scPathVQ: A Pathway-Guided Discrete Latent Interface for Single-Cell Understanding and Generation

Anonymous authors
Paper under double-blind review

Abstract

Effective single-cell transcriptomic representations are fundamental to modeling cellular states in virtual cells. Existing methods typically tokenize individual genes, but are inherently constrained by the high-dimensional, sparse, and unordered nature of transcriptomic inputs. Moreover, cell embeddings are often obtained by flattening and pooling latent sequences, which can obscure the functional gene modules that collectively shape cellular states. We introduce **scPathVQ**, a pathway-guided discrete latent interface that provides a compact representation while preserving functional gene-module structure for both cellular-state understanding and generation. Its biology-structured tokenizer organizes transcriptome-wide genes into pathway-induced units that capture functional gene modules, complemented by highly variable genes that provide cell-specific expression context. A latent bottleneck compresses these contextualized features into a fixed-length sequence. Cell-state vector quantization maps each latent component to a shared codebook. The resulting quantized sequence preserves functional gene-module structure for perturbation response modeling, while gated pooling yields a compact cell level representation for downstream understanding tasks. We pretrain scPathVQ on 34 million human cells and evaluate several downstream tasks. The pooled representation supports strong transfer across cell-level tasks, while the quantized sequence enables accurate perturbation-response prediction. Generated expression profiles recover gene pathway responses and associated functional cell-state patterns. Ablation studies further identify the contribution of pathway-guided organization to the learned representation.

1 Introduction

Large-scale single-cell atlases and perturbation screens enable computational models to characterize cellular diversity and predict transcriptional responses to genetic interventions (Hao et al., 2024; Rosen et al., 2026; Dixit et al., 2016; Norman et al., 2019). Single-cell foundation models learn from these resources and transfer pretrained representations to annotation, integration, and response prediction (Theodoris et al., 2023; Cui et al., 2024; Zeng et al., 2025). These applications require representations that capture both cell identity and molecular variation across conditions. How transcriptomic information is organized and compressed therefore matters for the range of tasks a pretrained model can support.

We consider two complementary capabilities. Cell understanding extracts biological meaning from observed transcriptomes, including cell identity and similarity across batches. Cell generation predicts molecular expression under a specified condition, such as a genetic perturbation (Cui et al., 2024; Roohani et al., 2024). Understanding benefits from a stable cell level summary, whereas generation requires access to coordinated expression changes within and across cell identities. Cells sharing an annotated identity can exhibit distinct response programs, so identity separation alone does not ensure informative conditioning for generation. Supporting both raises a representation challenge: how can an encoder preserve functional molecular information in a compact form that serves whole-cell comparison and structured response prediction?

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.

The first limitation lies in *how transcriptomes are tokenized*. Many single-cell foundation models adapt the language-modeling paradigm by treating genes as tokens and each cell as a sentence-like sequence, incorporating gene identity and expression through ranking or value embeddings (Theodoris et al., 2023; Cui et al., 2024; Hao et al., 2024). This design provides a flexible basis for learning gene dependencies, but leaves functional organization implicit. Transcriptomic responses involve coordinated molecular programs whose evidence is distributed across interacting genes; neither expression rank nor gene-wise token boundaries explicitly encode this organization. The encoder must infer these groupings from sparse observations while learning contextual interactions. The second limitation concerns *the interface exposed to downstream tasks*. A common strategy summarizes encoder outputs into a single cell-level embedding through pooling or a dedicated summary token (Theodoris et al., 2023; Cui et al., 2024; Hao et al., 2024; Rosen et al., 2026). This interface is effective for tasks such as cell classification and representation alignment, where global cell-level similarity is the primary objective. However, when the same summary is used for conditional expression prediction, diverse sources of molecular variation must be compressed into a single aggregate representation. Such compression may preserve cell identity while attenuating response-relevant variation within a cell type, forcing the decoder to recover fine-grained molecular structure from an entangled embedding. Retaining the complete gene-token sequence preserves finer-grained information, but makes downstream computation scale with the input dimensionality. These limitations highlight the need for a compact representation interface that not only summarizes cellular identity, but also preserves reusable latent components for structured decoding.

We introduce **scPathVQ**, which addresses these two limitations through a biology-structured tokenization scheme and a reusable discrete latent interface. To make functional organization explicit, scPathVQ constructs pathway-induced patches that aggregate genes according to biological pathways, while a complementary highly variable gene (HVG) view provides additional expression evidence through cross-view attention. To avoid collapsing the transcriptome into a single summary embedding, scPathVQ further introduces a latent-state bottleneck that transforms the patch tokens into a fixed-length sequence of latent components, following the latent-array principle of Perceiver (Jaegle et al., 2021). Each component aggregates evidence across multiple patches and retains a distinct aspect of the cellular state. Cell-state vector quantization (VQ) then maps these components to a shared codebook of reusable discrete prototypes while preserving their slot structure (Van Den Oord et al., 2017). Because quantization is performed on cell-contextualized components rather than individual genes or predefined pathways, the resulting prototypes capture reusable molecular states at an intermediate level of abstraction. This discrete representation provides structured latent components for downstream decoding, while gated pooling over the same components produces a compact continuous embedding for cell-level understanding.

We pretrain scPathVQ on 34 million human cells and evaluate four downstream tasks: annotation and multi-batch integration for cell understanding, together with perturbation prediction and cell state inference for cell generation. Cell state inference reads out functional states from generated post-perturbation expression, connecting gene-level prediction to perturbation-induced pathway responses. Among the evaluated baselines, scPathVQ achieves the highest annotation scores across four datasets and biological-conservation scores across five integration benchmarks, and leads five of six perturbation dataset–metric comparisons. Generated responses capture pathway-associated cell-state patterns and show agreement with observed pathway responses under genetic perturbations. Ablation studies further support the effectiveness of biology-structured tokenization and the discrete latent interface in learning informative cellular representations.

Our contributions are as follows:

- We identify the complementary requirements of cell understanding and cell generation, linking their joint support to explicit functional organization and compression that preserves multiple state components.
- We introduce scPathVQ, which integrates biology-structured tokenization, a latent bottleneck, and cell-state vector quantization to construct reusable discrete latent codes and derive a continuous cell embedding for cell understanding.
- We evaluate four downstream tasks, demonstrate leading performance on annotation, biological conservation, and perturbation prediction, and show that generated expression supports inference of pathway-associated cell states. Component ablations establish the contribution of pathway organization to representation quality.

2 Related Work

Large-scale pretraining has established gene expression as a transferable source of supervision for single-cell analysis. scBERT and Geneformer adapt masked-token modeling to ranked or binned gene inputs, while scGPT, scFoundation, xTrimoGene, CellPLM, CellFM, and TABULA extend the scale, input processing, or pretraining objectives (Yang et al., 2022; Theodoris et al., 2023; Cui et al., 2024; Hao et al., 2024; Gong et al., 2023; Wen et al., 2024; Zeng et al., 2025; Ding et al., 2026). These models support tasks such as annotation, integration, and perturbation-related prediction, but their computation remains largely attached to gene-level inputs or a dense cell-level representation. A complementary family makes a continuous cell embedding the primary interface for retrieval, integration, or classification, including SCimilarity, SATURN, and UCE (Heimberg et al., 2025; Rosen et al., 2024; 2026). scPathVQ addresses the interface between these views by retaining a continuous readout while also exposing a compact sequence for structured prediction.

Biological priors have been introduced through interpretable latent variables, structured attention, external gene representations, and ontology-based supervision. VEGA and TOSICA use gene modules, pathways, or regulons as structured units (Seninge et al., 2021; Chen et al., 2023); SATURN and sciLaMA incorporate protein or multimodal gene representations (Rosen et al., 2024; Hu et al., 2025); and GeneCompass, scCello, and RegFormer use regulatory, ontology, or hierarchical information during pretraining (Yang et al., 2024; Yuan et al., 2024; Hu et al., 2026). These approaches show that prior knowledge can improve the meaning, geometry, or supervision of learned representations. In scPathVQ, pathway annotations instead define the computational units before contextual integration: KEGG-induced patches provide fixed functional coordinates, while the HVG view supplies cell-specific context beyond curated coverage. The distinction is the role and position of the prior in the representation pipeline, rather than the use of pathway information alone.

Latent arrays and vector-quantized autoencoders provide complementary tools for constructing compact interfaces. Perceiver decouples the number of latent positions from the number of input elements, while vector quantization maps continuous states to a reusable codebook (Jaegle et al., 2021; Van Den Oord et al., 2017). Recent single-cell models have begun to explore discrete representations: scDiVa uses masked discrete diffusion for joint identity and expression modeling, and CellVQ maps a cell representation to codebook-based cell codes (Wang et al., 2026b;a). scPathVQ applies cell state VQ to multiple latent components after pathway-guided contextualization and bottleneck integration, rather than assigning a single code to a pooled cell state. The shared quantized sequence can therefore support structured response modeling, while gated pooling of the same components yields a continuous embedding for cell-level understanding.

3 Methodology

3.1 Problem Formulation

Let $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^{N}$ be a corpus of normalized single-cell expression profiles over the harmonized gene universe $\mathcal{G} = \{g_1, \ldots, g_G\}$, where $\mathbf{x}_i \in \mathbb{R}^G$ represents cell i. A single-cell foundation model learns an encoder $E_\theta : \mathbb{R}^G \to \mathbb{R}^d$ and a decoder $D_\theta : \mathbb{R}^d \to \mathbb{R}^G$, producing $\mathbf{z}_i = E_\theta(\mathbf{x}_i)$ and $\widehat{\mathbf{x}}_i = D_\theta(\mathbf{z}_i)$. Pretraining follows a masked gene modeling objective:

$$\mathcal{L}_{\mathrm{pre}}(\theta) = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\left[\mathcal{L}_{\mathrm{rec}}(\widehat{\mathbf{x}}, \mathbf{x})\right] + \sum_r \lambda_r \mathcal{L}_{\mathrm{aux}}^{(r)}, \tag{1}$$

where $\mathcal{L}_{\mathrm{rec}}$ measures reconstruction error, and $\mathcal{L}_{\mathrm{aux}}^{(r)}$ denotes auxiliary objectives tailored to specific architectures, which impose regularization or representation-level constraints. The pretrained encoder supports two downstream task families. For cell understanding, a task-specific predictor C_ϕ maps the representation of an observed cell to a semantic output in $\mathcal{Y}$:

$$f_{\mathrm{und}} : \mathbb{R}^G \to \mathcal{Y}, \qquad \widehat{y}_i = f_{\mathrm{und}}(\mathbf{x}_i) = C_\phi(E_\theta(\mathbf{x}_i)). \tag{2}$$

For cell generation, a conditional predictor G_ψ maps a source transcriptome $\mathbf{x}_i^{\mathrm{src}}$ and condition $\mathbf{a}_i \in \mathcal{A}$ to a target cell state:

$$f_{\mathrm{gen}} : \mathbb{R}^G \times \mathcal{A} \to \mathbb{R}^G, \qquad \widehat{\mathbf{x}}_i^{\mathrm{tgt}} = f_{\mathrm{gen}}(\mathbf{x}_i^{\mathrm{src}}, \mathbf{a}_i) = G_\psi(E_\theta(\mathbf{x}_i^{\mathrm{src}}), \mathbf{a}_i). \tag{3}$$

Cell state inference then characterizes pathway-associated functional states from the generated target expression, evaluating conditional generation at the program level.

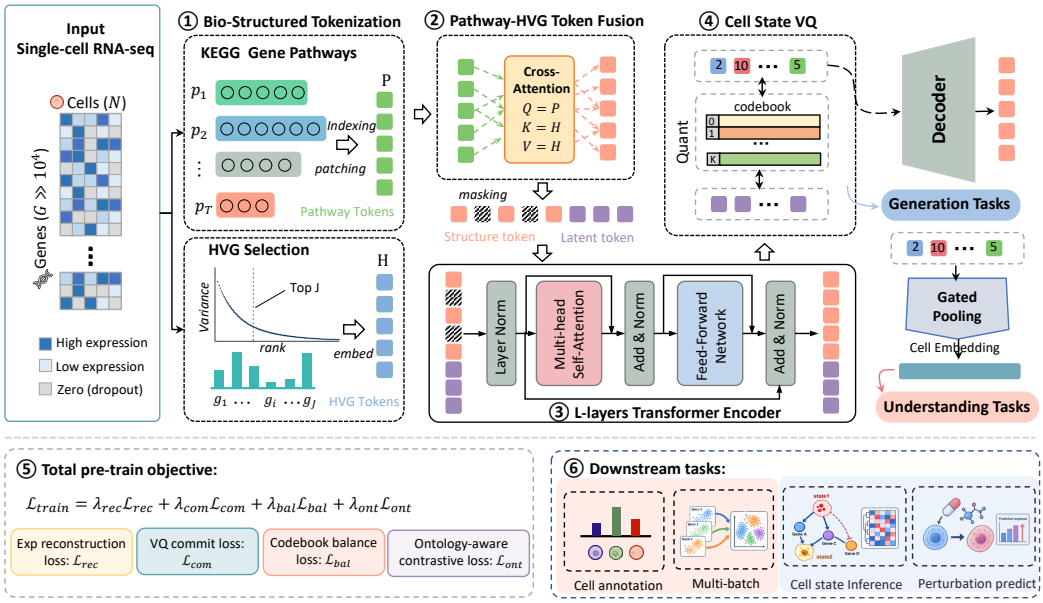


Figure 1: Overview of scPathVQ. scPathVQ first constructs pathway-induced patches and a fixed HVG view, then uses pathway-to-HVG cross-attention to contextualize the structured tokens. An L-layer Transformer with learnable latent tokens compresses them into a fixed-length sequence, whose final latent slots are quantized with a shared codebook. Pretraining combines masked expression reconstruction, VQ commitment, codebook-balance, and ontology-aware contrastive objectives. The quantized sequence supports perturbation prediction and functional cell state inference from generated expression, whereas gated pooling provides a continuous embedding for cell-type annotation and multi-batch integration.

3.2 Biology-Structured Tokenization

Gene expression is measured at individual genes, whereas cell states are shaped by coordinated molecular programs. A purely gene-wise tokenization leaves this organization implicit and allocates most input positions to sparse, isolated measurements. We instead construct two complementary views: pathway-induced patches form the structured backbone, while a corpus-level HVG view supplies cell-specific signals beyond curated pathway coverage before latent compression. Appendix A.1 provides the biological rationale and implementation details.

Pathway-induced patch tokens. Let $\mathcal{P} = \{\mathcal{P}_t\}_{t=1}^{T}$ denote a collection of gene pathways, where $\mathcal{P}_t \subseteq \mathcal{G}$ contains the genes associated with pathway t. Direct pathway-specific tokens would vary in length and duplicate genes because pathways differ in size and overlap. We instead fix the pathway and within-pathway gene orders, flatten the lists, and remove duplicates while retaining their first occurrence, yielding

$$\mathbf{g}' = \mathrm{uniq}(\mathrm{flatten}(\mathcal{P}_1, \ldots, \mathcal{P}_T)) = (g'_1, \ldots, g'_{G'}), \qquad G' = |\mathbf{g}'|. \tag{4}$$

We split $\mathbf{g}'$ into $P = \lceil G'/R_p \rceil$ contiguous, non-overlapping patches $\{p_m\}_{m=1}^{P}$ of R_p consecutive genes and pad the final patch with null positions. These patches serve as Transformer input tokens. Their patch and within-patch gene orders remain fixed across cells and define a reproducible coordinate system.

For cell i, let $\mathbf{X}_i^{\mathrm{p}} \in \mathbb{R}^{P \times R_p}$ contain one expression row per pathway-induced patch. A linear projection $\mathbf{W}_p \in \mathbb{R}^{d \times R_p}$ and learned patch-position embedding $\mathbf{W}_{\mathrm{pos}} \in \mathbb{R}^{P \times d}$ yield

$$\mathbf{P}_i = \mathbf{X}_i^{\mathrm{p}} \mathbf{W}_p^{\mathsf{T}} + \mathbf{W}_{\mathrm{pos}}, \qquad \mathbf{P}_i \in \mathbb{R}^{P \times d}. \tag{5}$$

Here, $\mathbf{W}_{\mathrm{pos}}$ encodes fixed patch coordinates rather than temporal order. We construct $\mathcal{P}$ from human KEGG pathway maps (Kanehisa & Goto, 2000; Kanehisa et al., 2025). Appendix A.1 lists the retained pathways and partition parameters.

HVG tokens. To retain cell-state signals beyond curated pathway coverage, we select a corpus-level panel $\mathcal{H} = \{h_j\}_{j=1}^{J}$ of the top-J highly variable genes. For cell i, the token for h_j combines gene identity and expression value:

$$\mathbf{h}_{ij} = \mathrm{LN}\big(E_{\mathrm{gene}}(h_j) + E_{\mathrm{value}}(x_{ih_j})\big), \qquad \mathbf{h}_{ij} \in \mathbb{R}^d. \tag{6}$$

The tokens form $\mathbf{H}_i = [\mathbf{h}_{ij}]_{j=1}^{J} \in \mathbb{R}^{J\times d}$. The gene embedding is shared with the expression decoder, and the fixed panel defines consistent data-driven coordinates across cells.

Selective cross-view fusion. Pathway-induced patches remain the encoder units, whereas HVGs provide data-driven context. We implement this asymmetry by letting each patch query the observed HVG tokens through multi-head cross-attention (Vaswani et al., 2017), with pre-normalization and a residual connection:

$$\mathbf{Z}_i = \mathrm{LN}(\mathbf{P}_i + \mathrm{MHA}(\mathrm{LN}(\mathbf{P}_i), \mathrm{LN}(\mathbf{H}_i), \mathrm{LN}(\mathbf{H}_i))), \qquad \mathbf{Z}_i \in \mathbb{R}^{P\times d}. \tag{7}$$

Here, MHA denotes multi-head attention and LN denotes layer normalization. This query direction preserves the pathway-induced coordinates while allowing each valid patch to incorporate HVG-derived, cell-specific variation before compression.

