or MLP decoder works effectively with minimal significant differences.

activation	perts	enc	dec
ReLU (diverges)	34.4*	26.7*	43.9
clamped ReLU	34.6*	32.4	**44.3**
rect. tanh	**37.3**	**32.6**	43.9
softmax	27.6*	16.2*	15.3*

(f) **Decoder count activation.** Our rectified tanh activation function improves inference representations.

backbone	perts	enc	dec
-S (57M)	33.2*	29.2*	43.9
-B (159M)	**37.3**	**32.6**	**43.9**
-L (403M)	37.2	28.2*	43.1*

(g) **Encoder backbone.** Scaling TxFM to Base architecture on DiverseRNA-1.4M performed best on these tasks.

train data (# samples) [compute]	perts	enc	dec
Bulk RNA-seq only (33k)	21.2*	11.6*	33.5*
DiRNA w/o bulk RNA (1.4M)	34.6*	33.1	42.4*
DiRNA w/o K562 cells (932K)	31.4*	31.5*	42.0*
DiRNA w/ K562 controls (1.0M)	34.3*	32.0*	42.2*
DiRNA w/ curated K562 (1.4M)	**37.3**	32.6	43.9
DiRNA w/ full K562 (2.8M)	35.6*	31.2*	**44.9***
DiRNA w/ full K562 (2.8M) [2x]	36.7	29.8*	44.7*

(h) **Dataset curation.** A phenoprints-oriented data curation strategy of DiverseRNA-1.4M is an effective way to train TxFM-B.

(a) Reconstruction loss. We experimented with 6 loss functions (§ A.3). Those based on the negative binomial (NB) distribution learn additional dispersion parameters per gene, and the zero-inflated (ZI) ZINB model also modeled gene- and sample-specific ZI probabilities. We also implemented a rank-based cross-entropy objective, similar to scGPT (Cui et al., 2024), with discretized, binned targets. Our Poisson-based loss function (Equation 2) empirically yields significant improvements over the alternatives. While the ZINB variant has an improved codebook, its decoder suffers.

(b) Count preprocessing. Similar to pixel normalization and self-standardization used for image MAEs (He et al., 2022; Kraus et al., 2024), we confirm that the standard practice of normalizing inputs by library size (Conesa et al., 2016) is important to achieve good performance.

(c) Mask ratio. TxFM's effective masking ratio is dataset-dependent on different numbers of sequenced genes in each dataset (Table 1), hence we describe this ablation in terms of the exact num-

(Equation 1), Poisson reconstruction loss (Equation 2), Base transformer architecture, trained on DiverseRNA-1.4M (Table 1) for 200 epochs, near 1,000 H100 GPU hours per training run. See § A.1, 7, C for hyperparameters and additional details.

ber of genes *unmasked* during SSL training. Unmasking 2048 genes works most effectively on the DiverseRNA-1.4M dataset, while unmasking 1024 only underperforms slightly but requires significantly less training compute. On this dataset, 2048 unmasked genes entails a nearly 90% masking ratio, based on the average number of genes per training sample.

(d) Masking strategy. We also tested frequency-weighted masking based on a gene's training set sparsity. Let m_i be the number of training samples where gene i has nonzero expression, and define sampling weights $w_i \propto m_i^\tau$ for a temperature τ. Positive τ preferentially unmasks frequently expressed genes, while negative τ favors rarely expressed genes. In practice, uniform masking ($\tau = 0$) was simplest and gave the best overall trade-off, so we use it by default.

(e) Decoder depth. We assess decoder complexity by comparing a linear mapping to varying-depth MLPs with residual connections. Results indicate minimal sensitivity to depth, confirming that the encoder captures a sufficiently rich representation in the CLS token to enable reconstruction.

(f) Decoder count activation. Our proposed *rectified tanh* activation (Equation 1) significantly improves perturbation consistency compared to three alternatives.

(g) Encoder backbone. We ablate encoder capacity by training Small (-S), Base (-B), and Large (-L) variants on DiverseRNA-1.4M. Results indicate a performance plateau on this dataset: moving from -S to -B improves both inference performance and gene token representations, but further scaling to -L did not yield inductive gains despite lower reconstruction loss (§ A.1.2).

(h) Dataset curation. We vary the curation of DiverseRNA's largest subset, the Replogle et al. (2022) K562 CRISPRi dataset (∼1.9M cells), alongside the inclusion of bulk RNA-seq data. Evaluating seven distinct strategies (§ C.2), we find that including K562 data improves overall performance even when restricted to only 72,000 unperturbed control cells. Our default *phenoprint* curation strategy, which prunes ∼75% of the K562 cells with less discriminative signal, effectively improves perturbation representations. Interestingly, including just 33k *bulk* RNA-seq samples from TCGA and GTEX improves both *single-cell* perturbation consistency and decoder parameter recall. Further, removing these 33,000 bulk samples from training degrades performance approximately as much as removing all 450,000 perturbed K562 cells. This highlights a powerful synergy unlocked by using SSL to jointly model single-cell and bulk transcriptomic modalities.

5 Conclusion

We present TxFM, a self-supervised masked autoencoder designed for effective transfer learning from transcriptomic count data, and DiverseRNA-1.4M, a curated public dataset designed for training SSL transcriptomics models, enabling faster iteration on fewer computational resources than large atlas-scale data requirements.

Training TxFM on DiverseRNA-1.4M consistently yields higher-signal gene representations than models trained on much larger atlas-scale datasets, for both inference-time perturbed cell embeddings and the gene features learned within model parameters. Architecturally, we find that our high ∼90% masking ratio, Poisson-based reconstruction loss, and a rectified tanh activation were each individually necessary for TxFM's strong transfer performance, resulting in the only model to meaningfully surpass normalized input counts on cellular perturbation representation tasks.

Lastly, we show that many transcriptomic models generally learn gene parameters that recover many known biological relationships without supervision. TxFM's learned decoder weights achieve the highest gene-gene relationship recall (recovering the most known pathways, protein complexes, etc.) among transcriptomic-based and sequence-based models evaluated, suggesting strong potential for novel biological relationship discovery in service of identifying new drug targets.

Limitations and reproducibility. Each TxFM checkpoint evaluated in this work was trained and evaluated exclusively on freely accessible public data, including our curated DiverseRNA-1.4M and the uncurated TF-Sapiens 57M dataset. While our oncology-oriented curation of DiverseRNA-1.4M improves both perturbational and non-perturbational gene representation learning, distributional biases may limit generalization to other biological contexts. Additional research is required to establish scaling laws for transcriptomics SSL. We release novel modeling components and benchmarking tools here: `https://github.com/recursionpharma/opentxfm`.

Acknowledgments and Disclosure of Funding

This work was fully funded and supported by Recursion Pharmaceuticals, which provided computational resources and compensation for the authors. All authors are current or former employees of Recursion Pharmaceuticals and may hold equity, stock, or stock options in the company. The authors declare no other competing financial interests or third-party funding.

References

Abhinav K. Adduri, Dhruv Gautam, Beatrice Bevilacqua, Alishba Imran, Rohan Shah, Mohsen Naghipourfar, Noam Teyssier, Rajesh Ilango, Sanjay Nagaraj, Mingze Dong, Chiara Ricci-Tam, Christopher Carpenter, Vishvak Subramanyam, Aidan Winters, Sravya Tirukkovular, Jeremy Sullivan, Brian S. Plosky, Basak Eraslan, Nicholas D. Youngblut, Jure Leskovec, Luke A. Gilbert, Silvana Konermann, Patrick D. Hsu, Alexander Dobin, Dave P. Burke, Hani Goodarzi, and Yusuf H. Roohani. Predicting cellular responses to perturbation across diverse contexts with state. *bioRxiv*, 2025. doi: 10.1101/2025.06.26.661135. URL https://www.biorxiv.org/content/early/2025/07/10/2025.06.26.661135.

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. URL https://doi.org/10.1038/s41592-025-02772-6.

Benedikt Alkin, Lukas Miklautz, Sepp Hochreiter, and Johannes Brandstetter. MIM-Refiner: A contrastive learning boost from intermediate pre-trained representations. *arXiv preprint arXiv:2402.10093*, 2024.

D. Michael Ando, Cory Y. McLean, and Marc Berndl. Improving phenotypic measurements in high-content imaging screens. *bioRxiv*, 2017. doi: 10.1101/161422.

Sercan Ö Arik and Tomas Pfister. Tabnet: Attentive interpretable tabular learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 6679–6687, 2021.

Ihab Bendidi, Shawn T Whitfield, Kian Kenyon-Dean, Hanene Ben Yedder, Yassir El Mesbahi, Emmanuel Noutahi, and Alisandra Kaye Denton. Benchmarking transcriptomics foundation models for perturbation analysis: one pca still rules them all. In *NeurIPS 2024 Workshop on AI for New Drug Modalities*, 2024.

Rebecca Boiarsky, Nalini M Singh, Alejandro Buendia, Ava P Amini, Gad Getz, and David Sontag. Deeper evaluation of a single-cell foundation model. *Nature Machine Intelligence*, 6(12):1443–1446, 2024.

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

Safiye Celik, Jan-Christian Hütter, Sandra Melo Carlos, Nathan H Lazar, Rahul Mohan, Conor Tillinghast, Tommaso Biancalani, Marta M Fay, Berton A Earnshaw, and Imran S Haque. Building, benchmarking, and exploring perturbative maps of transcriptional and morphological data. *PLOS Computational Biology*, 20(10):e1012463, 2024.

Yiqun Chen and James Zou. Genept: a simple but effective foundation model for genes and cells built from chatgpt. *bioRxiv*, pp. 2023–10, 2024.

Ana Conesa, Pedro Madrigal, Sonia Tarazona, David Gomez-Cabrero, Alejandra Cervera, Andrew McPherson, Michał Wojciech Szcześniak, Daniel J Gaffney, Laura L Elo, Xuegong Zhang, et al. A survey of best practices for rna-seq data analysis. *Genome biology*, 17(1):13, 2016.

GTEx Consortium. The GTEx Consortium atlas of genetic regulatory effects across human tissues. *Science*, 369(6509):1318–1330, 2020.

Gerold Csendes, Gema Sanz, Kristóf Z Szalay, and Bence Szalai. Benchmarking foundation cell models for post-perturbation rna-seq prediction. *BMC genomics*, 26(1):393, 2025.

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.

Cell Science Program CZI, Shibla Abdulla, Brian Aevermann, Pedro Assis, Seve Badajoz, Sidney M Bell, Emanuele Bezzi, Batuhan Cakir, Jim Chaffer, Signe Chambers, et al. CZ CELLxGENE Discover: a single-cell data platform for scalable exploration, analysis and modeling of aggregated data. *Nucleic acids research*, 53(D1):D886–D900, 2025.

Alan DenAdel, Madeline Hughes, Akshaya Thoutam, Anay Gupta, Andrew W Navia, Nicolo Fusi, Srivatsan Raghavan, Peter S Winter, Ava P Amini, and Lorin Crawford. Evaluating the role of pre-training dataset size and diversity on single-cell foundation model performance. *bioRxiv*, pp. 2024–12, 2024.

Antonio Fabregat, Steven Jupe, Lisa Matthews, Konstantinos Sidiropoulos, Marc Gillespie, Phani Garapati, Robin Haw, Bijay Jassal, Florian Korninger, Bruce May, et al. The reactome pathway knowledgebase. *Nucleic acids research*, 46(D1):D649–D655, 2018.

Felix Fischer, David S Fischer, Roman Mukhin, Andrey Isaev, Evan Biederstedt, Alexandra-Chloé Villani, and Fabian J Theis. sctab: Scaling cross-tissue single-cell annotation models. *Nature Communications*, 15(1):6611, 2024.

Philip Fradkin, Ruian "Ian" Shi, Taykhoom Dalal, Keren Isaev, Brendan J Frey, Leo J Lee, Quaid Morris, and Bo Wang. Orthrus: toward evolutionary and functional rna foundation models. *Nature Methods*, pp. 1–11, 2026.

Shreshth Gandhi, Farnoosh Javadi, Valentine Svensson, Umair Khan, Matthew G. Jones, John Yu, Daniele Merico, Hani Goodarzi, and Nima Alidoust. Tahoe-x1: Scaling perturbation-trained single-cell foundation models to 3 billion parameters. *bioRxiv*, 2025. doi: 10.1101/2025.10.23.683759. URL https://www.biorxiv.org/content/early/2025/10/23/2025.10.23.683759.

Gabriela Rapozo Guimarães, Giovanna Resk Maklouf, Cristiane Esteves Teixeira, Leandro de Oliveira Santos, Nayara Gusmão Tessarollo, Nayara Evelin de Toledo, Alessandra Freitas Serain, Cristóvão Antunes de Lanna, Marco Antônio Pretti, Jéssica Gonçalves Vieira da Cruz, et al. Single-cell resolution characterization of myeloid-derived cell states with implication in cancer outcome. *Nature Communications*, 15(1):5694, 2024.

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.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

Graham Heimberg, Rajat Bhatnager, Hana El-Samad, and Matt Thomson. Low-dimensionality in gene expression data enables the accurate extraction of transcriptional programs from shallow sequencing. *Cell Systems*, 2:239–250, 2016.

Nicholas Ho, Caleb N Ellington, Jinyu Hou, Sohan Addagudi, Shentong Mo, Tianhua Tao, Dian Li, Yonghao Zhuang, Hongyi Wang, Xingyi Cheng, et al. Scaling dense representations for single cell with transcriptome-scale context. *bioRxiv*, pp. 2024–11, 2024.

