ENHANCING THE EXPRESSIVITY OF SINGLE-CELL CHROMATIN ACCESSIBILITY REPRESENTATIONS THROUGH ADAPTIVE REGION EMBEDDINGS

Anonymous authors
Paper under double-blind review

ABSTRACT

Single-cell assay for transposase-accessible chromatin using sequencing (scATAC-seq) enables the characterization of epigenomic regulatory landscapes and cellular heterogeneity at single-cell resolution. However, learning effective representations from scATAC-seq data remains challenging because of its extreme sparsity, high dimensionality, and variability in genomic regions across datasets. Here, we introduce scChromHEARTs, a representation learning framework for **s**ingle-**c**ell **Chrom**atin accessibility based on **H**igh-**E**xpressivity **A**daptive **R**egion-aware embedding **T**ransformation**s** and a tailored self-supervised learning strategy. Rather than assigning fixed embeddings to predefined genomic regions, scChromHEARTs generates adaptive embeddings for arbitrary genomic intervals from their chromosomal identities and genomic coordinates. Combined with subset-based contrastive learning, this design enables efficient representation learning across the full genomic region space. scChromHEARTs achieved strong performance across diverse downstream applications, including cell-type annotation, perturbed cell generation, batch correction, and cross-dataset transferability. Ablation studies further demonstrate that input embeddings designed to finely encode genomic structure substantially improve representation quality. These results highlight the importance of designing input embeddings for effective representation learning of epigenomic data. Code is available at `https://github.com/anonymous/ChromHEARTs`.

1 INTRODUCTION

Single-cell assay for transposase-accessible chromatin using sequencing (scATAC-seq) provides a genome-wide view of chromatin accessibility at single-cell resolution, enabling the characterization of cellular heterogeneity and gene regulatory landscapes (Cusanovich et al., 2018; Klemm et al., 2019; Minnoye et al., 2021; Terekhanova et al., 2023). Computational methods for scATAC-seq have progressed from conventional dimensionality reduction to deep learning and, more recently, language-model-based approaches. Early approaches commonly used latent semantic indexing (LSI) (Cusanovich et al., 2018; Stuart et al., 2021; Granja et al., 2021). Subsequent methods introduced alternative representations, including latent regulatory topics in cisTopic (Bravo González-Blas et al., 2019) and genome-wide cell–cell accessibility similarities in SnapATAC (Fang et al., 2021). Deep learning methods such as SCALE (Xiong et al., 2019) and PeakVI (Ashuach et al., 2022) further learned nonlinear latent representations from sparse accessibility profiles, while scBasset (Yuan & Kelley, 2022) incorporated the underlying DNA sequence using convolutional neural networks.

The recent success of language-model-based representation learning in single-cell transcriptomics (Theodoris et al., 2023; Cui et al., 2024; Hao et al., 2024; Rosen et al., 2026) has motivated similar approaches for scATAC-seq. In this paradigm, accessible genomic regions are treated as tokens and processed by sequence models to learn cell representations. For example, EpiAgent (Chen et al., 2025) represents accessible cis-regulatory elements (cCREs) using a predefined genomic vocabulary and processes them with a Transformer encoder (Vaswani et al., 2017), whereas ChromFound (Jiao et al., 2026) adopts genome-aware tokenization with a Mamba-based architecture (Gu & Dao, 2023) without relying on a predefined genomic vocabulary.

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

However, conventional token embedding schemes make effective representation learning more challenging for scATAC-seq than for scRNA-seq. This is largely due to the much larger and sparser feature space of scATAC-seq. In detail, scATAC-seq can contain 100,000 to over 1,000,000 genomic regions, with only about 1,000–10,000 accessible in each cell, whereas scRNA-seq typically contains 20,000–30,000 genes, with roughly 1,000–5,000 detected per cell. Moreover, dependencies among accessible regions are generally weaker and more localized than those among genes, further limiting the ability to learn relationships between regions. Because conventional lookup tables assign an independent embedding to each region without explicitly encoding genomic relationships, these relationships must be learned largely from the training data. This reliance on data-driven learning is particularly challenging for scATAC-seq, where available training corpora remain relatively small; for example, the Human-scATAC-Corpus (Chen et al., 2026) contains approximately 5.4 million cells, whereas large-scale scRNA-seq corpora can exceed 50 million cells (Hao et al., 2024).

These challenges are reflected in recent language-model-based scATAC-seq models. EpiAgent relies on a predefined cCRE vocabulary, which limits flexibility and requires relationships among regions to be learned largely from pre-training data. ChromFound provides more flexible coordinate-based tokenization, but encodes chromosome identity and genomic position separately and combines them additively. It provides only a limited representation of the structural relationship between chromosome identity and genomic position. These limitations motivate a more structured region representation that jointly captures chromosome identity and genomic position.

To this end, we propose scChromHEARTs, a representation learning framework for **s**ingle-**c**ell **Chrom**atin accessibility based on **H**igh-**E**xpressivity **A**daptive **R**egion-aware embedding **T**ransformation**s** (Fig. 1a). Instead of a large predefined vocabulary, scChromHEARTs represents genomic regions using a compact set of chromosome-specific token, bias, and gate embeddings. The chromosome token embedding provides the base representation of chromosome identity, while genomic coordinates are incorporated through rotary transformation (Su et al., 2024) applied at the region boundaries. The chromosome-specific gate and bias embeddings are further transformed according to the region position and provide multiplicative and additive modulation, respectively. Together, these components produce expressive genomic region embeddings. Our region embeddings are aggregated by a Transformer encoder into cell-level representations, which are learned through contrastive pre-training (Park & Lee, 2024) and genomic region accessibility prediction.

Our main contributions are threefold: (1) we introduce a flexible and expressive genomic region embedding scheme that combines chromosome-level token, bias, and gate embeddings with rotary transformations that encode genomic coordinates; (2) through ablation studies, we demonstrate that the proposed embeddings enable effective representation learning; and (3) we demonstrate strong performance across multiple downstream tasks. Together, our findings highlight input embedding expressivity as a key factor in enabling effective learning from sparse and high-dimensional scATAC-seq data.

2 Related work

Existing scATAC-seq representation methods include both large-scale pre-trained models and dataset-specific approaches. EpiAgent (Chen et al., 2025) uses a Transformer over accessible cCREs defined by a fixed genomic vocabulary, whereas ChromFound (Jiao et al., 2026) uses genome-aware tokenization with a Mamba-based architecture (Gu & Dao, 2023). Dataset-specific methods such as PeakVI (Ashuach et al., 2022) and PoissonVI (Martens et al., 2024) learns latent representations directly from the target accessibility matrix. CellCano (Ma et al., 2023) performs supervised cell-type annotation using gene-level accessibility scores and self-knowledge distillation, while SANGO (Zeng et al., 2024) learns cell representations by reconstructing the binary accessibility matrix with genomic sequence information. Additional details on related methods are provided in Appendix A.

3 Method

Rather than relying on a large learnable embedding lookup table, scChromHEARTs represents arbitrary genomic regions using chromosome-specific token, bias, and gate embeddings transformed by their genomic coordinates. A Transformer encoder is pretrained using contrastive learning and

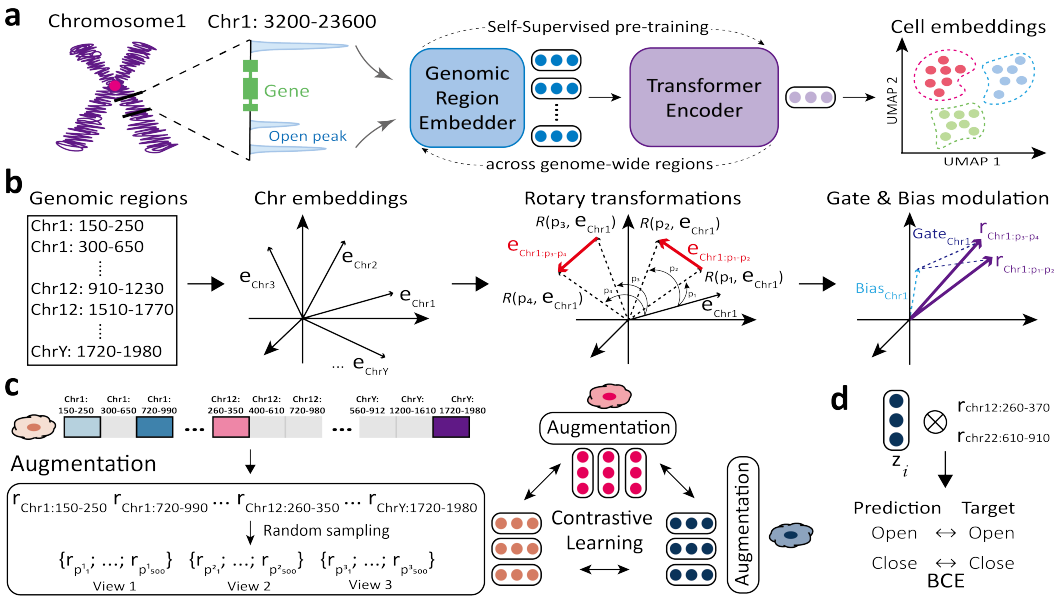

Figure 1: (a) The overview of scChromHEARTs. (b) Genomic region embedder of scChromHEARTs (Section 3.1). (c,d) Pre-training objectives based on multi-positive contrastive learning and genomic region accessibility prediction, respectively (Section 3.2).

genomic region accessibility prediction. The following subsections describe the genomic region embedder, encoder, and pre-training strategy.

