BIOLOGICALLY INFORMED GENERATION OF PROTEIN LEVELS WITH FLOW MATCHING

Anonymous authors
Paper under double-blind review

ABSTRACT

Cellular protein levels are governed by RNA and other complex factors making them hard to predict computationally. Limited protein experimental data, particularly at single cell (sc) and spatial resolution, often necessitates the use of scRNA-seq as a proxy despite poor direct correlation. Current methods estimating single cell protein levels largely use discriminative predictors while omics data is inherently stochastic. Since proteins measured are often few and widely differ across experiments, existing methods require repeated training and tuning with limited generalizablity. They fail to incorporate batch and sample effects; phenotypic priors such as cell type, disease, or condition; and mechanistic priors such as structure or function encoded in sequence. Most methods fail to incorporate spatial locations of cells in tissue slices. We address these issues with RnaProFlo, a conditional generative foundation model using rectified flow matching. It generates surface or intracellular protein levels at single cell or spatial spot resolution. Protein generation is conditioned on RNA expression, cell technical and biological priors, sequence and spatial location, if present. We develop a tokenizer that modulates expression levels based on protein language model derived sequence embeddings which inherently contain structure and functional information. It enables learning from non-overlapping protein feature sets across experiments and supports generalization to proteins unseen during training. Batch, disease, cell type and data source abstract metadata are embedded using a large language model (LLM) and globally condition the transformer ensuring dataset specific technical effects and biological conditions are incorporated. We further introduce gated graph structured token level cross attention with spatial distance bias during fine-tuning to enable generation of spatially informed proteomic data despite limited spatial training data. RnaProFlo outperforms other methods on paired single cell and spatial RNA-protein data effectively predicting held-out protein levels. Detailed ablation tests, zero-shot and fine-tuning experiments across seen and unseen proteins, tissues, biological conditions, sequencing technologies and species, collectively show RnaProFlo's superiority in estimating cellular and spatial protein from scRNA-seq or spatial transcriptomics (ST) expression and metadata.

1 INTRODUCTION

Proteins regulate gene expression, cell processes, metabolite production, intra and inter cellular communication and form the structural backbone of tissue. Single cell (sc) and spatially resolved protein measurement could reveal the mechanisms behind these biological processes. While DNA and RNA have been extensively measured, the proteome is vast with over 20,000 proteins and millions of poorly characterized variants or proteoforms. Experimental efforts are limited by cost, technological access and the number of proteins measured, particularly at single cell or spatial resolution. Curated experiment specific protein measurement further limits data reusability. To resolve these experimental limitations, RNA is largely used as a proxy for proteomic meaurement, despite low correlation between RNA and its corresponding translated proteins (Vogel & Marcotte, 2012).

Computational methods learn to predict protein levels from RNA expression with limited paired training data. Methods use linear models (Hanhart et al., 2024), kernel ridge regression (KRR) (Lance et al., 2022), recurrent neural networks (RNNs) (Lakkis et al., 2022), variational autoencoders (VAEs) (Gayoso et al., 2021; Wang et al., 2024b), adversarial networks (Yang et al.,

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

2023), residual networks (Zhou et al., 2024), sequence-guided RNA-protein multi-head cross attention (Zhao et al., 2026), diffusion (Yu et al., 2024) and transformer-based foundation models (Liu et al., 2025; Ji et al., 2025). Many (Hu et al., 2024) treat this as a cross modality data integration problem (Lotfollahi et al., 2022) while some focus on protein prediction from RNA (Liu et al., 2025; Ji et al., 2025; Xu et al., 2024; Zhao et al., 2026). Further discussion of related work is in A.1.

Current methods have several limitations. First, while RNA abundance informs protein levels, post translational modifications, translation efficiency, degradation and measurement noise introduce variability making RNA-protein mapping a many-many, stochastic, multimodal, high-dimensional problem poorly suited to data integration or discriminative problem formulations. Second, with the limited number of often non-overlapping protein features measured across different experiments and the diversity in sequencing technology, tissue, cell-type, species, and biological condition, adaptation to scenarios unseen during training such as new genes, proteins, tissue, cell-type, species, disease state, sequencing technology or perturbation with limited or no additional training is critical. Third, current models often represent cell- and dataset-level metadata as categorical covariates or special conditioning signals rather than semantically enriched representations which challenges single cell foundation model (scFM) generalization (Baek et al., 2025). Fourth, while scFMs are widely used, they are limited in their integration of biological priors and expression. Two genes or proteins with similar abundance need not map to similar downstream representations because their sequences encode different structural or functional contexts. scFM tokenizers additively combine gene and protein language model (PLM) embeddings (Kalfon et al., 2025) or using PLMs to encode genes and sample based on expression (Rosen et al., 2023). The former treats expression and sequence as independent components while the latter is suited for modeling cellular context but does not directly preserve the original quantitative expression values. Fifth, paired RNA-protein spatial data is extremely limited for training. Therefore, scFMs for spatial proteomic data generation must fine-tune on spatial spots while incorporating spatial tissue layout.

To address the aforementioned challenges, we develop RNAProFlo, a conditional generative multimodal foundation model for single cell and spatial proteomics. We integrate RNA and protein sequence embeddings, single cell or spatial expression profiles along with large language model (LLM) derived metadata embeddings to capture RNA to protein translation within cellular context. We model the problem with rectified flow matching using a transformer to learn denoising flow velocity. Cell and dataset metadata embeddings comprising batch, biological condition, cell type and associated contextual information globally condition the generative process through adaptive layer normalization (adaLN) of transformer layers. We integrate sequence and expression by modulating expression with protein language model (PLM) embeddings to generate token inputs for the transformer that are later decoded back to expression. We employ a three-stage strategy, training the encoder-decoder, flow matching transformer and end-to-end model in each of the three stages. For stage 3, we introduce an expression rollout loss so the decoder weights are updated based on generated latent proteins. Our design choices enable generalization in zero-shot mode or with limited finetuning to new proteins, species, sequencing technologies, and perturbation and spatial assays. RNAProFlo is trained on 3.2 million cells across 38 human and mouse datasets. We achieve state-of-the-art performance compared to related methods on average improving cosine similarity by 13.9% while reducing MSE by 48.2% relative to the strongest external method.We introduce a gated spatial niche token-level cross attention layer for fine-tuning our model on limited paired spatial spot based RNA-protein data, improving spatial cosine similarity by 3.3% higher reducing MSE by 23.3% on average. Our key contributions are:

- **Single cell and spatial proteomics generation via a flow matching foundation model** To our knowledge, RnaProFlo is the first conditional generative multi-modal foundation model for single cell and spatial proteomics and the first flow matching based scFM (Jiang et al., 2026). It generalizes to unseen genes, proteins, tissue, cell-type, species, disease state, sequencing technology or perturbation in zero-shot mode or with minimal fine-tuning.

- **Sequence-based modulation of RNA and protein expression** We develop a tokenizer with sequence based expression modulation to effectively integrate expression with sequence embeddings from a protein language model (PLM).

- **Global conditioning with LLM-derived cell and dataset metadata embeddings** RnaProFlo adaptively learns layer normalization weights from LLM embeddings of cell and metadata prompts for global conditioning of the protein denoising steps. The model

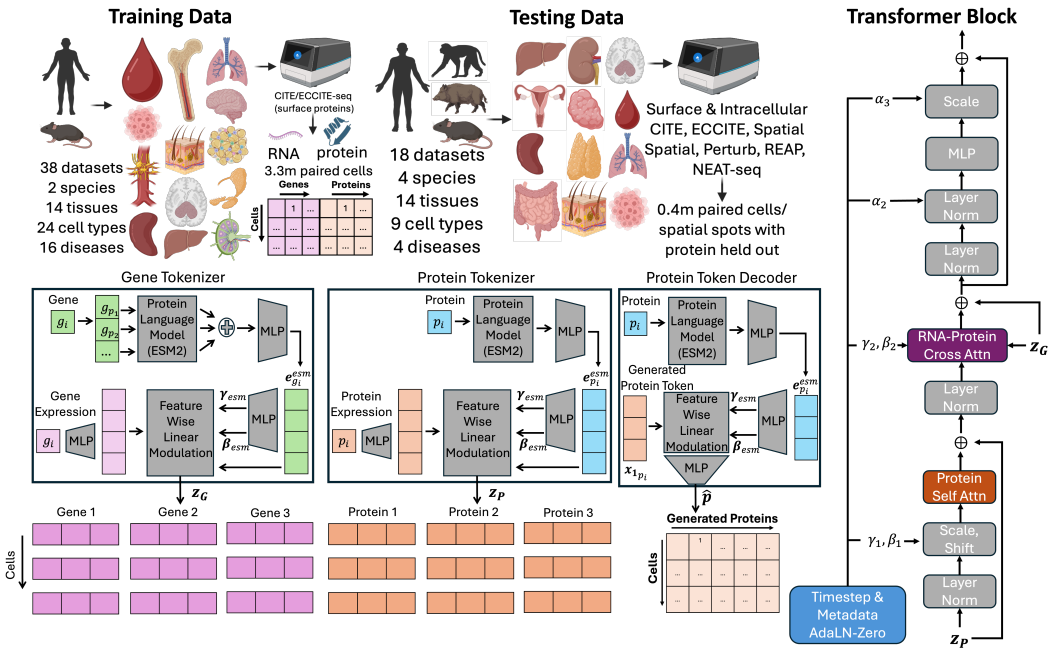


Figure 1: Overview of RnaProFlo

thus adapts to non-standard metadata across different datasets and captures underlying metadata semantics without dataset specific training.

- **Gated spatial niche token level cross attention leverages tissue structure during fine-tuning** We introduce a gated cross attention layer with spatial distance bias for spatially conditioned protein generation. Each protein token of a spot attends to every gene token of neighboring spots. The gate and spatial bias are learned from limited spatial data thus modeling interactions between protein levels at a spot and neighboring spot RNA.

2 METHODS

RNAProFlo is a multi-modal foundation model for single cell or spatial proteomics data generation conditioned on scRNA-seq or ST data, cellular context, and spatial tissue structure. RnaProFlo comprises four modules (Fig. 1): (i) Metadata embedder with LLM generated cell and dataset metadata embeddings (ii) Tokenizer with sequenced-based RNA and protein expression modulation (iii) Transformer blocks learning flow matching velocity conditioned on metadata and (iv) Protein sequence modulated decoder mapping latent protein tokens to expression.

2.1 PROBLEM FORMULATION

Given a dataset of single cells with RNA and protein measurements each cell i consists of normalized RNA expression $\mathbf{r}_i \in \mathbb{R}^G$, normalized protein expression $\mathbf{p}_i \in \mathbb{R}^P$, metadata embeddings $\mathbf{c}_i, \mathbf{d}_i \in \mathbb{R}^{d_{\text{meta}}}$ obtained from an LLM and corresponding to cell- and dataset-level information, respectively. Here, G denotes the number of highly variable genes selected, P denotes the maximum number of proteins that can be predicted by our model. We obtain sequence-derived embeddings from a protein language model (PLM) for each gene $\mathbf{E}^{\text{rna}} \in \mathbb{R}^{G \times d_{\text{esm}}}$ and protein $\mathbf{E}^{\text{prot}} \in \mathbb{R}^{P \times d_{\text{esm}}}$. Our goal is to learn a transcriptomic expression, sequence, and metadata conditioned protein generative model:

$$p(\mathbf{p} \mid \mathbf{r}, \mathbf{c}, \mathbf{d}, \mathbf{E}^{\text{rna}}, \mathbf{E}^{\text{prot}}). \tag{1}$$

Details on the dataset and cell metadata embeddings are in A.2.1

2.2 SEQUENCE-AWARE TOKENIZATION

Different genes or proteins with identical expression levels have distinct regulatory or functional roles depending on their underlying molecular properties. Prior work has shown that such properties are encoded in sequence representations derived from protein and RNA language models (Rives et al., 2021; Lin et al., 2023; Yin et al., 2025; Wang et al., 2024c). Instead of sampling sequence tokens based on expression (Rosen et al., 2023) or treating sequence and expression as independent components of a gene token (Kalfon et al., 2025), RNAProFlo uses protein sequence representations to modulate RNA and protein expression during tokenization. This allows genes or proteins with similar expression to effect generated protein levels differently, based on learned sequence dependent roles. It enables RnaProFlo to predict expression levels for new proteins, for example in new species or previously unmeasured in the training data, conditioned on genes that may not be expressed in training data, thus generalizing to new datasets. ESM2 (Lin et al., 2023) derived sequence embeddings are used with further details in A.2.2.

2.3 METADATA CONDITIONING WITH ADAPTIVE LAYER NORMALIZATION

We leverage metadata contextual cues for protein generation by globally conditioning transformer layers with adaptive layer normalization (AdaLN-Zero) (Peebles & Xie, 2023). For cell i, we concatenate cell and dataset embeddings $\mathbf{c}_i, \mathbf{d}_i \in \mathbb{R}^{d_{\text{meta}}}$ into $\mathbf{m}_i = [\mathbf{c}_i; \mathbf{d}_i]$, project them onto the token embedding space $\mathbb{R}^{d_t}$ with $\mathbf{W}_\mathbf{m} \in \mathbb{R}^{d_t \times 2d_{\text{meta}}}$, $\mathbf{b}_m \in \mathbb{R}^{d_t}$ and apply layer normalization:

$$\mathbf{z}_i^{\text{meta}} = \text{LayerNorm}(\mathbf{W}_m \mathbf{m}_i + \mathbf{b}_m), \tag{2}$$

The metadata representation is then fused with the sinusoidal timestep embedding $\mathbf{e}_t = \text{MLP}(\text{Sinusoidal}(t))$, which indicates the current position along the denoising trajectory of the flow matching transformer, to form a global conditioning representation:

$$\mathbf{g_i}_t = \phi([\mathbf{e}_t; \mathbf{z}_i^{\text{meta}}]) \in \mathbb{R}^{d_t}, \tag{3}$$

where $\phi(\cdot)$ is an MLP with a linear layer and SiLU activation. Further details on $\mathbf{e}_t$ and ϕ are in A.2.3. $\mathbf{g}_{i_t}$ is used throughout the transformer backbone to generate feature-wise affine modulation parameters.

Modulation is applied before the self-attention (parameters $\boldsymbol{\gamma}_s, \boldsymbol{\beta}_s$), cross-attention ($\boldsymbol{\gamma}_c, \boldsymbol{\beta}_c$) and final MLP layer ($\boldsymbol{\gamma}_{\text{mlp}}, \boldsymbol{\beta}_{\text{mlp}}$) in each transformer block allowing metadata and diffusion time to independently regulate the different computational components of the denoising process. For cell i:

$$[\boldsymbol{\gamma}_{i,s}, \boldsymbol{\beta}_{i,s}, \boldsymbol{\gamma}_{i,c}, \boldsymbol{\beta}_{i,c}, \boldsymbol{\gamma}_{i,\text{mlp}}, \boldsymbol{\beta}_{i,\text{mlp}}] = \mathbf{W}^{\text{AdaLN}} \mathbf{g}_{i_t} + \mathbf{b}^{\text{AdaLN}} \tag{4}$$

where $\mathbf{W}^{\text{AdaLN}} \in \mathbb{R}^{6d_t \times d_t}$ and $\mathbf{b}^{\text{AdaLN}} \in \mathbb{R}^{6d_t}$. The modulation weights $\mathbf{W}^{\text{AdaLN}}, \mathbf{b}^{\text{AdaLN}}$ are initialized to 0 improving optimization stability during early training. For a layer normalized hidden representation $\mathbf{h}_{i,l}$ of cell i in layer $l = \{s, c, \text{mlp}\}$, AdaLN applies

$$\text{AdaLN}(\mathbf{h}_{i,l}, \boldsymbol{\gamma}_{i,l}, \boldsymbol{\beta}_{i,l}) = \mathbf{h}_{i,l} \odot (1 + \boldsymbol{\gamma}_{i,l}) + \boldsymbol{\beta}_{i,l}, \tag{5}$$

Dataset embeddings capture broad cohort-specific effects, including assay and tissue variation, while cell embeddings encode finer-grained biological context such as cellular state. Injecting them with AdaLN allows the model to dynamically adapt its latent denoising trajectory according to experimental context and cell identity without incurring the quadratic cost associated with attention tokens.

