

CellLENS enables cross-domain information fusion for enhanced cell population delineation in single-cell spatial omics data

Received: 9 August 2024

Accepted: 14 April 2025

Published online: 22 May 2025

 Check for updates

Bokai Zhu^{1,2,3,18}, Sheng Gao^{4,18}, Shuxiao Chen^{4,18}, Yuchen Wang^{5,6}, Jason Yeung⁵, Yunhao Bai², Amy Y. Huang^{2,3,7}, Yao Yu Yeo⁵, Guanrui Liao^{5,8}, Shulin Mao^{5,9,10}, Zhenghui G. Jiang¹¹, Scott J. Rodig¹², Ka-Chun Wong⁶, Alex K. Shalek^{1,2,13,14,15}, Garry P. Nolan^{16,19}✉, Sizun Jiang¹⁰✉ & Zongming Ma^{17,19}✉

Delineating cell populations is crucial for understanding immune function in health and disease. Spatial omics technologies offer insights by capturing three complementary domains: single-cell molecular biomarker expression, cellular spatial relationships and tissue architecture. However, current computational methods often fail to fully integrate these multidimensional data, particularly for immune cell populations and intrinsic functional states. We introduce Cell Local Environment and Neighborhood Scan (CellLENS), a self-supervised computational method that learns cellular representations by fusing information across three spatial omics domains (expression, neighborhood and image). CellLENS markedly enhances de novo discovery of biologically relevant immune cell populations at fine granularity by integrating individual cells' molecular profiles with their neighborhood context and tissue localization. By applying CellLENS to diverse spatial proteomic and transcriptomic datasets across multiple tissue types and disease settings, we uncover unique immune cell populations functionally stratified according to their spatial contexts. Our work demonstrates the power of multi-domain data integration in spatial omics to reveal insights into immune cell heterogeneity and tissue-specific functions.

There has been a recent surge in the development of multiplexed imaging technologies capable of simultaneously evaluating 40–100 protein targets^{1,2} or several hundred to thousands mRNA targets^{3,4} at single-cell or subcellular resolution, in their native tissue context. These high-dimensional 'spatial omics' data often require sophisticated approaches for data analysis to uncover biologically meaningful insights. In what follows, we refer to spatial omics data at single-cell or finer resolution as single-cell spatial omics data. Most of the information in these data is contained in the following three complementary domains: (1) single-cell expression levels of measured biomarkers; (2) relative locations of cells; and (3) image of measured

tissues, providing information related to cellular morphology and tissue architecture, through multiple channels. For simplicity, we refer to them hereafter as (1) expression, (2) location and (3) image domains, respectively.

Analysis of single-cell spatial omics data begins with cell phenotyping: single-cell boundary masks are generated from raw images using cell segmentation approaches (for example, ref. 5), which are then used to extract the expression level of molecular features from each cell to create a cell-by-feature data matrix. Then, the process involves applying a clustering algorithm (for example, refs. 6–8) on the matrix, followed by cell-type annotation.

A full list of affiliations appears at the end of the paper. ✉e-mail: gnolan@stanford.edu; sjiang3@bidmc.harvard.edu; zongming.ma@yale.edu

Conventionally, the foregoing cell population delineation (that is, clustering cells into groups with distinct biomolecular profiles or biological functionalities) and identification (that is, cell-type annotation) process rely solely on feature expression levels, while image-level information (for example, cell relative location, marker or tissue image information) is only used during downstream analyses, such as cellular neighborhood or tissue schematic identification^{9,10}.

Tissue image information at different scales can facilitate invaluable insights into cellular-level processes, including but not limited to immunological cell-type and activation state. This is exemplified by how pathologists can identify certain immune cell populations from hematoxylin and eosin (H&E) images, based on information contained in cell morphology, location and surrounding tissue architecture^{11,12}. In addition, spatial information in an image can further differentiate subpopulations of cells, which may not be fully captured by the cell's measured features alone, but rather in concert with the distinct microenvironments that drive immunological cell states beyond cell identities.

Furthermore, image-level information can complement the molecular profile information confined by the antibody or RNA probe panel size. This is exemplified by the distinctive effector and suppressor activities of T cells in close contact with tumors, compared to those farther away^{13,14}. In parallel, B cells located in different layers of niches in the germinal center also represent different populations with distinct functionality and cell state changes^{15,16}. Therefore, ideal cell population delineation and identification in spatial omics data should take advantage of spatial and tissue image information availability.

When one has a set of predefined immune cell populations in the form of an annotated training dataset, new methods^{17,18} have been proposed to leverage the relative locations of cells in a spatial omics dataset when classifying the cells to the prespecified populations. However, as classification methods, they are supervised learning by nature; hence, they do not directly accommodate the definition and discovery of new cell populations unseen in the labeled training data or the partitioning of a population into multiple biologically distinct subpopulations. In addition, image domain information is ignored. For cell population delineation, recent methodologies explored fusing single-cell morphology (that is, individual cell shape information)¹⁹ and spot spatial adjacency information (that is, a form of relative location information among cells)²⁰ to the original expression profiles to generate new numerical representations of cells that could better represent their differences than the measured molecular profiles.

Compared with classification, the primary advantage of applying unsupervised clustering methods on these cell representation vectors is the possibility of *de novo* cell population discovery: clustering methods make no assumption about the existing cell populations in data, making the identification of previously unseen immunological events more accessible. In view of the gap between the available information in data and that leveraged by state-of-the-art computational methods, we postulated that integrating information contained in the location and the image domains of spatial omics data with single-cell molecular profiles would notably enhance granularity when delineating cell populations.

Thus, we present Cell Local Environment and Neighborhood Scan (CellLENS), an unsupervised information fusion algorithm, broadly applicable to different single-cell spatial omics data modalities to learn cross-domain integrative single-cell representation vectors. In particular, CellLENS-learned representation vectors incorporate information from expression, location and image domains. Existing unsupervised clustering algorithms, such as Leiden clustering and its peers, can operate directly on the CellLENS-learned representations rather than conventional feature expressions, enabling spatially informed, fine-grained delineation and *de novo* discovery of immunological and other cell populations across diverse imaging modalities and biological samples.

Results

Overview of CellLENS

The following is a brief description of the CellLENS pipeline (Fig. 1) (see Methods for technical details). CellLENS is an information fusion algorithm operating on single-cell spatial omics modalities, including but not necessarily limited to both spatial proteomics (for example, CODEX²) and spatial transcriptomics (for example, CosMx SMI²¹, Xenium²²). To start, CellLENS curates three pieces of inputs for each cell corresponding to the three information domains: (1) expression domain: an expression profile vector; (2) location domain: a cellular neighborhood context vector, that is, a vector recording composition information of neighboring cells around each cell; and (3) image domain: an image tensor recording local tissue images in multiple channels.

To access the neighborhood context information, we assume that each cell has been given a coarse initial cluster label. Hence, its neighborhood context can be represented by the proportions of its spatial nearest neighbors with different initial cluster labels. When *a priori* human expert annotation is not available, one can obtain initial cluster labels by grouping the cells according to their expression profile similarities. When human expert cell-type annotations are available, the annotated labels can serve as initial cluster labels. For all results reported in this article, we used Leiden clustering⁸ with a resolution level of 0.5 on a feature-expression-induced nearest neighbor graph of each dataset for generating initial cluster labels, without human intervention.

CellLENS leverages a new neural network architecture that we termed LENS-graph neural network (GNN)-duo (Extended Data Fig. 1), for orchestrated information integration across three domains. The architecture consists of two parallel GNNs²³ with an identical node set but distinct network topology, which are connected by an overarching multilayer perceptron (MLP)²⁴ head. In both GNNs, nodes represent cells in the measured tissue section(s) with 1:1 correspondence. One GNN (the spatial GNN in Fig. 1) is constructed on a spatial adjacency graph, where for each node its nodal feature is initialized with the local tissue image encoding the vector of the cell; pairs of nodes are connected if their spatial locations in the tissue section are close. The other GNN (the expression GNN in Fig. 1) is built on a feature similarity graph, where for each node its nodal feature is initialized with the cell's expression profile and pairs of cells are connected if their expression profiles are similar.

To compute the local tissue image encoding vector for each cell, we trained from scratch a convolutional neural network (CNN) model²⁵, which we call LENS-CNN (Extended Data Fig. 1), as an image encoder. LENS-CNN takes each cell's local tissue image, processes it through an AlexNet-like architecture²⁶ and predicts the cell's cellular neighborhood context vector. The resulting fitted encoding vectors from training (a prespecified hidden layer in the trained LENS-CNN) are used as local tissue image encoding vectors for individual cells. LENS-CNN supplies values for initializing LENS-GNN-duo; its training (Methods) is performed before that of the LENS-GNN-duo. After initialization and parallel message passing on the LENS-GNN-duo, the updated nodal vectors of the two GNNs are concatenated and used as inputs for the final MLP head. The MLP head trains and predicts the target vector for each cell, which is the concatenation of the cell's cellular neighborhood context vector and a one-hot vector recording the cell's initial cluster label. Finally, after training, a designated hidden layer of the MLP head is used as the output representation vector of the CellLENS pipeline (Extended Data Fig. 1).

By design, the CellLENS representation fuses single-cell expression, cellular neighborhood and local tissue image information. In the downstream analysis, an existing clustering algorithm^{6–8} can be directly applied to the CellLENS representation vectors for cell population delineation at fine granularity, with no further modification required. For benchmarking purposes, we applied Leiden clustering⁸

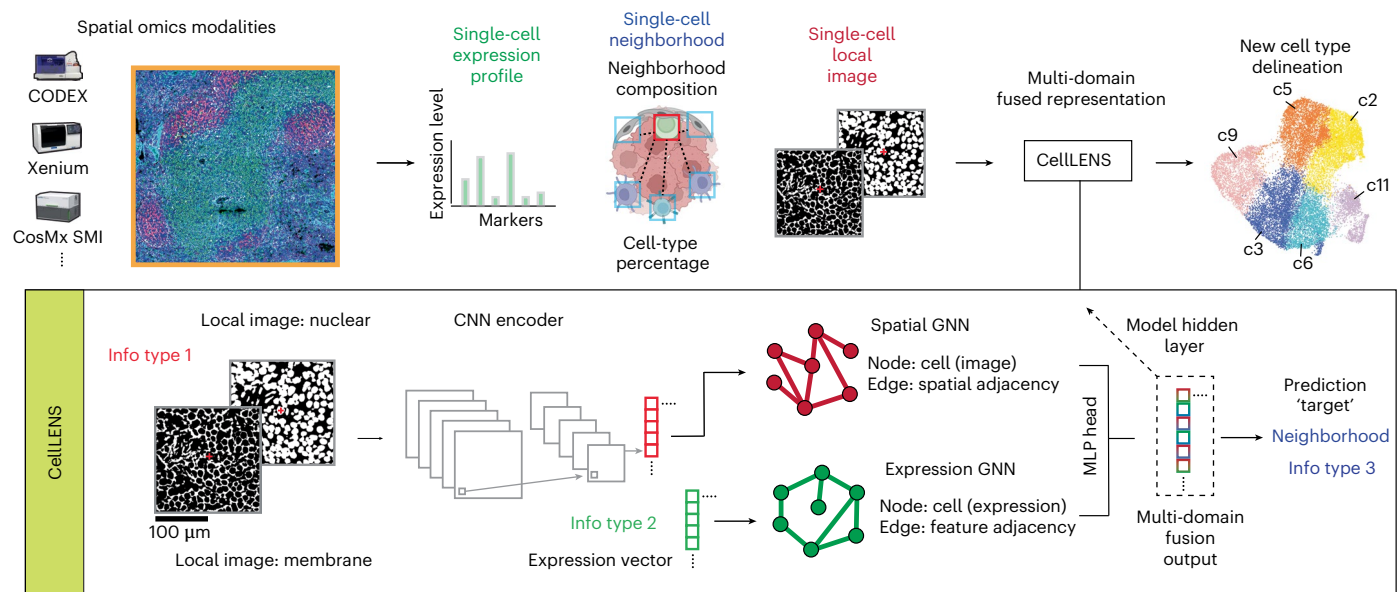


Fig. 1 | Illustration of the CellLENS pipeline. CellLENS is compatible with imaging-based spatial omics modalities with single-cell or finer resolutions (for example, CODEX, Xenium and CosMx). Information from three domains is extracted from each individual cell and its surroundings: (1) single-cell expression profile (for example, measured protein or mRNA features); (2) single-cell location information (for example, cellular neighborhood composition); and (3) single-cell local tissue image information (for example, local images from nuclear and membrane channels). CellLENS takes these three types of information as input. It first uses a CNN encoder to extract features from images of local tissues surrounding each cell. Next, two separate GNN models are constructed in parallel: (1) a ‘spatial GNN’, where each node represents a cell, with the initial node vector assigned as CNN-extracted local image features; nodes are connected according to spatial adjacency; and (2) an ‘expression GNN’, where each node

represents a cell, with the initial node vector assigned as the expression profile; nodes are connected according to expression similarity. The two GNNs are connected by an overarching MLP head that combines the message-passing outputs of the two GNNs for predicting the target vector of each cell, that is, the concatenation of the cell’s feature-based population identity (one-hot) and its neighborhood composition (percentage) vectors. After training, the last layers of the two GNN models are extracted, combined and reduced (via singular value decomposition) to form the final, tri-domain integrated representation vector for each cell. This multi-domain fused representation vector is then used in the downstream analysis for cell-type identification purposes, which is compatible with commonly used unsupervised clustering methods (for example, Leiden clustering). Detailed illustration of the model architecture can be found in Extended Data Fig. 1. Cell illustrations created using [BioRender.com](https://www.biorender.com).

on the resulting CellLENS representation in this study to showcase its applications. Human experts or machine learning algorithms can then determine the biological states of the CellLENS clusters and perform further downstream investigations.

Application to CODEX spleen tissue data

We applied CellLENS to mouse spleen CODEX data (30 markers, 53,500 cells)². To quantitatively evaluate the capacity to delineate cell populations of various cell representation methods, we implemented four metrics: silhouette score; Calinski–Harabasz index, Davies–Bouldin index and modularity score (see Methods for details^{27–30}). These metrics do not require ground-truth cell population information; thus, they are suitable for objectively benchmarking performance for de novo cell-type identification. We computed these metrics for different methods, including: (1) feature: each cell’s feature expression vector; (2) concatenation: the concatenation of each cell’s feature expression vector and neighborhood composition vector; (3) each cell’s canonical correlation analysis (CCA)-transformed representation vector; (4) MOFA+: each cell’s MOFA+³¹ representation vector; (5) SPICEMIX²⁰: each cell’s SPICEMIX representation vector; (6) MUSE¹⁹: each cell’s MUSE representation vector; and (7) CellLENS: each cell’s CellLENS representation vector. We observed improved clustering performance with the CellLENS representation compared to other methods on the mouse spleen CODEX data (Fig. 2a).

We next performed a more nuanced comparison of cell-type identifications based on clustering results on: (1) CellLENS representation; or (2) feature-only expression profile. First, coarse cell-type annotations were generated based on the protein profiles from either CellLENS or feature-only clusters (Fig. 2b and Supplementary Fig. 1). Subsequently, we focused on clusters annotated as B cells. The CellLENS and

feature expression representations produced seven and six Leiden clusters, which were annotated as B cells respectively (Fig. 2c). B cell clusters from the feature expression representation were more intermixed when visualized using UMAP, and less distinctive with regard to B cell-related protein marker expression (Fig. 2c, right); B cell clusters from the CellLENS representation were more well structured when visualized using UMAP, and more distinctive in terms of B cell-related protein marker expression (Fig. 2c, left).

To further validate our findings that B cell subpopulations were better stratified using CellLENS, we investigated the biological relevance of the identified B cell clusters by observing their spatial localization (Fig. 2d). While the conventional feature-only approach could partially delineate B cell subpopulations, including the identification of germinal center (GC) B cells with medium or high CD21 and CD35 expression, most Leiden clusters from feature representation (fc1, fc4, fc6) were mixtures of different B cell subgroups (Fig. 2d, bottom). In comparison, Leiden clusters from the CellLENS representation spatially delineated the B cell subgroups successfully, including not only GC B cells with medium or high CD21 and CD35 expression as identified in the feature-only approach, but also GC CD21-negative and CD35-negative B cell (cc6), splenic zone B cell (cc0) and marginal zone B cell subgroups (cc1, cc10) (Fig. 2d, top)^{32,33}.