3.1 Expressive Genomic Region Embedder

Chromosome token, bias, and gate embeddings Features in scATAC-seq are defined by genomic intervals, each specified by a chromosome identity and genomic coordinates (e.g., `chr1:5200-9000`). Since the human genome contains only 24 chromosome types, chromosome identities can be efficiently represented using a small learnable embedding lookup table. Specifically, chromosome token, bias, and gate embeddings are defined by trainable embedding matrices $E = [\mathbf{e}_c]_{c\in\mathcal{C}} \in \mathbb{R}^{24\times d_e}$, $B = [\mathbf{b}_c]_{c\in\mathcal{C}} \in \mathbb{R}^{24\times d}$, and $G = [\mathbf{g}_c]_{c\in\mathcal{C}} \in \mathbb{R}^{24\times d}$, where $d_e > d$.

To incorporate genomic coordinates, we apply a rotary transformation $R(p, \mathbf{x})$ (Su et al., 2024) to an embedding vector $\mathbf{x}$ according to its genomic position p (details in Appendix C.1). Accordingly, a genomic position p on chromosome c is represented as $R(p, \mathbf{e}_c)$. We then represent a genomic interval by the difference between the chromosome token embeddings rotated at its end and start coordinates. Specifically, we represent each scATAC-seq feature as a genomic interval spanning p_{start} to p_{end} on chromosome c, whose embedding is defined as

$$\mathbf{x}_{c:\text{start-end}} = W_e \left(R(p_{\text{end}}, \mathbf{e}_c) - R(p_{\text{start}}, \mathbf{e}_c) \right), \tag{1}$$

where $W_e \in \mathbb{R}^{d\times d_e}$ is a trainable projection matrix. This construction enables embeddings to be generated for arbitrary genomic intervals according to their chromosomal locations.

However, because the rotary transformation itself is parameter-free, it provides a fixed positional transformation and does not introduce additional trainable modulation for individual genomic regions. Moreover, subtracting two position-dependent embeddings weakens the smooth positional structure. As shown in Fig. A1, rotated chromosome embeddings preserve shared positional structure, whereas subtraction-based interval representations become nearly orthogonal. To increase the expressivity of genomic region representations while retaining genomic positional information, we introduce chromosome gate and bias embeddings. These embeddings are also learned at the chromosome level and transformed according to genomic position, but provide multiplicative and additive position-dependent modulation to the interval representation, respectively. Specifically, the chromosome gate and bias embeddings are defined as

$$\mathbf{g}_{c:\text{mid}} = W_g R(p_{\text{mid}}, \mathbf{g}_c), \qquad \mathbf{b}_{c:\text{mid}} = R(p_{\text{mid}}, \mathbf{b}_c), \tag{2}$$

Table 1: Cell type annotation performance across scATAC-seq datasets. Values are reported as mean ± standard deviation. The best result is highlighted in **bold**, and the second-best result is underlined. * denotes results obtained from a single fold rather than five-fold cross-validation due to the substantial computational cost.

Dataset	Tissue	Metric	scChromHEARTs w corpus PT	scChromHEARTs w/o corpus PT	EpiAgent	ChromFound	CellCano	LSI	SANGO	PeakVI	PoissonVI
Ameen2022	Cardiac	F1	**0.83** ± 0.007	0.80 ± 0.005	0.68 ± 0.005	0.02 ± 0.003	0.69 ± 0.011	0.82 ± 0.009	0.80 ± 0.010	0.81 ± 0.014	0.77 ± 0.016
		Acc	**0.85** ± 0.004	0.84 ± 0.001	0.76 ± 0.003	0.15 ± 0.005	0.74 ± 0.008	**0.85** ± 0.006	0.84 ± 0.007	**0.85** ± 0.009	0.82 ± 0.006
Kanemaru2023	Cardiac	F1	0.73 ± 0.010	0.69 ± 0.003	**0.76** ± 0.017	0.68 *	0.68 ± 0.030	0.70 ± 0.003	0.67 ± 0.003	0.71 ± 0.004	0.72 ± 0.010
		Acc	**0.97** ± 0.001	**0.97** ± 0.001	0.93 ± 0.002	0.91 *	0.93 ± 0.001	**0.97** ± 0.001	0.95 ± 0.002	**0.97** ± 0.001	**0.97** ± 0.001
Granja2019	Bone marrow	F1	0.92 ± 0.011	**0.94** ± 0.010	0.71 ± 0.017	0.88 ± 0.015	0.86 ± 0.007	0.81 ± 0.012	0.80 ± 0.032	0.80 ± 0.041	0.80 ± 0.023
		Acc	0.94 ± 0.001	**0.95** ± 0.002	0.82 ± 0.005	0.90 ± 0.012	0.89 ± 0.006	0.89 ± 0.006	0.90 ± 0.009	0.88 ± 0.009	0.90 ± 0.007
Buenrostro2018	Bone marrow	F1	0.86 ± 0.020	0.88 ± 0.016	0.78 ± 0.032	0.60 ± 0.330	0.76 ± 0.011	**0.89** ± 0.010	0.88 ± 0.021	0.72 ± 0.042	0.85 ± 0.019
		Acc	0.86 ± 0.015	0.88 ± 0.014	0.80 ± 0.021	0.66 ± 0.246	0.82 ± 0.011	**0.90** ± 0.007	**0.90** ± 0.020	0.80 ± 0.018	0.87 ± 0.014
Granja2019	PBMC	F1	**0.93** ± 0.016	**0.93** ± 0.009	0.86 ± 0.051	0.77 ± 0.074	0.86 ± 0.010	0.87 ± 0.020	0.88 ± 0.022	0.86 ± 0.028	0.81 ± 0.026
		Acc	**0.96** ± 0.006	**0.96** ± 0.002	0.93 ± 0.009	0.88 ± 0.038	0.92 ± 0.006	0.92 ± 0.003	0.95 ± 0.010	0.94 ± 0.005	0.91 ± 0.007
Muto2021	Kidney	F1	0.93 ± 0.005	0.94 ± 0.006	0.84 ± 0.013	0.88 ± 0.037	0.91 ± 0.015	**0.95** ± 0.002	0.91 ± 0.033	0.83 ± 0.053	0.91 ± 0.013
		Acc	**0.97** ± 0.001	**0.97** ± 0.003	0.92 ± 0.007	0.95 ± 0.011	0.96 ± 0.004	**0.97** ± 0.002	0.96 ± 0.004	0.95 ± 0.007	0.96 ± 0.004
Li2023	Brain	F1	0.79 ± 0.014	0.58 ± 0.042	**0.81** ± 0.004	0.67 *	0.71 ± 0.007	0.66 ± 0.005	0.60 ± 0.031	0.74 ± 0.030	0.72 ± 0.020
		Acc	**0.94** ± 0.004	0.89 ± 0.012	0.93 ± 0.001	0.90 *	0.90 ± 0.002	0.92 ± 0.002	0.85 ± 0.012	0.92 ± 0.005	**0.93** ± 0.006

where $p_{\text{mid}} = \frac{p_{\text{start}}+p_{\text{end}}}{2}$ and $W_g \in \mathbb{R}^{d\times d}$. The genomic region embedding, $\mathbf{r}$, is then adaptively transformed as

$$\mathbf{r}_{c:\text{start-end}} = \sigma(\mathbf{g}_{c:\text{mid}}) \odot \mathbf{x}_{c:\text{start-end}} + \mathbf{b}_{c:\text{mid}}, \tag{3}$$

where $\sigma(\cdot)$ and $\odot$ denote the sigmoid function and element-wise multiplication, respectively. The chromosome bias embedding provides a position-dependent additive shift, whereas the chromosome gate embedding provides multiplicative modulation of the region representation. Together, these components allow scChromHEARTs to combine a coordinate-aware chromosome token representation with trainable position-dependent modulation (Fig. 1b).