2.4 FLOW MATCHING FORMULATION

Motivated by the success of text-conditioned image generation (Esser et al., 2024) we model the conditional distribution of protein tokens using a continuous-time generative framework based on rectified flow matching. Let $\mathbf{z}^{\text{prot}} \in \mathbb{R}^{P \times d_t}$ denote clean protein tokens and $\mathbf{z}_0 \sim \mathcal{N}(0, I)$ denote Gaussian noise. We can define linear interpolation:

$$\mathbf{z}_t = (1 - t)\mathbf{z}_0 + t\mathbf{z}^{\text{prot}}, \quad t \sim \mathcal{U}(0, 1). \tag{6}$$

The corresponding target velocity field is constant along this path:

$$\mathbf{v}^*(\mathbf{z}_t) = \frac{d\mathbf{z}_t}{dt} = \mathbf{z}^{\text{prot}} - \mathbf{z}_0. \tag{7}$$

The model learns a parameterized velocity field $\mathbf{v}_\theta(\mathbf{z}_t, t, \cdot)$ by minimizing:

$$\mathcal{L}_{\text{flow}} = \mathbb{E}_{t,\mathbf{z}_0}\left[\left\|\mathbf{v}_\theta(\mathbf{z}_t, t, \cdot) - (\mathbf{z}^{\text{prot}} - \mathbf{z}_0)\right\|^2\right]. \tag{8}$$

Clean protein tokens $\hat{\mathbf{z}}^{\text{prot}}$ are obtained from sampled noise by Euler integration of the learned velocity field starting from $\mathbf{z}_0 \sim \mathcal{N}(0, I)$ and integrating to $t = 1$:

$$\mathbf{z}_{t+\Delta t} = \mathbf{z}_t + \Delta t\, \mathbf{v}_\theta(\mathbf{z}_t, t), \tag{9}$$

We parameterize $\mathbf{v}_\theta$ using a Transformer. Cross and self attention ensure that protein tokens attend to RNA and protein tokens respectively when learning the denoising velocity.

2.5 Decoder

A 2-layer MLP decoder recovers protein expression $\hat{p}$ from clean tokens $\hat{\mathbf{z}}^{\text{prot}}$. With the encoder protein ESM2 embedding $\mathbf{h}_p^{\text{esm}}$ from 17 we learn sequence-based expression modulation parameters $\boldsymbol{\gamma}_{\text{esm}-\text{dec}}, \boldsymbol{\beta}_{\text{esm}-\text{dec}}$ to recover protein expression with a 2-layer MLP A.4:

$$\hat{\mathbf{p}} = \text{Softplus}(\text{MLP}(\boldsymbol{\gamma}_{\text{esm}-\text{dec}} \odot \hat{\mathbf{z}}^{\text{prot}} + \boldsymbol{\beta}_{\text{esm}-\text{dec}})). \tag{10}$$

2.6 Staged Training Objective

RNAProFlo is trained in three stages since it depends on the combination of informative latents and an accurate transport model. First, we train the RNA and protein encoder and protein decoder with a reconstruction and contrastive loss to preserve measured protein expression and align RNA and protein latents. Second, we freeze the encoders and train the RNA- and metadata-conditioned flow model in the latent space with the flow matching loss, avoiding simultaneous drift in the representation and velocity field. Finally, we apply low rank adaptation (LoRA) based joint training of the transformer blocks and decoder with a composite loss, addressing errors that may be small in latent space but consequential after decoding.
Our **Stage 1** loss is:

$$\mathcal{L}_{\text{stage1}} = \lambda_{\text{recon}}\mathcal{L}_{\text{recon}} + \lambda_{\text{contrast}}\mathcal{L}_{\text{contrast}} \tag{11}$$

Contrastive loss aligns cell-level aggregates of RNA tokens for cell i $\tilde{\mathbf{z}}_i^{\text{rna}}$ and protein tokens for cell j $\tilde{\mathbf{z}}_j^{\text{prot}}$ in latent space A.6 and reconstruction loss measures mean square error (MSE) in decoded protein expression space. **Stage 2** learns protein denoising velocity in the latent space with the encoder fixed and a flow matching loss according to 8. **Stage 3** jointly adapts the flow matching transformer and decoder with LoRA A.6.3. With base weights frozen, adapters are added to self attention, cross attention, MLP and AdaLN layers in transformer blocks as well as the sequence-based expression modulator and MLP layer of the decoder. Metadata and timestep conditioning and the final transformer projection layer are fully updated without LoRA.

$$\mathcal{L}_{\text{stage3}} = \lambda_{\text{expr}}\mathcal{L}_{\text{expr}} + \lambda_{\text{contrast}}\mathcal{L}_{\text{contrast}} \tag{12}$$

To ensure protein profiles generated by Euler integration remain close to observed values we introduce $\mathcal{L}_{\text{expr}}$. For each cell, we generate K predictions from independent initial noise samples. These samples provide a Monte Carlo estimate of expected latent and endpoint accuracy. To adapt the Stage 3 pretrained model to specific target datasets, users can optionally perform additional fine-tuning with the stage 3 objective and LoRA adapters.

2.7 Gated Spatial Niche Cross Attention with fourier distance bias

We develop RNAProFloS, the spatial extension of RnaProFlo, to generate spatial location conditioned proteomic data:

$$p(\mathbf{p} \mid \mathbf{r}, \mathbf{c}, \mathbf{d}, \mathbf{E}^{\text{rna}}, \mathbf{E}^{\text{prot}}, \mathbf{s}). \tag{13}$$

where $\mathbf{s}_i \in \mathbb{R}^2$ denote the query cell's spatial coordinates. To incorporate local tissue context, we identify the K nearest spatial neighbors $\mathcal{N}_K(i)$ of each query cell i within the same tissue section. We encode the RNA expression of each neighbor $j \in \mathcal{N}_K(i)$ as a sequence of gene tokens

$\{\mathbf{r}_{i,j,g}\}_{g=1}^{G}$ using the RNA embedder. This preserves gene-level information from neighboring cells rather than pooling each neighbor into a single token. In each transformer block, the query cell's protein tokens attend to all gene tokens from its spatial neighbors. For protein token p, gene token g in neighbor j, and attention head h, the attention logit is

$$a_{i,p,j,g}^{(h)} = \frac{\left(\mathbf{q}_{i,p}^{(h)}\right)^{\top}\mathbf{k}_{i,j,g}^{(h)}}{\sqrt{d_h}} + \beta_h(d_{i,j}), \qquad d_{i,j} = \|\mathbf{s}_i - \mathbf{s}_j\|_2, \tag{14}$$

where $\mathbf{q}_{i,p}^{(h)}$ is projected from the query cell's protein token, $\mathbf{k}_{i,j,g}^{(h)}$ is projected from the neighboring gene token, and d_h is the dimension of each attention head. The distance bias is shared across the gene tokens of a given neighbor but varies across neighbors and attention heads. We map each scalar distance to Fourier features and use an MLP to obtain the head-specific bias:

$$\boldsymbol{\phi}(d) = \left[\sin(2^f\pi d), \cos(2^f\pi d)\right]_{f=0}^{F-1}, \quad \boldsymbol{\beta}(d) = \mathrm{MLP}(\boldsymbol{\phi}(\mathrm{d})) \tag{15}$$

Attention weights are normalized jointly over all gene tokens in all K neighbors. Consequently, each protein token can select specific gene-level signals from neighboring cells while conditioning attention on their spatial distance. We insert this attention module between protein self-attention and cross-attention to the query cell's own RNA tokens. Its output is added through a learned scalar gate:

$$\widetilde{\mathbf{H}}_b = \mathbf{H}_b + \tanh(\eta)\, W_O\, \mathrm{Attn}_{\mathrm{niche}}(\mathbf{H}_b), \qquad \eta = 0 \text{ at initialization.} \tag{16}$$

The zero-initialized gate makes the new spatial branch an identity mapping at initialization, allowing its contribution to be learned during spatial fine-tuning.

3 Experimental Results

RNAProFlo is trained on 3.3 million cells from 38 datasets, 2 sequencing technologies, 2 species, 14 tissue types, 24 primary cell types and 16 disease states. We evaluate RnaProFlo on the similarity of generated protein profiles with true held-out profiles across all experimentally measured proteins on over 436,500 cells across 18 datasets encompassing various scenarios. Test set details including number of evaluated proteins are at Table 7. Training set information is at A.7 and Table 6 with a summary of the evaluations A.8. A detailed ablation study is included in 3.5. Unless otherwise mentioned, values reported are mean and standard deviation across cells and RNAProFlo metrics were computed from the average of 10 independently generated protein expression profiles per cell. Since this is a regression task, we also train **RnaProDisc** A.9, a discriminative mode of RnaProFlo where the transformer learns masked protein values with an MSE reconstruction loss instead of learning flow matching velocity. RnaProDisc applies a similar 3-stage training strategy as RnaProFlo with scRNA-seq, sequence, metadata and spatial location conditioning, while excluding the timestep embedding. Both RnaProFlo and RnaProDisc outperform benchmarked methods in zero-shot and fine-tuned mode. They handle proteins, species, technologies and perturbations not encountered during training. They also estimate spot based proteomic data conditioned on spatial location.The deterministic RnaProDisc and generative RnaProFlo perform similarly on evaluated tasks. We therefore recommend the use of RnaProDisc for deterministic tasks and RnaProFlo for generative tasks like dataset simulation.

3.1 Zero-shot Single Cell Protein Generation

RnaProFlo and the discriminative variant RnaProDisc were evaluated on 4 datasets (Table 7), 3 of which were unseen during training. Dataset103_1 was split $90-10\%$ between training and testing. Data spanned 4 human tissues, 2 sequencing technologies, 61 to 228 proteins and 271,825 cells in all. All protein levels for all cells were held-out for testing. While almost all methods outperform the baseline (Table 1), RNAProDisc followed by RnaProFlo achieve top performance on all datasets.

3.2 Fine-tuned Single Cell Protein Generation

We evaluate fine-tuning performance using 10% of data for finetuning with testing on the remaining 90%. Fine-tuning RNAProFlo (Ours-FT) improves the performance over the zero-shot ver-

Table 1: Zero-shot performance of RNAProFlo (Ours), the discriminative mode RNAProDisc (Ours-D) and other methods. Values report mean (standard deviation). Best in **bold**, second-best underlined. N and P denote number of test cells and proteins respectively. BMMC:Bone marrow mononuclear cells

	Dataset103_1 (CITE-seq) (Blood $N = 161764, P = 228$)			Dataset108 (ECCITE-seq) (Blood/skin $N = 37819, P = 61$)			Dataset78_1.3 (CITE-seq) (Lung $N = 1254, P = 81$)			GSE194122 (CITE-seq) (BMMC $N = 70988, P = 140$)		
Method	Cos (↑)	MSE (↓)	Cos(P) (↑)	Cos (↑)	MSE (↓)	Cos(P) (↑)	Cos (↑)	MSE (↓)	Cos(P) (↑)	Cos (↑)	MSE (↓)	Cos(P) (↑)
Ours	0.913 (0.029)	0.184 (0.069)	0.892 (0.104)	**0.786** (0.040)	**0.414** (0.105)	**0.802** (0.152)	**0.875** (0.025)	**0.240** (0.045)	**0.905** (0.091)	0.564 (0.073)	0.730 (0.261)	0.530 (0.215)
Ours-D	**0.961** (0.013)	**0.082** (0.029)	**0.917** (0.098)	0.683 (0.080)	0.546 (0.101)	0.704 (0.179)	0.842 (0.031)	0.297 (0.053)	0.890 (0.095)	0.541 (0.083)	**0.792** (0.389)	**0.531** (0.216)
Baseline	0.200 (0.092)	0.987 (0.016)	0.317 (0.224)	0.252 (0.146)	0.977 (0.324)	0.168 (0.130)	0.153 (0.081)	0.993 (0.049)	0.289 (0.187)	0.589 (0.079)	1.208 (0.429)	0.340 (0.297)
CAPTAIN	0.539 (0.128)	0.708 (0.136)	0.495 (0.252)	0.304 (0.202)	1.175 (0.439)	0.241 (0.241)	0.189 (0.191)	1.265 (0.319)	0.234 (0.134)	0.076 (0.128)	1.234 (0.147)	0.073 (0.127)
scTranslator	0.556 (0.106)	0.796 (0.283)	0.398 (0.397)	0.370 (0.128)	0.900 (0.150)	0.480 (0.245)	0.267 (0.075)	1.255 (0.189)	0.384 (0.357)	**0.624** (0.058)	4.229 (1.927)	0.185 (0.741)
totalVI	0.695 (0.148)	1.611 (0.781)	0.657 (0.176)	0.489 (0.173)	3.873 (0.888)	0.335 (0.176)	0.224 (0.067)	1.064 (0.588)	0.530 (0.149)	0.484 (0.096)	0.835 (0.194)	0.526 (0.158)
scProTrans	0.564 (0.290)	0.791 (0.259)	0.537 (0.050)	0.649 (0.255)	0.762 (0.239)	0.640 (0.120)	0.776 (0.233)	0.668 (0.190)	0.753 (0.085)	0.580 (0.153)	0.801 (0.160)	0.505 (0.096)

Table 2: Performance of finetuned RNAProFlo (Ours-FT), finetuned RnaProDisc (Ours-D-FT) and other finetuned methods on the same datasets from Table 1 with 10% of all cells used for finetuning and the remaining for testing. M, N denotes number of finetune and test cells for each dataset. RnaProFlo (Ours-ZS) and RnaProDisc (Ours-D-ZS) in zero-shot mode similar to 1 are included on the same data points here for comparison.