Application to CODEX and Xenium tonsil tissue data

We further tested CellLENS on human tonsil CODEX data (46 markers, 102,574 cells)³⁴. Quantitative metrics showed improved clustering using CellLENS (Extended Data Fig. 2a). Cell-type annotation (Extended Data Fig. 2b and Supplementary Fig. 2) revealed an additional replicating non-GC B cell population uniquely identified by CellLENS (red) (Extended Data Fig. 2c). This replicating non-GC B cell population

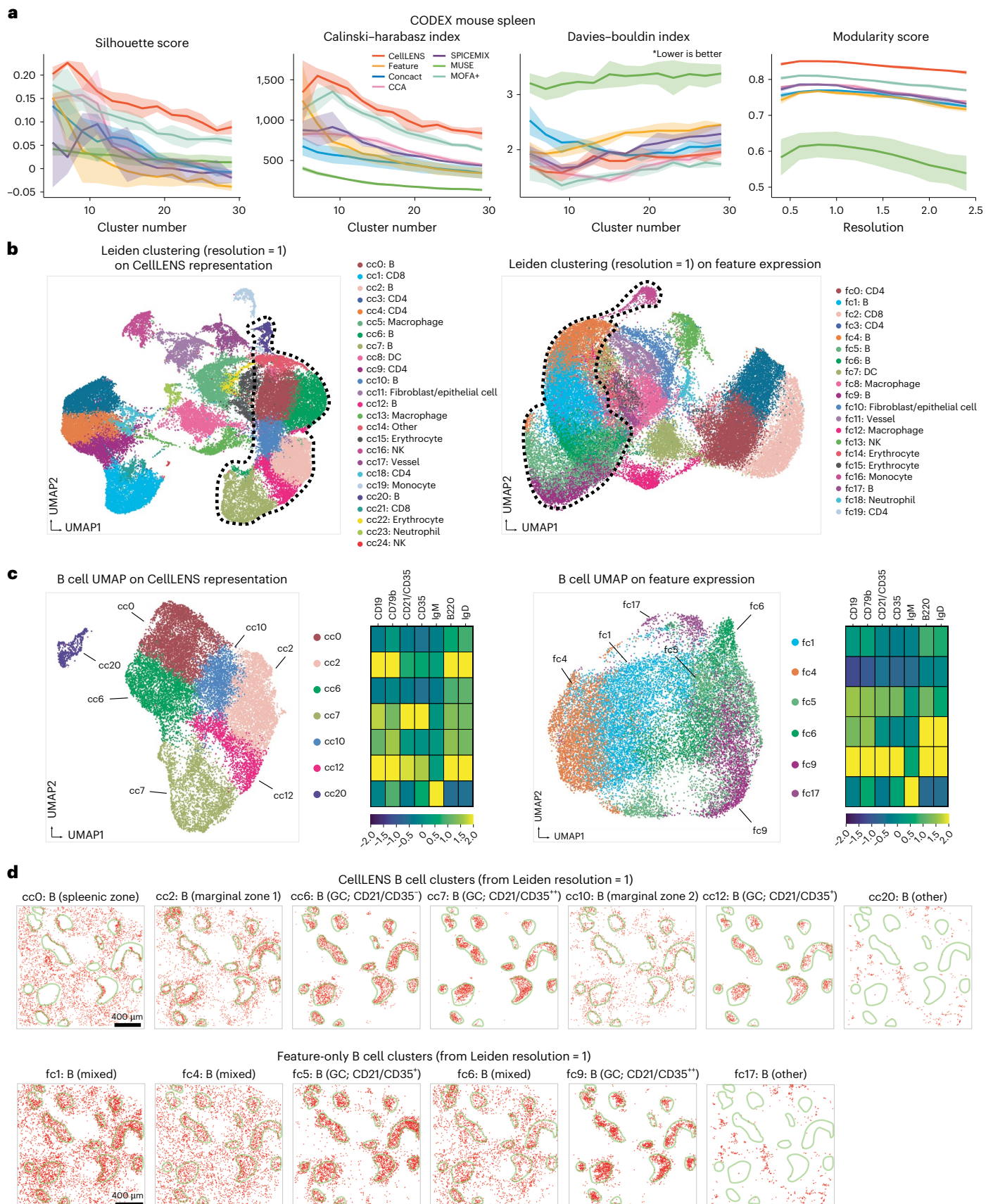


Fig. 2 | Refined B cell subpopulations discovered by CellLENS in a healthy mouse spleen CODEX dataset. **a**, Metric-based evaluation of cell population delineation performances on CODEX mouse spleen tissue. Representations of cells, from seven different methods, were used as input: CellSNP; feature (protein expression table); conncat (protein expression + neighborhood composition table); CCA; MOFA+; SPICEMIX; and MUSE (Methods). Five batches, each with 10,000 randomly selected cells, were tested. The solid line indicates the average and the shading indicates the 95% confidence interval (CI) of the scores. **b**, Uniform manifold approximation and projection (UMAP) visualization of representations and Leiden clustering (cc, CellLENS clusters; fc, feature-only clusters). The cell types of the CellLENS or feature-only clusters were annotated based on the average expression profiles of the clusters. Left, CellLENS

representation. Right, feature expression. The dashed line indicates the B cell subpopulations. **c**, UMAP visualization and B cell-related protein expression profiles of clusters annotated as B cells. Left, B cell clusters from CellLENS and their expression heatmap. Right, B cell clusters from feature expression and their expression heatmap. **d**, Comparison of spatial locations of different B cell clusters identified using CellLENS representation versus feature expression in the spleen tissue. In each plot, the red dots indicate cells from a specific cluster; the green outlines indicate GC boundaries. Top, Spatial locations of B cell subpopulations identified by the CellLENS representation clusters. Bottom, Spatial locations of B cell subpopulations identified using the feature expression clusters. Annotations shown in parentheses summarize the key features and spatial characteristics of the respective clusters. NK, natural killer (cell).

was identified using CellLENS as a distinctive Leiden cluster (cc10 in Extended Data Fig. 2b), but was intermixed with other B cells in a feature-only Leiden cluster (fc8 in Extended Data Fig. 2b), as shown in Extended Data Fig. 2d. This probably reflects similar protein expression profiles as both GC (cc8) and replicating non-GC B cells (cc10) expressed high Ki-67 (Extended Data Fig. 2e). A close visual inspection of the spatial localization of cells in this replicating non-GC B cell population (that is, CellLENS cc10 cluster cells) confirmed their distinctive arrangements bordering GCs (Extended Data Fig. 2f).

To further validate the performance of CellLENS we observed in the CODEX tonsil tissue, we subsequently performed a comparable analysis on a Xenium tonsil dataset (158,259 cells, 377 genes)³⁵. Like the CODEX tonsil data results, we observed improved clustering performance based on the quantitative benchmarking metrics (Fig. 3a). Subsequently, we performed cell-type annotation on both CellLENS representation and feature-only representation (Supplementary Fig. 3a–c), and observed highly comparable improvement of cell-type delineation in B cell populations. This same replicating non-GC B cell population was identified using CellLENS as a distinctive Leiden cluster, but was intermixed with other B cells in a feature-only Leiden cluster (Fig. 3b,c). Subsequent evaluation of the spatial locations of these cells further confirmed our finding, where proliferating B cells clustered on feature-only representation (fc3) showed a mixed spatial pattern (Fig. 3d, left) but proliferating B cells clustered on CellLENS representation showed distinct spatial patterns, including ones only inside GCs (Fig. 3d, middle) and ones only outside GCs (Fig. 3d, right). While similar gene markers were identified and shared among these clusters (Supplementary Fig. 3d), indicating that they are all replicating B cells, we identified genes with significantly differential expression levels across cc7 and cc15, including higher expression of *TCL1A*, *BASPI* and *CD79A* in cc7. Consistent with the previous literature, this indicates that these spatially distinct subgroups have alternative functionalities^{36,37}. Replication across the Xenium and CODEX datasets highlights CellLENS robustness across spatial omics technologies. Given the larger number of features available in the Xenium RNA panel over the CODEX antibody panel, the improvement of CellLENS over other methods and beyond features alone is evident, highlighting the utility of additional leverage of location and image domains.

Fig. 3 | Refined immune cell subpopulations discovered by CellLENS in a tonsil and LN Xenium dataset. **a**, Metrics evaluation of clustering performances on CODEX human tonsil tissue. Embeddings and representations of cells, from seven different methods, were used as input: CellLENS; feature (protein expression table); conncat (protein expression + neighborhood composition table); CCA; MOFA+; SPICEMIX; and MUSE. Clusters (for the calculation of the silhouette score, and the Calinski–Harabasz and Davies–Bouldin indices) were generated with Leiden clustering using the same parameters. The solid line indicates the average and the shading indicates the 95% CI of the scores. **b**, Zoom-in of the UMAP visualization of B cells (green), GC B cells (purple) and replicating non-GC cells (red) on CellLENS embedding (left) and feature-only embedding (right). Cells are colored according to the cell populations identified using the CellLENS

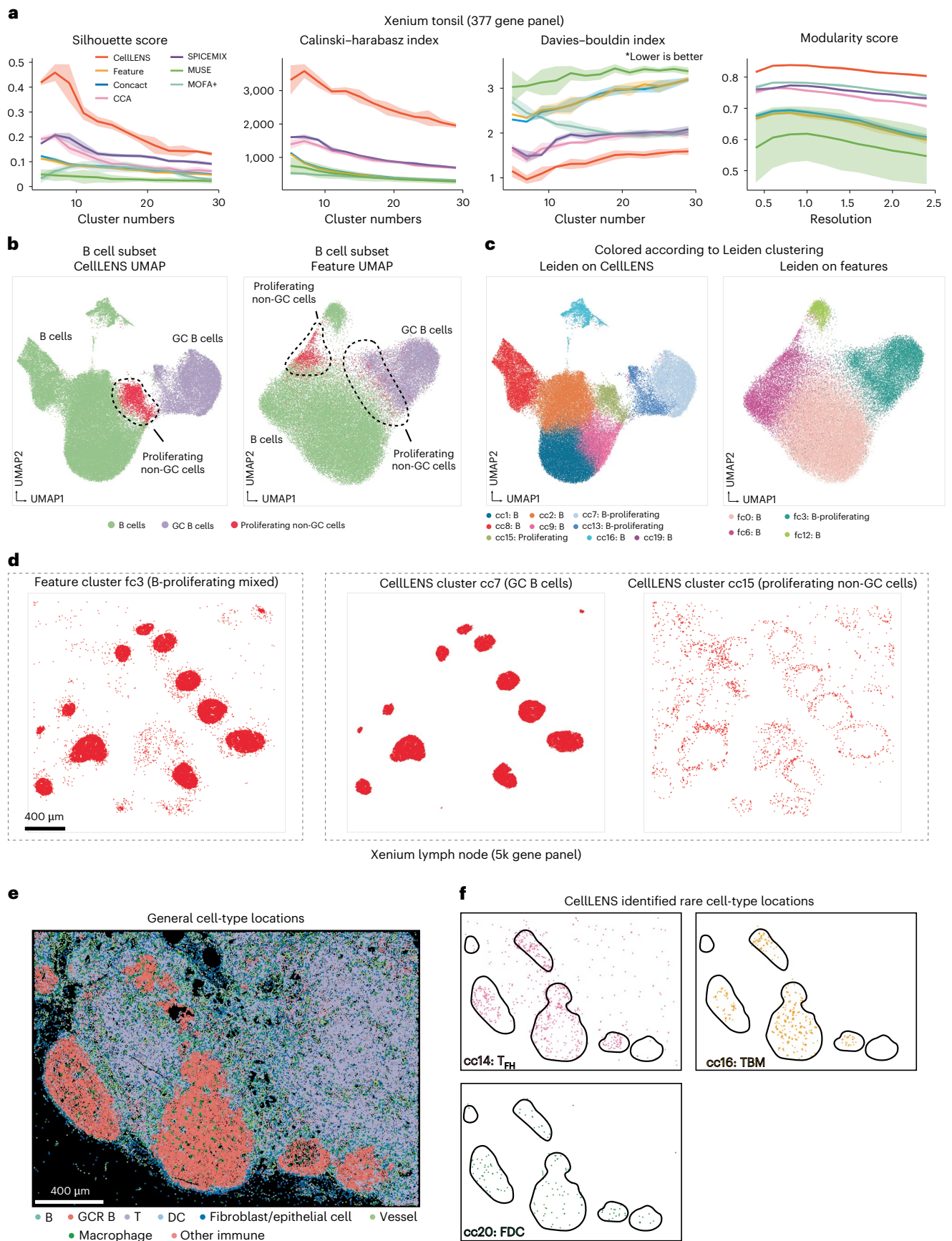
Application to lymph node tissue Xenium data with the 5k gene panel

We continued to evaluate the performance of CellLENS on delineating low-frequency immune cell populations. We tested the pipeline on a human formalin-fixed paraffin-embedded lymph node (LN) tissue measured using Xenium with a gene panel of 5k (ref. 38, Fig. 3e). As in the previous examples, we generated cell embeddings from both the CellLENS and conventional feature-only expression profiles, and both were followed by Leiden clustering and cell-type annotation. The general immune cell populations, including T cells, B and GC B cells, macrophages and others, can be identified using both methods (Extended Data Fig. 3a–c). However, CellLENS enabled the unique delineation of rare but canonical immune cell types of follicular helper T (T_{FH}) cells, tingible body macrophages (TBMs) and follicular dendritic cells (FDCs) by augmenting the identification power with fused spatial and tissue morphological information (Fig. 3f). In contrast, the conventional method failed to discern these low-frequency cell types with information from the measured transcripts only (Supplementary Fig. 4a), even when the panel size is as large as 5k. This example further highlights the novelty and robustness of our CellLENS pipeline.

Application to classic Hodgkin lymphoma cancer tissue CODEX data

We next evaluated CellLENS in the classic Hodgkin lymphoma (cHL) tumor microenvironment (TME). We applied CellLENS on a CODEX cHL dataset (45 markers, 143,730 cells)³⁹. We quantitatively compared CellLENS and four other methods (Fig. 4a), performing cell annotation on CellLENS and feature-only representations (Fig. 4b and Supplementary Fig. 5a). We previously linked systemic T cell dysfunction in cHL TMEs directly to tumor proximity using spatial proteomics and transcriptomics¹⁴. We postulated that CellLENS would be an effective approach to facilitate the finer delineation of T cell subpopulations in the context of cHL. Clustering on the CellLENS representation generated six distinct CD4 T cell subpopulations, each occupying distinctive spatial regions relative to the tumor in the TME (Fig. 4c,d). In contrast, CD4 T cells obtained from clustering on the feature-only representation did not exhibit these diverse spatial localizations and niche profiles (Supplementary Fig. 5b). Two interesting CD4 T cell subpopulations

clusters in both UMAPs. **c**, Zoom-in of the UMAP visualization of the same cell types shown in **c** but colored according to the cluster numbers from CellLENS embedding (left) and feature-only embedding (right). **d**, Visualization of the B-proliferating cluster locations. Left, B-proliferating cells from the feature-only representation clustering (fc3), showing a mixed spatial pattern. Middle, B-proliferating cells from the CellLENS representation clustering (cc7), showing a GC spatial pattern. Right, B-proliferating cells from the CellLENS representation clustering (cc15), showing a non-GC spatial pattern. **e**, General cell-type locations on the Xenium human LN data (5k gene panel). **f**, Uniquely identified rare immune cell populations using CellLENS and their corresponding locations with the GC boundaries outlined in black.



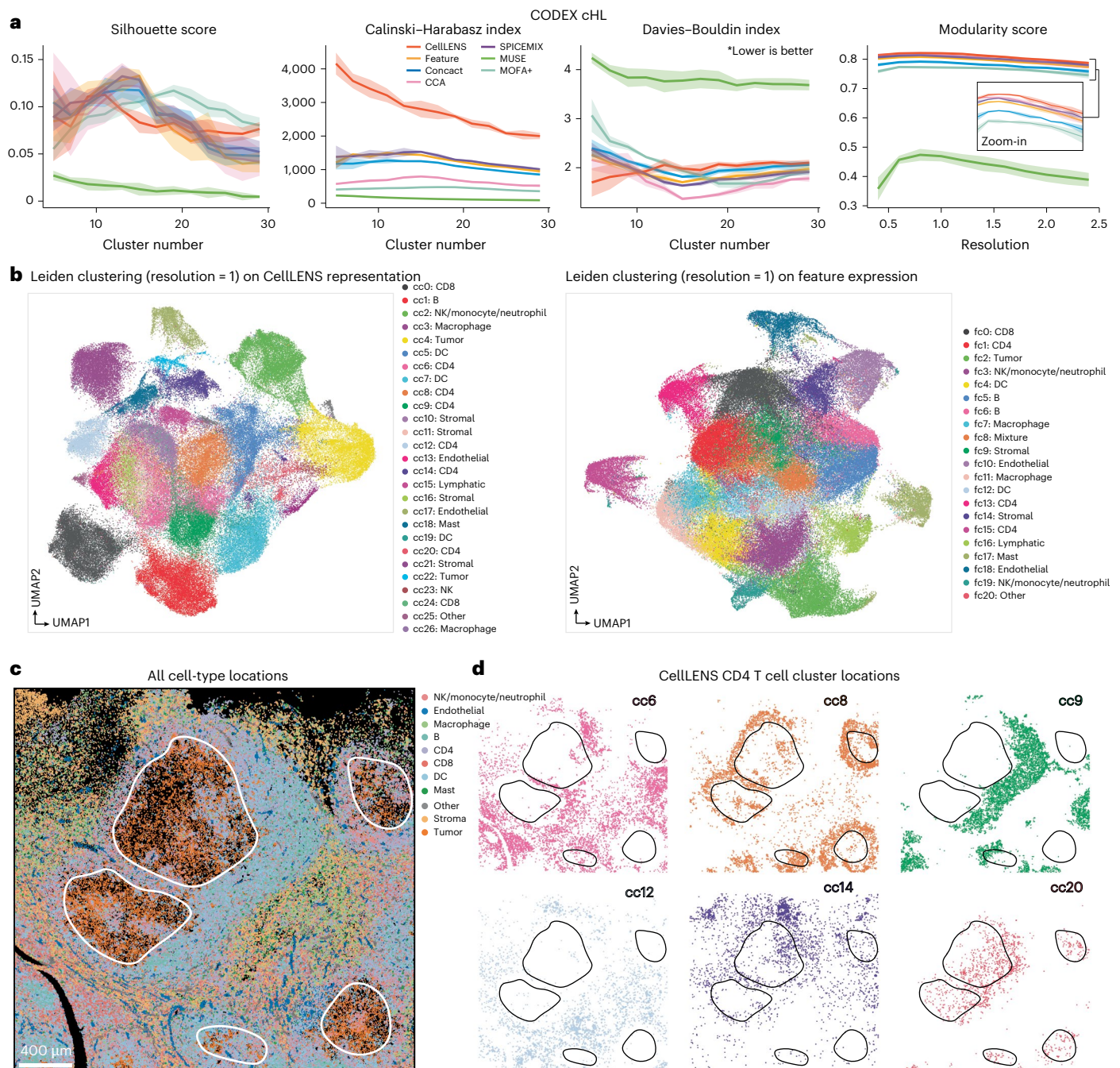


Fig. 4 | Refined T cell subpopulations in TMEs discovered using CelliENS in a cHL tumor CODEX dataset. **a**, Metric-based evaluations of cell population delineation performances on CODEX human cHL tissue. Representations of cells, from seven different methods, were used as input: CelliENS; feature (protein expression table); connect (protein expression + neighborhood composition table); CCA; MOFA+; SPICEMIX; and MUSE (Methods). Five batches, each with 10,000 randomly selected cells were tested. The solid line indicates the average and the shading indicates the 95% CI of the scores. **b**, UMAP visualization of the representations and Leiden clustering (cc, CelliENS clusters; fc, feature-

only clusters). The cell types of the CelliENS or feature-only clusters were annotated based on the average expression profiles of the clusters. Left, CelliENS representation. Right, feature expression. **c**, Visualization of cell-type spatial locations in the cHL tissue, colored according to the annotations on the CelliENS clusters. The black regions are empty spaces. The white outlines indicate the borders of the cHL tumor regions. **d**, Visualization of the spatial locations of different CD4 T cell subpopulations identified by the CelliENS representation clusters. The black outlines indicate the borders of the cHL tumor regions.

identified using CelliENS representation are (Fig. 4d): CelliENS cc8 (top row, middle), a subpopulation that occupies the boundary of the cHL tumor patch; and CelliENS cc20 (bottom row, right), a subpopulation that infiltrated inside the cHL tumor patch. We further observed that only the tumor-infiltrated CD4 T cell subpopulation (CelliENS cc20), but not the tumor boundary CD4 T cell subpopulation (CelliENS cc8), exhibited an elevated expression of the T cell

dysfunctional markers LAG3 and PD-1 (Supplementary Fig. 5c). These results further elucidate the importance of understanding T cell dysregulation in the cHL TME^{40,41}. More importantly, the CelliENS-enabled analysis, alongside other recent spatial omics studies¹⁴, shed light on the importance of spatial aspects in dysfunctional T cell-cHL TME interactions in the orchestrated immunological responses to tumor in cHL and beyond.