3.2 scChromHEARTs

Encoder scChromHEARTs employs a Transformer encoder (Vaswani et al., 2017) to aggregate genomic region embeddings into a cell-level representation. For each cell, a special `SEP` token is prepended to the sequence of accessible genomic region embeddings, and the output representation corresponding to this token is used as the cell embedding:

$$\mathbf{c} = \text{Encoder}\left([\mathbf{x}_{\text{SEP}}; v_{p_1}\mathbf{r}_{p_1}; v_{p_2}\mathbf{r}_{p_2}; \ldots; v_{p_n}\mathbf{r}_{p_n}]\right)[0], \tag{4}$$

where $\mathbf{x}_{\text{SEP}}$ denotes the embedding of the special `SEP` token, while v_{p_k} and $\mathbf{r}_{p_k}$ denote the accessibility value and embedding of the k-th genomic region, respectively.

Pre-training Strategy To mitigate the effects of scATAC-seq sparsity and learn robust representations from diverse genomic regions, we adopt contrastive learning of scRobust (Park & Lee, 2024), where multiple augmented views of each cell are constructed by randomly sampling subsets of accessible regions (e.g., 500 regions per view). scRobust was originally developed for scRNA-seq and encourages different views of the same cell to produce consistent representations. In scRNA-seq, correlations among genes provide shared signals across different sampled subsets of input genes, facilitating the learning of consistent representations across views. In scATAC-seq, however, greater sparsity and weaker inter-region correlations make such cross-view consistency more difficult to learn using conventional lookup-table embeddings. By incorporating shared structural information across genomic regions, our coordinate-aware embeddings facilitate effective contrastive learning in scATAC-seq.

We jointly optimize a multi-positive contrastive objective and a genomic region accessibility prediction objective (Fig. 1c,d). The contrastive objective treats multiple views of the same cell as positives, encouraging the encoder to recover a consistent representation from different subsets of regions. The accessibility objective predicts region accessibility from the cell representation, helping retain region-level information. The full loss formulations are provided in Appendix C.2.

4 EXPERIMENTS

We first pretrain an 8-layer scChromHEARTs model on 4.59M cells from the Human scATAC Corpus (Chen et al., 2026). For target-dataset adaptation, we initialize a smaller 1- or 2-layer model using the corresponding layers from the pretrained encoder and perform a second stage of pretraining on the target dataset. We use these shallower architectures at this stage to reduce computational cost and training time. The scChromHEARTs encoder is then frozen, and separate task-specific models are trained on the cell embeddings for downstream evaluation. Detailed hyperparameters and dataset descriptions are provided in Appendix C.3 and D.

4.1 DOWNSTREAM TASKS

Cell type annotation For cell type annotation, we compared scChromHEARTs with EpiAgent (Chen et al., 2025), ChromFound (Jiao et al., 2026), CellCano (Ma et al., 2023), SANGO (Zeng et al., 2024), PeakVI (Ashuach et al., 2022), PoissonVI (Martens et al., 2024), and LSI (Deerwester et al., 1990) using five-fold cross-validation. We additionally evaluated scChromHEARTs without human corpus pretraining by randomly initializing the model and pretraining it only on the target dataset, allowing us to assess the contribution of large-scale corpus pretraining. We used a two-layer encoder for all datasets except Buenrostro2018 (Buenrostro et al., 2018), for which a one-layer encoder was used because of its relatively small size (2,034 cells).

For each cell, we generated its embedding using all genomic regions with nonzero accessibility. We used a batch size of one during embedding extraction, allowing the number of input regions to vary naturally across cells according to their accessibility profiles. EpiAgent, ChromFound, and CellCano perform cell-type prediction in an end-to-end manner. In contrast, the remaining methods first generate cell representations independently of the prediction task. For these methods, we used the graph Transformer adopted in SANGO as a common prediction model.

Several commonly used scATAC-seq datasets (Granja et al., 2019; Ameen et al., 2022; Muto et al., 2021) define cell labels based on LSI (Deerwester et al., 1990). ChromFound performs cell aggregation using LSI-derived neighborhoods constructed from all cells in the dataset during preprocessing. Because cells connected in this LSI space are aggregated to have more similar representations, training and test cells can indirectly share information when their neighborhoods overlap. To avoid the leakage between the training, we disabled the cell aggregation procedure in ChromFound.

As shown in Table 1, scChromHEARTs achieved the best or second-best performance on most datasets. Notably, pretraining on the human corpus yielded more consistently strong performance across datasets than pretraining on the target datasets alone. As expected, LSI also showed strong performance and achieved the best results on Muto2021 (Muto et al., 2021) and Buenrostro2018. In contrast, EpiAgent and ChromFound did not consistently achieve high performance across datasets. These results demonstrate that scChromHEARTs generates cell embeddings that effectively capture cell type information across diverse datasets.

Trajectory-guided perturbed cell generation Predicting how an individual cell transitions from a control state to a perturbed state is important for characterizing cell-level regulatory responses to perturbation. However, the same individual cell cannot be observed before and after perturbation, making it impossible to establish ground-truth one-to-one pairs between control and perturbed cells. To address this problem, we generate progressively perturbed cells by iteratively applying a learned perturbation mapping in the latent space of pre-trained scATAC-seq models. Starting from a control-cell embedding, a model predicts a perturbation-induced shift, and the updated embedding is repeatedly used as the input to the next step. This produces a trajectory that gradually transforms the control- toward the perturbed cells. If cell embeddings capture perturbation effects, the final generated embedding should resemble real perturbed cells.

In this experiment, we used the Lareau2019 bone marrow dataset (Lareau et al., 2019) to evaluate perturbation prediction. For each scATAC-seq model, we first generated cell embeddings and then applied a simple neural network to predict perturbation-induced shifts in the embedding space. We compared the two-layer scChromHEARTs with ChromFound, SANGO, PeakVI, PoissonVI, and LSI. To evaluate generalization to unseen cell types, NK and CD8 cells were completely excluded from training the perturbation predictor and used only for testing (details in Appendix E and Fig.

A2). As shown in Figs. 2 and A3, the benchmark methods generated highly mixed cell embeddings across cell types and perturbation stages, without clearly distinguishable cell-type-specific perturbation trajectories. In contrast, scChromHEARTs generated cell-type-specific perturbation trajectories that preserved cell-type identity while progressively transitioning from control to perturbed states. For both unseen NK and CD8 cells, the generated embeddings moved systematically toward the corresponding observed perturbed-cell distributions over successive prediction steps.

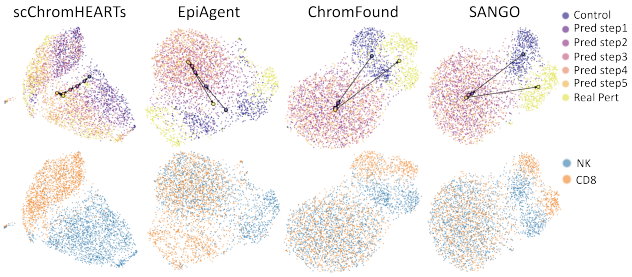

Figure 2: UMAP visualization of cell-type-specific perturbed cell generation along the predicted trajectory.

We evaluate the generated perturbed cell embeddings from two complementary perspectives. First, we evaluate how well the predicted perturbed states recover plausible control–perturbed cell pairs, based on the expectation that an accurately predicted perturbed state should be closest to the real perturbed cells corresponding to that control cell in the latent space. If the inferred pairs are biologically meaningful, they should also reveal a larger number of genomic regions that exhibit consistent perturbation-associated accessibility changes across paired cells. Second, we decode the generated embeddings back into scATAC-seq accessibility profiles and compare them with those reconstructed from real perturbed cell embeddings. This allows us to assess whether the generated embeddings reproduce the chromatin accessibility changes induced by perturbation.

