Tissue Reassembly with Generative AI

Anonymous Authors[1]

Abstract

The spatial arrangement of cells plays a critical role in determining their functions and interactions within tissues. However, single-cell RNA sequencing dissociates cells from their native tissue context, resulting in a loss of spatial information. Here, we show that complex tissue structures can be reassembled from the gene expression profiles of dissociated cells. To achieve this, we developed LUNA, a generative AI model that reconstructs tissues conditioned solely on the gene expression of cells by learning spatial priors over existing spatially resolved datasets. We show that LUNA effectively reconstructs slices from the MERFISH whole mouse brain atlas with over 1.2 million cells. Applying LUNA to the mouse central nervous system scRNA-seq atlas, we show that LUNA is applicable for de novo generation of tissue structures. We envision that AI-driven tissue reassembly can help to overcome current technological limitations and advance our understanding of tissue organization and function.

1. Introduction

Single-cell RNA sequencing (scRNA-seq) technologies have enabled the high-throughput profiling of cellular transcriptomes, making it possible to identify and explore unique transcriptional profiles within individual cells (Tang et al., 2009; Macosko et al., 2015). These technologies have revolutionized our understanding of cellular diversity, providing unprecedented insights into the complexity of cells and tissues (Patel et al., 2014; Consortium* et al., 2022; Lu et al., 2023; Li et al., 2022; Sikkema et al., 2023; Hickey et al., 2023). However, a major limitation of scRNA-seq technologies is the loss of spatial context, as cells are dissociated from their native microenvironments during the sequencing process. The spatial context is crucial for understanding how cells interact with their neighbors and the surrounding microenvironment, influencing their roles in healthy tissue function and disease development. Although spatially resolved sequencing technologies aim to address this limitation (Chen et al., 2015; Codeluppi et al., 2018; Wang et al., 2018), they are restricted in the number of genes that can be measured.

To address the limitations of individual technologies, computational methods have been developed to infer spatial context from gene expression of single-cells. Early approaches relied on in situ hybridization (ISH)-based reference atlases of landmark gene expression and leveraged them to infer the spatial location of cells in scRNA-seq datasets (Achim et al., 2015; Satija et al., 2015). More recently, reference mapping methods such as Tangram (Biancalani et al., 2021), CytoSPACE (Vahid et al., 2023), and CeLEry (Zhang et al., 2023b), proposed mapping dissociated tissue data to reference spatial datasets. Although these methods have shown promising results, they rely on an exact spatial reference for mapping scRNA-seq datasets, thus primarily functioning as data integration tools. On the other hand, novoSpaRc (Nitzan et al., 2019; Moriel et al., 2021) enabled a *de novo* spatial reconstruction of single-cell gene expression. However, novoSpaRc assumes that cells in close physical proximity exhibit similar gene expression profiles — an assumption that is often violated in complex tissue architectures.

Here, we present LUNA (Location reconstrUction using geNerative Ai), a generative AI model that reassembles complex tissue structures from gene expression of cells by learning spatial priors over spatial transcriptomics datasets (Figure 1). LUNA learns cell representations that capture cellular interactions globally and locally across the entire tissue slice, enabled by an attention mechanism (Vaswani et al., 2017; Shen et al., 2021b) that takes into account interactions across all cells. LUNA operates as a diffusion model (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song & Ermon, 2019) – during training it learns to denoise corrupted cell coordinates, while during inference it starts from random noise and reconstructs the physical locations of cells *de novo* solely from their gene expression.

We apply LUNA to reassemble the MERFISH whole mouse brain atlas, which consists of 1.23 million cells spanning all brain regions (Zhang et al., 2023a), as well as the scRNA-

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

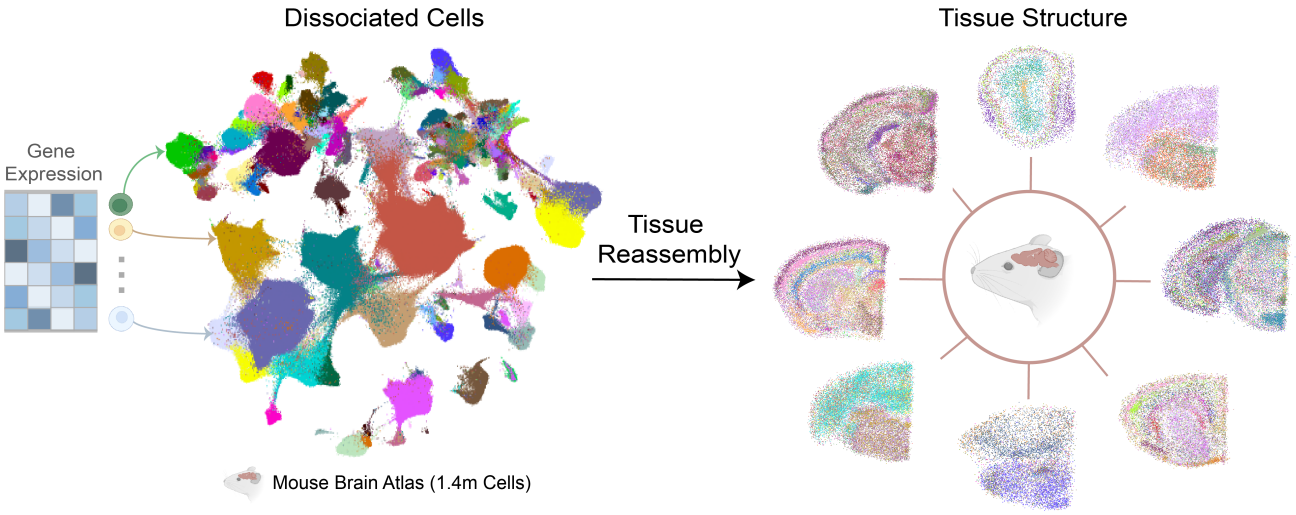

Figure 1. Given gene expressions of cells, the tissue reassembly process involves predicting the complex tissue architecture. We visualize the UMAP projection of cells based on their gene expression profiles (1, 122 genes) from the MERFISH whole mouse Allen Brain Cell (ABC) Atlas (Zhang et al., 2023a) (left). Each cell is colored according to one of 338 distinct subclasses. LUNA's predictions of the spatial tissue structure is displayed on the right with eight slices randomly chosen from a total of 66 slices.

seq mouse central nervous system atlas of 1.08 million cells (Zeisel et al., 2018). LUNA effectively reconstructs complex mouse brain tissue across a wide range of brain regions and captures spatial gene expression patterns. Furthermore, LUNA outperforms existing methods, achieving an 100% improvement on MERFISH mouse cortex atlas. We apply LUNA to simulate the knockout of individual genes *in silico*, allowing us to investigate their impact on the spatial organization of cells within tissues. LUNA is highly scalable, with linear time and memory complexity relative to the number of input cells, and it can infer the spatial coordinates of tens of thousands of cells in less than a few minutes on a single GPU.

2. Method

LUNA is a diffusion-based generative model designed to generate spatial locations of cells from their gene expression. Using an attention-based mechanism, LUNA learns how each individual cell should attend to other cells according to their relevance to a cell's own location and molecular features as shown in Figure 2. During training, LUNA leverages spatial transcriptomics data with ground truth cell locations to learn to generate cell locations conditioned on gene expression. During inference, the input to LUNA is the gene expression of dissociated cells, and the model then generates cell locations from pure noise conditioned on gene expression, enabling *de novo* inference of cell locations.

In the training phase, the input to LUNA is spatial transcriptomics data from single or multiple slices. LUNA learns cell embeddings that capture both the local and global tissue structure using a multi-head self-attention mechanism (Vaswani et al., 2017; Shen et al., 2021b) that enables cells to learn to attend to other cells. We designed the loss in LUNA as an SE(2)-invariant function (Segol & Lipman, 2020; Torres-Mendez et al., 2000; Romero et al., 2020; Wu et al., 2016), *i.e.*, it is robust to arbitrary rotations, translations and reflections of the predicted locations, only enforcing the preservation of relative distances between cells within each slice.

Training LUNA involves two main steps: *(i)* a corruption process that iteratively adds noise to the ground truth cell locations, and *(ii)* a denoising process that generates the ground truth locations from the noise-corrupted locations. During the denoising process, LUNA learns to reverse the corruption process, gradually transitioning from noisier to less noisy cell locations, ultimately reconstructing true cell locations from pure noise.

2.1. Learning cell representations

LUNA learns cell representations that capture the relationship between the molecular features of individual cells and their spatial context within tissue. These embeddings form the foundation for predicting the locations of cells.