Application to hepatocellular carcinoma tissue CosMx SMI data with the 6k gene panel

We extended CellLENS from spatial proteomics to spatial transcriptomics in a hepatocellular carcinoma (HCC) CosMx SMI dataset. We deployed CellLENS and other methods on an HCC tumor CosMx SMI dataset (54,867 cells, 997 genes)⁴². CellLENS outperformed feature-only embeddings across all quantitative clustering metrics (Fig. 5a). We then performed cell-type annotation based on the Leiden clustering results, either from CellLENS representation or feature-only representation (Extended Data Fig. 4a,b). Between the two methods, CellLENS allowed finer delineation of cell subpopulations. For instance, two clusters of macrophages were identified using feature-only representation (Extended Data Fig. 4a,b). In contrast, an increased number of macrophage clusters, each with a distinctive niche occupation, were obtained from the CellLENS representation (Fig. 5b). We observed that the macrophage subpopulation CellLENS cc6 exhibited a unique spatial occupation of the tumor–immune interface, which was missing in the feature-only results (Fig. 5b and Extended Data Fig. 4c). Further investigation consolidated the functional differences of subpopulation CellLENS cc6 from other macrophages in this dataset. Differential expression analysis showed cc6 uniquely upregulated genes, including *CIQA*, *CIQB*, *CIQC*, *LYZ* and *HLA* family members (*HLA.DQA1*, *HLA.DPA1*) (Fig. 5c).

To explore functional differences, we performed module scoring using pan-specific and liver-specific macrophage gene lists^{43,44} (Fig. 5d). We found that CellLENS cc6 macrophages skewed toward a more ‘M1-like’ and pro-inflammatory phenotype, while the other macrophages displayed a more immunoregulatory phenotype. Given their proximity and pro-inflammatory nature, we hypothesized active cc6 macrophage–tumor interactions. Spatial ligand–receptor (LR) analysis using SpatialDM⁴⁵ revealed enriched LR interactions specifically at the cc6 macrophage–tumor boundary (Fig. 5e). We further performed differential expression analysis on LR pairs between CellLENS cc6 and other macrophages, to further support a model in which the CellLENS cc6 macrophages had increased LR interaction profiles ($P_{\text{adj}} < 0.05$, Benjamini–Hochberg correction), including integrin receptors and their respective ligands compared to other macrophages (Fig. 5f). CellLENS identified cc6 as tumor–boundary macrophages with unique functions, LR profiles and improved tumor delineation (Supplementary Fig. 6). We next postulated that the HCC tumor subpopulations infiltrated by these CellLENS cc6 macrophages (CellLENS cc7, cc8, cc17) would also possess different gene programs and pathway usages compared to other tumor populations even in the same tissue. To test this, we performed unsupervised gene expression program (GEP) identification using consensus nonnegative matrix factorization (CNMF)⁴⁶ (Supplementary Fig. 7a) and selected the most significantly upregulated or downregulated GEPs for comparison between tumor clusters with macrophage infiltrates (CellLENS cc7,

cc8, cc17), compared to other tumor cells in the same tissue. We then annotated the GEPs according to their top 20 contributing genes, using Gene Ontology analysis (Fig. 5g), and observed elevated gene program activation and usage related to immune cell chemotaxis, cytokine signaling and extracellular matrix (ECM) disassembly in macrophage-infiltrated tumor cells. In contrast, other tumor cells showed enhanced gene program usage related to NOS (nitric oxide synthase) synthesis, interleukin-10 (IL-10) regulation and proliferation (Supplementary Fig. 7b). In summary, we demonstrated that CellLENS representation facilitated the delineation and identification of a tumor-associated macrophage (TAM) subpopulation with infiltration tendencies and pro-inflammatory functionalities. This TAM subpopulation was potentially altering tumor programs via spatial interactions, consistent with findings from previous studies in HCC tissues^{47,48}. Remarkably, the discovery of this TAM subpopulation was entirely unsupervised; no training or reference dataset has been involved.

Comprehensive benchmarking

To evaluate the robustness of CellLENS with respect to different tuning parameter choices, we tested ranges of values for four different tuning parameters on the mouse spleen CODEX dataset (Supplementary Fig. 8): (1) resolution used in Leiden clustering for acquiring the cell identity cluster numbers and cell neighborhood composition; (2) number of neighbors (K) used when searching for the nearest neighborhood to calculate the cell neighborhood composition; (3) image size (in pixels) for acquiring the LENS-CNN encoding; and (4) binarization threshold value for acquiring the LENS-CNN encoding. To evaluate the advantage of the LENS-GNN-duo architecture, we benchmarked the model training performance, using LOSS values calculated on randomly selected test datasets as a proxy, of the full LENS-GNN-duo pipeline against the single-GNN alternatives (that is, using only the expression GNN or the spatial GNN with an MLP head) (Supplementary Fig. 9).

While CellLENS provides an unsupervised approach for de novo identification of delineated cell types from spatial omics data, supervised methods, for example, label transfer from single-cell RNA sequencing annotations^{49–51}, may also be useful in certain scenarios. However, their performance can be influenced by several factors, sometimes resulting in subpar labeling accuracy for specific immune cell populations (Supplementary Fig. 10).

The CellLENS pipeline allows users to directly input initial cell-type labels (for example, labels predicted using label transfer) instead of performing coarse Leiden clustering at the start. While CellLENS still performs as expected, we advise caution because results may be influenced by the accuracy of the input labels (Supplementary Fig. 11). We further examined the impact of the resolution level of Leiden clustering used during CellLENS initialization. The performance of CellLENS remained stable with respect to this initialization tuning

Fig. 5 | CellLENS-enabled delineation of biologically distinct macrophage subpopulations in an HCC tumor CosMx SMI dataset. **a**, Metric-based evaluation of cell population delineation performances on CosMx SMI human HCC tissue. Representations of cells, from seven different methods, were used as input: CellSNP; feature (protein expression table); concast (protein expression + neighborhood composition table); CCA; MOFA+; SPICEMIX; and MUSE (Methods). Five batches, each with 10,000 randomly selected cells were tested. The solid line indicates the average and the shading indicates the 95% CI of the scores. **b**, Visualization of spatial locations of different cell populations, including all cell types (first panel colored according to the cell-type annotation obtained from the CellLENS clusters; black regions indicate empty spaces) and different macrophage subpopulations identified by the CellLENS representation clusters (second to fourth panels). In each of the second to fourth panels, all tumor cells, and macrophage cells from a specific CellLENS cluster, are colored, while other cells and empty spaces are shown in black. **c**, Volcano plot of differentially expressed genes between the CellLENS cc6 cluster and other macrophage

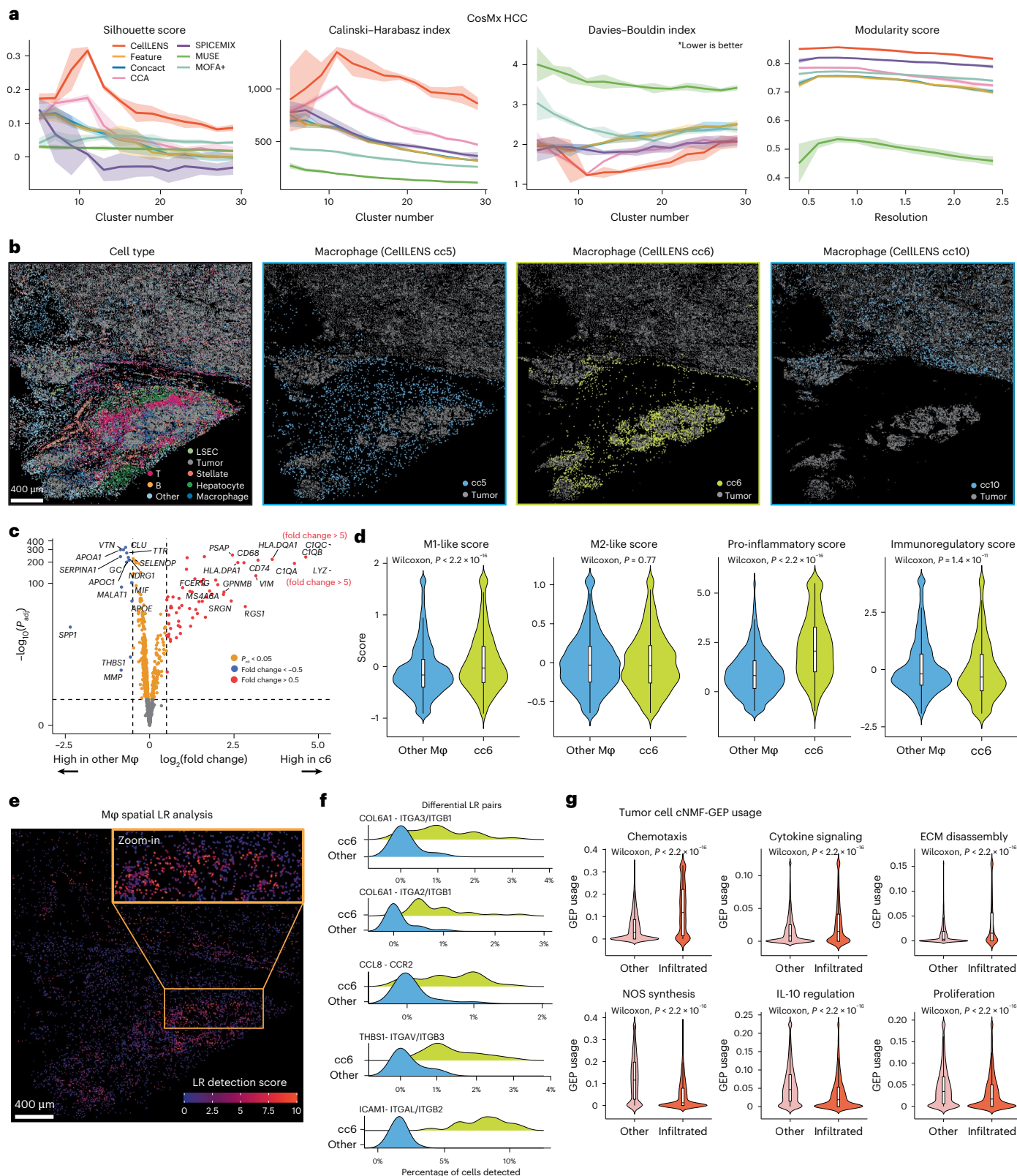
clusters. Statistics were generated using Limma. **d**, Comparison of module score values⁵¹ between CellLENS cc6 ($n = 2,569$) and all other macrophage cells ($n = 5,089$). ‘M1-like’ and ‘M2-like’ scores were calculated using genes from ref. 43. Splenic macrophage-specific ‘pro-inflammatory’ and ‘immunoregulatory’ scores were calculated using the genes from ref. 44. An unpaired two-sided Wilcoxon rank-sum test was used to determine the P values. The box plot indicates the 25th to 75th percentiles (box boundary), the median values (line in the box), and the minimum and maximum values (whiskers). **e**, Visualization of the spatial distribution of all macrophages and their respective LR interaction detection score levels. The detection score was calculated based on significant LR interaction pairs between macrophages and tumor cells⁴⁵. **f**, Top ten most frequent LR interaction pairs associated with CellLENS cc6 macrophages. **g**, GEP usage scores⁴⁶ among tumor cells, stratified according to infiltration (by macrophage) status. Infiltrated tumors $n = 5,953$; other tumors $n = 18,737$. An unpaired two-sided Wilcoxon rank-sum test was used to determine the P values. The setup of the box plots is the same as in **d**. LSEC, liver sinusoidal endothelial cell.

parameter on both the CODEX spleen and Xenium tonsil datasets, except at extreme resolution levels (Supplementary Figs. 12 and 13).

In our comparisons presented in the previous sections, we fixed the resolution level (resolution = 1) for the final Leiden clustering on both CellLENS and feature-only embeddings. Alternatively, we also tested fixing the total number of resulting clusters across comparisons on both the CODEX mouse spleen and Xenium human tonsil datasets

by increasing the resolution levels for feature-only embeddings. Strikingly, we observed that further increasing the resolution in Leiden clustering of feature-only embeddings did not recover the initially missed populations (Supplementary Figs. 14 and 15).

To evaluate the stability of the CellLENS results, we next tested the pipeline on two adjacent human tonsil sections, both investigated using CODEX (Supplementary Fig. 16). While these two sections



underwent different epitope retrieval conditions⁵², which resulted in different expression intensity distributions across measured biomarkers, CellLENS identified largely consistent cell populations between these sections (Extended Data Fig. 5), showcasing the stability of the algorithm.

In addition to the methods outlined in Figs. 2a, 3a, 4a and 5a, we benchmarked CellLENS against multiple other state-of-the-art algorithms for cell population discovery, using four clustering evaluation metrics. The additional methods are SpaGCN⁵³, stLearn⁵⁴, SEDR⁵⁵, BANKSY⁵⁶ and CellCharter⁵⁷. While some of these methods were not originally designed for single-cell-level cell-type identification (that is, they were developed for spot-level data or focused on spatial domain identification), we found it pertinent to include them during benchmarking. Thus, we assessed 11 methods across five spatial omics datasets: CODEX mouse spleen; Xenium tonsil; CODEX tonsil; CODEX cHL; and CosMx SMI HCC. CellLENS consistently outperformed other methods across these datasets (Extended Data Fig. 6 and Supplementary Figs. 17–19).

SpaGCN, ranked second, also had a consistently high performance across the board, although it was limited to no more than approximately 50k cells at the time of testing. (Thus, we were only able to perform the analysis on a subregion of the data when the number of cells exceeded this threshold.) To further investigate the difference of biological insights provided by SpaGCN and CellLENS in cell-type delineation, we analyzed the identified cell types from Leiden clustering on SpaGCN's embeddings, where it was feasible to run on the full dataset (that is, CODEX mouse spleen and CosMx SMI HCC), according to the approach described in the previous sections. We found that while SpaGCN had good metric performances, it was unable to identify critical spatially resolved cell subpopulations specifically identified by CellLENS (Extended Data Figs. 7 and 8).

We benchmarked the influence of several factors on image morphology feature extraction in the context of the LENS-CNN model on the CODEX mouse spleen dataset. First, we quantified model losses when using 'none', 'single-channel' or 'dual-channel' image information: (1) CellLENS without the CNN component, that is, completely ignoring the image domain; (2) CellLENS with nuclear-only images; and (3) CellLENS with both nuclear and membrane images (the default in this study) (Supplementary Fig. 20). Second, we tested whether more sophisticated computer vision models (ResNet50 (ref. 58), ViT⁵⁹) with frozen pretrained weights could be used to extract image features for CellLENS instead of retraining weights in the simpler AlexNet-like architecture (the default in CellLENS) on the spatial omics data (Extended Data Fig. 9a). Third, we evaluated the impact of replacing the original AlexNet-like CNN architecture with more complex architectures (ResNet50 and ViT) when training the CNN encoder in CellLENS (Extended Data Fig. 9b). We found that simply using pretrained ResNet50 or ViT led to worse performance. In addition, replacing the AlexNet-like CNN encoder in CellLENS with a retrained ViT resulted in comparable performance but with a higher training time (Extended Data Fig. 10) because of the extra model complexity, whereas replacing it with a retrained ResNet50 resulted in worse performance. For all these reasons, the dataset underwent an 80/20 training/test split. Training was performed on the training set only and prediction performances were evaluated using the test set only. All analyses described in this paragraph were done on the Xenium tonsil dataset.

We evaluated the scalability of the CellLENS pipeline to large spatial datasets. To this end, we generated a series of datasets with varying cell counts (approximately 10k, 50k, 150k, 500k and 1,500k) by spatially aware subsetting or duplication of the Xenium human tonsil dataset. Benchmarking was performed on an NVIDIA A5000 graphics processing unit and a PCIe 4.0 solid-state drive disk. Our results demonstrate that CellLENS can process several million cells within a reasonable time frame using affordable, easy-to-access computational resources for most biological laboratories (Extended Data Fig. 10).

Discussion

In this study, we developed a new geometric deep learning pipeline, CellLENS, which integrates complementary information from feature expressions (including proteins or RNAs), neighborhood context and local tissue image, to produce for each cell a comprehensive representation vector suitable for a wide range of downstream analyses, including but not limited to clustering analysis for spatial-and-tissue-context-aware cell population delineation and identification, and the analysis of their functional differences.