	Dataset103_1 ($M = 1630, N = 14547$)			Dataset108 ($M = 3147, N = 28586$)			Dataset78_1.3 ($M = 135, N = 1101$)			GSE194122 ($M = 21806, N = 63889$)		
Method	Cos (↑)	MSE (↓)	Cos(P) (↑)	Cos (↑)	MSE (↓)	Cos(P) (↑)	Cos (↑)	MSE (↓)	Cos(P) (↑)	Cos (↑)	MSE (↓)	Cos(P) (↑)
Ours-FT	**0.964** (0.012)	**0.074** (0.027)	0.920 (0.098)	0.931 (0.027)	0.135 (0.062)	0.898 (0.158)	0.961 (0.016)	0.082 (0.032)	0.910 (0.092)	0.894 (0.067)	0.288 (0.660)	0.598 (0.233)
Ours-D-FT	**0.964** (0.012)	**0.074** (0.026)	**0.921** (0.097)	**0.932** (0.029)	**0.134** (0.063)	0.900 (0.160)	**0.962** (0.015)	**0.078** (0.031)	**0.912** (0.090)	0.888 (0.070)	0.294 (0.632)	0.596 (0.233)
Ours-ZS	0.913 (0.029)	0.184 (0.069)	0.892 (0.104)	0.786 (0.040)	0.414 (0.105)	0.802 (0.152)	0.875 (0.025)	0.240 (0.046)	0.906 (0.091)	0.564 (0.073)	0.729 (0.258)	0.530 (0.215)
Ours-D-ZS	0.961 (0.013)	0.082 (0.029)	0.917 (0.098)	0.683 (0.080)	0.546 (0.101)	0.704 (0.179)	0.842 (0.031)	0.297 (0.053)	0.891 (0.095)	0.541 (0.083)	0.791 (0.385)	0.531 (0.216)
CAPTAIN	0.832 (0.069)	0.319 (0.113)	0.792 (0.178)	0.883 (0.124)	0.223 (0.195)	**0.958** (0.079)	0.836 (0.082)	0.332 (0.140)	0.559 (0.422)	0.345 (0.156)	0.874 (0.111)	0.303 (0.203)
scTranslator	0.910 (0.073)	0.209 (0.190)	0.629 (0.308)	0.892 (0.103)	0.248 (0.235)	0.143 (0.494)	0.912 (0.103)	0.199 (0.257)	0.155 (0.484)	**0.937** (0.029)	**0.146** (0.080)	**0.834** (0.078)
totalVI	0.281 (0.177)	2.374 (12.043)	0.486 (0.175)	0.655 (0.182)	4.893 (121.049)	0.390 (0.212)	0.324 (0.295)	1.829 (14.526)	0.249 (0.212)	0.356 (0.126)	0.900 (0.372)	0.340 (0.185)

sion (Ours) with a similar trend for RnaProDisc (Ours-D-FT and Ours-D). Ours-FT and Ours-D-FT show almost identical results suggesting that the direct transformer and flow matching transformer learn almost the function on this predictive task with fine-tuning. Fine-tuned RNAProFlo and RNAProDisc excel compared to benchmarked methods. We demonstrate the capability of RnaProFlo on limited fine-tuning data in Table 10.

3.3 Generalization to new species, technologies, and perturbation assays

We test the ability of RnaProFlo to generalize to new species, sequencing technologies, proteins, and perturbation conditions unseen during training. 2 Reap-Seq datasets demonstrate performance on a new technology unseen during training while Perturb-CITE-seq data tests generalization to unseen perturbations. We test cross species generalization with Wild boar and Rhesus Monkey data comprising 198 and 13 proteins respectively. While these proteins are not captured during training their human/ mouse orthologs with identical names are present allowing other methods to make predictions. Finally, we test generalization to completely unseen proteins on new sequencing technology. Neat-seq dataset is an intracellular protein sequencing technology capturing 7 new proteins.
In zero-shot mode, RNAProFlo surpasses others, likely due to sequence-modulation and metadata providing additional context for unseen proteins. Next, we test generalization to completely unseen proteins on a new technology. CAPTAIN and totalVI are unable to make predictions while

Table 3: Performance comparison of zero-shot RNAProFlo (Ours-ZS), RnaProDisc (Ours-D-ZS), finetuned RnaProFlo (Ours-FT) and RnaProDisc (Ours-D-FT) and other methods across datasets in zero-shot and fine-tuned settings. Values report mean (standard deviation). Best values are in bold and second-best values are underlined. The out of training distribution feature being tested is listed below each dataset. Neat-Seq is a new technology dataset containing $p = 7$ proteins unseen during training. No fraction of these datasets was seen during training. Dataset sizes are in 9

		Ours-FT	Ours-D-FT	Ours-ZS	Ours-D-ZS	Baseline	CAPTAIN-FT	CAPTAIN-ZS	scTranslator-FT	scTranslator-ZS	totalVI-ZS	totalVI-FT
GSE100501_1 Reap-Seq Tech	Cos	**0.883** (0.077)	0.875 (0.083)	0.634 (0.112)	0.630 (0.061)	0.241 (0.229)	0.702 (0.152)	0.301 (0.221)	0.849 (0.163)	0.470 (0.077)	0.665 (0.196)	0.709 (0.225)
	MSE	**0.224** (0.134)	0.237 (0.148)	0.849 (0.366)	0.621 (0.083)	0.989 (0.053)	0.507 (0.213)	0.928 (0.205)	0.303 (0.279)	2.996 (1.512)	1.539 (6.769)	38.741 (307.772)
GSE100501_2 Reap-Seq Tech	Cos	0.954 (0.029)	0.955 (0.024)	0.706 (0.065)	0.683 (0.035)	0.274 (0.116)	0.629 (0.153)	0.079 (0.198)	**0.968** (0.064)	0.548 (0.042)	0.801 (0.110)	0.939 (0.147)
	MSE	0.090 (0.054)	0.089 (0.045)	0.639 (0.133)	0.841 (0.124)	0.985 (0.054)	0.596 (0.194)	1.123 (0.183)	**0.071** (0.113)	2.678 (0.527)	0.479 (0.136)	1.429 (24.471)
Dataset 86 Perturb-CS Perturbation	Cos	0.963 (0.022)	0.963 (0.021)	0.814 (0.058)	0.843 (0.042)	0.750 (0.106)	0.693 (0.099)	0.736 (0.090)	**0.977** (0.018)	0.752 (0.038)	0.324 (0.107)	0.442 (0.045)
	MSE	0.073 (0.042)	0.073 (0.039)	0.454 (0.071)	0.504 (0.057)	1.015 (0.712)	0.518 (0.142)	0.624 (0.326)	**0.051** (0.041)	0.879 (0.213)	1.750 (0.374)	0.829 (0.058)
Dataset76_5 Boar Species	Cos	0.826 (0.054)	**0.830** (0.053)	0.636 (0.056)	0.633 (0.059)	0.140 (0.102)	0.406 (0.129)	0.021 (0.094)	0.758 (0.136)	0.599 (0.063)	0.264 (0.093)	0.519 (0.199)
	MSE	0.320 (0.106)	**0.313** (0.103)	0.978 (0.338)	0.691 (0.044)	3.919 (13.339)	0.836 (0.141)	1.307 (0.225)	0.545 (0.549)	2.382 (2.056)	1.751 (1.103)	0.769 (0.170)
Dataset85 Monkey Species	Cos	0.731 (0.165)	0.813 (0.160)	0.642 (0.126)	0.682 (0.147)	0.490 (0.288)	**0.906** (0.068)	0.301 (0.358)	0.773 (0.183)	0.482 (0.188)	0.610 (0.269)	0.326 (0.288)
	MSE	0.477 (0.281)	0.339 (0.255)	0.797 (0.044)	0.778 (0.058)	16.289 (116.363)	**0.176** (0.114)	1.016 (0.423)	0.441 (0.342)	1.391 (0.739)	0.720 (0.553)	1.138 (1.637)
GSE178707 Neat-Seq Tech+Protein	Cos	**0.968** (0.015)	**0.968** (0.015)	0.824 (0.044)	0.795 (0.049)	0.269 (0.266)	- (-)	- (-)	0.679 (0.290)	0.364 (0.158)	- (-)	- (-)
	MSE	**0.064** (0.029)	0.065 (0.029)	0.500 (0.034)	0.406 (0.055)	0.994 (0.008)	- (-)	- (-)	0.653 (0.624)	1.472 (0.426)	- (-)	- (-)

Table 4: Performance comparison of RnaProFloS (Ours-Sp), the spatial variant of RnaProFlo-D (Ours-D-Sp), zero-shot RnaProFlo (Ours-ZS) and RnaProFlo-D (Ours-D-ZS) and other methods across spot-based spatial CITE-seq **human** datasets. For each dataset, finetuning was performed on 10% of all cells across all available proteins.

		Ours-Sp	Ours-D-Sp	Ours-ZS	Ours-D-ZS	Baseline	CAPTAIN FT	CAPTAIN	scTranslator FT	scTranslator	totalVI FT	totalVI
GSE213264_5 (Tonsil)	Cos (↑)	**0.968** (0.013)	0.967 (0.012)	0.765 (0.021)	0.610 (0.021)	0.370 (0.083)	0.794 (0.066)	-0.007 (0.111)	0.950 (0.049)	0.739 (0.023)	0.513 (0.320)	0.297 (0.044)
	MSE (↓)	**0.067** (0.029)	0.069 (0.021)	0.450 (0.045)	0.846 (0.065)	0.966 (0.017)	0.371 (0.104)	1.115 (0.099)	0.105 (0.093)	1.163 (0.155)	0.817 (0.340)	3.627 (1.812)
GSE213264_6 (Spleen)	Cos (↑)	0.936 (0.029)	0.930 (0.027)	0.638 (0.036)	0.547 (0.032)	0.177 (0.198)	0.589 (0.058)	0.022 (0.114)	**0.960** (0.035)	0.659 (0.042)	0.776 (0.241)	0.249 (0.050)
	MSE (↓)	0.130 (0.060)	0.137 (0.052)	0.609 (0.053)	0.721 (0.042)	0.995 (0.007)	0.677 (0.087)	1.121 (0.097)	**0.082** (0.068)	2.521 (0.382)	0.707 (0.233)	17.036 (12.866)
GSE213264_7 (Thymus)	Cos (↑)	**0.973** (0.012)	0.971 (0.011)	0.726 (0.016)	0.628 (0.014)	0.390 (0.114)	0.841 (0.059)	0.023 (0.104)	0.956 (0.024)	0.732 (0.021)	0.736 (0.259)	0.292 (0.051)
	MSE (↓)	**0.056** (0.026)	0.059 (0.020)	0.476 (0.026)	0.641 (0.026)	0.976 (0.013)	0.314 (0.110)	1.132 (0.106)	0.093 (0.058)	1.372 (0.144)	0.743 (0.267)	4.168 (2.283)
GSE213264_8 (Skin)	Cos (↑)	**0.878** (0.059)	0.858 (0.056)	0.608 (0.066)	0.444 (0.051)	0.127 (0.143)	0.614 (0.090)	0.067 (0.188)	0.774 (0.120)	0.552 (0.072)	0.538 (0.243)	0.232 (0.105)
	MSE (↓)	**0.246** (0.118)	0.271 (0.095)	0.635 (0.075)	0.819 (0.056)	0.991 (0.018)	0.624 (0.117)	1.278 (0.344)	0.432 (0.211)	3.778 (1.749)	0.941 (0.491)	29.338 (36.150)
GSE213264_9 (Glioblastoma)	Cos (↑)	**0.956** (0.021)	0.940 (0.013)	0.635 (0.020)	0.558 (0.021)	0.243 (0.148)	0.871 (0.038)	0.024 (0.104)	0.936 (0.046)	0.686 (0.024)	0.514 (0.251)	0.327 (0.049)
	MSE (↓)	**0.087** (0.039)	0.119 (0.023)	0.600 (0.036)	0.689 (0.023)	0.983 (0.016)	0.246 (0.063)	1.261 (0.233)	0.197 (0.214)	1.447 (0.286)	0.971 (1.308)	9.973 (7.864)

scTranslator, with its protein identifier lookup, predicts poorly compared to RnaProFlo. On perturbation data, RnaProFlo shows performance improvements compared to its zero-shot version and other fine-tuned methods even when fine-tuned on just 224 cells (0.1% setting) (Table 10).

3.4 RNAPROFLO FOR SPOT-BASED SPATIAL OMICS

Spatial-proteomics measures protein levels at specific locations in tissue slices. We evaluate whether RnaProFloS, the spatial extension of RnaProFlo, can effectively leverage spatial tissue structure to improve protein generation. A primary challenge is the lack of single cell resolution RNA-protein paired spatial data. Paired spatial CITE-seq (Liu et al., 2023) captures groups of cells or spots in tissue slices alongwith 2D spatial coordinates in the tissue. We obtain 4 mouse and 5 human datasets across 9 tissues totaling 16,334 cells and 189 to 273 proteins. RnaProFlo and RnaProDisc are fine-tuned with gated spatial niche cross attention and fourier distance bias 2.7 on 10% of all data and evaluated on the rest. RnaProFloS and RnaProFlo-D-S, the discriminative mode, outperform benchmarked methods on all but 1 dataset across all metrics. We further evaluate finetuning performance in the limited data scenario on 1% of all data (Table 11).

Table 5: Ablation results on validation set $N = 32,007$ cells grouped by ablation category

	Original	Sequence		Tokenizer		Metadata					Loss			Stage
Metric	Orig	ESM(P)	ESM(R)	FiLM(P)	FiLM(R)	Cell	Data	Cell&Data	Abstract	Celltype	Cont	Expr	1-step	Stage2 only
MSE (↓)	**0.352**	1.335	0.379	1.334	0.359	0.429	0.353	0.432	0.356	0.368	0.366	3.395	2.187	3.499
Cos (↑)	**0.939**	0.777	0.933	0.777	0.937	0.927	0.937	0.925	0.937	0.939	0.936	0.649	0.760	0.653

3.5 Ablations

After establishing the performance of RnaProFlo, RnaProDisc and RnaProFloS in zero-shot and fine-tuned modes across perturbation, sequencing technology, species, and proteins unseen during training, we ask: (i) Is flow matching learning to generate protein data? UMAP visualizations of generated embeddings and decoded proteins shows meaningful grouping by celltype and pathway 6. Latent velocity cosine and seed similarity for the generative model is also high (Table 12). Additionally, we interpret RnaProFlo self-attention showing that attention focuses on known protein-protein interaction pairs (Table 13). We therefore conclude that the learned flows are meaningful. (ii) Are the sequence and metadata priors critical? We ablate the ESM2 embeddings, sequence-modulation of expression, and various metadata components (Table 5). While all priors improve performance, protein sequence modulated tokenization components are key. (iii) Are the design choices improving performance? From Table 5 we observe that staged training, expression loss and the Euler rollout through the flow model while computing expression loss are critical.

Further ablations and associated details are in A.12.2.

4 Conclusion and Future Work

RnaProFlo is the first conditional generative multi-modal foundation model for the single cell and spatial proteomics generation. It is the also the first flow matching based scFM. We leverage the diffusion transformer architecture (Peebles & Xie, 2023) to train a foundation model that learns the flow matching velocity with RNA-protein crossmodal attention and protein-protein self attention. Our protein-protein attention layer focuses on known protein-protein interaction pairs. Our tokenizer modulates RNA and protein expression with ESM2 derived sequence embeddings incorporating underlying structural and functional biology (Baranowski et al., 2025) in ESM2 (Lin et al., 2023). We further encode cell and dataset metadata by designing large language model prompts and generating embeddings with Qwen3-Embedding-8B (Zhang et al., 2025) which we globally inject into our transformer layers via adaptive layer normalization. We extensively evaluate our method against a baseline and multiple state of the art methods in zero-shot and fine tuned settings. We empirically demonstrate excellent out of distribution performance on new species, sequencing technologies, proteins and perturbations. RnaProDisc, he discriminative variant of RnaProFlo, achieves competitive performance as well and can be used for lightweight prediction tasks when generative stochasticity is not needed. We extend RnaProFlo to spatial data by introducing a gated spatial niche token level cross attention layer, enabling spatially conditioned protein generation.

Despite its excellent performance, RnaProFlo has limitations. Like other flow-matching transformer models, its interpretability is limited, which would be useful in the context of understanding the underlying biological process of protein translation. Extending RnaProFlo to spatial proteomics imaging data is also the subject of future work.