Robert C Jones, Jim Karkanias, Mark A Krasnow, Angela Oliveira Pisco, Stephen R Quake, Julia Salzman, Nir Yosef, Bryan Bulthaup, Phillip Brown, et al. The Tabula Sapiens: A multiple-organ, single-cell transcriptomic atlas of humans. *Science*, 376(6594):eabl4896, 2022.

Jérémie Kalfon, Jules Samaran, Gabriel Peyré, and Laura Cantini. scprint: pre-training on 50 million cells allows robust gene network predictions. *Nature Communications*, 16(1):3607, 2025.

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):101, 2025.

Kian Kenyon-Dean, Zitong Jerry Wang, John Urbanik, Konstantin Donhauser, Jason Hartford, Saber Saberian, Nil Sahin, Ihab Bendidi, Safiye Celik, Juan Sebastián Rodríguez Vera, Imran Haque, and Oren Kraus. Vitally consistent: Scaling biological representation learning for cell microscopy. In *Forty-second International Conference on Machine Learning*, volume 267 of *PMLR*, pp. 29735–29752, 2025.

Vladislav Kim, Nikolaos Adaloglou, Marc Osterland, Flavio M Morelli, Marah Halawa, Tim König, David Gnutt, and Paula A Marin Zapata. Self-supervision advances morphological profiling by unlocking powerful image representations. *Scientific Reports*, 15(1):4876, 2025.

Oren Kraus, Kian Kenyon-Dean, Saber Saberian, Maryam Fallah, Peter McLean, Jess Leung, Vasudev Sharma, Ayla Khan, Jia Balakrishnan, Safiye Celik, Dominique Beaini, Maciej Sypetkowski, Chi Vicky Cheng, Kristen Morse, Maureen Makes, Ben Mabey, and Berton Earnshaw. Masked autoencoders for microscopy are scalable learners of cellular biology. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11757–11768, 2024.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Sal Candido, et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *BioRxiv*, 2022:500902, 2022.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

Elon Litman, Tyler Myers, Vinayak Agarwal, Ekansh Mittal, Orion Li, Ashwin Gopinath, and Timothy Kassis. Genejepa: A predictive world model of the transcriptome. *bioRxiv*, 2025. doi: 10.1101/2025.10.14.682378. URL https://www.biorxiv.org/content/early/2025/10/15/2025.10.14.682378.

Romain Lopez, Jeffrey Regier, Michael B Cole, Michael I Jordan, and Nir Yosef. Deep generative modeling for single-cell transcriptomics. *Nature methods*, 15(12):1053–1058, 2018.

Michael I Love, Wolfgang Huber, and Simon Anders. Moderated estimation of fold change and dispersion for rna-seq data with deseq2. *Genome biology*, 15(12):550, 2014.

Malte D Luecken, Maren Büttner, Kridsadakorn Chaichoompu, Anna Danese, Marta Interlandi, Michaela F Müller, Daniel C Strobl, Luke Zappia, Martin Dugas, Maria Colomé-Tatché, et al. Benchmarking atlas-level data integration in single-cell genomics. *Nature methods*, 19(1):41–50, 2022.

James M McFarland, Brenton R Paolella, Allison Warren, Kathryn Geiger-Schuller, Tsukasa Shibue, Michael Rothberg, Olena Kuksenko, William N Colgan, Andrew Jones, Emily Chambers, et al. Multiplexed single-cell transcriptional response profiling to define cancer vulnerabilities and therapeutic mechanism of action. *Nature communications*, 11(1):4296, 2020.

Karol Nowicki-Osuch, Lizhe Zhuang, Tik Shing Cheung, Emily L Black, Neus Masqué-Soler, Ginny Devonshire, Aisling M Redmond, Adam Freeman, Massimilliano di Pietro, Nastazja Pilonis, et al. Single-cell rna sequencing unifies developmental programs of esophageal and gastric intestinal metaplasia. *Cancer discovery*, 13(6):1346–1363, 2023.

James D Pearce, Sara E Simmonds, Gita Mahmoudabadi, Lakshmi Krishnan, Giovanni Palla, Ana-Maria Istrate, Alexander Tarashansky, Benjamin Nelson, Omar Valenzuela, Donghui Li, et al. A cross-species generative cell atlas across 1.5 billion years of evolution: The transcriptformer single-cell model. *bioRxiv*, pp. 2025–04, 2025.

Livia Perfetto, Leonardo Briganti, Alberto Calderone, Andrea Cerquone Perpetuini, Marta Iannuccelli, Francesca Langone, Luana Licata, Milica Marinkovic, Anna Mattioni, Theodora Pavlidou, et al. Signor: a database of causal relationships between biological entities. *Nucleic acids research*, 44(D1):D548–D554, 2016.

Joseph M Replogle, Reuben A Saunders, Angela N Pogson, Jeffrey A Hussmann, Alexander Lenail, Alina Guna, Lauren Mascibroda, Eric J Wagner, Karen Adelman, Gila Lithwick-Yanai, et al. Mapping information-rich genotype-phenotype landscapes with genome-scale perturb-seq. *Cell*, 185(14):2559–2575, 2022.

Till Richter, Mojtaba Bahrami, Yufan Xia, David S Fischer, and Fabian J Theis. Delineating the effective use of self-supervised learning in single-cell genomics. *Nature Machine Intelligence*, 7 (1):68–78, 2025.

Davide Risso, Fanny Perraudeau, Svetlana Gribkova, Sandrine Dudoit, and Jean-Philippe Vert. A general and flexible method for signal extraction from single-cell rna-seq data. *Nature communications*, 9(1):284, 2018.

Syed Asad Rizvi, Daniel Levine, Aakash Patel, Shiyang Zhang, Eric Wang, Curtis Jamison Perry, Nicole Mayerli Constante, Sizhuang He, David Zhang, Cerise Tang, Zhuoyang Lyu, Rayyan Darji, Chang Li, Emily Sun, David Jeong, Lawrence Zhao, Jennifer Kwan, David Braun, Brian Hafler, Hattie Chung, Rahul M. Dhodapkar, Paul Jaeger, Bryan Perozzi, Jeffrey Ishizuka, Shekoofeh Azizi, and David van Dijk. Scaling large language models for next-generation single-cell analysis. *bioRxiv*, 2025. doi: 10.1101/2025.04.14.648850. URL https://www.biorxiv.org/content/early/2025/10/25/2025.04.14.648850.

Yanay Rosen, Yusuf Roohani, Ayush Agarwal, Leon Samotorčan, Tabula Sapiens Consortium, Stephen R Quake, and Jure Leskovec. Universal cell embeddings: A foundation model for cell biology. *bioRxiv*, pp. 2023–11, 2023.

Cristian Ruiz-Moreno, Sergio Marco Salas, Erik Samuelsson, Sebastian Brandner, Mariette E.G. Kranendonk, Mats Nilsson, and Hendrik G. Stunnenberg. Harmonized single-cell landscape, intercellular crosstalk and tumor architecture of glioblastoma. *bioRxiv*, 2022. doi: 10.1101/2022.08.27.505439. URL https://www.biorxiv.org/content/early/2022/08/27/2022.08.27.505439.

Monica Schenone, Vlado Dančík, Bridget K Wagner, and Paul A Clemons. Target identification and mechanism of action in chemical biology and drug discovery. *Nature chemical biology*, 9(4): 232–240, 2013.

Sanjay R Srivatsan, José L McFaline-Figueroa, Vijay Ramani, Lauren Saunders, Junyue Cao, Jonathan Packer, Hannah A Pliner, Dana L Jackson, Riza M Daza, Lena Christiansen, et al. Massively multiplex chemical transcriptomics at single-cell resolution. *Science*, 367(6473):45–51, 2020.

Artur Szałata, Karin Hrovatin, Sören Becker, Alejandro Tejada-Lapuerta, Haotian Cui, Bo Wang, and Fabian J Theis. Transformers in single-cell omics: a review and new perspectives. *Nature methods*, 21(8):1430–1443, 2024.

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, et al. Transfer learning enables predictions in network biology. *Nature*, 618(7965):616–624, 2023.

John N Weinstein, Eric A Collisson, Gordon B Mills, Kenna R Shaw, Brad A Ozenberger, Kyle Ellrott, Ilya Shmulevich, Chris Sander, and Joshua M Stuart. The cancer genome atlas pan-cancer analysis project. *Nature genetics*, 45(10):1113–1120, 2013.

Hongzhi Wen, Wenzhuo Tang, Xinnan Dai, Jiayuan Ding, Wei Jin, Yuying Xie, and Jiliang Tang. Cellplm: Pre-training of cell language model beyond single cells. *BioRxiv*, pp. 2023–10, 2023.

Frederik Wenkel, Wilson Tu, Cassandra Masschelein, Hamed Shirzad, Liam Hodgson, Ihab Bendidi, Cian Eastwood, Shawn T Whitfield, Craig Russell, Yassir El Mesbahi, et al. Txpert: using multiple knowledge graphs for prediction of transcriptomic perturbation effects. *Nature Biotechnology*, pp. 1–8, 2026.

Daniel R Wong, Abby S Hill, and Rob Moccia. Simple controls exceed best deep learning algorithms and reveal foundation model effectiveness for predicting genetic perturbations. *Bioinformatics*, 2025.

Sunny Z Wu, Ghamdan Al-Eryani, Daniel Lee Roden, Simon Junankar, Kate Harvey, Alma Andersson, Aatish Thennavan, Chenfei Wang, James R Torpy, Nenad Bartonicek, et al. A single-cell and spatially resolved atlas of human breast cancers. *Nature genetics*, 53(9):1334–1347, 2021.

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.

Xinyu Yuan, Zhihao Zhan, Zuobai Zhang, Manqi Zhou, Jianan Zhao, Boyu Han, Yue Li, and Jian Tang. Cell ontology guided transcriptome foundation model. *Advances in Neural Information Processing Systems*, 37:6323–6366, 2024.

Appendix contents

A TxFM training **17**

A.1 TxFM training hyperparameters 17

A.2 Epochwise and layerwise analysis of TxFM 19

A.3 Loss functions 19

A.4 Gradient analysis of loss functions 21

A.5 Gene count statistics are encoded into TxFM embedding space 23

B Evaluation methodology **23**

B.1 Perturbation representation benchmarks 23

B.2 Additional analysis: cell type clustering and classification with scIB 26

C Ablation experiments **30**

C.1 Activation functions 30

C.2 Dataset curation 30

D Dataset-specific benchmarking results **31**

E NeurIPS Paper Checklist **33**

A TxFM training

A.1 TxFM training hyperparameters

All TxFM models were trained with data-distributed parallel methods over H100 GPUs on a large-scale compute cluster. Table 7 describes relevant training hyperparameters. Each model used a one-cycle cosine learning rate decay schedule with 10% warm-up using the AdamW optimizer with max learning rate 1e-3, betas (0.9, 0.999), epsilon 1e-6, weight decay of 1 divided by the learning rate times the total number of training steps, bfloat16-mixed precision, with additional standard techniques such as LayerScale. The models with 1.4M samples were trained on our curated DiverseRNA-1.4M (Table 1). The TxFM-B trained on 57M samples used the much larger CellxGene Transcriptformer-Sapiens dataset (Pearce et al., 2025); preliminary investigations indicated that $K = 1024$ was more suitable for the sparser scRNA-seq data in this larger dataset vs the denser DiverseRNA-1.4M data. All other model parameter settings, if not otherwise stated, follow the TxFM-B Default model as per the ablations in § 4.4.

Table 7: Training hyperparameters for TxFM variations.

hyperparameter	TxFM-S	TxFM-B Default	TxFM-B CxG	TxFM-L
training dataset # samples	1.4M	1.4M	57M	1.4M
training epochs	200	200	50	200
K (# unmasked tokens)	2048	2048	1024	2048
global batch size	1536	1536	6144	1536
total training GPU-hours	400	864	4096	2500
# encoder blocks	6	12	12	24
# MHSA heads	6	12	12	16
model embedding dim.	384	768	768	1024
stochastic depth rate	0.1	0.1	0.1	0.3
total learnable parameters	57M	159M	185M	403M

A.1.1 Transductive SSL-finetuning

In Table 3 we evaluate the impact of training unsupervised models directly on the evaluation data to address biological use cases pertaining to individual dataset analysis, where PCA is typically used as the representation. (This is in contrast to the zero-shot setting in Table 2, where the evaluation data is entirely unseen during unsupervised training and where we sought to determine how strong pretrained SSL models generalize to new cellular contexts.) Therefore, we evaluated performance when training PCA and scVI models directly on each of the evaluation datasets (HEPG2, Jurkat, RPE1) and compared to self-supervised finetuning the default TxFM-B on those datasets. Our SSL finetuning performs masked reconstruction on the evaluation dataset with the same setup as pretraining (2048 unmasked genes, 200 epochs). In Figure 3 we visualize the difference in our Poisson reconstruction loss between training a TxFM-B from scratch on RPE1 versus finetuning. We observe that the pretrained model is already very close to high performance on RPE1, and the benefit of finetuning is made clear as it obtains better generalization that the from-scratch TxFM, as the validation set contains held out experimental batches from the assay.

A.1.2 Scaling on DiverseRNA-1.4M

Figure 4 shows training loss curves for the three different TxFM backbones trained on DiverseRNA-1.4M (Table 6 (g), training hyperparameters in Table 7). As we can see, TxFM-L obtains the best reconstruction loss on the validation set during training. However, this unfortunately did not translate to better downstream transfer learning performance. It is possible that the parameter count for TxFM-B saturates DiverseRNA-1.4M, and therefore more curated data should be obtained to effectively scale to TxFM-L. While we would have liked to explore doing so in this work, we sought to instead prioritize our compute budget to comprehensive architecture ablations on the TxFM-B backbone, which already required considerable training compute per standard run. Future work should explore scaling laws and data curation further.