We examined whether these predicted trajectories could provide informative control–perturbed cell pairing (details in Appendix E.6). For each method, we constructed control and perturbed meta-cell pairs using three pairing strategies: trajectory-based, random, and OT-based pairing. To evaluate the quality of the inferred pairs, we counted the peaks showing consistent perturbation-associated accessibility changes across the resulting pairs (Table A4). scChromHEARTs identified 34 consistent perturbation-responsive peaks with trajectory-based pairing, compared with only 18 and 19 peaks using random and OT-based pairing, respectively. In contrast, the benchmark methods showed no comparable improvement with trajectory-based pairing and generally identified only around 20 or fewer consistent perturbation-responsive peaks across all pairing strategies. Thus, using the trajectories predicted by scChromHEARTs enables the construction of more plausible control–perturbed cell pairs than the benchmark methods.

Table 2: Pearson delta correlations between real and predicted peak-wise accessibility changes for CD8 and NK cells.

PCC	scChromHEARTs	EpiAgent	ChromFound	LSI	SANGO	PeakVI	PoissonVI
CD8	**0.610**	-0.279	-0.073	0.212	-0.184	-0.046	-0.075
NK	**0.574**	-0.410	-0.063	0.236	-0.306	-0.064	-0.020

To assess whether the predicted perturbed embeddings provide reliable representations of the underlying chromatin accessibility state, we decoded the cell embeddings back into scATAC-seq accessibility profiles (details in Appendix E.8). For each scATAC model, a simple decoder was trained using only the training cell types to reconstruct the accessibility of 2,000 genomic peaks from the cell embeddings. We then decoded the control, intermediate, and real perturbed cells for the held-out CD8 and NK cell types. First, we compared the decoded accessibility profiles of each trajectory state with those of the real perturbed cells using Euclidean distance, energy distance, and maximum mean discrepancy (MMD), implemented with the Pertpy library (Heumos et al., 2026) (Fig. A4). Second, we computed peak-wise accessibility changes across the 2,000 peaks from the decoded values relative to the control cell embeddings, and measured the Pearson delta correlation between the changes obtained from real and generated perturbed-cell embeddings (Table 2 and Fig. A5). Overall, only scChromHEARTs produced reasonable and consistent results across both validation analyses, while the benchmark methods generally showed weak or inconsistent correspondence with the real perturbation.

The superior performance of scChromHEARTs across these analyses can be explained by the quality of its predicted perturbation trajectories. As shown in Figs. 2, scChromHEARTs was the only model that generated coherent trajectories in which the generated cells progressively moved from the control cells toward the real perturbed cells. In Fig. A4, the decoded scATAC-seq profiles of

scChromHEARTs also became progressively closer to those of the real perturbed cells across successive trajectory steps, whereas the benchmark methods generally showed little convergence or even increasing distributional distances. Consequently, the perturbed cells generated by scChromHEARTs showed substantially higher Pearson delta correlations with real perturbed cells for decoded peak-wise accessibility changes. Together, these results indicate that scChromHEARTs generates perturbed cell representations that closely resemble those of real perturbed cells.

Batch effect correction We further evaluated the quality of the learned cell representations by assessing whether they preserve biologically meaningful cell-type structure while minimizing technical variation associated with experimental batches. This analysis was performed on GSE194122 Luecken et al. (2021) and GSE204682 Zhu et al. (2023). As shown in Fig. A6 and Table A3, most methods formed distinguishable cell-type clusters, indicating that cell-type-related biological variation was broadly retained. We then quantified the extent to which the same representations were influenced by batch effects by comparing Leiden cluster assignments with batch labels using the Adjusted Rand Index (ARI) and Normalized Mutual Information (NMI). Lower ARI and NMI values indicate weaker alignment between the embedding structure and batch identity, and therefore less batch-associated variation. For each method, we report the maximum ARI and NMI across Leiden clustering resolutions from 0.1 to 2.0 to conservatively capture the strongest batch-associated structure. For all datasets, we used a one-layer encoder and randomly sampled 2,000 genomic regions per cell to generate cell embeddings.

As shown in Table 3, conventional methods, including LSI, SANGO, PeakVI, and PoissonVI, generally showed higher batch-based ARI and NMI scores than foundation models (EpiAgent and ChromFound). scChromHEARTs achieved comparable results to the foundation models. Together, these results suggest that scChromHEARTs preserves cell-type structure while reducing batch-associated variation compared with conventional methods.

Table 3: Batch clustering performance (ARI / NMI; lower is better). Best results are shown in **bold**, and second-best results are underlined.

Dataset	Metric	Ours	EpiAgent	ChromFound	LSI	SANGO	PeakVI	PoissonVI
GSE194122	ARI ↓	0.105	**0.044**	0.107	0.103	0.107	0.137	0.127
	NMI ↓	0.144	**0.135**	0.208	0.240	0.230	0.241	0.250
GSE204682	ARI ↓	0.229	0.230	**0.226**	0.280	0.235	0.279	0.287
	NMI ↓	**0.374**	0.397	0.388	0.512	0.456	0.493	0.504

Cross-dataset Transferability Because scChromHEARTs is primarily trained with contrastive learning, its representations may preferentially capture relative differences among cells within a target dataset. We therefore investigated whether our model also learns biological structure that generalizes across datasets from the same tissue. Successful transfer to an unseen dataset without target-specific adaptation would suggest that scChromHEARTs captures biological information shared across datasets, rather than only dataset-specific differences among cells.

To evaluate cross-dataset transferability, we applied scChromHEARTs to unseen datasets from the same tissue without any additional training on the evaluation datasets. We compared its performance with EpiAgent, which requires an additional training stage to obtain cell embeddings. In this evaluation setting, EpiAgent was initialized from its model pre-trained on large-scale human scATAC-seq data and subsequently trained on the designated tissue-matched pre-training dataset before being applied to the unseen evaluation datasets. In contrast, scChromHEARTs was trained only on the designated tissue-matched pre-training dataset without using weights pre-trained on the human corpus. We deliberately excluded the human corpus pre-trained weights for scChromHEARTs to avoid any potential data leakage.

Table 4: Cross-dataset clustering performance (ARI / NMI) on unseen scATAC-seq datasets. Best results are shown in **bold**.

Dataset	Tissue	scChromHEARTs	EpiAgent
Ameen2022	Cardiac	0.329 / 0.464	**0.508 / 0.638**
Hocker2021		**0.897 / 0.864**	0.768 / 0.820
Zhang2021		**0.805 / 0.822**	0.740 / 0.796
Buenrostro2018	Bone marrow	**0.456 / 0.629**	0.403 / 0.599
Lareau2019		**0.625 / 0.741**	**0.625** / 0.667

For cardiac tissue, Kanemaru2023 (Kanemaru et al., 2023) was used for pre-training, and Ameen2022 (Ameen et al., 2022), Hocker2021 (Hocker et al., 2021), and Zhang2021 (Zhang et al., 2021) were used for the unseen dataset transferability. For bone marrow, Granja2019 (Granja et al., 2019) was used as the pre-training dataset, while Buenrostro2018 (Buenrostro et al., 2018) and Lareau2019 (Lareau et al., 2019) were used as unseen evaluation datasets. We selected the best ARI and NMI scores across Leiden clustering resolutions ranging from 0.1 to 2.0. As shown in Table 4,

scChromHEARTs achieved better performance than EpiAgent on all datasets except Ameen2022, despite being trained exclusively on the designated tissue-matched dataset without the human corpus pre-training. These results suggest that scChromHEARTs learns generalizable genomic representations that capture biological structure shared across datasets within the same tissue.

4.2 Ablation Study

Comparison of genomic region embedding designs scChromHEARTs represents arbitrary genomic regions using chromosome token, bias, and gate embeddings combined with the rotary transformation (Su et al., 2024). To evaluate the effectiveness of our embedding design, we compared it with ablated variants as well as alternative region embedding strategies. Using cell type annotation as the evaluation task, we compare six embedding variants: (1) using chromosome token, bias, and gate embeddings; (2) using chromosome token and bias embeddings; (3) using chromosome token and gate embeddings; (4) using chromosome token embeddings only; (5) a conventional feature-specific embedding lookup table; and (6) the embedding scheme used in ChromFound (see Appendix B for details). For a fair comparison, all variants use the same 2-layer Transformer encoder and are pre-trained from random initialization using 500 randomly sampled accessible genomic regions per cell. For downstream cell type annotation, we intentionally restrict the input to 2,000 randomly sampled accessible genomic regions per cell, creating a more challenging evaluation setting. This limited-input setting is designed to assess the effectiveness of contrastive pre-training in learning informative cell representations.