Overall, RnaProFlo, with its ability to generate protein levels conditioned on scRNA-seq or ST expression along with sequence, metadata and spatial location, enables proteomics data simulation in varied biological conditions paving the way for accelerating drug target discovery and disease understanding with ubiquitous societal implications for biomedical research.

AI use statement

In this work, we used generative AI tools for polishing the manuscript draft. LLM embeddings were also used as components of the model. We have not used generative AI tools to generate synthetic data sets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, assist with translation, clean and reformat dataset, support qualitative

and thematic data analysis, or interpret results and the rest of the required disclosure tasks are not applicable to this work. We have reviewed all AI-assisted work. We thoroughly read through and revised manuscript edits suggested by AI. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

Ethics statement

All datasets are publicly available as reported in their original studies.

Reproducibility statement

All our training and testing data, code and model weights will be publicly available at https://anonymous.4open.science/r/RNAProFlo-ADC3. The manuscript uses publicly available data. Details of the publicly available datasets used in the study, preprocessing and modeling steps are provided in the Appendix.

References

Uniprot: the universal protein knowledgebase in 2025. *Nucleic acids research*, 53(D1):D609–D617, 2025.

Seungbyn Baek, Kyungwoo Song, and Insuk Lee. Single-cell foundation models: bringing artificial intelligence into cell biology. *Experimental & Molecular Medicine*, 57(10):2169–2181, 2025.

Catherine Baranowski, Hector Garcia Martin, Diego A Oyarzún, Aviv Spinner, Bijoy Desai, Christopher J Petzold, Evangelos-Marios Nikolados, Sebastian Jaaks-Kraatz, Aljaž Gaber, Robert J Chalkley, et al. Can protein expression be 'solved'? *Trends in biotechnology*, 2025.

Haiyang Bian, Yixin Chen, Xiaomin Dong, Chen Li, Minsheng Hao, Sijie Chen, Jinyi Hu, Maosong Sun, Lei Wei, and Xuegong Zhang. scmulan: a multitask generative pre-trained language model for single-cell analysis. In *International Conference on Research in Computational Molecular Biology*, pp. 479–482. Springer, 2024a.

Haiyang Bian, Yixin Chen, Erpai Luo, Xinze Wu, Minsheng Hao, Lei Wei, and Xuegong Zhang. General-purpose pre-trained large cellular models for single-cell transcriptomics. *National Science Review*, 11(11):nwae340, 2024b.

Amy F Chen, Benjamin Parks, Arwa S Kathiria, Benjamin Ober-Reynolds, Jorg J Goronzy, and William J Greenleaf. Neat-seq: simultaneous profiling of intra-nuclear proteins, chromatin accessibility and gene expression in single cells. *Nature Methods*, 19(5):547–553, 2022.

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.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Adam Gayoso, Zoë Steier, Romain Lopez, Jeffrey Regier, Kristopher L Nazor, Aaron Streets, and Nir Yosef. Joint probabilistic modeling of single-cell multi-omic data with totalvi. *Nature methods*, 18(3):272–282, 2021.

Daniel Hanhart, Federico Gossi, Maria Anna Rapsomaniki, Marianna Kruithof-de Julio, and Panagiotis Chouvardas. Sclinear predicts protein abundance at single-cell resolution. *Communications biology*, 7(1):267, 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.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Yinlei Hu, Siyuan Wan, Yuanhanyu Luo, Yuanzhe Li, Tong Wu, Wentao Deng, Chen Jiang, Shan Jiang, Yueping Zhang, Nianping Liu, et al. Benchmarking algorithms for single-cell multi-omics prediction and integration. *Nature Methods*, 21(11):2182–2194, 2024.

Boya Ji, Tingting Hu, Jiawen Wang, Mengmeng Liu, Liwen Xu, Qinhao Zhang, Siyun Zhong, Libo Qiao, Yan Zhang, Shaoliang Peng, et al. Captain: A multimodal foundation model pretrained on co-assayed single-cell rna and protein. *bioRxiv*, pp. 2025–07, 2025.

Shiyu Jiang, Zhaoyu Fang, Yujie Zhang, Xuting Zhang, Jérémie Kalfon, Weixu Wang, Xi Fu, Aakash Patel, Syed Rizvi, Yuancheng Ryan Lu, Siyu He, Yixin Wang, Kejun Ying, Peter Pao-Huang, Yifan Lu, Weize Xu, Mengchen Wang, Ziyang Miao, Jianhui Lin, Jimmy Ding, Zitong Jerry Wang, Wei Ouyang, Tianlong Chen, Guoxian Yu, Min Li, Jiayi Ma, Fei Wang, Yuying Xie, Jiliang Tang, Raul Rabadan, David van Dijk, Pengtao Xie, Peng He, Emily B. Fox, Le Song, Fabian J. Theis, Eric Xing, Christina V. Theodoris, Xiaojie Qiu, and Jiayuan Ding. The landscape of single-cell foundation models: Design principles, applications, and open challenges. *Preprints*, August 2026. doi: 10.20944/preprints202608.1166.v1. URL https://doi.org/10.20944/preprints202608.1166.v1.

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.

Justin Lakkis, Amelia Schroeder, Kenong Su, Michelle YY Lee, Alexander C Bashore, Muredach P Reilly, and Mingyao Li. A multi-use deep learning method for cite-seq and single-cell rna-seq data integration with cell surface protein prediction and imputation. *Nature machine intelligence*, 4(11):940–952, 2022.

Christopher Lance, Malte D Luecken, Daniel B Burkhardt, Robrecht Cannoodt, Pia Rautenstrauch, Anna Laddach, Aidyn Ubingazhibov, Zhi-Jie Cao, Kaiwen Deng, Sumeer Khan, et al. Multimodal single cell data integration challenge: results and lessons learned. *BioRxiv*, pp. 2022–04, 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.

Linjing Liu, Wei Li, Fang Wang, Yiming Li, Long-Kai Huang, Ka-Chun Wong, Fan Yang, and Jianhua Yao. A pre-trained large generative model for translating single-cell transcriptomes to proteomes. *Nature Biomedical Engineering*, pp. 1–20, 2025.

Tianyu Liu, Kexing Li, Yuge Wang, Hongyu Li, and Hongyu Zhao. Evaluating the utilities of foundation models in single-cell data analysis. *Advanced Science*, pp. e14490, 2024.

Tianyu Liu, Tianqi Chen, Wangjie Zheng, Xiao Luo, Yiqun Chen, and Hongyu Zhao. Embeddings from language models are good learners for single-cell data analysis. *Patterns*, 7(2), 2026.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Yang Liu, Marcello DiStasio, Graham Su, Hiromitsu Asashima, Archibald Enninful, Xiaoyu Qin, Yanxiang Deng, Jungmin Nam, Fu Gao, Pino Bordignon, et al. High-plex protein and whole transcriptome co-mapping at cellular resolution with spatial cite-seq. *Nature Biotechnology*, 41 (10):1405–1409, 2023.

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.

Mohammad Lotfollahi, Mohsen Naghipourfar, Malte D Luecken, Matin Khajavi, Maren Büttner, Marco Wagenstetter, Žiga Avsec, Adam Gayoso, Nir Yosef, Marta Interlandi, et al. Mapping single-cell data to reference atlases by transfer learning. *Nature biotechnology*, 40(1):121–130, 2022.

Hiren Madhu, João Felipe Rocha, Tinglin Huang, Siddharth Viswanath, Smita Krishnaswamy, and Rex Ying. Heist: a graph foundation model for spatial transcriptomics and proteomics data. In *International Conference on Learning Representations*, volume 2026, pp. 65453–65476, 2026.

Alex Morehead, Lazar Atanackovic, Akshata Hegde, Yanli Wang, Frimpong Boadu, Joel Selvaraj, Alexander Tong, Aditi Krishnapriyan, and Jianlin Cheng. Flow matching for generative modelling in bioinformatics and computational biology. *Nature Machine Intelligence*, pp. 1–18, 2026.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

Ethan Perez et al. Film: Visual reasoning with a general conditioning layer. *AAAI*, 2018.

Vanessa M Peterson, Kelvin Xi Zhang, Namit Kumar, Jerelyn Wong, Lixia Li, Douglas C Wilson, Renee Moore, Terrill K McClanahan, Svetlana Sadekova, and Joel A Klappenbach. Multiplexed quantification of proteins and transcripts in single cells. *Nature biotechnology*, 35(10):936–939, 2017.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Prajit Ramachandran, Barret Zoph, and Quoc Le. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2017.

Alexander Rives et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *PNAS*, 2021.

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.

Yanay Rosen, Maria Brbić, Yusuf Roohani, Kyle Swanson, Ziang Li, and Jure Leskovec. Toward universal cell embeddings: integrating single-cell rna-seq datasets across species with saturn. *Nature methods*, 21(8):1492–1500, 2024.

Damian Szklarczyk, Rebecca Kirsch, Mikaela Koutrouli, Katerina Nastou, Farrokh Mehryary, Radja Hachilif, Annika L Gable, Tao Fang, Nadezhda T Doncheva, Sampo Pyysalo, et al. The string database in 2023: protein–protein association networks and functional enrichment analyses for any sequenced genome of interest. *Nucleic acids research*, 51(D1):D638–D646, 2023.

Alejandro Tejada-Lapuerta, Anna C Schaar, Robert Gutgesell, Giovanni Palla, Lennard Halle, Mariia Minaeva, Larsen Vornholz, Leander Dony, Francesca Drummer, Till Richter, et al. Nicheformer: a foundation model for single-cell and spatial omics. *Nature Methods*, pp. 1–14, 2025.

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.

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *arXiv preprint arXiv:2302.00482*, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Christine Vogel and Edward M Marcotte. Insights into the regulation of protein abundance from proteomic and transcriptomic analyses. *Nature reviews genetics*, 13(4):227–232, 2012.

Chloe Wang, Haotian Cui, Andrew Zhang, Ronald Xie, Hani Goodarzi, and Bo Wang. scgpt-spatial: Continual pretraining of single-cell foundation model for spatial transcriptomics. *bioRxiv*, pp. 2025–02, 2025.

Fang Wang, Chunpu Liu, Jiawei Li, Fan Yang, Jiangning Song, Tianyi Zang, Jianhua Yao, and Guohua Wang. Spdb: a comprehensive resource and knowledgebase for proteomic data at the single-cell resolution. *Nucleic acids research*, 52(D1):D562–D571, 2024a.

Lifei Wang, Rui Nie, Xuexia Miao, Yankai Cai, Anqi Wang, Hanwen Zhang, Jiang Zhang, and Jun Cai. Inclust+: the deep generative framework with mask modules for multimodal data integration, imputation, and cross-modal generation. *BMC bioinformatics*, 25(1):41, 2024b.

Ning Wang, Jiang Bian, Yuchen Li, Xuhong Li, Shahid Mumtaz, Linghe Kong, and Haoyi Xiong. Multi-purpose rna language modelling with motif-aware pretraining and type-guided fine-tuning. *Nature Machine Intelligence*, 6(5):548–557, 2024c.

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.

Jing Xu, De-Shuang Huang, and Xiujun Zhang. scmformer integrates large-scale single-cell proteomics and transcriptomics data by multi-task transformer. *Advanced Science*, 11(19):2307835, 2024.

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.

Yongjian Yang, Yu-Te Lin, Guanxun Li, Yan Zhong, Qian Xu, and James J Cai. Interpretable modeling of time-resolved single-cell gene–protein expression with crossmodalnet. *Briefings in Bioinformatics*, 24(6):bbad342, 2023.

Weijie Yin, Zhaoyu Zhang, Shuo Zhang, Liang He, Ruiyang Zhang, Rui Jiang, Gan Liu, Jingyi Wang, Xuegong Zhang, Tao Qin, et al. Ernie-rna: an rna language model with structure-enhanced representations. *Nature Communications*, 16(1):10076, 2025.

Hanlei Yu, Yuanjie Zheng, and Xinbo Yang. scdm: A deep generative method for cell surface protein prediction with diffusion model. *Journal of Molecular Biology*, 436(12):168610, 2024.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, et al. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025.

Mengyuan Zhao, Jiawei Li, Yanlin Jiang, Jiahui Yan, Xinyue Tang, Cheng Liang, Jijun Tang, and Fei Guo. A sequence knowledge-guided deep learning method for single-cell multi-omics translation. *Genome Biology*, 2026.

Suyuan Zhao, Yizhen Luo, Ganbo Yang, Yan Zhong, Hao Zhou, and Zaiqing Nie. Stofm: a multi-scale foundation model for spatial transcriptomics. *arXiv preprint arXiv:2507.11588*, 2025.

Songqi Zhou, Yang Li, Wenyuan Wu, and Li Li. scmmt: a multi-use deep learning approach for cell annotation, protein prediction and embedding in single-cell rna-seq data. *Briefings in Bioinformatics*, 25(2):bbad523, 2024.

Xin Zou and Xiujuan Lei. Spatialpeft: a parameter-efficient fine-tuning framework for spatial transcriptomics foundation models. *Bioinformatics*, 42(7):btag503, 2026.

A Appendix

A.1 Related Work

We discuss related work in single cell protein level estimation, single cell foundation models (scFMs), Flow Matching and paired single cell and spatial RNA-protein technologies.

A.1.1 Methods for Protein Level Estimation

The top-performing method from NeurIPS 2021 (Lance et al., 2022) applied truncated SVD followed by kernel ridge regression (KRR) for multi-modal integration. Similarly, scLinear (Hanhart et al., 2024) utilizes multivariate regression for similar tasks. Multi-task sciPENN (Lakkis et al., 2022) tackles batch effects and missing data with feedforward networks, recurrent cells and censored loss for missing proteins. CrossmodalNet (Yang et al., 2023) employs Fader networks to disentangle time-specific embeddings from the shared embedding in temporal CITE-seq data using a Discriminator. scMMT (Zhou et al., 2024) uses several ResNet blocks, an embedding block and separate cell type prediction and protein expression blocks with GradNorm gradient loss to balance gradients during backpropagation across the two tasks. VAEs serve as generative models and have proven highly successful in single-cell omics, integrating data from multiple modalities into a shared latent space (Lotfollahi et al., 2022). TotalVI (Gayoso et al., 2021), the multiomic extension of scVI (Lopez et al., 2018), models the joint distribution of RNA and protein expression. It incorporates several components, including one-hot encoded batch vectors, a shared cell embedding for both RNA and protein data, an RNA-specific size factor to adjust for sequencing depth, and a protein-specific prior to account for background noise in protein measurements. Similarly, Inclust+ (Wang et al., 2024b) enhances VAE-based frameworks by adding additional embeddings to handle batch encoding and by using masking techniques to filter out missing or irrelevant values. The diffusion model scDM (Yu et al., 2024), encodes RNA and protein expression as text and leverages a pretrained denoising diffusion probabilistic model (DDPM) (Ho et al., 2020). They separately train a classifier on labeled RNA-protein data and inject the parameters into the transformer. However, their training data and experiments are limited.Recently, scProTrans (Zhao et al., 2026) used a sequence based gene and protein encoder, an scRNA-seq expression autoencoder for cell embeddings, and cross attention to estimate protein levels from RNA. The scTranslator (Liu et al., 2025) foundation model is pre-trained on paired bulk RNA-protein as well as paired single-cell RNA-protein datasets with optional fine-tuning. They use the Performer transformer to encode RNA with a reindexed gene positional encoding (GPE) to encode gene names and a transformer decoder to learn protein levels. They achieve state-of-the-art performance and are particularly well-suited for many single cell omics data integration and cell-type prediction tasks. The model is large and computationally expensive using 32 GPUs, each with 32 GB of memory, with a total training time of approximately 1 month (Liu et al., 2025). CAPTAIN (Ji et al., 2025) leverages the pretrained scFM scGPT (Cui et al., 2024) as an scRNA-seq encoder, augments it with a prior gene knowledge module capturing gene-gene relationships, and learns an RNA grounded protein embedding with a transformer.