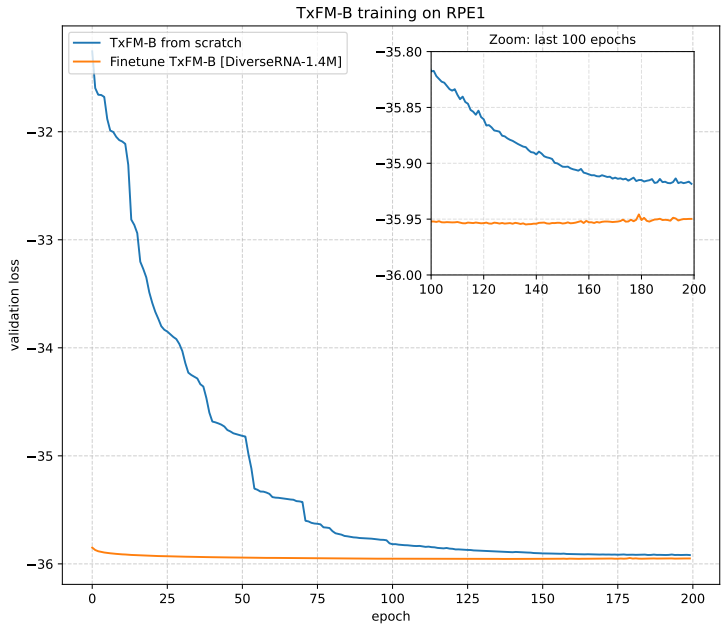


Figure 3: Example of SSL training a TxFM-B on the RPE1 dataset from scratch versus SSL fine-tuning TxFM-B Default (pretrained on DiverseRNA-1.4M) on the RPE1 dataset.

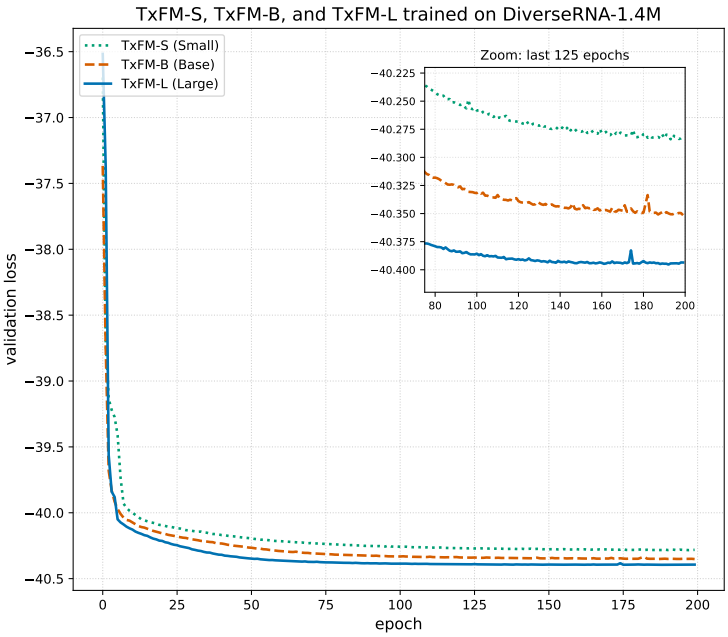


Figure 4: Validation set (held out experimental batches from each of the composite datasets) reconstruction loss curves for the three different TxFM backbones evaluated in Table 6 (g).



Figure 5: **Epochwise analysis of TxFM during training.** Gene-gene relationship recall (Kraus et al., 2024) performance on 5 databases (plus all combined) of gene representations extracted from the Default TxFM-B codebook and decoder, as a function of training epoch. Dashed lines indicate the recall for the equivalent layers of earlier versions of the scVI and PCA encoder baselines trained on DiverseRNA-1.4M.

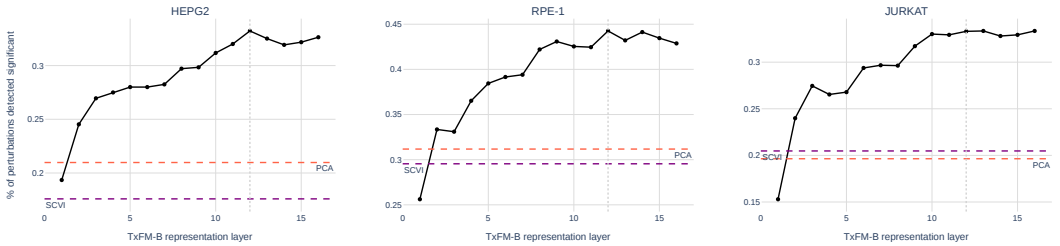


Figure 6: **Layerwise inference-time analysis of TxFM.** Perturbation consistency on HEPG2, RPE-1, and Jurkat datasets for each layer of our Default TxFM-B (encoder layers followed by decoder layers) compared to scVI and PCA (each trained on DiverseRNA-1.4M).

A.2 Epochwise and layerwise analysis of TxFM

TxFM learns gene representations in both its encoder token embeddings and the decoder reconstruction matrix; cosine similarities between these gene representations recover known biological relationships. Figure 5 tracks relationship recall during training of TxFM-B, compared to scVI and PCA trained on the same data. Decoder-based relationships emerge early, while codebook relationships improve more gradually. On Signor and Reactome (Perfetto et al., 2016; Fabregat et al., 2018), decoder recall peaks in the first quarter of training and then declines. Interestingly, Signor is the only database where the final epoch's codebook exceeds the decoder, a trend also seen for scVI. TxFM's learned gene weights outperform those of scVI, indicating potential for novel biological relationship discovery for the purpose of finding new drug targets (Schenone et al., 2013).

Figure 6 evaluates layerwise inference by aggregating perturbation (CLS) representations for three unseen cell lines, extracted from each encoder layer and decoder MLP layer. Representation quality improves after the first encoder layer and continues to increase through the 12th (default) layer, with little change in the decoder MLP. TxFM-B's perturbation representations outperform those of scVI and PCA. The best layer is the final transformer block, unlike some large vision MAEs which find optimal performance at intermediate blocks (Alkin et al., 2024; Kenyon-Dean et al., 2025).

A.3 Loss functions

Preprocessing. We preprocess raw gene expression counts by first performing library size normalization. Library size is the total count summed across all genes in a sample. As library size varies between e.g. single cells, it is a common step to normalize all cells to have equal library size L so that individual gene counts are comparable to each other. In this work, we set $L = 10^5$. We then log-transform normalized gene counts, with a prior addition of a pseudocount. This is another common step in RNA-seq analysis, which enables log-transform in the presence of zero gene counts

(which are very often found in scRNA-seq data).

$$\tilde{x}_{ng} = \frac{x_{ng}}{\sum_{i=1}^{G} x_{ni}} \cdot L \tag{3}$$

$$\tilde{x}_{ng} = \log(\tilde{x}_{ng} + 1) \tag{4}$$

x_{ng} is the raw gene expression count for gene g in sample n, G is the total number of genes profiled in that sample, and $\log$ is the natural logarithm.

Poisson-based loss. For an expression count x_{ng}, the Poisson negative log-likelihood (NLL) is defined as follows, while we drop the constant below:

$$\begin{aligned} -\log p(x_{ng}|\lambda) &= -\log\left(\frac{\lambda^{x_{ng}} e^{-\lambda}}{x_{ng}!}\right) \\ &= -x_{ng}\log\lambda + \lambda \end{aligned} \tag{5}$$

Setting the learned rate parameter to our model's prediction as $\hat{\mathbf{x}}_g = \log\lambda$ for the target log-count $\mathbf{x}_g$ (preprocessed as explained above), we recover the loss:

$$\mathcal{L}_{\text{Poisson}}(\mathbf{x}_g, \hat{\mathbf{x}}_g) = e^{\hat{\mathbf{x}}_g} - \hat{\mathbf{x}}_g \cdot e^{\mathbf{x}_g} \tag{6}$$

We model genes in a sample as i.i.d. and thus sum loss values over genes to obtain $\mathcal{L}_{\text{Poisson}}(\mathbf{x}, \hat{\mathbf{x}})$ for an input gene expression vector $\mathbf{x}$. We used library size-normalized counts with pseudocounts as targets.

Gradient of the loss. Putting together our rectified tanh activation function ϕ and the Poisson-based loss, we can write our loss for a gene count $\mathbf{x}_i$ and its model-generated logit z as:

$$\mathcal{L} = e^{\phi(z)} - \phi(z)e^{\mathbf{x}_i} \tag{7}$$

Taking the gradient, we obtain:

$$\frac{\delta\mathcal{L}}{\delta z} = e^{\phi(z)}\phi'(z) - e^{\mathbf{x}_i}\phi'(z) \tag{8}$$

$$= \phi'(z)\left(e^{\phi(z)} - e^{\mathbf{x}_i}\right) \tag{9}$$

The derivative of the rectified tanh (§ C.1) is given by:

$$\frac{\delta\phi}{\delta z} = \frac{\delta}{\delta z}\left(\log(L+1)\text{ReLU}\left(\tanh\left(\frac{z}{4e}\right)\right)\right) \tag{10}$$

$$= \begin{cases} \frac{\log(L+1)}{4e}\operatorname{sech}^2\left(\frac{z}{4e}\right) & z > 0 \\ 0 & z \leq 0 \end{cases} \tag{11}$$

Putting this together and setting $\alpha = \log(L+1), \beta = 4e$ for brevity, we obtain the gradient of the loss, where the hyperbolic secant function acts as a saturating gate on the Poisson error term:

$$\frac{\delta\mathcal{L}}{\delta z} = \begin{cases} \frac{\alpha}{\beta}\operatorname{sech}^2\left(\frac{z}{\beta}\right)\left(e^{\alpha\tanh\left(\frac{z}{\beta}\right)} - e^{\mathbf{x}_i}\right) & z > 0 \\ 0 & z \leq 0 \end{cases} \tag{12}$$

Negative binomial-based loss. For the ablation experiments, we implemented a negative binomial loss function using the Pytorch `distributions` package. We followed the parameterization by Risso et al. (2018) that models the mean and inverse dispersion as parameters of the distribution and allowed gene-specific inverse dispersion parameters θ_g shared across samples. The distribution is then defined as follows for a gene count x_{ng}:

$$p(x_{ng} \mid \mu_{ng}, \theta_g) = \binom{x_{ng} + \theta_g - 1}{x_{ng}}\left(\frac{\theta_g}{\theta_g + \mu_{ng}}\right)^{\theta_g}\left(\frac{\mu_{ng}}{\theta_g + \mu_{ng}}\right)^{x_{ng}} \tag{13}$$

We scaled the learned mean μ_{ng} for each gene count by the cell size factor defined as $s_n = \frac{\sum_g x_{ng}}{L}$. This implies the following reparameterization: $r = \theta_g$ for the number of failures r and $p = \frac{s_n \mu_{ng}}{s_n \mu_{ng} + \theta_g}$ for the probability of success p. The loss for a given gene count is given by the negative log-likelihood of $\text{NB}(r, p)$ and loss values are again summed over genes for a given sample. We used raw integer counts as targets as library size is accounted for by the size factor s_n.

Zero-inflated negative binomial-based loss. For the ablation experiments, we implemented a zero-inflated negative binomial loss function building on the NB loss described above. We modeled gene- and cell-specific zero inflation (or dropout) probabilities following Lopez et al. (2018). We implemented them as a separate linear layer from the CLS token to G zero-inflation logits for each cell. The ZINB distribution is defined as:

$$p(x_{ng} \mid \mu_{ng}, \theta_g, \pi_{ng}) = \begin{cases} \pi_{ng} + (1 - \pi_{ng}) \cdot \text{NB}(0 \mid \mu_{ng}, \theta_g) & \text{if } x_{ng} = 0 \\ (1 - \pi_{ng}) \cdot \text{NB}(x_{ng} \mid \mu_{ng}, \theta_g) & \text{if } x_{ng} > 0 \end{cases} \tag{14}$$

We used raw integer counts as targets as library size is accounted for by the size factor s_n.

Cross-entropy loss. For the ablation experiments, we implemented a rank-based cross-entropy loss. We discretized the log1p-normalized targets into 51 equal-width bins (one dedicated bin for zero expression plus 50 other bins, following the scGPT approach) when computing the loss. We extended our default decoder architecture (a 4-layer MLP generating gene-specific predictions) with a 2-layer nonlinear shared projection from each scalar to 51 bin logits. This modification added negligible number of model parameters ($\sim$3k). We kept the encoder, input representation, masking strategy, and all hyperparameters the same as the default model.

We considered avoiding the scalar bottleneck and experimented with a decoder architecture outputting 51-dimensional logit vector per gene, with a narrower final residual block to control for parameter explosion. However, we found this implementation of the cross-entropy loss to perform worse than the results we report in the ablation study.

A.4 Gradient analysis of loss functions

NB vs. Poisson. Figure 7 shows the Poisson and NB NLLs for a single example observation. $\text{NB}(r, p)$ is parameterized with the number of failures r and probability of success p, where $r = \theta_g$ for inverse dispersion θ_g. Thus, lower r values correspond to genes with higher dispersion. The loss curves for high-dispersion genes have increasingly small gradients when the mean predictions overestimate the target. In contrast, the loss curves for low-dispersion genes (high r) closely approximate the Poisson curve, as expected. For those curves, the gradients are larger for overestimated predictions, providing signal for the optimizer.