As shown in Figs. 3 and A7, the full model using chromosome token, bias, and gate embeddings achieved the best performance on most datasets, except for Ameen2022. The variant using chromosome token and bias embeddings showed comparable performance in several datasets. Interestingly, on Ameen2022, variants including the chromosome bias embedding performed worse than those without it. One possible explanation is that the bias embedding leads the model to overfit the contrastive objective, primarily increasing inter-sample separation rather than preserving biologically meaningful relationships among cells. In contrast, incorporating the gate embedding alleviated this degradation and yielded more stable performance across datasets. These results suggest that the bias embedding is the primary component for constructing effective genomic region representations, while the gate embedding further refines these representations and improves robustness across datasets.

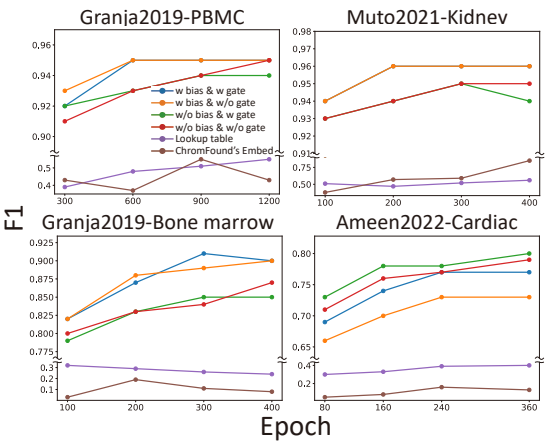

Figure 3: Ablation of genomic region embedding components based on F1 scores across four scATAC-seq datasets.

The conventional embedding lookup table and the ChromFound embedding scheme showed substantially lower performance than the proposed embedding variants. As shown in Fig. A8, the test contrastive loss of the lookup-table and ChromFound scheme exhibited little to no decrease during pre-training, in contrast to the scChromHEARTs variants. These results suggest that the lookup-based and ChromFound embedding schemes are less amenable to optimization with the contrastive learning objective on sparse scATAC-seq data. We attribute this difference, at least in part, to the structured genomic representation introduced by scChromHEARTs. Specifically, scChromHEARTs transforms chromosome token embeddings according to genomic coordinates using the rotary transformation. This incorporates region-specific positional information while preserving the shared chromosome-specific structure. In addition, the chromosome bias embedding introduces position-dependent shifts that encourage nearby genomic regions to occupy similar locations in the embedding space, while the gate embedding adaptively modulates each region representation according to its genomic position. Together, these components explicitly incorporate chromosome identity and genomic position into the region embeddings, providing structured genomic priors be-

fore large-scale pre-training. This structured representation enables scChromHEARTs to generate high-quality cell embeddings, even under the extreme sparsity of scATAC-seq data.

Structural comparison of genomic region embeddings. To examine how different embedding designs organize genomic regions, we visualized the region embeddings using UMAP before and after pre-training on Granja2019-PBMC. Before pre-training (Fig. A9), genomic intervals were embedded by each randomly initialized region embedder. scChromHEARTs variants using the bias embedding showed local grouping only among closely located genomic regions, while the overall positional structure remained fragmented and disconnected across longer genomic distances. This fragmentation was more pronounced in the bias–gate variant than in the bias-only variant. After pre-training (Fig. 4), the previously fragmented positional organization became substantially more continuous across longer genomic distances. In particular, the bias-gate variant, which showed stronger fragmentation before pre-training, formed a more compact and coherent positional structure than the bias-only variant.

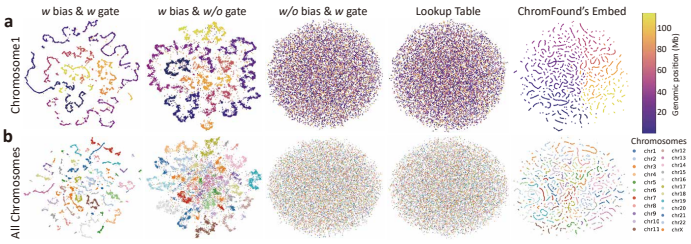

Figure 4: UMAP visualization of genomic region embeddings under different embedding designs after pre-training on Granja2019-PBMC: (a) chromosome 1 embeddings colored by genomic position and (b) embeddings from all chromosomes colored by chromosome identity.

In contrast, the overall structure of the ChromFound embeddings changed relatively little before and after pre-training. Although nearby genomic regions showed similar embeddings, this similarity gradually weakened with increasing genomic distance. Thus, the similarity between nearby regions appears to arise mainly from the sinusoidal encoding of their coordinates, rather than from broader chromosome-level organization. ChromFound also showed little apparent chromosome-specific clustering, either before or after pre-training. This may indicate that its embedding formulation provides a relatively weak inductive structure for preserving chromosome-level information. Because ChromFound independently constructs chromosome and positional embeddings and then combines them additively, its initial representation may therefore provide a weaker structural basis.

Interestingly, scChromHEARTs variants without the bias embedding showed little visually apparent similarity among genomic region embeddings, yet achieved more stable optimization and better downstream performance than the ChromFound embedding scheme (Fig. A8; Figs. 3 and A7). This suggests that visual proximity among region embeddings does not necessarily reflect how effectively they capture informative structure from scATAC-seq data. Even without the bias embedding, the combination of chromosome tokens and rotary transformations may preserve relational information that can be effectively utilized by the encoder during pre-training.

5 Conclusion

scChromHEARTs introduces flexible genomic region embeddings that explicitly encode chromosome identity and genomic coordinates. This design enables diverse genomic regions to be represented with a lightweight Transformer architecture while providing shared genomic structure that facilitates effective representation learning from sparse scATAC-seq data. By combining contrastive learning with region-subset sampling, scChromHEARTs can learn from a broad range of accessible genomic regions throughout pre-training without processing all regions simultaneously. Although contrastive learning enables efficient subset-based pre-training, scChromHEARTs generally requires target-specific pre-training to obtain well-adapted representations. This limits its direct use as a fully general-purpose foundation model on unseen datasets. Future work could explore more transferable pre-training objectives that retain the efficiency of subset-based learning while reducing or eliminating the need for target-specific adaptation.

AI USE STATEMENT

In this work, we used generative AI tools to assist with language editing and improving the clarity and readability of the manuscript. Generative AI tools were also used to assist with code debugging and refinement. All AI-assisted content was reviewed and revised by the authors. The authors verified the technical correctness of the manuscript, experimental procedures, and results, and take full responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work uses previously published and publicly available single-cell chromatin accessibility datasets. We did not collect new human-subject data or conduct additional experiments involving human participants. All datasets were used in accordance with their original terms of access and the ethical approvals reported in the corresponding studies. Our study focuses on representation learning for single-cell epigenomic data and does not introduce additional privacy or safety risks beyond those associated with the use of the original datasets.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the model architecture, training procedure, hyperparameters, datasets, data preprocessing, and evaluation protocols in the main text and Appendix. Source code and configuration files are available at an anonymous repository. These materials are intended to facilitate reproduction of the experimental results reported in this work.

REFERENCES

Mohamed Ameen, Laksshman Sundaram, Mengcheng Shen, Abhimanyu Banerjee, Soumya Kundu, Surag Nair, Anna Shcherbina, Mingxia Gu, Kitchener D Wilson, Avyay Varadarajan, et al. Integrative single-cell analysis of cardiogenesis identifies developmental trajectories and non-coding mutations in congenital heart disease. *Cell*, 185(26):4937–4953, 2022.

Tal Ashuach, Daniel A Reidenbach, Adam Gayoso, and Nir Yosef. Peakvi: A deep generative model for single-cell chromatin accessibility analysis. *Cell reports methods*, 2(3), 2022.

Carmen Bravo González-Blas, Liesbeth Minnoye, Dafni Papasokrati, Sara Aibar, Gert Hulselmans, Valerie Christiaens, Kristofer Davie, Jasper Wouters, and Stein Aerts. cistopic: cis-regulatory topic modeling on single-cell atac-seq data. *Nature methods*, 16(5):397–400, 2019.

Jason D Buenrostro, M Ryan Corces, Caleb A Lareau, Beijing Wu, Alicia N Schep, Martin J Aryee, Ravindra Majeti, Howard Y Chang, and William J Greenleaf. Integrated single-cell analysis maps the continuous regulatory landscape of human hematopoietic differentiation. *Cell*, 173(6):1535–1548, 2018.

Xiaoyang Chen, Keyi Li, Xuejian Cui, Zian Wang, Qun Jiang, Jiacheng Lin, Zhen Li, Zijing Gao, Hairong Lv, and Rui Jiang. Epiagent: foundation model for single-cell epigenomics. *Nature Methods*, 22(11):2316–2327, 2025.