A.1.2 Related work in scFMs

Single-cell foundation models (scFMs) (Cui et al., 2024; Theodoris et al., 2023; Hao et al., 2024; Rosen et al., 2023), inspired by language models, learn general-purpose representations of cells, genes, and biological states from large-scale single-cell datasets. Learned representations are then leveraged in zero-shot or fine-tuned mode for downstream tasks such as such as cell type annotation, batch correction, perturbation prediction, and gene network inference. They do not explicitly perform conditional generation or prediction of one modality from another. With a few exceptions in proteomics, (Liu et al., 2025; Ji et al., 2025), scFMs have been largely applied to scRNA-seq, scATAC-seq, and ST (Baek et al., 2025). Key limitations include the computational demands from training attention layers on thousands of genes and limited robustness to batch effects – distribution shifts in expression profiles due to technical discrepancies across datasets and experiments. Recent evaluations also suggest that while scFMs can perform well in some zero-shot and fine-tuned settings, their gains are task- and dataset-dependent, and conventional baselines may remain competitive for certain single-cell tasks (Liu et al., 2024). Key design choices in scFMs include tokenization, model architecture, training objectives and finetuning (Baek et al., 2025).

Tokenization in scFMs Tokenization in scFMs converts unordered, heterogeneous molecular measurements into structured sequences that can be processed by transformers. A common design treats each gene (protein or peak) as a token comprising (i) a gene identity embedding which maybe a learned lookup (Cui et al., 2024; Hao et al., 2024) or pretrained sequence-derived embedding (Rosen et al., 2023) (ii) an expression embedding either by discretization into bins (Cui et al., 2024) or by continuous projection through a small neural network (Hao et al., 2024) and (iii) optional metadata or context embeddings derived from one-hot covariates projected into continuous space (Cui et al., 2024). Because gene expression is inherently unordered, scFMs impose a sequence structure via strategies such as (i) fixed gene ordering using genomic order (Rosen et al., 2023) or a predefined vocabulary (Cui et al., 2024)) (ii) rank-based ordering by sorting genes by expression per cell (Theodoris et al., 2023) or (iii) omitting ordering (Yang et al., 2022) by treating cell tokens as an unordered set. Together, these design choices define how biological signals are factorized into tokens, directly impacting the model's ability to capture feature interactions, handle sparsity, and generalize across datasets.

Model architecture in scFMs Architectures in single-cell foundation models (scFMs) broadly follow three transformer-inspired paradigms. Decoder-only models treat gene tokens as a sequence and learn masked or autoregressive objectives over a fixed vocabulary, capturing gene–gene dependencies within a unified token space (Cui et al., 2024). Encoder–decoder architectures (Liu et al., 2025) leverage this paradigm in multimodal settings by encoding one modality and decoding another analogous to sequence-to-sequence transformers (Vaswani et al., 2017). Encoder-centric designs (Yang et al., 2022; Theodoris et al., 2023) with shallow decoders learn a shared latent representation of cellular state and address downstream tasks with task specific objectives.

Training objectives in scFMs Transformer-based models adopt masked modeling objectives, reconstructing held-out genes using cross-entropy for discretized expression (Cui et al., 2024) or mean squared error (MSE) for continuous values (Hao et al., 2024), enabling learning of feature dependencies from partially observed inputs. Multimodal models (Ji et al., 2025) incorporate contrastive losses like InfoNCE (Oord et al., 2018; Radford et al., 2021) to couple representations across modalities. Several models (Cui et al., 2024; Hao et al., 2024) add additional losses associated with specific downstream tasks. Autoregressive objectives that predict the next token which are common in decoder only LLMs are of limited use in scFMs due to the lack of inherent ordering in genes.

Refining scFMs with fine-tuning A key advantage of scFMs is their ability to leverage knowledge acquired during training for a variety of downstream tasks with no additional training (in zero-shot mode) or limited additional training (fine-tuning) on the new dataset. Most of the transformer layers are frozen to preserve general knowledge and prevent overfitting with updates to certain layers. Finetuning dataset size can vary from hundreds to millions of cells, depending on the task and data. To limit the computational expense, parameter-efficient fine-tuning techniques such as low-rank adaptation (LoRA), which adds a few additional trainable parameters to each attention layer instead of updating the full weight matrix are used (Baek et al., 2025).

Encoding metadata in scFMs Differing metadata across datasets and experiments introduces batch which affects the performance of scFMs (Baek et al., 2025). Most existing models either do not explicitly model this (Rosen et al., 2023) or use fixed encoded covariates (Cui et al., 2024; Hao et al., 2024) losing complex patterns encoded in the metadata such as similarity between different cell types, tissues, or diseases. For example, (Gayoso et al., 2021) introduces categorical covariates and (Bian et al., 2024a; Tejada-Lapuerta et al., 2025) introduce additional contextual tokens. Recent methods have begun to incorporate richer semantic information from metadata by additively combining LLM-derived metadata embeddings with the cell embedding (Liu et al., 2026).

Incorporating spatial tissue structure in scFMs Unimodal scFMs have access to a large corpus of spatial training data. Many methods incorporate spatial coordinates via coordinate embeddings (Wen et al., 2023; Zou & Lei, 2026; Madhu et al., 2026). (Wang et al., 2025) introduce a spatial neighborhood loss term. (Zhao et al., 2025) combine relative spatial distances with cell embeddings Like us, (Zou & Lei, 2026) addresses the problem during finetuning. However, unlike us, they add an adaptive spatial coordinate embedding to the cell embedding.

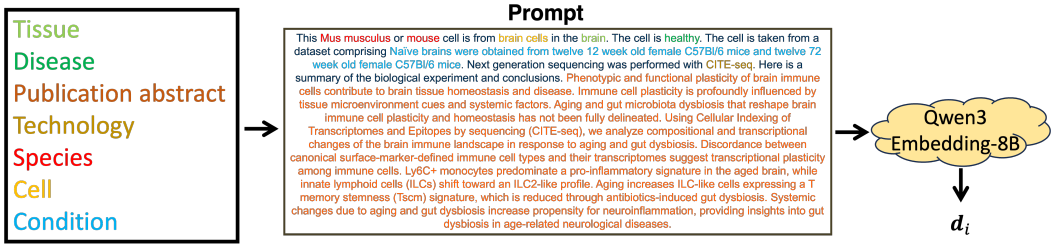


Figure 2: Example dataset prompt. Colors correspond to various dataset metadata fields as follows: red: species, orange: cell type, light green: tissue, green: disease, light blue: condition, gold: sequencing technology, and dark orange: abstract

A.1.3 Related work in Flow Matching

Flow matching (FloM) learns to transport samples using probability paths between arbitrary pairs of distributions consisting of an easy-to-sample source and an empirical target distribution (Morehead et al., 2026). Unlike diffusion models, the source can be any distribution (Tong et al., 2023), though it is often chosen to be Gaussian. Rectified flow matching (Liu et al., 2022), assumes an independent coupling between source and target samples and defines probability paths through linear interpolation between paired samples, yielding simple straight-line transport trajectories that approximate optimal transport behavior without requiring explicit OT optimization. This simple, scalable and stable formulation has been adopted successfully in Stable Diffusion (Esser et al., 2024).

A.1.4 Experimental measurement of paired sc- and ST- paired RNA-protein

CITE-Seq and Reap-Seq simultaneously profile RNA and surface proteins at single cell resolution. Variants include Perturb-CITE-Seq which also allows CRISPR perturbation tracking by measuring guide RNA expression and ECCITE-Seq, extending Perturb-CITE-Seq with cell hashing for multiplexed sample readout, and immune receptor profiling (TCR/BCR). Spatial-CITE-seq jointly profiles RNA and protein levels in groups of cells or spots in tissue slices while additionally measuring 2D spatial coordinates. Neat-seq measures intracellular proteins alongwith RNA and chromatin. Since data is limited in each experiment, an effective RNA-conditioned protein generator would adapt to varying sequencing technologies in zero-shot mode or with very limited data fine-tuning.

A.2 RNAProFlo Details

A.2.1 Dataset and cell metadata embeddings

Single cell RNA and protein expression levels are influenced by intrinsic biological state (cell-cycle, sequence, genetics) as well as cellular state (cell-type, tissue, disease, age, species) and technical artifacts (sequencing batch, dataset, experiment, laboratory, sequencing technology). Different datasets have diverse types and categories of metadata which poses a challenge for data integration. Our metadata embedding module unifies metadata across datasets, captures semantics in metadata content, handles cellular and technical variation, and provides context for protein generation without dataset specific training. We curate a set of common metadata fields and design custom descriptive prompts which we encode into $d_{\text{meta}} = 4096$ dimensional embeddings $\mathbf{d}_i \in \mathbb{R}^{d_{\text{meta}}}$ and $\mathbf{c}_i \in \mathbb{R}^{d_{\text{meta}}}$ with Qwen3-Embedding-8B (Zhang et al., 2025). Dataset fields include tissue, disease, sequencing technology, species, cell type and condition metadata. We further obtain the publication abstract from the source publication. For the cell embedding, we select common sc-omics metadata fields such as Batch, Donor ID, Donor, SampleName, original identifier or Orig.Ident, Patient, Batch indices, Sample ID, Sample, DatasetID, seuratCluster, Coarse and high resolution cell type, cell type annotation levels 1, 2, and 3, cluster name, and additional dataset specific fields like covid diagnosis start to sample and development stage. We additionally incorporate the dataset identifier. Example prompts are in Fig. 2 and Fig. 3. Missing fields are omitted and new fields can be incorporated by adding to the prompt.

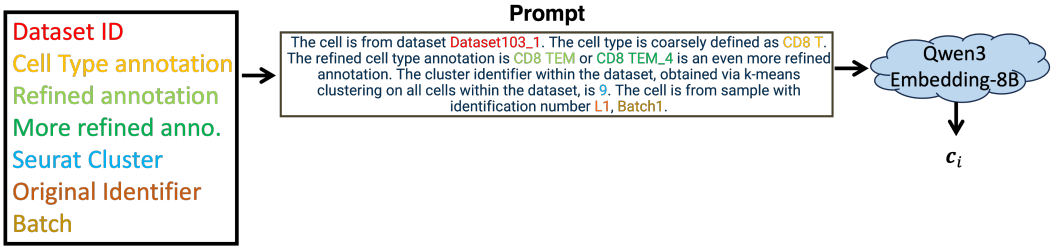


Figure 3: Example cell prompt. Colors correspond to cell metadata fields as follows: red: dataset ID, orange: cell type, light green: (refined) cell type annotation level 2 (L2), green: (more refined) cell type annotation level 3 (L3), light blue: seurat cluster, dark orange: orig.ident, and gold:batch

A.2.2 Sequence embeddings

The underlying mechanism of protein translation is complex and poorly understood. However, sequence embeddings from RNA and protein language models have been shown to capture information related to molecular structure, function, localization, evolutionary relationships, and, in some cases, regulatory mechanisms influencing translation efficiency and post-translational modification propensity (Rives et al., 2021; Lin et al., 2023; Yin et al., 2025; Wang et al., 2024c). The UCE scFM (Rosen et al., 2024) incorporates sequence embeddings into their tokenizer. Further, the number and set of proteins measured in each experiment varies widely, ranging from 7 to over 250 proteins. The set of important or highly variable genes in each dataset also differs and incorporating the entire set of over $20,000$ genes is computationally expensive and can introduce noise (Bian et al., 2024b; Cui et al., 2024). Hence, instead of learned lookup tables encoding gene names or identities, we leverage the RNA and protein sequence embeddings to encode feature identity.

A key challenge is mapping experimental dataset specific protein identifiers to their corresponding sequence. We accomplish this by matching the protein identifier to UniProt (uni, 2025) name, gene, and species fields to recover protein sequences and identifiers. We leverage top-performing protein language model ESM2 (Lin et al., 2023). Since the maximum protein sequence length in ESM2 is 1022, we aggregate resulting embeddings obtained from a sliding window of size 1022 with a step size of 511.

While several RNA-sequence specific foundation models exist (Yin et al., 2025; Wang et al., 2024c), ESM2 (Lin et al., 2023) has been widely adopted in the sequence language modeling space, with demonstrated applications on RNA sequence as well(Rosen et al., 2023; 2024). This is achieved by mean aggregation of all the protein embeddings that the gene translates to. In RNAProFlo, to prevent underlying differences in the sequence embedders from driving the RNA-protein cross attention layer weights, we adopt a similar approach as that in protein, leveraging ESM2 precomputed gene embeddings from UCE (Rosen et al., 2024).

Sequence embedding compression and normalization We learn a shared compression to the d_t-dimensional token space on the ESM2-sequence-derived embedding vectors across all genes $\mathbf{e}_{g_{\text{rna}}} \in \mathbb{R}^{d_{\text{esm}}}$ and proteins $\mathbf{e}_{p_{\text{prot}}} \in \mathbb{R}^{d_{\text{esm}}}$. This operation is applied identically across all genes and proteins, yielding normalized sequence-aware token representations. Specifically, we learn $\mathbf{W}_e \in \mathbb{R}^{d_{\text{esm}} \times d_t}$ and $\mathbf{b}_e \in \mathbb{R}^{d_t}$. For $o = \{g_{\text{rna}}, p_{\text{prot}}\}$ we have:

$$\mathbf{h}_o^{\text{esm}} = \text{LayerNorm}(\mathbf{W}_e^T \mathbf{e}_o + \mathbf{b}_e). \tag{17}$$

where LayerNorm denotes standard layer normalization defined for clarity in A.2.4.

RNA and Protein Tokens Each gene g in cell i is represented by expression scalar r_{g_i}. We linearly map scalar expression to d_t-dimensional token space with a shared mapping across all genes and cells. We learn $\mathbf{W}_r \in \mathbb{R}^{d_t}$ and $\mathbf{b}_r \in \mathbb{R}^{d_t}$ and apply it identically across all genes in each cell:

$$\mathbf{h}_{g_i}^{\text{expr}} = \mathbf{W}_\mathbf{r} r_{g_i} + \mathbf{b}_\mathbf{r} \tag{18}$$

Similarly, for each protein p in cell i represented by expression scalar p_{p_i}, we learn $\mathbf{W}_p \in \mathbb{R}^{d_t}$ and $\mathbf{b}_p \in \mathbb{R}^{d_t}$ and apply it identically across all proteins in each cell:

$$\mathbf{h}_{p_i}^{\text{expr}} = \mathbf{W}_{\mathbf{p}} p_{p_i} + \mathbf{b}_{\mathbf{p}} \tag{19}$$

Tokenization with sequence-based expression modulation We modulate expression with sequence embeddings by adapting Feature-wise Linear Modulation (FiLM) (Perez et al., 2018). We learn modulation parameters for genes $\boldsymbol{\gamma}_{g_{\text{rna}}}, \boldsymbol{\beta}_{g_{\text{rna}}} \in \mathbb{R}^{d_t}$ and proteins $\boldsymbol{\gamma}_{p_{\text{prot}}}, \boldsymbol{\beta}_{p_{\text{prot}}} \in \mathbb{R}^{d_t}$ as follows:

$$[\boldsymbol{\gamma}_{g_{\text{rna}}}, \boldsymbol{\beta}_{g_{\text{rna}}}] = \mathbf{W}_{\text{film}}^{\text{rna}} \mathbf{h}_{g_{\text{rna}}}^{\text{esm}} + \mathbf{b}_{\text{film}}^{\text{rna}}, \quad [\boldsymbol{\gamma}_{p_{\text{prot}}}, \boldsymbol{\beta}_{p_{\text{prot}}}] = \mathbf{W}_{\text{film}}^{\text{prot}} \mathbf{h}_{p_{\text{prot}}}^{\text{esm}} + \mathbf{b}_{\text{film}}^{\text{prot}} \tag{20}$$

where $\mathbf{W}_{\text{film}}^{\text{rna}}, \mathbf{W}_{\text{film}}^{\text{prot}} \in \mathbb{R}^{2d_t \times d_t}$ and $\mathbf{b}_{\text{film}}^{\text{rna}}, \mathbf{b}_{\text{film}}^{\text{prot}} \in \mathbb{R}^{2d_t}$.