We can also see this analytically by considering the NLLs (with constants w.r.t. to the predicted mean dropped):

$$\mathcal{L}_{\text{Poisson}} = \lambda - x \log \lambda \tag{15}$$

$$\mathcal{L}_{\text{NB}} = -r \log \frac{r}{\mu + r} - x \log \frac{\mu}{\mu + r} \tag{16}$$

The gradient of each loss w.r.t. to the mean (λ for Poisson and μ for NB) are:

$$\frac{\partial \mathcal{L}}{\partial \lambda} = 1 - \frac{x}{\lambda} \tag{17}$$

$$\frac{\partial \mathcal{L}_{\text{NB}}}{\partial \mu} = \frac{r}{\mu + r} - \frac{xr}{\mu(\mu + r)} = \frac{r(\mu - x)}{\mu(\mu + r)} \tag{18}$$

We observe that the NB gradient is the Poisson gradient scaled by $\frac{r}{r+\mu}$. Let's consider two cases: when the model underestimates ($\mu \ll x$) and overestimates the target ($\mu \gg x$).

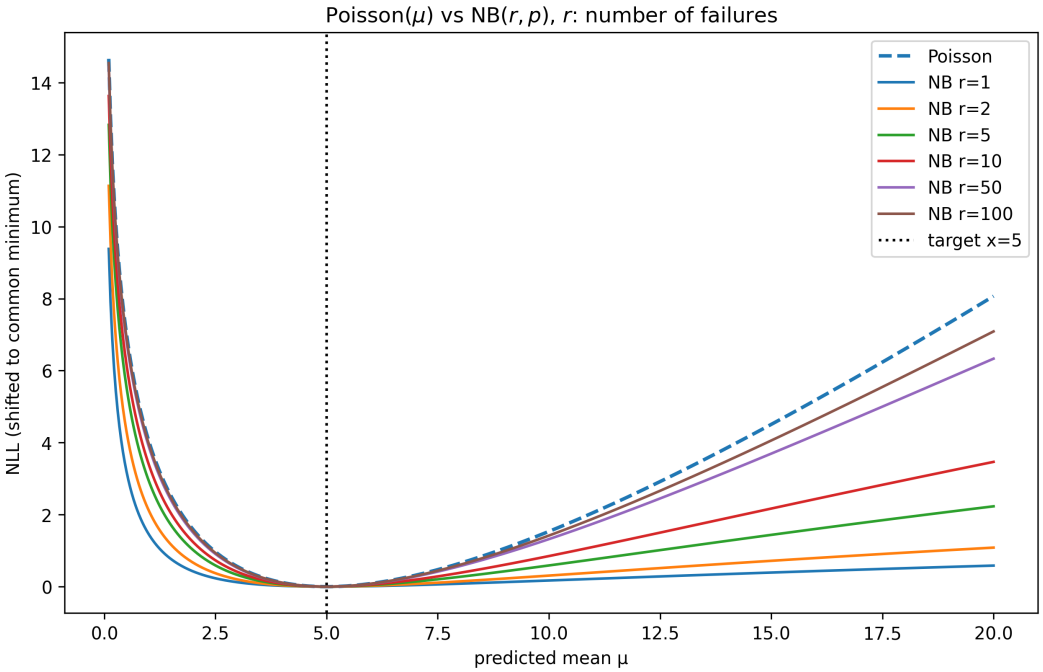


Figure 7: **Poisson vs NB NLL for a single observation and a range of r values for NB.** The mean of each distribution is plotted on the x-axis and the target is shown with a dashed line. Each curve is shifted on the y-axis so all minima line up at 0 as the gradients are invariant to this shift and for easier comparison of the slopes.

- Overestimation under Poisson: $\lim_{\lambda\to\infty}\left(1-\frac{x}{\lambda}\right)=1$ (gradient is constant, never vanishes)
- Overestimation under NB: $\lim_{\mu\to\infty}\frac{r(\mu-x)}{\mu(\mu+r)}=\lim_{\mu\to\infty}\frac{r/\mu-rx/\mu^2}{1+r/\mu}=\frac{0-0}{1+0}=0$ (gradient vanishes)
- Underestimation under Poisson: $\lim_{\lambda\to 0^+}\left(1-\frac{x}{\lambda}\right)=-\infty$ (strong negative gradient)
- Underestimation under NB: $\lim_{\mu\to 0^+}\frac{r(\mu-x)}{\mu(\mu+r)}=-\infty$ (strong negative gradient)

We can also see that for large μ in the overestimation case, the gradient of the NB loss $\frac{r(\mu-x)}{\mu(\mu+r)}\approx\frac{r}{\mu}$, meaning that for small r (high dispersion genes), the rate of gradient vanishing is more severe. As a consequence, predictions that overestimate the mean for high-dispersion genes persist under the NB loss. In contrast, the Poisson loss corrects overestimation with a gradient that approaches 1 regardless of dispersion. Both losses can correct underestimation of the mean with strong gradient signal.

ZINB vs NB. For non-zero targets $x>0$, the ZINB likelihood equals to the NB likelihood scaled by $1-\pi$ factor. Thus, the same gradient analysis applies as described for the NB loss above, as the zero inflation parameter π does not affect the gradient w.r.t. μ.

For $x=0$, the ZINB NLL is:

$$\mathcal{L}_{\text{ZINB}}=-\log\left(\pi+(1-\pi)\cdot\text{NB}(0;\mu,r)\right)=-\log\left(\pi+(1-\pi)\left(\frac{r}{r+\mu}\right)^r\right) \tag{19}$$

When μ is small, it is close to the zero target so the prediction is good and there is no error to correct.

As μ increases (the overestimation case) and if π is large, the NB component tends to 0 and the NLL saturates at at $-\log(\pi)$. The gradient vanishes because the zero-inflation component π explains

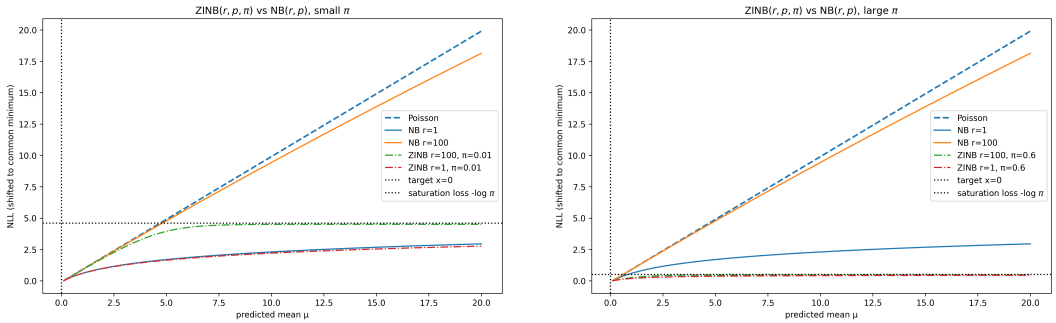


Figure 8: **ZINB vs NB NLL for a single observation and a range of r values for NB and π values for ZINB.** Left: small π, right: large π. The mean of each distribution is plotted on the x-axis and the target is shown with a dashed line. Each curve is shifted on the y-axis so all minima line up at 0 as the gradients are invariant to this shift and for easier comparison of the slopes.

the observed zero regardless of predicted mean or dispersion. However, when π is small, the NB component dominates, though still subject to the same ceiling of $-\log(\pi)$.

Figure 8 shows that for large π, the loss curves saturate to $-\log(\pi)$ very quickly for both low- and high-dispersion genes, leading to vanishing gradients. However, for small π, the high-dispersion ($r = 1$) ZINB loss saturates less quickly than the low-dispersion one, and closely approximates the NB loss with the corresponding r. On the other hand, the low-dispersion ($r = 100$) ZINB loss reaches the saturating ceiling quicker and diverges from the corresponding NB loss, which in turn approximates the Poisson loss.

In summary, the ZINB loss inherits the NB vanishing gradient behaviours and adds an additional saturation mechanism for zero-count genes via the $-\log(\pi)$ ceiling on the loss.

A.5 Gene count statistics are encoded into TxFM embedding space

We regressed gene count mean (mu) and variance (var) independently and jointly against projections of TxFM's gene codebook embeddings onto their principal components to obtain estimates of the explained variance captured by these statistics. We repeated this procedure 100 times using shuffled embeddings to obtain a reference null (shown with grey distributions). The actual variance explained is significantly higher than the null for each statistic independently as well as jointly (Figure 9). Moreover, TxFM encodes information from each statistic, as evidenced by the increased variance explained when fitted jointly compared to each independently.

B Evaluation methodology

B.1 Perturbation representation benchmarks

B.1.1 Evaluation setup

We adopt the hierarchical evaluation framework introduced in Bendidi et al. (2024), which assesses model performance on gene perturbation tasks using a biologically motivated metric suite. Each metric evaluates a different aspect of model utility, with aggregation occurring across runs, metrics, and tasks: results of evaluations performed at 5 different seeds are first averaged together, then groups of metrics constituting a downstream task are averaged together for one final metric per downstream task. We separately apply different post-processing techniques to the embeddings and count baselines before evaluation: centering on negative controls, standardization, and TVN (Ando et al., 2017). For each downstream task and evaluated model pair, we pick the post-processing technique resulting in the best performing average score.

Batch Effect Correction (ilisi): To quantify batch mixing in the latent space, we use the Integration Local Inverse Simpson's Index (iLISI). A higher iLISI indicates better mixing of batches and

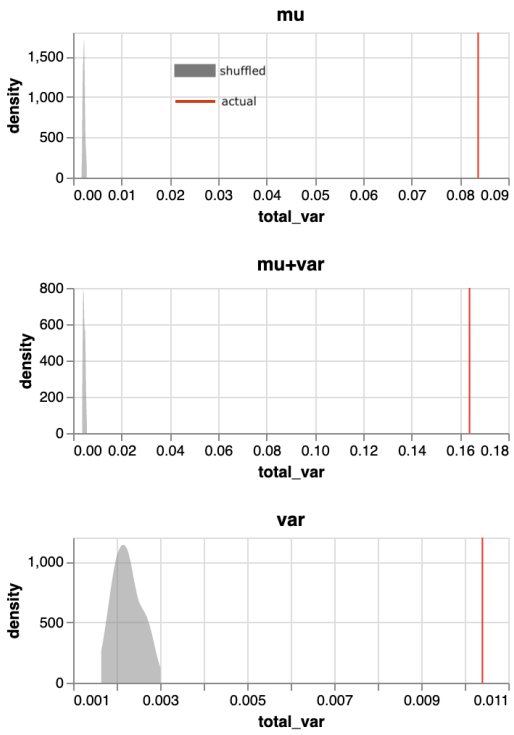


Figure 9: **Variance explained captured by gene mean and gene variance against principal components of gene representations from TxFM codebook parameters.** Top: variance explained captured by gene mean. Bottom: variance explained captured by gene variance. Middle: variance explained captured by gene mean and variance jointly. Grey densities represent estimates obtained using shuffled gene statistics, red line shows explained variance using actual gene statistics.

hence more effective batch effect reduction. The final iLISI score for each model is averaged across all batches.

Linear Separability (lin): This metric assesses how linearly distinguishable different perturbations are in the latent space using a linear classifier trained on known perturbation labels. Top-1 and Top-5 accuracy using unseen batches for evaluation are reported after being averaged together.

Latent Space Organization (knn): Local structure is assessed via k-nearest neighbor retrieval of perturbation-matched samples across disjoint biological batches. Accuracy is reported for an average of Top-1 and Top-5 retrieval.

Perturbation Consistency (p.cst): This metric measures whether embeddings of the same perturbation are more similar to each other than expected by chance. We compute perturbation consistency by comparing a perturbation's similarity to a null distribution. Here, we refer to all cells that received a given gene knockout (regardless of the guide used) when we say "perturbation". To assemble a null distribution of cosine similarities, we sample N random cells and compute cosine similarity between their embeddings and the mean embedding of each gene perturbation (computed across corresponding cells). We remove pairwise similarities that correspond to the random cell's perturbation to exclude self-self similarities. In our experiments, we set $N = 10,000$. Then, for each perturbation, we compute mean leave-one-out cosine similarity across the cell embeddings. Finally, for each perturbation, we compute empirical p-value by comparing its similarity to the null distribution. Perturbations with $p < 0.01$ are considered significantly consistent. The final score is the fraction of consistent perturbations.

Perturbation consistency is computed post-alignment consisting of a PCA transform fit on the non-targeting control samples and followed by standardizing the transformed embeddings within each batch.

Biological Relationship Recall (bmdb): To test zero-shot biological reasoning, cosine similarities between perturbation embeddings are used to predict gene-gene relationships. Predicted links (top 5% most similar/dissimilar pairs) are compared to curated databases (e.g., CORUM, HuMAP). Recall is computed per database and averaged across databases.

Latent Space Interpretability (inv): This is composed of an average of two metrics: Spearman correlation and Structural Integrity. Spearman correlation measures how well latent embeddings can be linearly decoded back into gene expression values. This tests biological interpretability of learned representations. Structural Integrity assesses how well perturbation-induced gene expression changes are preserved in reconstructed profiles. For each batch, we compute the Frobenius norm of the difference between the negative control-centered predicted and the negative control-centered actual gene expression matrices:

$$\text{Structural Distance} = \frac{1}{B}\sum_{b=1}^{B}\frac{1}{n_b}\left\|\tilde{Y}^{(b)}_{\text{pred}} - \tilde{Y}^{(b)}_{\text{actual}}\right\|_F$$

where B is the number of batches, n_b the number of samples in batch b, and $\tilde{Y}$ denotes gene expression centered by the batch-specific control. The Structural Integrity is then defined as:

$$\text{Structural Integrity} = 1 - \frac{\text{Structural Distance}}{\text{Structural Distance}_{\text{max}}}$$

where $\text{Structural Distance}_{\text{max}}$ is an upper bound estimated from the actual data. Higher scores indicate better preservation of perturbation-induced gene expression structure.