The model is built on the premise that a cell's location is influenced not only by its own molecular features but also by those of other cells in the tissue. To effectively model these interactions, LUNA employs a multi-head self-attention mechanism (Vaswani et al., 2017; Shen et al., 2021b), which

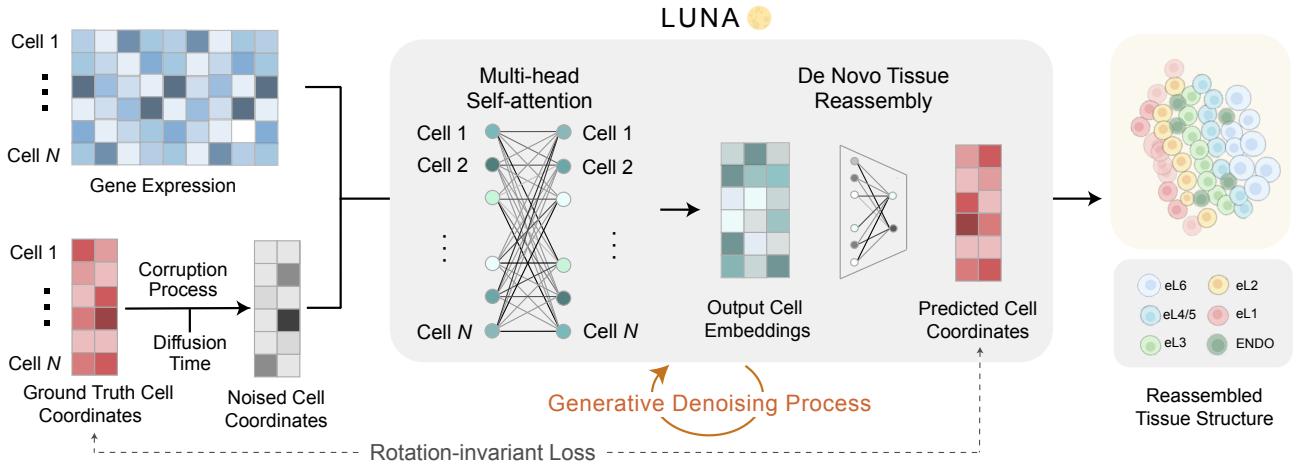

Figure 2. Overview of the training stage of LUNA. LUNA takes as input spatial transcriptomics data and corrupts the spatial locations of cells by adding noise. Using the attention mechanism, LUNA learns how to position each individual cell with respect to other cells based on their gene expressions. The loss function in LUNA ensures invariance to the rotations and reflections of the predicted slice.

dynamically integrates information from all cells in the slice. This mechanism allows the model to assign attention scores to different cells based on their relevance to the prediction of each cell's location, capturing both local and global cellular interactions. Once the cell embeddings are learned, LUNA projects each cell's embedding into a 2-dimensional space to predict its locations.

Formally, LUNA takes as input *(i)* molecular features of cells $\mathbf{X}_s \in \mathbb{R}^{m \times d}$ in a given slice s where m denotes the number of cells and d denotes the number of genes, *(ii)* time step $t \in \mathbb{N}$ uniformly sampled from 0 to the maximum diffusion time T, and *(iii)* corrupted cell locations $\tilde{\mathbf{R}}_{s,t} \in \mathbb{R}^{m \times 2}$ at time step t. The output of the model are the denoised cell locations $\hat{\mathbf{R}}_s \in \mathbb{R}^{m \times 2}$ for a given slice s. We omit slice index s for the ease of notation.

LUNA is composed of multiple transformer layers, each integrating fully connected layers followed by the self-attention block. At each layer l given a fixed diffusion time t, LUNA learns *(i)* cell embeddings $\mathbf{F}^{(l)} \in \mathbb{R}^{m \times d_x^{(l)}}$, *(ii)* a diffusion time embedding $\gamma_t^{(l)} \in \mathbb{R}$, and *(iii)* cell location embeddings $\mathbf{R}^{(l)} \in \mathbb{R}^{m \times d_r^{(l)}}$. The cell embeddings $\mathbf{F}^{(0)}$ in layer 0 are initialized via fully connected neural network layer that maps the molecular features of cells into the latent embedding space with dimensionality $d_x^{(0)}$:

$$\mathbf{F}^{(0)} = FCN(\mathbf{X}_s), \tag{1}$$

where *FCN* denotes a fully connected neural network layer. The diffusion time embedding $\gamma_t^{(0)}$ is initialized to the value of the diffusion time itself:

$$\gamma_t^{(0)} = t. \tag{2}$$

Finally, the cell location embeddings are initialized as the corrupted cell locations obtained from the corruption process at time step t:

$$\mathbf{R}^{(0)} = \tilde{\mathbf{R}}_t. \tag{3}$$

Then, at each layer $l \in \{1, \ldots, L\}$, LUNA first transforms cell embeddings, a diffusion time embedding and cell location embeddings using fully connected layers:

$$\mathbf{F}_f^{(l)} = FCN\left(\mathbf{F}^{(l-1)}\right), \tag{4}$$

$$\gamma_t^{(l)} = FCN\left(\gamma_t^{(l-1)}\right), \tag{5}$$

$$\mathbf{R}_r^{(l)} = FCN\left(\mathbf{R}^{(l-1)}\right). \tag{6}$$

The transformed cell embeddings, diffusion time embedding and cell location embeddings are concatenated to form a unified representation for each cell:

$$\mathbf{H}^{(l)} = \left[\mathbf{F}_f^{(l)}, \gamma_t^{(l)} \mathbf{1}_{m \times 1}, \mathbf{R}_r^{(l)}\right], \tag{7}$$

where $\mathbf{1}_{m \times 1}$ is a column vector of ones used to replicate the diffusion time embedding $\gamma_t^{(l)}$ across all m cells. To obtain final cell embeddings $\mathbf{F}^{(l)}$ in layer l, the unified representation $\mathbf{H}^{(l)}$ is passed through a self-attention block which computes attention weights between all pairs of cells (Shen et al., 2021b):

$$\mathbf{F}^{(l)} = \text{SelfAttention}^{(l)}\left(\mathbf{H}^{(l)}\right). \tag{8}$$

This self-attention operation enables the model to aggregate information from all cells in a slice, capturing how

each cell's location is influenced by others. To ensure scalability, we implemented the efficient attention (Shen et al., 2021b) – an approximation of attention computation with a linear complexity. This approach reduces memory usage and computational complexity by circumventing the need for explicit pairwise interactions between every query-key pair (Appendix A).

Finally, the cell embeddings $\mathbf{F}^{(l)}$ are used to predict the locations of each cell. The cell locations are generated from the embeddings using a fully connected layer that projects the location of each cell from the latent embedding space to the two-dimensional physical space:

$$\mathbf{R}^{(l)} = FCN\left(\mathbf{F}^{(l)}\right). \tag{9}$$

At each layer, the updated locations are centered by subtracting the mean, ensuring that the cell locations remain stable and consistent relative to the tissue structure. At the final layer, the output cell locations are the predicted clean cell locations $\hat{\mathbf{R}} = \mathbf{R}^{(L)}$.

2.2. Loss objective

To optimize LUNA, we introduce a specific loss objective that is SE(2)-invariant. This geometric invariance is crucial because the spatial arrangement of cells in the tissue may undergo transformations due to experimental artifacts, while the underlying gene expression profiles remain unchanged.

To ensure SE(2)-invariance, LUNA introduces a pairwise loss function which focuses on preserving the relative distances between all cells in the slice defined as follows:

$$\mathcal{L}_{\text{diff}} = \frac{1}{m^2}\|\Delta(\hat{\mathbf{R}}) - \Delta(\mathbf{R}_{gt})\|_F^2, \tag{10}$$

where $\Delta(\cdot) : \mathbb{R}^{m\times 2} \to \mathbb{R}^{m\times m}$ computes the pairwise squared Euclidean distances between the locations of all cells and $\mathbf{R}_{gt} \in \mathbb{R}^{m\times 2}$ represents the ground true cell locations.

The function $\Delta(\hat{\mathbf{R}})_{ij} = \|\hat{r}_i - \hat{r}_j\|_2^2$ computes the squared distance between the predicted locations of cells i and j, and similarly, $\Delta(\mathbf{R}_{gt})_{ij} = \|r_i - r_j\|_2^2$ computes the squared distance between the true locations. The pairwise loss then becomes:

$$\mathcal{L}_{\text{diff}} = \frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m}(\|\hat{r}_i - \hat{r}_j\|_2^2 - \|r_i - r_j\|_2^2)^2, \tag{11}$$

which penalizes deviations between the predicted and true pairwise distances, ensuring that the spatial relationships between cells are preserved.

The final loss is computed over all slices $s \in \{1, ...S\}$ where S denotes the total number of slices:

$$\mathcal{L} = \frac{1}{S}\sum_{s=1}^{S}\mathcal{L}_{\text{diff}}^{(s)}. \tag{12}$$

2.3. Corruption process

The diffusion model training begins with a corruption process, where LUNA progressively introduces noise to the true cell locations in an autoregressive manner. This step gradually transitions the true cell locations into pure noise, with the resulting corrupted locations serving as inputs for the subsequent denoising process.

We denote the initial diffusion time step, where no noise has been introduced to the cell locations, as time step 0, where ground-truth spatial information is fully retained. Conversely, the maximum diffusion time step, denoted as T, represents the state where no ground-truth information is preserved, and the locations are sampled from pure Gaussian noise. The time step t is randomly selected from $\{0, 1, 2, \ldots, T\}$. The corrupted cell locations at time step t for the slice s are denoted as $\tilde{\mathbf{R}}_t \in \mathbb{R}^{m\times 2}$ where again we omit the slice index s for the ease of notation. From time step $t-1$ to t, Gaussian noise is added to the corrupted cell locations $\tilde{\mathbf{R}}_{t-1}$, yielding noisier locations $\tilde{\mathbf{R}}_t$ according to a predefined noise scheduler (Ho et al., 2020; Song et al., 2021; Sohl-Dickstein et al., 2015). At time step T, the true cell locations $\mathbf{R}_{gt} = \tilde{\mathbf{R}}_0$ are transformed into corrupted cell locations $\tilde{\mathbf{R}}_T$ that have standard Gaussian distribution. The conditional distribution is defined as:

$$q_\phi(\tilde{\mathbf{R}}_t|\tilde{\mathbf{R}}_{t-1}) = \mathcal{N}(\tilde{\mathbf{R}}_t|\sqrt{\alpha_t}\tilde{\mathbf{R}}_{t-1}, (1-\alpha_t)\mathbf{I}), \tag{13}$$

where $q_\phi(\cdot|\cdot)$ represents the corruption process with a noise scheduler ϕ that conditionally transitions the cell location distribution from less corrupted to more corrupted states. $\mathcal{N}(\cdot|\cdot,\cdot)$ denotes a Gaussian distribution, while $\mathbf{I}$ is the identity matrix. The noise scheduler ϕ, regulated by hyperparameters $\{\alpha_t\}$, controls the balance between retaining the original cell locations and adding noise.

