LeCellModel: Interpretable Density Estimation over the Gene Expression Manifold

Gil Karin* **Artemii Bakulin*** **Nir Yosef**
Weizmann Institute of Science
{gil.karin, artemii.bakulin, nir.yosef}@weizmann.ac.il

Abstract

Representation learning for single-cell RNA-seq has been evaluated by the identities representations capture — cell types, lineages, perturbations — rather than how faithfully they model the distribution of cell states on the gene expression manifold. As a result, density estimation, rare-state detection, perturbation scoring, and gene-level attribution are bolted onto frozen embeddings as separate models, disconnected from the geometry of the data. We argue that modeling the distribution of cell states should be the primary objective, and that representation quality follows from it. Building on LeJEPA, whose SIGReg objective enforces a provably optimal isotropic Gaussian latent, we introduce LeCellModel, which reads the encoder's Jacobian log-volume (JEPA-SCORE) under this Gaussian latent as a relative typicality score. Across several datasets, LeCellModel outperforms established methods on the scIB benchmark for representation quality. Furthermore, we show that LeCellModel enables detection of rare expression states using JEPA-SCORE as a typicality measure, applying it to a controlled cytokine panel, where it identifies stimulated cells across minority fractions and tracks pathway-response magnitude. We introduce gene-level interpretability of the typicality score — to our knowledge, the first method for analyzing JEPA-SCORE in feature space. Leveraging this framework, LeCellModel recapitulates the known association of a profibrotic macrophage program with idiopathic pulmonary fibrosis, and discovers a $CD177^{+}$ activated regulatory T cell subpopulation previously described only in tumor contexts. Both findings emerge directly from the geometry of the learned embedding, without relying on supervised annotations. The code is available at `https://github.com/gilk717/lecellmodel_paper`.

1 Introduction

Single-cell RNA sequencing measures gene expression across thousands of individual cells, producing high-dimensional count data that capture the molecular state of each cell. A central computational task is to learn representations of these states that organize cells into biologically meaningful structure: separating cell types, resolving developmental trajectories, and exposing responses to perturbation. The dominant approach has been to train encoder-decoder models with reconstruction objectives, from variational autoencoders such as scVI [Lopez et al., 2018] and scANVI [Xu et al., 2021] to BERT-style foundation models such as Geneformer [Theodoris et al., 2023] and scGPT [Cui et al., 2024]. Reconstruction, however, operates entirely in gene space, which encourages the encoder to retain nuisance variation such as batch effects and dropout patterns alongside biological signal. Adversarial batch correction methods like scDREAMER [Shree et al., 2023] partly offset this, but the adversary works against a reconstruction objective that itself rewards retaining batch-related variation to minimize error. Finally, there are emerging approaches, such as GeneJEPA [Litman et al., 2025] and CellJEPA [ElSheikh et al., 2026], following the Joint-Embedding Predictive Architecture (JEPA)

*Equal contribution.

paradigm that forgo the decoder entirely, shaping the embedding space through latent prediction rather than reconstruction. Crucially, all these methods are judged primarily by the identities their representations capture, rather than by how faithfully they describe the distribution of cell states.

A remarkable yet largely unexploited property of JEPAs is that, when the latent space is constrained to an isotropic Gaussian, the encoder provably estimates the density of the training distribution [Balestriero et al., 2025]. We are the first to leverage this property in a practical single-cell setting, introducing LeCellModel: a JEPA-based model for learning the distribution of cell states. LeCellModel builds on LeJEPA [Balestriero and LeCun, 2025], which unlike previous single-cell JEPAs avoids heuristics such as stop-gradient to prevent collapse, replacing them with a principled isotropic Gaussian latent enforced via Sketched Isotropic Gaussian Regularization (SIGReg) [Balestriero and LeCun, 2025]. In addition, rather than scaling to large transformer architectures, we show that a lightweight MLP encoder is sufficient for single-cell data, keeping the model tractable and the Jacobian computable. Across several datasets on the scIB benchmark [scverse contributors, 2024], LeCellModel achieves state-of-the-art representation quality, while its typicality score, read within a cell type, enables practical detection of rare cell states. Moreover, we show that differentiating the score with respect to input gene expression attributes local changes in typicality to individual genes, yielding per-gene attribution in the input's native coordinates. Thus, a single trained encoder serves simultaneously as a representation, a typicality score, and an attribution map.

Contributions.

- **A density-estimation strategy for cell states.** We propose a strategy for estimating the relative density of cell states that relies on JEPA-SCORE [Balestriero et al., 2025], the implicit density estimator of Gaussian-embedding JEPAs. Combined with biologically meaningful data augmentations and batch-effect controls, JEPA-SCORE yields a relative estimate of cell-state density on the learned manifold, which we read within a cell type as typicality, despite the stochastic, sparse, and batch-confounded nature of the data.
- **Differentiable gene-level attribution of density.** A key novelty of LeCellModel is that the density score is itself differentiable with respect to gene expression. We therefore differentiate it to identify, for each cell, the genes whose local expression changes most affect its JEPA-SCORE, turning a cell-level relative density estimate into direct gene-level attribution. Unlike differential expression, it requires no predefined contrast group.
- **LeCellModel: a Gaussian-latent JEPA for single-cell RNA-seq.** We instantiate this idea in a teacher-free, single-network JEPA combining a lightweight MLP backbone, gene-space augmentations, SIGReg [Balestriero and LeCun, 2025], and adversarial batch correction. The architecture is light enough to train per-dataset and to retain a tractable Jacobian, which is what makes the density readout practical (Section 2.3).
- **Cell identity preserved at atlas scale.** LeCellModel preserves cell identity while integrating across batches at atlas scale, leading scIB Total on HLCA Core (584K cells) and Heart v2 (704K cells, 32 preparations), and tying scVI on Pancreas; it also leads GeneJEPA trained on the same data and the foundation models scGPT and UCE (Section 3.2; App. K).
- **Density-driven discovery in disease tissue.** JEPA-SCORE recovers stimulated cells in a controlled cytokine panel and tracks pathway-response magnitude. Applied to idiopathic pulmonary fibrosis lung, it surfaces a profibrotic monocyte-derived macrophage program and a $CD177^{+}$ activated-Treg subpopulation previously described only in tumors — both identified purely from the geometry of the embedding (Sections 3.3–3.4).

2 Methods

LeCellModel is built on the Joint Embedding Predictive Architecture (JEPA) framework [Balestriero and LeCun, 2025]. JEPAs learn representations by requiring augmented views of the same input to be mutually predictable in embedding space — without reconstructing the input. For single-cell data, this is natural: since a cell's identity is encoded across many correlated genes, randomly masked and Poisson-resampled views retain enough signal to agree on the same cell state. In LeCellModel, all augmented views are trained to converge to a shared projection center; no momentum teacher or asymmetric predictor head is used. SIGReg regularizes the projection distribution toward an isotropic Gaussian — preventing collapse, enforcing diversity both across embedding dimensions and between cells, and enabling JEPA-score as a density proxy (Section 2.4). To handle an atlas-scale batch

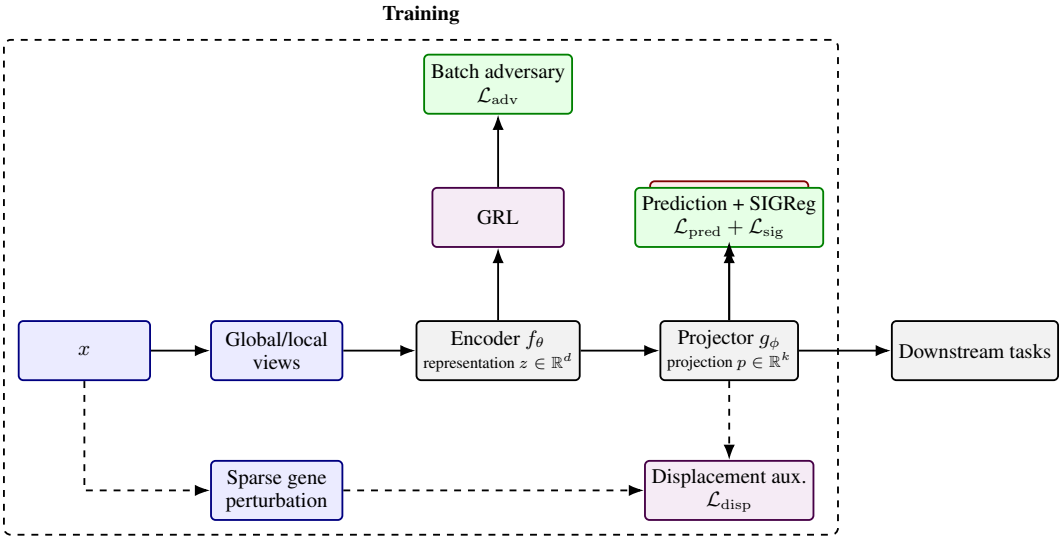


Figure 1: **Overview of LeCellModel.** Input cells are augmented into global and local views. The encoder f_θ maps each view to a representation $z \in \mathbb{R}^d$, and the projector g_ϕ maps z to a projection $p \in \mathbb{R}^k$. During training, batch supervision acts on z through an explicit gradient-reversal layer (GRL), JEPA prediction and SIGReg regularization act on p, and an optional displacement auxiliary predicts projection shifts induced by sparse gene-space perturbations applied to the input cell. The projection p can be used directly as an embedding for downstream tasks, while the JEPA-score is computed separately from the Jacobian of $F = g_\phi \circ f_\theta$.

structure, adversarial batch correction penalizes the encoder for retaining batch-predictive features, freeing it from encoding technical variation. Together, these components yield both reliable cell-state embeddings and a principled density readout (Figure 1).

2.1 Problem setup and notation

Let $x \in \mathbb{R}^G$ denote a cell's raw expression vector over G highly variable genes, $b \in \{1, \ldots, B\}$ its batch label, and $c \in \{1, \ldots, C\}$ an optional biological grouping (cell type, or a coarser grouping such as lineage). We train an encoder $f_\theta : \mathbb{R}^G \to \mathbb{R}^d$ and a projector $g_\phi : \mathbb{R}^d \to \mathbb{R}^k$, and write $F = g_\phi \circ f_\theta$ for their composition, so that $z = f_\theta(x)$ and $p = F(x)$. An adversary head $h_\psi : \mathbb{R}^d \to \mathbb{R}^B$ predicts batch from the representation. Encoder, projector, and adversary head are all lightweight two-layer MLPs; full architectures are in App. C.

2.2 View generation and augmentation

Each cell yields $n_g = 1$ global and $n_\ell = 6$ local views through biologically motivated augmentations on raw counts: Poisson resampling (sequencing-depth stochasticity), library-size normalization followed by $\log(1 + x)$, and random gene masking at rates $m_g = 0$ (globals) and $m_\ell = 0.6$ (locals, more aggressively perturbed). Poisson resampling is omitted on the atlas-scale benchmarks (HLCA Core, Heart v2). The global view is left unmasked so that it serves as an information-rich target for the masked local views (Eq. 1); two global views, masked less than the local views, give similar scIB scores (App. I).

2.3 Training objective

2.3.1 JEPA prediction loss

Let p^g denote the projection of the global view and $\{p_j^\ell\}_{j=1}^{n_\ell}$ the projections of the local views. We define the prediction target as $\hat{c} = p^g$ (more generally, $\hat{c} = \frac{1}{n_g}\sum_{i=1}^{n_g} p_i^g$ when $n_g > 1$) and minimize

$$\mathcal{L}_{\text{pred}} = \frac{1}{n_g + n_\ell} \sum_v \|p_v - \hat{c}\|^2, \tag{1}$$

averaged over the batch, where the sum runs over all $n_g + n_\ell$ view projections. Unlike I-JEPA and prior single-cell JEPAs, LeCellModel has no momentum/EMA teacher and no asymmetric predictor head: the same encoder–projector produces both $\hat{c}$ and the predicting views, and gradients flow symmetrically through both with no stop-gradient on $\hat{c}$. Collapse prevention is provided entirely by SIGReg (Eq. 2): without it, a constant projection minimizes $\mathcal{L}_{\text{pred}}$.

2.3.2 SIGReg regularization

Following LeJEPA, we regularize the projection distribution toward an isotropic Gaussian $\mathcal{N}(0, I_k)$ using an Epps–Pulley characteristic-function test along $M = 1024$ random one-dimensional projections, evaluated per view-slot and averaged:

$$\mathcal{L}_{\text{sig}} = \frac{1}{(n_g + n_\ell)\, M} \sum_v \sum_{m=1}^{M} T_{\text{EP}}^{(v)}(\omega_m). \tag{2}$$

The empirical CF, weighting, truncation, and numerical integration are given in App. E. The isotropic Gaussian target also fixes the geometry under which the Jacobian log-volume is read as relative density [Balestriero et al., 2025]; replacing SIGReg with VICReg [Bardes et al., 2022], which constrains only the first two moments, trains less stably and weakens both scIB and the score (Section 4).