3.3 Learnable Latent Bottleneck

The cross-view fusion in Equation 7 produces P pathway-induced states per cell, whose number depends on the upstream gene grouping. To obtain a fixed-length interface without collapsing multiple functional components, we introduce S learnable latent tokens with $S \ll P$. Their roles are learned through attention to the fused cell states, following the latent-array principle (Jaegle et al., 2021). Let $\mathbf{M} \in \mathbb{R}^{S\times d}$ denote a learnable latent-token template, which is appended to the fused sequence of every cell:$\mathbf{Z}_i^{(0)} = [\mathbf{Z}_i; \mathbf{M}] \in \mathbb{R}^{(P+S)\times d}$.

An L-layer pre-normalized Transformer encoder (Vaswani et al., 2017) jointly updates the pathway tokens and latent positions. For layer $\ell = 1, \ldots, L$,

$$\mathbf{A}_i^{(\ell)} = \mathbf{Z}_i^{(\ell-1)} + \mathrm{MHA}_\ell\Big(\mathrm{LN}(\mathbf{Z}_i^{(\ell-1)})\Big), \quad \mathbf{Z}_i^{(\ell)} = \mathbf{A}_i^{(\ell)} + \mathrm{FFN}_\ell\Big(\mathrm{LN}(\mathbf{A}_i^{(\ell)})\Big). \tag{8}$$

After the final layer, we discard the pathway-induced states and retain the S latent positions as $\tilde{\mathbf{Z}}_i \in \mathbb{R}^{S\times d}$. This fixed-length sequence provides a contextual summary of multiple cell states and is passed to the vector-quantization module.

3.4 Cell-State Vector Quantization

The bottleneck produces S cell-contextualized state/slot. For cell i and slot s, let $\mathbf{z}_{is} = (\tilde{\mathbf{Z}}_i)_{s,:} \in \mathbb{R}^d$. We quantize each slot independently against a shared spherical codebook $\mathcal{C} = \{\mathbf{c}_k\}_{k=1}^{K} \subset \mathbb{R}^{d_q}$ of unit-norm prototypes, preserving the ordered slot structure (Van Den Oord et al., 2017). With $\mathbf{W}_q \in \mathbb{R}^{d_q\times d}$, cosine assignment, and a straight-through estimator,

$$\bar{\mathbf{z}}_{is} = \frac{\mathbf{W}_q\,\mathrm{LN}(\mathbf{z}_{is})}{\|\mathbf{W}_q\,\mathrm{LN}(\mathbf{z}_{is})\|_2}, \quad k_{is} = \arg\max_{k\in\{1,\ldots,K\}} \bar{\mathbf{z}}_{is}^{\top}\mathbf{c}_k, \quad \mathbf{q}_{is} = \bar{\mathbf{z}}_{is} + \mathrm{sg}(\mathbf{c}_{k_{is}} - \bar{\mathbf{z}}_{is}). \tag{9}$$

Here, $\mathrm{sg}(\cdot)$ stops gradients, so $\mathbf{q}_{is} = \mathbf{c}_{k_{is}}$ in the forward pass while gradients reach $\bar{\mathbf{z}}_{is}$. The indices $(k_{i1}, \ldots, k_{iS})$ form the discrete code, and $\mathbf{Q}_i = (\mathbf{q}_{i1}, \ldots, \mathbf{q}_{iS})$ is the corresponding prototype sequence retained for structured prediction. Codebook entries are updated by exponential moving averages of their assigned normalized latents. Implementation details are given in Appendix A.4.

Two auxiliary objectives regularize the assignments. The commitment loss keeps projected slots close to their selected prototypes:

$$\mathcal{L}_{\mathrm{com}} = \mathbb{E}_{i\sim\mathcal{D}}\left[\frac{1}{S}\sum_{s=1}^{S}\|\bar{\mathbf{z}}_{is} - \mathrm{sg}(\mathbf{c}_{k_{is}})\|_2^2\right]. \tag{10}$$

To discourage codebook collapse, we average temperature-τ_q soft assignments across cells and slots into $\bar{\boldsymbol{\pi}}$ and penalize deviation from uniform code usage:

$$\mathcal{L}_{\mathrm{bal}} = D_{\mathrm{KL}}\big(\bar{\boldsymbol{\pi}} \,\big\|\, K^{-1}\mathbf{1}_K\big). \tag{11}$$

Soft assignments affect only $\mathcal{L}_{\mathrm{bal}}$. The exposed sequence $\mathbf{Q}_i$ remains hard-quantized. Appendix A.4 gives the exact soft-assignment definition, quantization dimensions, codebook size, EMA decay, temperature, and training schedule.

3.5 PRETRAINING STRATEGY

We pretrain the discrete and continuous interfaces from the same post-VQ sequence $\mathbf{Q}_i$. The reconstruction objective preserves information at two resolutions. For cell i, let $\mathcal{T}_i$ contain the masked gene targets, $\hat{x}_g$ denote the uncorrupted expression target for gene g, and $v_{im} \in \{0, 1\}$ indicate whether pathway patch m is valid. We define

$$\mathcal{L}_{\text{rec}} = \mathbb{E}_{i\sim\mathcal{D}}\left[\underbrace{\sum_{g\in\mathcal{T}_i} -\log p_\theta(\hat{x}_g \mid \mathbf{Q}_i, g)}_{\mathcal{L}_{\text{gene}}} \quad + \underbrace{\sum_{m\in P} v_{im}\left\|\widehat{\mathbf{Z}}_{im} - \text{sg}(\mathbf{Z}_{im})\right\|^2}_{\mathcal{L}_{\text{state}}}\right]. \tag{12}$$

The gene term models masked expression, while the state term reconstructs valid fused pathway states with stop-gradient targets. Both decoders use only state-VQ representations, requiring the quantized sequence to preserve information at both resolutions. Appendix A.5 gives the likelihood factorization, decoder architectures, and expanded losses.

The quantized slots also provide a continuous readout for cell-level supervision and downstream understanding. Let $\mathbf{u}_{is} = \mathbf{W}_o\mathbf{q}_{is} + \mathbf{e}_s^{\text{lat}}$ combine the projected code vector with a learned slot embedding. A gated pooling layer computes

$$\alpha_{is} = \text{softmax}_s(\psi_{\text{gate}}(\mathbf{u}_{is})), \quad \mathbf{h}_{\text{cell},i} = \text{LN}\left(\sum_{s=1}^{S}\alpha_{is}\psi_{\text{val}}(\mathbf{u}_{is})\right). \tag{13}$$

The ordered sequence $\mathbf{Q}_i$ remains available to structured prediction heads, whereas cell-level heads use $\mathbf{h}_{\text{cell},i}$.

Ontology supervision organizes the pooled embedding when labels are available. Let $\mathcal{I}_{\text{lab}}$ index labeled cells in the distributed minibatch, y_i be the harmonized Cell Ontology identifier, and $\bar{\mathbf{h}}_i$ a normalized projection of $\mathbf{h}_{\text{cell},i}$. With $\mathcal{I}_i^+ = \{j \in \mathcal{I}_{\text{lab}} \setminus \{i\} : y_j = y_i\}$ and $\mathcal{I}_{\text{anc}} = \{i \in \mathcal{I}_{\text{lab}} : |\mathcal{I}_i^+| > 0\}$, the supervised contrastive objective is:

$$\mathcal{L}_{\text{ont}} = -\mathbb{E}_{i\sim\mathcal{I}_{\text{anc}}}\mathbb{E}_{j\sim\mathcal{I}_i^+}\left[\log\frac{\exp(\bar{\mathbf{h}}_i^\top\bar{\mathbf{h}}_j/\tau)}{\sum_{a\in\mathcal{I}_{\text{lab}}\setminus\{i\}}\exp(\bar{\mathbf{h}}_i^\top\bar{\mathbf{h}}_a/\tau)}\right], \tag{14}$$

Unlabeled cells are excluded only from $\mathcal{L}_{\text{ont}}$ and continue to optimize the reconstruction and VQ objectives. Appendix A.5 gives the contrastive-set construction and projection-head details.

Combining reconstruction with the commitment and balance losses in Equations 10 and 11 gives the complete objective

$$\mathcal{L}_{\text{pre}} = \lambda_{\text{rec}}\mathcal{L}_{\text{rec}} + \lambda_{\text{com}}\mathcal{L}_{\text{com}} + \lambda_{\text{bal}}\mathcal{L}_{\text{bal}} + \lambda_{\text{ont}}\mathcal{L}_{\text{ont}}. \tag{15}$$

$\mathcal{L}_{\text{rec}}$ preserves decoded information, $\mathcal{L}_{\text{com}}$ and $\mathcal{L}_{\text{bal}}$ regularize discretization, and $\mathcal{L}_{\text{ont}}$ structures the pooled cell space. All coefficients remain fixed throughout pretraining, with no loss-weight warm-up. Appendix A.5 reports their values and selection protocol together with the corruption and optimization settings.

4 EXPERIMENTS

We evaluate scPathVQ on four tasks spanning cell understanding and cell generation. The pooled embedding supports cell-type annotation and multi-batch integration. For cell generation, perturbation prediction measures gene-level response accuracy, and cell state inference assesses functional states and pathway responses derived from generated expression. Component ablations test the contribution of each representation module. Dataset descriptions, evaluation protocols, and extended analyses are provided in the Appendix.

4.1 CELL TYPE ANNOTATION

Table 1: Cell-type annotation results on 4 datasets. The superscripts (s) and (f) denote the train-from-scratch and full-fine-tuning settings, respectively. Values are mean ± std over 5 seeds.

Model	PBMC68K		Liver cancer		Lung		Melanoma	
	Macro-F1↑	Accuracy↑	Macro-F1↑	Accuracy↑	Macro-F1↑	Accuracy↑	Macro-F1↑	Accuracy↑
SVM	0.746±0.037	0.789±0.021	0.729±0.025	0.764±0.052	0.816±0.029	0.854±0.018	0.685±0.061	0.733±0.031
scmap	0.796±0.062	0.815±0.019	0.753±0.032	0.786±0.103	0.825±0.107	0.874±0.082	0.709±0.053	0.758±0.037
scBERT	0.846±0.026	0.875±0.042	0.829±0.043	0.874±0.027	0.893±0.036	0.924± 0.056	0.768±0.037	0.837±0.053
scGPT	0.862±0.031	0.891±0.024	0.847±0.038	0.889±0.021	0.912±0.041	0.941±0.033	0.785±0.045	0.852±0.063
Geneformer	0.859±0.034	0.897±0.059	0.855±0.031	0.876±0.025	0.891±0.038	0.921±0.032	0.795±0.047	0.854±0.128
scFoundation	0.885±0.012	0.901±0.015	0.902±0.021	0.935±0.018	0.935±0.031	0.966±0.013	0.792±0.054	0.879±0.022
CellFM	0.896±0.026	0.904±0.038	0.905±0.042	0.934±0.026	0.938±0.028	0.950±0.073	0.798±0.081	0.883±0.035
UCE	0.887±0.014	0.912±0.016	0.888±0.024	0.942±0.019	0.945±0.035	0.948±0.012	0.802±0.058	0.887±0.023
scPathVQ$^{(s)}$	0.878±0.013	0.914±0.018	0.895±0.022	0.934±0.017	0.938±0.034	0.959±0.014	0.798±0.056	0.886±0.021
scPathVQ$^{(f)}$	**0.926±0.016**	**0.945±0.009**	**0.923±0.022**	**0.958±0.017**	**0.963±0.032**	**0.991±0.005**	**0.855±0.020**	**0.932±0.016**

To evaluate whether the pooled embedding $\mathbf{h}_{\text{cell}}$ supports cell understanding, we perform cell-type annotation on four tissue and disease datasets. We compare zero-shot inference, the corresponding architecture trained from scratch, and full fine-tuning, measuring model generalization, the benefit of pretraining, and task-specific adaptation, respectively. We report Macro-F1 for class-balanced evaluation and accuracy as a complementary metric. Detailed dataset and evaluation protocols are provided in Appendix C.1.

With full fine-tuning, scPathVQ consistently achieves the highest Macro-F1 and accuracy across all four datasets, resulting in the best mean performance (Table 1). On melanoma, it improves over the next-best method by **5.3%** in Macro-F1 and **4.5%** in accuracy. The corresponding margins on PBMC68K are 3.0 and 3.1 points, with scPathVQ also achieving the best results on liver cancer and lung. Moreover, the frozen linear probe attains the highest mean Macro-F1 across all datasets without updating the backbone (Figure 2). These results show that the pooled representation learned by scPathVQ captures cell-state information that remains effective under both frozen and fine-tuned settings.

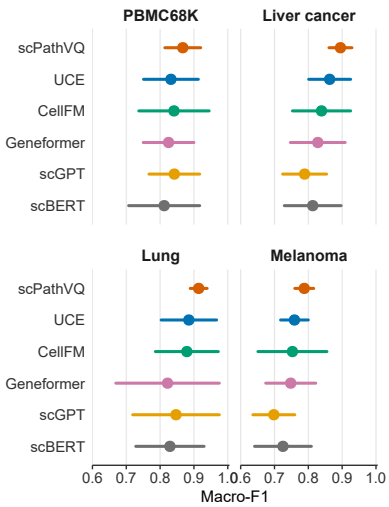


Figure 2: Zero-shot cell annotation in Macro-F1. Points and bars show mean ± std over five seeds.

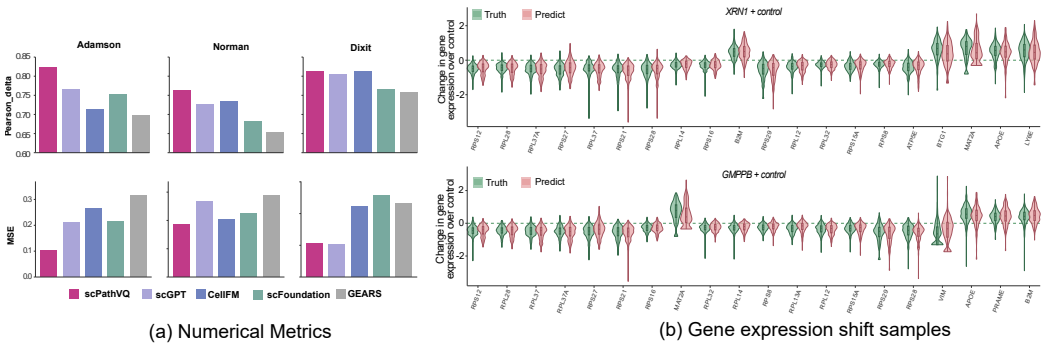
4.2 Perturbation Prediction

To assess scPathVQ for cell generation, we test whether the post-quantization sequence $\mathbf{Q}$ preserves biologically meaningful response patterns when predicting responses to perturbation identities held out during downstream training on Adamson, Norman, and Dixit. With the backbone frozen, a task-specific predictor combines $\mathbf{Q}$ with a pretrained gene or gene-pair embedding to estimate control-relative expression changes. We report condition-level Delta Pearson for gene-wise response patterns and MSE for prediction magnitude. Full protocol details are provided in Appendix C.2.

scPathVQ achieves the best value in five of the six dataset–metric comparisons (Figure 3(a)), leading both metrics on Adamson (Delta Pearson 0.823, MSE 0.102) and Norman (0.764, 0.186), including combinatorial perturbations. On Dixit, it attains the highest Delta Pearson (0.814), while its MSE (0.105) remains close to the best value (0.103). In two representative Adamson examples, the predicted profiles recover the overall direction and central tendency of control-relative shifts among the top differentially expressed genes, despite differences in individual magnitudes (Figure 3(b)). Overall, these results show that $\mathbf{Q}$ preserves response-relevant structure for predicting transcriptional responses to held-out perturbations.

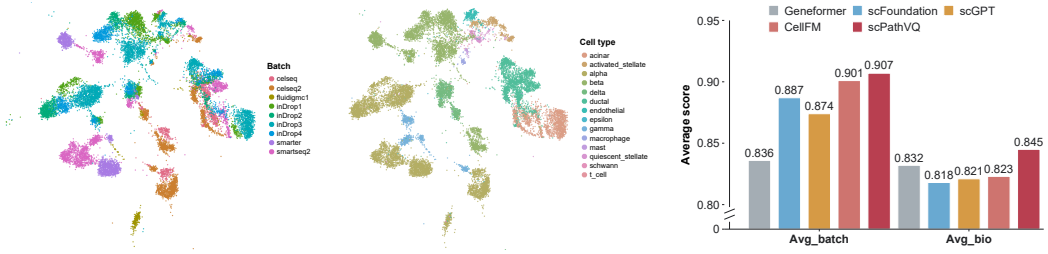
4.3 Multi-Batch Integration

We evaluate whether the pooled embedding $\mathbf{h}_{\text{cell}}$ preserves cell identity while reducing batch variation across five benchmarks, comparing scPathVQ with Harmony and four single-cell foundation



(a) Numerical Metrics (b) Gene expression shift samples

Figure 3: Perturbation prediction results. (a) Comparison against other models using Δ Pearson and MSE metrics. (b) Predicted vs. observed expression shifts in the Adamson dataset. Distributional changes for the top 20 differentially expressed genes under *XRN1* and *GMPPB* perturbations.