B.1.2 Benchmarked approaches

We evaluate TxFM across a range of baselines including: commonly used existing methods (PCA and scVI (Lopez et al., 2018)); simple transformations applied directly to the test data (raw counts, library size + log normalization, and its variants with 5000 or 1024 highly variable genes); a baseline of 1024 highly variable genes with randomly shuffled labels (Random label shuffle); models using ChatGPT-derived embeddings (GenePT Large, Chen & Zou (2024)); and large pretrained models trained on CellxGene or similar datasets (Geneformer (Theodoris et al., 2023), CellPLM (Wen et al., 2023), scTab (Fischer et al., 2024), UCE (Rosen et al., 2023), scCello (Yuan et al., 2024), scPrint-L (Kalfon et al., 2025), TranscriptFormer-Sapiens (Pearce et al., 2025), scGPT (Cui et al., 2024), AIDO.Cell-100M (Ho et al., 2024)), GeneJEPA (Litman et al., 2025), Tahoe-x1 (Gandhi et al., 2025), Cell2Sentence (Rizvi et al., 2025), STATE (Adduri et al., 2025).

We report results in two settings: (i) zero-shot, where models are trained on data disjoint from the test set and (ii) fit-on-evaluation-data, where models are trained or fine-tuned on the test set. Raw counts baseline uses unprocessed gene counts, Lib+Log applies library size normalization followed by a log-transform, HVG variants subset the genes to either 5000 or 1024 highly variable ones, and Random label shuffle keeps gene inputs fixed while randomizing the perturbation labels. PCA and scVI are evaluated in both settings using the standard benchmark splits.

Our TxFM models include TxFM-B DiverseRNA-1.4M, trained on a curated 1.4 million sample dataset, and TxFM-B TF-Sapiens, trained on a larger ~50 million sample dataset. These models allow to isolate the effects of data curation versus training architecture and enable a direct comparison with FMs trained on the same data such as TranscriptFormer-Sapiens.

For the zero-shot setting, we trained multiple 232M-parameter scVI models on DiverseRNA-1.4M with the following configuration: latent dimensionality of 768, a single hidden layer with 1024 units, AdamW optimizer, lr=0.0005, three variants of likelihoods (negative binomial, zero-inflated negative binomial, Poisson), two settings for weight decay (default value of 1e-06 or 0.03), and other settings set to default values. For the Poisson model, we adjusted the learning rate to 0.0001 and used default weight decay to help with training stability. We compared these scVI models for inductive performance on the held-out data using gene-gene relationship recall and perturbation consistency benchmarks (described in § B.1.1 and under *Perturbation Consistency* above). The model trained with the negative binomial likelihood and default weight decay performed best on

both benchmarks for all three datasets; we used this model to report scVI inductive performance and evaluate its codebook and decoder representations. For the transductive SSL setting, we trained scVI on each dataset following the procedure used in Bendidi et al. (2024): using 8000 highly variable genes, latent dimensionality of 256, a single hidden layer with 512 units, no dropout, a zero-inflated negative binomial likelihood and gene-level dispersion parameters.

For Cell2Sentence (Rizvi et al., 2025), we use the 2 billion parameter pretrained version and tried several different prompting strategies. We found that their default recommended prompting strategy performed better than prompting it to predict the perturbation, arriving at the following template:

"The following is a list of N gene names ordered by descending expression level in a Homo sapiens cell. Your task is to give the cell type which this cell belongs to based on its gene expression. Cell sentence: $g_1, g_2, \ldots, g_N$. The cell type corresponding to these genes is:",

such that N is the number of genes (sorted by expression value in a given sample) we include in the prompt. The sample-level embedding taken is the last token of the model's hidden state of the sequence. We found that $N = 1000$ seemed to perform best in perturbation consistency on the HEPG2 data (i.e., $N = 200$ yielded 23.3% p. cst, $N = 500$ yielded 24.7%, $N = 1000$ yielded 24.9%, but $N = 2000$ yielded 23.5%).

B.2 Additional analysis: cell type clustering and classification with scIB

We evaluate embeddings on clustering and batch integration tasks on the five single-cell datasets[3] used in Kedzierska et al. (2025). Additionally, we assess representation quality through classification probing using both linear classifiers and 2-layer MLPs trained on the learned embeddings. For clustering and batch integration, we employ metrics introduced by Luecken et al. (2022) and implemented in the `scib` package[4].

B.2.1 Clustering task

The clustering task assesses whether cell embeddings preserve meaningful biological structure by measuring how well cells cluster according to their true cell types. We employ Louvain clustering with resolution optimization: we test 10 different resolution parameters and select the resolution that maximizes NMI between discovered clusters and ground truth labels. Results are obtained by subsampling 10,000 cells with 10 random seeds and reporting the average. Three complementary metrics evaluate different aspects of biological preservation:

Normalized Mutual Information (NMI): Measures information overlap between discovered clusters $\mathcal{C}$ and true cell type labels $\mathcal{L}$:

$$\mathrm{NMI}(\mathcal{C}, \mathcal{L}) = \frac{2 \cdot I(\mathcal{C}, \mathcal{L})}{H(\mathcal{C}) + H(\mathcal{L})} \tag{20}$$

where $I(\mathcal{C}, \mathcal{L}) = \sum_{c,\ell} p(c, \ell) \log \frac{p(c,\ell)}{p(c)p(\ell)}$ is mutual information and $H(\cdot)$ denotes entropy. High NMI indicates that knowing cluster assignments provides substantial information about true cell types.

Adjusted Rand Index (ARI): Measures pairwise agreement between clusterings, corrected for chance:

$$\mathrm{ARI} = \frac{\mathrm{RI} - \mathbb{E}[\mathrm{RI}]}{\max(\mathrm{RI}) - \mathbb{E}[\mathrm{RI}]} \tag{21}$$

where $\mathrm{RI} = \frac{n_S + n_D}{\binom{n}{2}}$ counts pairs that are consistently assigned (same cluster in both clusterings: n_S; different clusters in both: n_D). ARI is more conservative than NMI, requiring precise boundary correspondence for high scores.

Average Silhouette Width (ASW): Measures separation of true cell types in embedding space:

$$\mathrm{ASW} = \frac{1}{n} \sum_{i=1}^{n} \frac{b_i^{(l)} - a_i^{(l)}}{\max(a_i^{(l)}, b_i^{(l)})} \tag{22}$$

[3] We note that the *Pancreas* dataset contains non-integer values, implying prior normalization. As the original raw counts are not available and we expect negligible impact on performance, these values were used as input for all models.

[4] `https://scib.readthedocs.io/`

Table 8: Per-dataset results for the Kedzierska et al. (2025) benchmarks in Table 9.

	Clustering			Batch Int.	Classification			
					Linear Probe		2-layer MLP	
	ASW ↑	NMI ↑	ARI ↑	$\text{ASW}_{l/b}$ ↑	Acc ↑	F1 ↑	Acc ↑	F1 ↑
Pan-immune								
(Lib+Log)Norm+2000HVG	50.7 ± 0.1	58.0 ± 0.6	33.2 ± 1.0	92.0 ± 0.2	86.0	78.6	87.2	80.3
PCA [DiverseRNA-1.4M]	50.4 ± 0.1	53.4 ± 0.4	30.5 ± 1.5	93.5 ± 0.2	82.9	75.1	84.3	79.2
scGPT	52.7 ± 0.1	62.8 ± 0.5	40.6 ± 0.8	88.3 ± 0.2	85.1	81.7	85.9	83.0
AIDO.Cell	49.5 ± 0.1	56.8 ± 0.5	32.0 ± 0.9	82.6 ± 0.4	90.3	86.4	90.7	87.6
scTab	58.5 ± 0.2	83.6 ± 0.7	76.8 ± 1.7	88.4 ± 0.4	89.9	80.2	90.4	81.1
scVI [DiverseRNA-1.4M]	50.5 ± 0.2	53.5 ± 0.4	29.7 ± 1.7	84.7 ± 0.4	81.9	78.0	83.6	81.1
STATE	50.8 ± 0.1	60.4 ± 0.5	34.2 ± 1.3	96.4 ± 0.2	88.1	81.4	88.6	83.3
Transcriptformer	52.6 ± 0.1	72.0 ± 0.5	55.1 ± 1.6	84.9 ± 0.4	90.0	86.3	90.3	86.8
TxFM-B [TF-Sapiens data]	55.7 ± 0.2	66.0 ± 0.5	36.5 ± 0.9	81.4 ± 0.3	90.2	86.7	90.7	87.6
TxFM-S [DiverseRNA-1.4M]	52.9 ± 0.2	61.7 ± 0.7	36.1 ± 1.9	84.5 ± 0.3	87.4	83.0	88.6	85.1
TxFM-B [DiverseRNA-1.4M]	53.1 ± 0.1	62.0 ± 0.6	35.8 ± 1.6	86.3 ± 0.4	88.3	83.9	89.4	85.9
Pancreas								
(Lib+Log)Norm+2000HVG	55.4 ± 0.0	75.0 ± 0.4	56.0 ± 0.4	93.0 ± 0.1	71.9	60.8	75.4	70.4
PCA [DiverseRNA-1.4M]	49.8 ± 0.0	39.9 ± 0.6	17.4 ± 2.0	83.7 ± 0.6	95.0	85.7	94.8	83.8
scGPT	50.9 ± 0.2	55.3 ± 0.6	36.3 ± 1.9	79.9 ± 1.4	90.3	75.8	87.8	77.8
AIDO.Cell	45.3 ± 0.3	48.4 ± 0.8	22.3 ± 1.8	71.7 ± 0.5	95.6	83.2	93.8	77.7
scTab	51.5 ± 0.1	54.4 ± 0.8	50.2 ± 4.8	90.6 ± 0.2	75.2	50.0	78.5	60.0
scVI [DiverseRNA-1.4M]	45.6 ± 0.2	41.6 ± 0.2	17.2 ± 2.0	70.3 ± 1.5	73.2	61.0	76.2	67.6
STATE	51.6 ± 0.0	60.7 ± 0.5	26.8 ± 0.8	93.4 ± 0.2	93.4	82.4	94.3	82.6
Transcriptformer	48.1 ± 0.2	56.0 ± 0.5	34.3 ± 1.5	73.5 ± 0.8	75.7	57.2	73.7	52.4
TxFM-B [TF-Sapiens data]	51.8 ± 0.1	50.9 ± 0.4	25.4 ± 2.5	75.2 ± 1.1	89.4	82.2	89.9	83.2
TxFM-S [DiverseRNA-1.4M]	51.1 ± 0.2	50.6 ± 0.3	22.8 ± 1.2	80.0 ± 0.9	90.9	71.4	91.3	72.5
TxFM-B [DiverseRNA-1.4M]	51.1 ± 0.2	50.6 ± 0.3	22.8 ± 1.2	80.0 ± 0.9	90.3	73.0	90.7	72.6
PBMC 12k								
(Lib+Log)Norm+2000HVG	51.4 ± 0.0	69.1 ± 0.4	59.9 ± 0.7	99.3 ± 0.0	94.8	90.9	95.5	92.2
PCA [DiverseRNA-1.4M]	51.6 ± 0.0	68.4 ± 0.7	61.4 ± 3.6	99.0 ± 0.0	93.7	87.4	94.4	88.4
scGPT	62.9 ± 0.1	83.8 ± 0.8	85.8 ± 3.7	96.8 ± 0.2	97.2	95.9	97.1	95.8
AIDO.Cell	58.7 ± 0.1	76.7 ± 0.9	65.1 ± 1.6	93.1 ± 0.2	97.5	96.2	97.5	96.6
scTab	57.8 ± 0.0	79.4 ± 0.4	82.5 ± 0.6	98.3 ± 0.1	95.0	90.2	95.1	91.7
scVI [DiverseRNA-1.4M]	59.0 ± 0.1	74.4 ± 0.5	66.8 ± 2.0	96.0 ± 0.1	95.5	91.0	96.1	92.3
STATE	51.2 ± 0.1	74.4 ± 0.7	65.0 ± 1.3	99.2 ± 0.1	95.7	92.2	96.0	92.3
Transcriptformer	61.0 ± 0.1	79.8 ± 2.1	74.5 ± 9.6	94.3 ± 0.2	97.2	96.1	97.4	96.5
TxFM-B [TF-Sapiens data]	59.3 ± 0.0	86.6 ± 0.9	89.1 ± 2.5	96.5 ± 0.1	97.5	96.2	97.5	96.2
TxFM-S [DiverseRNA-1.4M]	66.0 ± 0.0	87.7 ± 0.7	92.3 ± 0.6	96.0 ± 0.2	97.1	95.1	97.2	95.0
TxFM-B [DiverseRNA-1.4M]	66.0 ± 0.0	87.7 ± 0.7	92.3 ± 0.6	96.0 ± 0.2	97.1	94.2	97.1	95.1
PBMC 68k								
(Lib+Log)Norm+2000HVG	48.5 ± 0.1	46.1 ± 0.6	27.5 ± 1.5	96.9 ± 0.2	78.2	68.9	82.2	73.0
PCA [DiverseRNA-1.4M]	49.8 ± 0.1	43.8 ± 2.3	21.4 ± 0.4	96.5 ± 0.5	65.7	55.0	67.0	54.6
scGPT	51.0 ± 0.2	50.5 ± 0.6	32.8 ± 0.9	91.6 ± 0.7	70.3	58.7	70.3	57.1
AIDO.Cell	47.7 ± 0.2	44.0 ± 1.2	23.5 ± 2.9	88.3 ± 0.7	72.6	61.6	73.1	61.9
scTab	50.2 ± 0.1	49.0 ± 0.4	19.9 ± 0.5	94.2 ± 0.6	68.5	56.7	69.0	56.8
scVI [DiverseRNA-1.4M]	49.9 ± 0.2	41.9 ± 1.1	22.2 ± 0.4	91.2 ± 0.9	68.3	56.2	68.4	57.1
STATE	50.1 ± 0.0	45.8 ± 1.1	24.4 ± 1.7	98.6 ± 0.2	68.0	58.3	69.0	59.3
Transcriptformer	49.9 ± 0.2	49.0 ± 0.4	21.6 ± 1.6	90.1 ± 1.0	73.5	62.8	72.8	61.6
TxFM-B [TF-Sapiens data]	50.1 ± 0.2	49.8 ± 0.5	28.4 ± 1.3	89.7 ± 0.8	74.5	64.2	74.6	63.2
TxFM-S [DiverseRNA-1.4M]	50.5 ± 0.2	50.1 ± 0.6	24.5 ± 0.6	91.2 ± 0.8	70.4	58.3	70.7	59.0
TxFM-B [DiverseRNA-1.4M]	50.0 ± 0.2	46.0 ± 1.3	23.4 ± 0.5	92.6 ± 0.6	69.1	58.3	69.8	57.5
Tabula Sapiens								
(Lib+Log)Norm+2000HVG	46.1 ± 1.0	65.7 ± 0.5	37.8 ± 0.9	86.7 ± 0.6	64.7	13.7	67.1	20.4
PCA [DiverseRNA-1.4M]	48.4 ± 0.2	70.9 ± 0.4	52.1 ± 1.1	83.2 ± 0.5	76.6	27.2	79.1	30.9
scGPT	50.2 ± 0.3	75.1 ± 0.4	58.8 ± 0.7	78.3 ± 0.7	77.2	30.5	78.3	31.4
AIDO.Cell	36.8 ± 0.5	65.0 ± 0.5	38.4 ± 1.9	65.7 ± 1.1	76.4	30.0	78.2	36.6
scTab	49.8 ± 0.4	75.2 ± 0.3	56.8 ± 0.5	84.1 ± 0.7	75.4	17.7	75.7	21.2
scVI [DiverseRNA-1.4M]	47.0 ± 0.4	72.4 ± 0.4	51.7 ± 1.1	60.4 ± 1.4	74.2	20.9	76.3	25.5
STATE	50.8 ± 0.1	78.2 ± 0.2	59.2 ± 1.6	89.2 ± 0.4	80.1	38.2	81.7	40.2
Transcriptformer	49.7 ± 0.3	73.4 ± 0.2	51.1 ± 1.4	71.2 ± 0.9	78.5	33.3	79.8	34.1
TxFM-B [TF-Sapiens data]	52.6 ± 0.2	75.8 ± 0.4	53.4 ± 2.1	69.4 ± 1.0	76.6	28.9	79.8	33.2
TxFM-S [DiverseRNA-1.4M]	52.1 ± 0.3	75.2 ± 0.5	55.1 ± 1.1	76.2 ± 0.9	77.4	29.2	77.8	32.4
TxFM-B [DiverseRNA-1.4M]	51.6 ± 0.2	74.9 ± 0.3	55.6 ± 0.7	73.5 ± 1.0	78.6	32.0	79.4	33.8