2.3.3 Adversarial batch correction

We train h_ψ to predict the batch label b from the representation through a gradient-reversal layer GRL_λ [Ganin and Lempitsky, 2015]:

$$\mathcal{L}_{\text{adv}} = \text{CE}\big(h_\psi(\text{GRL}_\lambda(z)), b\big), \tag{3}$$

with λ ramping from 0 over a warmup. A conditional variant that preserves coarse biological structure and two stabilizers (spectral normalization on h_ψ, label smoothing in cross-entropy) are described in App. F. SIGReg does not integrate batches; removing the adversary lowers Total scIB by 0.09–0.20, with the largest drop in Batch (App. I).

2.3.4 Displacement auxiliary

View prediction primarily constrains the projector along directions induced by the augmentations, leaving sensitivity to other local gene-space perturbations under-determined. We therefore optionally add a displacement auxiliary: a small predictor estimates the projection shift $\Delta p = F(x') - p$ induced by a sparse additive gene-space perturbation, trained with a combined cosine-direction and magnitude loss on a 25% batch subsample after warmup. This provides a modest boost of scIB (+0.02 Total on atlas-scale datasets), but regularizing the Jacobian may reduce the precision of the JEPA-score as a typicality signal; the loss form, perturbation procedure, and ablation are in App. H.

2.3.5 Combined objective

The final training loss is

$$\mathcal{L} = \mathcal{L}_{\text{pred}} + \lambda_{\text{sig}}\,\mathcal{L}_{\text{sig}} + \lambda_{\text{adv}}\,\mathcal{L}_{\text{adv}} + \lambda_{\text{disp}}\,\mathcal{L}_{\text{disp}}. \tag{4}$$

The encoder parameters minimize $\mathcal{L}$ via the GRL; adversary parameters minimize $\mathcal{L}_{\text{adv}}$ directly. The coefficients λ_{adv} and λ_{disp} ramp linearly from 0 over a warmup; λ_{sig} is fixed. Defaults are in the appendix.

2.4 JEPA-score

Definition. For a trained model $F = g_\phi \circ f_\theta$, we define the per-cell score

$$S(x) \;=\; \sum_{i=1}^{k} \log \max\big(\sigma_i(J(x)),\, \varepsilon\big), \qquad J(x) = \frac{\partial F(x)}{\partial x} \in \mathbb{R}^{k \times G}, \tag{5}$$

where $\sigma_i(J(x))$ are the singular values of $J(x)$ and $\varepsilon = 10^{-7}$ floors near-zero singular values for numerical stability of the logarithm. The Jacobian is taken on the composed map $F = g_\phi \circ f_\theta$, i.e. with respect to the projector output.

Interpretation. Balestriero et al. [2025] showed that, under isotropic Gaussian regularization, the Jacobian log-volume of a trained JEPA approximates $\log p(x)$ at optimality, validated globally on synthetic and image data. Single-cell data departs from the assumptions of this result (sparse counts, masking and resampling views, adversarial batch correction), so we read $S(x)$ as a relative rather than a calibrated log-density. Cells also form a structured manifold of related types with very uneven abundances: across types the score tracks each type's abundance in the dataset, which reflects sampling as well as biology (Section 4), and the biologically interesting signal is rarely a globally rare cell but a coherent rare *program* inside a known type. We therefore read the score within a cell type, as typicality.

Two mechanisms drive low scores onto rare-and-predictive states. SIGReg's empirical statistic is averaged across the batch, so a sub-population influences it only in proportion to its mixture weight; the rare component is therefore less strongly constrained by the global Gaussian-matching objective. If, additionally, a rare state's cross-view predictability is carried by a compact gene program, the prediction loss is satisfied with few features and points the same way. Rare-and-predictive states therefore concentrate onto few dominant Jacobian directions, yielding a low log-volume score; we confirm this empirically in Section 4, and Section 3.4 uses the score to localize disease-enriched programs and reads them gene-by-gene through $\nabla_x S$.

Computation. The stabilized score is computed in a single vectorized pass using composable functional transforms (`vmap`, `jacrev`, `svdvals`); details in App. G. For a transformer the size of GeneJEPA (over 100M parameters), the per-cell Jacobian costs roughly $2{,}000\times$ more than for our MLP, which makes atlas-scale scoring impractical.

2.5 Gene-level gradient attribution

For a low-scoring cell x, we compute the gradient of the score with respect to gene expression,

$$\nabla_x S(x) \;=\; \nabla_x \left[\sum_{i=1}^{k} \log \max\left(\sigma_i\left(\frac{\partial F(x)}{\partial x} \right), \varepsilon \right) \right] \;\in\; \mathbb{R}^G, \tag{6}$$

implemented by second-order automatic differentiation, composing the Jacobian (via `jacrev`) with an outer `grad` over the stabilized log-volume.

Each entry of $\nabla_x S(x)$ measures the local sensitivity of $S(x)$ to the corresponding gene's expression; negative entries identify genes whose expression locally lowers the score.

3 Experiments

3.1 Datasets and evaluation

Datasets. We evaluate LeCellModel on three integration benchmarks that span orders of magnitude in scale and differ in the dominant source of technical variation: the Human Lung Cell Atlas (HLCA) Core [Sikkema et al., 2023] (584,944 cells, 14 studies); the Heart Cell Atlas v2 [Litviňuková et al., 2020] (704,296 cells, 22 donors across 3 sequencing modalities, 32 preparations total); and a multi-study human pancreas benchmark [Luecken et al., 2022] (16,382 cells, 9 batches spanning six sequencing protocols). For rare-program discovery we additionally assemble two disease-targeted subsets drawn from the extended HLCA release [Sikkema et al., 2023], combining healthy and IPF donors at roughly 8% disease cells each: the *HLCA monocyte–macrophage* subset (257,893 cells, 125 donors, 13 studies, 20,313 IPF cells) and the *HLCA lymphoid* subset (159,652 cells, 171 donors,

Table 1: Batch-integration performance across the three benchmark datasets, mean ± standard deviation across 4 random seeds (42, 1, 2, 3). scIB aggregates per method's output embedding (definitions in Section 3.2 preamble); bold marks the best per-dataset entry; per-metric tables in Appendix L.

	HLCA Core			Heart v2			Pancreas		
Method	Bio	Batch	Total	Bio	Batch	Total	Bio	Batch	Total
PCA	0.682±.008	0.332±.003	0.542±.006	0.763±.006	0.314±.002	0.583±.003	0.638±.000	0.254±.000	0.485±.000
Scanorama	0.707±.000	0.388±.017	0.580±.006	0.762±.000	0.342±.009	0.594±.004	0.770±.004	0.556±.006	0.684±.004
Harmony	0.685±.007	0.516±.013	0.618±.009	0.843±.005	0.554±.007	0.727±.004	0.727±.001	0.587±.000	0.671±.001
scVI	**0.729**±.001	0.551±.006	0.658±.002	0.818±.001	0.543±.005	0.708±.002	**0.831**±.003	0.594±.007	0.736±.003
LeCellModel	0.727±.005	**0.632**±.003	**0.689**±.003	**0.849**±.002	**0.627**±.001	**0.760**±.002	0.807±.010	**0.640**±.007	**0.741**±.008

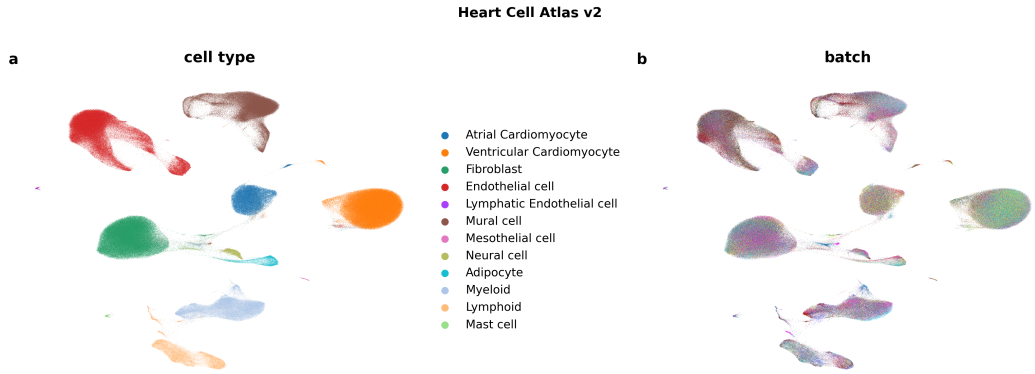


Figure 2: **LeCellModel embedding of the Heart Cell Atlas v2.** UMAP of 704,296 cells encoded by LeCellModel. **(a)** Colored by cell type. Twelve major cardiac populations form well-separated clusters, preserving biological structure. **(b)** Colored by batch (32 preparations across scRNA-seq, snRNA-seq, and 10x Multiome). Cells from different batches mix within each cluster, demonstrating effective correction of multi-modality batch effects.

19 studies, 10,913 IPF cells); subsetting criteria are detailed in Appendix B. The Parse Biosciences cytokine panel is described in Section 3.3.

Baselines. We compare LeCellModel to four standard baselines: scVI [Lopez et al., 2018], a variational autoencoder with a negative-binomial likelihood and an explicit batch-conditional decoder; Harmony [Korsunsky et al., 2019], an iterative batch-correction method operating in PCA space; Scanorama [Hie et al., 2019], a mutual-nearest-neighbor alignment in reduced space; and raw PCA as an unintegrated lower bound. We also evaluated GeneJEPA [Litman et al., 2025] (trained from scratch and fine-tuned), scGPT [Cui et al., 2024] (zero-shot and fine-tuned) and UCE [Rosen et al., 2026] (zero-shot). None was competitive: GeneJEPA scored below PCA, and scGPT and UCE below Harmony; single-run results are in Appendix K.

Batch-integration metrics. We follow the `scib_metrics` [scverse contributors, 2024] reimplementation of scIB [Luecken et al., 2022] and report ten metrics: *bio-conservation* — Isolated Labels, Leiden [Traag et al., 2019] NMI, Leiden ARI, Silhouette Label, cLISI; *batch correction* — BRAS [Rautenstrauch and Ohler, 2025], iLISI, kBET, Graph Connectivity, PCR comparison. Bio and Batch are the arithmetic means of each group; Total is their scIB-convention weighted average (0.6 Bio, 0.4 Batch). All methods are evaluated across four random seeds; hyperparameters are listed in the appendix.

3.2 Dataset integrations

Table 1 reports the Bio-Conservation, Batch-Correction, and Total aggregates for LeCellModel and all baselines on the three benchmark datasets; per-metric breakdowns are in Appendix L.

Summary. **LeCellModel attains the highest Total score on all three benchmarks**, with the lead at atlas scale being many times larger than the cross-seed standard deviation: HLCA Core $\Delta = +0.031$ over scVI, Heart v2 $\Delta = +0.033$ over Harmony. On small-scale Pancreas, LeCellModel remains competitive, tying scVI on Total (0.741 ± 0.008 vs. 0.736 ± 0.003) with LeCellModel leading Batch-Correction and scVI leading Bio-Conservation — a regime where most atlas-scale methods give up ground. scGPT, UCE and GeneJEPA trail LeCellModel on both Bio and Batch (App. K).

3.3 Validating rare-state detection with ground truth

The JEPA-score targets unsupervised discovery, so its validation requires a setting where a rarity label exists but is withheld from training. The Parse Biosciences PBMC cytokine-stimulation panel [Parse Biosciences, 2026] provides this: stimulated and resting cells are independently labeled by exposure condition — though within the stimulated population the magnitude and character of the response varies, and not every labeled-stimulated cell exhibits a detectable cytokine signature. We subsample stimulated cells to fractions $f \in \{0.01, 0.03, 0.05, 0.10\}$ of the training mix and train without using the label. We report AUROC and AUPRC against the held-out label, and Spearman ρ between the score and the canonical pathway signature (STAT3 targets for IL-10; STAT6 for IL-4) within stimulated cells only — asking whether the score also ranks cells by *degree* of response, not merely stim vs. resting.

We evaluate four (cytokine, cell type) pairs spanning a range of effect sizes: IL-10 × CD16 Mono, IL-10 × CD14 Mono, IL-4 × CD14 Mono, and IL-4 × CD16 Mono (weaker IL-4 response; pair descriptions and cell counts are in Appendix N). Baselines are seven density- and reconstruction-based anomaly scores — PCA-kNN, PCA-Mahal, PCA-IForest, scVI-IForest, scVI-kNN, scVI-Mahal, and scVI-recon — all fit without labels.