The CellLENS pipeline uses a new architecture that consists of two GNNs for coding expression similarity and spatial proximity among cells in parallel, thus allowing smooth information diffusion of biomarker expressions and local tissue images in their respective natural domains for better information fusion. CellLENS exhibits robustness in enhancing cell population delineation across diverse spatial omics datasets collected from different tissues and disease settings using diverse technologies and different panel sizes. Notably, CellLENS's compatibility with current de novo cell-type identification processes allows for straightforward applications of existing clustering algorithms to the learned representations. Thus, we anticipate the adoption of CellLENS in the spatial omics community.

We showcased the application of CellLENS on imaging-based spatial modalities (CODEX, CosMx SMI and Xenium). It is conceivable that this pipeline would also be compatible with sequencing-based spatial modalities (for example, Slide-seq, Stereo-seq, high-definition spatial transcriptomics^{60–62}), as such datasets are usually accompanied by H&E or fluorescent images of the same or an adjacent tissue section, on top of the spatially resolved genomic readouts; hence, inputs to the CellLENS pipeline can be further curated and tested in the future.

While the current study focused on showcasing the efficacy of CellLENS in delineating immune cell populations and their associated functional states, the utility of the learned representation vectors can be generalized to other biological contexts. For instance, they can serve as engineered features in diagonal integration tasks, complementing other recently developed methodologies^{15,16}. In addition, they can serve as inputs to spatial neighborhood analysis pipelines^{63,64} to identify different tumor-immune microenvironments and other neighborhoods of biological interest. Furthermore, the representation vectors can serve as inputs to machine learning models aimed at predicting disease outcomes directly from single-cell information. Overall, the learned all-encompassing cell representation that integrates expression, location and image domains summarizes key information provided by spatial omics datasets and holds great promise for improved and better-informed downstream analyses with diverse biological objectives.

Online content

Any methods, additional references, Nature Portfolio reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information; details of author contributions and competing interests; and statements of data and code availability are available at <https://doi.org/10.1038/s41590-025-02163-1>.

References

- Giesen, C. et al. Highly multiplexed imaging of tumor tissues with subcellular resolution by mass cytometry. *Nat. Methods* **11**, 417–422 (2014).
- Goltsev, Y. et al. Deep profiling of mouse splenic architecture with codex multiplexed imaging. *Cell* **174**, 968–981 (2018).
- Chen, K. H., Boettiger, A. N., Moffitt, J. R., Wang, S. & Zhuang, X. Spatially resolved, highly multiplexed RNA profiling in single cells. *Science* **348**, aaa6090 (2015).
- Wang, X. et al. Three-dimensional intact-tissue sequencing of single-cell transcriptional states. *Science* **361**, eaat5691 (2018).

5. Greenwald, N. F. et al. Whole-cell segmentation of tissue images with human-level performance using large-scale data annotation and deep learning. *Nat. Biotechnol.* **40**, 555–565 (2022).
6. Samusik, N., Good, Z., Spitzer, M. H., Davis, K. L. & Nolan, G. P. Automated mapping of phenotype space with single-cell data. *Nat. Methods* **13**, 493–496 (2016).
7. Levine, J. H. et al. Data-driven phenotypic dissection of AML reveals progenitor-like cells that correlate with prognosis. *Cell* **162**, 184–197 (2015).
8. Traag, V. A., Waltman, L. & Van Eck, N. J. From Louvain to Leiden: guaranteeing well-connected communities. *Sci. Rep.* <https://doi.org/10.1038/s41598-019-41695-z> (2019).
9. Bhate, S. S., Barlow, G. L., Schürch, C. M. & Nolan, G. P. Tissue schematics map the specialization of immune tissue motifs and their appropriation by tumors. *Cell Syst.* **13**, 109–130 (2022).
10. Schürch, C. M. et al. Coordinated cellular neighborhoods orchestrate antitumoral immunity at the colorectal cancer invasive front. *Cell* **182**, 1341–1359 (2020).
11. Kumar, V., Abbas, A. & Aster, J. *Robbins & Cotran Pathologic Basis of Disease (Robbins Pathology)* (Saunders, 2014).
12. Eroschenko, V. P. *DiFiore's Atlas of Histology with Functional Correlations* (Lippincott, Williams & Wilkins, 2013).
13. Phillips, D. et al. Immune cell topography predicts response to PD-1 blockade in cutaneous T cell lymphoma. *Nat. Commun.* **12**, 6726 (2021).
14. Yeo, Y. Y. et al. Epstein–Barr virus orchestrates spatial reorganization and immunomodulation within the classic Hodgkin lymphoma tumor microenvironment. Preprint at *bioRxiv* <https://doi.org/10.1101/2024.03.05.583586> (2024).
15. Chen, S. et al. Integration of spatial and single-cell data across modalities with weakly linked features. *Nat. Biotechnol.* **42**, 1096–1106 (2024).
16. Zhu, B. et al. Robust single-cell matching and multimodal analysis using shared and distinct features. *Nat. Methods* **20**, 304–315 (2023).
17. Brbić, M. et al. Annotation of spatially resolved single-cell data with STELLAR. *Nat. Methods* **19**, 1411–1418 (2022).
18. Amitay, Y. et al. CellSighter: a neural network to classify cells in highly multiplexed images. *Nat. Commun.* **14**, 4302 (2023).
19. Bao, F. et al. Integrative spatial analysis of cell morphologies and transcriptional states with MUSE. *Nat. Biotechnol.* **40**, 1200–1209 (2022).
20. Chidester, B., Zhou, T., Alam, S. & Ma, J. SPICEMIX enables integrative single-cell spatial modeling of cell identity. *Nat. Genet.* **55**, 78–88 (2023).
21. He, S. et al. High-plex imaging of RNA and proteins at subcellular resolution in fixed tissue by spatial molecular imaging. *Nat. Biotechnol.* **40**, 1794–1806 (2022).
22. Janesick, A. et al. High resolution mapping of the tumor microenvironment using integrated single-cell, spatial and in situ analysis. *Nat. Commun.* **14**, 8353 (2023).
23. Kipf, T. N. & Welling, M. Semi-supervised classification with graph convolutional networks. Preprint at <https://arxiv.org/abs/1609.02907> (2016).
24. Goodfellow, I., Bengio, Y. & Courville, A. *Deep Learning* (MIT, 2016).
25. Krizhevsky, A., Sutskever, I. & Hinton, G. E. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25* https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf (NeurIPS, 2012).
26. Krizhevsky, A., Sutskever, I. & Hinton, G. E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **60**, 84–90 (2017).
27. Rousseeuw, P. J. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* **20**, 53–65 (1987).
28. Davies, D. L. & Bouldin, D. W. A cluster separation measure. *IEEE Trans. Pattern Anal. Mach. Intell.* **2**, 224–227 (2009).
29. Caliński, T. & Harabasz, J. A dendrite method for cluster analysis. *Commun. Stat. Theory Methods* **3**, 1–27 (1974).
30. Newman, M. E. J. Modularity and community structure in networks. *Proc. Natl Acad. Sci. USA* **103**, 8577–8582 (2006).
31. Argelaguet, R. et al. MOFA+: a statistical framework for comprehensive integration of multi-modal single-cell data. *Genome Biol.* <https://doi.org/10.1186/s13059-020-02015-1> (2020).
32. Cumpelik, A. et al. Dynamic regulation of B cell complement signaling is integral to germinal center responses. *Nat. Immunol.* **22**, 757–768 (2021).
33. Steiniger, B. S. Human spleen microanatomy: why mice do not suffice. *Immunology* **145**, 334–346 (2015).
34. Kennedy-Darling, J. et al. Highly multiplexed tissue imaging using repeated oligonucleotide exchange reaction. *Eur. J. Immunol.* **51**, 1262–1277 (2021).
35. Human tonsil data: Xenium human multi-tissue and cancer panel 1 standard. *10x Genomics* www.10xgenomics.com/datasets/human-tonsil-data-xenium-human-multi-tissue-and-cancer-panel-1-standard (2024).
36. Herling, M. et al. High TCL1 levels are a marker of B-cell receptor pathway responsiveness and adverse outcome in chronic lymphocytic leukemia. *Blood* **114**, 4675–4686 (2009).
37. Papa, I. & Vinuesa, C. G. Synaptic interactions in germinal centers. *Front. Immunol.* **9**, 1858 (2018).
38. FFPE human lymph node with 5k pan tissue and pathways panel. *10x Genomics* www.10xgenomics.com/datasets/preview-data-xenium-prime-gene-expression (2024).
39. Shaban, M. et al. MAPS: pathologist-level cell type annotation from tissue images through machine learning. *Nat. Commun.* **15**, 28 (2024).
40. Veldman, J. et al. CD4⁺ T cells in classical Hodgkin lymphoma express exhaustion associated transcription factors TOX and TOX2: characterizing CD4⁺ T cells in Hodgkin lymphoma. *Oncoimmunology* **11**, 2033433 (2022).
41. Nagasaki, J. et al. The critical role of CD4⁺ T cells in PD-1 blockade against MHC-II-expressing tumors such as classic Hodgkin lymphoma. *Blood Adv.* **4**, 4069–4082 (2020).
42. CosMx SMI human liver FFPE dataset. *NanoString* <https://nanosttring.com/products/cosmx-spatial-molecular-imager/ffpe-dataset/human-liver-rna-ffpe-dataset/> (2025).
43. Cheng, S. et al. A pan-cancer single-cell transcriptional atlas of tumor infiltrating myeloid cells. *Cell* **184**, 792–809 (2021).
44. MacParland, S. A. et al. Single cell RNA sequencing of human liver reveals distinct intrahepatic macrophage populations. *Nat. Commun.* **9**, 4383 (2018).
45. Li, Z., Wang, T., Liu, P. & Huang, Y. SpatialDM for rapid identification of spatially co-expressed ligand–receptor and revealing cell–cell communication patterns. *Nat. Commun.* **14**, 3995 (2023).
46. Kotliar, D. et al. Identifying gene expression programs of cell-type identity and cellular activity with single-cell RNA-seq. *eLife* **8**, e43803 (2019).
47. Yuan, Y. et al. Mechanisms of tumor-associated macrophages affecting the progression of hepatocellular carcinoma. *Front. Pharmacol.* **14**, 1217400 (2023).
48. Tian, Z., Hou, X., Liu, W., Han, Z. & Wei, L. Macrophages and hepatocellular carcinoma. *Cell Biosci.* <https://doi.org/10.1186/s13578-019-0342-7> (2019).
49. Mages, S. et al. TACCO unifies annotation transfer and decomposition of cell identities for single-cell and spatial omics. *Nat. Biotechnol.* **41**, 1465–1473 (2023).

50. Aran, D. et al. Reference-based analysis of lung single-cell sequencing reveals a transitional profibrotic macrophage. *Nat. Immunol.* **20**, 163–172 (2019).
51. Hao, Y. et al. Integrated analysis of multimodal single-cell data. *Cell* **184**, 3573–3587 (2021).
52. Lee, J. L. & Schaffner, L. Benchmarking tissue preparation_pilotdata. *Zenodo* <https://doi.org/10.5281/zenodo.11410274> (2024).
53. Hu, J. et al. SpaGCN: integrating gene expression, spatial location and histology to identify spatial domains and spatially variable genes by graph convolutional network. *Nat. Methods* **18**, 1342–1351 (2021).
54. Pham, D. et al. Robust mapping of spatiotemporal trajectories and cell–cell interactions in healthy and diseased tissues. *Nat. Commun.* **14**, 7739 (2023).
55. Xu, H. et al. Unsupervised spatially embedded deep representation of spatial transcriptomics. *Genome Med.* **16**, 12 (2024).
56. Singhal, V. et al. BANKSY unifies cell typing and tissue domain segmentation for scalable spatial omics data analysis. *Nat. Genet.* **56**, 431–441 (2024).
57. Varrone, M., Tavernari, D., Santamaria-Martínez, A., Walsh, L. A. & Ciriello, G. CellCharter reveals spatial cell niches associated with tissue remodeling and cell plasticity. *Nat. Genet.* **56**, 74–84 (2024).
58. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition* 770–778 (IEEE, 2016).
59. Dosovitskiy, A. An image is worth 16×16 words: transformers for image recognition at scale. Preprint at <https://arxiv.org/abs/2010.11929> (2020).
60. Rodriques, S. G. et al. Slide-seq: a scalable technology for measuring genome-wide expression at high spatial resolution. *Science* **363**, 1463–1467 (2019).
61. Chen, A. et al. Spatiotemporal transcriptomic atlas of mouse organogenesis using DNA nanoball-patterned arrays. *Cell* **185**, 1777–1792 (2022).
62. Vickovic, S. et al. High-definition spatial transcriptomics for in situ tissue profiling. *Nat. Methods* **16**, 987–990 (2019).
63. Kim, J. et al. Unsupervised discovery of tissue architecture in multiplexed imaging. *Nat. Methods* **19**, 1653–1661 (2022).
64. Hu, Y. et al. Unsupervised and supervised discovery of tissue cellular neighborhoods from cell phenotypes. *Nat. Methods* **21**, 267–278 (2024).

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

© The Author(s), under exclusive licence to Springer Nature America, Inc. 2025

¹Ragon Institute of MGH, MIT, and Harvard, Cambridge, MA, USA. ²Broad Institute of MIT and Harvard, Cambridge, MA, USA. ³Massachusetts Institute of Technology, Cambridge, MA, USA. ⁴Department of Statistics and Data Science, The Wharton School, University of Pennsylvania, Philadelphia, PA, USA. ⁵Center for Virology and Vaccine Research, Beth Israel Deaconess Medical Center, Harvard Medical School, Boston, MA, USA. ⁶Department of Computer Science, City University of Hong Kong, Hong Kong, People's Republic of China. ⁷Dana-Farber Cancer Institute, Boston, MA, USA. ⁸Center of Hepato-Pancreato-Biliary Surgery, The First Affiliated Hospital of Sun Yat-sen University, Guangzhou, People's Republic of China. ⁹Division of Genetics and Genomics, Boston Children's Hospital, Harvard Medical School, Boston, MA, USA. ¹⁰Program in Biological and Biomedical Sciences, Harvard Medical School, Boston, MA, USA. ¹¹Division of Gastroenterology/Liver Center, Beth Israel Deaconess Medical Center, Harvard Medical School, Boston, MA, USA. ¹²Department of Pathology, Brigham and Women's Hospital, Harvard Medical School, Boston, MA, USA. ¹³Institute for Medical Engineering and Science, Massachusetts Institute of Technology, Cambridge, MA, USA. ¹⁴Department of Chemistry, Massachusetts Institute of Technology, Cambridge, MA, USA. ¹⁵Koch Institute for Integrative Cancer Research, Massachusetts Institute of Technology, Cambridge, MA, USA. ¹⁶Department of Pathology, Stanford University, Stanford, CA, USA. ¹⁷Department of Statistics and Data Science, Yale University, New Haven, CT, USA. ¹⁸These authors contributed equally: Bokai Zhu, Sheng Gao, Shuxiao Chen. ¹⁹These authors jointly supervised this work: Garry P. Nolan, Sizun Jiang, Zongming Ma. ✉e-mail: gnolan@stanford.edu; sjiang3@bidmc.harvard.edu; zongming.ma@yale.edu

Methods

CellLENS input preparation

CellLENS integrates single-cell feature expressions (for example, protein or mRNA), cell neighborhood context and local tissue image information for cell population differentiation and discovery. We first describe how to curate inputs to the CellLENS pipeline from a typical spatial omics dataset with single-cell or subcellular resolution.

Assume there are n cells in the dataset and we have the following information. First, we have a cell-by-feature matrix $X \in \mathbb{R}^{n \times d}$ recording d biomarkers (for example, protein, gene expression) for each cell. The i th row of X , $x_i \in \mathbb{R}^d$, records these biomarkers for the i th cell. In addition, we have the spatial locations of the cells. When all cells are within the same field of view (FOV), the spatial locations can be represented as a $n \times 2$ matrix storing the x and y coordinates of cell centroids within the FOV. When the dataset encompasses multiple FOVs, a FOV identifier is also recorded for each cell.

Spatial proximity and feature similarity graphs. We construct two graphs that share the same set of nodes; each node corresponds to a cell in the dataset. The first graph is a spatial proximity graph among cells. In this graph, each cell is connected to k_s spatial nearest neighbors within the same FOV, based on the spatial coordinates of cells. We denote this graph by G_s and its adjacency matrix A_s .

The second graph is a feature similarity graph among cells. For its construction, we first perform principal component analysis (PCA) on the feature matrix X to obtain a reduced-dimension representation $X^{PC} \in \mathbb{R}^{n \times d_0}$. We denote its i th row by x_i^{PC} for $i = 1, \dots, n$. Then, we calculate the pairwise similarity of cells using Pearson correlations (or other similarity measure that the user chooses) between row pairs of X^{PC} and form a k_f -nearest-neighbor graph by connecting each cell to k_f cells whose expression profiles are most similar. We denote this graph as G_f and its adjacency matrix by A_f .

Cellular neighborhood composition. We partition all cells into disjoint clusters by applying a clustering method on the feature similarity graph G_f . If human expert cell-type annotations are available, we can regard them as cluster labels and skip the clustering step. For all results reported in this study, we used Leiden clustering⁸ with a resolution level of 0.5 on G_f without human intervention for initial cell cluster label generation. These initial labels were in turn used for the cellular neighborhood composition calculations.

Suppose there are in total p different initial cell cluster labels. Fix a positive neighborhood size k_i . For the i th cell, its cellular neighborhood composition vector is $y_i \in \mathbb{R}^p$, which records the proportions of its k_i spatial nearest neighbors belonging to each of the p clusters. The spatial nearest neighbors of each cell are determined using the spatial coordinates of cells in the same FOV. We stack the y_i values as row vectors to form $Y \in \mathbb{R}^{n \times p}$.