To mitigate computational complexity, we compute the conditional distribution of corrupted cell locations given the true cell locations $\mathbf{R}_{gt}$, rather than from the previous diffusion step (Ho et al., 2020; Chan, 2024):

$$q_\phi(\tilde{\mathbf{R}}_t|\tilde{\mathbf{R}}_0) = \mathcal{N}(\tilde{\mathbf{R}}_t|\sqrt{\overline{\alpha}_t}\tilde{\mathbf{R}}_0, (1-\overline{\alpha}_t)\mathbf{I}). \tag{14}$$

Sampling from this conditional distribution is equivalent to adding Gaussian noise to the true cell locations. As shown in (Chan, 2024), the corrupted cell locations are computed as:

$$\tilde{\mathbf{R}}_t = \sqrt{\overline{\alpha}_t}\tilde{\mathbf{R}}_0 + \sqrt{(1-\overline{\alpha}_t)}\epsilon, \quad (15)$$

where $\overline{\alpha}_t = \prod_{i=1}^{t} \alpha_i$, and ϵ is drawn from a Gaussian distribution with mean 0 and variance $\mathbf{I}$, *i.e.*, $\epsilon \sim \mathcal{N}(0, \mathbf{I})$. As t progresses from 0 to T, the cell locations become increasingly corrupted until they are indistinguishable from white noise. At $t = T$, the corrupted locations $\tilde{\mathbf{R}}_T$ are sampled purely from Gaussian noise, with $\overline{\alpha}_T = 0$. To ensure robustness to translations, we additionally subtract the center of mass from the noise ϵ and the data slices $\tilde{\mathbf{R}}_0$, thus working with distributions defined in the subspace where the center of mass is fixed to 0 (Xu et al., 2022; Hoogeboom et al., 2022).

2.4. Denoising process

Given the corrupted locations, we train the LUNA model to recover the true cell locations by reversing the corruption process using gene expression profiles. We denote the model as $\mu_\theta(\cdot,\cdot,\cdot)$ where θ represents all model parameters. LUNA progressively denoises the corrupted cell locations $\tilde{\mathbf{R}}_t$ producing less corrupted locations $\tilde{\mathbf{R}}_{t-1}$ given time step t, more corrupted locations $\tilde{\mathbf{R}}_t$ at the time step t, and the molecular features $\mathbf{X}$.

Eventually, LUNA learns to generate clean cell locations from pure white noise, transitioning from $t = T$ to $t = 0$. To achieve this, the model first predicts the clean cell locations $\hat{\mathbf{R}}$ using the learned network $\mu_\theta(\cdot,\cdot,\cdot)$. It then interpolates between the corrupted locations $\tilde{\mathbf{R}}_t$ and the predicted clean locations $\hat{\mathbf{R}}$, producing less corrupted locations for the next diffusion step:

$$\tilde{\mathbf{R}}_{t-1} = \frac{(1-\overline{\alpha}_{t-1})\sqrt{\alpha_t}}{1-\overline{\alpha}_t}\tilde{\mathbf{R}}_t + \frac{(1-\alpha_t)\sqrt{\overline{\alpha}_{t-1}}}{1-\overline{\alpha}_t}\hat{\mathbf{R}}. \quad (16)$$

Inference phase. Once the model $\mu_\theta(\cdot,\cdot,\cdot)$ is trained, LUNA generates cell locations based on the molecular features of the cells in the test set $\mathbf{X}_{ts} \in \mathbb{R}^{m_{ts} \times d}$. The inference process starts by sampling locations from a normal distribution $\epsilon \sim \mathcal{N}(0, \mathbf{I})$, where $\epsilon \in \mathbb{R}^{m_{ts} \times 2}$. The noisy locations, $\mathbf{R}_{ts,T} = \epsilon$ are then refined through a sequence of diffusion steps.

At each step t, the model takes the noisy locations $\mathbf{R}_{ts,t}$, the molecular features $\mathbf{X}_{ts}$, and the time step t as inputs to predict the denoised locations for the next time step $t-1$:

$$\mathbf{R}_{ts,t-1} = \frac{(1-\overline{\alpha}_{t-1})\sqrt{\alpha_t}}{1-\overline{\alpha}_t}\mathbf{R}_{ts,t} + \frac{(1-\alpha_t)\sqrt{\overline{\alpha}_{t-1}}}{1-\overline{\alpha}_t}\mu_\theta(\mathbf{R}_{ts,t}, \mathbf{X}_{ts}, t). \quad (17)$$

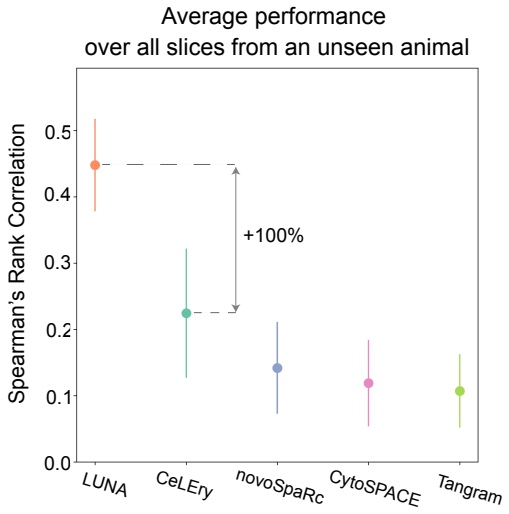

Figure 3. Performance comparison on the cross-animal generalization task between LUNA and alternative baselines on the MERFISH mouse primary motor cortex atlas (Zhang et al., 2021). The performance is measured as the median Spearman's rank correlation between the ground truth spatial coordinates and the predicted spatial coordinates over all slices from an unseen mice (31 slices, 118, 036 cells). Higher correlation indicate better prediction performance. Error bars represent standard deviation across 31 slices.

This process is repeated for each time step, iterating from $t = \{T, ..., 0\}$ until the model generates the final clean cell locations at $t = 0$.

3. Results

3.1. LUNA significantly outperforms other methods

LUNA is a unique method in its capability to reconstruct complex tissue structures. Existing methods are reference mapping methods and require the pre-selection of a single reference slice to which then new cells are mapped based on the similarity of the gene expression between the cells in the reference slice and the input gene expression profiles. This mapping is performed either by explicitly learning a mapping matrix, as in novoSpaRc (Nitzan et al., 2019), Tangram (Biancalani et al., 2021), and CytoSPACE (Vahid et al., 2023), or by implicitly regressing from gene expression space to the physical space, as in CeLEry (Zhang et al., 2023b) (Appendix C).

We compared the performance of LUNA to alternative methods on the spatially resolved cell atlas of the mouse primary motor cortex sequenced by MERFISH (Zhang et al., 2021). We used 31 slices (118, 036 cells) from one animal as the test set and trained LUNA on 33 slices (158, 379 cells) from another mouse to learn spatial priors across multiple slices (Appendix B). During inference, all methods were conditioned only on the gene expression. Since existing methods

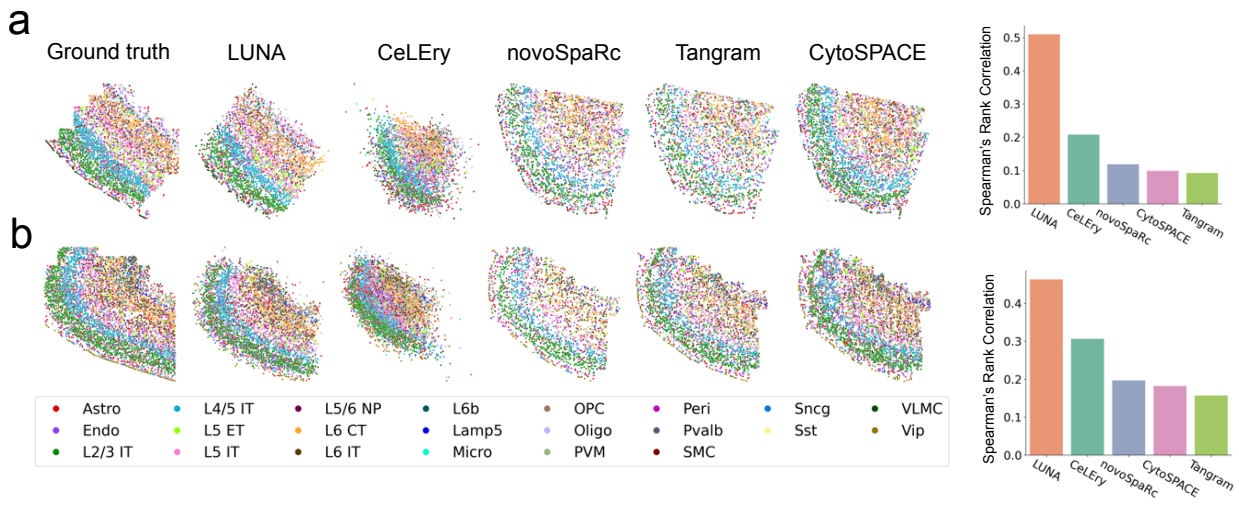

Figure 4. Visualization of ground-truth locations and predictions using LUNA and alternative baselines for one example slices ($5,235$ cells) **(a)** using randomly selected reference slice and **(b)** using the best matched slice as a reference for alternative baselines. Colors denote cell class labels. Performance comparison on that slice using Spearman's rank correlation between the ground truth spatial coordinates and the predicted spatial coordinates as a metric (far right).

can not accept multiple slices as input, we randomly selected a single slice from the training mouse to serve as the reference, repeating the procedure for each slice of the testing mouse (Appendix C).