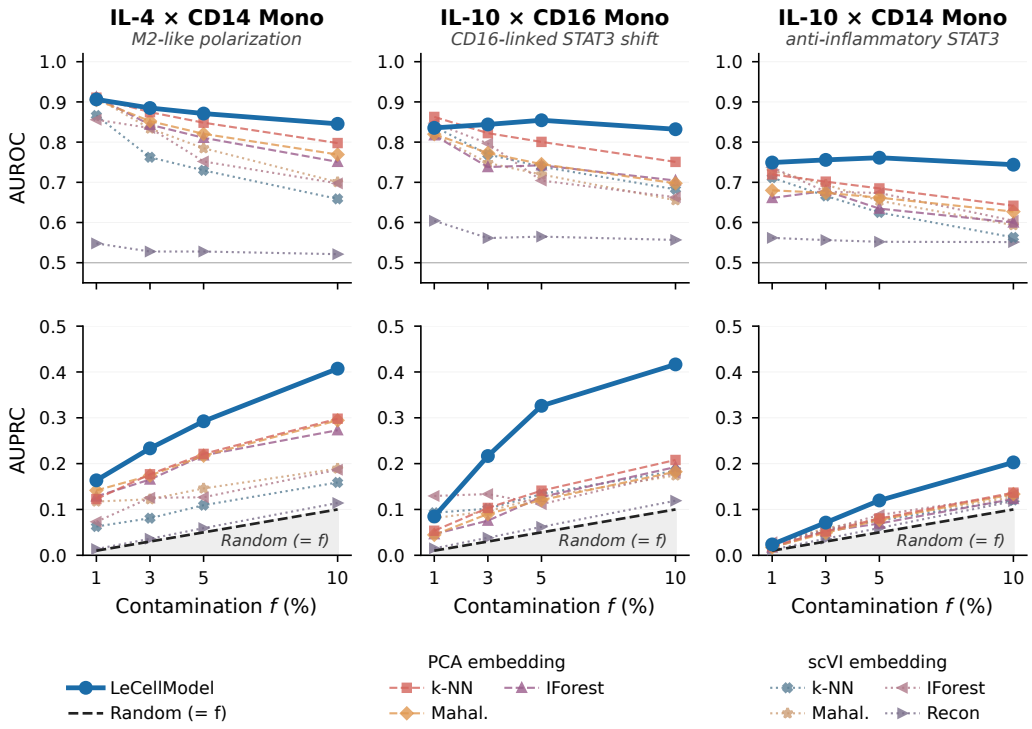


Figure 3: **Rare-state detection benchmark.** Stimulated cells at minority fractions $f \in \{1\%, 3\%, 5\%, 10\%\}$; no label seen during training. AUROC (top) and AUPRC (bottom). Three pairs have established biological responses; the weaker IL-4 response in CD16 monocytes is reported in Appendix M.

Each density-based baseline rests on a single geometric assumption, and as f grows, the stimulated minority inflates these density estimates and performance deteriorates substantially (e.g. scVI-kNN drops $0.839 \rightarrow 0.683$ AUROC on IL-10 $\times$ CD16). The JEPA score remains stable across f, leading on all three established-response pairs in both AUROC and AUPRC beyond $f = 1\%$ (Figure 3; Tables 12, 13). Computing the same kinds of scores in LeCellModel's own embedding does not reproduce this: on the three established-response pairs, kNN density and Mahalanobis distance fall below chance, and isolation forest trails JEPA-SCORE, coming close only on IL-4 $\times$ CD14 at low f (Table 15).

Within stimulated cells, the JEPA-score correlates with response *magnitude* (Spearman ρ against the pathway signature, z-scored on resting cells, without access to the gene list); LeCellModel leads on the biologically stronger pairs (Table 14; pair biology in App. N).

3.4 Rare program discovery

Within a cell type, low-JEPA-score cells flag rare-and-predictive states (Section 2.4). We can recover the gene program of such a region from two complementary directions: **gene** $\rightarrow$ **cells**, by finding genes whose expression spatially co-locates with low scores on the embedding kNN graph (bivariate Moran's I; App. O, Table 16), or **cells** $\rightarrow$ **genes**, by taking a low-score neighborhood and ranking genes by their local effect on the score ($\nabla_x S$; Section 2.5). The two views converge on the program where they overlap, but $\nabla_x S$ operates per-cell, surfacing score *drivers* rather than co-occurring markers — a distinction no population-level correlation can make. We apply this to the HLCA monocyte–macrophage cohort (13 studies) and the HLCA lymphoid normal+IPF cohort (19 studies); batch correction across these contributors is what makes a shared program recoverable.

Monocyte-derived macrophages. Within monocyte-derived macrophages, LeCellModel isolates a low-JEPA-score subpopulation corresponding to a profibrotic macrophage program (SPP1, CHI3L1, CHIT1, ADAMDEC1) [Morse et al., 2019], consistent with the fibrosis-associated SPP1$^+$ state highlighted in the HLCA disease analysis [Sikkema et al., 2023] and further extended here by matrix-remodeling and immunosuppressive effectors (CTSK, SIGLEC15). This population is enriched for IPF cells (31.7% vs. 12.5% baseline within monocyte-derived macrophages; 2.53-fold enrichment), spanning 53 donors (16 IPF).

CD4 T cells. Within CD4 T cells, LeCellModel isolates a low-JEPA-score population defined by FOXP3, CD177, CTLA4, CCR8, and LAYN — the *CD177$^+$ activated-Treg* phenotype previously characterized in tumor microenvironments [Kim et al., 2021]. Its appearance in *non-cancerous*, fibrotic IPF lung is, to our knowledge, previously unreported. LeCellModel identifies it from embedding geometry alone: Tregs appear only in a single research group's per-study labels (Meyer 2019/2021) — and not at all in HLCA's harmonized vocabulary — so a label-based analysis of this cohort would likely have missed the population entirely. The recovered program overlaps strongly with the CD177$^+$ tumor-infiltrating Treg state described by Kim et al. [2021], including the highly activated and suppressive phenotype associated with recent tissue infiltration. It is also strongly enriched for IPF cells (34.7% vs. 5.3% baseline within CD4 T cells; 6.57-fold enrichment), spanning 95 donors (17 IPF). Restricting this cluster to CD177$^+$ cells further strengthens the disease association, raising the IPF fraction to 56.9% (10.76-fold over the CD4 baseline), across 51 donors (13 IPF) (Figure 4).

Gene-level attribution. Given a low-JEPA-score subpopulation C, we rank genes by the absolute cluster-mean input$\times$gradient, $|\mathbb{E}_{x \in C}[x \odot \nabla_x S]|$, where the per-cell gradient $\nabla_x S(x) \in \mathbb{R}^G$ is defined in Section 2.5. Unlike integrated gradients on reconstruction or classification losses, which tend toward cell-type markers, this highlights genes that shape a cell's *local typicality and rare-state geometry* — extending the JEPA-score from a per-cell anomaly detector to a gene-level interpretation tool without any predefined gene list. In CD4 T cells, CD177 and CCR8 carry the two largest raw-gradient magnitudes ($|\mathbb{E}_{x \in C}[\partial S / \partial x]| \approx 11.0, 9.4$) — the score is most locally sensitive to the defining markers of the activated-Treg subpopulation. In monocyte-derived macrophages, the gradient elevates matrix-remodeling and checkpoint effectors (MMP9, CTSK, SIGLEC15), while SPP1 — the most abundant marker of the recovered program — falls outside the gradient top: the gradient tends to surface what locally drives the score rather than general markers. The ranking recovers known pathway targets and complements differential expression: on the CD14 cytokine pairs it places the STAT3 and STAT6 targets F13A1 and ALOX15 first, and in IPF it ranks the program genes CD177

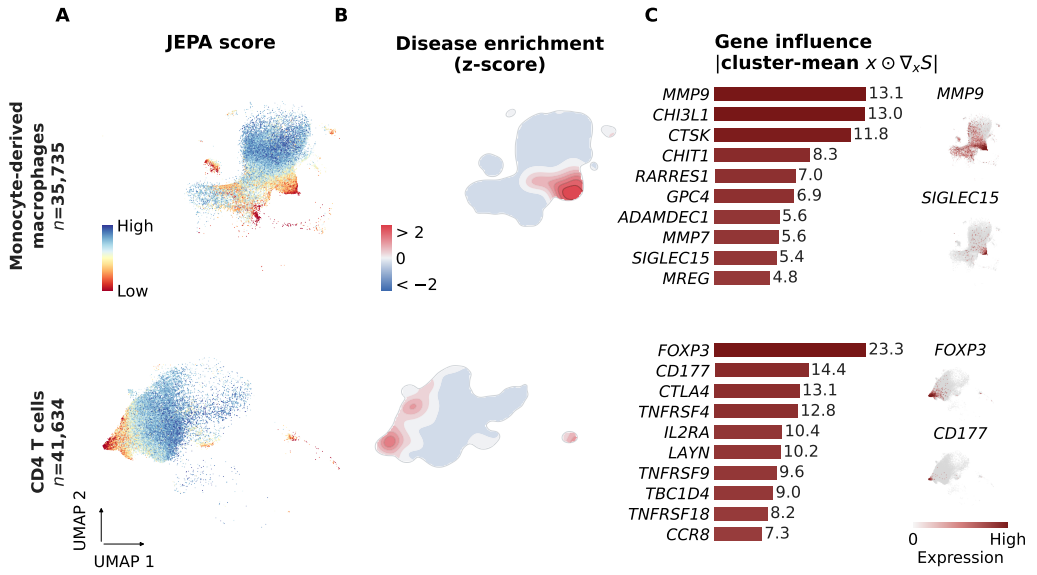


Figure 4: **Rare-program discovery and score-based gene attribution in IPF lung.** For each cohort, UMAP is computed on all cells; each row shows the indicated cell type at its coordinates in that full embedding. **(A) JEPA score:** Low-JEPA-score cells (red) form a coherent neighborhood within the cell type. **(B) disease enrichment:** kNN-smoothed disease enrichment z-score; the low-JEPA-score region is enriched for IPF cells relative to the cell-type baseline (monocyte-derived macrophages: 31.7% vs. 12.5%, 2.53×; CD4 T cells: 34.7% vs. 5.3%, 6.57×). **(C) gene influence:** Top-10 genes by absolute cluster-mean input×gradient over the low-JEPA-score cells; representative gene-expression UMAPs are shown to the right of each row. In monocyte-derived macrophages, the low-JEPA-score region recovers a profibrotic macrophage program, while the gradient prioritizes remodeling and checkpoint genes such as *MMP9*, *CTSK*, and *SIGLEC15*, rather than the most abundant marker, *SPP1*. In CD4 T cells, the same analysis recovers a CD177$^+$ activated-Treg program centered on *FOXP3*, *CD177*, *CTLA4*, *CCR8*, and *LAYN*. Cluster selection uses only the JEPA-score and embedding geometry; no gene list is used as input.

and SIGLEC15 2nd and 9th, against 25th and 43rd by DE (App. R). The underlying scores are stable across seeds and augmentation settings (mean pairwise per-cell Spearman $\rho = 0.965$ and 0.843 across four seeds in the macrophage and lymphoid cohorts; App. Q).

4 Analysis

Predictive selectivity. At the cell-type scale, the score behaves as the global density reading of Balestriero et al. [2025] predicts: on Heart v2, mean JEPA-score increases with cell-type abundance (Spearman $\rho = 0.86$ between $\log_{10}$ frequency and mean score). Section 2.4 predicts a finer mechanism *within* a cell type: under SIGReg, rare states whose cross-view predictability is carried by a compact gene program should concentrate their representation onto fewer dominant directions than common states do. We test this directly: for each population, we measure how well its embedding can be approximated by the one or two best-aligned genes (best-1 and best-2 cosine similarity, $\cos_1$, $\cos_2$).

The prediction holds. For CD177$^+$ Tregs (FOXP3 > 1 and CD177 > 1, $n = 182$), LeCellModel reaches $\cos_1/\cos_2 = 0.812/0.867$ against $0.367/0.477$ on the broader CD4 T-cell pool ($n = 500$); profibrotic monocyte-derived macrophages show the same gap ($0.717/0.821$ vs. $0.385/0.534$). The effect is selective to the SIGReg-induced geometry: under scVI, the rare-vs-common contrast almost vanishes (CD177$^+$ Treg $0.468/0.593$ vs. CD4 $0.359/0.473$; profibrotic monocyte-derived macrophages $0.326/0.465$ vs. general monocyte-derived macrophages $0.335/0.450$). The score's

geometric mechanism, not just embedding quality, is what makes within-type rare-state recovery work.

Regularizer ablation. Replacing SIGReg with VICReg [Bardes et al., 2022], which constrains only the first two moments, weakens both the score and the representation: the CD177$^+$ Treg share in the rarest 1% falls from 69% to 37%, IL-10 AUROC drops by 0.03 on average, and HLCA Core scIB Total falls from 0.689 to 0.659 (App. I).