Xiaoyang Chen, Zijing Gao, Keyi Li, Zian Wang, Qun Jiang, Xuejian Cui, Zhen Li, and Rui Jiang. Human-scatac-corpus: a comprehensive database of scatac-seq data. *Nucleic Acids Research*, 54 (D1):D175–D183, 2026.

Haotian Cui, Chloe Wang, Hassaan Maan, Kuan Pang, Fengning Luo, Nan Duan, and Bo Wang. scgpt: toward building a foundation model for single-cell multi-omics using generative ai. *Nature methods*, 21(8):1470–1480, 2024.

Darren A Cusanovich, James P Reddington, David A Garfield, Riza M Daza, Delasa Aghamirzaie, Raquel Marco-Ferreres, Hannah A Pliner, Lena Christiansen, Xiaojie Qiu, Frank J Steemers, et al. The cis-regulatory dynamics of embryonic development at single-cell resolution. *Nature*, 555(7697):538–542, 2018.

Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41 (6):391–407, 1990.

Rongxin Fang, Sebastian Preissl, Yang Li, Xiaomeng Hou, Jacinta Lucero, Xinxin Wang, Amir Motamedi, Andrew K Shiau, Xinzhu Zhou, Fangming Xie, et al. Comprehensive analysis of single cell atac-seq data with snapatac. *Nature communications*, 12(1):1337, 2021.

Jeffrey M Granja, Sandy Klemm, Lisa M McGinnis, Arwa S Kathiria, Anja Mezger, M Ryan Corces, Benjamin Parks, Eric Gars, Michaela Liedtke, Grace XY Zheng, et al. Single-cell multiomic analysis identifies regulatory programs in mixed-phenotype acute leukemia. *Nature biotechnology*, 37 (12):1458–1465, 2019.

Jeffrey M Granja, M Ryan Corces, Sarah E Pierce, S Tansu Bagdatli, Hani Choudhry, Howard Y Chang, and William J Greenleaf. Archr is a scalable software package for integrative single-cell chromatin accessibility analysis. *Nature genetics*, 53(3):403–411, 2021.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Minsheng Hao, Jing Gong, Xin Zeng, Chiming Liu, Yucheng Guo, Xingyi Cheng, Taifeng Wang, Jianzhu Ma, Xuegong Zhang, and Le Song. Large-scale foundation model on single-cell transcriptomics. *Nature methods*, 21(8):1481–1491, 2024.

Lukas Heumos, Yuge Ji, Lilly May, Tessa D Green, Stefan Peidli, Xinyue Zhang, Xichen Wu, Johannes Ostner, Antonia Schumacher, Karin Hrovatin, et al. Pertpy: an end-to-end framework for perturbation analysis. *Nature methods*, 23(2):350–359, 2026.

James D Hocker, Olivier B Poirion, Fugui Zhu, Justin Buchanan, Kai Zhang, Joshua Chiou, Tsui-Min Wang, Qingquan Zhang, Xiaomeng Hou, Yang E Li, et al. Cardiac cell type–specific gene regulatory programs and disease risk association. *Science advances*, 7(20):eabf1444, 2021.

Yifeng Jiao, Yuchen Liu, Yu Zhang, Xin Guo, Yushuai Wu, Chen Jiang, Jiyang Li, Hongwei Zhang, Limei Han, Xin Gao, et al. Chromfound: Towards a universal foundation model for single-cell chromatin accessibiltiy data. *Advances in Neural Information Processing Systems*, 38:76986–77023, 2026.

Kazumasa Kanemaru, James Cranley, Daniele Muraro, Antonio MA Miranda, Siew Yen Ho, Anna Wilbrey-Clark, Jan Patrick Pett, Krzysztof Polanski, Laura Richardson, Monika Litvinukova, et al. Spatially resolved multiomics of human cardiac niches. *Nature*, 619(7971):801–810, 2023.

Sandy L. Klemm, Zohar Shipony, and William J. Greenleaf. Chromatin accessibility and the regulatory epigenome. *Nature Reviews Genetics*, 20(4):207–220, 2019. doi: 10.1038/s41576-018-0089-8. URL https://doi.org/10.1038/s41576-018-0089-8.

Caleb A Lareau, Fabiana M Duarte, Jennifer G Chew, Vinay K Kartha, Zach D Burkett, Andrew S Kohlway, Dmitry Pokholok, Martin J Aryee, Frank J Steemers, Ronald Lebofsky, et al. Droplet-based combinatorial indexing for massive-scale single-cell chromatin accessibility. *Nature biotechnology*, 37(8):916–924, 2019.

Junhao Li, Manoj K Jaiswal, Jo-Fan Chien, Alexey Kozlenkov, Jinyoung Jung, Ping Zhou, Mahammad Gardashli, Luc J Pregent, Erica Engelberg-Cook, Dennis W Dickson, et al. Divergent single cell transcriptome and epigenome alterations in als and ftd patients with c9orf72 mutation. *Nature communications*, 14(1):5714, 2023.

Malte D Luecken, Daniel Bernard Burkhardt, Robrecht Cannoodt, Christopher Lance, Aditi Agrawal, Hananeh Aliee, Ann T Chen, Louise Deconinck, Angela M Detweiler, Alejandro A Granados, et al. A sandbox for prediction and integration of dna, rna, and proteins in single cells. In *Thirty-fifth conference on neural information processing systems datasets and benchmarks track (Round 2)*, 2021.

Wenjing Ma, Jiaying Lu, and Hao Wu. Cellcano: supervised cell type identification for single cell atac-seq data. *Nature communications*, 14(1):1864, 2023.

Laura D Martens, David S Fischer, Vicente A Yépez, Fabian J Theis, and Julien Gagneur. Modeling fragment counts improves single-cell atac-seq analysis. *Nature Methods*, 21(1):28–31, 2024.

Liesbeth Minnoye, Georgi K. Marinov, Thomas Krausgruber, Lixia Pan, Alexandre P. Marand, Stefano Secchia, William J. Greenleaf, Eileen E. M. Furlong, Keji Zhao, Robert J. Schmitz, Christoph Bock, and Stein Aerts. Chromatin accessibility profiling methods. *Nature Reviews Methods Primers*, 1(1):10, 2021. doi: 10.1038/s43586-020-00008-9. URL https://doi.org/10.1038/s43586-020-00008-9.

Yoshiharu Muto, Parker C Wilson, Nicolas Ledru, Haojia Wu, Henrik Dimke, Sushrut S Waikar, and Benjamin D Humphreys. Single cell transcriptional and chromatin accessibility profiling redefine cellular heterogeneity in the adult human kidney. *Nature communications*, 12(1):2190, 2021.

Sejin Park and Hyunju Lee. Robust self-supervised learning strategy to tackle the inherent sparsity in single-cell rna-seq data. *Briefings in Bioinformatics*, 25(6):bbae586, 2024.

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

Tim Stuart, Avi Srivastava, Shaista Madad, Caleb A Lareau, and Rahul Satija. Single-cell chromatin state analysis with signac. *Nature methods*, 18(11):1333–1341, 2021.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Nadezhda V Terekhanova, Alla Karpova, Wen-Wei Liang, Alexander Strzalkowski, Siqi Chen, Yize Li, Austin N Southard-Smith, Michael D Iglesia, Michael C Wendl, Reyka G Jayasinghe, et al. Epigenetic regulation during cancer transitions across 11 tumour types. *Nature*, 623(7986):432–441, 2023.

Christina V Theodoris, Ling Xiao, Anant Chopra, Mark D Chaffin, Zeina R Al Sayed, Matthew C Hill, Helene Mantineo, Elizabeth M Brydon, Zexian Zeng, X Shirley Liu, et al. Transfer learning enables predictions in network biology. *Nature*, 618(7965):616–624, 2023.

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

F Alexander Wolf, Philipp Angerer, and Fabian J Theis. Scanpy: large-scale single-cell gene expression data analysis. *Genome biology*, 19(1):15, 2018.

Lei Xiong, Kui Xu, Kang Tian, Yanqiu Shao, Lei Tang, Ge Gao, Michael Zhang, Tao Jiang, and Qiangfeng Cliff Zhang. Scale method for single-cell atac-seq analysis via latent feature extraction. *Nature communications*, 10(1):4576, 2019.

Han Yuan and David R Kelley. scbasset: sequence-based modeling of single-cell atac-seq using convolutional neural networks. *Nature Methods*, 19(9):1088–1096, 2022.