For each cell i with true label ℓ_i, $a_i^{(l)}$ is the mean distance to other cells with the same label, and $b_i^{(l)}$ is the mean distance to cells from the nearest different label. High ASW indicates well-separated cell types.

B.2.2 Batch integration task

Batch integration evaluates the dual challenge of removing technical artifacts (batch effects) while preserving biological signal. Batches represent different experimental conditions that should ideally contain identical cell type distributions. Results are obtained by subsampling 10,000 cells with 10 random seeds and reporting the average. Two metrics assess different aspects of this balance:

Batch-Corrected Biological Preservation ($\mathrm{ASW}_{l/b}$)**:** Measures whether cell types remain distinguishable within individual batches:

$$\mathrm{ASW}_{l/b} = \frac{1}{n}\sum_{i=1}^{n}\frac{b_i^{(l/b)} - a_i^{(l/b)}}{\max(a_i^{(l/b)}, b_i^{(l/b)})} \tag{23}$$

For each cell i with label ℓ_i and batch β_i, $a_i^{(l/b)}$ is the mean distance to other cells with the same label within the same batch, and $b_i^{(l/b)}$ is the mean distance to cells with different labels but within the same batch. This metric restricts comparisons to within-batch only, testing whether biological structure generalizes across experimental conditions.

Principal Component Regression comparison ($\mathrm{PCR_c}$)**:** Measures batch effect reduction by comparing variance explained by batch labels before and after integration [5] :

$$\mathrm{PCR_c} = \max\left(0, \frac{R^2_{\text{pre}} - R^2_{\text{post}}}{R^2_{\text{pre}}}\right) \tag{24}$$

where R^2_{pre} and R^2_{post} are the variance in the first 50 principal components explained by batch labels in the log-normalized data and embedding spaces, respectively. The $\max(0, \cdot)$ operation ensures the score is non-negative when integration increases batch effects. Values closer to 1 indicate higher batch effect removal.

B.2.3 Classification probing task

While not part of the original benchmark by Kedzierska et al. (2025), classification probing provides a complementary assessment of representation quality. Linear probes evaluate whether cell type information is linearly separable in the embedding space, while MLP probes test if this information can be recovered through simple non-linear transformations. The probe's performance reflects how well the model has organized biologically meaningful structure within its learned representations.

We follow an evaluation protocol with batch-aware data splitting to prevent information leakage. Data is split at the batch level with 75% for training and 25% for testing, ensuring no batch appears in both sets. Within each training fold, we further reserve 20% of training data for validation using random sampling (not batch-aware).

For preprocessing, we apply StandardScaler fit on training data and applied to test data. The training protocol employs early stopping based on validation loss and ReduceLROnPlateau scheduling. Model selection is performed through hyperparameter grid search over learning rates $\{1 \times 10^{-4}, 5 \times 10^{-4}, 1 \times 10^{-5}\}$ with fixed weight decay 1×10^{-4}, using AdamW optimizer and batch size 2048. The configuration maximizing the combined metric $(\mathrm{Acc} + \mathrm{F1\,macro})/2$ is selected.

We evaluate two architectures: linear probes and 2-layer MLPs with hidden dimensions $[512, 256]$ and dropout rate 0.5. Performance is measured using two complementary metrics:

Accuracy (Acc)**:**

$$\mathrm{Acc} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

Macro-averaged F1 Score ($\mathrm{F1\,macro}$)**:**

$$\mathrm{F1\,macro} = \frac{1}{C}\sum_{c=1}^{C}\mathrm{F1}_c$$

where $\mathrm{F1}_c = \frac{2 \cdot \mathrm{Precision}_c \cdot \mathrm{Recall}_c}{\mathrm{Precision}_c + \mathrm{Recall}_c}$ for class c, and C is the number of classes. Unlike accuracy, $\mathrm{F1\,macro}$ provides equal weight to all cell types regardless of frequency, making it particularly suitable for imbalanced single-cell datasets where rare cell types are as important as abundant ones.

Table 9: **Cell type representation performance averaged across the five datasets from the Kedzierska et al. (2025) benchmark.** We evaluate clustering (NMI, ARI, ASW), batch integration ($ASW_{l/b}$), and classification probing accuracy. $\diamondsuit$ and $\star$ denote, respectively, simple count baseline (no training) and models pretrained without test distribution overlap (zero-shot). For each metric, the best performance is in **bold**, second best is underlined. The rank column expresses the average rank of a method across metrics and is used to sort methods from best to worst. Table 8 shows per-dataset results.

		Clustering			Batch Int.	Classification			
						Linear Probe		2-layer MLP	
Rank↓	model	ASW ↑	NMI ↑	ARI ↑	$ASW_{l/b}$ ↑	Acc ↑	F1 ↑	Acc ↑	F1 ↑
3.1	TxFM-B [TF-Sapiens data]	53.9±0.2	65.8±0.6	46.6±1.8	82.4±0.6	85.6	**71.6**	86.5	**72.7**
4.4	STATE-SE	50.9±0.1	63.9±0.6	41.9±1.3	**95.4**±0.2	85.1	70.5	85.9	71.5
4.6	scGPT	53.5±0.2	65.5±0.6	50.9±1.6	87.0±0.6	84.0	68.5	83.9	69.0
4.7	TxFM-B [DiverseRNA-1.4M] ⋆	**54.4**±0.1	64.2±0.7	46.0±0.9	85.7±0.6	84.7	68.3	85.3	69.0
5.0	TxFM-S [DiverseRNA-1.4M] ⋆	**54.5**±0.2	65.1±0.5	46.2±1.1	85.6±0.6	84.6	67.4	85.1	68.8
6.0	AIDO.Cell	47.6±0.3	58.2±0.8	36.3±1.9	80.3±0.6	**86.5**	71.5	**86.7**	72.1
6.2	scTab	53.6±0.2	**68.3**±0.5	**57.2**±1.6	91.1±0.4	80.8	59.0	81.7	62.2
6.2	Transcriptformer [TF-Sapiens data]	52.3±0.1	66.0±0.8	47.3±3.1	82.8±0.7	83.0	67.1	82.8	66.3
7.8	(Lib+Log)Norm+2000HVG $\diamondsuit$	50.4±0.2	62.8±0.5	42.9±0.9	93.6±0.2	79.1	62.6	81.5	67.3
7.9	PCA [DiverseRNA-1.4M] ⋆	50.0±0.1	55.3±0.9	36.6±1.8	91.2±0.4	82.8	66.1	83.9	67.4
9.9	scVI [DiverseRNA-1.4M] ⋆	50.4±0.2	56.8±0.5	37.5±1.4	80.5±0.9	78.6	61.4	80.1	64.7

B.2.4 Cell type representation benchmarking, scIB results

Table 9 presents an aggregate assessment of inference-time model performance on cell-type representation learning tasks, revealing a tension between geometric and informational objectives. While unsupervised clustering rewards models that compress cells into dense, local neighborhoods, this can sometimes come at the cost of global separability. Our results show that a representation can appear geometrically scattered (lower clustering scores) while still encoding robust, linearly separable cell identities (high probing accuracy), necessitating a holistic view of performance.

Aggregate performance and zero-shot significance. Navigating this trade-off, TxFM-B trained on TF-Sapiens emerges as the most capable generalist, delivering consistent top-tier performance across both clustering and classification tasks. This highlights the scaling efficiency of our architecture: when trained on large-scale atlases, TxFM-B outperforms STATE-SE Adduri et al. (2025), a model with nearly $4\times$ the learnable parameters trained on a dataset roughly $3\times$ larger. It is crucial to contextualize these results within the training distributions. Models trained on large atlases can benefit from partial test leakage, as evaluation datasets like Tabula Sapiens (Jones et al., 2022) are frequently subsets of their self-supervised training corpora. In contrast, the $\star$ marked versions of TxFM operate in a strict zero-shot setting, yet perform near the top of the ranking. This underscores that representations learned from curated perturbational data can generalize very effectively.

Task-specific inductive biases. Some models show performance skewed toward specific metrics, revealing distinct design trade-offs. scTab Fischer et al. (2024) excels at clustering. Its TabNet backbone Arik & Pfister (2021) uses sequential attention which, similarly to decision trees, can create sharp partitions ideal for discrete clustering metrics (ASW/NMI/ARI). However, this can result in a fragmented geometry which is not conducive to strong classification. AIDO.Cell (Ho et al., 2024), conversely, excels at classification but struggles with clustering, suggesting its embeddings capture rich signal but distribute it with high intra-class variance (e.g., retaining library size noise). Supervised probes can learn to ignore this noise, but unsupervised clustering algorithms cannot.

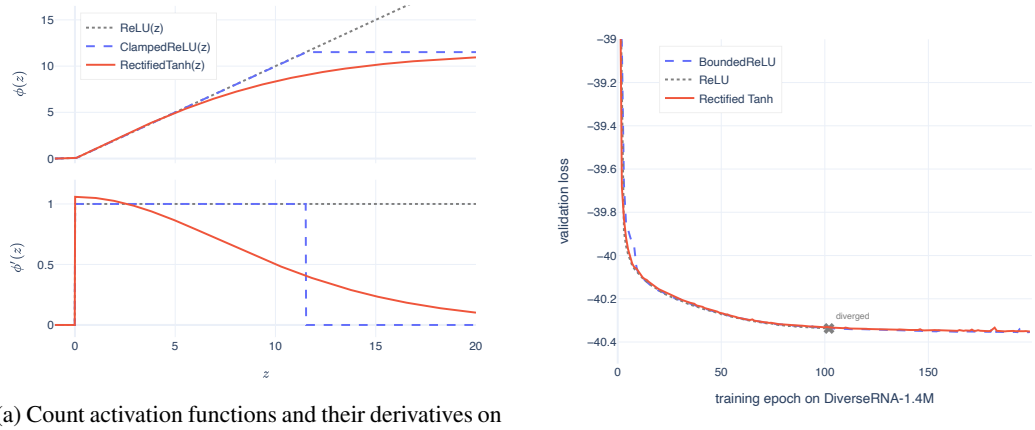
Ultimately, no single model dominates every metric, but TxFM offers an effective compromise: retaining the local geometry needed for clustering and the invariance required for batch correction, while ensuring cell types remain extractable for downstream classification.