The generated fixed dimensional cellular neighborhood composition vector for each cell will then be concatenated with each cell's one-hot cluster label vector. The stacked $2p$ -dimensional vectors $\{y_i^{tg} : i = 1, \dots, n\}$ will be used as the self-supervising target of prediction in the later LENS-GNN-duo model.

Local tissue image tensor. For the i th cell, we crop its corresponding FOV at each channel c of interest, centered at the centroid of the cell and with a window size of $L \times L$, resulting in a matrix $T_{0,i}^{(c)} \in \mathbb{R}^{L \times L}$. We then dichotomize the cropped image at each channel c at the α th quantile s_c of all values in $T_{0,i}^{(c)}$ to obtain an $L \times L$ matrix whose (a, b) th entry is:

$$(T_i^{(c)})_{ab} = \begin{cases} 1, & \text{if } (T_{0,i}^{(c)})_{ab} \geq s_c, \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

We concatenate the cropped and dichotomized images for all selected channels centered at each cell to generate the final output of this step: a 0–1-valued local tissue image tensor $T_i \in \mathbb{R}^{C \times L \times L}$ for each cell $i \in \{1, 2, \dots, n\}$, where C is the total number of image channels used. In this study, the α value was set to 0.9.

Default tuning parameter choices for input curation. By default, we set $d_0 = 25$ and $k_f = 15$ to construct the feature similarity graph, and $k_s = 15$ for the spatial proximity graph. For initial cell clustering, we apply Leiden clustering with a resolution of 0.5. In addition, we set $k_i = 20$ when computing cellular neighborhood composition vectors. For image processing, the default choices are $\alpha = 0.9$, $L = 512$ (corresponding to 50–100 μm) and $C = 2$ (corresponding to nucleus and membrane channels only to achieve robustness with respect to spatial omics technology).

LENS-CNN for encoding local tissue image

The first step of the CellLENS pipeline trains a CNN²⁴, called LENS-CNN, that predicts each cell's neighborhood composition vector y_i , with its local tissue image tensor T_i . After LENS-CNN is trained, we take the fitted hidden state of each cell as an encoding vector of the cell's local tissue image information.

LENS-CNN encoder architecture. The LENS-CNN encoder $\text{enc}(\cdot)$ maps each local tissue image tensor T_i to a code $z_i \in \mathbb{R}^q$, that is:

$$z_i = \text{enc}(T_i) \in \mathbb{R}^q \quad (2)$$

We adapt the architecture of AlexNet²⁶ to form a six-layer CNN as the encoder. The kernel sizes of the convolutional layers are 2, 3, 4, 2, 2 and 2, respectively, with the corresponding feature map dimensions 512, 128, 63, 45, 7, 3 and 2, and channel dimensions 2, 16, 32, 64, 128, 256 and 512. The architecture also includes a 2×2 max-pooling function after each of the first five convolution layers. Following the convolution layers, we flatten the feature map and add three fully connected layers with the output dimensions of 1,024, 512 and 128, respectively. The final output dimension of the LENS-CNN encoder is thus $q = 128$. To map z_i to a predicted neighborhood composition vector $\tilde{y}_i \in \mathbb{R}^p$, we add a fully connected layer and a nonlinear activation function on top.

In particular, we set:

$$\tilde{y}_i = \tilde{W}^T \phi(z_i) + \tilde{b} \quad (3)$$

where $\tilde{W} \in \mathbb{R}^{q \times p}$ and $\tilde{b} \in \mathbb{R}^p$ are learnable parameters and ϕ is the rectified linear unit (ReLU) function:

$$\phi(x) = \max(x, 0) \quad (4)$$

(see Extended Data Fig. 1 for a schematic plot of the LENS-CNN architecture).

After training, this step outputs fitted coding vectors $\{z_i \in \mathbb{R}^q : i = 1, \dots, n\}$ for use in the subsequent steps of CellLENS.

LENS-CNN training. The learnable parameters in $\text{enc}(\cdot)$ and in equation (3) are trained jointly by minimizing the mean squared error between the true and predicted neighborhood composition vectors. In other words, the loss function for training LENS-CNN is

$$L_2(Y, \tilde{Y}) = \frac{1}{n} \sum_{i=1}^n \|\tilde{y}_i - y_i\|_2^2 \quad (5)$$

In our implementation, LENS-CNN is trained by the Adam optimizer with a learning rate of 10^{-4} , weight decay of 0 and batch size of 64 for 400 epochs.

LENS-GNN-duo for learning cell population representation

In the second step of CellLENS, which we call LENS-GNN-duo, we train a pair of GNNs with an overarching MLP head²⁴ to predict each cell's neighborhood-composition-plus-cell-cluster vectors, using both its feature expressions and its local tissue image encoding. The nodes of the graphs underlying the two GNNs are shared and correspond to the individual cells in the dataset. The underlying edge structures among the nodes are different and are given by the feature similarity graph G_f and the spatial proximity graph G_s , respectively. For the i th node (that is, cell), its nodal covariates in G_f include feature expressions $x_i \in \mathbb{R}^d$; its nodal covariates in G_s include local tissue image encoding $z_i \in \mathbb{R}^q$.

By training these learnable parameters in the LENS-GNN-duo architecture, we aim to smooth feature expression and local tissue image information based on message passing on G_f and G_s , respectively, and to integrate them via the overarching MLP to predict the target vector for each cell. The cell state representation will be the concatenation of the outcomes of message passing on the pair of graphs after the model is properly trained.

LENS-GNN-duo architecture. For message passing on the feature similarity graph G_f , we first curate the initial nodal covariate at node i as

$$h_i^{(1),f} = \phi([W^{(0),f}]^T x_i + b^{(0),f}) \quad (6)$$

Here, $W^{(0),f} \in \mathbb{R}^{d \times r_1}$ and $b^{(0),f} \in \mathbb{R}^{r_1}$ are learnable parameters, and the ReLU activation function ϕ (defined in equation (4)) is applied elementwisely on the entries of the vector in parentheses.

The collection of the initial nodal covariate vectors $\{h_i^{(1),f} : i = 1, \dots, n\}$ is used as the input to a GNN with network structure G_f (that is, expression GNN in Fig. 1). To enable message passing across neighboring cells, we adopt the notion of graph convolution layer from ref. 23, which can be viewed as a localized first-order approximation to spectral graph convolution. We stack $\{h_i^{(1),f} : i = 1, \dots, n\}$ as the row vectors of $H^{(1),f} \in \mathbb{R}^{n \times r_1}$. We define $\tilde{A}_f = A_f + I_n$, where A_f is the adjacency matrix of G_f and I_n is the n -by- n identity matrix. The matrix $\tilde{A}_f$ is the adjacency matrix of the graph $\tilde{G}_f$ obtained from adding a self-loop to each node in G_f . We further define $\tilde{D}_f$ as the diagonal matrix with $(\tilde{D}_f)_{ii} = \sum_{j=1}^n (\tilde{A}_f)_{ij}$, for $i = 1, \dots, n$, which records the degree of each node in $\tilde{G}_f$. With the foregoing definitions, the message-passing convolution on G_f can be written as:

$$\begin{aligned} H^{(2),f} &= \tilde{D}_f^{-\frac{1}{2}} \tilde{A}_f \tilde{D}_f^{-\frac{1}{2}} H^{(1),f} W^{(1),f} \\ H^{(3),f} &= \tilde{D}_f^{-\frac{1}{2}} \tilde{A}_f \tilde{D}_f^{-\frac{1}{2}} \phi(H^{(2),f}) W^{(2),f} \end{aligned} \quad (7)$$

Here, $W^{(1),f} \in \mathbb{R}^{r_1 \times r_2}$ and $W^{(2),f} \in \mathbb{R}^{r_2 \times r_3}$ are learnable parameters. The ReLU activation function ϕ is applied elementwisely on $H^{(2),f}$. The row vectors of $H^{(2),f}$ and $H^{(3),f}$ are denoted by $\{h_i^{(2),f} : i = 1, \dots, n\} \subset \mathbb{R}^{r_2}$ and $\{h_i^{(3),f} : i = 1, \dots, n\} \subset \mathbb{R}^{r_3}$, respectively.

In an analogous way, we define $\tilde{G}_s$, $\tilde{A}_s$ and $\tilde{D}_s$ based on the spatial proximity graph G_s . We stack the output of LENS-CNN for each cell into $Z \in \mathbb{R}^{n \times q}$. The initial nodal covariate and message passing on the graph neural network with network structure G_s (that is, spatial GNN in Fig. 1) is defined as:

$$\begin{aligned} H^{(1),s} &= \phi(ZW^{(0),s} + 1_n(b^{(0),s})^T) \\ H^{(2),s} &= \tilde{D}_s^{-\frac{1}{2}} \tilde{A}_s \tilde{D}_s^{-\frac{1}{2}} H^{(1),s} W^{(1),s} \\ H^{(3),s} &= \tilde{D}_s^{-\frac{1}{2}} \tilde{A}_s \tilde{D}_s^{-\frac{1}{2}} \phi(H^{(2),s}) W^{(2),s} \end{aligned} \quad (8)$$

Here, 1_n is the all-one vector in $\mathbb{R}^n$; $b^{(0),s} \in \mathbb{R}^{r_1}$, $W^{(1),s} \in \mathbb{R}^{q \times r_1}$, $W^{(2),s} \in \mathbb{R}^{r_1 \times r_2}$ and $W^{(3),s} \in \mathbb{R}^{r_2 \times r_3}$ are all learnable parameters; and the ReLU function ϕ is applied elementwisely in both instances. The row vectors

of $H^{(2),s}$ and $H^{(3),s}$ are denoted by $\{h_i^{(2),s} : i = 1, \dots, n\} \subset \mathbb{R}^{r_2}$ and $\{h_i^{(3),s} : i = 1, \dots, n\} \subset \mathbb{R}^{r_3}$, respectively.

Given the outcomes of message passing on both graphs, we define for $i = 1, \dots, n$:

$$h_i^{(0)} = \left[(h_i^{(3),f})^T, (h_i^{(3),s})^T \right]^T \in \mathbb{R}^{p_0} \quad (9)$$

as the input to the MLP head for predicting the neighborhood-composition-plus-cell-cluster-annotation vector. Here $p_0 = r_3 + t_3$. We stack $\{h_i^{(0)} : i = 1, \dots, n\}$ as rows of $H^{(0)} \in \mathbb{R}^{n \times p_0}$. The next layers of the MLP are defined as

$$\begin{aligned} H^{(1)} &= \phi(H^{(0)})W^{(0)} + 1_n(b^{(0)})^T \\ H^{(2)} &= \phi(H^{(1)})W^{(1)} + 1_n(b^{(1)})^T \end{aligned} \quad (10)$$

Here $W^{(0)} \in \mathbb{R}^{p_0 \times p_1}$, $W^{(1)} \in \mathbb{R}^{p_1 \times 2p}$, $b^{(0)} \in \mathbb{R}^{p_1}$ and $b^{(1)} \in \mathbb{R}^{2p}$ are all learnable parameters, and ReLU activation ϕ is applied elementwisely. Finally, we define $\hat{y}_i^{tg} = h_i^{(2)}$ as the predicted neighborhood-composition-plus-cell-cluster-annotation vector for the i th cell, where $h_i^{(2)}$ is the i th row vector of $H^{(2)}$ and

$$u_i = h_i^{(0)} \quad (12)$$

as its representation. The representation vectors $\{u_i : i = 1, \dots, n\}$ are used in the downstream analysis.

LENS-GNN-duo training. Given the predicted and ground-truth target vectors, that is, $\{\hat{y}_i^{tg} : i = 1, \dots, n\}$ and $\{y_i^{tg} : i = 1, \dots, n\}$, we define the loss function of LENS-GNN-duo as

$$L_2(Y^{tg}, \hat{Y}^{tg}) = \frac{1}{n} \sum_{i=1}^n \|\hat{y}_i^{tg} - y_i^{tg}\|_2^2 \quad (13)$$

where $Y^{tg} \in \mathbb{R}^{n \times 2p}$ is the matrix of the concatenated neighborhood composition and cell cluster vectors, and $\hat{Y}^{tg} \in \mathbb{R}^{n \times 2p}$ is a matrix with the i th row being the predicted vector $\hat{y}_i^{tg}$.

By default, the hidden dimensions of LENS-GNN-duo are set at $r_1 = r_2 = t_1 = t_2 = 32$, $r_3 = 33$, $t_3 = 11$ and $p_1 = 33$. By definition $p_0 = r_3 + t_3 = 44$. LENS-GNN-duo is trained by the Adam optimizer, with a learning rate of 10^{-3} and weight decay of 0, and with a single batch for 3,000 epochs.

To improve the stability of the learned embedding vectors $\{u_i : i = 1, \dots, n\}$, we propose training LENS-GNN-duo for m rounds. In the l th round, we stack the learned embeddings as row vectors of $U_{[l]} \in \mathbb{R}^{n \times p_0}$. After obtaining $U_{[l]}$, $l = 1, \dots, m$, we construct the following:

$$\bar{U} = [U_{[1]} \dots U_{[m]}] \in \mathbb{R}^{n \times (mp_0)} \quad (14)$$

Then, we compute the p_0 leading left singular vectors of $\bar{U}$, collected as the columns of $\bar{L} \in \mathbb{R}^{n \times p_0}$, and the corresponding singular values, collected as the diagonal entries of the diagonal matrix $\bar{D} \in \mathbb{R}^{p_0 \times p_0}$. Our final learned embedding vectors are taken to be the row vectors of $\bar{L}\bar{D} \in \mathbb{R}^{n \times p_0}$, that is,

$$\begin{bmatrix} u_1^T \\ \vdots \\ u_n^T \end{bmatrix} = \bar{L}\bar{D} \quad (15)$$

(see Extended Data Fig. 1 for a schematic plot of the LENS-GNN-duo architecture and the representation generation).

Cell population identification on CellLENS representations

The CellLENS representation vector of the i th cell is $u_i \in \mathbb{R}^{p_0}$. It combines complementary information from multiple domains in spatial

omics data. The additional information provided by cellular neighborhood and local tissue images enables improved cell population differentiation and discovery at a finer granularity than that enabled by feature expression alone.

To this end, we construct a new graph G_c with cells as nodes. Each cell i is connected to its k_c -nearest neighbors measured using Euclidean distance in the representation vector space. In other words, cell j is connected to i in G_c , if u_j is among the k_c closest points to u_i in $\mathbb{R}^{p_0}$, or vice versa. Then, we perform Leiden clustering⁸ on G_c to find cell clusters, followed by cell-type annotation processes, similar to the procedure performed in conventional single-cell studies. For benchmarking purposes, all final clusters presented in this study used the same set of parameters for Leiden clustering (resolution = 1).

Recall that each cell has a coarse initial cell population label. In the last step, we refine the clustering result on CellLENS representation by singling out in each cluster any initial cell population that is primarily contained in the current cluster, provided that the current cluster also includes a substantial fraction of cells from other initial cell populations (by default, when the composition of initial cell populations in the current cluster, when viewed as a discrete probability distribution, leads to a Shannon entropy > 0.75). Further details are documented in function `cluster_refine` in the GitHub repository (<https://github.com/ssgao/celllens/>).

Benchmarking against other methods

For all methods benchmarked initially: (1) for ‘feature’, the single-cell expression profile first underwent PCA dimension reduction, and the first 30 principal components (or maximum feature numbers) were used as input; (2) for ‘concat’, the 30-component feature PCA was directly concatenated with the neighborhood composition vector and used as input; (3) for ‘CCA’, the function `cross_decomposition` from the Python package ‘sklearn’ were applied on the feature expression vector and the neighborhood composition vector; subsequently the CCA embeddings of the feature data were used as the CCA representation; (4) for ‘MOFA+’, the feature expression vector and the neighborhood composition vector were used as the input for the R package ‘MOFA2’ with default parameters; (5) for ‘SPICEMIX’, single-cell expression profiles along with cell spatial adjacency information were used as the input for SPICEMIX²⁰, and default parameters were implemented and trained with epochs = 200; (6) for ‘MUSE’, we first extracted images from each individual cell, using the same nuclear and membrane channels as implemented in CellLENS. The single-cell level whole-cell segmentation masks were retrieved directly from the original data source. These images, alongside the single-cell expression profile, were used as input for MUSE¹⁹; the subsequent steps used default parameters.

During revision, we further expanded the collection of methods to benchmark against, which includes: (7) for SpaGCN⁵³, the nuclear and membrane images (combined together in RGB format) along with the cell locations and expression profiles were used as the input and run with default parameters (epoch = 250). Because of memory limit, SpaGCN can only be run on datasets with fewer than approximately 50k cells; thus, for datasets exceeding this number, a random subregion of the tissue was cropped and used for benchmarking; (8) for stLearn⁵⁴, cell locations and expression profiles were used as the input and run with default parameters (PCA = 30 or maximum feature numbers); (9) for SEDR⁵⁵, cell locations and expression profiles were used as the input and run with default parameters (PCA = 30) or maximum feature numbers; (10) for BANKSY⁵⁶, cell locations and expression profiles were used as the input and run with default parameters (PCA = 30) or maximum feature numbers (lambda = 0.2; k_geom = 15); (11) for CellCharter, cell locations and expression profiles were used as the input and run with default parameters (PCA = 30) or maximum feature numbers; n_layers = 3). Finally, to calculate the quantitative metrics for clustering performance evaluation in each dataset across methods, five batches, each with 10,000 randomly selected cells from the data, were used.