We evaluated the results using the Spearman's rank correlation between the predicted and ground truth locations, averaging the performance across all slices from the unseen mouse (Appendix D). LUNA achieved an average correlation coefficient of 44.8%, outperforming the best alternative baseline CeLEry by 100% (Figure 3). Performance gains of LUNA were retained using other evaluation metrics, such as precision and RSSD (Figure S1). We visualized the spatial locations predicted by different methods on a sample slice from the unseen test mouse (Figure 4a). LUNA accurately predicted the layer-wise structure of mouse primary motor cortex tissues and correctly inferred the spatial priors of the tissue, such as the contour of the tissue. The limitations of other methods stem from their reliance on pre-selected reference slices, which constrains their generalization capabilities.

Given the sensitivity of reference mapping methods to the choice of reference slices, we further explored alternative methods by selecting the most similar slice from another animal as the reference. To identify the most similar slice, we used cell class information. Specifically, for each of the 31 slices in the test set, we computed the cosine similarity between its cell class distribution and the cell class distribution of all other slices from another animal (Appendix C). Each test slice was then paired with the reference slice that exhibited the highest cosine similarity. We then reassessed the performance of alternative methods using this prior information. This prior information improved the performance of alternative baselines (Figure 4b); however, they still lagged significantly behind LUNA's performance. Even without any prior information, LUNA outperformed all baselines, achieving 37.5% better performance than the best alternative method, CeLEry (Figure S2).

3.2. Whole mouse brain atlas reconstruction

LUNA effectively scales to atlas-scale large datasets and accurately reconstructed a whole mouse brain of the Allen Brain Cell (ABC) MERFISH mouse brain atlas (Zhang et al., 2023a). We trained LUNA on 2.85 million cells across 147 slices from one mouse (Appendix B). We then applied it to reassemble cells from the whole brain of another mouse, never seen during model training, consisting of 1.23 million cells and 338 identified subclasses in 66 slices.

We evaluated LUNA's predictions using the ground-truth cell locations. We found that LUNA's predictions agree well with the ground-truth cell locations across 11 major regions identified in the ABC atlas, despite their distinct structural characteristics. For example, LUNA accurately captured the circular structure of the olfactory bulb region, the layered organization of the isocortex region, and the anatomical separation between the brain stem and cerebrum (Figure 5a). In the slice from the olfactory bulb region affected by sequencing artifacts, LUNA inferred the spatial distribution of cells in areas lacking ground truth spatial information.

To further evaluate the generation results, we compared

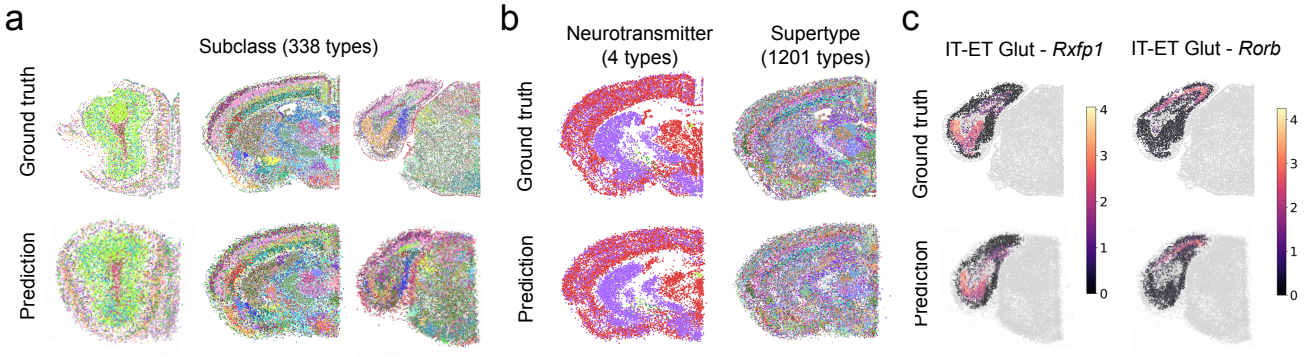

Figure 5. **(a)** Examples of tissue reassembly results using LUNA on three representative slices from distinct major brain regions of the MERFISH whole mouse Allen Brain Cell (ABC) atlas (Zhang et al., 2023a). Cells are coloured according to their cell type annotations with 338 distinct cell subclasses, showing ground truth (top) and predicted (bottom) locations. **(b)** LUNA's predictions across 4 neurotransmitter types and $1,201$ supertypes. Ground truth locations (top) and LUNA's predictions (bottom) for a slice with $35,738$ cells . **(c)** Spatial expression patterns of two spatially variable genes in the IT-ET Glut cell type. Representative genes include *Rxfp1* and *Rorb* for IT-ET Glut. Cells from other cell types are shown in gray color.

LUNA's predictions across various levels of categorization, including the neurotransmitter level (4 types) and cell supertype level ($1,201$ types). LUNA accurately positioned cells at different levels of categorization, from coarser to finer (Figure 5b).

We next assessed LUNA's ability to infer the spatial distribution of gene expression and correctly position cells within a cell type based on their gene expression profiles. We identified spatially variable genes by computing Moran's I statistic (Moran, 1950) and visualized the expression values of genes with the highest Moran's I values (Appendix D). LUNA successfully captured the spatial expression patterns of genes within individual cell types, demonstrating its ability to preserve biological information. For example, LUNA accurately positioned cells expressing *Rxfp1* and *Rorb* in extratelencephalic and intratelencephalic neurons containing the neurotransmitter Glut (Figure 5c).

3.3. De novo reconstruction of scRNA-seq data

We next applied LUNA for *de novo* generation of tissue structures of 1.08 million dissociated single cells across 13 coronal slices from the mouse central nervous system (CNS) scRNA-seq atlas (Zeisel et al., 2018) (Appendix B). We used the model trained on the ABC MERFISH mouse brain atlas to predict spatial locations for cells in the scRNA-seq mouse CNS atlas. Since the scRNA-seq atlas lacks spatial information, we validated LUNA's performance using estimated cell locations derived from the integration of the scRNA-seq atlas with the STARmap PLUS CNS spatial atlas (Shi et al., 2023). In (Shi et al., 2023), in situ sequencing method STARmap PLUS was performed to create the mouse CNS spatial atlas, which was then integrated with the scRNA-seq data. The integrated dataset underwent careful quality control by human experts, ultimately providing estimated cell locations of the scRNA-seq atlas (Shi et al., 2023). To validate LUNA's predictions, we used these estimated cell locations obtained after the integration. Notably, LUNA never used spatial information from the STARmap PLUS atlas.

LUNA 's predictions aligned closely with cell locations estimated through the integration data with the STARmap PLUS atlas at cell class level (27 types; Figure 6a). Additionally, we examined LUNA's predictions for specific neuronal subtypes including di- and mesencephalon excitatory neurons and telencephalon projecting excitatory neurons (Figure 6b), along with their respective sub-molecular classes. The results indicate that LUNA not only accurately placed cells across major cell classes but also precisely predicted the spatial relationships of complex sub-molecular classes within these neuronal cells. This underscores LUNA's capability to capture both broad and intricate cellular architectures within the CNS.

3.4. *In silico* gene knockout with LUNA

We use LUNA to simulate the knockout of individual genes *in silico*, allowing us to investigate their impact on the spatial organization of cells within tissues. During the inference stage of LUNA, we knockout a specific gene by masking its expression to zero and assess its contribution to spatial reconstruction compared to using all genes. To quantify the impact of excluding a certain gene, we measured the discrepancies of the physical spaces between predictions inferred with the complete gene panel and those with one gene knocked out. Using this approach, LUNA can reveal

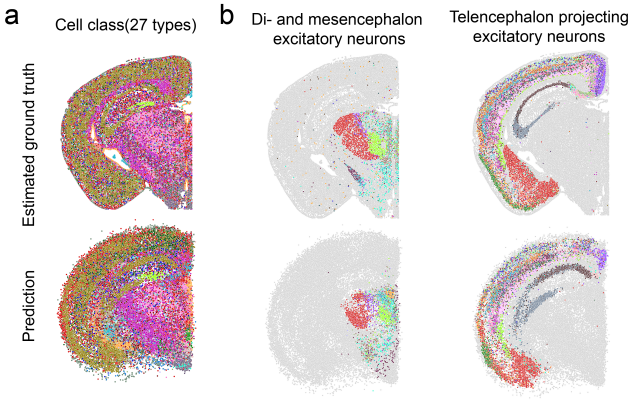

Figure 6. **(a)** Tissue reassembly of the scRNA-seq mouse central nervous system atlas using LUNA for one example slice (35, 738 cells). Cells are colored based on the cell classes (27 types). **(b)** Tissue reassembly of the scRNA-seq mouse central nervous system atlas using LUNA for di- and mesencephalon excitatory neurons (left) and telencephalon projecting excitatory neurons (right). The top plot shows estimated ground truth locations obtained by aligning scRNA-seq atlas with the STARmap atlas and the bottom plot shows LUNA's predictions. Cells from each cell type are colored by their sub-molecule class. Cells from other cell types are shown in gray color.

how specific genes affect the local tissue structure.