Yuansong Zeng, Mai Luo, Ningyuan Shangguan, Peiyu Shi, Junxi Feng, Jin Xu, Ken Chen, Yutong Lu, Weijiang Yu, and Yuedong Yang. Deciphering cell types by integrating scatac-seq data with genome sequences. *Nature Computational Science*, 4(4):285–298, 2024. doi: 10.1038/s43588-024-00622-7. URL https://doi.org/10.1038/s43588-024-00622-7.

Kai Zhang, James D Hocker, Michael Miller, Xiaomeng Hou, Joshua Chiou, Olivier B Poirion, Yunjiang Qiu, Yang E Li, Kyle J Gaulton, Allen Wang, et al. A single-cell atlas of chromatin accessibility in the human genome. *Cell*, 184(24):5985–6001, 2021.

Kaiyi Zhu, Jaroslav Bendl, Samir Rahman, James M Vicari, Claire Coleman, Tereza Clarence, Ovaun Latouche, Nadejda M Tsankova, Aiqun Li, Kristen J Brennand, et al. Multi-omic profiling of the developing human cerebral cortex at the single-cell level. *Science advances*, 9(41): eadg3754, 2023.

A Detailed related work

Conventional scATAC-seq representations Early scATAC-seq analysis pipelines commonly relied on latent semantic indexing (LSI), which applies a term frequency–inverse document frequency (TF–IDF) transformation to the cell-by-region accessibility matrix followed by singular value decomposition (SVD) (Cusanovich et al., 2018; Stuart et al., 2021; Granja et al., 2021). This approach provides a computationally efficient low-dimensional representation of cells and remains widely used in scATAC-seq analysis pipelines. However, because LSI is based on a linear factorization of the accessibility matrix, it does not explicitly model nonlinear relationships among genomic regions or incorporate their genomic coordinates.

Other early approaches introduced alternative representations of chromatin accessibility. cisTopic (Bravo González-Blas et al., 2019) models cells as mixtures of latent regulatory topics, where each topic captures a set of genomic regions that tend to exhibit coordinated accessibility patterns. Cell representations are consequently defined by their topic proportions. SnapATAC (Fang et al., 2021), in contrast, constructs cell representations based on genome-wide similarities between accessibility profiles. By binning the genome and quantifying pairwise cell similarities, SnapATAC obtains a cell–cell similarity matrix that can subsequently be reduced to a lower-dimensional space. These approaches provide effective representations for downstream analysis but learn primarily from accessibility patterns within the target dataset.

Deep learning-based representations Deep learning methods have been introduced to capture nonlinear structure in sparse scATAC-seq data. SCALE (Xiong et al., 2019) uses a deep generative framework to learn low-dimensional latent variables from binarized chromatin accessibility profiles. Its probabilistic formulation is designed to account for the sparsity and discrete nature of scATAC-seq measurements while providing latent representations for downstream tasks such as clustering.

PeakVI (Ashuach et al., 2022) similarly adopts a probabilistic latent-variable model, but explicitly models the probability that each genomic region is accessible in each cell. The model separates cell-specific latent biological variation from region-specific effects and learns a compact latent representation directly from the accessibility matrix. PoissonVI (Martens et al., 2024) provides an alternative probabilistic formulation based on count distributions and also learns low-dimensional representations from the target scATAC-seq dataset. Because these models are trained directly on each target dataset, the resulting representation space is specific to the dataset on which the model is fitted.

scBasset (Yuan & Kelley, 2022) takes a different approach by explicitly incorporating the underlying DNA sequence of accessible regions. It uses convolutional neural networks to learn sequence features associated with chromatin accessibility and predicts cell-specific accessibility from genomic sequence. This allows the model to capture regulatory sequence patterns that are not directly encoded in conventional cell-by-region accessibility matrices. However, its primary representation is learned through sequence-to-accessibility prediction rather than through modeling each cell as a sequence of accessible genomic-region tokens.

Supervised representation learning Several methods learn scATAC-seq representations primarily for supervised cell-type annotation. CellCano (Ma et al., 2023) learns latent features from chromatin accessibility profiles and uses them to distinguish cell types across scATAC-seq datasets. SANGO (Zeng et al., 2024) similarly learns cell representations optimized for cell-type prediction. In contrast to unsupervised or self-supervised representation learning approaches, these methods exploit cell-type labels during training, encouraging their latent spaces to emphasize cell-type-discriminative information. Although such representations can also be extracted and used for downstream analyses, their training objectives are primarily designed around supervised annotation.

Large-scale pre-trained scATAC-seq models EpiAgent (Chen et al., 2025) represents accessible cis-regulatory elements (cCREs) as discrete tokens defined by a fixed genomic-region vocabulary. A Transformer encoder (Vaswani et al., 2017) processes the resulting sequence, and a dedicated cell-level token is used to aggregate information across accessible regions. EpiAgent is pre-trained on approximately 5 million cells. To obtain representations for a target dataset, the pre-trained model is further adapted to that dataset before cell embeddings are extracted. This strategy allows the model

to transfer information learned from a large-scale corpus while adapting the representation space to the accessibility patterns of the target dataset.

ChromFound (Jiao et al., 2026) avoids a fixed genomic-region vocabulary by using genome-aware tokenization. Each genomic region is represented using its chromosome identity, genomic coordinates, and accessibility information and is processed with a Mamba-based sequence model (Gu & Dao, 2023). ChromFound is pre-trained on approximately 1.97 million cells. For downstream cell representation, its region-level encoder outputs are reduced to scalar values, concatenated across genomic regions, and subsequently projected to a low-dimensional space using PCA. This design enables the model to encode arbitrary genomic coordinates without requiring all possible regions to be enumerated in a predefined vocabulary.

B CHROMFOUND GENOMIC REGION EMBEDDING SCHEME

ChromFound Jiao et al. (2026) represents each genomic region by combining chromosome identity, genomic coordinates, and accessibility information. For genomic region j, the chromosome identity c_j is represented using a learnable chromosome embedding matrix $W_c \in \mathbb{R}^{|\mathcal{C}| \times d_{\text{model}}}$ as

$$\mathbf{e}_j^c = W_c[c_j, :], \tag{5}$$

where $|\mathcal{C}| = 25$, consisting of 24 human chromosomes and one padding token. The start and end coordinates of the genomic region, denoted by p_j^s and p_j^e, respectively, are independently encoded using sinusoidal positional embeddings. Specifically, for each sine–cosine dimension pair indexed by k,

$$\begin{aligned}
\left[\mathbf{e}_j^s\right]_{2k} &= \sin\left(\frac{p_j^s}{\text{temp} \cdot 10000^{2k/d_{\text{model}}}}\right), \quad \left[\mathbf{e}_j^s\right]_{2k+1} = \cos\left(\frac{p_j^s}{\text{temp} \cdot 10000^{2k/d_{\text{model}}}}\right), \\
\left[\mathbf{e}_j^e\right]_{2k} &= \sin\left(\frac{p_j^e}{\text{temp} \cdot 10000^{2k/d_{\text{model}}}}\right), \quad \left[\mathbf{e}_j^e\right]_{2k+1} = \cos\left(\frac{p_j^e}{\text{temp} \cdot 10000^{2k/d_{\text{model}}}}\right),
\end{aligned} \tag{6}$$

where $\text{temp} = 100000$, d_{model} is the embedding dimension, and $k = 0, \ldots, d_{\text{model}}/2 - 1$. The genomic coordinates are defined on the GRCh38 reference genome. ChromFound additionally projects the continuous accessibility value x_j into the embedding space using a linear transformation:

$$\mathbf{e}_j^a = \text{Linear}(x_j). \tag{7}$$

The final genomic region representation is obtained by summing the chromosome, start-position, end-position, and accessibility embeddings:

$$\mathbf{e}_j^{\text{region}} = \mathbf{e}_j^c + \mathbf{e}_j^s + \mathbf{e}_j^e + \mathbf{e}_j^a. \tag{8}$$

C DETAILS OF SCCHROMHEARTS

C.1 ROTARY TRANSFORMATION

Since the chromosome token embedding provides the main representation of a genomic region, we use a larger dimension for d_e than for the bias and gate embeddings, d_b and d_g. Given a genomic coordinate p and an embedding vector $\mathbf{x} \in \mathbb{R}^d$, the rotary transformation $R(p, \mathbf{x})$ is defined as

$$\begin{aligned}
\omega_i &= \beta^{-\frac{2i}{d}}, \qquad i = 0, \ldots, \frac{d}{2} - 1, \\
r_{2i} &= x_{2i} \cos(p\omega_i) - x_{2i+1} \sin(p\omega_i), \\
r_{2i+1} &= x_{2i} \sin(p\omega_i) + x_{2i+1} \cos(p\omega_i), \\
R(p, \mathbf{x}) &= [r_0, r_1, \ldots, r_{d-2}, r_{d-1}] \in \mathbb{R}^d,
\end{aligned} \tag{9}$$

where β denotes the base frequency of the rotary transformation, ω_i denotes the frequency associated with the i-th dimension pair, and $i = 0, \ldots, \frac{d}{2} - 1$ indexes the paired embedding dimensions. Using this transformation, genomic coordinates can be incorporated into the chromosome token embedding.