Evaluation metrics that do not require ground-truth labels were used for benchmarking each embedding’s performance during clustering. We used four metrics: the silhouette score²⁷; the Davies–Bouldin index²⁸; the Calinski–Harabasz index²⁹; and the modularity score³⁰. For the silhouette score, Davies–Bouldin index and Calinski–Harabasz index, the metrics were calculated by first generating a subset of data containing 10,000 randomly selected cells; then, we used Leiden clustering to produce K numbers of clusters on the subset, where K spanned from 5 to 30. Lastly, we used the functions `silhouette_score`, `davies_bouldin_score` and `calinski_harabasz_score` from the Python package `sklearn.metrics` to generate the respective metrics with each k clustering result. This process was then repeated five times. For the modularity score, the metric was calculated by first generating a subset data containing 10,000 randomly selected cells, then we used function `find_partition` (from the Python package `leidenalg`) with `partition_type = RBConfigurationVertexPartition` and different resolution γ values to generate the graph partitions. The γ values ranged from 0.5 to 2.6. Then, the modularity scores (based on the partition at each individual resolution value) were extracted. Note that this clustering process does not optimize for the modularity score itself. Lastly, the modularity score was extracted from the `find_partition` results, and this process was then repeated five times.

Scalability testing

We created a collection of datasets with ranging cell numbers by subsetting and duplicating the Xenium tonsil dataset: datasets with a size of approximately 10k and 50k were generated by randomly cropping a region in the whole tissue; a dataset of size approximately 150k is the original version; datasets of size approximately 500k and 1,500k were generated by duplicating the original dataset three times and ten times, respectively. Subsequently, we implemented the CellLENS process on these datasets, using an NVIDIA A5000 graphics processing unit and a PCIe 4.0 solid-state drive disk. The runtime and memory usage during different CellLENS steps were recorded using the `wandb` package and application programming interface interface.

Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

Data availability

This study did not generate any new experimental data. The CODEX spleen dataset was generated from ref. 2 and the original data can be accessed at <https://data.mendeley.com/datasets/zjnpwh8m5b/1>. The Xenium tonsil dataset was generated from ref. 35, and the original data can be accessed at <https://www.10xgenomics.com/datasets/human-tonsil-data-xenium-human-multi-tissue-and-cancer-panel-1-standard>. The Xenium LN dataset was generated from ref. 38, and the original data can be accessed at <https://www.10xgenomics.com/datasets/preview-data-xenium-prime-gene-expression>. The CODEX tonsil dataset was generated from ref. 34, and the data can be found in our Zenodo link <https://doi.org/10.5281/zenodo.14617085>. The CODEX cHL dataset was generated from ref. 39, and the original data is available via Zenodo at <https://doi.org/10.5281/zenodo.10067010> (ref. 66). The CosMx liver dataset was generated from ref. 42, and the original data can be accessed at <https://nanosttring.com/products/cosmx-spatial-molecular-imager/ffpe-dataset/human-liver-rna-ffpe-dataset/>. For convenient access to this study’s data, the files are available via Zenodo at <https://doi.org/10.5281/zenodo.14617085> (ref. 65).

Code availability

The CellLENS Python package, along with the code used in this study, is available via GitHub at <https://github.com/ssgao/celllens/>.

References

65. Zhu, B. Cross-domain information fusion for enhanced cell population delineation in single-cell spatial-omics data. *Zenodo* <https://doi.org/10.5281/zenodo.14617085> (2025).
66. Shaban, M. et al. Data for MAPS: pathologist-level cell type annotation from tissue images through machine learning. *Zenodo* <https://doi.org/10.5281/zenodo.10067010> (2023).

Acknowledgements

We thank the insightful discussion with lab members from the G.P.N., A.K.S., S.J. and Z.M. labs. G.P.N. is supported in part by the Rachford and Carlota A. Harris Endowed Professorship. A.K.S. is supported in part by a National Institutes of Health (NIH) grant no. P01AI177687, the Bill & Melinda Gates Foundation (INV-055706), an NIH grant no. 75N93019C00071, the Koch Institute Frontier Research Program, the Virginia and D.K. Ludwig Fund for Cancer Research and the Koch Institute Support (core) grant no. P30-CA014051 from the National Cancer Institute. S.J. is supported in part by NIH grant nos. DP2AI171139, P01AI177687, R01GM152585, R01AI149672 and U24CA224331, a Gilead's Research Scholars Program in Hematologic Malignancies, a Sanofi Award, the Bill & Melinda Gates Foundation, the Dye Family Foundation and previously by the Leukemia Lymphoma Society Career Development Program. Z.M. is supported by National Science Foundation grant nos. 2345215 and 2245575. S.J. and Z.G.J. acknowledge support by a Beth Israel Deaconess Medical Center Spark grant. This article reflects the views of the authors and should not be construed as representing the views or policies of the institutions that provided funding.

Author contributions

S.G., Z.M., S.C. and B.Z. conceptualized the study. S.G., Z.M., B.Z. and Y.W. developed and implemented the algorithm. B.Z., J.Y., Y.B., A.Y.H., Y.Y.Y., G.L., S.M. and Y.W. carried out the analysis. Z.G.J., S.J.R., K.W., G.P.N., A.K.S., S.J. and Z.M. provided the resources. G.P.N., A.K.S., S.J. and Z.M. supervised the study. B.Z., S.G. and S.C. contributed equally and have the right to list their names first in their CVs.

Competing interests

S.J. is a cofounder of Elucidate Bio, has received speaking honoraria from Cell Signaling Technology and has received research support from Roche, Novartis and Sanofi unrelated to this work. G.P.N. received research grants from Pfizer, Vaxart, Celgene and Juno Therapeutics during the time of and unrelated to this work. G.P.N. is a cofounder of Akoya Biosciences and Ionpath; an inventor on patent US9909167; and a scientific advisory board member for Akoya Biosciences. A.K.S. reports compensation for consulting or scientific advisory board membership from Honeycomb Biotechnologies, Cellarity, Ochre Bio, Relation Therapeutics, IntrECate Biotherapeutics, Bio-Rad Laboratories, Fog Pharma, Passkey Therapeutics and Dahlia Biosciences unrelated to this work. S.J.R. receives research support from Bristol Myers Squibb and KITE/Gilead. S.J.R. is a member of the scientific advisory board of Immunitas Therapeutics. The other authors declare no competing interests.

Additional information

Extended data is available for this paper at <https://doi.org/10.1038/s41590-025-02163-1>.

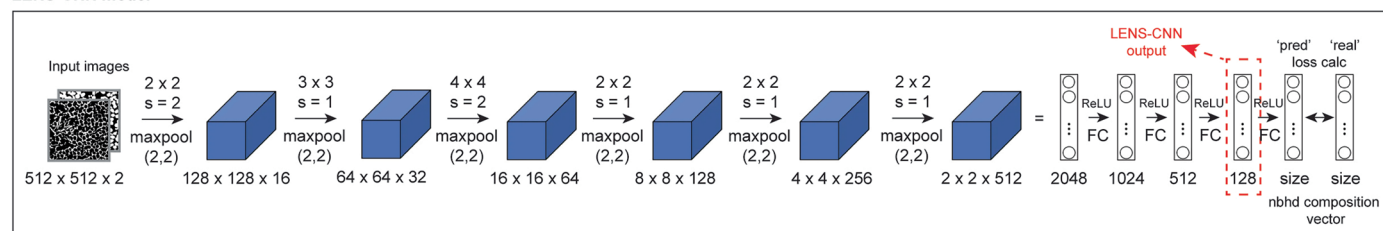
Supplementary information The online version contains supplementary material available at <https://doi.org/10.1038/s41590-025-02163-1>.

Correspondence and requests for materials should be addressed to Garry P. Nolan, Sizun Jiang or Zongming Ma.

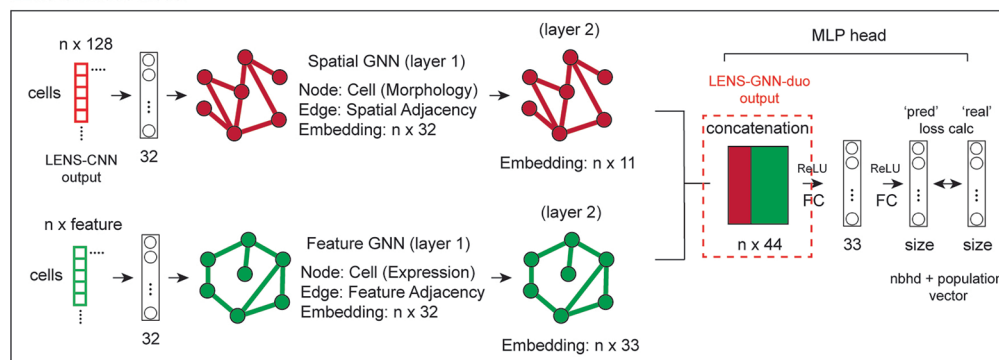
Peer review information *Nature Immunology* thanks Raphael Gottardo, Ahmed Mahfouz and the other, anonymous, reviewer(s) for their contribution to the peer review of this work. Peer reviewer reports are available. Primary Handling Editor: Nick Bernard, in collaboration with the *Nature Immunology* team.

Reprints and permissions information is available at www.nature.com/reprints.

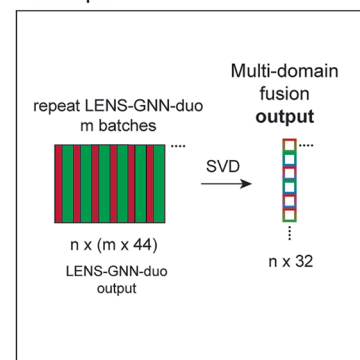
LENS-CNN model



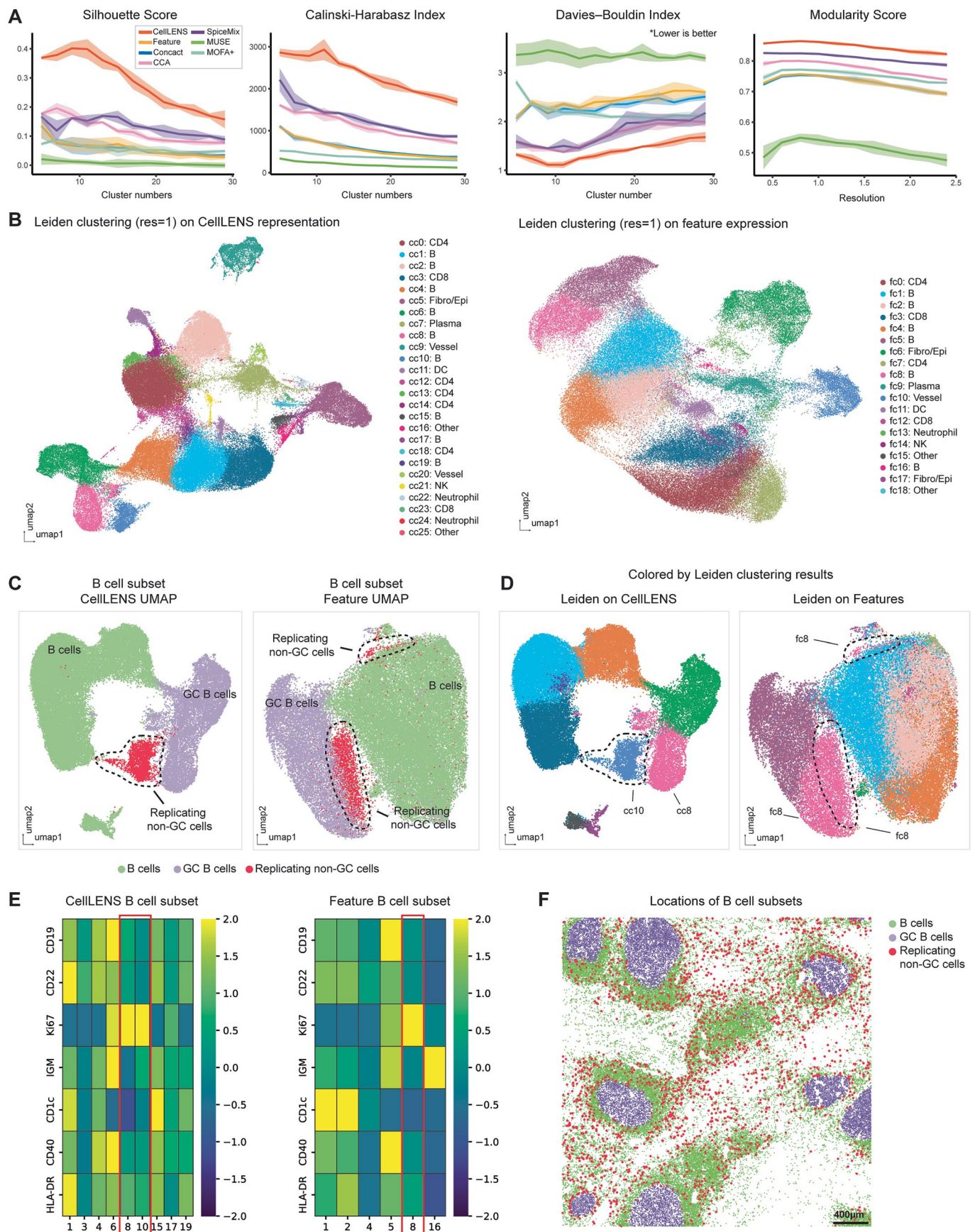
LENS-GNN-duo model



Final output



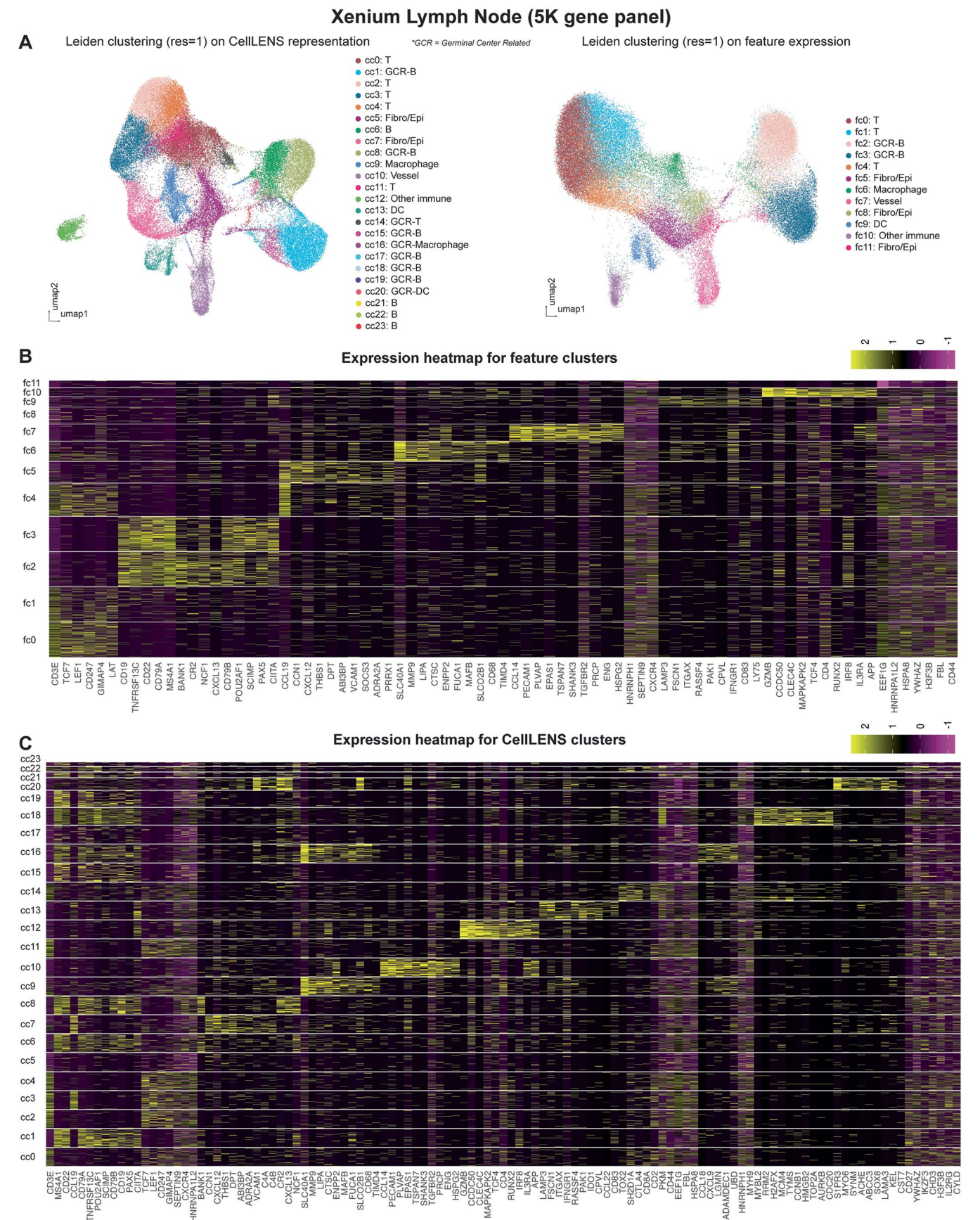
Extended Data Fig. 1 | Detailed illustration of LENS-CNN and LENS-GNN-duo model architectures in the CellLENS pipeline. The simplified versions are presented in Fig. 1. Additional details on CellLENS model architectures are described in the Methods section. Related code is deposited in the GitHub repository with documentation (see the Code Availability section for details).