5 Conclusion

The key design choice in LeCellModel is that SIGReg both prevents collapse and gives the Jacobian log-volume its meaning. The JEPA-score and its gradient therefore come directly from the trained model rather than being added afterward, and they are interpretable because of how the model was trained.

Toward atlas-scale reference embeddings. A natural next step is to use LeCellModel as an atlas reference: train once on a large multi-study cohort — potentially with a heavier encoder backbone such as a transformer — freeze the model, and map query cells into the resulting coordinate system. New samples would inherit an integrated embedding, a per-cell JEPA-score against a fixed reference geometry, and gene-level $\nabla_x S$ attribution, turning rare-state discovery into a query-time operation rather than a per-dataset retraining cycle. An open question is how the score's typicality interpretation transfers when query cells fall outside the training manifold.

Limitations. LeCellModel's cross-seed variance on Total scIB is larger in the low-data regime (Pancreas $\sigma = 0.008$) than at the atlas scale (Heart v2, HLCA Core $\sigma \leq 0.003$), so per-seed rankings on small datasets should be read with caution. The displacement-auxiliary ablation (App. H) uses slightly different adversary and SIGReg coefficients than the with-displacement runs, selected for stability without the auxiliary; the comparison is therefore not perfectly hyperparameter-matched. The JEPA-score is a geometric quantity of a trained model, not a guaranteed biological readout: it depends on training dynamics and on the augmentations the encoder was exposed to, so runs can disagree at the margin of the rare tail, though changed cells stay near the cutoff (App. Q). We cannot yet separate model uncertainty from genuine rarity or test the gradient against finite count perturbations, and our negative control (App. S) confirms only known phenotypes; new states need replication and experimental validation. Finally, $\nabla_x S$ identifies the genes whose local expression most strongly affects the score; these are interpretive markers of the rare-state geometry, not validated biological drivers.

Acknowledgments and Disclosure of Funding

The authors received no specific funding for this work and declare no competing interests.

References

Randall Balestriero and Yann LeCun. LeJEPA: Provable and scalable self-supervised learning without the heuristics. arXiv preprint arXiv:2511.08544, 2025.

Randall Balestriero, Nicolas Ballas, Mike Rabbat, and Yann LeCun. Gaussian embeddings: How JEPAs secretly learn your data density. arXiv preprint arXiv:2510.05949, 2025.

Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations (ICLR)*, 2022.

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

David DeTomaso and Nir Yosef. Hotspot identifies informative gene modules across modalities of single-cell genomics. *Cell Systems*, 12(5):446–456.e9, 2021. doi: 10.1016/j.cels.2021.04.005.

Ali ElSheikh, Rui-Xi Wang, Weimin Wu, Yibo Wen, Payam Dibaeinia, Jennifer Yuntong Zhang, Jerry Yao-Chieh Hu, Mei Knudson, Sudarshan Babu, Shao-Hua Sun, Aly A. Khan, and Han Liu. Cell-JEPA: Latent representation learning for single-cell transcriptomics. arXiv preprint arXiv:2602.02093, 2026.

Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *Proceedings of the 32nd International Conference on Machine Learning*, ICML, 2015. doi: 10.48550/arXiv.1409.7495.

Brian Hie, Bryan Bryson, and Bonnie Berger. Efficient integration of heterogeneous single-cell transcriptomes using scanorama. *Nature Biotechnology*, 37(6):685–691, 2019. doi: 10.1038/s41587-019-0113-3.

Myung-Chul Kim, Nicholas Borcherding, Kawther K. Ahmed, Andrew P. Voigt, Ajaykumar Vishwakarma, Ryan Kolb, Paige N. Kluz, Gaurav Pandey, Umasankar De, Theodore Drashansky, et al. Cd177 modulates the function and homeostasis of tumor-infiltrating regulatory T cells. *Nature Communications*, 12(1):5764, 2021. doi: 10.1038/s41467-021-26091-4.

Ilya Korsunsky, Nghia Millard, Jean Fan, Kamil Slowikowski, Fan Zhang, Kevin Wei, Yuriy Baglaenko, Michael Brenner, Po-Ru Loh, and Soumya Raychaudhuri. Fast, sensitive and accurate integration of single-cell data with harmony. *Nature Methods*, 16(12):1289–1296, 2019. doi: 10.1038/s41592-019-0619-0.

Elon Litman, Tyler Myers, Vinayak Agarwal, Ekansh Mittal, Orion Li, Ashwin Gopinath, and Timothy Kassis. Genejepa: A predictive world model of the transcriptome. *bioRxiv*, 2025. doi: 10.1101/2025.10.14.682378.

Monika Litviňuková, Carlos Talavera-López, Henrike Maatz, Daniel Reichart, Catherine L. Worth, Eric L. Lindberg, Masatoshi Kanda, Krzysztof Polanski, Matthias Heinig, Michael Lee, Emily R. Nadelmann, Kenny Roberts, Liz Tuck, Eirini S. Fasouli, Daniel M. DeLaughter, Barbara McDonough, Hiroko Wakimoto, Joshua M. Gorham, Sara Samari, Krishnaa T. Mahbubani, Kourosh Saeb-Parsy, Giannino Patone, Joseph J. Boyle, Hongbo Zhang, Hao Zhang, Anissa Viveiros, Gavin Y. Oudit, Omer Ali Bayraktar, J. G. Seidman, Christine E. Seidman, Michela Noseda, Norbert Hübner, and Sarah A. Teichmann. Cells of the adult human heart. *Nature*, 588(7838):466–472, 2020. doi: 10.1038/s41586-020-2797-4.

Romain Lopez, Jeffrey Regier, Michael B. Cole, Michael I. Jordan, and Nir Yosef. Deep generative modeling for single-cell transcriptomics. *Nature Methods*, 15(12):1053–1058, 2018. doi: 10.1038/s41592-018-0229-2.

Malte D. Luecken, Maren Büttner, Ketan Chaichoompu, Alessandro Danese, Marta Interlandi, Mirko F. Mueller, Daniel C. Strobl, Luke Zappia, Martin Dugas, Maria Colomé-Tatché, and Fabian J. Theis. Benchmarking atlas-level data integration in single-cell genomics. *Nature Methods*, 19(1):41–50, 2022. doi: 10.1038/s41592-021-01336-8.

Cameron Morse, Tracy Tabib, Jen Sembrat, Kristy L. Buschur, Hiam T. Bittar, Erika Valenzi, Yong Jiang, Dan J. Kass, Kevin Gibson, Wei Chen, and Robert Lafyatis. Proliferating spp1/mertk-expressing macrophages in idiopathic pulmonary fibrosis. *The European Respiratory Journal*, 54 (2):1802441, 2019. doi: 10.1183/13993003.02441-2018.

Parse Biosciences. 10 million human PBMCs in a single experiment. Dataset and analysis resource, 2026. URL `https://www.parsebiosciences.com/datasets/10-million-human-pbmcs-in-a-single-experiment/`. Accessed 2026-04-24.

Pia Rautenstrauch and Uwe Ohler. Shortcomings of silhouette in single-cell integration benchmarking. *Nature Biotechnology*, 2025. doi: 10.1038/s41587-025-02743-4.

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*, 656:183–191, 2026. doi: 10.1038/s41586-026-10689-z.

scverse contributors. scib-metrics: GPU-accelerated reimplementation of scIB single-cell integration benchmark metrics. GitHub repository, 2024. URL `https://github.com/YosefLab/scib-metrics`. Accessed 2026-04-24.

Ajita Shree, Musale Krushna Pavan, and Hamim Zafar. scdreamer for atlas-level integration of single-cell datasets using deep generative model paired with adversarial classifier. *Nature Communications*, 14(1):7781, 2023. doi: 10.1038/s41467-023-43590-8.

Lisa Sikkema, Ciro Ramírez-Suástegui, Daniel C. Strobl, Tessa E. Gillett, Luke Zappia, Elo Madissoon, Nikolay S. Markov, Laure-Emmanuelle Zaragosi, Yuge Ji, Meshal Ansari, Marie-Jeanne Arguel, Leonie Apperloo, Martin Banchero, Christophe Bécavin, Marijn Berg, Evgeny Chichelnitskiy, Mei-I Chung, Antoine Collin, Aurore C. A. Gay, Janine Gote-Schniering, Baharak Hooshiar Kashani, Kemal Inecik, Manu Jain, Theodore S. Kapellos, Tessa M. Kole, Sylvie Leroy, Christoph H. Mayr, Amanda J. Oliver, Michael von Papen, Lance Peter, Chase J. Taylor, Thomas Walzthoeni, Chuan Xu, Linh T. Bui, Carlo De Donno, Leander Dony, Alen Faiz, Minzhe Guo, Austin J. Gutierrez, Lukas Heumos, Ni Huang, Ignacio L. Ibarra, Nathan D. Jackson, Preetish Kadur Lakshminarasimha Murthy, Mohammad Lotfollahi, Tracy Tabib, Carlos Talavera-López, Kyle J. Travaglini, Anna Wilbrey-Clark, Kaylee B. Worlock, Masahiro Yoshida, Maarten van den Berge, Yohan Bossé, Tushar J. Desai, Oliver Eickelberg, Naftali Kaminski, Mark A. Krasnow, Robert Lafyatis, Marko Z. Nikolic, Joseph E. Powell, Jayaraj Rajagopal, Mauricio Rojas, Orit Rozenblatt-Rosen, Max A. Seibold, Dean Sheppard, Douglas P. Shepherd, Don D. Sin, Wim Timens, Alexander M. Tsankov, Jeffrey Whitsett, Yan Xu, Nicholas E. Banovich, Pascal Barbry, Thu Elizabeth Duong, Christine S. Falk, Kerstin B. Meyer, Jonathan A. Kropski, Dana Pe'er, Herbert B. Schiller, Purushothama Rao Tata, Joachim L. Schultze, Sara A. Teichmann, Alexander V. Misharin, Martijn C. Nawijn, Malte D. Luecken, and Fabian J. Theis. An integrated cell atlas of the lung in health and disease. *Nature Medicine*, 29:1563–1577, 2023. doi: 10.1038/s41591-023-02327-2.

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

Vincent A. Traag, Ludo Waltman, and Nees Jan van Eck. From Louvain to Leiden: guaranteeing well-connected communities. *Scientific Reports*, 9:5233, 2019. doi: 10.1038/s41598-019-41695-z.

Chenling Xu, Romain Lopez, Edouard Mehlman, Jeffrey Regier, Michael I. Jordan, and Nir Yosef. Probabilistic harmonization and annotation of single-cell transcriptomics data with deep generative models. *Molecular Systems Biology*, 17(1):e9620, 2021. doi: 10.15252/msb.20209620.

A Data availability

The HLCA Core is available from CZ CELLxGENE (`https://cellxgene.cziscience.com/collections/6f6d381a-7701-4781-935c-db10d30de293`); the exact dataset version used here is `b351804c-293e-4aeb-9c4c-043db67f4540.h5ad`. The HLCA extended atlas is available from the same portal; the monocyte–macrophage and lymphoid subsets used in Section 3.4 are obtained by manually subsetting the extended release to the corresponding lineages and to normal+IPF donors only. The Heart Cell Atlas v2 is available at `https://www.heartcellatlas.org/`. The multi-study human pancreas benchmark (`human_pancreas_norm_complexBatch.h5ad`) is distributed by Luecken *et al.* at `https://figshare.com/articles/dataset/Benchmarking_atlas-level_data_integration_in_single-cell_genomics_-_integration_task_datasets_Immune_and_pancreas_/12420968`. The Parse Biosciences PBMC cytokine-stimulation panel is available at `https://www.parsebiosciences.com/datasets/10-million-human-pbmcs-in-a-single-experiment/`; for the rare-state detection experiments (Section 3.3) we use the 2,000-HVG version distributed by CellFlow at `https://cellflow.readthedocs.io/en/stable/user/genapi/cellflow.datasets.pbmc_cytokines.html`.

B Dataset preprocessing details

Raw count matrices are used as input; library-size normalization and `log1p` are applied inside the view generator (Section 2.2).

HVG selection. We use $2{,}000$ HVGs for the three integration benchmarks and $3{,}000$ HVGs for the HLCA monocyte–macrophage and lymphoid subsets, selected with the batch-aware `seurat_v3` flavor on raw counts.

HLCA cohort subsetting. Both subsets were extracted from the HLCA extended release by filtering to the relevant lineage cell types and normal+IPF donors, excluding Kaminski *et al.* 2020 from both cohorts, and dropping datasets contributing fewer than $5{,}000$ (myeloid) or $1{,}000$ (lymphoid) cells. This yields 13 studies for the monocyte–macrophage cohort and 19 for the lymphoid cohort.