(a) Pancreas: batch (b) Pancreas: cell type (c) Metric comparison

Figure 4: Multi-batch integration on Pancreas. (a,b) PCA projections of the same scPathVQ cell embeddings, colored by acquisition batch and cell type, respectively, using shared coordinates. (c) Comparison with Geneformer, scFoundation, scGPT, and CellFM in batch mixing (Avg_batch) and biological conservation (Avg_bio). Higher values indicate better performance for both metrics.

models. Protocols, full comparison results, and extended visualizations for additional datasets appear in Appendices C.3 and D.3 (Table 8; Figure 12). scPathVQ leads 9/10 metric comparisons, ranking first in biological conservation on all five datasets and batch mixing on four. On Lung, batch mixing trails CellFM (0.924 versus 0.928), while biological conservation remains highest (0.819). On Pancreas, scPathVQ achieves the best batch-mixing and biological-conservation scores of 0.907 and 0.845, respectively (Figure 4(c)). The corresponding PCA projections show coherent cell populations and central batch overlap, with some batch-enriched peripheral regions (Figure 4(a,b)). These results support improved batch mixing with preserved biological identity.

4.4 Cell State Inference

We evaluate whether generated profiles preserve perturbation-induced functional programs at the condition and single-cell levels. At the condition level, we compare expression-derived pathway-response scores between predicted and measured profiles (Appendix C.4). Across all 33,708 evaluable pathway–condition pairs from held-out Norman perturbations, Pearson and Spearman correlations reach 0.700 and 0.641, respectively. In the subset shown in Figure 5, predictions generally recover positive responses in porphyrin metabolism and negative responses in the proteasome and cysteine and methionine metabolism pathways, with differences in magnitude.

At single-cell resolution, we examine held-out CRISPR activation of *BAK1*, a pro-apoptotic BCL-2-family effector, in Norman (Norman et al., 2019). scPathVQ generates post-perturbation expression from each source cell's quantized representation, then re-encodes generated and measured profiles through the pathway-induced tokenizer and encoder to obtain pathway-associated attention scores. In shared UMAP coordinates, predictions reproduce the broad spatial patterns observed in measured cells: cell-cycle and senescence states localize primarily to the left, apoptosis and ferroptosis to the lower right, and necroptosis to the upper right (Figure 6). Together, the quantitative pathway-response agreement and qualitative cell-state patterns support the preservation of functional information through the quantized representation for perturbation-conditioned generation.

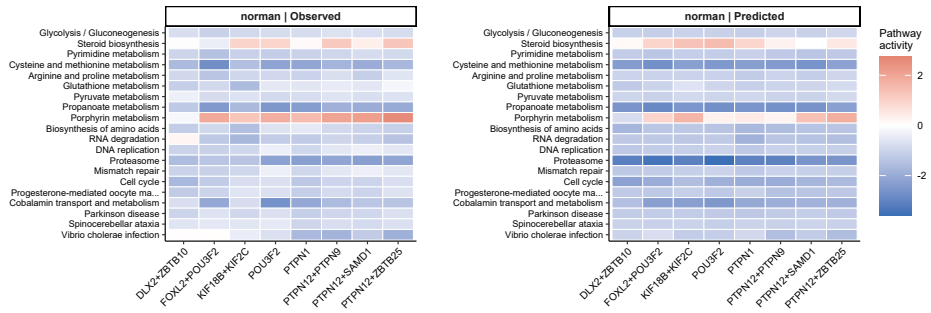


Figure 5: Observed (left) and predicted (right) pathway responses across held-out Norman perturbations. Rows show 20 selected pathways and columns show eight perturbation conditions. A shared color scale indicates positive (red) and negative (blue) relative responses.

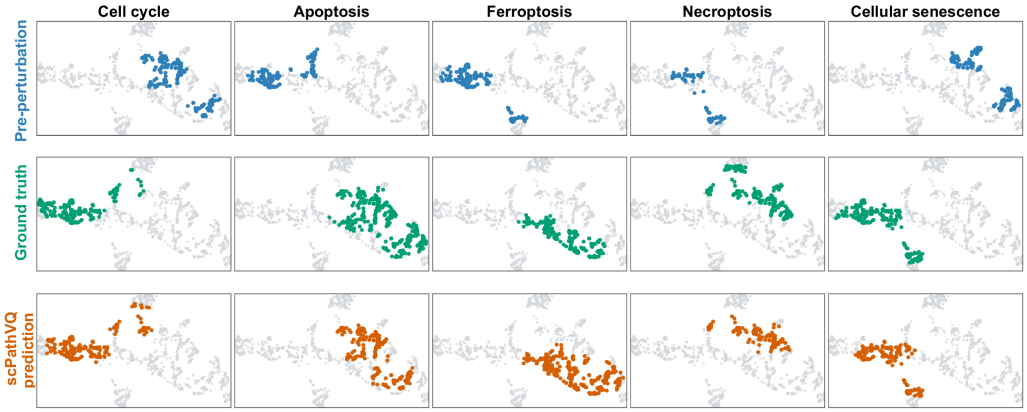


Figure 6: Pathway-associated cell states under held-out *BAK1* activation in Norman. Rows show pre-perturbation cells, measured post-perturbation cells, and scPathVQ predictions in shared UMAP coordinates. Colored points indicate elevated pathway attention relative to fixed controls.

4.5 Ablation Study

Table 2: Component ablations for scPathVQ (Macro-F1).

Configuration	PBMC68K	Liver cancer	Lung	Melanoma	Avg. drop
scPathVQ	**0.926**	**0.923**	**0.963**	**0.855**	–
w/o pathway tokens	0.885	0.879	0.924	0.817	4.4%
w/o HVG tokens	0.902	0.898	0.932	0.836	2.7%
w/o latent bottleneck	0.894	0.886	0.927	0.841	3.2%
w/o VQ	0.912	0.904	0.936	0.832	2.3%
w/o ontology loss	0.914	0.907	0.931	0.847	1.8%

We evaluate component contributions through cell-type annotation ablations (Table 2). Every ablation reduces Macro-F1 across all four datasets. Removing pathway tokens produces the largest mean relative decrease (4.4%), followed by ablations of the latent bottleneck (3.2%), HVG tokens (2.7%), VQ (2.3%), and ontology loss (1.8%). These results support integrating pathway structure and complementary expression context before latent compression and quantization. Appendix A.3 examines the effect of latent-token count (Table 3), while Appendix E provides additional pathway, attention, and input-occlusion analyses.

5 Conclusion

In this paper, we introduced scPathVQ, which organizes gene expression around biological programs before compression, producing a quantized sequence for cell generation and a pooled embedding for cell understanding. Experiments demonstrate its effectiveness across cell annotation, integration, held-out perturbation prediction, and cell state inference. Ablation studies further support the contributions of pathway-guided tokenization and the discrete latent interface. Agreement between predicted and observed pathway responses, together with complementary single-cell state patterns, connects expression generation to functional interpretation. These findings support biologically structured compression as a reusable interface for preserving cellular identity and perturbation-associated functional information in single-cell understanding and response modeling.

ETHICS STATEMENT

This study uses previously released single-cell transcriptomic datasets and does not involve new participant recruitment or biological sample collection. We analyze these datasets for research purposes under their documented access conditions and do not attempt to re-identify individual donors. We have read and adhere to the ICLR Code of Ethics.

REPRODUCIBILITY STATEMENT

Appendix A describes the model architecture, training objectives, optimization procedure, and implementation details. Appendix B identifies the data sources and processing steps, while Appendix C specifies the downstream protocols, data splits, metrics, and statistical definitions. Code, configuration files, and scripts required to reproduce the reported experiments will be released in an anonymized repository at `https://anonymous.4open.science/r/scPathVQ-03B2`.

AI USE STATEMENT

Generative AI tools were used to assist with manuscript drafting and language editing. All AI-assisted text was reviewed, revised, and verified by the authors. The authors take responsibility for the final content of this submission, including its text, claims, citations, and artifacts.

REFERENCES

Britt Adamson, Thomas M Norman, Marco Jost, Min Y Cho, James K Nuñez, Yuwen Chen, Jacqueline E Villalta, Luke A Gilbert, Max A Horlbeck, Marco Y Hein, et al. A multiplexed single-cell crispr screening platform enables systematic dissection of the unfolded protein response. *Cell*, 167(7):1867–1882, 2016.

Jiawei Chen, Hao Xu, Wanyu Tao, Zhaoxiong Chen, Yuxuan Zhao, and Jing-Dong J Han. Transformer for one stop interpretable cell type annotation. *Nature Communications*, 14(1):223, 2023.

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.

Alexander D Diehl, Terrence F Meehan, Yvonne M Bradford, Matthew H Brush, Wasila M Dahdul, David S Dougall, Yongqun He, David Osumi-Sutherland, Alan Ruttenberg, Sirarat Sarntivijai, et al. The cell ontology 2016: enhanced content, modularization, and ontology interoperability. *Journal of biomedical semantics*, 7(1):44, 2016.

Jiayuan Ding, Jianhui Lin, Shiyu Jiang, Yixin Wang, Ziyang Miao, Zhaoyu Fang, Jiliang Tang, Min Li, and Xiaojie Qiu. Tabula: A tabular self-supervised foundation model for single-cell transcriptomics. *Advances in Neural Information Processing Systems*, 38:103789–103808, 2026.

Atray Dixit, Oren Parnas, Biyu Li, Jenny Chen, Charles P Fulco, Livnat Jerby-Arnon, Nemanja D Marjanovic, Danielle Dionne, Tyler Burks, Raktima Raychowdhury, et al. Perturb-seq: dissecting molecular circuits with scalable single-cell rna profiling of pooled genetic screens. *cell*, 167(7): 1853–1866, 2016.

Jing Gong, Minsheng Hao, Xingyi Cheng, Xin Zeng, Chiming Liu, Jianzhu Ma, Xuegong Zhang, Taifeng Wang, and Le Song. xtrimogene: an efficient and scalable representation learner for single-cell rna-seq data. *Advances in Neural Information Processing Systems*, 36:69391–69403, 2023.

Sonja Hänzelmann, Robert Castelo, and Justin Guinney. Gsva: gene set variation analysis for microarray and rna-seq data. *BMC bioinformatics*, 14(1):7, 2013.

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.

Graham Heimberg, Tony Kuo, Daryle J DePianto, Omar Salem, Tobias Heigl, Nathaniel Diamant, Gabriele Scalia, Tommaso Biancalani, Shannon J Turley, Jason R Rock, et al. A cell atlas foundation model for scalable search of similar human cells. *Nature*, 638(8052):1085–1094, 2025.

Hongru Hu, Shuwen Zhang, Yongin Choi, Venkat S Malladi, and Gerald Quon. scilama: A single-cell representation learning framework to leverage prior knowledge from large language models. *bioRxiv*, 2025.

Luni Hu, Hua Qin, Yilin Zhang, Yi Lu, Ping Qiu, Zhihan Guo, Lei Cao, Wenjian Jiang, Yixin Shen, Qianqian Chen, et al. Regformer: a single-cell foundation model powered by gene regulatory hierarchies. *Nature Communications*, 2026.

Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *International conference on machine learning*, pp. 4651–4664. PMLR, 2021.

Minoru Kanehisa and Susumu Goto. Kegg: kyoto encyclopedia of genes and genomes. *Nucleic acids research*, 28(1):27–30, 2000.

Minoru Kanehisa, Miho Furumichi, Yoko Sato, Yuriko Matsuura, and Mari Ishiguro-Watanabe. Kegg: biological systems database as a model of the real world. *Nucleic acids research*, 53(D1): D672–D677, 2025.

Nayoung Kim, Hong Kwan Kim, Kyungjong Lee, Yourae Hong, Jong Ho Cho, Jung Won Choi, Jung-Il Lee, Yeon-Lim Suh, Bo Mi Ku, Hye Hyeon Eum, et al. Single-cell rna sequencing demonstrates the molecular and cellular reprogramming of metastatic lung adenocarcinoma. *Nature communications*, 11(1):2285, 2020.

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

Lichun Ma, Maria O Hernandez, Yongmei Zhao, Monika Mehta, Bao Tran, Michael Kelly, Zachary Rae, Jonathan M Hernandez, Jeremy L Davis, Sean P Martin, et al. Tumor cell biodiversity drives microenvironmental reprogramming in liver cancer. *Cancer cell*, 36(4):418–430, 2019.

Thomas M Norman, Max A Horlbeck, Joseph M Replogle, Alex Y Ge, Albert Xu, Marco Jost, Luke A Gilbert, and Jonathan S Weissman. Exploring genetic interaction manifolds constructed from rich single-cell phenotypes. *Science*, 365(6455):786–793, 2019.

Aditya Pratapa, Amogh P Jalihal, Jeffrey N Law, Aditya Bharadwaj, Murali, and TM. Benchmarking algorithms for gene regulatory network inference from single-cell transcriptomic data. *Nature methods*, 17(2):147–154, 2020.

Yusuf Roohani, Kexin Huang, and Jure Leskovec. Predicting transcriptional outcomes of novel multigene perturbations with gears. *Nature biotechnology*, 42(6):927–935, 2024.

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.

Yanay Rosen, Yusuf Roohani, Ayush Agrawal, Leon Samotorčan, Tabula Sapiens Consortium, Stephen R Quake, and Jure Leskovec. Universal cell embedding provides a foundation model for cell biology. *Nature*, pp. 1–9, 2026.

Lucas Seninge, Ioannis Anastopoulos, Hongxu Ding, and Joshua Stuart. Vega is an interpretable generative model for inferring biological network activity in single-cell transcriptomics. *Nature communications*, 12(1):5684, 2021.

Aravind Subramanian, Pablo Tamayo, Vamsi K Mootha, Sayan Mukherjee, Benjamin L Ebert, Michael A Gillette, Amanda Paulovich, Scott L Pomeroy, Todd R Golub, Eric S Lander, et al. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the national academy of sciences*, 102(43):15545–15550, 2005.

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.

Itay Tirosh, Benjamin Izar, Sanjay M Prakadan, Marc H Wadsworth, Daniel Treacy, John J Trombetta, Asaf Rotem, Christopher Rodman, Christine Lian, George Murphy, et al. Dissecting the multicellular ecosystem of metastatic melanoma by single-cell rna-seq. *Science*, 352(6282):189–196, 2016.

Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

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.

Jue Wang, Cheng Tan, Zhangyang Gao, Sida Shao, Shiping Liu, and Stan Z Li. Illuminating cell states by a comprehensive and interpretable single cell foundation model. *Nature Communications*, 17(1):4037, 2026a.

Mingxuan Wang, Cheng Chen, Gaoyang Jiang, Zijia Ren, Chuangxin Zhao, Lu Shi, and Yanbiao Ma. Scdiva: Masked discrete diffusion for joint modeling of single-cell identity and expression. *arXiv preprint arXiv:2602.03477*, 2026b.

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. In *International Conference on Learning Representations*, volume 2024, pp. 5649–5673, 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.

Xiaodong Yang, Guole Liu, Guihai Feng, Dechao Bu, Pengfei Wang, Jie Jiang, Shubai Chen, Qinmeng Yang, Hefan Miao, Yiyang Zhang, et al. Genecompass: deciphering universal gene regulatory mechanisms with a knowledge-informed cross-species foundation model. *Cell Research*, 34 (12):830–845, 2024.

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

Yuansong Zeng, Jiancong Xie, Ningyuan Shangguan, Zhuoyi Wei, Wenbing Li, Yun Su, Shuangyu Yang, Chengyang Zhang, Jinbo Zhang, Nan Fang, et al. Cellfm: a large-scale foundation model pre-trained on transcriptomics of 100 million human cells. *Nature Communications*, 16(1):4679, 2025.

Grace XY Zheng, Jessica M Terry, Phillip Belgrader, Paul Ryvkin, Zachary W Bent, Ryan Wilson, Solongo B Ziraldo, Tobias D Wheeler, Geoff P McDermott, Junjie Zhu, et al. Massively parallel digital transcriptional profiling of single cells. *Nature communications*, 8(1):14049, 2017.

A Additional Methodological Details

A.1 Rationale and Construction of Pathway-Induced Tokenization

Biological definition and organizational role. A gene pathway is a curated description of a biological process or signaling program in terms of the genes, gene products, and reactions that jointly participate in that program. For transcriptomic modeling, we represent pathway t by a gene set $\mathcal{P}_t \subseteq \mathcal{G}$ and define the corresponding incidence relation by $A_{tg} = \mathbb{1}[g \in \mathcal{P}_t]$. A pathway collection is therefore an overlapping family of sets rather than a partition of the genome: a gene can participate in several processes, and distinct pathways can share modules or signaling components. This overlap is biologically meaningful because cellular phenotypes are generated by coupled programs

rather than by independent genes. Set-level methods such as GSEA and GSVA use this organization to detect coordinated changes that may be weak or inconsistent at the level of any single gene (Subramanian et al., 2005; Hänzelmann et al., 2013). In single-cell models, curated network or pathway structure has likewise been used to obtain interpretable program-level representations (Seninge et al., 2021; Chen et al., 2023). We use pathway membership as an inductive bias for organizing computation; it is not treated as a complete causal model or as a direct estimate of pathway activity.

Why organization is needed for single-cell inputs. For a cell i, the measured transcriptome is a vector $\mathbf{x}_i \in \mathbb{R}^G$, but the coordinates do not form a sequence with a biological syntax. Ordering genes by expression produces a cell-dependent serialization, whereas alphabetical, genomic, or vocabulary order is computationally convenient but does not imply functional proximity. Once positional encodings are added, either choice can make the ease of learning a dependency depend on the serialization. The problem is compounded by the dimensionality and sparsity of atlas-scale data. In our pretraining corpus, $N = 34{,}017{,}503$ cells are represented in an active vocabulary of $G = 9{,}159$ genes. The stored expression matrix contains $65{,}721{,}557{,}551$ nonzero entries, giving

$$s = 1 - \frac{\mathrm{nnz}(\mathbf{X})}{NG} = 0.7891, \qquad \frac{\mathrm{nnz}(\mathbf{X})}{N} = 1{,}931.99 \text{ nonzero genes per cell on average.} \quad (16)$$

Thus, 78.91% of stored gene–cell entries are zero and only 21.09% are nonzero on average. A full-vocabulary gene-token implementation would expose 9,159 sequence positions for every cell. The pathway view exposes $P = 158$ patch positions, a $9{,}159/158 = 57.97$-fold reduction in sequence length. For a hypothetical self-attention layer whose cost is quadratic in the number of positions, the corresponding nominal pairwise-interaction count is reduced by approximately $(9{,}159/158)^2 = 3.36 \times 10^3$. This nominal comparison isolates the structured pathway branch; the complete dual-view input additionally contains 2,000 individual HVG context tokens, so the model processes 158 pathway queries and 2,000 context positions before cross-view fusion. The reduction is therefore a reduction in contextual-token length, not a deletion of the gene values: the expression values of the mapped genes are still used inside their assigned patches.