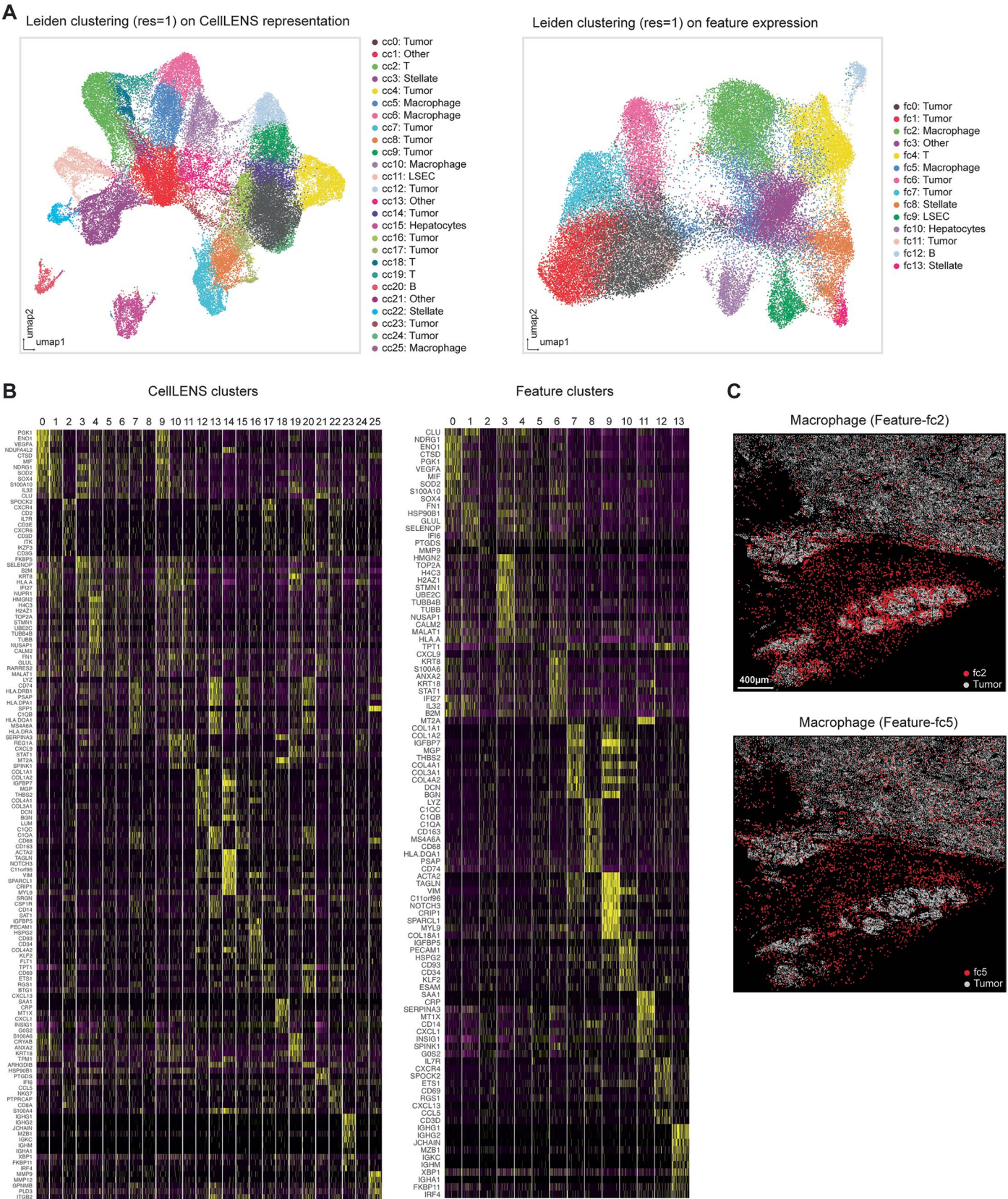


Extended Data Fig. 2 | See next page for caption.

Extended Data Fig. 2 | Refined B cell subpopulations discovered by CellLENS in a tonsil CODEX dataset. (A) Metrics evaluation of clustering performances on CODEX human tonsil tissue. Embeddings/representations of cells, from 7 different methods, were used as input: CellSNP representation, feature (protein expression table), concast (protein expression + neighborhood composition table), CCA representation, MOFA+ representation, SpiceMix representation, and MUSE representation. Clusters (for the calculation of Silhouette score, CH, and DB index) were generated with Leiden clustering using the same parameters. (B) UMAP visualization of the embedding and Leiden clustering result (cc: CellLENS clusters; fc: feature-only clusters). Left panel: CellLENS embedding; Right panel: feature expression table. (C) Zoom-in of the UMAP visualization of B cells (green), germinal center B cells (purple), and replicating non-GC cells

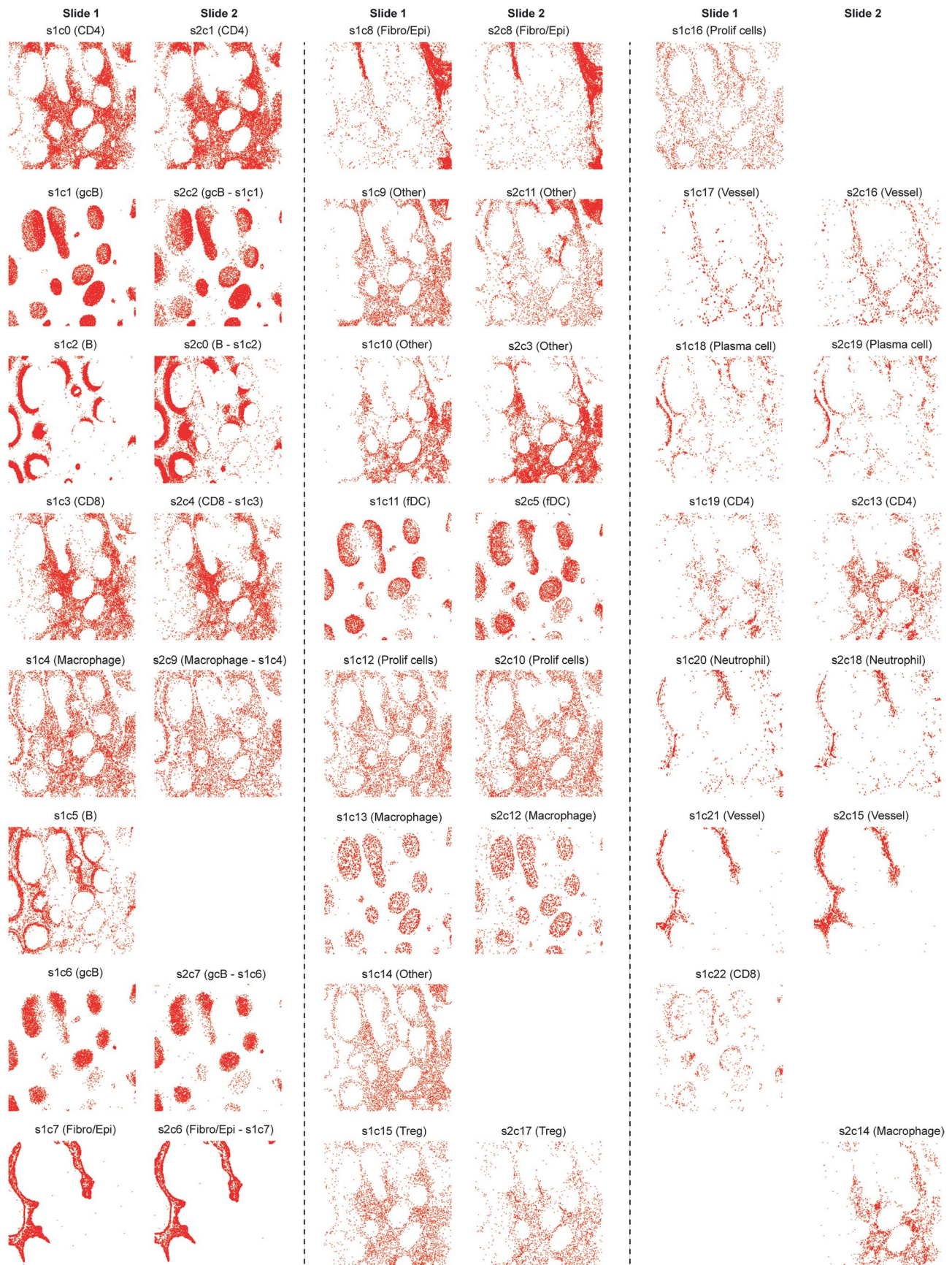
(red) on CellLENS embedding (left) and feature-only embedding (right). Cells colored according to cell populations identified using CellLENS clusters in both UMAPs. (D) Zoom-in of the UMAP visualization of the same cell types shown in C, but colored by cluster numbers from CellLENS embedding (left) and feature-only embedding (right). CellLENS successfully separated replicating non-GC cells from GC B cells (c10 and c8), where feature-only failed (c8 partial). (E) B cell and replication-related marker expression (z-normed) heatmap. Left panel: clusters from CellLENS; Right panel: clusters from feature-only. (F) Visualization of cell type locations: B cells (green), GC B cells (purple), and replicating non-GC cells (red). Based on the location, replicating non-GC cells should not be mixed with GC B cells, as shown in the clustering result from feature-only.





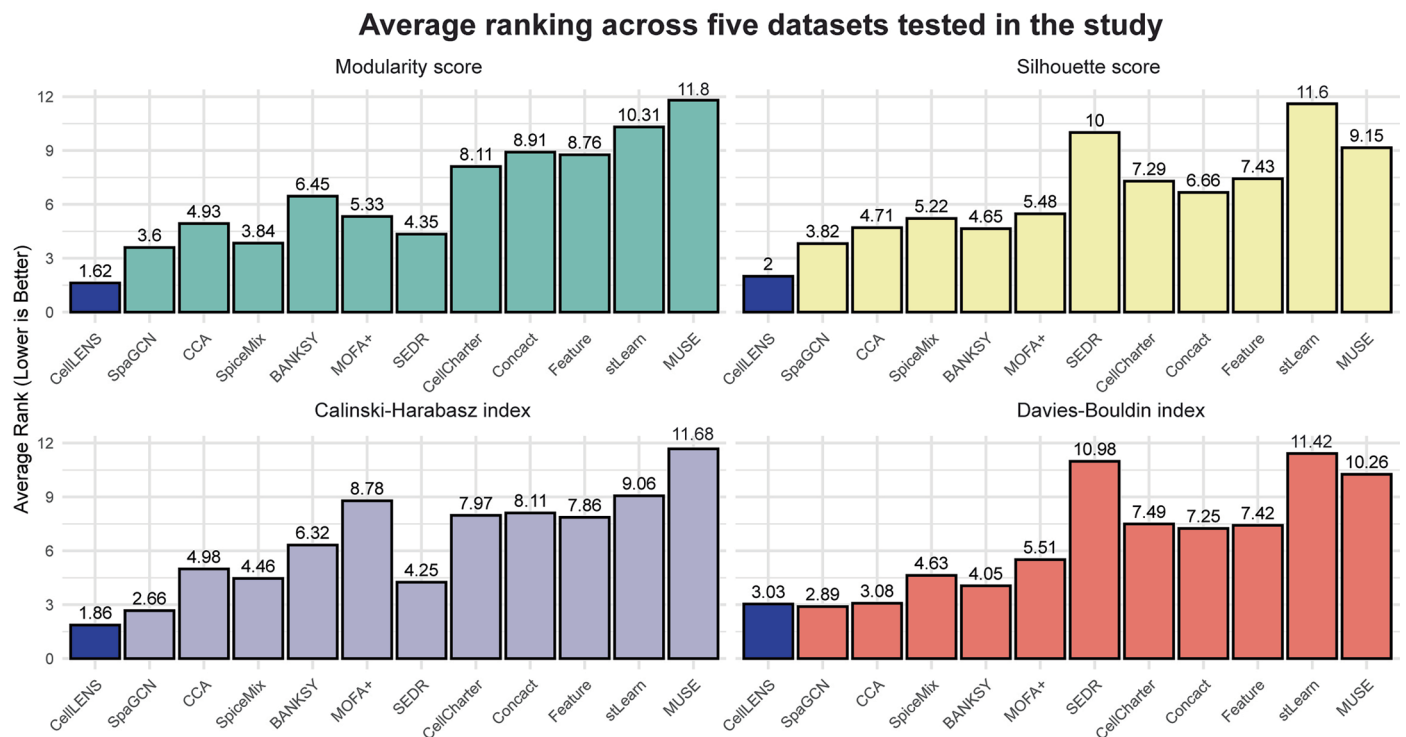
Extended Data Fig. 4 | Supplementary plots illustrating the application of CellLENS on the human HCC CosMx-SMI data in Fig. 5. (A) UMAP visualization of the embedding and Leiden clustering result (cc: CellLENS clusters; fc: feature-only clusters). Left panel: CellLENS embedding; Right panel: feature expression table. **(B)** Heatmaps of top 10 mRNA marker genes for each cluster. Genes were identified via function 'FindAllMarker' in R package 'Seurat'. Left panel:

clusters from CellLENS representation; Right panel: clusters from feature-only representation. **(C)** Visualization of spatial locations in the HCC tissue of different Macrophage subpopulations identified by feature-only representation. In each plot, cells in a target cluster are colored in red, HCC tumor cells are colored in grey, and other cells and empty spaces are colored in black.



Extended Data Fig. 5 | Evaluation of CellLENS performance stability on adjacent CODEX tonsil sections: correspondence of CellLENS clusters identified individually on two sections. CODEX tonsil tissue from two adjacent sections⁵² were used to evaluate the robustness of CellLENS performance. Shown

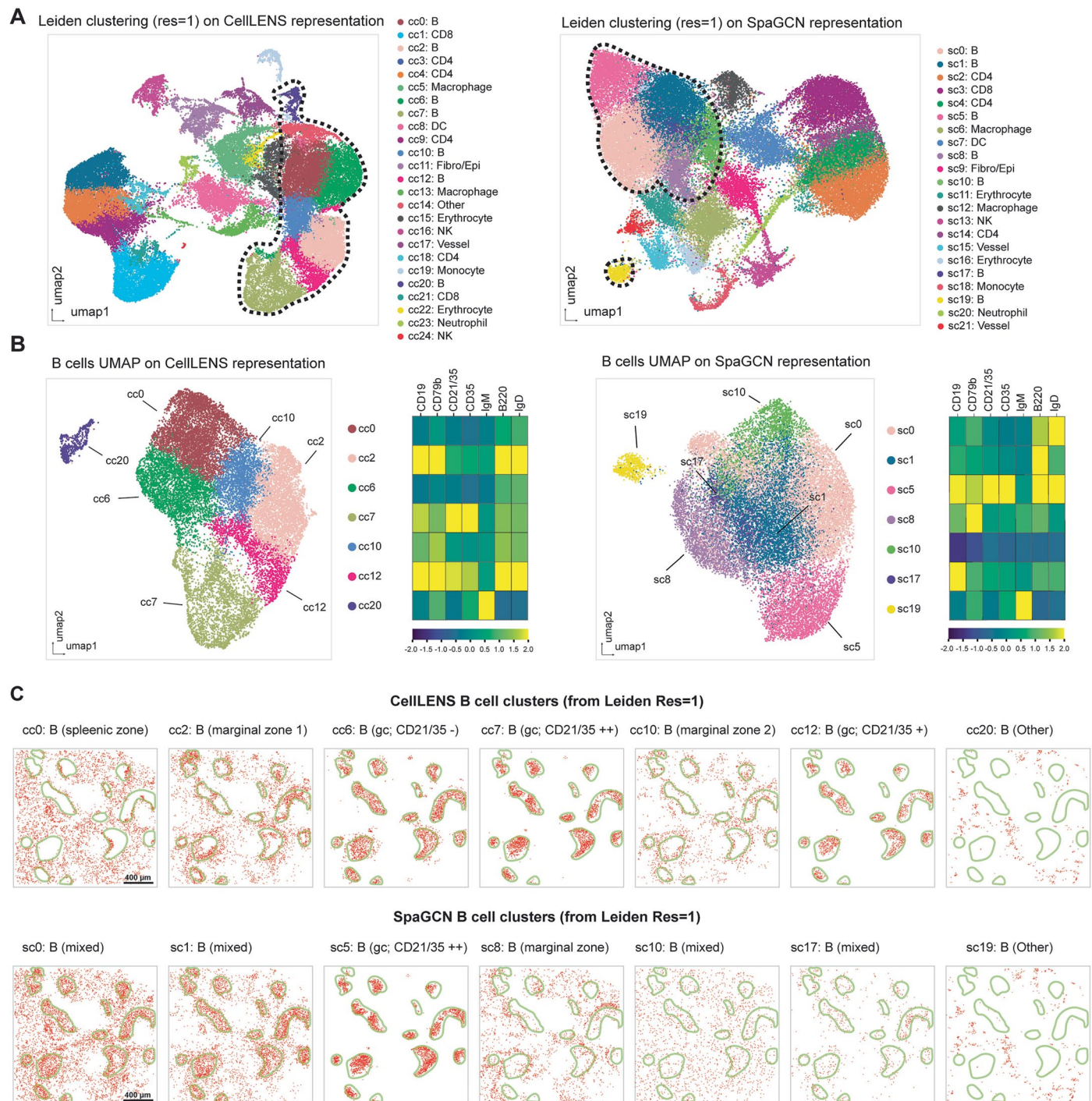
here are Leiden clustering results (resolution = 1) based on CellLENS embeddings. In each of the three columns, left are clusters identified from CellLENS embedding of slide 1, and right are corresponding clusters in slide 2 aligned to their counterparts in slide 1.



Extended Data Fig. 6 | Comprehensive benchmarking with additional methods. A total of 12 methods are benchmarked here: CellLENS (our method). SpaGCN: Designed for spot-level spatial modalities (to identify spatial domains). StLearn: Designed for spatial modalities (to identify cell populations). SEDR: Designed for spatial modalities (to identify cell populations). MUSE: Designed for single-cell spatial modalities (to identify cell populations). SpiceMix: Designed for spatial modalities (to identify cell populations). BANKSY: Designed for spatial modalities (to identify cell populations or spatial domains). CellCharter: Designed for spatial modalities (to identify spatial domains). MOFA+: Designed for general modalities. CCA: General statistical procedure with canonical correlation analysis. Concatenation: Direct concatenation

between feature and location matrix. Feature-only profile: Conventional way of cell type identification. We applied all 12 methods to five datasets presented in our manuscript (CODEX spleen, Xenium tonsil, CODEX tonsil, CODEX cHL, and CosMx HCC) and evaluated them using four different metrics (See Methods for details). Here we aggregated all the results across metrics and datasets into one summary figure. Each subpanel represents a specific metric (for example, Modularity score). The Y-axis indicates the average ranking of a method across benchmarking conditions (for example, K clusters or resolution numbers). On the X-axis, methods are arranged by their average ranking across all four metrics, such that methods on the left perform the best overall. In the summary figure, the rankings were averaged across all five datasets.