Atlas-scale view generation. Poisson resampling is omitted on atlas-scale datasets (HLCA Core, Heart v2) where dataset-level depth variability already provides sufficient view diversity.

C Architecture

The **encoder** f_θ is a two-layer MLP with a trailing LayerNorm:

$$\mathsf{Linear}(G \to h) \to \mathsf{LayerNorm}(h) \to \mathsf{GELU} \to \mathsf{Dropout} \to \mathsf{Linear}(h \to d) \to \mathsf{LayerNorm}(d),$$

with default $d = 256$, $h = 512$.

The **projector** g_ϕ is a two-layer MLP *without* a trailing norm:

$$\mathsf{Linear}(d \to 512) \to \mathsf{LayerNorm}(512) \to \mathsf{GELU} \to \mathsf{Dropout} \to \mathsf{Linear}(512 \to k),$$

with default $k = 128$.

The **adversary head** h_ψ is a two-layer MLP with a leading LayerNorm and LeakyReLU activation, producing logits over B batches:

$$\mathsf{LayerNorm}(d) \to \mathsf{Linear}(d \to d) \to \mathsf{LayerNorm}(d) \to \mathsf{LeakyReLU} \to \mathsf{Dropout}(0.1) \to \mathsf{Linear}(d \to B).$$

The conditional adversary prepends a learned cell-type embedding $e_c \in \mathbb{R}^{32}$ to the input, replacing the first LayerNorm and Linear with $\mathsf{LayerNorm}(d + 32) \to \mathsf{Linear}(d + 32 \to d)$. The conditional variant is not used in any experiment reported in this paper; all results use the unconditional adversary.

D View generation details

Global and local view augmentations are applied to raw counts $x \in \mathbb{R}^G$: (i) Poisson resampling, $\tilde{x}_j \sim \mathrm{Poisson}(x_j)$ per gene; (ii) library-size normalization to target sum 10^4 followed by $\log(1 + \cdot)$;

(iii) random gene masking at rate m_g (global) or m_ℓ (local). (m_g, m_ℓ) are $(0, 0.6)$ for all three integration benchmarks; the IPF and rare-state detection experiments use $(0.5, 0.7)$ (see Tables 5 and 6).

E SIGReg details

Let $\{p_n\}_{n=1}^N \subset \mathbb{R}^k$ denote the projections of a single view-slot across a batch of size N, with $p \in \mathbb{R}^k$ as defined in Section 2.1.

Random directions. We draw $M = 1024$ unit directions $\{\omega_m\}_{m=1}^M \subset S^{k-1}$ uniformly on the unit sphere (resampled each step) and reduce isotropy in $\mathbb{R}^k$ to M univariate normality tests on the scalar projections $u_n^{(m)} = \omega_m^\top p_n$.

Empirical characteristic function. For each direction ω_m, the empirical CF of the scalar projections $\{u_n^{(m)}\}_{n=1}^N$ is

$$\hat{\varphi}_{\omega_m}(t) \;=\; \frac{1}{N}\sum_{n=1}^{N} \exp\big(i\, t\, u_n^{(m)}\big). \tag{7}$$

Epps–Pulley statistic. We compare $\hat{\varphi}_{\omega_m}$ to the standard-normal CF $\varphi_{\mathcal{N}}(t) = \exp(-t^2/2)$ through a weighted squared-error integral,

$$T_{\mathrm{EP}}(\omega_m) \;=\; N \int_{-T}^{T} \big|\hat{\varphi}_{\omega_m}(t) - \varphi_{\mathcal{N}}(t)\big|^2\, w(t)\, dt, \tag{8}$$

with weighting $w(t) = \exp(-t^2/2)$ and truncation $T = 5$; the integral is approximated by a 17-point trapezoidal rule on a uniform grid over $[-T, T]$.

Aggregation across view-slots. At each training step we evaluate Eq. 8 once per view-slot ($n_g + n_\ell$ evaluations) and average across view-slots and directions to obtain $\mathcal{L}_{\mathrm{sig}}$ in Eq. 2. Per-view-slot evaluation preserves the i.i.d. assumption across $\{p_n\}_{n=1}^N$ that pooling projections across view-slots would violate.

F Adversarial batch correction details

Conditional variant. We condition the adversary on a biological grouping c (cell type, or a coarser grouping such as lineage — chosen per dataset):

$$\mathcal{L}_{\mathrm{adv}}^{\mathrm{cond}} = \mathrm{CE}\big(h_\psi([\mathrm{GRL}_\lambda(z)\,;\, e_c]), b\big), \tag{9}$$

where e_c is the learned group embedding defined in App. C. This removes batch effects *within* each group while preserving biological structure *between* groups, useful when high-confidence biological grouping is available. The conditional variant is not used in any experiment reported in this paper; all results use the unconditional adversary.

Stabilization. Adversarial training with a growing λ schedule is prone to instability. We apply two standard stabilizers: spectral normalization on h_ψ's linear layers (bounding its Lipschitz constant) and label smoothing in the cross-entropy (softening overconfident batch predictions). Both are applied in all experiments; label smoothing uses a value of 0.1.

G JEPA-score computation

We compute the stabilized score (Eq. 5) in a single vectorized pass. For a minibatch of N cells, reverse-mode differentiation mapped over cells (`vmap`, `jacrev`) gives the stacked Jacobians, and a batched singular-value decomposition (`svdvals`) gives their singular values:

$$\mathbf{J} = \big(J(x_1), \ldots, J(x_N)\big) \in \mathbb{R}^{N\times k\times G}, \qquad S(x_n) = \sum_{i=1}^{k} \log\max\big(\sigma_i(\mathbf{J}_n),\, \varepsilon\big), \quad n = 1, \ldots, N.$$

The floor is a numerical stabilization for near-degenerate singular directions and does not alter the intended geometric interpretation away from that regime.

H Displacement auxiliary details

Let $F = g \circ f$ denote the composed encoder–projector and $R \in \mathbb{R}^{G \times r}$ a fixed random matrix (seeded; held constant across training) with ℓ_2-normalized columns, where $r = 64$. The auxiliary is applied to a 25% subsample of each batch after warmup. For each cell x (log-normalized counts), the procedure is:

Algorithm 1 Displacement auxiliary (per cell $x \in \mathbb{R}^G$; per-gene std of log-normalized counts s; scale σ; subset size n_s). $\mathrm{sg}(\cdot)$ denotes stop-gradient.

Sample gene subset $S \subset \{1, \ldots, G\}$ uniformly at random, $|S| = n_s$; form indicator $v \in \{0, 1\}^G$
Sample $\alpha \sim \mathcal{N}(0, \sigma^2)$ ▷ scalar, per cell
$x' \leftarrow \max(x + \alpha \cdot (v \odot s),\, 0)$ ▷ scaled by per-gene batch std
$p \leftarrow F(x)$ ▷ with gradient
$p' \leftarrow F(x')$ ▷ no gradient
$\Delta p \leftarrow \mathrm{sg}(p') - \mathrm{sg}(p)$; $\bar{v} \leftarrow v/\|v\|$; $u \leftarrow R^\top \bar{v} \,/\, \|R^\top \bar{v}\|$
$\widehat{\Delta p} \leftarrow h_\theta(p, u)$ ▷ predictor input: p with gradient

h_θ takes the concatenation $[p;\, u] \in \mathbb{R}^{k+r}$ and outputs $\widehat{\Delta p} \in \mathbb{R}^k$ via

$$\mathsf{Linear}(k + r \to 256) \to \mathsf{LayerNorm}(256) \to \mathsf{GELU} \to \mathsf{Linear}(256 \to k).$$

We use $n_s = 25$ and $\sigma = 1.0$.

The loss combines a cosine-direction term and a scalar magnitude term, evaluated only on cells with $\|\Delta p\| > \tau$ ($\tau = 10^{-4}$):

$$\mathcal{L}_{\mathrm{disp}} = \mathbb{E}_{x,v}\Big[\, 1 - \cos\big(\widehat{\Delta p},\, \Delta p\big) \;+\; \gamma \cdot \big(\|\widehat{\Delta p}\| - \|\Delta p\|\big)^2 \,\Big], \qquad \gamma = 0.1. \tag{10}$$

The entire target Δp is detached, so the predictor matches it but the encoder does not shrink displacements to minimize it. Gradient flows into F only through p in $\widehat{\Delta p} = h_\theta(p, u)$; p' is computed without gradient, so the objective pressures the encoder to make small gene-space perturbations predictable in the latent space.

Ablation: scIB without the displacement auxiliary. Table 2 reports LeCellModel with and without the displacement auxiliary on the three integration benchmarks. Removing the auxiliary lowers Total on HLCA Core and Heart v2; on Pancreas, Total is essentially unchanged (a slight Bio gain offsets a Batch decline). In the main text, we report the configuration with the auxiliary enabled. The no-displacement runs use slightly different coefficients, selected for stability without the auxiliary: $\lambda_{\mathrm{sig}} = 0.4$ (vs. 0.5); adversary λ progress factor 1.6 (HLCA Core) / 1.6 (Heart v2) / 1.3 (Pancreas) vs. $1.8/1.8/1.5$; and Pancreas adversary weight 0.8 (vs. 1.0).

Table 2: LeCellModel with and without the displacement auxiliary across the three integration benchmarks. Baseline row from Table 1 (mean across 4 seeds); ablation row is single-seed (seed 42).

	HLCA Core			**Heart v2**			**Pancreas**		
Configuration	Bio	Batch	Total	Bio	Batch	Total	Bio	Batch	Total
Baseline (w/ disp.)	0.727	0.632	0.689	0.849	0.627	0.760	0.807	0.640	0.741
No displacement aux.	0.716	0.603	0.671	0.845	0.577	0.738	0.820	0.629	0.744

I Additional ablations

We ablate two design choices on all three integration benchmarks against the main-paper configuration: removing the adversarial batch-correction head, and changing the view-augmentation regime to

Table 3: LeCellModel ablations across the three integration benchmarks. Baseline row from Table 1 (mean across 4 seeds); ablation rows are single-seed (seed 42).

	HLCA Core			Heart v2			Pancreas		
Configuration	Bio	Batch	Total	Bio	Batch	Total	Bio	Batch	Total
Baseline	0.727	0.632	0.689	0.849	0.627	0.760	0.807	0.640	0.741
No adversary	0.685	0.472	0.600	0.727	0.446	0.614	0.656	0.371	0.542
$n_g = 2$, masks $0.5/0.7$	0.714	0.619	0.676	0.849	0.596	0.748	0.794	0.637	0.731

$n_g = 2$ globals with mask rates $m_g = 0.5, m_\ell = 0.7$ (instead of $n_g = 1$, $m_g = 0$, $m_\ell = 0.6$). The baseline row reproduces the multi-seed Total from Table 1; ablation rows are single-seed (seed 42).

The adversary is the dominant single-component contributor across all three datasets: removing it lowers Total by 0.089, 0.146, and 0.199 on HLCA Core, Heart v2, and Pancreas, respectively, through both Batch and Bio, with the larger drop in Batch. The alternative view regime is roughly Total-neutral on Heart v2 (-0.012) and gives small regressions on HLCA Core (-0.013) and Pancreas (-0.010); the lighter $n_g = 1$ regime is retained in the main paper.

Regularizer ablation. For VICReg we set the adversary progress factor to 1.0 (vs. 1.6 for SIGReg), as training was unstable at 1.6; all other settings follow the Figure 4 configuration (seed 1; VICReg $\lambda = 1.0$). The CD177$^+$ Treg program here is defined as CD4 T cells with FOXP3 > 2 and CD177 > 2 ($n = 162$); its median score percentile moves from 0.55% under SIGReg (seed 1; 69.1% in the rarest 1%) to 1.49% under VICReg (37.0%). On the cytokine panel, compared at matched seeds (1–2), VICReg lowers AUROC by 0.03 on average (up to 0.06) on both IL-10 pairs and changes it by at most 0.022 on the IL-4 pairs. On HLCA Core, VICReg reaches scIB Total 0.659 (mean of seeds 42, 1, 2 at $\lambda = 2.0$). The same Jacobian score computed on scVI's encoder stays near chance (AUROC 0.50–0.61 at $f = 5\%$ on all four pairs, seed 42, against 0.71–0.87 for LeCellModel), so the readout depends on the SIGReg-shaped representation.

J Training hyperparameters

LeCellModel. Across all datasets we use a two-layer MLP encoder and projector with $d = 256$, $k = 128$, AdamW (lr $= 5 \times 10^{-4}$, weight decay 10^{-2}), OneCycle LR with 10-epoch warmup, gradient clipping at norm 10, and bf16 mixed precision. Spectral normalization on the adversary and label smoothing (0.1) are enabled throughout. Per-dataset settings are in Table 4.