For cell i modulated gene g and protein p tokenized representation respectively is:

$$\mathbf{z}_{g_i}^{\text{rna}} = \text{LayerNorm}(\mathbf{h}_{g_{\text{rna}}}^{\text{esm}} + \boldsymbol{\gamma}_{g_{\text{rna}}} \odot \mathbf{h}_{g_i}^{\text{expr}} + \boldsymbol{\beta}_{g_{\text{rna}}}), \tag{21}$$

$$\mathbf{z}_{p_i}^{\text{prot}} = \text{LayerNorm}(\mathbf{h}_{p_{\text{prot}}}^{\text{esm}} + \boldsymbol{\gamma}_{p_{\text{prot}}} \odot \mathbf{h}_{p_i}^{\text{expr}} + \boldsymbol{\beta}_{g_{\text{prot}}}) \tag{22}$$

where $\odot$ denotes Hadamard multiplication. Our formulation allows the model to interpret expression values in a gene- or protein-specific manner, where sequence-derived features dynamically control the scaling and shifting of expression embeddings. Combining expression embedding $\mathbf{h}^{\text{expr}}$ with molecular identity from the sequence embedding $\mathbf{h}^{\text{esm}}$, enables the model to generalize across genes or proteins unseen during training.

A.2.3 Sinusoidal Timestep Embedding

Following Transformer positional encodings (Vaswani et al., 2017) and DiT (Peebles & Xie, 2023), we represent the continuous flow timestep using deterministic sinusoidal Fourier features. Let $t \in [0, 1]$ denote the flow matching interpolation timestep and d_t denote the token dimension, where d_t is even. We first define a set of exponentially-spaced frequencies:

$$\omega_i = 10000^{-\frac{i}{d_t/2-1}}, \qquad i = 0, \ldots, \frac{d_t}{2} - 1. \tag{23}$$

The timestep embedding is then constructed as

$$\tilde{\mathbf{e}}_{\mathbf{t}} = [\sin(\omega_0 t), \ldots, \sin(\omega_{d/2-1} t), \cos(\omega_0 t), \ldots, \cos(\omega_{d_t/2-1} t)] \in \mathbb{R}^{d_t} \tag{24}$$

This formulation provides a multi-scale continuous representation of time, where low-frequency components encode coarse trajectory information while high-frequency components capture fine-grained temporal variation. Since nearby timesteps yield nearby embeddings, the representation varies smoothly along the flow trajectory and facilitates stable conditioning of the transformer velocity field. Following DiT (Peebles & Xie, 2023), the fixed sinusoidal embedding is projected through a learnable multilayer perceptron (MLP) to obtain the learned timestep conditioning vector:

$$\mathbf{e}_t = W_2 \,\text{SiLU}\,(W_1 \tilde{\mathbf{e}}_{\mathbf{t}} + b_1) + b_2, \quad \mathbf{e}_t \in \mathbb{R}^{d_t} \tag{25}$$

In our architecture, the timestep embedding is fused with the metadata embedding from 2:

$$\mathbf{f}_{i,t} = [\mathbf{e}_t; \mathbf{z}_i^{\text{meta}}] \tag{26}$$

and projected with an MLP with SiLU activation:

$$\mathbf{g}_{i_t} = \text{SiLU}\,(W \mathbf{f}_{i,t} + b), \quad \mathbf{g}_{i_t} \in \mathbb{R}^{d_t} \tag{27}$$

The resulting conditioning vector is used to modulate transformer blocks through Adaptive Layer Normalization with Zero Initialization (AdaLN-Zero).

A.2.4 Layer Normalization

We use the standard formulation with learnable affine parameters $\boldsymbol{\gamma}, \boldsymbol{\beta} \in \mathbb{R}^d$:

$$\mathbf{h}_l = \text{LayerNorm}(\tilde{\mathbf{h}}_l) = \frac{\tilde{\mathbf{h}}_l - \mu_l}{\sigma_l} \odot \boldsymbol{\gamma} + \boldsymbol{\beta}, \tag{28}$$

with mean and variance computed across the feature dimension and a stabilizing constant $\epsilon = 10^{-5}$:

$$\mu_l = \frac{1}{d} \sum_{k=1}^{d} \tilde{h}_{l,k}, \quad \sigma_l = \sqrt{\frac{1}{d} \sum_{k=1}^{d} (\tilde{h}_{l,k} - \mu_l)^2 + \epsilon}, \tag{29}$$

A.3 TRANSFORMER

We embed time using sinusoidal embeddings A.2.3 and fuse with metadata 3. Each transformer block applies:

$$\mathbf{z} \leftarrow \mathbf{z} + \mathrm{SelfAttn}(\mathrm{AdaLN}(\mathbf{z}, \boldsymbol{\gamma}_s, \boldsymbol{\beta}_s)), \tag{30}$$

$$\mathbf{z} \leftarrow \mathbf{z} + \mathrm{CrossAttn}(\mathrm{AdaLN}(\mathbf{z}, \boldsymbol{\gamma}_c, \boldsymbol{\beta}_c), \mathbf{z}^{\mathrm{rna}}), \tag{31}$$

$$\mathbf{z} \leftarrow \mathbf{z} + \mathrm{MLP}(\mathrm{AdaLN}(\mathbf{z}, \boldsymbol{\gamma}_{\mathrm{MLP}}, \boldsymbol{\beta}_{\mathrm{MLP}})). \tag{32}$$

We resolve the computational attention bottleneck by limiting the model to the top 2000 highly variable genes per dataset while sequence based expression modulation handles non-overlapping genes and proteins across datasets.

A.4 DECODER

As described in 10, protein expression levels are recovered with:

$$\hat{\mathbf{p}} = \mathrm{Softplus}(\mathrm{MLP}(\mathbf{Y})), \quad \mathbf{Y} = \boldsymbol{\gamma}_{\mathrm{esm-dec}} \odot \hat{\mathbf{z}}^{\mathrm{prot}} + \boldsymbol{\beta}_{\mathrm{esm-dec}} \tag{33}$$

where the 2-layer MLP is:

$$\mathrm{MLP}(\mathbf{Y}) = \mathbf{W}_2 \,\mathrm{LayerNorm}\left(\mathrm{GELU}\left(\mathbf{W}_1 \mathbf{Y} + \mathbf{b}_1\right)\right) + \mathbf{b}_2, \tag{34}$$

where $\mathbf{W}_1, \mathbf{W}_2, \mathbf{b}_1, \mathbf{b}_2$ are learned weights and biases.

A.5 GATED SPATIAL NICHE CROSS ATTENTION

The 2-layer MLP in 15 learns the head-specific bias $\beta_h(d)$:

$$\boldsymbol{\beta}(d) = W_2 \,\mathrm{SiLU}(W_1 \boldsymbol{\phi}(d) + \mathbf{b}_1) + \mathbf{b}_2, \quad \beta_h(d) = [\boldsymbol{\beta}(d)]_h. \tag{35}$$

A.6 STAGE 3 LOSS

To encourage alignment between transcriptomic and proteomic representations, we employ a symmetric contrastive objective over pooled RNA and protein embeddings. Specifically, RNA and protein token representations are first aggregated across tokens and ℓ_2 normalized to obtain cell-level embeddings:

$$\mathbf{z}_i^{\mathrm{rna}} = \mathrm{Normalize}\left(\frac{1}{N_g}\sum_{k=1}^{N_g} \mathbf{Z}_{ik}^{\mathrm{rna}}\right), \tag{36}$$

$$\mathbf{z}_i^{\mathrm{prot}} = \mathrm{Normalize}\left(\frac{1}{N_p}\sum_{k=1}^{N_p} \mathbf{Z}_{ik}^{\mathrm{prot}}\right), \tag{37}$$

where N_g and N_p denote the number of RNA and protein tokens, respectively. Pairwise similarities between RNA and protein embeddings across a minibatch are then computed using scaled cosine similarity:

$$S_{ij} = \frac{(\mathbf{z}_i^{\mathrm{rna}})^\top \mathbf{z}_j^{\mathrm{prot}}}{\tau}, \tag{38}$$

where τ is a temperature parameter controlling the sharpness of the similarity distribution. The contrastive loss is formulated as a symmetric InfoNCE objective:

$$\mathcal{L}_{\mathrm{contrast}} = \frac{1}{2}\left(\mathrm{CE}(S, I) + \mathrm{CE}(S^\top, I)\right), \tag{39}$$

where CE denotes cross-entropy loss and I represents the identity labels indicating matched RNA–protein pairs from the same cell. The first term encourages each RNA embedding to align with its corresponding protein embedding, while the second enforces the reciprocal protein-to-RNA alignment. This objective promotes a shared multimodal latent space in which paired transcriptomic and proteomic representations are maximally similar while non-matching cells are separated. The expression loss is:

$$\mathcal{L}_{\mathrm{expr}} = \frac{1}{KB}\sum_{k,b} \|\hat{\mathbf{p}}_{b,k} - \mathbf{p}_b\|_2 \tag{40}$$

A.6.1 Model Details

RnaProFlo uses 2 diffusion transformer (DiT) style blocks with 2 attention heads and a feed-forward expansion ratio of 2. It has 1.1 million parameters and took approximately 6 hours to train on an NVIDIA H200 GPU with 144GB of available memory, with peak memory usage of 71.5GB. Our model component embedding dimensions are $d_{esm} = 5120$ from ESM2(Lin et al., 2023), $d_{meta} = 4096$ from Qwen3-Embedding-8B (Zhang et al., 2025), and $t = 64$ for model tokens. We use a batch size of 128. We use AdamW as the training optimizer with learning rate 10^{-4} and weight decay 0.01. Gradients are clipped to norm 1.0. During training, we split the training dataset into 99% training and 1% validation. We apply early stopping when the cosine distance on the validation set does not increase by more than 0.0001 between epochs. In this way, all training runs terminated after 1 epoch. During validation and inference, we generate protein representations by integrating the learned velocity field using Euler ODE integration with 10 sampling steps. Contrastive loss temperature $\tau = 0.1$.

All linear layers were initialized with Xavier-uniform weights and zero biases. In addition, the final prediction head and adaLN modulation layers were zero-initialized so the input to the residual path of the transformer block is effectively zero at the start to stabilize training, following the adaLN-Zero design (Peebles & Xie, 2023). LayerNorm affine parameters, where present, used PyTorch defaults. For stage 1, reconstruction loss weight $\lambda_{\text{recon}} = 1$, contrastive weight $\lambda_{\text{contrast}} = 0.01$. For stage 3, expression weight $\lambda_{\text{expr}} = 1$ and contrastive weight $\lambda_{\text{contast}} = 0.05$. The number of rollout steps $= 8$. All stage 3 hyperaparameters were set by running a hyperparameter sweep on 15% of an epoch and selecting the combination with the highest cosine distance on the validation set. For stage 1, we our hyperparameter sweep measured reconstruction MSE and the RNA-protein top-K ($K = 1, 5, 10$) match. For spatial niche fine-tuning we use neighborhood size $K = 8$ based on a parameter sweep (Table 14).

LoRA fine-tuning was performed without defining a validation set or early stopping for a fixed number of 50 training epochs. 6.9% of model weights are adapted in stage3. All hyperparameters were the same as stage 3 training except the AdamW learning rate which was increased to 10^{-2} since LoRA adapters operate in a constrained low-dimensional subspace and larger learning rates enable rapid adaptation while maintaining stable optimization. LoRA parameter A is initialized with Kaiming uniform initialization and B is initialized to zero, ensuring that the fine-tuned model initially reproduces the pretrained function exactly.

Dropout of 0.1 is applied for regularization and to reduce dataset specific overfitting within the attention and feedforward pathways. Specifically, attention dropout is applied to normalized attention weights in both self- and cross-attention layers, while feedforward dropout is applied after the Gaussian Error Linear Unit (GELU) activation in the transformer MLP blocks. During LoRA fine-tuning, additional dropout is applied within the low-rank adaptation branch.

Details on the activation functions are in A.6.2

A.6.2 Model Activations

Transformer feedforward blocks use the Gaussian Error Linear Unit (GELU) activation (Hendrycks & Gimpel, 2016) $\mathrm{GELU}(x) = x\Phi(x)$ where $\Phi(x)$ is the Gaussian cumulative distribution function. GELU provides smooth nonlinear gating and improves optimization stability in transformer architectures.

Timestep-conditioning and adaptive normalization layers use the Sigmoid Linear Unit (SiLU) activation (Ramachandran et al., 2017) $\mathrm{SiLU}(x) = x\sigma(x)$ where $\sigma(x)$ is the sigmoid function. SiLU stabilizes conditioning pathways in continuous-time generative models.

The decoder uses a Softplus output activation $\mathrm{Softplus}(x) = \log(1+e^x)$ which ensures non-negative protein expression predictions while maintaining smooth gradients.

A.6.3 Finetuning with LoRA

For a frozen linear layer with weight matrix $W \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, LoRA parameterizes the fine-tuned layer as:

$$\mathrm{LoRA}(x) = Wx + \frac{\alpha}{r} BAx, \tag{41}$$

where $A \in \mathbb{R}^{r \times d_{\text{in}}}$ and $B \in \mathbb{R}^{d_{\text{out}} \times r}$ are trainable low-rank matrices, r is the LoRA rank, and α is a scaling hyperparameter. LoRA adapters are inserted into the self-attention query-key-value pro-

jection, self-attention output projection, cross-attention query projection, cross-attention key-value projection, cross-attention output projection, and the final token-space velocity prediction layer. During fine-tuning, we freeze the pretrained backbone parameters and optimize only the LoRA adapter weights and decoder parameters.

A.7 Training and Test Data

We create a training dataset by downloading all available paired human and mouse sc-RNA-protein data from scProteomicsDB (Wang et al., 2024a) and obtaining their corresponding article abstracts from PubMed. Our data comprises 3.3 million cells from 38 datasets covering 2 sequencing technologies (CITE-seq and ECCITE-seq), 2 species (human and mouse), 14 tissue types, 24 primary cell types, and 16 disease states. Pie plots illustrating the composition are at Fig 4. Number of cells in a dataset range from 588 to $632,100$ cells with varying number of proteins from 11 to 279. Fig 5 shows the distribution of cells and proteins. Details of the training data are in Table 6. Datasets were entirely used for either training (with a validation split) or testing (with subsequent further splits for finetuning) except for Dataset103_1 which was split with 90% for training and 10% for testing.
We test our model in various scenarios - zero-shot, finetuning, generalization to unseen species, technologies, spot-based spatial omics, and perturbation data. We use a total of 18 datasets from scProteomicsDB (Wang et al., 2024a), Spatial CITE-Seq (Liu et al., 2023), Reap-Seq (Peterson et al., 2017), Neat-Seq (Chen et al., 2022) with a total of $436,504$ cells. The number of cells in each dataset range from 902 to $218,331$ and the number of proteins vary from 7 to 273. Testing data details are at Table 7.