The zero-heavy regime also motivates a group-level unit. For a full patch with $R_p = 50$ real members, let D_{im} be the number of detected, nonzero genes in patch m for cell i. Using the corpus-wide marginal detection fraction $q = 1 - s = 0.2109$ as a simple reference, $\mathbb{E}[D_{im}] \approx 50q = 10.55$. Under an independence approximation, the probability that all 50 positions are zero would be $(1-q)^{50} \approx 7.2 \times 10^{-6}$, so a full patch usually contains at least one detected transcript even when individual genes are frequently undetected. The numerical probability is only an illustrative calculation because genes in a pathway are correlated and zero values mix biological absence with sampling and technical non-detection. Its role is to show the statistical intuition: a program-level token can pool several partially redundant observations, making the representation less dependent on any one dropout-prone coordinate. For example, the KEGG T-cell-receptor signaling map contains multiple components of the same program, including CD3D, CD3E, CD247, LCK, and ZAP70 (Kanehisa et al., 2025). A cell in which one of these transcripts is zero can still provide program evidence through the other observed components; the patch association metadata preserves the relevant pathway label even if the genes fall on opposite sides of a patch boundary.

Expected benefit for large-scale pretraining. Pathway-induced units provide three aligned benefits. First, they create semantic locality: the same patch coordinate refers to a fixed neighborhood in the pathway-organized gene layout for every cell. The fixed patch coordinates allow the Transformer layers and downstream heads to reuse contextual patterns across cells instead of relearning relationships under a cell-specific gene order. Second, they provide a denoising inductive bias. Jointly projecting the expression values within each patch allows reconstruction to use coordinated evidence from several members of a program when individual gene values are masked. Third, the shorter sequence makes contextualization and subsequent latent compression feasible on tens of millions of cells. Because pathways overlap, deduplicating the physical gene slots does not discard multi-pathway membership: each patch retains an association vector over the full pathway collection, and a gene can contribute to the metadata of every pathway that contains it. These properties make pathway organization useful before cross-view attention and latent quantization, while the fixed HVG view retains salient signals that are not represented by the curated collection.

Choice and representativeness of KEGG. We use the human KEGG pathway collection as the reference source because it provides a long-standing, curated representation of biological systems

with explicit human gene memberships and broad coverage of core metabolic, signaling, cellular, and disease-related programs (Kanehisa & Goto, 2000; Kanehisa et al., 2025). KEGG is also a common reference for gene-set enrichment and variation analyses, which makes the resulting units recognizable and independently interpretable (Subramanian et al., 2005; Hänzelmann et al., 2013). The choice is intended to provide a single, reproducible source of functional organization, not to claim that KEGG is exhaustive or uniquely correct. In the active vocabulary used here, the filtered KEGG union contains 7,895 genes, so its coverage is $7{,}895/9{,}159 = 86.20\%$. The remaining 1,264 genes are retained through the complementary 2,000-gene HVG panel; the counts imply 736 genes shared by the KEGG union and the HVG panel, corresponding to 36.80% of the HVGs and 9.32% of the KEGG-covered genes. This coverage profile is broad enough to anchor the majority of the input while leaving an explicit data-driven route for genes outside KEGG.

Deterministic construction. Let $\mathcal{P} = \{\mathcal{P}_t\}_{t=1}^{T}$ denote the filtered human pathway collection. We first harmonize pathway gene identifiers with the fixed model vocabulary and remove entries that do not survive the preprocessing filters. The resulting collection contains $T = 341$ pathways and $G' = 7{,}895$ unique mapped genes. Because the pathways overlap, this union count is not an average pathway size. We then fix the pathway order and the within-pathway gene order, flatten the lists, and remove duplicate genes while preserving the first occurrence. This produces a deterministic sequence $\mathbf{g}' = (g'_1, \ldots, g'_{G'})$ that is shared by all cells. The sequence is partitioned into contiguous, non-overlapping patches

$$p_m = \big(g'_{(m-1)R_p+1}, \ldots, g'_{\min(mR_p, G')}\big), \qquad m = 1, \ldots, P, \qquad P = \left\lceil \frac{G'}{R_p} \right\rceil, \tag{17}$$

with $R_p = 50$. Since $G' = 7{,}895$, we obtain $P = 158$ patches and append five null positions to the final patch so that every patch has the same tensor shape. A patch is therefore a fixed-width computational unit, not necessarily a complete pathway. For traceability, we retain the binary patch–pathway support $B_{mt} = \mathbb{1}[p_m \cap \mathcal{P}_t \neq \varnothing]$, which records every KEGG pathway intersecting the patch even though each mapped gene has only one physical owner slot after deduplication. The tokenizer also stores a normalized composition vector $\boldsymbol{\pi}_m \in \mathbb{R}^T$, whose entries are

$$\pi_{mt} = \frac{\sum_{g \in p_m} A_{tg}}{\sum_{t'=1}^{T} \sum_{g \in p_m} A_{t'g}}, \qquad B_{mt} = \mathbb{1}[\pi_{mt} > 0], \tag{18}$$

when the denominator is nonzero, and zero otherwise. For cell i, the expression values in p_m are projected jointly with the patch identity and the linear expression projection in Equation 5; the implementation additionally adds the composition embedding, summary statistics of the patch expression values, and a learned positional embedding. The position encodes this fixed computational coordinate system, not a temporal order.

The fixed HVG panel is selected once from the pretraining corpus and remains an individual-token view. Genes shared by KEGG and the HVG panel use the same underlying expression values in both views, while HVG-only genes are not assigned to a fabricated pathway. Pathway-induced patches are used as queries and HVG tokens as cell-specific keys and values in the cross-view block described in Appendix A.2. This separation preserves the functional prior where it is supported by annotation and preserves data-driven variation where annotation coverage is incomplete.

A.2 HVG View and Cross-View Attention

The HVG view complements the curated pathway view with a data-driven coordinate system. Highly variable genes are ranked by expression variability across the pretraining cells rather than by pathway membership, and therefore retain state-, lineage-, or condition-related signals that may be absent from, or only weakly represented by, the KEGG collection. This design follows the practical use of a bounded HVG panel with Seurat-style preprocessing in scGPT (Cui et al., 2024). We keep the panel fixed across cells and datasets: position j always denotes the same gene, the input length is bounded, and the resulting coordinates can be shared by the gene embedding and decoder. Each token combines gene identity with its expression value.

We construct the panel once from the training corpus and reuse it without modification for validation or downstream data. For each training dataset, at most 10,000 cells are sampled with a deterministic dataset-aware hash; Scanpy's Seurat v3 procedure then produces a within-dataset HVG rank,

which is converted to a percentile to make scores comparable across datasets. We average each gene's percentile over the training datasets and select the $J = 2{,}000$ genes with the smallest mean percentile, breaking ties by gene identifier. Validation datasets are excluded from this aggregation to avoid using held-out data for feature selection. The value $J = 2{,}000$ fixes a bounded context of individual-gene tokens while retaining a broad data-driven complement to the $P = 158$ pathway-induced positions; it is a computational and coverage compromise rather than an assertion that this value is universally optimal.

Cross-view attention is deliberately asymmetric. Each pathway-induced patch is a structured query, whereas the HVG tokens supply cell-specific keys and values, as formalized in Equation 7. The query direction preserves the pathway coordinate system and the fixed output length of P functional positions, while allowing every position to retrieve expression evidence from the wider transcriptome, including HVG-only genes. Thus, cross-view fusion conditions a shared functional unit on the current cell instead of concatenating two homogeneous sequences or compressing the HVG view independently. The corruption mask is applied before the two views are gathered, so a shared gene is masked consistently in both views. Applying the fusion before the latent bottleneck lets each pathway position acquire cell-specific context before the sequence is reduced to the S latent slots; the attention weights are used as contextual routing coefficients, not as estimates of pathway activity or causal influence.

A.3 Learnable Latent Bottleneck

The bottleneck is motivated by the object that the model must expose: a reusable cell state rather than a long list of input tokens. After pathway–HVG fusion, each cell is represented by $P = 158$ contextualized functional units. Retaining all of them would tie the downstream interface to the tokenizer's granularity, whereas pooling them into one vector would mix distinct state factors before the model can discretize or decode them. Perceiver motivates the use of a fixed learned latent array to bridge a large structured input and a compact interface (Jaegle et al., 2021). scPathVQ adopts this interface principle rather than the Perceiver computation graph: a template $\mathbf{M} \in \mathbb{R}^{S \times d}$ is appended to the pathway sequence, and the latent positions and functional tokens are updated jointly through self-attention. The latent positions are not assigned to predefined pathways; their roles are learned from the pretraining objectives and can aggregate complementary evidence from the complete cell context.

From the cell-state perspective, the retained latent sequence is a multi-slot summary of coordinated molecular programs. Each slot can exchange information with every pathway-induced position before the pathway states are discarded, so the bottleneck preserves several components of a cell state instead of forcing all variation into a single global statistic. This is related in spirit to CellVQ, which introduces a Single-Cell Discretization module to map a cell representation to a compact cell code for recognition and analysis (Wang et al., 2026a). scPathVQ differs in the object and timing of compression: cross-view contextualization precedes the bottleneck, and the model retains S latent states that are quantized slot-wise and later pooled for cell-level tasks. The resulting sequence is therefore a fixed-length interface for structured prediction, while its pooled readout remains available for annotation and integration. The initialization is defined in the main text, and Equation 8 specifies the joint Transformer updates. The implementation uses $L = 8$ pre-normalized Transformer layers with 8 attention heads, hidden dimension $d = 512$, feed-forward dimension 2,048, and dropout 0.1. Only the final S latent states are retained for vector quantization and downstream representation construction.

The number of latent tokens controls the capacity of this cell-state interface. Table 3 evaluates $S \in \{4, 8, 16, 32, 128, 256\}$ under the same pretraining and downstream protocols. Very small arrays give progressively larger validation reconstruction losses (1.2306, 0.6325, and 0.3286 for $S = 4, 8, 16$), indicating insufficient capacity for the pretraining target even though their measured dead-code rates are zero. Increasing the array beyond 32 does not produce a monotonic gain: validation reconstruction loss rises to 0.4327 and 1.1025 for $S = 128$ and 256, while the dead-code rate increases to 14.06% and 43.75%, respectively. Downstream optima are task-dependent, with $S = 16$ best for Lung annotation and $S = 128$ best for the two Pancreas integration scores and Adamson Delta Pearson, whereas $S = 32$ gives the lowest Adamson MSE (0.102). We therefore use $S = 32$, which combines the lowest validation reconstruction loss (0.1583), a dead-code rate of 3.13%, and competitive performance across the downstream tasks.

Table 3: Ablation over the number of learnable latent tokens S. Validation reconstruction loss and dead-code rate are measured during pretraining; the downstream metrics are evaluated on Lung, Pancreas, and Adamson using the same protocol as the main experiments. Lower is better for reconstruction loss, dead-code rate, and MSE; higher is better for the remaining metrics.

Setting	**Metric**	$S = 4$	$S = 8$	$S = 16$	$S = 32$	$S = 128$	$S = 256$
Pretrain	Val. reconstruction loss	1.2306	0.6325	0.3286	**0.1583**	0.4327	1.1025
	Val. dead-code rate (%)	**0.0000**	**0.0000**	**0.0000**	0.0313	0.1406	0.4375
Cell annotation	Lung (Macro-F1)	0.8570	0.8950	**0.9640**	0.9630	0.9530	0.8770
	Lung (Accuracy)	0.8840	0.9020	**0.9950**	0.9910	0.9890	0.8960
Multi-batch integration	Pancreas (Avg. batch)	0.8490	0.8870	0.8950	0.9070	**0.9100**	0.8740
	Pancreas (Avg. bio.)	0.7850	0.8230	0.8120	0.8450	**0.8510**	0.8260
Perturbation prediction	Adamson (Delta Pearson)	0.7390	0.7680	0.8070	0.8230	**0.8310**	0.5890
	MSE	0.2860	0.2270	0.1350	**0.1020**	0.1090	0.3420

A.4 Cell-State Vector Quantization

Vector quantization is applied after the latent bottleneck has assembled a contextualized cell state. A continuous latent vector preserves fine expression variation, but by itself it does not define a finite vocabulary or a controlled bitrate for comparing and modeling cell states across cells and source datasets. VQ replaces this unconstrained space with a learned, finite vocabulary of state prototypes. The VQ-VAE formulation established this encoder–codebook–decoder pattern for turning continuous representations into discrete symbols that remain usable for reconstruction and generation (Van Den Oord et al., 2017). In scPathVQ, the codebook is shared across all cells and all latent slots, so recurrent configurations of the contextualized cell state can share prototype indices. The bottleneck therefore controls representational capacity while providing a stable coordinate system for pretraining on heterogeneous transcriptomes; the commitment and usage-balance terms further make the assignment problem explicit rather than leaving the continuous encoder free to drift.

For cell i and slot s, we project the final latent state to $d_q = 32$ dimensions, apply layer normalization and unit-norm normalization, and select the nearest vector in a shared spherical codebook $\mathcal{C} = \{\mathbf{c}_k\}_{k=1}^{K}$ with $K = 512$ by cosine similarity (Equation 9). Normalization makes the assignment depend on the direction of the projected state rather than its magnitude, while the shared codebook lets slots from different cells reuse the same state vocabulary. After the 3,000-step continuous warm-up described in Appendix A.5, the forward representation is hard-quantized, but the straight-through estimator passes gradients to the projected latent. We use exponential-moving-average updates with decay $\gamma = 0.99$ and a temperature-0.1 soft assignment only for the codebook-balance loss. Assignment counts and vector sums are aggregated across workers before each update. These choices preserve end-to-end pretraining while making the exposed representation genuinely discrete after warm-up.

The discrete object is a sequence of cell-state components rather than one global cell label. The model exposes the slot-wise code-vector sequence $\mathbf{Q}_i = (\mathbf{q}_{i1}, \ldots, \mathbf{q}_{iS})$, its symbolic index sequence $\boldsymbol{\kappa}_i = (k_{i1}, \ldots, k_{iS})$, and the gated pooled vector $\mathbf{h}_{\mathrm{cell},i}$. This multi-slot construction connects understanding and generation through one representation: cell-annotation and integration heads pool $\mathbf{Q}_i$ into $\mathbf{h}_{\mathrm{cell},i}$, whereas the gene-reconstruction and perturbation heads retain slot positions and condition on $\mathbf{Q}_i$ to model structured expression changes. CellVQ similarly motivates treating a molecular profile as a compact, discretizable cell state for recognition and analysis (Wang et al., 2026a); our design extends this state-level view by quantizing multiple contextualized slots with one shared codebook and retaining both the sequence and its pooled readout. Quantization is thus performed on an integrated cell state, not on individual genes or pathway patches, so the same learned state vocabulary can support cell-level understanding and generation-oriented decoding.

A.5 Pretraining Objectives and Optimization

Pretraining corrupts a complete expression tensor defined on the 9,159-gene union of the KEGG and HVG vocabularies. For each cell, we sample 40% of the nonzero coordinates and at most an

equal number of zero-valued coordinates, with a maximum of 512 masked targets. The selected coordinates are replaced by a mask identifier before either view is gathered, so a gene shared by the pathway-induced and HVG views is masked consistently. The corrupted views are then tokenized, fused, compressed, and passed through the quantizer. Both reconstruction decoders receive the quantizer output, which is continuous during warm-up and hard-quantized afterward.

Reconstruction objective. For $g \in \mathcal{T}_i$, let $\hat{x}_g$ denote the uncorrupted expression target used in Equation 12. The expression decoder reconstructs this value from the quantizer output and the gene identity. The gene-level negative log-likelihood is

$$\mathcal{L}_{\text{gene}} = \mathbb{E}_{i\sim\mathcal{D}}\left[\sum_{g\in\mathcal{T}_i} -\log p_\theta(\hat{x}_g \mid \mathbf{Q}_i, g)\right]. \tag{19}$$

Only the selected masked coordinates contribute to this term. Sampling zero-valued coordinates in proportion to nonzero coordinates prevents zero-valued targets from dominating the selected target set. The gene decoder uses the gated quantizer representation as global context, the reconstructed owner patch as local context for KEGG-covered genes, and a shared gene embedding. HVG-only genes use the same decoder with a zero local-patch term.

The second decoder recovers the fused pathway states from the quantizer output. A two-layer bidirectional Transformer combines projected slots with patch queries constructed from patch identity and KEGG composition, producing $\widehat{\mathbf{Z}}_{im}$ for pathway patch m. Let $v_{im} = 1$ when patch m contains at least one nonpadding gene coordinate and 0 otherwise. The state-level component is

$$\mathcal{L}_{\text{state}} = \mathbb{E}_{i\sim\mathcal{D}}\left[\sum_{m=1}^{P} v_{im}\left\|\widehat{\mathbf{Z}}_{im} - \text{sg}(\mathbf{Z}_{im})\right\|_2^2\right]. \tag{20}$$

The target $\mathbf{Z}_{im}$ is the fused state from Equation 7 and is stop-gradient in this loss. Padding-only patches contribute zero through v_{im}. Reconstructing the fused states from the quantizer output requires the bottleneck to preserve the pathway-structured context assembled before compression. The reconstructed states also provide the local context used by the gene decoder. The two components enter the reconstruction objective with equal weight,

$$\mathcal{L}_{\text{rec}} = \mathcal{L}_{\text{gene}} + \mathcal{L}_{\text{state}}. \tag{21}$$

This expanded definition matches Equation 12 and corresponds to the reconstruction term in Equation 15.