Random seeds. Integration benchmark results (Table 1) are averaged over seeds $\{42, 1, 2, 3\}$ — the framework default and the three smallest positive integers.

Table 4: LeCellModel training configuration per dataset. "Disp." is the displacement auxiliary. Epochs$= 150$ for all datasets.

	HLCA Core	Heart v2	Pancreas
Batch size	1024	1024	512
λ_{sig}	0.5	0.5	0.5
Adversary weight	0.5	0.5	1.0
Adv. λ progress factor	1.8	1.8	1.5
Adv. warmup epochs	0	0	0
n_g/n_ℓ	1 / 6	1 / 6	1 / 6
Mask rates m_g/m_ℓ	0 / 0.6	0 / 0.6	0 / 0.6
Poisson resampling	off	off	on
Projector dim k	128	128	128
Disp. (weight)	2.0	2.0	2.0

Rare-program discovery cohorts. The HLCA monocyte–macrophage and HLCA lymphoid (normal+IPF) subsets used in Section 3.4 are trained with the configuration in Table 5. They share the same encoder architecture (App. C) but use a larger projection dimension ($k = 256$) and a different

view-augmentation regime ($n_g = 2$ globals, mask rates $0.5/0.7$) than the integration benchmarks; both also use $3{,}000$ HVGs (App. B).

Table 5: LeCellModel training configuration for the IPF rare-program discovery cohorts (Section 3.4). Conventions match Table 4.

	HLCA mono–mac	HLCA lymphoid
Batch size	1024	1024
λ_{sig}	0.5	0.5
Adversary weight	0.7	0.5
Adv. λ progress factor	1.6	1.6
Adv. warmup epochs	0	0
n_g/n_ℓ	2 / 6	2 / 6
Mask rates m_g/m_ℓ	0.5 / 0.7	0.5 / 0.7
Poisson resampling	on	on
Projector dim k	256	256
Disp. (weight)	off	off

Rare-state detection experiments (Section 3.3). The four cytokine $\times$ cell-type runs use the configuration in Table 6. Each run trains LeCellModel from scratch on a contaminated mix at one of $f \in \{0.01, 0.03, 0.05, 0.10\}$ stim fractions, with no displacement auxiliary; the adversary uses donor as the batch label.

Table 6: LeCellModel training configuration for the cytokine rare-state detection benchmark (Section 3.3). Identical across all minority fractions; adv. λ progress factor differs between CD14 and CD16 pairs.

	Rare-state detection
Batch size	1024
λ_{sig}	0.35
Adversary weight	0.4
Adv. λ progress factor	1.0 (CD16 pairs); 1.4 (CD14 pairs)
Adv. warmup epochs	5
n_g/n_ℓ	2 / 6
Mask rates m_g/m_ℓ	0.5 / 0.7
Poisson resampling	on
Projector dim k	256
Disp. (weight)	off

Baselines. scVI is trained with `n_layers` $= 2$, `n_latent` $= 40$, `n_hidden` $= 256$ for up to 200 epochs with early stopping. Harmony and Scanorama are run on the standard scanpy log-normalization pipeline (10^4 target sum, log1p; Harmony additionally scales to unit variance at max value 10 and takes the top-50 PCs as input); both use library defaults. PCA uses 50 components on log-normalized, scaled expression.

Compute. All LeCellModel training runs use a single GPU per job, drawn from a shared institutional cluster with a mixed pool: NVIDIA A40, NVIDIA A100 (40 GB and 80 GB PCIe), and NVIDIA H100 NVL. Training and inference both use bf16 mixed precision. A full 150-epoch training run takes roughly 1 hour for the atlas-scale benchmarks (HLCA Core, Heart v2) and roughly 3 minutes for Pancreas; exact wall-clock varies with GPU type. The reported runs represent the final configurations; hyperparameter development required additional runs of comparable scale.

K GeneJEPA and foundation-model baselines

We evaluate GeneJEPA [Litman et al., 2025] zero-shot (published Tahoe-100M checkpoint) and, with our own training scripts, fine-tuned and trained from scratch on our benchmarks. All three settings trail LeCellModel and do not separate cell types (Leiden NMI 0.06–0.24); Figure 5 shows the zero-shot UMAP on HLCA Core. We also report scGPT [Cui et al., 2024] (zero-shot and fine-tuned)

and UCE [Rosen et al., 2026] (zero-shot, 4-layer model). UCE's 1280-dimensional embedding makes scIB's silhouette metrics infeasible on the full HLCA Core, so it is scored on a stratified 100k-cell subsample, with LeCellModel scored on the identical cells. Table 7 gives the per-metric breakdown for zero-shot GeneJEPA and Table 8 the aggregates for all settings.

Table 7: GeneJEPA zero-shot scIB on HLCA Core (584,944 cells), single run. LeCellModel row reproduced from Table 9 for reference.

	Bio-Conservation						**Batch-Correction**						
Method	Iso. Lab.	NMI	ARI	Sil.	cLISI	**Bio**	BRAS	iLISI	kBET	GC	PCR	**Batch**	**Total**
GeneJEPA (zero-shot)	0.450	0.090	0.031	0.440	0.935	0.389	0.793	0.231	0.205	0.089	0.685	0.401	0.394
LeCellModel	0.617	0.804	0.682	0.533	1.000	0.727	0.815	0.119	0.399	0.905	0.921	0.632	0.689

Table 8: GeneJEPA and foundation-model baselines, scIB aggregates, single run each. LeCellModel row from Table 1 (mean over four seeds). *UCE on HLCA Core is scored on a stratified 100k-cell subsample, against LeCellModel on the same cells.

	HLCA Core			**Pancreas**		
Method	Bio	Batch	Total	Bio	Batch	Total
GeneJEPA (zero-shot)	0.389	0.401	0.394	0.389	0.381	0.386
GeneJEPA (from scratch)	0.466	0.507	0.482	0.403	0.158	0.305
GeneJEPA (fine-tuned)	0.386	0.396	0.390	0.386	0.377	0.382
scGPT (zero-shot)	0.598	0.487	0.554	0.682	0.490	0.605
scGPT (fine-tuned)	0.706	0.414	0.589	0.760	0.496	0.655
UCE (zero-shot)	0.698*	0.349*	0.558*	0.600	0.373	0.510
LeCellModel	0.727	0.632	0.689	0.807	0.640	0.741
LeCellModel (100k subsample)	0.732*	0.664*	0.705*	–	–	–

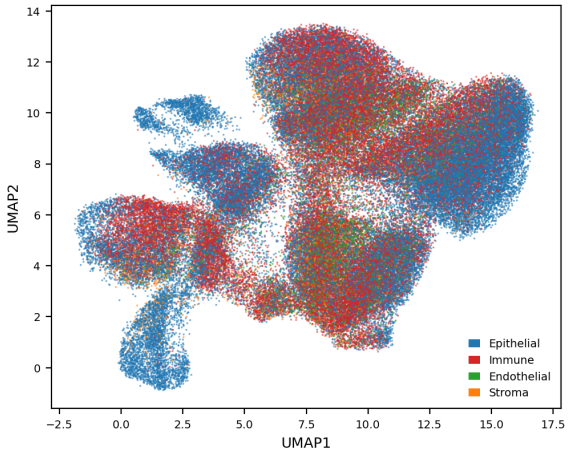


Figure 5: **GeneJEPA zero-shot embedding of HLCA Core, colored by cell type.** The frozen encoder does not produce separated clusters on primary tissue, consistent with the Tahoe-100M → HLCA domain gap.

L Full per-metric scIB tables

Tables 9–11 report the ten individual scIB metrics for each method on each benchmark. Bio-Conservation is the mean of Isolated Labels, Leiden NMI, Leiden ARI, Silhouette Label, and cLISI; Batch-Correction is the mean of BRAS (Batch-label Adjusted Silhouette, a cell-type-stratified silhouette score that measures within-label batch mixing; `scib_metrics.bras`), iLISI, kBET, Graph Connectivity, and PCR comparison; Total is the scIB-convention weighted average (0.6 Bio $+ 0.4$ Batch). Column aggregates match Table 1 up to rounding.

Table 9: HLCA Core (584,944 cells): per-metric scIB results, mean ± std across 4 seeds. Iso. Lab. = Isolated Labels; NMI/ARI on Leiden; Sil. = Silhouette Label; GC = Graph Connectivity; PCR = PCR comparison.

	Bio-Conservation						Batch-Correction						
Method	Iso. Lab.	NMI	ARI	Sil.	cLISI	**Bio**	BRAS	iLISI	kBET	GC	PCR	**Batch**	**Total**
PCA	0.556±.001	0.763±.015	0.545±.023	0.549±.003	1.000	0.682±.008	0.634±.002	0.001±.001	0.144±.006	0.881±.008	0.000	0.332±.003	0.542±.006
Scanorama	0.562±.001	0.795±.002	0.635	0.545	1.000	0.707	0.778±.002	0.019	0.199±.001	0.791±.003	0.153±.088	0.388±.017	0.580±.006
Harmony	0.538	0.755±.005	0.598±.028	0.537	0.999	0.685±.007	0.812±.001	0.063	0.318±.001	0.830±.001	0.558±.065	0.516±.013	0.618±.009
scVI	0.605±.004	0.810±.001	0.707±.004	0.521±.001	1.000	**0.729**±.001	0.822±.001	0.014	0.184±.001	0.924±.002	0.813±.029	0.551±.006	0.658±.002
LeCellModel	0.617±.017	0.804±.003	0.682±.008	0.533	1.000	0.727±.005	0.815±.003	0.119±.001	0.399±.005	0.905±.006	0.921±.012	**0.632**±.003	**0.689**±.003

Table 10: Heart Cell Atlas v2 (704,296 cells): per-metric scIB results, mean ± std across 4 seeds.

	Bio-Conservation						Batch-Correction						
Method	Iso. Lab.	NMI	ARI	Sil.	cLISI	**Bio**	BRAS	iLISI	kBET	GC	PCR	**Batch**	**Total**
PCA	0.726±.012	0.820	0.692±.001	0.578±.020	1.000	0.763±.006	0.552±.016	0.021±.001	0.090±.003	0.907±.001	0.000	0.314±.002	0.583±.003
Scanorama	0.642	0.841	0.754	0.572	1.000	0.762	0.672	0.044	0.117	0.796	0.083±.048	0.342±.009	0.594±.004
Harmony	0.691	0.931±.010	0.948±.015	0.643	1.000	0.843±.005	0.851	0.097	0.271±.003	0.854	0.696±.035	0.554±.007	0.727±.004
scVI	0.672±.004	0.938±.001	0.956	0.524±.001	1.000	0.818±.001	0.835±.001	0.066±.001	0.114±.001	0.900±.001	0.801±.025	0.543±.005	0.708±.002
LeCellModel	0.835±.004	0.935±.005	0.952±.007	0.524±.001	1.000	**0.849**±.002	0.777±.003	0.155±.001	0.330±.006	0.919±.007	0.953±.005	**0.627**±.001	**0.760**±.002

Table 11: Pancreas (16,382 cells): per-metric scIB results, mean ± std across 4 seeds.

	Bio-Conservation						Batch-Correction						
Method	Iso. Lab.	NMI	ARI	Sil.	cLISI	**Bio**	BRAS	iLISI	kBET	GC	PCR	**Batch**	**Total**
PCA	0.656	0.653	0.355	0.528	1.000	0.638	0.461	0.004	0.114	0.689	0.000	0.254	0.485
Scanorama	0.524±.002	0.860±.010	0.904±.010	0.559±.001	1.000	0.770±.004	0.780±.004	0.126	0.367±.005	0.799±.022	0.711±.003	0.556±.006	0.684±.004
Harmony	0.666	0.768±.002	0.617±.003	0.582	1.000	0.727±.001	0.714	0.203	0.457	0.762	0.800	0.587	0.671±.001
scVI	0.719±.010	0.918±.003	0.952±.002	0.568±.002	1.000	**0.831**±.003	0.782±.001	0.131±.001	0.347±.030	0.879±.009	0.829±.005	0.594±.007	0.736±.003
LeCellModel	0.642±.051	0.908±.004	0.940±.003	0.547±.001	1.000	0.807±.010	0.714±.007	0.232±.006	0.416±.017	0.916±.008	0.923±.005	**0.640**±.007	**0.741**±.008

M Full rare-state detection results

Full AUROC, AUPRC, and Spearman ρ (within stimulated cells, against the pathway signature z-scored on resting cells) for all four (cytokine, cell type) pairs across all four minority fractions (seed 42, one run per pair).

Table 12: Full AUROC by (pair, f, method). Bold marks the best entry per row.