A.8 Evaluations

We evaluate the performance of RNAProFlo and 4 state of the art (SOTA) models. We utilized two foundation models CAPTAIN (Ji et al., 2025) and scTranslator (Liu et al., 2025) and included a top performing VAE model totalVI (Gayoso et al., 2021) since its unimodal scRNA-seq counterpart scVI (Lopez et al., 2018) has been shown to outperform scFMs on some tasks (Kedzierska et al., 2025). We also include the recent (April 2026) transformer-based scProTrans (Zhao et al., 2026) due to its use of sequence based gene and protein encoders. Finally, since RNA expression is often used as a proxy for protein levels when studying biological systems and processes such as transcription-factor regulation, we include a baseline (Baseline) by setting the predicted protein level to the corresponding gene expression value from which it was translated. All methods were run with default parameters with details included in A.11. Methods were evaluated across 3 established metrics in this domain (Liu et al., 2025): Relative Mean Square Error (MSE), Cosine similarity (Cos) across cells and Cosine similarity Cos(P) across proteins. Detailed definitions are included in A.10.
For zero-shot evaluations, the author-trained scFM models (CAPTAIN and scTranslator) were utilized. However, since totalVI and scProTrans are not foundation models, we trained them on our 3.2 million training dataset with default parameters. For fine-tuning experiments, the scFM models (CAPTAIN and scTranslator) were finetuned as recommended by the authors. Additionally, to evaluate the performance of a non scFM with limited out of distribution training data, we used the trained totalVI model and performed additional training with the corresponding training data used in fine-tuning experiments.

A.9 RnaProDisc

RnaProDisc follows the same 3-stage training and losses as RnaProFlo with the exception of Stage2. Instead of flow matching, we use a masked reconstruction loss in latent space with full masking. We observed with a parameter sweep on 15% data subset that full masking performed better than partial masking.

A.10 Evaluation Metrics

Let $\hat{\mathbf{y}}_i \in \mathbb{R}^P$ denote the predicted protein expression vector for cell i, and $\mathbf{y}_i \in \mathbb{R}^P$ denote the corresponding ground-truth protein expression vector, where P is the number of proteins.

Table 6: Summary of paired sc-RNA-protein datasets used for training in this study

Dataset	Technology	Cells	Proteins	Species	Tissue	Cell Type	Disease
Dataset100	CITE-seq	355,433	38	Human	Blood	Immune cells	Multiple sclerosis
Dataset101	CITE-seq	632,100	138	Human	Blood	PBMC	COVID-19
Dataset103_1	CITE-seq	145,587	228	Human	Blood	Immune cells	None
Dataset103_2	ECCITE-seq	49,147	54	Human	Blood	Immune cells	None
Dataset104	CITE-seq	4,712	34	Mouse	Tumor	Immune cells	Breast cancer
Dataset105	CITE-seq	4,451	75	Mouse	Aorta	CD45+ cells	Aortic aneurysm
Dataset106	CITE-seq	24,320	11	Mouse	Spleen	Dendritic cells	None
Dataset107	CITE-seq	100,620	143	Human	Bone marrow	BM cells	AML
Dataset74	CITE-seq	124,448	193	Human	Blood	PBMC	MIS-C/COVID-19
Dataset75_1	CITE-seq	26,919	268	Human	Glioblastoma	Myeloid cells	Glioblastoma
Dataset75_2	CITE-seq	13,635	174	Mouse	Glioblastoma	Myeloid cells	Glioblastoma
Dataset76_1	CITE-seq	43,662	198	Human	Liver	Liver cells	Obesity
Dataset76_2	CITE-seq	91,040	173	Mouse	Liver	Liver cells	None
Dataset76_3	CITE-seq	55,457	115	Mouse	Liver	Liver cells	NAFLD
Dataset77_1	CITE-seq	16,828	110	Mouse	Spleen/LN	Immune cells	None
Dataset77_2	CITE-seq	15,820	207	Mouse	Spleen/LN	Immune cells	None
Dataset78_1.1	CITE-seq	955	21	Human	Lung	Dendritic cells	NSCLC
Dataset78_1.2	CITE-seq	588	58	Human	Lung	Dendritic cells	NSCLC
Dataset80_1	CITE-seq	246,964	192	Human	Blood	Adaptive immune cells	COVID-19
Dataset80_2	CITE-seq	125,117	192	Human	Blood	Innate immune cells	COVID-19
Dataset81	CITE-seq	4,426	19	Human	Bone marrow	AML cells	AML
Dataset82_1	CITE-seq	23,287	18	Human	Blood	CAR-T cells	ALL
Dataset82_2	CITE-seq	31,105	18	Human	Blood	CAR-T cells	ALL
Dataset82_3	CITE-seq	30,484	18	Human	Blood	CAR-T cells	ALL
Dataset82_4	CITE-seq	16,450	18	Human	Blood	CAR-T cells	ALL
Dataset83	CITE-seq	53,201	87	Human	Blood	Immune cells	None
Dataset84	CITE-seq	10,253	64	Mouse	Mesenteric LN	MLN cells	None
Dataset88	CITE-seq	44,041	84	Human	Bone marrow	BMMCs	B-ALL
Dataset89_1	CITE-seq	5,798	32	Mouse	Bone marrow	Progenitors	None
Dataset89_2	CITE-seq	16,015	82	Mouse	Bone marrow	Progenitors	None
Dataset91	CITE-seq	500,089	31	Human	Blood	Memory T cells	Tuberculosis
Dataset92	CITE-seq	85,000	16	Human	Brain	Microglia/immune cells	Epilepsy
Dataset93	CITE-seq	211,442	279	Human	Bone marrow	BM mononuclear cells	AML
Dataset94	CITE-seq	38,652	185	Human	Blood	Immune cells	COVID-19
Dataset95	CITE-seq	26,407	18	Human	Fetal liver	Fetal liver cells	None
Dataset96	CITE-seq	52,945	29	Mouse	Bone marrow	HSPCs	None
Dataset97	CITE-seq	33,322	16	Mouse	Adipose tissue	Leukocytes	None
Dataset98	CITE-seq	2,298	56	Mouse	Salivary gland	Tumor cells	SCC
Dataset99	CITE-seq	13,052	34	Mouse	Brain	Brain cells	None

The relative mean squared error (MSE) for cell i is computed as:

$$\mathrm{RelMSE}_i = \frac{\sum_{j=1}^{P}(\hat{y}_{ij} - y_{ij})^2}{\sum_{j=1}^{P} y_{ij}^2 + \epsilon}, \tag{42}$$

where $\epsilon = 10^{-8}$ is a small constant for numerical stability.

Cell-wise cosine similarity (Cos) is computed as:

$$\mathrm{Cosine}_i = \frac{\hat{\mathbf{y}}_i^\top \mathbf{y}_i}{\|\hat{\mathbf{y}}_i\|_2 \|\mathbf{y}_i\|_2}. \tag{43}$$

Protein-wise cosine similarity (Cos(P)) evaluates agreement across cells for each protein independently. Let $\hat{\mathbf{y}}_{\cdot j}$ and $\mathbf{y}_{\cdot j}$ denote the predicted and ground-truth expression vectors for protein j across all N cells:

$$\mathrm{CosineProt}_j = \frac{\hat{\mathbf{y}}_{\cdot j}^\top \mathbf{y}_{\cdot j}}{\|\hat{\mathbf{y}}_{\cdot j}\|_2 \|\mathbf{y}_{\cdot j}\|_2}. \tag{44}$$

A.11 Baseline Methods

Since totalVI (Gayoso et al., 2021) and scProTrans (Zhao et al., 2026) are not scFMs, we trained them from scratch on our 3.2 million cell dataset. All methods was run with its recommended default parameters. scTranslator was finetuned and zero-shot evaluated on an NVIDIA H200 GPU. Finetune parameters were selected automatically by "stage3_fine-tune.py" for each dataset. Zero-shot was

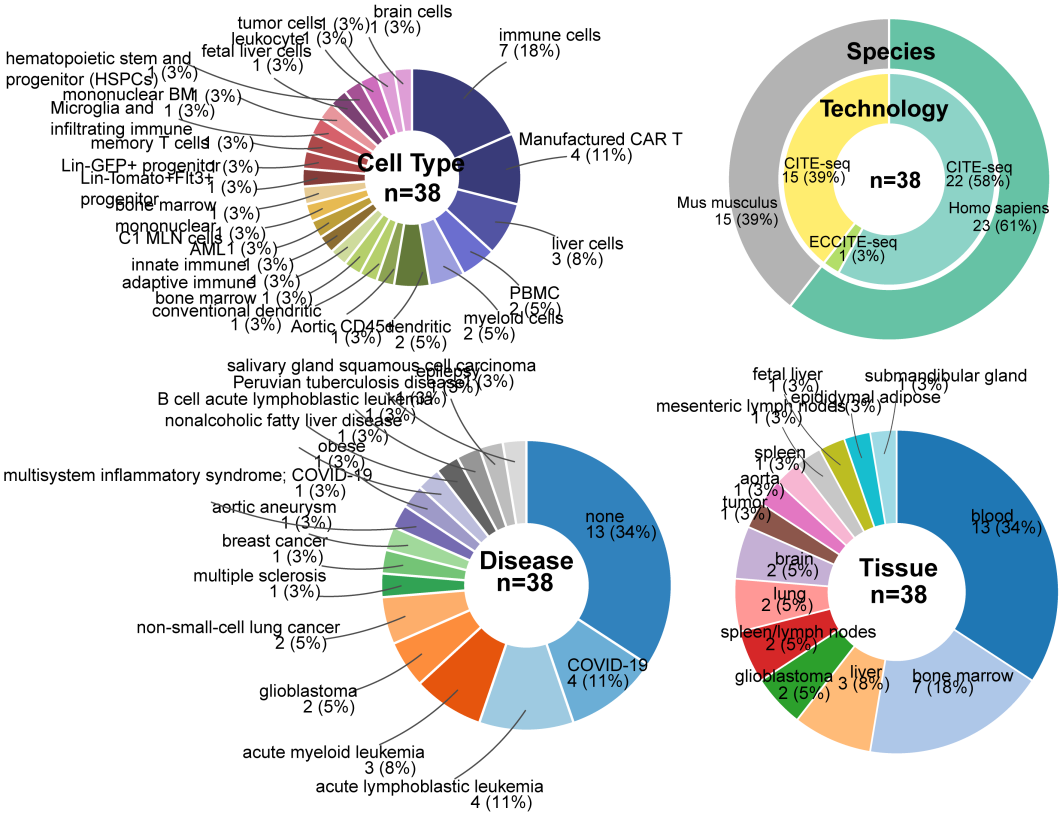


Figure 4: Pie plots illustrating the cell type, species, technology, disease and tissue composition of our training dataset

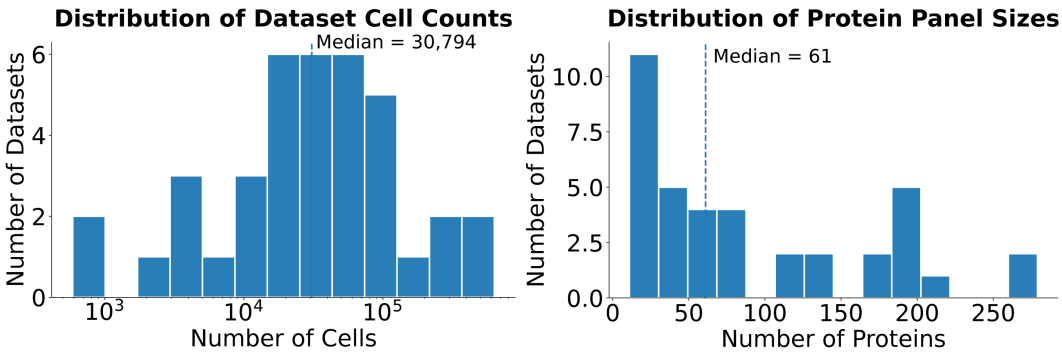


Figure 5: Distributions of the number of cells and protein panel size across datasets

Table 7: Summary of paired sc-RNA-protein datasets used for testing in this study. The corresponding experiment it was used in is listed. ZS denotes zero-shot, FT denotes fine-tuned.

Dataset	Technology	Cells	Proteins	Species	Tissue	Cell type	Disease	Experiment
Dataset76_5	CITE-seq	21,907	198	*Sus scrofa*	Liver	Liver cells	None	Species
Dataset85	CITE-seq	13,761	13	*Macaca mulatta*	Vaginal tissue	Vaginal epithelium	SHIV infection	Species
GSE213264_1	Spatial CITE-seq	1,303	189	*Mus musculus*	Spleen	–	None	Spatial
GSE213264_2	Spatial CITE-seq	2,037	189	*Mus musculus*	Colon	–	None	Spatial
GSE213264_3	Spatial CITE-seq	902	189	*Mus musculus*	Intestine	–	None	Spatial
GSE213264_4	Spatial CITE-seq	2,419	189	*Mus musculus*	Kidney	–	None	Spatial
GSE213264_5	Spatial CITE-seq	2,492	273	*Homo sapiens*	Tonsil	–	None	Spatial
GSE213264_6	Spatial CITE-seq	2,494	273	*Homo sapiens*	Spleen	–	None	Spatial
GSE213264_7	Spatial CITE-seq	2,500	273	*Homo sapiens*	Thymus	–	None	Spatial
GSE213264_8	Spatial CITE-seq	1,691	273	*Homo sapiens*	Skin	–	None	Spatial
GSE213264_9	Spatial CITE-seq	1,710	273	*Homo sapiens*	Glioblastoma	–	None	Spatial
GSE100501_1	REAP-seq	15,701	44	*Homo sapiens*	Blood	PBMC	None	Technology
GSE100501_2	REAP-seq	14,546	76	*Homo sapiens*	Blood	$CD8^+$ naive T cells	None	Technology
GSE178707	NEAT-seq	8,472	7	*Homo sapiens*	Blood	Primary CD4 memory T cells	None	Technology
Dataset103_1	CITE-seq	16,177	228	*Homo sapiens*	Blood	Immune cells	None	ZS, FT
Dataset78_1.3	CITE-seq	1,254	81	*Homo sapiens*	Lung	Dendritic cells	NSCLC	ZS, FT
Dataset108	ECCITE-seq	37,819	61	*Homo sapiens*	Blood/Skin	PBMCs and skin cells	CTCL	ZS, FT
Dataset86	Perturb-CITE-seq	218,331	20	*Homo sapiens*	Tumor	Melanoma/Cas9 cells	Melanoma	Perturbation

Table 8: Performance comparison of RnaProFloS (Ours-Sp), the spatial variant of RnaProFlo-D (Ours-D-Sp), zero-shot RnaProFlo (Ours-ZS) and RnaProFlo-D (Ours-D-ZS) and other methods across spot-based spatial CITE-seq **mouse** datasets. For each dataset, finetuning was performed on 10% of all cells across all available proteins.