Vector-quantization regularization. The commitment loss $\mathcal{L}_{\text{com}}$ and the balance loss $\mathcal{L}_{\text{bal}}$ are defined in Equations 10 and 11. The temperature-τ_q assignments and their mean usage are

$$\begin{aligned}\pi_{isk} &= \frac{\exp(\bar{\mathbf{z}}_{is}^\top \mathbf{c}_k/\tau_q)}{\sum_{k'=1}^{K}\exp(\bar{\mathbf{z}}_{is}^\top \mathbf{c}_{k'}/\tau_q)},\\ \bar{\pi}_k &= \mathbb{E}_{i\sim\mathcal{D}}\left[\frac{1}{S}\sum_{s=1}^{S}\pi_{isk}\right].\end{aligned} \tag{22}$$

We use $\tau_q = 0.1$. The commitment term penalizes the squared distance between each projected latent and its selected stop-gradient code vector. The balance term computes the KL divergence between $\bar{\boldsymbol{\pi}}$ and the uniform distribution over the 512 codes, discouraging persistent concentration on a small subset of the vocabulary. Soft assignments affect only this regularizer and diagnostics; after warm-up, the exposed representation uses the selected hard code vector. Code vectors are updated by exponential moving averages with decay 0.99. We log soft-assignment sharpness but assign its loss coefficient a value of zero.

During the first 3,000 optimization steps, the quantizer directly forwards the continuous normalized latents and keeps $\mathcal{L}_{\text{com}}$ inactive. Hard quantization and the commitment term are enabled after this warm-up. All other loss coefficients remain fixed throughout pretraining.

Ontology-aware contrastive supervision. Let $\mathcal{I}_{\text{lab}}$ denote cells with harmonized Cell Ontology identifiers y_i, and let $\mathcal{I}_i^+ = \{j \in \mathcal{I}_{\text{lab}} \setminus \{i\} : y_j = y_i\}$ be the positives for anchor i. With $\mathbf{h}_i$ denoting a normalized projection of $\mathbf{h}_{\text{cell},i}$, the finite-sum form of Equation 14 is

$$\mathcal{L}_{\text{ont}} = -\frac{1}{|\mathcal{I}_{\text{anc}}|} \sum_{i \in \mathcal{I}_{\text{anc}}} \frac{1}{|\mathcal{I}_i^+|} \sum_{j \in \mathcal{I}_i^+} \log \frac{\exp(\bar{\mathbf{h}}_i^\top \bar{\mathbf{h}}_j / \tau)}{\sum_{i' \in \mathcal{I}_{\text{lab}} \setminus \{i\}} \exp(\bar{\mathbf{h}}_i^\top \bar{\mathbf{h}}_{i'} / \tau)}, \tag{23}$$

where $\mathcal{I}_{\text{anc}} = \{i \in \mathcal{I}_{\text{lab}} : |\mathcal{I}_i^+| > 0\}$. Embeddings and labels are gathered across workers to enlarge the contrastive set. Unlabeled cells are excluded from $\mathcal{L}_{\text{ont}}$ but continue to train all reconstruction and VQ terms. A labeled anchor is likewise skipped when no same-label positive occurs in the distributed batch. The contrastive temperature is 0.1. Its two-layer projection head is used only during pretraining and is discarded when exporting $\mathbf{h}_{\text{cell}}$.

Coefficient selection. We use reconstruction as the reference scale and fix $\lambda_{\text{rec}} = 1.0$ without tuning it. We search $\lambda_{\text{com}} \in \{0.01\} \cup \{0.05, 0.10, \ldots, 0.50\}$ and $\lambda_{\text{bal}}, \lambda_{\text{ont}} \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1\}$. We select $(\lambda_{\text{com}}, \lambda_{\text{bal}})$ on held-out pretraining data by validation reconstruction subject to codebook-health constraints, then select λ_{ont} by ontology-label retrieval without degrading reconstruction. The selected coefficients are fixed for the complete run.

With the selected fixed values, the pretraining objective is

$$\mathcal{L}_{\text{pre}} = 1.0\,\mathcal{L}_{\text{rec}} + 0.25\,\mathcal{L}_{\text{com}} + 0.01\,\mathcal{L}_{\text{bal}} + 0.1\,\mathcal{L}_{\text{ont}}. \tag{24}$$

Within $\mathcal{L}_{\text{rec}}$, the gene-level and state-level components have equal weight because they supervise complementary resolutions of the same bottleneck: the former preserves individual measurements, whereas the latter preserves the pathway-structured state from which those measurements are decoded. The smaller VQ coefficients regularize discretization without replacing reconstruction as the primary signal. The 3,000-step quantizer warm-up is the only stage-dependent component of the objective.

Optimization. We optimize for 1,000,000 logical steps on four Tesla V100-SXM2 32-GB workers with AdamW, using a microbatch of 256 cells per worker, one accumulation step, and a global batch size of 1024. The learning rate is 2×10^{-4} with $\beta_1 = 0.9$ and $\beta_2 = 0.95$. Weight decay is 0.05 for matrix-valued parameters and zero for normalization parameters and biases. The learning rate increases linearly for the first 0.5% of training (5,000 steps) and then follows cosine decay. Training uses FP16 mixed precision, while LayerNorm, VQ assignment and codebook updates, EMA statistics, and loss reductions are evaluated in FP32. Gradients are clipped to a maximum norm of 1.0. Validation and checkpointing are performed every 1,000 logical steps, and all pretraining runs use seed 2026.

A.6 End-to-End Pretraining Algorithm

Algorithm 1 summarizes the complete optimization procedure. The corruption mask is applied to the union expression tensor before either input view is gathered, and the codebook is updated from assignment statistics aggregated across workers rather than by AdamW.

Algorithm 1 End-to-end pretraining of scPathVQ

Require: Corpus $\mathcal{D}$; pathway patch map; fixed HVG panel $\mathcal{H}$; Steps T; global batch size B
Ensure: Optimized trainable parameters Θ and EMA codebook $\mathcal{C}$

Initialize Θ, latent template $\mathbf{M}$, codebook $\mathcal{C}$, and EMA statistics
for $t = 1, \ldots, T$ **do**
  Sample a distributed minibatch $\mathcal{B}_t \subset \mathcal{D}$ with available ontology labels
  for all $i \in \mathcal{B}_t$ **do**
    Sample $\mathcal{T}_i$ from 40% of nonzeros and at most as many zeros; cap at 512
    Mask $\mathcal{T}_i$ before gathering the pathway and HVG views
    $(\mathbf{P}_i, \mathbf{H}_i) \leftarrow \textsc{Tokenize}(\mathbf{x}_i^{\text{mask}})$
    $\mathbf{Z}_i \leftarrow \textsc{CrossViewAttention}(\mathbf{P}_i, \mathbf{H}_i)$
    $\widetilde{\mathbf{Z}}_i \leftarrow \textsc{Transformer}([\mathbf{Z}_i; \mathbf{M}])_{P+1:P+S}$
    Project and normalize $\widetilde{\mathbf{Z}}_i$; assign indices $\{k_{is}\}_{s=1}^{S}$ in $\mathcal{C}$
    $\mathbf{Q}_i \leftarrow \{\bar{\mathbf{z}}_{is} + \text{sg}(\mathbf{c}_{k_{is}} - \bar{\mathbf{z}}_{is})\}_{s=1}^{S}$
  end for
  All-reduce assignment statistics and update $\mathcal{C}$ by EMA
  Decode masked genes and fused pathway states from $\{\mathbf{Q}_i\}$; compute $\mathcal{L}_{\text{rec}}$
  Compute $\mathcal{L}_{\text{bal}}$ and $\mathcal{L}_{\text{com}}$ (zero during warm-up)
  Gate-pool $\{\mathbf{Q}_i\}$; all-gather labeled embeddings and compute $\mathcal{L}_{\text{ont}}$
    (set $\mathcal{L}_{\text{ont}} = 0$ if no anchor has a same-label positive)
  $\mathcal{L}_{\text{pre}} \leftarrow \lambda_{\text{rec}}\mathcal{L}_{\text{rec}} + \lambda_{\text{com}}\mathcal{L}_{\text{com}} + \lambda_{\text{bal}}\mathcal{L}_{\text{bal}} + \lambda_{\text{ont}}\mathcal{L}_{\text{ont}}$
  Update Θ with AdamW, the scheduled rate, and unit-norm gradient clipping
end for

B Pretraining and Downstream Datasets

We use an atlas-scale corpus for pretraining and task-specific collections for four downstream tasks. Cell understanding uses the pooled embedding $\mathbf{h}_{\text{cell}}$ for cell-type annotation and multi-batch integration. Cell generation uses the quantized sequence $\mathbf{Q}$ to predict post-perturbation expression, which is evaluated at the gene level through perturbation prediction and at the functional-program level through cell state inference. The two generation tasks share the perturbation screens and held-out condition splits.

B.1 Pretraining Corpus

The pretraining corpus contains 34,017,503 human cells collected from 252 source datasets in the May and July 2023 CELLxGENE releases. The May release contributes most of the cells. We use 31,389,539 cells for optimization and reserve 2,627,964 cells for validation. Table 4 reports the source composition, split, and ontology-label coverage.

Cells with a valid `cell_type_ontology_term_id` are harmonized into 609 Cell Ontology identifiers (Diehl et al., 2016). Identifier equality defines the positive pairs used by the ontology-aware contrastive objective in Equation 23. Cells without a valid identifier remain in the corpus and contribute to reconstruction and vector-quantization training. The tokenizer operates on an active vocabulary of 9,159 genes shared across the corpus.

The source datasets span hematopoietic, nervous, respiratory, digestive, reproductive, and other organ systems (Table 5). Each cell is counted once in this summary; collections that cover several organs are assigned to the multi-tissue category. This breadth exposes the pretrained interface to cell identity and state variation across healthy and disease contexts rather than concentrating it in one tissue.

B.2 Datasets for Cell Understanding

Cell understanding evaluates whether $\mathbf{h}_{\text{cell}}$ summarizes an observed transcriptome in a form that supports semantic comparison. Cell-type annotation tests discrimination among biological identities, while multi-batch integration tests whether the same embedding preserves those identities when acquisition source and technology vary. Table 6 summarizes the six datasets used for these evaluations.

Table 4: Composition, split, and ontology coverage of the 34,017,503-cell pretraining corpus. Percentages are relative to the complete corpus.

Subset	Cells	Percentage
CELLxGENE	34,017,503	100%
Training split	31,389,539	92.27%
Validation split	2,627,964	7.73%
Valid Cell Ontology labels	32,605,001	95.85%
Complete corpus	34,017,503	100.00%

Table 5: Tissue scope of the pretraining corpus. Each cell contributes to one category, and sources spanning several organs are grouped as multi-tissue collections.

Tissue scope	Cells	Percentage	Tissue scope	Cells	Percentage
Multi-tissue collections	11,458,461	33.68%	Cardiovascular	1,405,738	4.13%
Hematopoietic and lymphoid	6,848,749	20.13%	Digestive tract	1,361,934	4.00%
Nervous system	6,702,587	19.70%	Reproductive and placental	768,395	2.26%
Respiratory system	2,649,519	7.79%	Eye	551,005	1.62%
Skin, adipose, and breast	1,806,031	5.31%	Liver and biliary	308,003	0.91%
Pancreas	96,119	0.28%	Kidney and urinary	60,962	0.18%

PBMC68K contains 68,579 peripheral-blood mononuclear cells measured by droplet-based scRNA-seq and assigned to 11 cell-type labels (Zheng et al., 2017). The annotation task therefore tests whether the pooled representation separates related immune populations within one tissue. The liver-cancer dataset is the 5,115-cell discovery cohort of GSE125449, profiled from 12 primary liver-cancer patients (Ma et al., 2019). After excluding "unclassified" cells, its seven labels cover B cells, T cells, tumor-associated macrophages, tumor endothelial cells, cancer-associated fibroblasts, malignant cells, and hepatic progenitor-like cells. This collection extends annotation from circulating immune cells to interacting malignant, immune, and stromal compartments.

The Lung dataset is the GSE131907 lung-adenocarcinoma atlas, with 208,506 cells from 58 samples and 44 patients (Kim et al., 2020). The samples span primary tumors, metastases, normal lung tissue, and pleural effusions. We use seven refined classes for annotation and ten biological labels for integration. Sample identity defines 58 batch units in the latter setting. The Melanoma dataset represents a second tumor microenvironment and contains 6,879 cells across 32 patient/sample batches after cells marked with the "?" label are removed. Its nine retained cell types include malignant and non-malignant populations, providing an annotation setting with strong inter-sample variation (Tirosh et al., 2016).

Multi-batch integration reuses PBMC68K, Lung, and Melanoma and adds Pancreas and Immune. The Pancreas benchmark contains 16,382 cells, 14 cell types, and nine acquisition batches generated with CEL-Seq, CEL-Seq2, Fluidigm C1, inDrop, SMARTer, and SMARTer2. Its technology diversity makes batch mixing and biological conservation jointly necessary. The Immune benchmark contains 33,506 human immune cells from ten batches and 16 cell types. Together, these collections evaluate whether $\mathbf{h}_{\mathrm{cell}}$ aligns acquisition-specific measurements without erasing tissue- or lineage-level structure under the common scIB-style protocol (Luecken et al., 2022).

B.3 Datasets for Quantized-Sequence Evaluation

These evaluations test the functional information retained by $\mathbf{Q}$ for conditional generation. A model receives a source cell state and a genetic intervention and predicts the target expression state. Perturbation prediction measures gene-level response accuracy, while cell state inference assesses pathway-associated states and pathway responses derived from the generated profiles. Table 7 summarizes the three perturbation screens. Norman supplies the cell-state examples in the main text, and the additional pathway-response analyses extend to Adamson and Dixit.

Adamson is a CRISPR interference screen designed to resolve transcriptional responses within the unfolded protein response (Adamson et al., 2016). The processed benchmark contains 68,603 cells

Table 6: Datasets used for cell understanding. Sparsity is the zero-entry fraction of the processed expression matrix. For Lung, the two cell-type counts correspond to annotation and integration, respectively.

Dataset	Evaluation	Cells	Genes	Cell types	Sparsity
PBMC68K	Annotation and integration	68,579	12,787	11	0.9667
Liver cancer	Annotation	5,115	14,969	7	0.9037
Lung	Annotation and integration	208,506	29,634	7 / 10	0.9463
Melanoma	Annotation and integration	6,879	17,614	9	0.7922
Pancreas	Integration	16,382	19,093	14	0.8221
Immune	Integration	33,506	12,303	16	0.8752

Table 7: Perturbation screens used to evaluate generation from the quantized sequence. The same screens support gene-level response prediction and functional cell state inference. Entries report canonical perturbation conditions, including controls where specified.

Dataset	Evaluation	Cells	Genes	Conditions	Sparsity
Adamson	Perturbation response	68,603	5,060	87 (86 + control)	0.7932
Dixit	Perturbation response	44,735	5,012	20 (19 + control)	0.8051
Norman	Perturbation response	91,205	5,045	237 single-/double-gene conditions	0.9189

and 87 canonical conditions, comprising 86 single-gene perturbations and a control. Dixit uses pooled CRISPR perturbations with single-cell RNA profiling to connect genetic interventions to molecular phenotypes (Dixit et al., 2016). Its benchmark matrix contains 44,735 cells and 20 conditions, including 19 perturbation identities and a control. These two datasets test whether $\mathbf{Q}$ supports prediction of held-out single-gene responses rather than only reconstruction of observed cells.

Norman uses CRISPR activation to study genetic interactions from rich single-cell phenotypes (Norman et al., 2019). It contains 91,205 cells and 237 canonical conditions spanning single- and double-gene interventions, so it extends perturbation prediction from individual effects to combinatorial state changes. For all three perturbation datasets, the evaluation follows condition-level splits and control-relative prediction used by GEARS (Roohani et al., 2024). Appendix C.2 specifies the downstream protocol.

B.4 Dataset Processing and Statistical Definitions

All counts in Tables 6 and 7 refer to the processed matrices used by the corresponding evaluation. Sparsity is computed from each stored sparse expression matrix as $1 - \mathrm{nnz}/(N \times G)$, where N is the number of cells and G is the number of genes. The pretraining matrix contains 65,721,557,551 nonzero entries over its 9,159-gene active vocabulary, giving a sparsity of 0.7891. Task-specific label spaces may differ across evaluations. The two Lung cell-type counts in Table 6 follow the annotation/integration order. Reported counts incorporate the label filtering described above, including removal of "unclassified" liver-cancer cells and "?" melanoma cells. A dash denotes a statistic that is not part of the corresponding evaluation summary.

C Downstream Experimental Protocols and Evaluation Metrics

The downstream evaluation is organized by the representation consumed by each task. Cell understanding uses the pooled embedding $\mathbf{h}_{\mathrm{cell}}$ for annotation and integration. Cell generation uses the slot-wise code-vector sequence $\mathbf{Q}$ for perturbation prediction, followed by functional state inference from generated expression. Each subsection specifies the input representation, split unit, downstream computation, and evaluation readout.