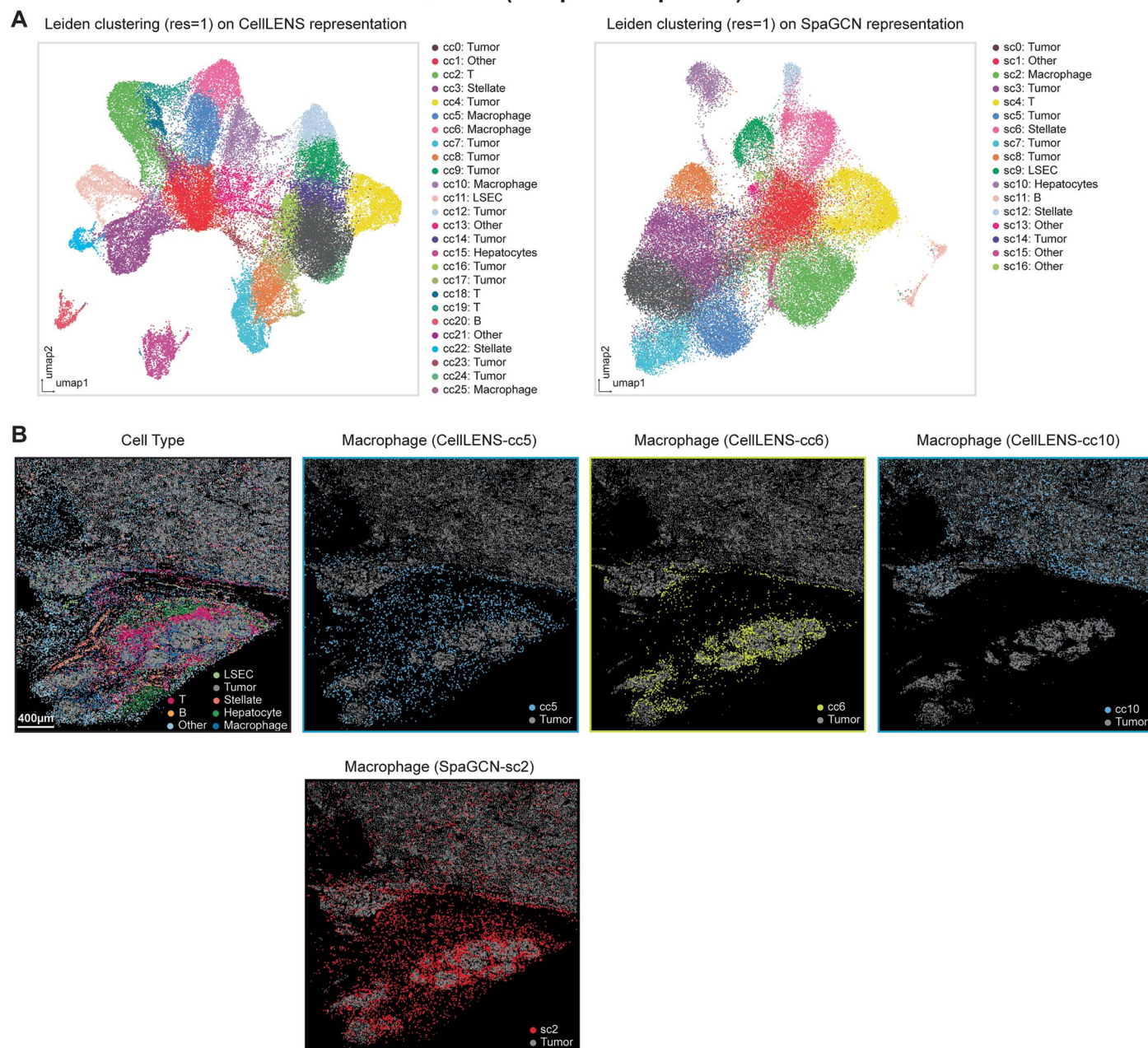
CODEX mouse spleen (compare to SpaGCN)



Extended Data Fig. 7 | Comparison between embeddings generated by CellLENS and SpaGCN on CODEX mouse spleen data. While initially designed for spot-level spatial-omics data, SpaGCN could run on the whole CODEX mouse spleen data (~50k cells). We compared the cell type delineation ability between

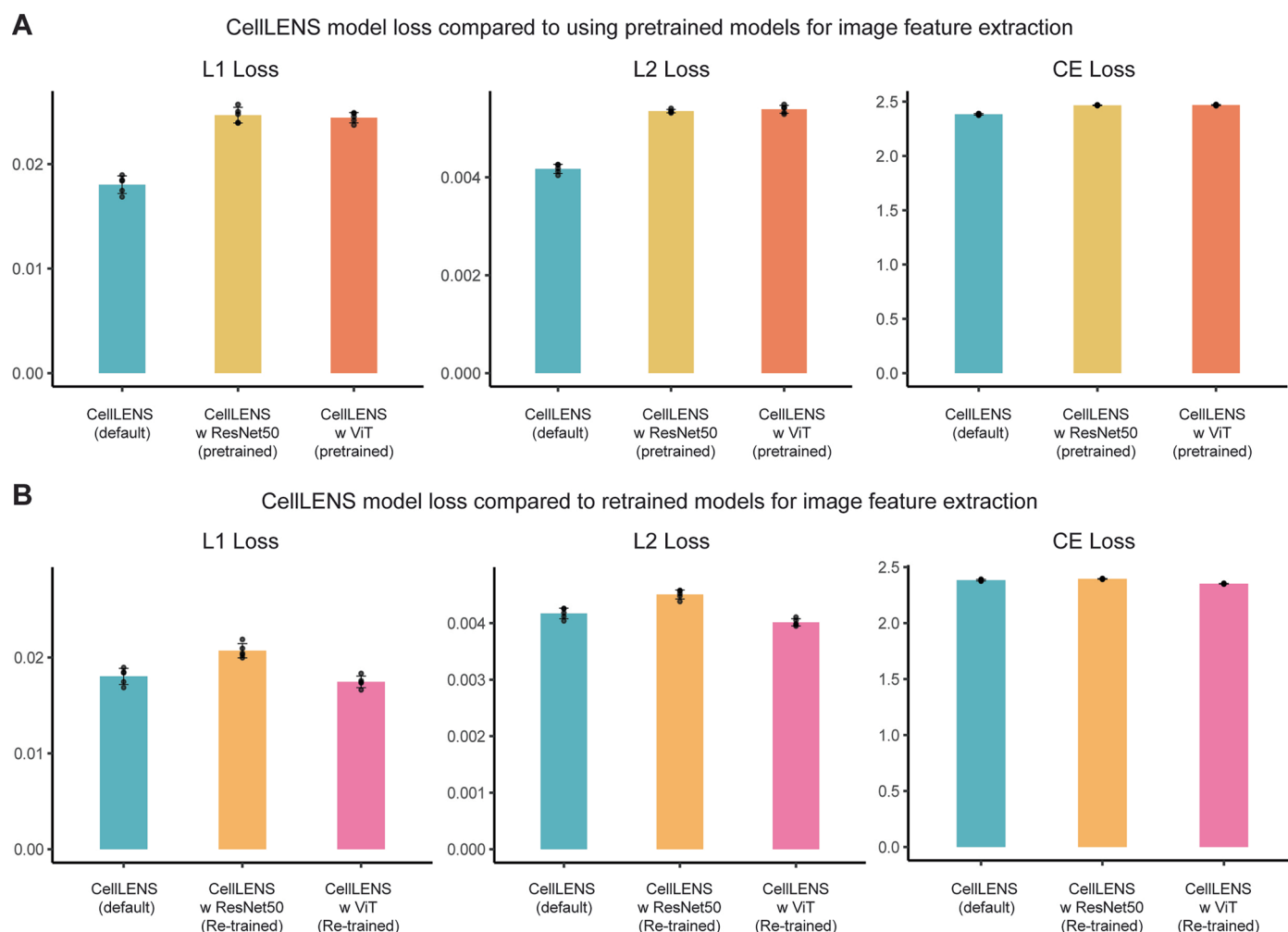
the embeddings from CellLENS and SpaGCN. The figure setup is the same as described in Fig. 2. SpaGCN failed to identify the various B cell subpopulations captured by CellLENS.

CosMx HCC (compare to SpaGCN)



Extended Data Fig. 8 | Comparison between embeddings generated by CelliLENS and SpaGCN on CosMx HCC data. While initially designed for spot-level spatial-omics data, SpaGCN could run on the whole CosMx-SMI human HCC data (~50k cells). We compared the cell type delineation ability

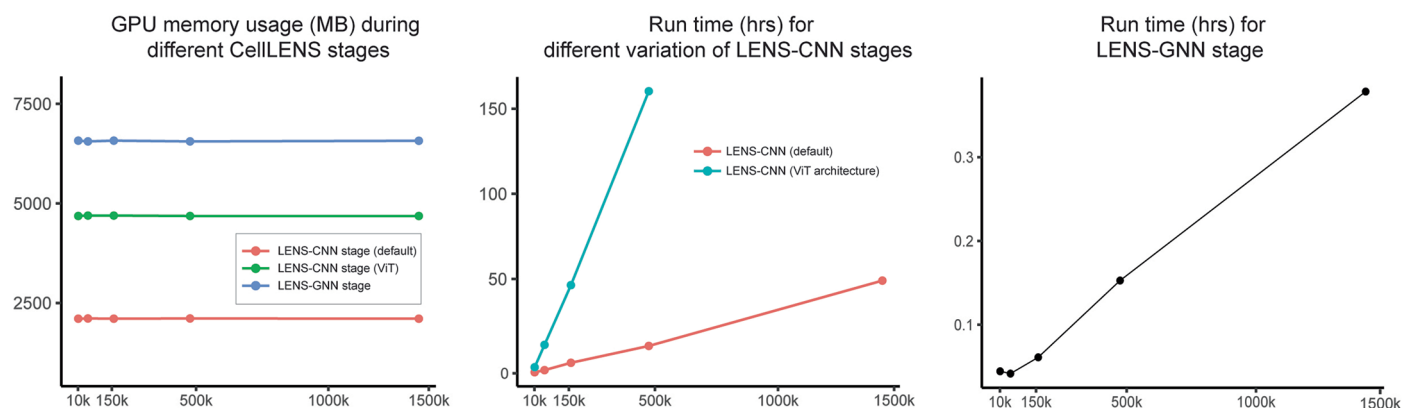
between the embeddings from CelliLENS and SpaGCN. The figure setup is the same as described in Fig. 5. SpaGCN failed to identify the various Macrophage subpopulations captured by CelliLENS.



Extended Data Fig. 9 | Loss quantification of the CellLENS model when using different image feature extraction processes. (A) We compared the CellLENS model training losses, across three CellLENS variations: 1) Default CellLENS, where the imaging feature extraction part is done by training an Alex-Net like CNN encoder (supervised by local cell type neighborhood composition vector). 2) CellLENS with a pre-trained ResNet50, where the image features were directly extracted with the pre-trained ResNet 50 model, flattened, and reduced to a vector with 128 dimensions. This vector is swapped with the original image feature vector obtained from the retrained Alex-Net model, and the rest of the CellLENS training process remains the same; 3) CellLENS with a pre-trained ViT

(transformer), using a similar process as the pre-trained ResNet50 in (2). (B) We compared the losses from three CellLENS variations: 1) CellLENS default with Alex-Net as described above. 2) CellLENS but swapping out the Alex-Net architecture with a ResNet50 architecture, and retraining its weights (initialized at pretrained weights). 3) CellLENS but swapping out the Alex-Net architecture with a ViT architecture, and retraining its weights (initialized at pretrained weight). The model loss was calculated the same as described in the Methods section paragraph 'Information retrieval efficacy evaluation of the LENS-GNN duo module'. In these cases, we implemented a 80/20 train test data split. Retraining was only done on the train data, and loss values were calculated on test data.

CellLENS run time and memory usage benchmark (with NVIDIA A5000 and SSD disk)



Extended Data Fig. 10 | Memory and run-time benchmarking for training CellLENS on datasets of different sizes. We ran CellLENS on increasing numbers of cells: -10k, -50k, -150k, -500k, -1.5 mil, and recorded the memory and run-time statistics. The benchmarking was performed on an NVIDIA A5000 GPU with an SSD disk (PCIe 4.0) for I/O. Left panel: GPU memory usage during CNN and GNN

training stages of CellLENS, across variable cell numbers. Middle panel: Run-times of CellLENS CNN training stage. Recorded run-time for both default LENS-CNN or LENS-CNN with a ViT architecture. Right panel: Run-time of CellLENS GNN training stage.

Reporting Summary

Nature Portfolio wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Portfolio policies, see our [Editorial Policies](#) and the [Editorial Policy Checklist](#).

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a Confirmed

- | | | |
|-------------------------------------|-------------------------------------|--|
| ☐ | ☒ | The exact sample size (n) for each experimental group/condition, given as a discrete number and unit of measurement |
| ☐ | ☒ | A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly |
| ☐ | ☒ | The statistical test(s) used AND whether they are one- or two-sided
<i>Only common tests should be described solely by name; describe more complex techniques in the Methods section.</i> |
| ☒ | ☐ | A description of all covariates tested |
| ☐ | ☒ | A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons |
| ☐ | ☒ | A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals) |
| ☐ | ☒ | For null hypothesis testing, the test statistic (e.g. F , t , r) with confidence intervals, effect sizes, degrees of freedom and P value noted
<i>Give P values as exact values whenever suitable.</i> |
| ☒ | ☐ | For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings |
| ☒ | ☐ | For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes |
| ☒ | ☐ | Estimates of effect sizes (e.g. Cohen's d , Pearson's r), indicating how they were calculated |

Our web collection on [statistics for biologists](#) contains articles on many of the points above.

Software and code

Policy information about [availability of computer code](#)

Data collection No code or software were used for data collection: all data were generated from separate studies, and directly used for analysis in our manuscript.

Data analysis All analysis code used in this manuscript is available online as a GitHub repository: <https://github.com/sggao/celllens/>

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Portfolio [guidelines for submitting code & software](#) for further information.

Data

Policy information about [availability of data](#)

All manuscripts must include a [data availability statement](#). This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A description of any restrictions on data availability
- For clinical datasets or third party data, please ensure that the statement adheres to our [policy](#)

This study did not generate any new experimental data:

CODEX spleen dataset was generated from Goltsev et al. 2018;

Xenium tonsil dataset was generated from <https://www.10xgenomics.com/datasets/human-tonsil-data-xenium-human-multi-tissue-and-cancer-panel-1-standard>;

Xenium lymph node dataset was generated from <https://www.10xgenomics.com/datasets/preview-data-xenium-prime-gene-expression>;

CODEX tonsil dataset was generated from Kennedy et al. 2021;

CODEX cHL dataset was generated from Shaban et al. 2024;

CosMx liver dataset was generated from <https://nanostring.com/products/cosmx-spatial-molecular-imager/ffpe-dataset/human-liver-rna-ffpe-dataset/>;

For more convenient access to this study's data, we have deposited the files under Zenodo: <https://zenodo.org/records/14617085> for free public access.

Research involving human participants, their data, or biological material

Policy information about studies with [human participants or human data](#). See also policy information about [sex, gender \(identity/presentation\), and sexual orientation](#) and [race, ethnicity and racism](#).

Reporting on sex and gender	This information is not available and not required in our manuscript, as such information is not related to any analysis/investigation performed in the study.
Reporting on race, ethnicity, or other socially relevant groupings	This information is not available and not required in our manuscript, as such information is not related to any analysis/investigation performed in the study.
Population characteristics	This information is not available and not required in our manuscript, as such information is not related to any analysis/investigation performed in the study.
Recruitment	This information is not available and not required in our manuscript, as such information is not related to any analysis/investigation performed in the study.
Ethics oversight	This information is not available and not required in our manuscript, as such information is not related to any analysis/investigation performed in the study.

Note that full information on the approval of the study protocol must also be provided in the manuscript.

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☒ Life sciences ☐ Behavioural & social sciences ☐ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see [nature.com/documents/nr-reporting-summary-flat.pdf](https://www.nature.com/documents/nr-reporting-summary-flat.pdf)

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size	All data utilized in this manuscript are generated from other published studies, thus in the context of study design, calculation of sample size was not performed in our study, as the sample size requirement is not relevant in this context. Tissue wise, all datasets used in our manuscript is consisted of one tissue. Cell number wise, the minimal cell size is ~ 50k cells, and the largest cell size is ~150k cells.
Data exclusions	All data utilized in this manuscript are generated from other published studies, thus in the context of study design, calculation of sample size was not performed in our study. All the analysis and comparisons were performed within tissue samples, and no exclusion was performed for data points within the investigated tissue regions.
Replication	No replication was performed in the context of life science study design, as no experimental data were produced in our study. However, replication was performed in the context of computational comparisons between methods by randomly selecting 10,000 cells five times from the same dataset. All attempts at replication were successful.
Randomization	All data utilized in this manuscript were generated from other published studies. Thus, in the context of study design, no randomization was performed. Additionally, since only one tissue per dataset was used in our study, randomization could not be performed at the sample level and was therefore irrelevant.
Blinding	All data utilized in this manuscript are generated from other published studies, thus in the context of study design, no blinding was performed. In the context of computational analysis, all the comparisons were performed within samples, thus blinding was not available and not required.

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

n/a	Involved in the study
☒	☐ Antibodies
☒	☐ Eukaryotic cell lines
☒	☐ Palaeontology and archaeology
☒	☐ Animals and other organisms
☒	☐ Clinical data
☒	☐ Dual use research of concern
☒	☐ Plants

Methods

n/a	Involved in the study
☒	☐ ChIP-seq
☒	☐ Flow cytometry
☒	☐ MRI-based neuroimaging

Plants

Seed stocks

Not involved in this study.

Novel plant genotypes

Not involved in this study.

Authentication

Not involved in this study.