We investigated how the exclusion of individual genes affects the predictions of physical locations within major cell classes of the ABC atlas. We train LUNA on all slices from the Animal 1 with the whole gene panel and during inference exclude individual genes from Animal 2. To quantify the effect, we calculated the Spearman Rank Correlation (SRC) within each cell class and determined gene contributions in a cell-class-specific manner. Our analysis revealed that certain genes have a particularly strong effect on prediction accuracy, including *Igfbpl1*, *Cldn11*, *Vgf*, *Nefh*, and *Nr2f1* (Figure 7). These genes showed the highest impact within the olfactory bulb–intracerebral migrating stream GABA neurons (OB-IMN GABA), oligodendrocyte precursor cells and oligodendrocytes (OPC-Oligo), cortex–medial ganglionic eminence GABA neurons (CTX-MGE GABA), and cortex–caudal ganglionic eminence GABA neurons (CTX-CGE GABA), respectively.

Functionally, these genes are closely tied to cell identity and function: *Igfbpl1* is involved in regulating cell growth and GABAergic transmission in mice (Gonda et al., 2007; Butti et al., 2022); *Cldn11* encodes for oligodendrocyte-specific protein essential for myelin sheath formation (Bronstein et al., 2000); *Vgf* and *Nefh* genes are biomarkers related to neurodegenerative diseases and neuronal damage (Quinn et al., 2021; Theunissen et al., 2022); *Nr2f1* plays a crucial role in neurodevelopment, significantly influencing cortical development and network activity maturation (Zhang et al., 2020; Tocco et al., 2021). Interestingly, previous work has shown that myelin mutant mice lacking expression of the *Claudin11* gene in oligodendrocytes exhibit central auditory deficits, reduced anxiety-like behavior and neurotransmitter imbalances (Maheras et al., 2018).

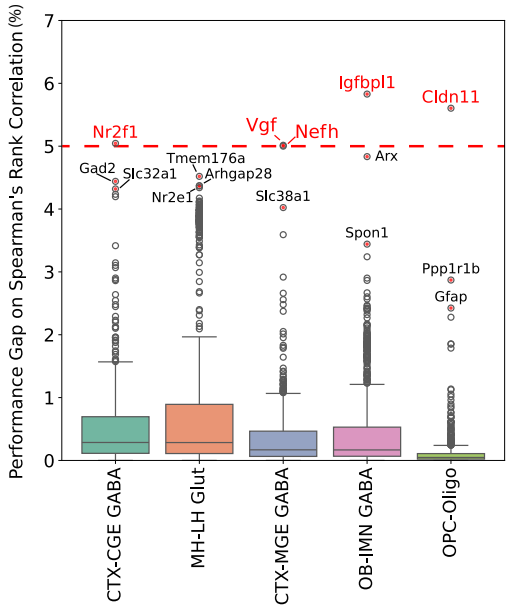

Figure 7. Performance difference caused by excluding each gene for different cell classes. The performance is measured using the Spearman's Rank Correlation (SRC) coefficient computed within each cell class. The top and middle bars of each box represent the 25% quantile and the mean performance difference, respectively.

4. Conclusion

LUNA is a generative AI model that enables *de novo* generation of tissue architectures from their gene expression. By learning spatial priors from existing spatial transcriptomics datasets, LUNA captures the underlying patterns of tissue organization, enabling it to reconstruct complex tissue architectures. LUNA's scalability, accuracy, and versatility position it as a powerful resource for studying tissue spatial organization across a wide range of biological contexts, paving the way towards the construction of comprehensive, spatially resolved single-cell transcriptome atlases and generation of virtual tissue models. We envision that AI-driven tissue reassembly can help to overcome current technological limitations and advance our understanding of tissue organization and function, paving the way towards virtual tissue models.

Software and Data

All analyzed datasets are publicly available. MERFISH Whole Mouse Brain Atlas (ABC Atlas) for Animal 1 is available at https://alleninstitute.github.io/abc_atlas_access/descriptions/Zhuang-ABCA-1.html and for Animal 2 is available at https://alleninstitute.github.io/abc_atlas_access/descriptions/Zhuang-ABCA-2.html. MERFISH Mouse Primary Motor Cortex Atlas is available at the Brain Image Library: https://doi.brainimagelibrary.org/doi/10.35077/g.21. scRNA-seq Mouse Central Nervous System Atlas is available at the Single Cell Portal: https://singlecell.broadinstitute.org/single_cell/study/SCP1830. Slide-tags Datasets for all the tissues are available at the Broad Institute Single Cell Portal under the following accession numbers: SCP2170 (mouse E14), SCP2171 (human melanoma), SCP2169 (human tonsil) and SCP2167 (human brain).

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. We envision that AI-driven tissue reassembly holds great potential for advancing our understanding of complex biological processes and can drive future discoveries in cellular and tissue architecture, ultimately enhancing our ability to map and interpret the spatial structure across multiple biological systems in healthy and disease states.

References

Achim, K., Pettit, J.-B., Saraiva, L. R., Gavriouchkina, D., Larsson, T., Arendt, D., and Marioni, J. C. High-throughput spatial mapping of single-cell RNA-seq data to tissue of origin. *Nature Biotechnology*, 33(5):503–509, 2015.

Biancalani, T. et al. Deep learning and alignment of spatially resolved single-cell transcriptomes with Tangram. *Nature Methods*, 18(11):1352–1362, 2021.

Bronstein, J. M., Tiwari-Woodruff, S., Buznikov, A. G., and Stevens, D. B. Involvement of OSP/Claudin 11 in oligodendrocyte membrane interactions: Role in biology and disease. *Journal of Neuroscience Research*, 59(6): 706–711, 2000.

Butti, E., Cattaneo, S., Bacigaluppi, M., Cambiaghi, M., Scotti, G. M., Brambilla, E., Ruffini, F., Sferruzza, G., Ripamonti, M., Simeoni, F., et al. Neural precursor cells tune striatal connectivity through the release of IGFBPL1. *Nature Communications*, 13(1):7579, 2022.

Chan, S. H. Tutorial on diffusion models for imaging and vision. *arXiv preprint arXiv:2403.18103*, 2024.

Chen, K. H., Boettiger, A. N., Moffitt, J. R., Wang, S., and Zhuang, X. Spatially resolved, highly multiplexed RNA profiling in single cells. *Science*, 348(6233):aaa6090, 2015.

Codeluppi, S., Borm, L. E., Zeisel, A., La Manno, G., van Lunteren, J. A., Svensson, C. I., and Linnarsson, S. Spatial organization of the somatosensory cortex revealed by osmFISH. *Nature Methods*, 15(11):932–935, 2018.

Consortium*, T. T. S., Jones, R. C., Karkanias, J., Krasnow, M. A., Pisco, A. O., Quake, S. R., Salzman, J., Yosef, N., Bulthaup, B., Brown, P., et al. The Tabula Sapiens: A multiple-organ, single-cell transcriptomic atlas of humans. *Science*, 376(6594):eabl4896, 2022.

Gonda, Y., Sakurai, H., Hirata, Y., Tabata, H., Ajioka, I., and Nakajima, K. Expression profiles of Insulin-like growth factor binding protein-like 1 in the developing mouse forebrain. *Gene Expression Patterns*, 7(4):431–440, 2007.

Hickey, J. W., Becker, W. R., Nevins, S. A., Horning, A., Perez, A. E., Zhu, C., Zhu, B., Wei, B., Chiu, R., Chen, D. C., et al. Organization of the human intestine at single-cell resolution. *Nature*, 619(7970):572–584, 2023.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 2020.

Hoogeboom, E., Satorras, V. G., Vignac, C., and Welling, M. Equivariant diffusion for molecule generation in 3D. In *International Conference on Machine Learning*, 2022.

Korsunsky, I., Millard, N., Fan, J., Slowikowski, K., Zhang, F., Wei, K., Baglaenko, Y., Brenner, M., Loh, P.-r., and Raychaudhuri, S. Fast, sensitive and accurate integration of single-cell data with Harmony. *Nature Methods*, 16 (12):1289–1296, 2019.

Lawrence, J., Bernal, J., and Witzgall, C. A purely algebraic justification of the kabsch-umeyama algorithm. *Journal of Research of the National Institute of Standards and Technology*, 124:1, 2019.

Li, H., Janssens, J., De Waegeneer, M., Kolluru, S. S., Davie, K., Gardeux, V., Saelens, W., David, F. P., Brbić, M., Spanier, K., et al. Fly Cell Atlas: A single-nucleus transcriptomic atlas of the adult fruit fly. *Science*, 375(6584): eabk2432, 2022.

Lu, T.-C., Brbić, M., Park, Y.-J., Jackson, T., Chen, J., Kolluru, S. S., Qi, Y., Katheder, N. S., Cai, X. T., Lee, S.,

et al. Aging Fly Cell Atlas identifies exhaustive aging features at cellular resolution. *Science*, 380(6650):eadg0934, 2023.

Macosko, E. Z., Basu, A., Satija, R., Nemesh, J., Shekhar, K., Goldman, M., Tirosh, I., Bialas, A. R., Kamitaki, N., Martersteck, E. M., et al. Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. *Cell*, 161(5):1202–1214, 2015.