Pair	f	**LeCellModel**	PCA-kNN	PCA-Mahal	PCA-IForest	scVI-IForest	scVI-recon	scVI-Mahal	scVI-kNN
IL-10 × CD16	0.01	0.835	**0.863**	0.820	0.817	0.847	0.604	0.819	0.839
	0.03	**0.844**	0.822	0.773	0.738	0.797	0.561	0.748	0.769
	0.05	**0.854**	0.801	0.745	0.742	0.704	0.565	0.719	0.739
	0.10	**0.832**	0.751	0.698	0.704	0.663	0.557	0.656	0.683
IL-10 × CD14	0.01	**0.749**	0.720	0.680	0.661	0.737	0.562	0.729	0.711
	0.03	**0.756**	0.702	0.674	0.679	0.677	0.556	0.694	0.666
	0.05	**0.761**	0.685	0.662	0.635	0.673	0.552	0.654	0.625
	0.10	**0.744**	0.642	0.627	0.600	0.604	0.551	0.593	0.563
IL-4 × CD14	0.01	0.906	0.911	0.905	**0.912**	0.856	0.548	0.910	0.866
	0.03	**0.885**	0.874	0.852	0.843	0.835	0.528	0.834	0.762
	0.05	**0.871**	0.848	0.821	0.811	0.752	0.528	0.785	0.730
	0.10	**0.845**	0.798	0.769	0.751	0.697	0.521	0.701	0.659
IL-4 × CD16 (weak)	0.01	0.667	0.724	0.721	0.720	0.714	0.467	0.765	**0.775**
	0.03	0.686	**0.729**	0.702	0.694	0.675	0.467	0.713	0.682
	0.05	**0.705**	0.701	0.661	0.658	0.646	0.464	0.680	0.667
	0.10	**0.690**	0.667	0.618	0.606	0.619	0.465	0.608	0.626

To separate the readout from the representation, we compute kNN density, Mahalanobis distance and isolation forest in LeCellModel's own projection, on the same cells and runs as JEPA-SCORE (Table 15). On the three established-response pairs, the two distance scores fall below chance at every fraction (e.g. AUROC 0.27 at $f = 1\%$ on IL-10 × CD16, against 0.84 for JEPA-SCORE): under SIGReg, rare-and-predictive states occupy a compact, low-volume region that distance-based scores read as dense. Isolation forest is competitive only on IL-4 × CD14 at low contamination (0.89 at $f = 1\%$) and falls to 0.75 at $f = 10\%$, while JEPA-SCORE stays between 0.84 and 0.91 on that pair. On the weaker IL-4 × CD16 response, all four scores are closer to chance.

Table 13: Full AUPRC by (pair, f, method). Bold marks the best entry per row.

Pair	f	**LeCellModel**	PCA-kNN	PCA-Mahal	PCA-IForest	scVI-IForest	scVI-recon	scVI-Mahal	scVI-kNN
IL-10 × CD16	0.01	0.085	0.054	0.045	0.046	**0.129**	0.015	0.081	0.093
	0.03	**0.217**	0.103	0.090	0.076	0.134	0.038	0.096	0.102
	0.05	**0.326**	0.141	0.121	0.127	0.112	0.062	0.127	0.134
	0.10	**0.416**	0.208	0.182	0.193	0.180	0.119	0.174	0.185
IL-10 × CD14	0.01	0.024	0.020	0.017	0.017	**0.028**	0.012	0.026	0.024
	0.03	**0.071**	0.052	0.049	0.052	0.055	0.036	0.059	0.055
	0.05	**0.120**	0.080	0.077	0.069	0.088	0.059	0.080	0.076
	0.10	**0.203**	0.137	0.134	0.123	0.132	0.118	0.127	0.121
IL-4 × CD14	0.01	**0.164**	0.123	0.142	0.128	0.073	0.014	0.117	0.063
	0.03	**0.233**	0.177	0.174	0.165	0.125	0.036	0.122	0.081
	0.05	**0.292**	0.221	0.217	0.218	0.127	0.059	0.146	0.109
	0.10	**0.407**	0.298	0.294	0.273	0.187	0.114	0.189	0.159
IL-4 × CD16 (weak)	0.01	0.017	0.025	0.029	0.029	0.027	0.009	**0.045**	0.038
	0.03	0.058	0.067	**0.073**	0.071	0.059	0.026	**0.073**	0.058
	0.05	**0.106**	0.092	0.089	0.084	0.087	0.043	0.095	0.087
	0.10	**0.190**	0.151	0.138	0.134	0.144	0.087	0.138	0.144

Table 14: Spearman ρ between per-cell score and pathway signature, computed within stimulated cells only. Bold marks the best entry per row.

Pair	f	**LeCellModel**	PCA-kNN	PCA-Mahal	PCA-IForest	scVI-IForest	scVI-recon	scVI-Mahal	scVI-kNN
IL-10 × CD16	0.01	0.532	0.514	0.565	**0.611**	0.414	0.252	0.384	0.390
	0.03	0.463	0.378	**0.470**	0.449	0.279	0.235	0.275	0.238
	0.05	**0.473**	0.280	0.361	0.339	0.151	0.239	0.183	0.126
	0.10	**0.428**	0.182	0.255	0.232	0.083	0.249	0.071	0.032
IL-10 × CD14	0.01	**0.363**	0.172	0.250	0.265	0.142	0.252	0.133	0.072
	0.03	**0.315**	0.122	0.240	0.238	0.104	0.250	0.029	-0.048
	0.05	**0.312**	0.091	0.210	0.222	0.011	0.255	-0.030	-0.101
	0.10	0.251	0.031	0.138	0.102	-0.030	**0.254**	-0.097	-0.167
IL-4 × CD14	0.01	**0.688**	0.344	0.659	0.605	0.359	0.179	0.410	0.051
	0.03	**0.535**	0.281	0.519	0.484	0.270	0.192	0.218	-0.099
	0.05	**0.473**	0.232	0.430	0.435	0.156	0.180	0.151	-0.104
	0.10	**0.414**	0.186	0.330	0.315	0.056	0.163	0.058	-0.139
IL-4 × CD16 (weak)	0.01	0.468	0.522	0.524	**0.525**	0.367	0.145	0.365	0.343
	0.03	0.442	0.515	**0.523**	0.481	0.238	0.206	0.218	0.078
	0.05	0.464	0.498	**0.516**	0.492	0.144	0.193	0.178	0.052
	0.10	0.428	**0.463**	0.436	0.429	0.210	0.185	0.164	0.102

Table 15: Rare-state detection with simple scores computed in LeCellModel's projection: AUROC by pair and contamination fraction f, mean over seeds 1–3 (s.d. ≤ 0.07). Bold marks the best entry per row.

Pair	f	JEPA-SCORE	kNN	Mahal.	IForest
IL-10 × CD14	0.01	**0.746**	0.312	0.314	0.525
	0.03	**0.758**	0.277	0.288	0.549
	0.05	**0.761**	0.272	0.289	0.601
	0.10	**0.744**	0.280	0.293	0.569
IL-10 × CD16	0.01	**0.837**	0.273	0.273	0.576
	0.03	**0.865**	0.184	0.223	0.617
	0.05	**0.864**	0.178	0.219	0.639
	0.10	**0.840**	0.203	0.250	0.625
IL-4 × CD14	0.01	**0.907**	0.140	0.290	0.893
	0.03	**0.882**	0.153	0.249	0.863
	0.05	**0.866**	0.164	0.264	0.849
	0.10	**0.844**	0.185	0.288	0.753
IL-4 × CD16 (weak)	0.01	**0.630**	0.604	0.577	0.495
	0.03	**0.675**	0.510	0.509	0.497
	0.05	**0.701**	0.412	0.447	0.499
	0.10	**0.688**	0.383	0.421	0.506

N Cytokine-stimulation panel: pair details

All pairs draw from a fixed pool of 16,710 resting CD16 monocytes and 110,999 resting CD14 monocytes. We consider four cytokine–cell-type settings: *IL-4 × CD14 Mono*, a canonical M2-like IL-4/STAT6 response; *IL-10 × CD14 Mono*, a canonical anti-inflammatory IL-10/STAT3 response; *IL-10 × CD16 Mono*, an IL-10/STAT3 response in a distinct $CD16^+$ background; and *IL-4 × CD16 Mono*, a weaker IL-4-response setting reported in Appendix M. Density-based baselines use scVI with `n_latent=30`, 100 epochs.

Pathway signatures. The Spearman ρ in Table 14 uses the mean z-score (z-scored on resting cells) of an up-set minus a down-set: **IL-10**, up = {CD163, CD163L1, IRAK3, MARCO, MRC1, F13A1} (STAT3 targets), down = {IL1B, IL6, CXCL10, CCL3, CCL4, CXCL8} (IL-10-suppressed pro-inflammatory mediators); **IL-4**, up = {CCL17, CCL22, MRC1, ALOX15, CCL18, CHI3L1, F13A1} (STAT6 targets), no down-set.

O Rare-program discovery: statistics and cluster selection

Bivariate Moran's I on the embedding graph. For each gene g, bivariate Moran's I_g measures whether gene expression and JEPA score are spatially anti-correlated on the embedding graph: $I_g < 0$ means g is elevated in cells whose neighbors have low JEPA score, marking it as a candidate gene of the rare subpopulation. W is the Gaussian-kernel-weighted kNN graph ($k = 15$, row-stochastic) on the LeCellModel projection; significance uses the analytical z-score from Hotspot [DeTomaso and Yosef, 2021], with BH-FDR < 0.05.

Before computing I_g we apply a logistic transform to the raw score, centering at the 5th-percentile score $c = \hat{F}_{5\%}(S)$, scaling by the score standard deviation σ_S, and squashing with a sigmoid of steepness $\beta = 2$:

$$\tilde{S}(x) \;=\; \sigma\!\left(\beta\,\frac{S(x) - c}{\sigma_S}\right).$$

This maps the rare anomalous tail ($S \ll c$) toward 0 and the bulk ($S \gg c$) toward 1. JEPA-coupled genes ($I_g < 0$, BH-FDR < 0.05) are clustered into modules by their per-cell coupling profile $\mathbf{q}_g \in \mathbb{R}^N$, where $(\mathbf{q}_g)_n = \tilde{x}_{g,n} \cdot (W\tilde{S})_n$ is the product of gene g's standardized expression and the neighborhood-smoothed score at cell n. Genes in the same module are elevated in the same low-score cells, not merely co-expressed globally. Module 1 (ranked by $|I| \cdot \sqrt{n_{\text{genes}}}$) is the recovered program reported in Section 3.4 (Table 16).

Cluster selection for gradient attribution. Gradient attribution requires a concrete cell cluster. We identify connected neighborhoods of low-JEPA-score cells via a pit-finding routine; the exact cluster boundary does not materially affect the attributed genes. The routine operates on a cosine kNN graph ($k = 10$) restricted to the bottom 10% of cells by JEPA-score (the *subgraph*), with a data-adaptive distance threshold δ — the 10^{th} percentile of kNN edge distances within the subgraph — controlling which neighbors are traversable.

1. Smooth the JEPA-score: replace each cell's score with the mean of itself and neighbors at distance $\leq \delta$, one pass.
2. From seeds at the bottom 0.5% of JEPA-score (all cells), follow the lowest-scoring neighbor at distance $\leq \delta$ until no lower neighbor exists; the landing cell is a local minimum ("pit bottom").
3. Group seeds by pit bottom; discard pits with fewer than $\max(3,\ 5\%$ of seeds$)$ converging seeds.
4. Iteratively merge the least-prominent adjacent pit pair: the prominence of pit B is the lowest cross-boundary score between A and B minus the score of B's bottom (i.e. how much the score rises to leave B); merge until all remaining pits are separated by more than $0.3\,\sigma_S$, where σ_S is the score standard deviation within the subgraph.
5. Expand each surviving pit into a core cluster by BFS within distance δ.

Table 16: Top-15 members of the recovered program (Module 1) per cohort, ranked by $|I_g|$ within the module. I_g is the raw bivariate Moran's I (range $[-1, 1]$); negative values indicate the gene is elevated in low-JEPA-score (rare) cells. Module 1 is the dominant co-coupling cluster among JEPA-coupled genes (BH-FDR < 0.05, $I_g < 0$). Mono–mac Module 1: $n_{\text{genes}} = 47$, mean $I = -0.732$. CD4 Module 1: $n_{\text{genes}} = 48$, mean $I = -0.784$.