[5]While the original benchmark implementation used a single call to `pcr()`, which measures absolute remaining batch effects, the paper explicitly describes PCR as comparing "the proportion of the variance that is explained by the batch variable between the original dataset and the embeddings of the model", which would instead correspond to `pcr_comparison()`. With either implementation, PCR values are highly variable and inconsistent between models and datasets. Due to this ambiguity, we opted not to report this metric.

C Ablation experiments

When testing differences in perturbation consistency (perts), recall for encoder gene tokens (enc), and recall for decoder gene weights (dec) between different versions of TxFM, we compared the metric corresponding to each ablation against the Default setting. For perts comparisons, we used a 2-proportion z-test. For enc and dec comparisons, we used a 2-tailed t-test. All resulting p-values were adjusted with Benjamini-Hochberg multiple test correction at 0.05 error rate across all pairs of ablations within each of the three metrics.

C.1 Activation functions



(a) Count activation functions and their derivatives on logits z.

(b) Model validation loss per epoch.

Figure 10: Comparing TxFM decoder count activation functions for models trained in ablation (f). A pure ReLU activation diverges to NaN midway through training. Clamping ReLU prevents divergence and achieves comparable loss, but has lower zero-shot performance in representing perturbations compared to our rectified tanh activation function (Table 6).

We use a novel activation function called *rectified tanh* (Equation 1) in the decoder to safeguard against predicting counts outside of the training distribution, by leveraging the normalization procedure we employ in preprocessing. Our activation function asymptotically tends to the library size, i.e. the largest possible value an entire expression vector of a preprocessed sample can sum up to. We add 1 to the library size L to mirror the pseudocounts added to input data in our preprocessing. (The activation can be generalized to $\phi(z) = \alpha\text{ReLU}(\tanh(\frac{z}{\beta}))$ and unit-size gradient steps can be ensured by setting $\alpha = \beta$; we recommend setting $\alpha = \log(L+1)$ where L is the library size the data is pre-normalized to ($L = 10^5$). In practice, our original implementation set $\alpha = \log(L+1)$ and $\beta = 4e$.)

Indeed, given that the count data is always non-negative, we first tried ReLU as the count activation function ϕ, but, halfway through training, the model diverges. While a softmax activation struggled to train effectively, we found that simply clamping the ReLU activation based on the maximum possible library size could prevent divergence:

$$\phi_{ClampedReLU}(z) = \min(\text{ReLU}(z), \log(L+1)).$$

Figure 10 compares rectified tanh to a ReLU and clamped ReLU. Our novel activation function achieved the best zero-shot performance in perturbation representation (Table 6).

C.2 Dataset curation

We evaluate these dataset curation strategies in Table 6 (h):

1. Removing all K562 cells from the training data, yielding 932K training samples.
2. Adding only the 72,000 non-targeting control K562 cells from the Replogle et al. (2022) dataset, in addition to the 932K non-K562 samples in the DiverseRNA dataset.

Table 10: **Backbone choice for existing models**. Average score across all benchmarking tasks in Bendidi et al. (2024) on RPE1, HEPG2, and Jurkat datasets for different backbone choices of existing FMs with multiple pretrained backbones. Grayed backbones are the best performing backbones used in Table 2.

Backbone	RPE1	HEPG2	Jurkat
GenePT ADA	39.29	35.45	32.07
GenePT Large	39.82	34.77	32.55

(a) **GenePT** ablation of best performing backbone.

Backbone	RPE1	HEPG2	Jurkat
AIDO.cell 3M	28.03	29.21	25.87
AIDO.cell 10M	31.01	31.84	26.47
AIDO.cell 100M	37.45	34.52	31.53

(b) **AIDO.cell** ablation of best performing backbone.

Backbone	RPE1	HEPG2	Jurkat
scPrint Small	25.70	22.03	24.15
scPrint Medium	30.84	24.22	26.58
scPrint Large	31.62	23.80	27.04

(c) **scPrint** ablation of best performing backbone.

Backbone	RPE1	HEPG2	Jurkat
Transcriptformer Exemplar	34.53	31.54	29.96
Transcriptformer Metazoa	34.52	31.51	30.00
Transcriptformer Sapiens	34.88	31.55	29.92

(d) **TranscriptFormer** ablation of best performing backbone.

3. Our default approach, which follows an established strategy shown to improve the performance of MAEs on biological experimental data (Kenyon-Dean et al., 2025) by filtering the perturbational K562 training data to only include cells from perturbations distinguishable from the rest. Specifically, we selected *phenoprint* perturbations by using an earlier version of TxFM to inference the data, computed perturbation consistency as described above (§ B.1.1), and kept perturbations with $p < 0.1$. This more permissive threshold allows us to increase the number of training samples while still keeping perturbations with reasonably distinct transcriptional profiles.
4. Training with the full uncurated dataset of K562 cells, yielding 2.8M training samples, and holding the compute budget constant by adjusting the number of training epochs according to dataset size; we also evaluate doubling the training compute on the 2.8M dataset to 200 epochs [2x], to about 2,000 H100 GPU hours.
5. Training TxFM-B with 4x more compute on a different dataset of 57 million cells sampled from 72 large-scale atlases from CZI et al. (2025).

D Dataset-specific benchmarking results

Figure 11 and Table 10 show ablations of the best backbone for each existing model and the best performing number of principal components for PCA on this benchmark. Tables 11, 12, 13 show the results on the Bendidi et al. (2024) benchmark for the RPE-1, HEPG2, and Jurkat datasets, correspondingly.

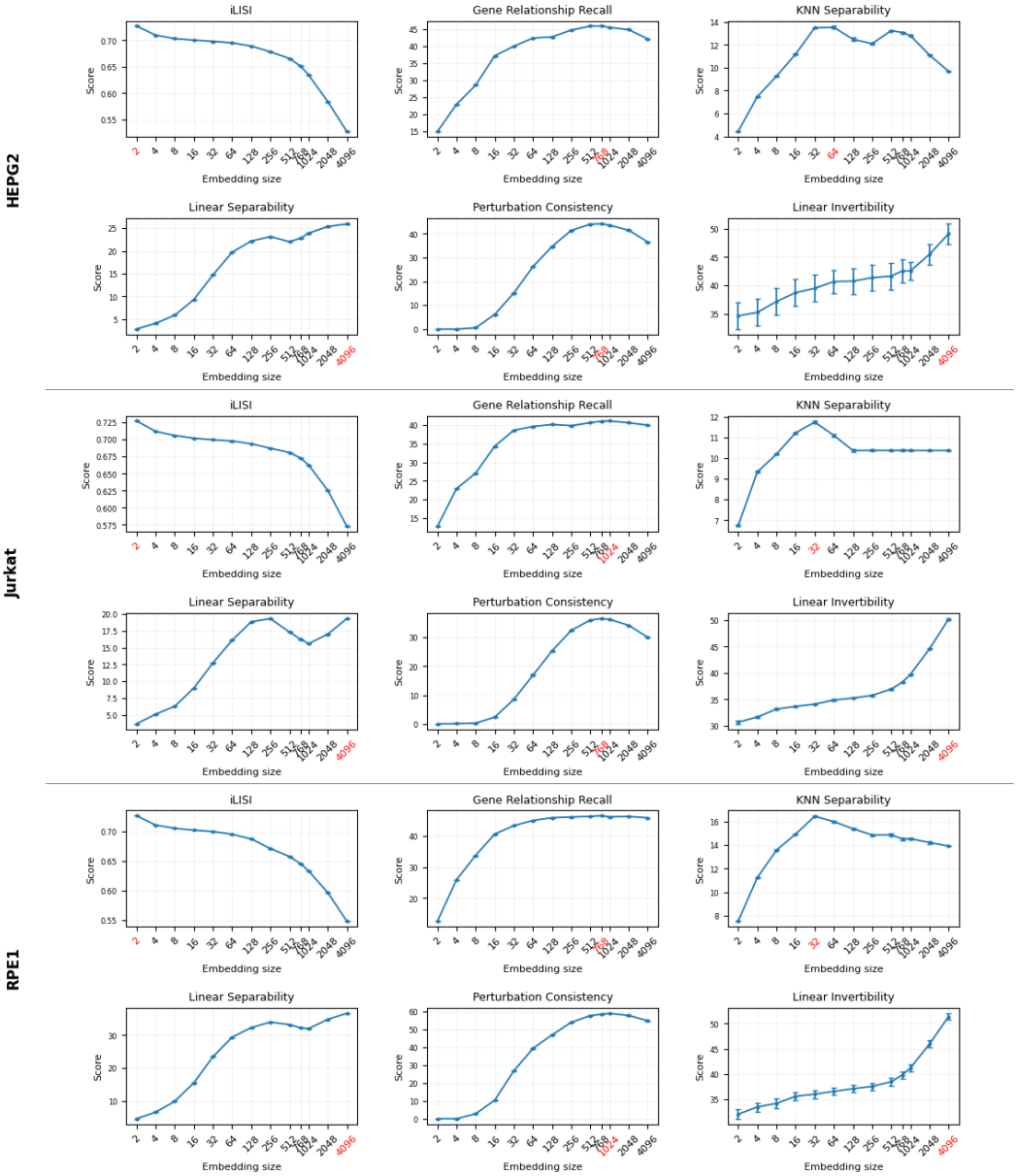


Figure 11: **Dimension size ablation for PCA.** Bendidi et al. (2024) benchmarking results on RPE1, HEPG2, and Jurkat datasets after applying PCA on raw counts with different numbers of principal components. Red is optimal size for each downstream task.

Table 11: Overview of model performance across different settings and perturbational downstream tasks on the scores for **RPE1** dataset. Metrics are mean ± std across 5 random seeds.

Models	ilisi↑	lin↑	knn↑	p.cst↑	bmdb↑	inv↑	avg↑
Transductive SSL models							
PCA	0.63±0.00	32.03±0.05	14.54±0.09	58.79±0.14	46.03±0.00	41.17±0.71	42.64
scVI	0.69±0.00	19.68±0.02	12.69±0.09	38.27±0.15	41.86±0.00	43.93±0.72	37.65
TxFM-B finetuned (ours)	0.70±0.00	42.10±0.01	17.82±0.07	48.73±0.28	47.27±0.00	42.64±0.71	44.85
Count data baselines							
Random label shuffle	0.70±0.00	0.85±0.01	8.20±0.01	0.00±0.00	11.17±0.00	25.34±0.80	19.40
Raw data	0.61±0.00	36.68±0.06	13.92±0.02	36.23±0.16	44.06±0.00	59.63±0.58	41.93
(Lib+Log)Norm	0.61±0.00	30.07±0.03	14.74±0.08	37.60±0.13	42.37±0.00	62.18±0.67	41.37
(Lib+Log)Norm+5k HVG	0.64±0.00	26.66±0.03	14.17±0.05	41.61±0.25	43.24±0.00	52.02±0.67	40.41
(Lib+Log)Norm+1024HVG	0.69±0.00	23.67±0.02	14.02±0.05	36.73±0.32	43.45±0.00	41.25±0.68	38.06
ChatGPT Embeddings							
GenePT-Large	0.73±0.00	27.49±0.03	12.14±0.09	43.46±0.25	44.35±0.00	37.55±0.75	39.82
Pretrained FMs							
Geneformer [CELLxGENE]	0.71±0.00	4.08±0.01	8.22±0.02	0.00±0.00	11.04±0.00	25.80±0.77	20.15
GeneJEPA [Tahoe-100M]	0.70±0.00	4.68±0.00	8.37±0.03	0.45±0.00	22.19±0.00	36.17±0.25	23.74
scTab [CELLxGENE]	0.70±0.00	6.69±0.01	8.76±0.03	1.73±0.03	31.60±0.00	35.33±0.70	25.80
CellPLM	0.71±0.00	7.32±0.00	9.50±0.06	2.18±0.01	23.74±0.00	36.89±0.78	25.20
UCE [CELLxGENE]	0.70±0.00	7.42±0.00	9.34±0.05	10.24±0.10	29.36±0.00	36.74±0.73	27.34
scCello [CELLxGENE]	0.71±0.00	14.04±0.01	10.52±0.10	1.37±0.00	31.03±0.00	37.16±0.85	27.53
scPrint-L [CELLxGENE]	0.70±0.00	16.26±0.02	11.38±0.04	16.38±0.19	37.59±0.00	37.44±0.76	31.62
scGPT [CELLxGENE]	0.71±0.00	12.58±0.01	10.34±0.03	18.26±0.11	34.19±0.00	37.55±0.74	30.67
Tahoe-x1 [Tahoe-100M]	0.71±0.00	17.16±0.00	11.59±0.01	22.64±0.39	37.23±0.00	39.16±0.24	33.15
TranscriptFormer-Sapiens [CELLxGENE]	0.71±0.00	18.65±0.01	10.90±0.05	34.22±0.19	35.08±0.00	39.40±0.72	34.88
AIDO.Cell-100M [CELLxGENE]	0.71±0.00	27.66±0.02	12.88±0.02	31.62±0.40	42.61±0.00	38.54±0.71	37.45
Cell2Sentence	0.70±0.00	26.73±0.00	13.10±0.00	40.09±0.20	43.38±0.00	39.83±0.29	39.01
TxFM-B [TF-Sapiens data] (ours)	0.71±0.00	27.54±0.00	12.55±0.07	36.48±0.39	42.11±0.00	41.47±0.70	38.57
STATE-SE [CxG + Tahoe-100M + scBC]	0.69±0.00	29.05±0.00	13.31±0.10	40.80±0.20	44.59±0.00	44.97±0.22	40.38
Fit on our public train set							
PCA [DiverseRNA-1.4M]	0.69±0.00	16.10±0.01	12.65±0.08	29.59±0.29	40.01±0.00	46.38±0.66	35.72
scVI [DiverseRNA-1.4M]	0.71±0.00	22.10±0.01	13.08±0.05	26.25±0.12	39.07±0.00	39.52±0.77	35.18
Linear AE [DiverseRNA-1.4M]	0.72±0.00	24.24±0.03	15.37±0.23	29.92±0.05	44.87±0.00	42.35±0.51	38.21
Ridge PCA [DiverseRNA-1.4M]	0.71±0.00	28.21±0.07	16.06±0.29	34.10±0.11	45.94±0.00	41.79±0.81	39.61
TxFM-B [DiverseRNA-1.4M] (ours)	0.70±0.00	37.61±0.02	15.13±0.04	42.71±0.61	45.33±0.00	41.54±0.73	42.17

E NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The benchmarking analysis (§ 4.1, § 4.2) and comprehensive ablation studies (§ 4.4)—representing 10,000s of hours of H100 GPU training and inference compute, comprising many training evaluations along different data and architecture training splits—accurately reflect the paper's claims (state-of-the-art inductive gene perturbation representation learning and gene relationship recall from model parameters) as being the result of careful SSL dataset curation and a novel SSL model architecture design (a transcriptomics MAE with transformer encoder and MLP decoder off CLS token).