		Ours-Sp	Ours-D-Sp	Ours-ZS	Ours-D-ZS	Baseline	CAPTAIN FT	CAPTAIN	scTranslator FT	scTranslator
GSE213264_1 (Spleen)	Cos	**0.928** (0.030)	0.912 (0.031)	0.660 (0.038)	0.548 (0.038)	0.102 (0.118)	0.743 (0.064)	-0.005 (0.140)	0.896 (0.058)	0.635 (0.045)
	MSE	**0.140** (0.058)	0.170 (0.055)	0.581 (0.043)	0.708 (0.048)	0.997 (0.048)	0.466 (0.108)	1.151 (0.150)	0.222 (0.134)	1.348 (0.281)
GSE213264_2 (Colon)	Cos	**0.942** (0.032)	0.929 (0.030)	0.750 (0.042)	0.661 (0.038)	0.131 (0.120)	0.791 (0.081)	-0.026 (0.139)	0.897 (0.071)	0.676 (0.042)
	MSE	**0.117** (0.067)	0.140 (0.054)	0.453 (0.053)	0.993 (0.123)	0.988 (0.023)	0.373 (0.119)	1.198 (0.200)	0.245 (0.209)	0.888 (0.210)
GSE213264_3 (Intestine)	Cos	**0.909** (0.054)	0.895 (0.052)	0.642 (0.085)	0.605 (0.078)	0.142 (0.129)	0.696 (0.143)	-0.011 (0.189)	0.818 (0.099)	0.616 (0.078)
	MSE	**0.183** (0.113)	0.208 (0.094)	0.590 (0.114)	0.643 (0.100)	1.001 (0.071)	0.553 (0.243)	1.158 (0.165)	0.372 (0.194)	1.368 (0.477)
GSE213264_4 (Kidney)	Cos	**0.952** (0.017)	0.940 (0.016)	0.663 (0.029)	0.639 (0.028)	0.362 (0.196)	0.830 (0.052)	-0.001 (0.144)	0.914 (0.057)	0.658 (0.034)
	MSE	**0.096** (0.034)	0.120 (0.029)	0.568 (0.043)	0.594 (0.038)	0.969 (0.043)	0.312 (0.087)	1.140 (0.140)	0.188 (0.136)	0.998 (0.163)

run with "stage3_inference_without_finetune.py" for each dataset. Finetuning took approximately 3 hours per dataset and zero shot evaluation approximately 1 hour. CAPTAIN zero-shot and finetune were run on a docker container with an NVIDIA A40 GPU. Finetuning each dataset took approximately 3 hours and zero-shot around 30 minutes. To evaluate datasets, we utilized the provided "zeroshot_generate.py" from the original authors while for finetuning we utilized their "finetune.py" script. Default parameters were used. Default parameters were used with the source dataset ID as metadata. Training took 2 hours and 17 minutes on an NVIDIA A40 GPU. For fineteuning we trained for 50 epochs per dataset taking an average of 1.5 hours per dataset. scProTrans was trained on a Google Colab T4 GPU with a batch size of 16 and 3 epochs which was sufficient for convergence and observe "elbow" behavior. All other parameters were set to default values. Training time was approximately 2 hours.

A.12 Additional Results

A.12.1 Fine-tuning on different dataset sizes

The results are shown in (Table 10). For the finetuning experiment, the total testset across all 4 datasets was split with $10\%, 1\%, 0.1\%, 0.01\%$ used for finetuning and the remaining for evaluation. The per-dataset breakdown after the split is shown in (Table 9).

A.12.2 Ablations

The protein ESM2 (ESM(P)) or RNA ESM2 (ESM(R)) ablations were obtained by setting the corresponding embeddings to 0. Tokenization ablations on the FiLM layer was obtained by omitting

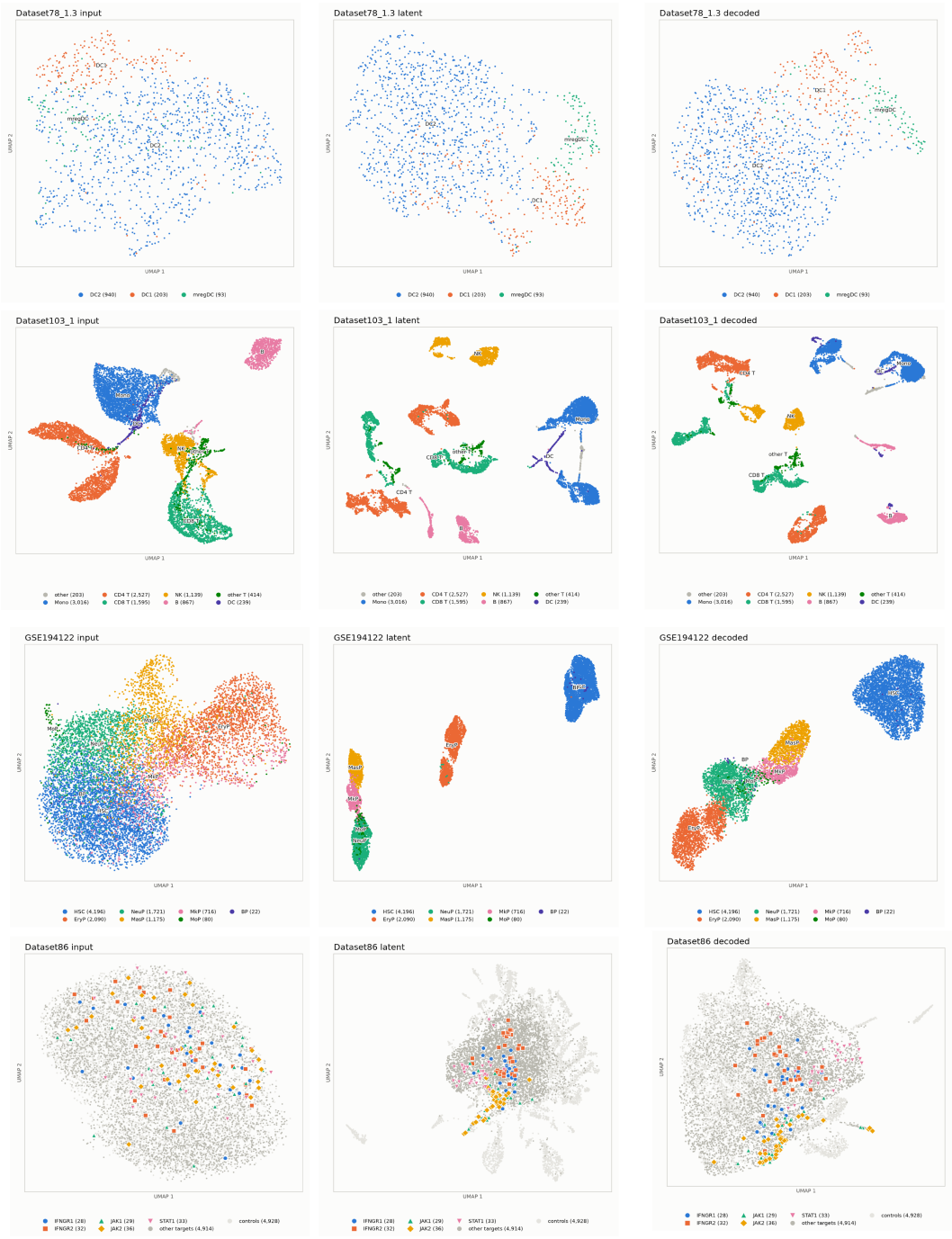


Figure 6: UMAPs showing input, latent and decoded cells colored by key features

Table 9: Number of fine-tuning (FT) and test cells at each data fraction for Tables 2 3 10

	Dataset103_1		Dataset108		Dataset78_1.3		Dataset86	
Fraction	FT	Test	FT	Test	FT	Test	FT	Test
10%	1,630	14,547	3,147	28,586	135	1,101	21,806	196,221
1%	166	16,011	320	31,413	9	1,227	2,177	215,850
0.1%	17	16,160	27	31,706	1	1,235	224	217,803

Table 10: Performance comparison of finetuned RNAProFlo (Ours-FT) and RnaProDisc (Ours-D-FT) alongwith other finetuned methods with various finetuning dataset sizes. Zero-shot RNAProFlo (Ours-ZS) and RnaProDisc (Ours-D-ZS) have been added for comparison. Size indicates the % of all cells used for finetuning, with the remaining for testing. Number of training and testing cells used is in Table 9. Values report mean.

		Dataset103_1		Dataset108		Dataset78_1.3		Dataset86	
Method	Size	Cos (↑)	MSE (↓)	Cos (↑)	MSE (↓)	Cos (↑)	MSE (↓)	Cos (↑)	MSE(↓)
Ours-FT	1%	0.962	0.076	0.922	0.153	0.915	0.168	0.957	0.088
	0.1%	0.957	0.092	0.887	0.233	0.823	0.361	0.946	0.116
Ours-D-FT	1%	0.960	0.084	0.920	0.157	0.932	0.138	0.957	0.085
	0.1%	0.955	0.092	0.904	0.213	0.845	0.292	0.953	0.094
Ours-ZS	1%	0.913	0.184	0.786	0.414	0.875	0.240	0.814	0.454
	0.1%	0.913	0.184	0.786	0.414	0.875	0.240	0.814	0.454
Ours-D-ZS	1%	0.961	0.082	0.683	0.546	0.842	0.296	0.843	0.504
	0.1%	0.961	0.082	0.683	0.546	0.842	0.297	0.843	0.504
CAPTAIN	1%	0.793	0.387	0.817	0.362	0.443	0.824	0.683	0.532
	0.1%	0.568	0.775	0.222	1.017	0.313	0.948	0.617	0.613
scTranslator	1%	0.854	0.315	0.775	0.442	0.607	0.680	0.974	0.058
	0.1%	0.662	0.624	0.926	0.152	0.935	0.139	0.872	0.246
totalVI	1%	0.341	0.982	0.657	0.783	0.148	0.991	0.445	0.825
	0.1%	0.375	1.106	0.450	1.651	0.103	1.272	0.521	0.743

ESM2 based FiLM modulation in 21 and aggregating the ESM2 and expression embeddings instead:

$$\mathbf{z}_{g_i}^{\text{rna}} = \text{LayerNorm}(\mathbf{h}_{g_{\text{rna}}}^{\text{esm}} + \mathbf{h}_{g_i}^{\text{expr}}), \quad \mathbf{z}_{p_i}^{\text{prot}} = \text{LayerNorm}(\mathbf{h}_{p_{\text{prot}}}^{\text{esm}} + \mathbf{h}_{p_i}^{\text{expr}}) \tag{45}$$

Ablations on loss terms were obtain by setting the term expresion (Expr) or Contrastive (Cont) coefficient to 0. Ablations on the cell (Cell), dataset (Data), and both (Cell&Data) metadata embeddings by setting the corresponding embeddings to 0. Abstract and celltype ablations were obtained by completely excluding them from all metadata.

A.13 Code Availability

All our training and testing data, code and model weights will be publicly available at https://anonymous.4open.science/r/RNAProFlo-ADC3.

Table 11: Spatial CITE-seq with limited fine-tuning data comprising 1% of all spatial data. Reported mean (standard deviation). Train and Test denote number of fine-tuning and testing cells.

				Ours-Sp		Ours-D-Sp		Ours-ZS		Ours-D-ZS	
Dataset	Species	Train	Test	Cos	MSE	Cos	MSE	Cos	MSE	Cos	MSE
GSE213264_1	Mouse	12	1,007	**0.901** (0.031)	**0.189** (0.055)	0.806 (0.034)	0.387 (0.070)	0.660 (0.038)	0.580 (0.043)	0.549 (0.038)	0.707 (0.048)
GSE213264_2	Mouse	18	1,928	**0.926** (0.031)	**0.143** (0.059)	0.845 (0.037)	0.290 (0.070)	0.750 (0.041)	0.452 (0.053)	0.661 (0.038)	0.992 (0.122)
GSE213264_3	Mouse	10	831	**0.894** (0.058)	**0.210** (0.111)	0.799 (0.075)	0.384 (0.155)	0.641 (0.086)	0.591 (0.115)	0.604 (0.078)	0.644 (0.101)
GSE213264_4	Mouse	24	2,339	**0.938** (0.018)	**0.121** (0.034)	0.835 (0.024)	0.313 (0.046)	0.663 (0.029)	0.568 (0.043)	0.639 (0.028)	0.594 (0.038)
GSE213264_5	Human	24	2,464	**0.964** (0.012)	**0.076** (0.021)	0.906 (0.016)	0.180 (0.030)	0.765 (0.021)	0.450 (0.045)	0.610 (0.021)	0.846 (0.065)
GSE213264_6	Human	25	2,427	**0.923** (0.025)	**0.155** (0.043)	0.830 (0.031)	0.313 (0.054)	0.638 (0.035)	0.609 (0.052)	0.547 (0.032)	0.721 (0.042)
GSE213264_7	Human	23	2,400	**0.967** (0.011)	**0.067** (0.021)	0.923 (0.012)	0.148 (0.023)	0.726 (0.016)	0.476 (0.026)	0.628 (0.014)	0.641 (0.026)
GSE213264_8	Human	16	1,536	**0.849** (0.051)	**0.304** (0.081)	0.779 (0.064)	0.429 (0.135)	0.608 (0.066)	0.635 (0.075)	0.445 (0.051)	0.819 (0.056)
GSE213264_9	Human	18	1,232	**0.941** (0.013)	**0.121** (0.024)	0.784 (0.016)	0.396 (0.022)	0.634 (0.019)	0.601 (0.036)	0.557 (0.021)	0.690 (0.023)

Table 12: Latent velocity and between-seed cosine similarity aggregated by evaluation pool in zero-shot RnaProFlo

Metric	Testset	More	Species	Tech	Spatial
Latent velocity cosine	0.823	0.843	0.811	0.772	0.817
Seed cosine	0.993	0.997	0.996	0.996	0.997

Table 13: In RNAProFlo without dataset-specific fine-tuning, protein self-attention assigns higher mean weight to protein pairs annotated as interactions in STRING Szklarczyk et al. (2023) (combined score > 700) than to pairs without such annotations. The mean attention weight ratio divides the average attention weight on annotated pairs by that on non-annotated pairs. For each K, we report the fold enrichment of STRING-annotated pairs among the top-K attended edges relative to the background frequency of annotated pairs among eligible edges.

Block and head	Mean attention weight ratio	$K = 1$	$K = 5$	$K = 10$	$K = 20$
Block 0, head 0	1.377	1.19×	1.15×	1.14×	1.12×
Block 0, head 1	1.359	1.11×	1.09×	1.09×	1.09×
Block 1, head 0	1.465	1.26×	1.21×	1.19×	1.16×
Block 1, head 1	1.417	1.18×	1.16×	1.15×	1.12×

Table 14: Validation performance for spatial neighborhood ablations on the validation set $N =$. N.N. denotes no niche, N.P. niche permutation, D.B. no distance bias, and D.P. distance permutation.

	Orig	Neighborhood size					Niche controls		Distance-bias controls	
Metric	$K = 10$	$K = 2$	$K = 4$	$K = 6$	$K = 8$	$K = 20$	N.N.	N.P.	D.B.	D.P.
Val MSE	0.254	0.255	0.256	0.271	0.264	0.255	0.265	0.271	0.255	0.261
Val Cos	0.945	0.945	0.945	0.943	0.944	0.945	0.944	0.943	0.944	0.944