Maheras, K. J., Peppi, M., Ghoddoussi, F., Galloway, M. P., Perrine, S. A., and Gow, A. Absence of Claudin 11 in CNS myelin perturbs behavior and neurotransmitter levels in mice. *Scientific Reports*, 8(1):3798, 2018.

Moran, P. A. Notes on continuous stochastic phenomena. *Biometrika*, 37(1/2):17–23, 1950.

Moriel, N. et al. NovoSpaRc: Flexible spatial reconstruction of single-cell gene expression with optimal transport. *Nature Communications*, 12(1):1–12, 2021.

Nitzan, M., Karaiskos, N., Friedman, N., and Rajewsky, N. Gene expression cartography. *Nature*, 576(7785): 132–137, 2019.

Patel, A. P., Tirosh, I., Trombetta, J. J., Shalek, A. K., Gillespie, S. M., Wakimoto, H., Cahill, D. P., Nahed, B. V., Curry, W. T., Martuza, R. L., et al. Single-cell RNA-seq highlights intratumoral heterogeneity in primary glioblastoma. *Science*, 344(6190):1396–1401, 2014.

Quinn, J. P., Kandigian, S. E., Trombetta, B. A., Arnold, S. E., and Carlyle, B. C. VGF as a biomarker and therapeutic target in neurodegenerative and psychiatric diseases. *Brain Communications*, 3(4):fcab261, 2021.

Romero, D., Bekkers, E., Tomczak, J., and Hoogendoorn, M. Attentive group equivariant convolutional networks. In *International Conference on Machine Learning*, 2020.

Satija, R., Farrell, J. A., Gennert, D., Schier, A. F., and Regev, A. Spatial reconstruction of single-cell gene expression data. *Nature Biotechnology*, 33(5):495–502, 2015.

Segol, N. and Lipman, Y. On universal equivariant set networks. In *International Conference on Learning Representations*, 2020.

Shen, Z., Zhang, M., Zhao, H., Yi, S., and Li, H. Efficient attention: Attention with linear complexities. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021a.

Shen, Z., Zhang, M., Zhao, H., Yi, S., and Li, H. Efficient attention: Attention with linear complexities. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021b.

Shi, H., He, Y., Zhou, Y., Huang, J., Maher, K., Wang, B., Tang, Z., Luo, S., Tan, P., Wu, M., et al. Spatial atlas of the mouse central nervous system at molecular resolution. *Nature*, 622(7983):552–561, 2023.

Sikkema, L., Ramírez-Suástegui, C., Strobl, D. C., Gillett, T. E., Zappia, L., Madissoon, E., Markov, N. S., Zaragosi, L.-E., Ji, Y., Ansari, M., et al. An integrated cell atlas of the lung in health and disease. *Nature Medicine*, 29(6): 1563–1577, 2023.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, 2015.

Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 2019.

Tang, F., Barbacioru, C., Wang, Y., Nordman, E., Lee, C., Xu, N., Wang, X., Bodeau, J., Tuch, B. B., Siddiqui, A., et al. mRNA-Seq whole-transcriptome analysis of a single cell. *Nature Methods*, 6(5):377–382, 2009.

Theunissen, F., Anderton, R. S., Mastaglia, F. L., James, I., Bedlack, R., and Akkari, P. A. Intronic NEFH variant is associated with reduced risk for sporadic ALS and later age of disease onset. *Scientific Reports*, 12(1):14739, 2022.

Tocco, C., Bertacchi, M., and Studer, M. Structural and functional aspects of the neurodevelopmental gene NR2F1: From animal models to human pathology. *Frontiers in Molecular Neuroscience*, 14:767965, 2021.

Torres-Mendez, L. A., Ruiz-Suarez, J. C., Sucar, L. E., and Gómez, G. Translation, rotation, and scale-invariant object recognition. *IEEE Transactions on Systems, Man, and Cybernetics*, 30(1):125–130, 2000.

Vahid, A. et al. High-resolution mapping of the spatial organization of single cells in tissues using CytoSPACE. *Nature Methods*, 20(3):362–373, 2023.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

Wang, X., Allen, W. E., Wright, M. A., Sylwestrak, E. L., Samusik, N., Vesuna, S., Evans, K., Liu, C., Ramakrishnan, C., Liu, J., et al. Three-dimensional intact-tissue sequencing of single-cell transcriptional states. *Science*, 361(6400):eaat5691, 2018.

Wolf, F. A., Angerer, P., and Theis, F. J. SCANPY: Large-scale single-cell gene expression data analysis. *Genome Biology*, 19:1–5, 2018.

Wu, Y., Löwe, H., Carricato, M., and Li, Z. Inversion symmetry of the Euclidean group: theory and application to robot kinematics. *IEEE Transactions on Robotics*, 32 (2):312–326, 2016.

Xu, M., Yu, L., Song, Y., Shi, C., Ermon, S., and Tang, J. GeoDiff: A geometric diffusion model for molecular conformation generation. In *International Conference on Learning Representations*, 2022.

Zeisel, A., Hochgerner, H., Lönnerberg, P., Johnsson, A., Memic, F., Van Der Zwan, J., Häring, M., Braun, E., Borm, L. E., La Manno, G., et al. Molecular architecture of the mouse nervous system. *Cell*, 174(4):999–1014, 2018.

Zhang, K., Yu, F., Zhu, J., Han, S., Chen, J., Wu, X., Chen, Y., Shen, T., Liao, J., Guo, W., et al. Imbalance of excitatory/inhibitory neuron differentiation in neurodevelopmental disorders with an NR2F1 point mutation. *Cell Reports*, 31(3), 2020.

Zhang, M., Eichhorn, S. W., Zingg, B., Yao, Z., Cotter, K., Zeng, H., Dong, H., and Zhuang, X. Spatially resolved cell atlas of the mouse primary motor cortex by MERFISH. *Nature*, 598(7879):137–143, 2021.

Zhang, M., Pan, X., Jung, W., Halpern, A. R., Eichhorn, S. W., Lei, Z., Cohen, L., Smith, K. A., Tasic, B., Yao, Z., et al. Molecularly defined and spatially resolved cell atlas of the whole mouse brain. *Nature*, 624(7991):343–354, 2023a.

Zhang, W. et al. Leveraging single-cell spatial transcriptomics with CeLEry for spatial reconstruction. *Nature Biotechnology*, 41(2):245–256, 2023b.

A. Efficient Attention

LUNA learns cell representations that capture cellular interactions globally and locally across entire tissue slice. To ensure scalability, we implemented Efficient Attention (Shen et al., 2021a). Different from traditional dot-product attention mechanism, Efficient Attention is an approximation of attention computation with linear complexity.

Dot-product Attention. Dot-product attention (Vaswani et al., 2017), the traditional method for implementing attention mechanisms, captures pairwise dependencies between input elements by transforming each input feature vector $x_i \in \mathbb{R}^d$ into three distinct vectors: a query $q_i \in \mathbb{R}^{d_k}$, a key $k_i \in \mathbb{R}^{d_k}$, and a value $v_i \in \mathbb{R}^{d_v}$. These transformations are performed through linear layers, which apply learned weight matrices to the input feature vectors. The similarities between the query and key vectors are computed by taking their dot products, capturing the interactions between different input features.

In matrix form, the set of queries, keys, and values are represented as $Q \in \mathbb{R}^{n \times d_k}$, $K \in \mathbb{R}^{n \times d_k}$, and $V \in \mathbb{R}^{n \times d_v}$, respectively, where n represents the number of input feature vectors (in our case, the number of cells). The core operation of dot-product attention is expressed as:

$$D(Q, K, V) = \rho\left(QK^{\mathsf{T}}\right) V$$

Here, $\rho(Y)$ is a normalization function, typically a row-wise softmax, denoted as $\sigma_{\text{row}}(Y)$, which normalizes the raw dot-product scores across each row of the matrix $Y = QK^{\mathsf{T}}$. This ensures that the attention scores are positive and sum to 1 across each row, allowing the model to focus on the most relevant keys for each query while still considering all input features. The row-wise softmax function $\sigma_{\text{row}}(Y)$ for a given row i is defined as:

$$\sigma_{\text{row}}(Y)_i = \frac{\exp(Y_{ij})}{\sum_{j'} \exp(Y_{ij'})}$$

where Y_{ij} represents the element in the i-th row and j-th column of the matrix Y. This transformation converts the raw similarity scores into normalized attention weights. The resulting attention weights are then used to compute a weighted sum of the value vectors V, where the weight assigned to each value vector is determined by its corresponding attention score.

Thus, the final output of dot-product attention becomes:

$$D(Q, K, V) = \sigma_{\text{row}}\left(QK^{\mathsf{T}}\right) V$$

This mechanism enables the model to dynamically adjust its focus on different parts of the input sequence, ensuring that it captures the most important interactions between elements while maintaining a global view of the data through the attention weights.

Dot-product attention is effective but requires calculating all pairwise similarities, which leads to high memory and computational demands. Specifically, the memory complexity is $O(n^2)$, and the computational complexity is $O(d_k n^2)$, where n represents the number of input feature vectors. These complexities grow quadratically with n, making it challenging to scale to large inputs.