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Table 12: Overview of model performance across different settings and perturbational downstream tasks on the scores for **HEPG2** dataset. Metrics are mean ± std across 5 random seeds.

Models	ilisi↑	lin↑	knn↑	p.cst↑	bmdb↑	inv↑	avg↑
Transductive SSL models							
PCA	0.63±0.00	23.90±0.09	12.81±0.05	43.56±0.10	45.57±0.00	42.54±1.57	38.63
scVI	0.69±0.00	12.61±0.04	11.47±0.14	27.84±0.07	45.22±0.00	47.03±2.27	35.63
TxFM-B finetuned (ours)	0.70±0.00	29.27±0.04	15.81±0.05	38.40±0.19	46.24±0.00	44.55±3.18	40.78
Count data baselines							
Random label shuffle	0.70±0.00	0.66±0.01	5.45±0.05	0.00±0.00	12.95±0.00	28.14±2.58	19.68
Raw data	0.60±0.00	23.21±0.74	9.75±0.06	2.60±0.12	32.15±0.00	56.38±2.01	30.81
(Lib+Log)Norm	0.65±0.00	21.51±0.05	11.93±0.05	10.49±0.22	35.32±0.00	60.09±1.68	34.12
(Lib+Log)Norm+5k HVG	0.67±0.00	19.22±0.05	11.38±0.05	8.09±0.13	34.75±0.00	50.54±2.00	31.88
(Lib+Log)Norm+1024HVG	0.69±0.00	15.04±0.04	10.74±0.03	24.13±0.08	44.06±0.00	44.46±2.16	34.65
ChatGPT Embeddings							
GenePT-Large	0.73±0.00	18.65±0.06	8.28±0.03	24.78±0.05	42.01±0.00	40.91±2.40	34.77
Pretrained FMs							
Geneformer [CELLxGENE]	0.71±0.00	2.21±0.00	5.44±0.02	0.04±0.00	11.30±0.00	28.41±2.45	19.89
GeneJEPA [Tahoe-100M]	0.70±0.00	2.64±0.00	5.74±0.00	0.41±0.00	23.68±0.00	38.86±1.82	23.64
scTab [CELLxGENE]	0.70±0.00	4.59±0.02	6.67±0.02	1.93±0.03	32.89±0.00	38.80±2.37	25.91
CellPLM	0.71±0.00	5.12±0.02	7.52±0.03	3.68±0.07	23.03±0.00	40.68±2.29	25.25
UCE [CELLxGENE]	0.70±0.00	5.23±0.01	7.20±0.03	6.31±0.01	31.32±0.00	40.47±2.31	26.92
scCello [CELLxGENE]	0.71±0.00	8.50±0.03	7.44±0.03	6.95±0.03	36.77±0.00	40.14±2.38	28.50
scPrint-L [CELLxGENE]	0.71±0.00	2.96±0.00	5.93±0.03	0.16±0.00	26.89±0.00	35.57±2.47	23.80
scGPT [CELLxGENE]	0.70±0.00	7.21±0.01	8.51±0.04	11.46±0.03	36.69±0.00	41.30±2.27	29.35
Tahoe-x1 [Tahoe-100M]	0.70±0.00	9.53±0.02	9.83±0.05	15.39±0.04	39.30±3.28	43.25±1.51	31.36
TranscriptFormer-Sapiens [CELLxGENE]	0.70±0.00	12.16±0.02	8.94±0.04	19.33±0.03	35.14±0.00	42.77±2.26	31.55
AIDO.Cell-100M [CELLxGENE]	0.71±0.00	16.98±0.03	10.26±0.04	25.05±0.05	41.42±0.00	42.28±2.27	34.52
Cell2Sentence	0.70±0.00	16.12±0.03	9.99±0.08	24.52±0.00	42.24±0.00	43.44±1.85	34.49
TxFM-B [TF-Sapiens data] (ours)	0.70±0.00	17.81±0.04	10.59±0.08	25.28±0.19	43.31±0.00	43.87±2.48	35.30
STATE-SE [CxG + Tahoe-100M + scBC]	0.69±0.00	17.20±0.00	10.79±0.05	23.13±0.08	43.94±0.00	45.15±1.77	35.10
Fit on our public train set							
PCA [DiverseRNA-1.4M]	0.69±0.00	11.11±0.05	11.50±0.06	20.53±0.13	43.34±0.00	48.21±2.10	34.00
scVI [DiverseRNA-1.4M]	0.70±0.00	14.16±0.03	10.83±0.00	17.28±0.05	40.00±0.00	43.17±2.25	32.69
Linear AE [DiverseRNA-1.4M]	0.72±0.00	16.59±0.07	11.82±0.03	20.45±0.31	45.74±0.00	45.63±0.16	35.44
Ridge PCA [DiverseRNA-1.4M]	0.71±0.00	20.23±0.09	12.43±0.11	22.87±0.22	46.92±0.00	46.16±0.24	36.76
TxFM-B [DiverseRNA-1.4M] (ours)	0.70±0.00	25.93±0.07	13.14±0.06	32.47±0.21	45.26±0.00	44.40±2.21	38.63

Justification: The limitations section appears immediately after the conclusion in § 5.

Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

Table 13: Overview of model performance across different settings and perturbational downstream tasks on the scores for **Jurkat** dataset. Metrics are mean ± std across 5 random seeds.

Models	ilisi↑	lin↑	knn↑	p.cst↑	bmdb↑	inv↑	avg↑
Transductive SSL models							
PCA	0.66±0.00	15.59±0.02	10.37±0.03	36.27±0.10	41.13±0.00	39.70±0.11	34.89
scVI	0.69±0.00	9.95±0.01	9.92±0.08	23.26±0.10	40.50±0.00	42.39±0.07	32.62
TxFM-B finetuned (ours)	0.69±0.00	25.64±0.03	11.99±0.01	36.56±0.10	42.64±0.00	41.52±0.10	38.01
Count data baselines							
Random label shuffle	0.70±0.00	0.54±0.00	7.46±0.04	0.04±0.00	11.59±0.00	24.77±0.08	19.20
Raw data	0.63±0.00	20.47±0.05	10.36±0.04	11.55±0.02	35.25±0.00	57.28±0.14	33.06
(Lib+Log)Norm	0.62±0.00	16.73±0.02	11.07±0.04	14.69±0.07	34.71±0.00	60.50±0.19	33.37
(Lib+Log)Norm+5k HVG	0.64±0.00	13.07±0.01	10.76±0.01	20.95±0.06	37.29±0.00	50.61±0.13	32.94
(Lib+Log)Norm+1024HVG	0.68±0.00	9.67±0.01	10.38±0.03	16.95±0.10	38.68±0.00	39.44±0.09	30.67
ChatGPT Embeddings							
GenePT-Large	0.74±0.00	14.30±0.02	8.58±0.01	24.47±0.16	38.11±0.00	35.74±0.48	32.55
Pretrained FMs							
Geneformer [CELLxGENE]	0.71±0.00	2.97±0.00	7.44±0.03	0.00±0.00	10.60±0.00	25.00±0.32	19.63
GeneJEPA [Tahoe-100M]	0.70±0.00	3.14±0.01	7.59±0.04	0.04±0.00	14.96±0.00	33.56±0.21	21.66
scTab [CELLxGENE]	0.70±0.00	3.70±0.01	7.72±0.04	0.12±0.00	21.12±0.00	32.79±0.14	22.69
CellPLM	0.71±0.00	4.27±0.00	8.19±0.03	1.79±0.07	23.99±0.00	34.81±0.17	24.06
UCE [CELLxGENE]	0.71±0.00	4.97±0.00	7.97±0.02	2.95±0.01	28.61±0.00	34.87±0.16	25.07
scCello [CELLxGENE]	0.70±0.00	7.26±0.01	8.69±0.04	0.37±0.00	28.45±0.00	34.77±0.08	25.09
scPrint-L [CELLxGENE]	0.70±0.00	8.43±0.01	9.24±.01	5.13±0.06	33.34±0.00	35.48±0.13	27.04
scGPT [CELLxGENE]	0.71±0.00	7.66±0.01	8.61±0.05	5.53±0.17	32.45±0.00	35.70±0.12	26.86
Tahoe-x1 [Tahoe-100M]	0.71±0.00	9.61±0.01	9.40±0.00	10.15±0.00	34.02±0.00	36.37±0.15	28.47
TranscriptFormer-Sapiens [CELLxGENE]	0.70±0.00	10.77±0.01	8.68±0.04	20.40±0.04	31.38±0.00	37.30±0.14	29.92
AIDO.Cell-100M [CELLxGENE]	0.71±0.00	15.07±0.00	9.46±.05	19.54±0.07	37.08±0.00	36.69±0.12	31.53
Cell2Sentence	0.71±0.00	13.93±0.00	9.39±0.04	26.21±0.00	39.11±0.00	37.21±0.08	32.82
TxFM-B [TF-Sapiens data] (ours)	0.71±0.00	16.52±0.01	9.82±0.05	22.59±0.13	37.97±0.00	39.77±0.11	32.98
STATE-SE [CxG + Tahoe-100M + scBC]	0.69±0.00	17.15±0.00	9.92±0.04	24.45±0.00	39.44±0.00	42.65±0.01	33.81
Fit on our public train set							
PCA [DiverseRNA-1.4M]	0.69±0.00	7.60±0.00	10.04±0.05	16.83±0.03	37.44±0.00	44.56±0.09	30.91
scVI [DiverseRNA-1.4M]	0.71±0.00	12.97±0.01	10.43±0.02	16.18±0.09	36.16±0.00	37.36±0.11	30.68
Linear AE [DiverseRNA-1.4M]	0.71±0.00	14.13±0.05	11.13±0.14	17.45±0.25	40.18±0.00	40.23±0.36	32.49
Ridge PCA [DiverseRNA-1.4M]	0.71±0.00	16.53±0.09	11.94±0.11	20.84±0.19	42.08±0.00	40.26±0.53	33.85
TxFM-B [DiverseRNA-1.4M] (ours)	0.69±0.00	23.79±0.01	11.59±0.03	31.27±0.17	42.08±0.00	40.42±0.11	36.52

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: This paper is empirical in nature.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The training dataset is curated from fully public data sources and the evaluation code is from previously published work: Bendidi et al. (2024) for perturbation evaluation, and Celik et al. (2024) for gene relationship evaluation. The new TxFM model architecture is fully described and detailed. We also commit to releasing additional code resources upon acceptance.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: As described above, training data is public, evaluation data is public, and evaluation code is also public.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: This is fully described in both the paper text and in the appendix throughout.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Error bars are included in primary results either in the appendix or main text (e.g., Table 4).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Each newly trained model is clearly documented with its compute requirements, see § A.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The datasets and models in this work are used correctly and documented thoroughly throughout the work. No ethical dilemmas are present in this work.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: This work may positively impact biomedical research by improving representation learning for transcriptomic data, enabling more effective analysis of perturbational screens, gene-gene relationships, and potential discovery of disease mechanisms. These advances could support target discovery and more efficient biological hypothesis generation. Potential negative impacts are limited but include overinterpretation of learned gene relationships as causal mechanisms, biased conclusions from training data skewed toward oncology and public atlas datasets, and inappropriate use of representations in clinical or therapeutic decision-making without validation. We mitigate these risks by evaluating only on public research datasets, clearly framing the model as a research tool rather than a clinical system, reporting limitations around dataset bias and generalization, and planning to release code/checkpoints to support reproducibility and independent evaluation.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.

- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The data and many transcriptomics models are already publicly available.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite each data source used in this work, and all are already available on `https://cellxgene.cziscience.com/datasets`

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We will release code / model checkpoint upon publication with clear and appropriate documentation.

    Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.