C.1 Cell Annotation

Protocol. We evaluate cell-type annotation on PBMC68K, Liver cancer, Lung, and Melanoma using the harmonized labels described in Appendix B. The mCRC collection is excluded because

cross-file barcode–expression collisions prevent an unambiguous alignment between cells and expression profiles. Each dataset is partitioned into train, validation, and test sets with an 8:1:1 ratio. For frozen linear probing, a linear classifier receives $\mathbf{h}_{\text{cell}}$ and the encoder and codebook remain fixed. The train-from-scratch control uses the matched scPathVQ downstream architecture without pretrained parameters. For full fine-tuning, the encoder and classifier are optimized jointly, while the pretrained codebook and its exponential-moving-average updates remain fixed. The reported values are means and sample standard deviations over five fixed seeds (42–46).

Metrics. Let C be the number of cell types, and let P_c and R_c denote precision and recall for class c. We compute

$$F1_c = \frac{2P_cR_c}{P_c + R_c}, \qquad \text{Macro-F1} = \frac{1}{C}\sum_{c=1}^{C} F1_c. \tag{25}$$

Macro-F1 gives every cell type equal weight and therefore prevents the largest class from dominating the summary. Accuracy is the fraction of test cells whose predicted label equals the supplied label. Both metrics are higher-is-better and are computed on the held-out test partition.

C.2 Perturbation Prediction

Protocol. We evaluate Adamson, Dixit, and Norman Perturb-seq benchmarks. Adamson contains single-gene CRISPR interference conditions, Dixit contains single-gene CRISPR perturbations, and Norman contains single- and double-gene CRISPR activation conditions. Train, validation, and test partitions are defined over perturbation identities rather than individual cells (Roohani et al., 2024); cells belonging to a held-out identity are used only for evaluation. The frozen backbone exposes the post-quantization sequence $\mathbf{Q}_i = (\mathbf{q}_{i1}, \ldots, \mathbf{q}_{iS})$ for source cell i. A task-specific predictor combines this sequence with the pretrained embedding of the perturbed gene or gene pair and outputs a control-relative expression change for the target genes. For each held-out condition, observed and predicted profiles are averaged over its cells before metric computation.

Metrics. For held-out condition c and gene j, let Δ_{cj} and $\widehat{\Delta}_{cj}$ be the observed and predicted changes relative to the dataset control mean. The condition-level Delta Pearson score is

$$r_c = \text{Pearson}\left(\Delta_{c,:}, \widehat{\Delta}_{c,:}\right), \qquad \text{Delta Pearson} = \frac{1}{|\mathcal{C}_{\text{test}}|}\sum_{c\in\mathcal{C}_{\text{test}}} r_c, \tag{26}$$

where $\mathcal{C}_{\text{test}}$ is the set of held-out perturbation identities. This metric evaluates response direction and relative gene-wise pattern. We measure response magnitude with

$$\text{MSE} = \frac{1}{|\mathcal{C}_{\text{test}}|G}\sum_{c\in\mathcal{C}_{\text{test}}}\sum_{j=1}^{G}\left(\widehat{\Delta}_{cj} - \Delta_{cj}\right)^2. \tag{27}$$

Delta Pearson is higher-is-better, whereas MSE is lower-is-better. Averaging at the condition level gives each intervention equal weight rather than weighting the dataset by the number of cells observed for that intervention.

C.3 Multi-Batch Integration

Protocol. Multi-batch integration evaluates whether a representation reduces acquisition-specific variation while preserving cell identity or cell-state structure. We use Pancreas, PBMC68K, Lung, Melanoma, and Immune as the five integration collections. For every cell, we extract $\mathbf{h}_{\text{cell}}$ and use this same representation for metric computation and PCA visualization. The batch-colored and cell-type-colored panels use shared coordinates, so the visual comparison does not introduce a second representation. Batch mixing and biological conservation are reported jointly in Table 8.

Metrics. Let z_i denote the embedding of cell i. For a generic label partition y, let $a_i(y)$ be the mean distance from z_i to cells in its own group and let $d_i^\star(y)$ be the smallest mean distance to any other group. The silhouette width and its dataset average are

$$s_i(y) = \frac{d_i^\star(y) - a_i(y)}{\max(a_i(y), d_i^\star(y))}, \qquad \text{ASW}(y) = \frac{1}{N}\sum_{i=1}^{N} s_i(y). \tag{28}$$

Using batch labels b, we compute $\mathrm{ASW}_{\mathrm{batch}} = \mathrm{ASW}(b)$ and transform it into a mixing score:

$$\mathrm{Avg_batch} = 1 - \mathrm{ASW}_{\mathrm{batch}}. \tag{29}$$

Lower batch-label silhouette indicates stronger mixing, so the transformed score is higher when batch effects are weaker. For biological conservation, clustering the embeddings produces assignments $\widehat{c}_i$, which are compared with the supplied cell-type labels c_i using normalized mutual information

$$\mathrm{NMI}(\widehat{c}, c) = \frac{2I(\widehat{c}; c)}{H(\widehat{c}) + H(c)} \tag{30}$$

and the adjusted Rand index $\mathrm{ARI}(\widehat{c}, c)$. We also compute $\mathrm{ASW}_{\mathrm{bio}} = \mathrm{ASW}(c)$ and report the biological-conservation aggregate

$$\mathrm{Avg_bio} = \frac{\mathrm{NMI}(\widehat{c}, c) + \mathrm{ARI}(\widehat{c}, c) + \mathrm{ASW}_{\mathrm{bio}}}{3}. \tag{31}$$

Both integration aggregates are higher-is-better. The same distance and clustering configuration is used when comparing representations; the aggregate definitions themselves do not depend on a task-specific prediction head.

C.4 Cell State Inference

Evaluation setting. Cell state inference characterizes perturbation-induced functional states from generated post-perturbation expression. We use the same condition-level split as perturbation prediction, with seed 1, so test perturbation identities are held out during downstream training. The analysis has two resolutions: a cell-level attention readout for the Norman *BAK1* example and an expression-derived pathway-response score for comparisons across perturbation conditions. Both use the filtered human KEGG collection described in Appendix A.1.

From quantized codes to generated expression. For source cell i, let $(k_{i1}, \ldots, k_{iS})$ denote its discrete code indices and let $\mathbf{c}_k$ be codebook entry k. The corresponding code-vector sequence and generated expression under condition c are

$$\mathbf{Q}_i = (\mathbf{c}_{k_{i1}}, \ldots, \mathbf{c}_{k_{iS}}), \qquad \widehat{\mathbf{x}}_i^{(c)} = G_\psi(\mathbf{Q}_i, \mathbf{e}_c), \tag{32}$$

where $\mathbf{e}_c$ represents the perturbed gene or gene pair. Here, G_ψ denotes the full conditional expression prediction map, including the conversion of a control-relative prediction to an expression profile. For the *BAK1* analysis, this computation is performed for individual source cells, yielding a matrix of generated post-perturbation profiles.

Re-encoding and latent-to-patch attention. Each generated expression profile is processed by the same pathway-induced tokenizer, HVG contextualization block, and Transformer encoder used for measured profiles. We extract attention from the final Transformer layer, taking latent-token positions as queries and pathway-patch positions as keys. The following equations describe the readout for a generic input profile $\mathbf{x}_i$; applying them to $\widehat{\mathbf{x}}_i^{(c)}$ gives the corresponding generated-cell scores. Let $\mathbf{A}_{i,h}^{(L)}$ be the final-layer attention matrix for head h, with P patch positions followed by S latent positions. We normalize the extracted attention over the valid patch keys and then average equally over all H heads and S latent slots:

$$\begin{aligned}
\alpha_{i,h,s,m} &= \left[\mathbf{A}_{i,h}^{(L)}\right]_{P+s,m}, \qquad s = 1, \ldots, S, \quad m = 1, \ldots, P, \\
\widetilde{\alpha}_{i,h,s,m} &= \frac{\alpha_{i,h,s,m}}{\sum_{m'=1}^{P} \alpha_{i,h,s,m'}}, \\
u_{i,m} &= \frac{1}{HS} \sum_{h=1}^{H} \sum_{s=1}^{S} \widetilde{\alpha}_{i,h,s,m}.
\end{aligned} \tag{33}$$

The resulting $u_{i,m}$ describes the relative allocation of attention among pathway patches, with $\sum_m u_{i,m} = 1$. This readout uses latent-to-patch attention in the final encoder layer; the pathway-to-HVG attention concentration analyzed in Appendix E is a separate diagnostic.

Aggregation from patches to pathways. We map patch scores to biological pathways using the tokenizer's fixed gene-membership composition π_{mt} from Equation 18. This composition counts mapped genes shared by patch m and pathway t, excluding padding positions. We normalize the composition over patches for each pathway and compute a weighted average:

$$w_{m,t} = \frac{\pi_{mt}}{\sum_{m'=1}^{P} \pi_{m't}}, \qquad a_{i,t} = \sum_{m=1}^{P} w_{m,t} u_{i,m}. \tag{34}$$

The pathway weights satisfy $\sum_m w_{m,t} = 1$, so the readout does not increase merely because a pathway spans more patches. Gene memberships can overlap across pathways. The same weights are used for pre-perturbation, measured post-perturbation, and generated profiles.

Control-referenced cell-state scores. Let $\mathcal{C}_0$ be a fixed set of control cells, scored with the same attention readout. For each pathway t, we compute its reference mean and standard deviation:

$$\mu_t^0 = \frac{1}{|\mathcal{C}_0|} \sum_{j \in \mathcal{C}_0} a_{j,t}, \qquad \sigma_t^0 = \sqrt{\frac{1}{|\mathcal{C}_0|} \sum_{j \in \mathcal{C}_0} (a_{j,t} - \mu_t^0)^2}. \tag{35}$$

These reference statistics are held fixed across the three groups of profiles. We standardize each pathway readout relative to controls and apply a hyperbolic tangent to obtain a signed score bounded by $[-1, 1]$:

$$z_{i,t} = \frac{a_{i,t} - \mu_t^0}{\max(\sigma_t^0, \epsilon_{\text{att}})}, \qquad s_{i,t} = \tanh(z_{i,t}), \qquad m_{i,t} = \mathbb{I}[s_{i,t} > 0], \tag{36}$$

where $\epsilon_{\text{att}} = 10^{-8}$. A positive score selects a cell whose pathway-associated attention exceeds the control mean. For generated cells, the vector $\widehat{\mathbf{s}}_i^{(c)} = (\widehat{s}_{i,1}^{(c)}, \ldots, \widehat{s}_{i,T}^{(c)})$ describes the predicted functional state over the T pathways, and the collection of these vectors characterizes the state distribution under perturbation c. Pathways are scored separately, allowing their selected-cell sets to overlap. Figure 6 displays the selected cells in shared UMAP coordinates for cell cycle, apoptosis, ferroptosis, necroptosis, and cellular senescence; gray points provide the embedding background. The three rows correspond to pre-perturbation expression, measured post-perturbation expression, and expression generated for individual source cells.

Expression-derived pathway responses. We additionally evaluate agreement between expression-derived pathway responses in measured and generated profiles. Pathways come from the filtered human KEGG collection described in Appendix A.1; pathways with fewer than three mapped genes are excluded. For a held-out perturbation condition c, let $\mathcal{I}_c$ denote its measured cells and let $\mathcal{I}_{\text{ctrl}}$ denote the control cells used by the perturbation evaluation protocol. The observed response relative to the dataset control mean is

$$\Delta_{c,g}^{\text{obs}} = \frac{1}{|\mathcal{I}_c|} \sum_{i \in \mathcal{I}_c} x_{i,g} - \frac{1}{|\mathcal{I}_{\text{ctrl}}|} \sum_{i \in \mathcal{I}_{\text{ctrl}}} x_{i,g}, \tag{37}$$

where $x_{i,g}$ is the evaluation-preprocessed expression value of gene g in cell i. For the heatmaps, the predicted condition-level response is the output of the latent-transition head,

$$\widehat{\boldsymbol{\Delta}}_c = f_\phi\big(\text{vec}(\overline{\mathbf{Q}}_{\text{ctrl}}), \mathbf{e}_c\big), \qquad \overline{\mathbf{Q}}_{\text{ctrl}} = \frac{1}{|\mathcal{I}_{\text{ctrl}}^Q|} \sum_{i \in \mathcal{I}_{\text{ctrl}}^Q} \mathbf{Q}_i, \tag{38}$$

where $\mathbf{Q}_i$ is the hard post-VQ code-vector sequence, $\mathbf{e}_c$ is the perturbation embedding, and $\mathcal{I}_{\text{ctrl}}^Q$ is the control subset used to construct the downstream latent input. For either response vector $\boldsymbol{\Delta}_c$, we standardize across the evaluated gene set $\mathcal{G}$, where $G = |\mathcal{G}|$,

$$z_{c,g} = \frac{\Delta_{c,g} - \mu_c}{\max(\sigma_c, \epsilon)}, \qquad \mu_c = \frac{1}{G} \sum_{g=1}^{G} \Delta_{c,g}, \tag{39}$$

where σ_c is the across-gene standard deviation and ϵ is a numerical stabilizer. For pathway p with mapped gene set $\mathcal{P}_p$, the response score is

$$A_{c,p} = \frac{1}{|\mathcal{P}_p|} \sum_{g \in \mathcal{P}_p} z_{c,g} - \frac{1}{|\mathcal{G} \setminus \mathcal{P}_p|} \sum_{g \in \mathcal{G} \setminus \mathcal{P}_p} z_{c,g}. \tag{40}$$

Positive and negative values indicate relative upward and downward responses of pathway genes against the transcriptome-wide background. This expression-derived contrast provides a condition-level readout complementary to the attention-based cell-state score.

Agreement evaluation and heatmap selection. Pearson and Spearman correlations compare observed and predicted response scores over all evaluable test conditions and pathways. For visualization, we rank pathways by their mean absolute predicted score across test conditions and select the top 20. We then select up to eight conditions with the largest mean absolute predicted response over those pathways. Observed and predicted panels use the same pathways, conditions, and color scale. Figure 5 shows Norman, and Figure 14 shows Adamson and Dixit. Display selection does not restrict the pathway–condition pairs used to compute the reported correlations.

D Additional Experimental Results

D.1 Cell-Annotation Diagnostics

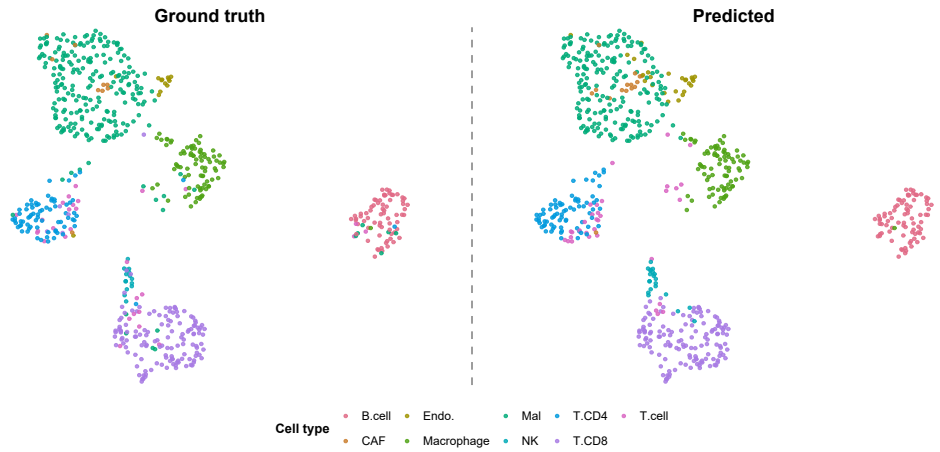
Figures 7 and 9 combine three views of the same seed-42 full-fine-tuning outputs for Melanoma and PBMC68K. The UMAPs compare ground-truth and predicted labels on the same fixed coordinates, the confusion matrices expose class-specific errors on the test partition, and the embedding heatmaps show how label-associated structure is distributed across the learned coordinates. In both datasets, the predicted labels largely align with the major regions defined by the ground truth. This alignment is visible for the separated malignant, myeloid, and lymphoid regions in Melanoma and across the broader immune manifold and dense T-cell neighborhoods in PBMC68K. The confusion matrices indicate that the remaining errors are concentrated among related immune states rather than being uniformly distributed: the Melanoma B-cell and macrophage classes have diagonal fractions of 1.00 and 0.95, whereas the generic T.cell class is split across the CD4 and CD8 labels. PBMC68K shows strong diagonals for relatively distinct populations such as CD34+ and CD56+ NK, while the finer CD4 and CD8 subtypes exhibit more mixing. The corresponding heatmaps contain label-associated blocks together with within-class variation, which is consistent with a representation that separates broad identity while retaining state heterogeneity. These panels provide qualitative diagnostics for the aggregate Macro-F1 and accuracy values in Table 1; they are not additional test sets.

D.2 Perturbation Diagnostics

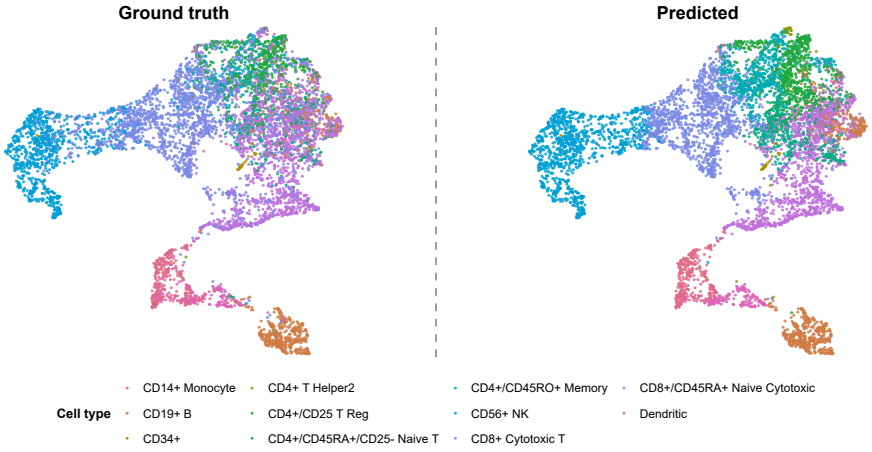
Figures 11 and 10 give complementary retrieval-level and gene-level views of perturbation prediction. In the Norman panels, the retrieved condition and the actual source condition are marked in the same pairwise response matrix for two combinatorial examples. The panels make the response-space ranking inspectable, but they use measured combinations as the retrieval pool and therefore do not replace the condition-level Delta Pearson and MSE evaluation. The Adamson panels compare observed distributions with visualizations centered on the model-predicted means for the top 20 differentially expressed genes under six single-gene perturbations. Across XRN1, GMPPB, AMIGO2, GNPNAT1, DERL2, and SYVN1, the predicted condition means generally track the direction and central tendency of the observed shifts, including positive responses among selected condition-responsive genes. Because Pred^* adds centered control-cell residuals, its violin width and tails are visualization devices rather than model-predicted distributional properties. These panels therefore provide a gene-level check of predicted condition means, while the aggregate metrics in Figure 3 remain the quantitative evaluation.