Monocyte-derived macrophages (Module 1)			**CD4 T cells (Module 1)**		
Rank	Gene	I_g	Rank	Gene	I_g
1	CHI3L1	-0.459	1	FOXP3	-0.484
2	ADAMDEC1	-0.429	2	TNFRSF18	-0.417
3	MMP9	-0.415	3	IL2RA	-0.416
4	CHIT1	-0.355	4	TNFRSF9	-0.408
5	CTSK	-0.353	5	TNFRSF4	-0.398
6	MREG	-0.327	6	TIGIT	-0.397
7	RARRES1	-0.325	7	CD177	-0.397
8	A2M	-0.325	8	LAYN	-0.385
9	SIGLEC15	-0.319	9	CTLA4	-0.370
10	GPC4	-0.317	10	LINC02099	-0.311
11	LILRB4	-0.315	11	TBC1D4	-0.303
12	MMP7	-0.315	12	IL1R2	-0.297
13	SPP1	-0.297	13	ENTPD1	-0.297
14	PALLD	-0.295	14	CCR8	-0.296
15	FCMR	-0.294	15	IKZF2	-0.281

For each core cluster C we compute the input×gradient attribution $x \odot \nabla_x S(x)$ at every cell via automatic differentiation, then rank genes by the absolute cluster mean, $|\mathbb{E}_{x \in C}[x \odot \nabla_x S]|$. Gene identity is not used in cluster selection; the procedure sees only the JEPA-score and the embedding geometry. All case-study results in Section 3.4 use these defaults.

P Predictive selectivity: computation details

For each cell x with full-cell projection $p = F(x)$ (log1p-CP10k input), $\cos_1 = \max_g \cos(F(x^{(g)}), p)$ where $x^{(g)}$ retains only gene g; $\cos_2$ further adds the best second gene on top of g^*, both exhaustive over all G genes, reported as population medians. The CD177$^+$ Treg cluster uses all $n = 182$ cells (FOXP3> 1 & CD177> 1); the CD4 T-cell pool and profibrotic monocyte-derived macrophage groups are each capped at $n = 500$ (uniform subsample, seed 42). The scVI baseline applies the same procedure to the posterior mean μ_z with raw-count inputs; scVI is trained with `n_latent`$= 40$, `n_layers`$= 2$, `n_hidden`$= 256$, donor batch key, up to 200 epochs with early stopping.

Q Stability of JEPA-score rankings

We retrain each IPF cohort with four seeds (42, 1, 2, 3) and, at seed 42, under two further augmentation settings (global/local mask rates $0.2/0.6$ and $0.1/0.4$, against $0.5/0.7$ in Table 5) and with Poisson resampling disabled. Runs score the same cells in the same order, so rankings are compared cell by cell (Table 17). Across seeds, per-cell scores correlate at mean pairwise Spearman $\rho = 0.965$ (monocyte–macrophage, 257,893 cells) and 0.843 (lymphoid, 159,652 cells). About 90% of each run's rarest 1% is shared with any other run. The cells that differ sit just past the cutoff: their median score percentile in the other run is 1.13–1.14%, and 99.9% (monocyte) and 99.6% (lymphoid) fall within its rarest 2%. Within the shared tail, the ordering is preserved (Spearman 0.82–0.89), and a consensus of three runs recovers 91–92% of the held-out run's rare set. The rare tail is nearly as stable across augmentation settings: of the cells whose rare-1% membership changes between mask regimes, 90.6% (monocyte) and 94.2% (lymphoid) stay within the rarest 2%, and disabling Poisson resampling gives the same picture.

Table 17: Stability of per-cell JEPA-scores in the IPF cohorts. ρ: mean pairwise Spearman correlation of per-cell scores; overlap: mean fraction of one run's rarest 1% that is also in the other run's; $\leq$2%: fraction of cells whose rare-1% membership changes that fall within the other run's rarest 2%.

	Monocyte–macrophage			**Lymphoid**		
Comparison	ρ	Overlap	$\leq$2%	ρ	Overlap	$\leq$2%
Seeds 42, 1, 2, 3	0.965	90.2%	99.9%	0.843	89.5%	99.6%
Mask rates $0.1/0.4$, $0.2/0.6$, $0.5/0.7$	0.936	80.6%	90.6%	0.877	88.2%	94.2%
Poisson resampling off vs. on	0.954	87.1%	95.0%	0.894	93.1%	99.5%

R Gradient attribution against pathway targets and differential expression

We rank genes by the absolute cluster-mean input×gradient, $|\mathbb{E}_{x\in C}[x \odot \nabla_x S]|$, and compare with the raw gradient, $|\mathbb{E}_{x\in C}[\nabla_x S]|$, and with Wilcoxon differential expression (Table 18). Without any pathway list, the gradient ranks canonical STAT3 and STAT6 targets at the top on the CD14 pairs (F13A1 and ALOX15 first), and lower on the CD16 pairs. Where the two rankings disagree, the gradient favors genes expressed in a subset of the cluster (CD177, SIGLEC15, MARCO, CCL18), while DE favors broadly expressed markers such as SPP1. The gradient thus captures local drivers of the score rather than abundance differences between groups, and needs no predefined contrast group.

Table 18: Gene ranks by input×gradient, raw gradient and differential expression (Wilcoxon rank-sum, ranked by test statistic). IPF: cluster against the rest of its cell type, ranks over 3,000 genes. Cytokine panel ($f = 5\%$): stimulated against control cells, ranks over 2,000 genes. Expressed: share of the cluster's (IPF) or stimulated (cytokine) cells with non-zero expression.

Gene	Setting	Input×grad.	Raw grad.	DE	Expressed
CD177	IPF, CD4 T cells	2	1	25	37%
SIGLEC15	IPF, mono.-derived macrophages	9	4	43	50%
SPP1	IPF, mono.-derived macrophages	414	2,875	8	92%
F13A1	IL-10 × CD14 (seed 1)	1	12	1	59%
ALOX15	IL-4 × CD14 (seed 1)	1	3	2	68%
CCL18	IL-4 × CD14 (seed 1)	8	25	93	14%
MARCO	IL-10 × CD16 (seeds 1–3)	3–7	1	26–35	28–31%
CD163	IL-10 × CD16 (seeds 1–3)	4–19	14–65	5–7	69–72%
F13A1	IL-10 × CD16 (seeds 1–3)	74–91	27–43	84–91	16–18%
ALOX15	IL-4 × CD16 (seeds 1–3)	8–19	1–2	43	14–15%

S Negative control on healthy heart tissue

As a negative control we use healthy tissue, where no disease program is present: the subsampled Heart Cell Atlas distributed with scvi-tools [Litviňuková et al., 2020]. We train with this dataset's own configuration: batch size 512, $\lambda_{\text{sig}} = 0.4$, adversary weight 0.8 with progress factor 2.0, one global and four local views with mask rates $0.2/0.5$, $d = k = 256$, no displacement auxiliary, 150 epochs. In this model, the lowest-scoring regions include an activated venous-endothelial state ($n = 46$; top input×gradient genes ACKR1, SELE and VCAM1). We then remove every cell at or below the 20th percentile of JEPA-score within its cell type (18,641 $\rightarrow$ 14,908 cells) and retrain on the remaining cells. The retrained model finds one low-score region instead of four; the remaining ventricular-cardiomyocyte region lies 0.67 standard deviations below the mean, against 1.71 before pruning, and the score distribution tightens (mean $-818 \rightarrow -747$, s.d. $88 \rightarrow 79$): homogeneous populations receive more uniform scores.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The five contributions listed in the Introduction (a relative density estimate of cell states via JEPA-score, read within a cell type as typicality; differentiable gene-level attribution of the score; the LeCellModel architecture; atlas-scale cell-identity preservation; and density-driven discovery in disease tissue) correspond to Section 2.4, Section 2.5 and App. R, Section 2.3, Section 3.2 (Table 1; App. K), and Sections 3.3–3.4 (Figures 3, 4); the abstract summarizes these without overstating scope.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The Limitations paragraph at the end of Section 5 covers cross-seed variance on small datasets, the hyperparameter mismatch in the displacement-auxiliary ablation, run-to-run variation at the margin of the rare tail (App. Q), the inability to separate model uncertainty from genuine rarity, the untested response of the gradient to finite count perturbations, the scope of the negative control (App. S), and the interpretive rather than causal status of $\nabla_x S$.

   Guidelines:

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

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

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

   Answer: [N/A]

   Justification: The paper does not introduce new theorems. The Jacobian-density connection from Balestriero et al. [2025] is referenced and informally discussed in Section 2.4 but not re-derived; the regime-shift argument for within-cell-type usage is empirical (Section 4).

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Architecture is given in App. C; view generation in Section 2.2 and App. D; training objective in Section 2.3; per-dataset hyperparameters for the integration benchmarks in Table 4 and for the IPF rare-program cohorts in Table 5 (App. J); displacement auxiliary algorithm in App. H; baseline configurations in App. J. All datasets are public and cited.

   Guidelines:

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

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: All datasets are publicly released and cited, with exact versions and URLs in App. A. Code, training scripts and configurations to reproduce the LeCellModel results and the PCA, Harmony, Scanorama and scVI baselines are available at `https://github.com/gilk717/lecellmodel_paper`. Not included: the scGPT, UCE and GeneJEPA embeddings, which were produced with their official pipelines (settings in App. K), and the code for the heart negative control (App. S).

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Optimizer (AdamW), learning rate, weight decay, schedule (OneCycle), warmup, gradient clipping, and mixed precision are stated in App. J; per-dataset configurations are in Tables 4 and 5; baseline configurations are in App. J. View-generation parameters are in Section 2.2 and App. D.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: Tables 1 and 9–11 report mean $\pm$ sample standard deviation across 4 random seeds (42, 1, 2, 3), per metric. Seed values are stated in the Table 1 caption. Standard deviation captures variability across model initialization and training-time stochasticity (augmentation, ordering); other sources of variability (e.g. dataset re-split) are not perturbed because the integration benchmarks are evaluated on the entire labeled cohort. The cytokine and rare-program experiments are case studies of the score's behavior rather than strict benchmarks; robustness is shown by sweeping across contamination fractions and cytokine–cell-type pairs.

   Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: GPU pool, mixed precision, and approximate per-run wall-clock for LeCellModel are reported in the "Compute" paragraph of App. J. The foundation-model baselines (App. K) used the same GPU pool; training GeneJEPA from scratch on HLCA Core took about 17 hours on a single GPU, and computing its per-cell Jacobian took about 16 s per cell, against about 0.007 s for LeCellModel (Section 2.4).

   Guidelines:

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

9. **Code of ethics**

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

Answer: [Yes]

Justification: The work uses publicly released, deidentified single-cell transcriptomic datasets and conforms with the NeurIPS Code of Ethics; no new human-subject data were collected and no high-risk dual-use models are released.

Guidelines:

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

10. **Broader impacts**

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

Answer: [N/A]

Justification: This is foundational ML methodology applied to public single-cell transcriptomic data; the intended use is improved batch integration and rare-cell-state discovery for biomedical research, with positive impact on translational disease analysis and no direct path to negative or dual-use applications.

Guidelines:

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

11. **Safeguards**

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

Answer: [N/A]

Justification: The paper does not release pre-trained generative models, scraped datasets, or content of high misuse risk. LeCellModel is trained per-dataset on public deidentified single-cell data.

Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: All datasets are cited via their original publications: HLCA Core and the IPF extension [Sikkema et al., 2023] (CC BY 4.0, via CELLxGENE), Heart Cell Atlas v2 [Litviňuková et al., 2020] (CC BY 4.0, via HCA Data Portal and CELLxGENE), the multi-study Pancreas benchmark [Luecken et al., 2022] (publicly available via NCBI GEO under GEO's unrestricted-use data policy; no formal CC license assigned), and the Parse Biosciences cytokine panel [Parse Biosciences, 2026] (CC BY-NC 4.0, as stated on the Parse Biosciences dataset page; used here for non-commercial academic research); exact dataset versions and URLs are in App. A. The negative control uses the subsampled Heart Cell Atlas distributed with scvi-tools [Litviňuková et al., 2020]. Software baselines are cited via their original papers: scVI [Lopez et al., 2018], Harmony [Korsunsky et al., 2019], Scanorama [Hie et al., 2019], the `scib_metrics` reimplementation [scverse contributors, 2024], and the foundation models GeneJEPA [Litman et al., 2025], scGPT [Cui et al., 2024] and UCE [Rosen et al., 2026], used through their published checkpoints and code under their released terms.

    Guidelines:

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

13. **New assets**

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

    Answer: [Yes]

    Justification: The new asset is the LeCellModel training and analysis code, released under the MIT License at `https://github.com/gilk717/lecellmodel_paper`. Documentation includes the architecture (App. C), training configurations (Tables 4, 5), the displacement-auxiliary algorithm (App. H), and the JEPA-score and gradient-attribution computation details (App. G, App. O). No new datasets are released.

Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or new human-subject data collection. All datasets are publicly released, deidentified single-cell transcriptomic data from prior studies cited in Section 3.2.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The authors did not collect new human-subject data. All datasets used were generated by prior studies; IRB approvals for those studies are described in their original publications.

    Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: LLMs are not used as a component of the methodology; the core method (encoder, projector, SIGReg, JEPA-score, gradient attribution) is fully specified in Section 2.3 and Sections 2.4–2.5 without LLM dependence.

Guidelines:

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