Efficient Attention. Efficient attention (Shen et al., 2021a) addresses the scaling issues of dot-product attention by reinterpreting the keys as global feature maps, eliminating the need to compute similarities for each individual pair of queries and keys. Instead, it uses the keys to summarize the input globally, and this summary is then applied to the queries. The output of efficient attention is computed as:

$$E(Q, K, V) = \rho(Q)\left(\rho(K)^{\mathsf{T}} V\right)$$

Efficient attention reduces memory and computation by avoiding the computation of pairwise interactions for each query-key pair. Instead, it aggregates global context vectors based on the keys. This leads to the fact that efficient attention scales linearly with n, making it a more practical solution for large-scale data without sacrificing performance. When using softmax normalization, the overall memory complexity becomes $O(dn + d^2)$ and the computational complexity becomes

$O(d^2n)$, assuming that $d_v = d_k = \frac{d}{2}$. In practice, $d \ll n$, therefore the complexity of Efficient Attention becomes linear to the number of cells n. Thus, making efficient attention a better option for processing large-scale data, from a computational standpoint.

Method	Memory Complexity	Computational Complexity
Dot-Product Attention	$O(n^2)$	$O(d_k n^2)$
Efficient Attention	$O(dn + d^2)$	$O(d^2 n)$

Table S1. Comparison of the memory and computational complexities of dot-product attention and efficient attention.

B. Datasets and preprocessing

In LUNA, we considered the following datasets:

- **MERFISH Whole Mouse Brain Atlas (ABC Atlas) (Zhang et al., 2023a)**: We utilized raw gene expression data from Animal 1 (2.85 million cells) and Animal 2 (1.23 million cells) from the ABC atlas dataset. The data was log2-transformed, and $1,122$ genes were selected as input features. All slices from Animal 1 were used for training, and all slices from Animal 2 were used for testing. In the zero-shot setting, we trained LUNA on all slices from Animal 1, excluding cells from randomly selected cell classes, and applied LUNA to predict tissue structures in Animal 2, including the unseen cells, which were not part of the training set. All slices from Animal 1 were used for training, and all slices from Animal 2 were used for testing.

- **MERFISH Mouse Primary Motor Cortex Atlas (Zhang et al., 2021)**: This dataset consisted of raw gene expression data from 33 slices ($158,379$ cells) from one animal and 31 slices ($118,036$ cells) from another animal. After applying a log2 transformation, 254 genes were used as input features. We trained LUNA using all 33 slices from the first animal and tested it using all 31 slices from the second animal.

- **scRNA-seq Mouse Central Nervous System Atlas (Zeisel et al., 2018)**: Wang *et al.* (Shi et al., 2023) constructed the Mouse Central Nervous System (CNS) atlas by integrating STARmap PLUS measurements (Shi et al., 2023) with a published single-cell RNA-sequencing atlas (Zeisel et al., 2018). This integration resulted in the imputation of expression profiles for $11,844$ genes and the estimation of spatial locations. We utilized this publicly available dataset from Wang *et al.* (Shi et al., 2023), which includes imputed transcriptomes and estimated spatial coordinates. By intersecting the gene panels from the ABC atlas and the CNS dataset, we identified 804 common genes. The CNS dataset comprises 13 coronal slices containing 1.08 million cells. Before integrating, both datasets were log2-transformed. For running LUNA, we integrated the gene expression matrices for these 804 common genes from both datasets—ABC Atlas (Animal 1) with 2.85 million cells and the CNS dataset with 1.08 million cells—using the Harmony method (Korsunsky et al., 2019). This integration, performed via the scanpy.external.pp.harmony_integrate function from the Scanpy library (Wolf et al., 2018), resulted in a 600-dimensional latent space. The ABC dataset served as the training set, while the CNS dataset was utilized for testing. The effectiveness of the model was assessed by comparing the results to the estimated ground truth locations provided in the work of Wang *et al.* (Shi et al., 2023).

All spatial locations were normalized to a range of -0.5 to 0.5 and used as the ground truth for cell spatial coordinates.

C. Baselines

We compared LUNA with four existing tissue spatial reconstruction methods—novoSpaRc (Nitzan et al., 2019), Tangram (Biancalani et al., 2021), CytoSPACE (Vahid et al., 2023), and CeLEry (Zhang et al., 2023b)—using the MERFISH Mouse Primary Cortex Atlas dataset (Zhang et al., 2021) with in total 254 genes. Notably, CeLEry is the only method capable of processing multiple slices simultaneously. Consequently, we trained CeLEry using all 33 slices from the first animal and tested it on all 31 slices from the second animal. For each method, we inferred locations for all slices from the test set, which includes 31 slices, under two scenarios:

1. For each slice in the test set, we randomly select one of the 33 slices from the training set as a reference using a fixed random seed.

2. For each slice in the test set, we calculate the cosine similarity between its cell class distribution and those of the 33 training set slices in LUNA. We then matched each test slice with the training slice that showed the highest cosine similarity which should be the best match as a reference slice.

We conducted extensive hyperparameter tuning for all methods. For novoSpaRc (Nitzan et al., 2019),we use the cell locations of the reference slice as the target mapping space. We conducted 50 trials of hyperparameter tuning to optimize the model's performance. During each trial, we explored different values for two key hyperparameters: α_{linear} and ϵ. The parameter α_{linear}, which controls the trade-off between prior spatial information and gene expression similarity, was tested across values ranging from 0.1 to 1 in increments of 0.1. The ϵ parameter, which governs the convergence of the reconstruction algorithm, was varied across values $5\times10^{-5}, 1\times10^{-4}, 5\times10^{-4}, 1\times10^{-3}, 5\times10^{-3}, 1\times10^{-2}, 5\times10^{-2}, 1\times10^{-1},$ and 1. For Tangram (Biancalani et al., 2021), we run it at the cell level with "uniform" density prior for MERFISH data by calling "tg.map_cells_to_space" and explored different values for the learning rate. The learning rate was tested across $5\times10^{-5}, 1\times10^{-4}, 5\times10^{-4}, 1\times10^{-3}, 5\times10^{-3}, 1\times10^{-2}, 5\times10^{-2}, 1\times10^{-1},$ and 1. For CytoSPACE (Vahid et al., 2023), we run it under default settings of single-cell mode since it has no hyperparameters to tune. For CeLEry (Zhang et al., 2023b), we performed 50 trials for hyperparameter optimization. During each trial, we explored different values for the learning rate, the dimension of the latent embedding, and the number of layers. The learning rate was tested across $5\times10^{-5}, 1\times10^{-4}, 5\times10^{-4}, 1\times10^{-3}, 5\times10^{-3}, 1\times10^{-2}, 5\times10^{-2}, 1\times10^{-1},$ and 1. The dimension of the latent embedding was explored with values of $32, 64, 128,$ and 256, while the number of layers was varied across $3, 4, 5, 7, 8, 9, 10$.

In the paper, we report the results from the hyperparameter configuration that achieved the best performance for each of the baseline models.

D. Evaluation Metrics

There are various ways to assess the quality of tissue spatial reconstruction results. A high-quality reconstruction should preserve the local neighborhood composition of each cell, accurately capture the spatial distribution across different cell types, reflect biologically meaningful spatial gradients for various genes, and be robust to artificial effects.

We evaluate the quality of the cell location predictions across these dimensions using several metrics:

Spearman's Rank Correlation (SRC). SRC quantifies the strength and direction of association between two ranked variables. In our study, SRC is computed by comparing the ranks of pairwise distances between predicted and actual spatial coordinates for a group of cells. Specifically, for each cell, we calculate the rank correlation between its predicted pairwise distances to all other cells and the actual (ground truth) pairwise distances:

$$\rho = 1 - \frac{6\sum d_i^2}{n(n^2-1)}$$

where d_i represents the difference between the ranks of each predicted and actual pairwise distance, and n is the total number of pairwise comparisons (cell pairs) in the group.

For each slice, we compute the Spearman Rank Correlation (SRC) for all the cells. We then aggregate these correlations across all cells in the slice by calculating either the average—referred to as **Spearman Rank Correlation (Average)**, or the median—referred to as **Spearman Rank Correlation (Median)**.

Moran's I. Moran's I is a measure of spatial autocorrelation used to assess whether the spatial distribution of gene expression is clustered, dispersed, or random. It quantifies the degree of similarity between gene expression values at nearby cells based on a weighted spatial graph. The formula for Moran's I is:

$$I = \frac{N\sum_{i,j} w_{i,j} z_i z_j}{S_0 \sum_i z_i^2}$$

where N is the total number of cells, $w_{i,j}$ represents the spatial weight (distance) between cells i and j, z_i and z_j are the deviations of the gene expression values from the mean for cells i and j, respectively, and S_0 is the sum of all spatial weights. In our case, $w_{i,j}$ corresponds to the distance between cell i and cell j, while z_i and z_j denote the expression levels of a specific gene in these cells. A high Moran's I value indicates spatial clustering of gene expression, whereas a low value suggests spatial dispersion.

Precision. The precision is for evaluating the performance of our model in predicting contacts between cells. In the contact prediction task, we classify pairs of cells as being in contact or not based on the distances between them. To define what constitutes a contact, we use a threshold determined by a specific percentile of the distance distribution, referred to as the **percentile**. For example, using the 10% percentile means we consider the top 10% closest cell pairs as contacts.

Once the threshold is established, we label cell pairs with distances less than the threshold as positive (contacts) and those with distances greater than or equal to the threshold as negative (non-contacts). With these labels, we calculate the **Precision** (P) measures the proportion of predicted contacts that are actual contacts and is defined as:

$$P = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}.$$

Root Sum Square Deviation (RSSD). The Root Sum Square Deviation (RSSD) is a metric used to quantify the total deviation between predicted and true cell positions in spatial data after optimal alignment. It provides an overall measure of how accurately the model predicts cell locations by comparing the predicted positions to the true ones.