D.3 Extended Integration Visualizations

Figure 12 extends the integration visualization beyond the Pancreas example in the main text. The batch-colored panels show substantial intermixing across the dominant Melanoma and Immune manifolds, while the matched cell-type panels preserve coherent malignant, stromal, myeloid, lymphoid, and progenitor populations. The Immune cell-type view also retains continuous branches connecting related monocyte and lymphoid states. Batch-enriched peripheral islands and elongated bridges remain visible in both collections, so the projections support partial batch correction rather than complete removal of acquisition structure. The visual evidence should be read together with the quantitative $\mathrm{Avg}_{\mathrm{batch}}$ and $\mathrm{Avg}_{\mathrm{bio}}$ values reported for all five datasets in Table 8.



(a) UMAP view for Melanoma



(b) UMAP view for PBMC68K

Figure 7: Ground-truth and predicted-label UMAPs for Melanoma and PBMC68K. Each pair uses the same coordinates.

Table 8: Multi-batch integration results on five datasets. $\mathrm{Avg}_{\mathrm{batch}}$ measures batch mixing and $\mathrm{Avg}_{\mathrm{bio}}$ measures biological-identity preservation; higher values are better for both metrics. The best and second-best values in each dataset–metric pair are shaded in dark and light blue, respectively.

Metric	Dataset	Harmony	Geneformer	scFoundation	scGPT	CellFM	scPathVQ
Avg_batch ↑	Pancreas	0.893	0.836	0.887	0.874	0.901	**0.907**
	PBMC68K	0.912	0.901	0.928	0.905	0.923	**0.931**
	Lung	0.901	0.889	0.912	0.905	**0.928**	0.924
	Melanoma	0.639	0.672	0.701	0.697	0.715	**0.732**
	Immune	0.951	0.825	0.942	0.931	0.949	**0.957**
Avg_bio ↑	Pancreas	0.765	0.832	0.818	0.821	0.823	**0.845**
	PBMC68K	0.782	0.801	0.822	0.805	0.819	**0.836**
	Lung	0.754	0.789	0.803	0.798	0.811	**0.819**
	Melanoma	0.651	0.712	0.748	0.741	0.759	**0.772**
	Immune	0.693	0.731	0.752	0.745	0.763	**0.779**

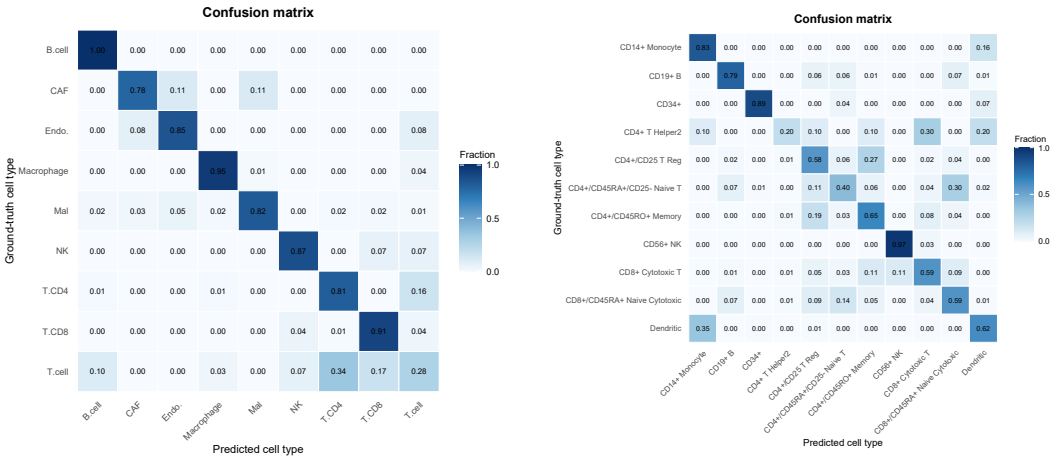
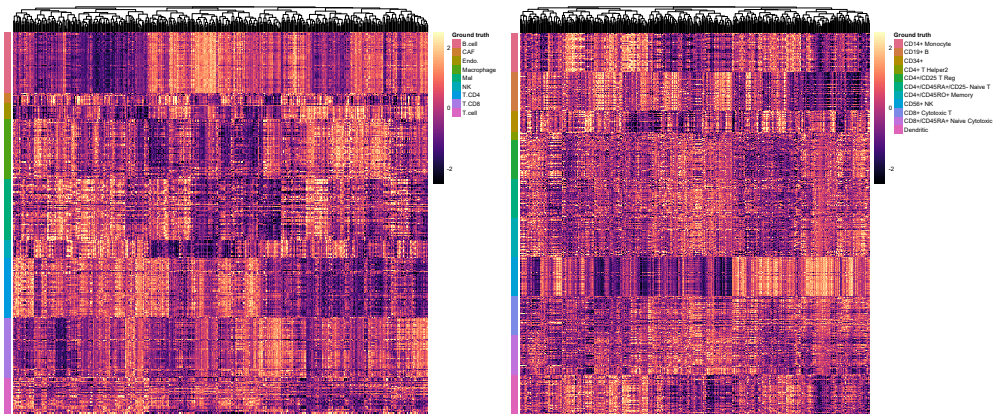


Figure 8: Row-normalized test-set confusion matrices, with ground-truth classes as rows and predicted classes as columns for Melanoma (left) and PBMC68K (right) datasets



(a) Embedding heatmaps

Figure 9: Hierarchically clustered embedding heatmaps with the ground-truth label strip. These two views assess class-specific errors and label-associated representation structure.

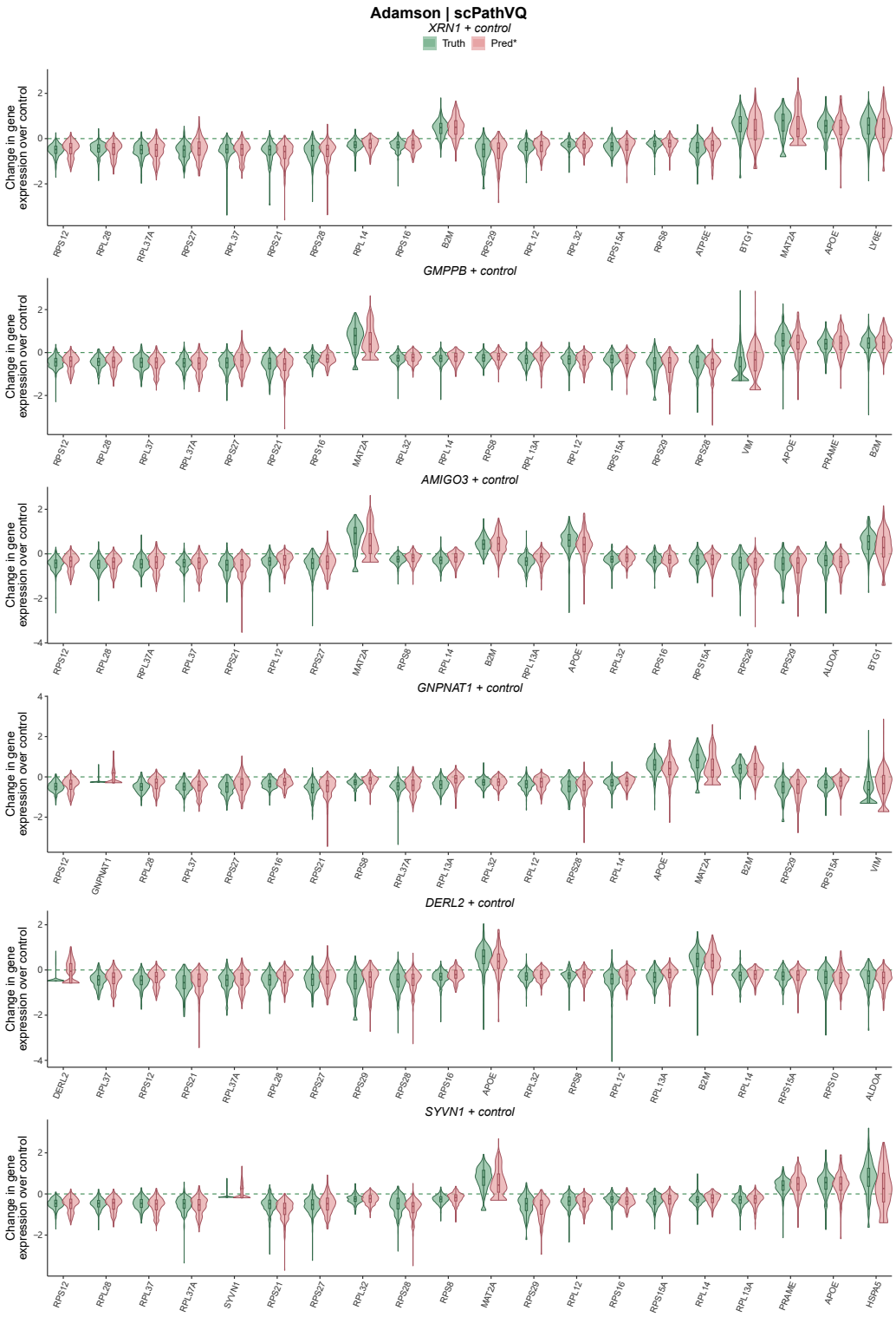


Figure 10: Observed and visualized predicted (Pred^*) Adamson distributions for the top 20 differentially expressed genes under XRN1, GMPPB, AMIGO2, GNPNAT1, DERL2, and SYVN1. Pred^* preserves the condition mean and adds centered control-cell residuals; violin width reflects residual spread, not predictive uncertainty.

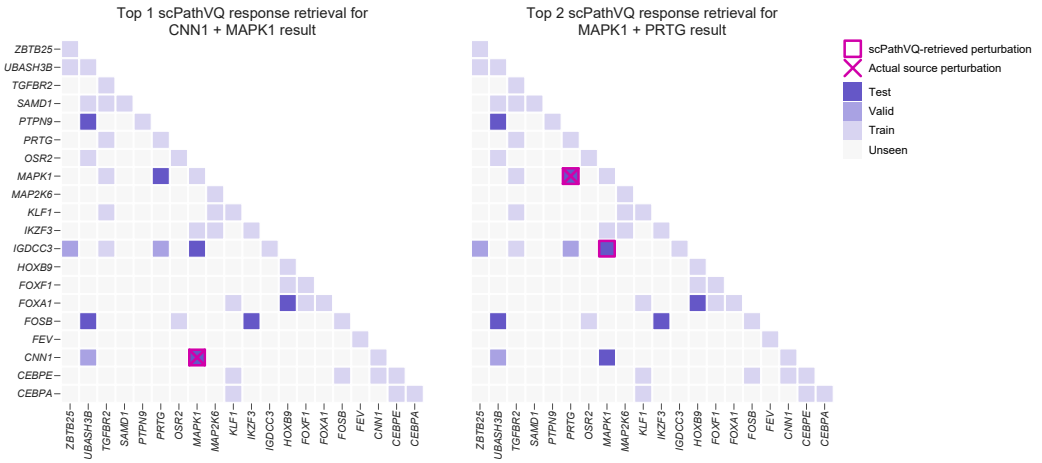
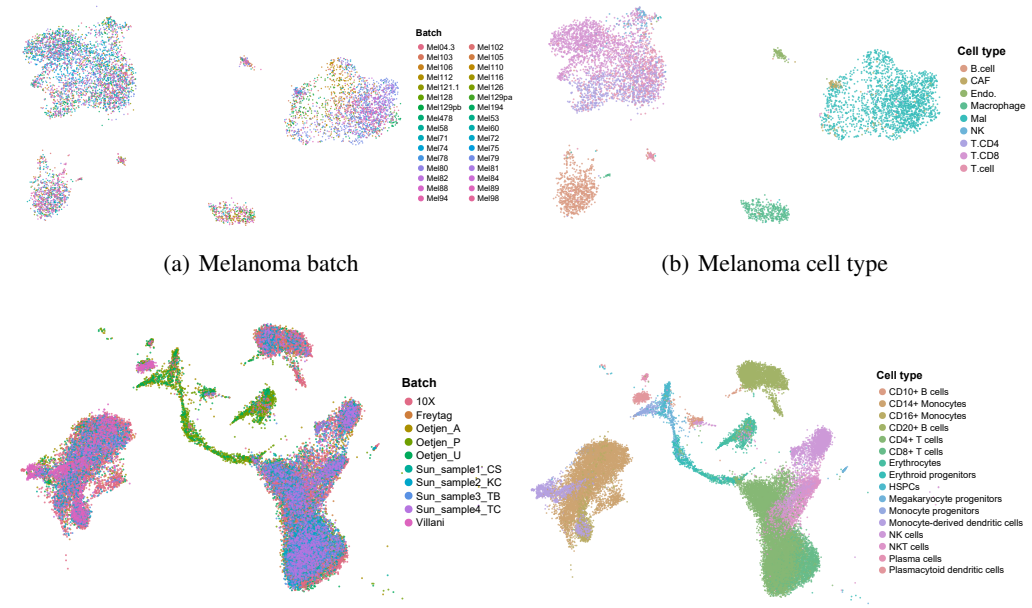


Figure 11: Norman response-retrieval examples for the *CNN1+MAPK1* and *MAPK1+PRTG* results. Squares mark scPathVQ retrievals and crosses mark source conditions; background colors denote train, validation, test, and unseen conditions. Because the pool contains measured combinations, these panels are illustrative response-space diagnostics.



(a) Melanoma batch

(b) Melanoma cell type

(c) Immune batch

(d) Immune cell type

Figure 12: Supplementary PCA visualizations for multi-batch integration on Melanoma and Immune. For each dataset, the left panel colors the shared coordinates by acquisition batch and the right panel colors the same coordinates by cell type. The paired views expose batch mixing and biological-identity preservation using one representation and one projection per dataset.

D.4 Gene Regulatory Network Inference

We examine gene regulatory network inference through an attention-based gene-association case study on hESC. The BEELINE hESC expression matrix contains 758 cells and 17,735 genes (Pratapa et al., 2020). Following the module-level network construction used by scDiVa (Wang et al., 2026b, Appendix E.5), we use scPathVQ attention scores as edge weights and visualize associations within selected gene modules. Figure 13 presents modules associated with *KDM2B*, *MAX*, *TFAP2A*, and *FOXA2*. Nodes represent genes, light blue edges show the module associations, and highlighted edges display selected strong associations with their scores.

The four modules contain overlapping gene neighborhoods with distinct local connections. *BCOR*, *TFAP2A*, and *FOXA2* recur across all four views, linking the selected modules through shared genes. The *FOXA2*-associated module highlights the *FOXA2–CER1* connection with a score of 0.66, while the *TFAP2A*-associated module highlights *TFAP2A–HAPLN1* at 0.66. These examples show how the learned attention structure organizes gene associations into connected modules. The networks provide a functional view of model-derived associations, with edge weights expressing association strength rather than activation or repression.

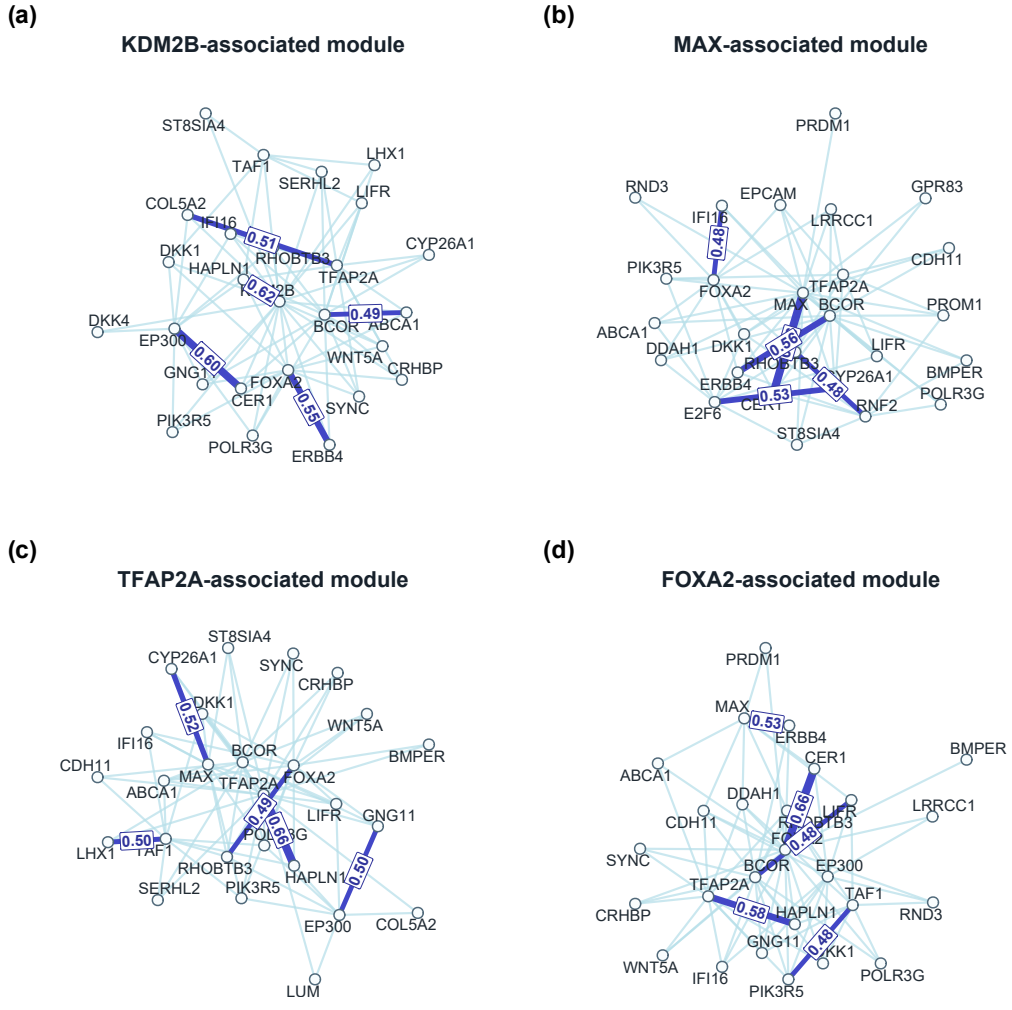


Figure 13: Attention-based gene-association networks in hESC. Panels show modules associated with (a) *KDM2B*, (b) *MAX*, (c) *TFAP2A*, and (d) *FOXA2*, following the module-level visualization used by scDiVa (Wang et al., 2026b, Appendix E.5). Nodes denote genes and edges represent associations derived from scPathVQ attention scores. Light blue edges show the module structure; thick blue edges highlight selected strong associations, with their scores annotated.

E MODEL ANALYSIS ON GENE PATHWAYS

We extend the cell state inference results with pathway-response comparisons and model diagnostics. The main text presents attention-defined cell states under Norman *BAK1* activation (Figure 6) and expression-derived pathway responses across Norman perturbations (Figure 5). Here, we provide the Adamson and Dixit heatmaps, agreement statistics across all three datasets, fusion-attention diagnostics, and pathway-input occlusion analyses. The condition-level split, pathway collection, and state and response readouts are specified in Appendix C.4.

The activity analysis included every evaluable held-out condition. For the attention and occlusion diagnostics, we sampled 64 control cells per dataset with random seed 1. Attention statistics were first computed for each cell and then averaged over cells, whereas occlusion used the mean quantized control-token sequence as input to the trained latent-transition head. We restricted occlusion to the 20 pathways with the largest mean absolute predicted activity and to the three test conditions with the largest ℓ_2 norm of the predicted gene-level response. These restrictions define the candidate set visualized in Figure 16(c,d); they do not affect the cross-dataset activity correlations.

Observed and predicted pathway activity. Figure 14 compares measured and generated responses on Adamson and Dixit using the expression-derived pathway score in Equation 40. The corresponding Norman comparison appears in Figure 5. Each display uses the activity-selected pathways and high-response conditions defined in Appendix C.4; the correlation analysis includes all evaluable pathway–condition pairs.

Across all evaluable pathway–condition pairs ($n = 7{,}304$, $1{,}680$, and $33{,}708$ for Adamson, Dixit, and Norman), the Pearson correlations between observed and predicted expression-derived scores were 0.697, 0.368, and 0.700, respectively; the corresponding Spearman correlations were 0.677, 0.381, and 0.641. The stronger agreement in Adamson and Norman and the weaker agreement in Dixit show that agreement between the two score sets was dataset dependent, even though the predicted maps retained structured condition-specific patterns.

Fusion attention as a selectivity diagnostic. We next quantified the selectivity of the pathway-to-HVG fusion block through attention concentration, a diagnostic of routing selectivity distinct from the cell-state readout in Appendix C.4. Each pathway-induced patch provides a query, and the individual HVG tokens provide cell-specific keys and values. Because the attention weights for each query sum to one, their raw mean across keys cannot rank pathway queries. Let $a_{i,h,r,j}$ denote the attention from pathway-induced patch r to valid HVG key j for control cell i and head h, and let $W_{r,p}$ be the learned composition weight from pathway p to patch r. We define patch-level attention concentration and its pathway aggregation as

$$
\begin{aligned}
C_r &= \frac{1}{|\mathcal{I}^A_{\mathrm{ctrl}}|H} \sum_{i \in \mathcal{I}^A_{\mathrm{ctrl}}} \sum_{h=1}^{H} \left(1 + \frac{\sum_{j \in \mathcal{K}_i} a_{i,h,r,j} \log a_{i,h,r,j}}{\log \max(|\mathcal{K}_i|, 2)}\right), \\
\widetilde{W}_{r,p} &= \frac{W_{r,p}}{\sum_{r'} W_{r',p}}, \qquad C_p = \sum_r \widetilde{W}_{r,p} C_r,
\end{aligned}
\tag{41}
$$

where $\mathcal{K}_i$ is the set of valid HVG keys and H is the number of attention heads. This is one minus normalized attention entropy, averaged over heads and the 64 control cells and then mapped from patches to pathways. Figure 15 reports C_p for the activity-selected pathways. Higher concentration indicates more selective routing of HVG evidence to a pathway-induced query, but does not establish pathway activity or causal importance. Across the 60 pathway–condition pairs per dataset used for occlusion, attention concentration did not align positively with the occlusion-based prediction shift: the descriptive Pearson correlations were -0.433, -0.409, and -0.086 for Adamson, Dixit, and Norman, respectively. We therefore treat attention concentration as a routing-selectivity diagnostic rather than as the primary attribution score.

Input occlusion and attribution faithfulness. To test whether pathway-associated inputs affected the generated response, we masked the member genes of one pathway before the frozen VQ encoder, recomputed the quantized control tokens, and reran the trained latent-transition head under the same perturbation condition. For a baseline predicted response $\widehat{\boldsymbol{\Delta}}_c$ and a masked response $\widehat{\boldsymbol{\Delta}}_{c,\setminus p}$, the

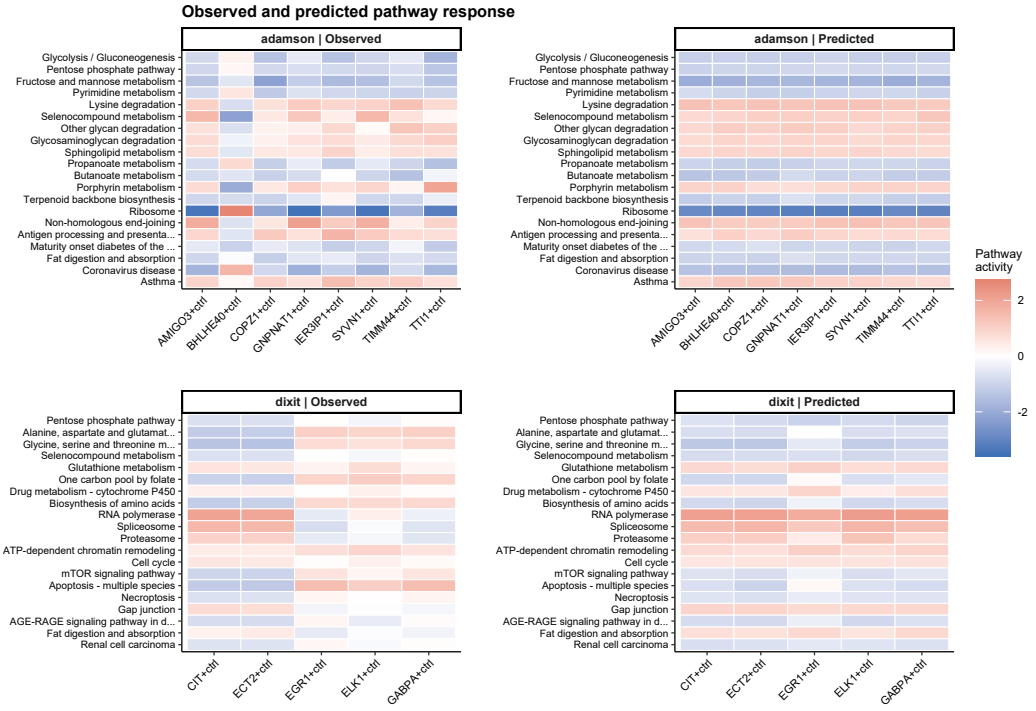


Figure 14: Observed and predicted pathway responses on Adamson and Dixit. Each row is a pathway and each column is a held-out perturbation condition. Observed panels use measured control-relative responses, and predicted panels use expression changes generated from the hard post-VQ sequence. Colors show the pathway–background contrast defined in Equation 40. Each dataset displays the 20 pathways with the largest mean absolute predicted score and up to eight conditions with the largest mean absolute predicted response over those pathways. The corresponding Norman heatmap is shown in Figure 5; correlations use all evaluable pathway–condition pairs.

pathway input-occlusion shift was

$$S_{c,p} = \frac{\left\| \widehat{\mathbf{\Delta}}_c - \widehat{\mathbf{\Delta}}_{c,\setminus p} \right\|_2}{\max\left(\left\| \widehat{\mathbf{\Delta}}_c \right\|_2, \epsilon \right)}. \tag{42}$$

This intervention measures the sensitivity of the model prediction to pathway-associated input features; it is not equivalent to a causal pathway knockout. Figure 16(c) shows the resulting shifts for the ten most sensitive pathways within the 20-pathway candidate set and the three highest-predicted-response test conditions in each dataset. The largest shifts were concentrated in a subset of signaling, cell-cycle, proteostasis, and disease-associated pathway gene sets, with the largest normalized shifts generally occurring in Dixit.

We assessed ranking faithfulness by progressively masking the top-k pathways ordered by $S_{c,p}$ and comparing the resulting prediction shift with one seed-controlled random permutation of the same candidate pathway set for each condition. Figure 16(d) shows the mean and standard error across the three highest-predicted-response test conditions per dataset for $k \in \{1, 3, 5, 10, 20\}$. Occlusion-ranked pathways produced larger shifts than the matched random ranking at low and intermediate k, most clearly in Dixit and Norman. The curves coincide at $k = 20$ because both rankings have then masked the same complete 20-pathway candidate set; gene overlap between pathways also makes successive shifts non-additive. Under this matched comparison, the occlusion ranking provides a descriptive diagnostic of model-sensitive pathway inputs rather than causal pathway importance.

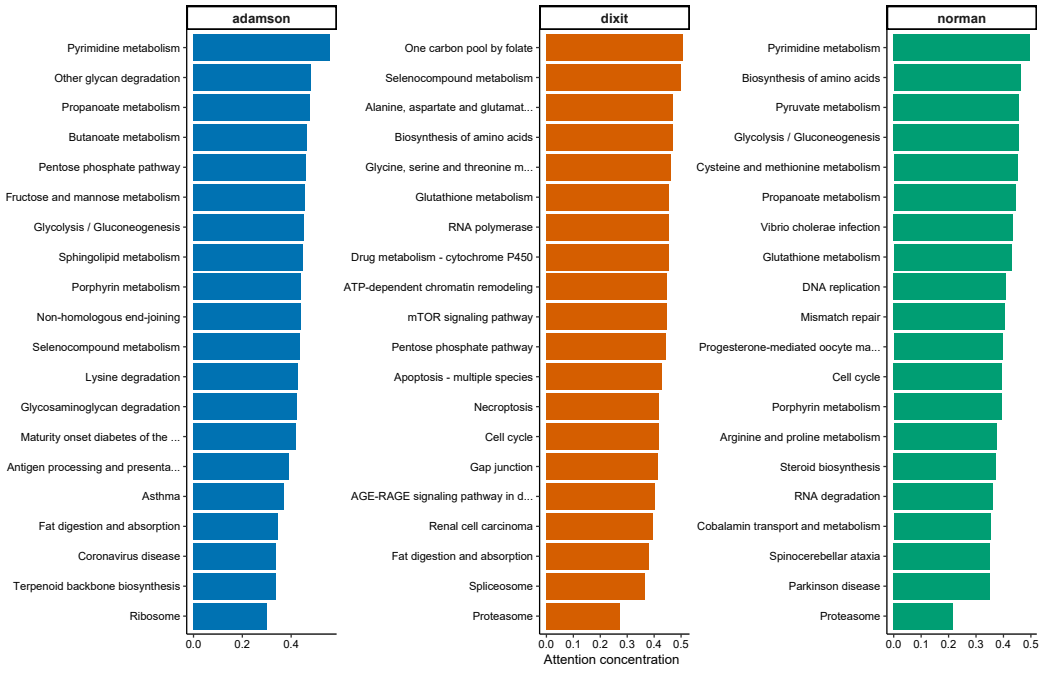


Figure 15: Fusion-attention selectivity diagnostic. Bars show pathway-level normalized attention concentration C_p from Equation 41, averaged over attention heads and 64 control cells and aggregated using normalized pathway–patch composition weights. The displayed pathways are the 20 activity-selected pathways for each dataset. This quantity measures routing selectivity and is not interpreted as pathway activity or causal pathway importance.

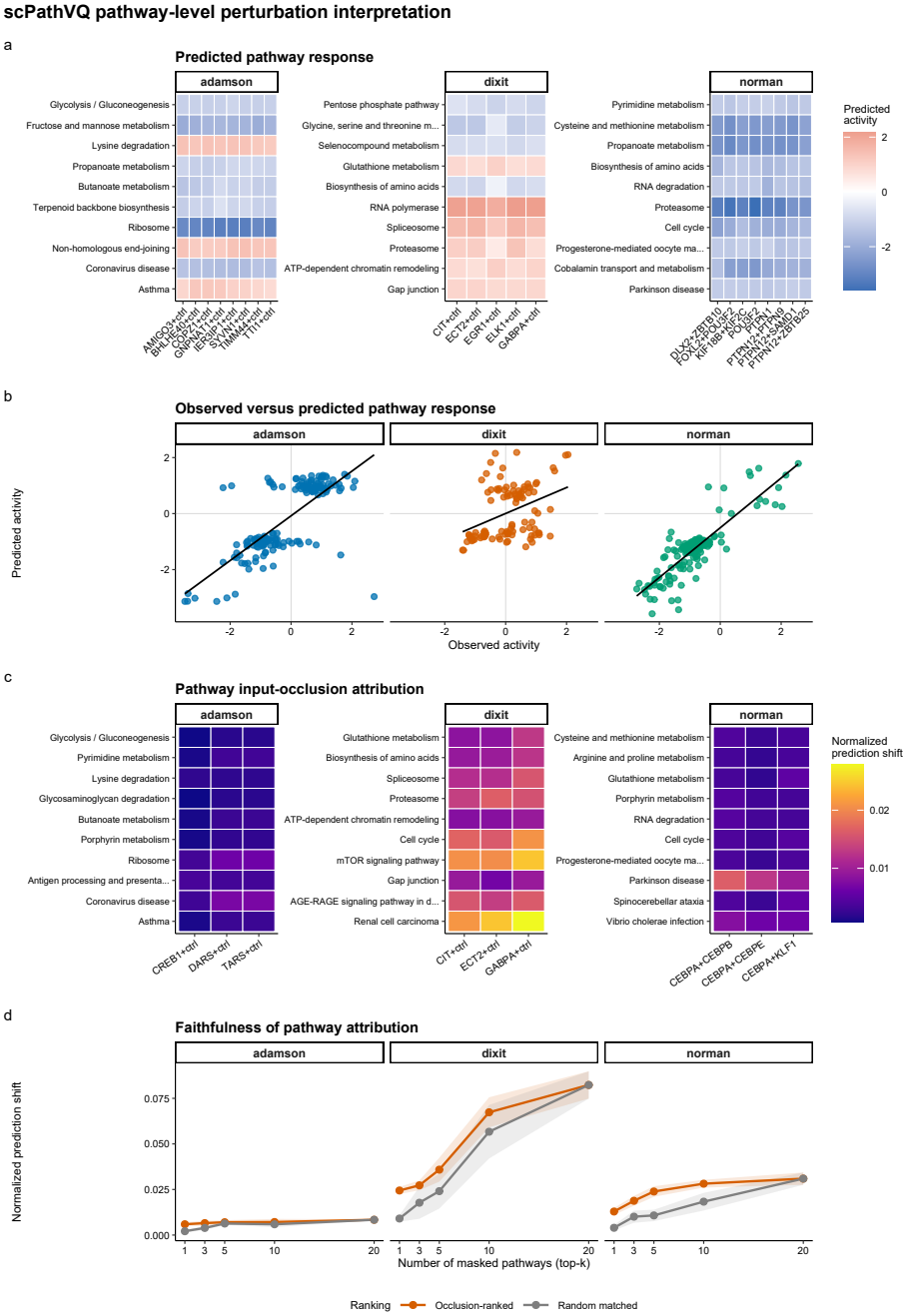


Figure 16: Integrated pathway-level model analysis. (a) Predicted standardized pathway–background responses for the ten pathways with the largest mean absolute predicted activity in each dataset. (b) Observed versus predicted expression-derived pathway-response scores for the 20 activity-selected pathways and up to eight high-response test conditions; each point is one pathway–condition pair, and the lines are dataset-specific least-squares fits. (c) Normalized prediction shifts after masking pathway genes before the frozen VQ encoder and rerunning the latent-transition head, shown for the ten most sensitive pathways within the candidate set and the three test conditions with the largest predicted-response norm in each dataset. (d) Faithfulness curves obtained by progressively masking occlusion-ranked pathways or one seed-controlled random permutation of the same candidate set per condition; lines show means across the three conditions, and ribbons show ±SEM. Occlusion measures model input sensitivity rather than causal pathway perturbation.