Given two sets of two-dimensional points, the true positions $\mathbf{V} = \{v_1, v_2, \ldots, v_n\}$ and the predicted positions $\mathbf{W} = \{w_1, w_2, \ldots, w_n\}$, where both sets contain n points (cells), the RSSD is calculated after aligning $\mathbf{W}$ to $\mathbf{V}$ using the Kabsch algorithm (Lawrence et al., 2019). This algorithm finds the optimal rotation (and optionally translation) that minimizes the deviation between the two sets.

The *Global RSSD*, denoted as **Absolute RSSD** ($\text{RSSD}_{\textbf{absolute}}$) , measures the total alignment error across all cells and is defined as:

$$\text{RSSD}_{\text{absolute}} = \sqrt{\sum_{i=1}^{n} \|v_i - w_i'\|^2},$$

where w_i' represents the predicted positions after optimal alignment. To evaluate the model's performance on individual cell types, we compute the *Per-Class RSSD* for each cell class c in the set of all classes $\mathcal{C}$. For a specific class c, we extract the subsets $\mathbf{V}_c$ and $\mathbf{W}_c$, representing the true and predicted positions of cells in class c, respectively. The Per-Class RSSD is then given by:

$$\text{RSSD}_c = \sqrt{\sum_{i=1}^{n_c} \|v_i^c - w_i^{c\prime}\|^2},$$

where n_c is the number of cells in class c, and $w_i^{c\prime}$ are the aligned predicted positions for class c. To summarize the performance across all classes, we compute two aggregate metrics: **Sum RSSD** ($\text{RSSD}_{\textbf{sum}}$), which reflects the cumulative deviation across all classes and **Mean RSSD** ($\text{RSSD}_{\textbf{mean}}$) which represents the average deviation per class, given by:

$$\text{RSSD}_{\text{sum}} = \sum_{c \in \mathcal{C}} \text{RSSD}_c.$$

$$\text{RSSD}_{\text{mean}} = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \text{RSSD}_c,$$

where $|\mathcal{C}|$ is the total number of cell classes.

E. Hyperparameters and Model Selection

Hyperparameters. In LUNA, we maintain a consistent learning rate of 5×10^{-4} across all experiments, with a maximum diffusion time of 1000 steps and a cosine noise schedule characterized by $\nu = 2$, which dictates the noise addition rate to the coordinates. The model consistently employs a latent dimension of 64 for position encoding and utilizes 16 attention heads in every experiment.

For large-scale atlas experiments such as the MERFISH Whole Mouse Brain Atlas, MERFISH Mouse Primary Motor Cortex Atlas, and scRNA-seq Mouse Central Nervous System Atlas, we implement 8 transformer layers.

The latent dimension of node features is set at 384 for the MERFISH Whole Mouse Brain Atlas and the scRNA-seq Mouse Central Nervous System Atlas. For the smaller MERFISH Mouse Primary Motor Cortex Atlas experiment, we adjust the latent dimension to 256.

Model Selection. For large-scale datasets, including the MERFISH Whole Mouse Brain Atlas and the scRNA-seq Mouse Central Nervous System Atlas, we adopt a model trained for up to 3500 epochs. This extensive training period accommodates the complexity and size of these datasets. Conversely, for smaller datasets such as the MERFISH Mouse Primary Motor Cortex Atlas experiment, the model is sufficiently trained after 1000 epochs, optimizing performance without overfitting to limited data.

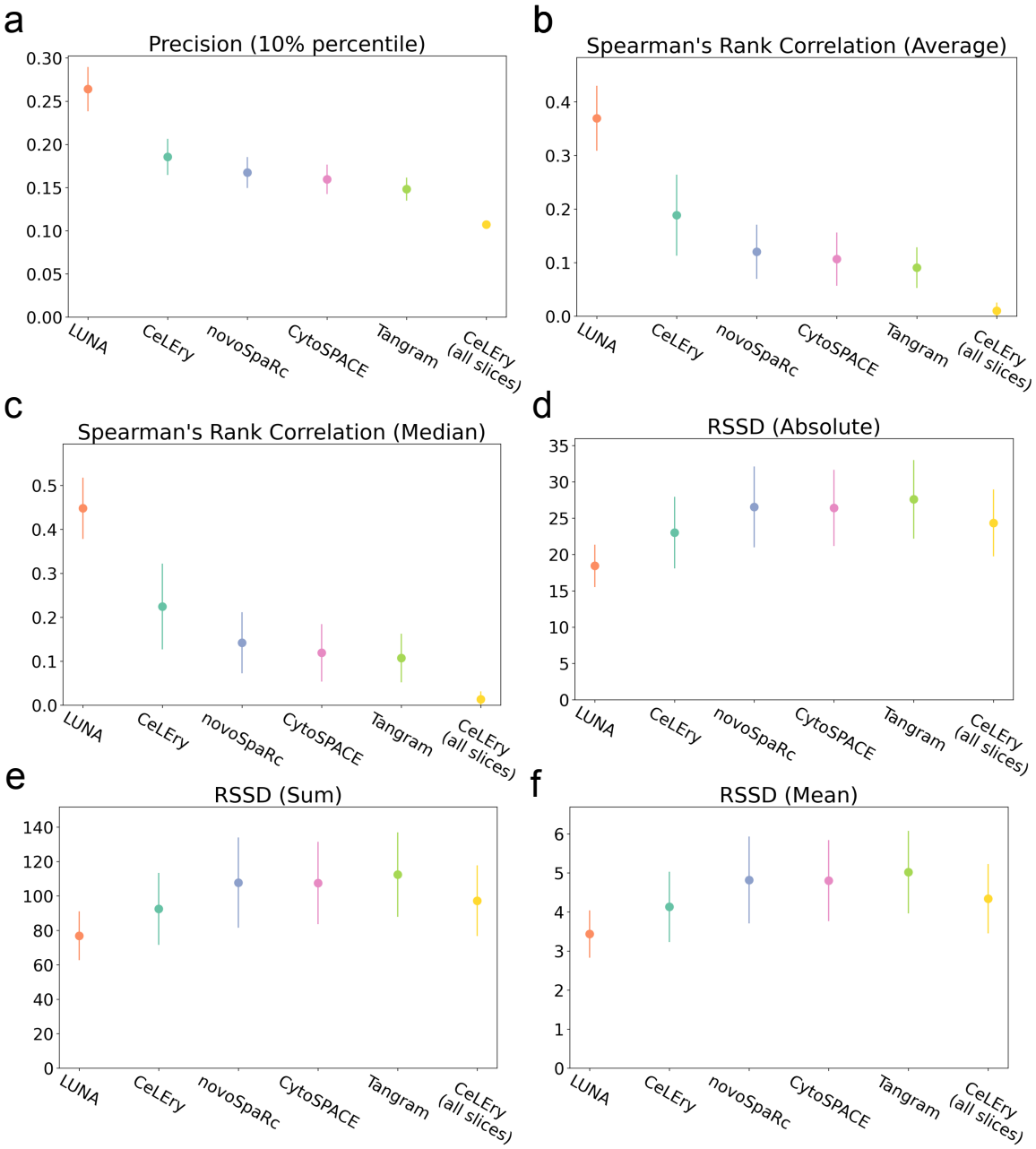

Figure S1. Performance comparison on the cross-animal generalization task between LUNA and alternative baselines on the MERFISH mouse primary motor cortex atlas (Zhang et al., 2021). Models were trained on slices from one mouse and evaluated on all slices from an unseen mouse (31 slices, 118, 036 cells). We evaluated performance based on precision at the 10% percentile threshold for predicting the closest cell pairs as contacts **(a)**, average and median Spearman's rank correlation across all cells within the same slice **(b, c)**, and absolute, sum, and mean Root Sum Square Deviation (RSSD) between the ground truth and predicted spatial coordinates **(d-f)**.

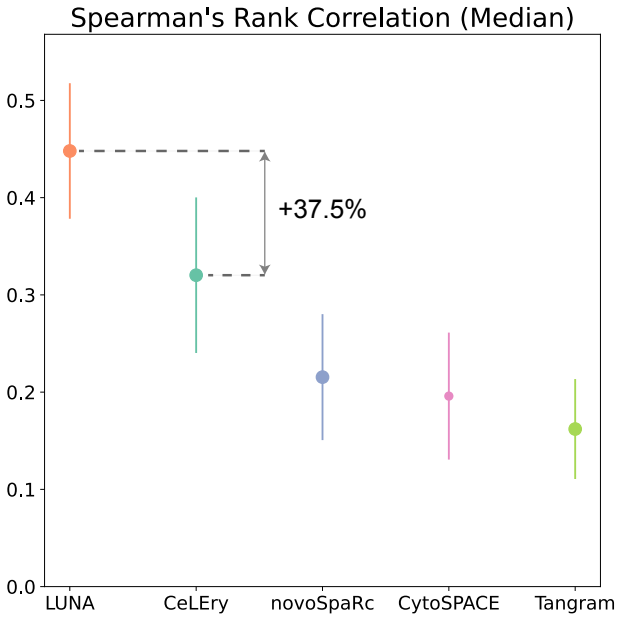

Figure S2. Evaluation of alternative methods using the best match of the reference slice. For each fixed test slice, the most similar slice from the train set (including 33 slices) was selected by computing the cosine similarity of the cell type distributions. The average performance of these methods was computed across all slices from an unseen mouse, comprising 31 slices and 118, 036 cells. Performance was evaluated based on the median Spearman's rank correlation across all cells within the same slice.