C.2 Loss functions

Let ℓ_i denote the identity of the original cell from which the i-th view is generated. The normalized contrastive representation and multi-positive contrastive loss are defined as

$$\begin{aligned} \mathbf{z}_i &= \frac{f_{\mathrm{CL}}(\mathbf{c}_i)}{\|f_{\mathrm{CL}}(\mathbf{c}_i)\|_2}, \qquad \mathcal{P}(i) = \{p \neq i \mid \ell_p = \ell_i\}, \\ \mathcal{L}_{\text{MP-NCE}} &= -\frac{1}{BN}\sum_{i=1}^{BN} \log \frac{\sum_{p\in\mathcal{P}(i)} \exp(\mathbf{z}_i^\top \mathbf{z}_p/\tau)}{\sum_{\substack{a=1 \\ a\neq i}}^{BN} \exp(\mathbf{z}_i^\top \mathbf{z}_a/\tau)}, \end{aligned} \tag{10}$$

where f_{CL} is a learnable projection layer, B is the batch size, N is the number of views per cell, and τ is the temperature parameter. We additionally predict genomic region accessibility from the cell representation. The prediction embedding, accessibility probability, and corresponding loss are defined as

$$\begin{aligned} \mathbf{z}_i^{\text{open}} &= \frac{f_{\text{open}}(\mathbf{c}_i)}{\|f_{\text{open}}(\mathbf{c}_i)\|_2}, \qquad \hat{p}_{ij} = \sigma\left((\mathbf{z}_i^{\text{open}})^\top \mathbf{r}_j\right), \\ \mathcal{L}_{\text{Access}} &= -\frac{1}{BM}\sum_{i=1}^{B}\sum_{j=1}^{M} \left[w_j y_{ij} \log \hat{p}_{ij} + (1-y_{ij}) \log(1-\hat{p}_{ij})\right], \end{aligned} \tag{11}$$

where $w_j = N_j^- / N_j^+$ compensates for class imbalance. The final pre-training objective is

$$\mathcal{L}_{\text{Total}} = \mathcal{L}_{\text{MP-NCE}} + \mathcal{L}_{\text{Access}}. \tag{12}$$

C.3 Hyperparameters

The hyperparameters used for target-dataset pretraining of scChromHEARTs are summarized in Table A1.

D Description of datasets

Table A2 summarizes the datasets used in each evaluation, together with their tissue origin, number of annotated cell types, and number of cells.

D.1 Data pre-processing

To reduce computational cost and processing time, we selected the 200,000 genomic regions with the highest number of accessible cells in each dataset. Accessibility values were normalized using `pp.normalize_total` from Scanpy Wolf et al. (2018) with `target_sum=1`. We used this normalization instead of term frequency–inverse document frequency (TF-IDF) because the inverse document frequency term depends on feature frequencies within each dataset, resulting in dataset-dependent transformations. In contrast, per-cell total normalization does not rely on dataset-level feature statistics, making it more suitable for evaluating representations across datasets with different genomic region sets.

E Trajectory-guided cell generation

Single-cell perturbation data provide separate populations of control and perturbed cells, but the same individual cell cannot be observed before and after perturbation. As a result, the cell-level transition from a control state to its perturbed state cannot be directly observed. To model this transition, we predict how each control cell moves through the cell-embedding space toward the perturbed state. Starting from a control-cell embedding, a perturbation predictor estimates a perturbation-induced shift, and the updated embedding is iteratively used to generate a sequence of progressively perturbed cell states. If the underlying representation captures perturbation-associated biological variation, the generated trajectory should progressively approach the distribution of real perturbed cells while preserving cell-type identity.

Based on this framework, we evaluate whether cell embeddings generated by different scATAC-seq models preserve perturbation-induced cell-state changes. For each model, control and perturbed cell embeddings are used as fixed inputs to the same perturbation prediction framework. We validate the predicted trajectories through trajectory-guided cell pairing and comparison of decoded accessibility profiles with real perturbed cells. The overall procedure is summarized in Fig. A2.

E.1 DATASET

We performed this evaluation using the Lareau2019 bone marrow dataset Lareau et al. (2019). The dataset profiles chromatin accessibility in bone marrow mononuclear cells under untreated control and ex vivo lipopolysaccharide (LPS)-stimulated conditions using droplet-based single-cell combinatorial indexing for ATAC-seq (dsciATAC-seq). It contains 15 annotated cell types: B, CD4, CD8, CLP, Collision, Ery-early, Ery-late, HSPC, HSPC-ery, Mono-1, Mono-2, NK, pDC, preB, and proB. The following sections describe the cell-type-level split, perturbation mapping, trajectory inference, control–perturbed pairing, and meta-cell-based evaluation in detail.

E.2 EVALUATION: CELL-TYPE-LEVEL SPLIT

To test whether perturbation responses learned from some cell types can be transferred to unseen cell types, we split the dataset at the cell-type level. For each pre-trained scATAC-seq model, we first generated cell embeddings for all control and perturbed cells and used these embeddings as fixed inputs to the perturbation predictor. Among the 15 cell types, CD8 and NK cells were completely held out from training the perturbation predictor, while the remaining 13 cell types were used for training. For both the training and held-out cell types, cells from both control and perturbed conditions were available. Thus, the perturbation mapping was learned only from the control-to-perturbed transitions of the 13 training cell types and was then evaluated on CD8 and NK cells, whose transitions were not observed during training. The same cell-type split was applied to the embeddings generated by all scATAC-seq models.

E.3 MODEL: LATENT PERTURBATION MAPPING

Given a control-cell embedding, $\mathbf{c}_{\mathrm{ctrl}} \in \mathbb{R}^d$, we use a simple neural network f_θ to predict how the cell embedding should move toward the perturbed state. The network takes the control-cell embedding as input and outputs two d-dimensional vectors: a shift vector $\Delta\mathbf{c}$ and a log-variance vector $\log \boldsymbol{\sigma}^2$:

$$f_\theta : \mathbb{R}^d \to \mathbb{R}^{2d}, \qquad f_\theta(\mathbf{c}_{\mathrm{ctrl}}) = \begin{bmatrix} \Delta\mathbf{c} \\ \log \boldsymbol{\sigma}^2 \end{bmatrix}. \tag{13}$$

The network consists of two fully connected layers with a GELU activation between them. Here, $\Delta\mathbf{c}$ represents the direction and magnitude of the predicted perturbation-induced movement in the embedding space, while $\log \boldsymbol{\sigma}^2$ represents the uncertainty associated with this movement. Therefore, the mean of the predicted perturbed embedding is obtained by adding the predicted shift to the original control-cell embedding:

$$\boldsymbol{\mu}_{\mathrm{pert}} = \mathbf{c}_{\mathrm{ctrl}} + \Delta\mathbf{c}. \tag{14}$$

The predicted perturbed embedding is then sampled as

$$\hat{\mathbf{c}}_{\mathrm{pert}} = \boldsymbol{\mu}_{\mathrm{pert}} + \boldsymbol{\epsilon} \odot \boldsymbol{\sigma}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{15}$$

where

$$\boldsymbol{\sigma} = \exp\left(\frac{1}{2} \log \boldsymbol{\sigma}^2\right). \tag{16}$$

E.4 TRAINING: CELL-TYPE-SPECIFIC PERTURBATION SHIFT

Because control and perturbed cells are not paired at the single-cell level, the perturbed cell corresponding to each control cell is unknown. We therefore estimate a perturbation direction for each cell type from the mean difference between its control and perturbed cells. This cell-type-specific shift is then added to each control-cell embedding to construct a reference perturbed embedding for training.

For each cell type c, we first calculate the mean embeddings of the control and perturbed cells:

$$\bar{\mathbf{c}}_{\text{ctrl}}^{(c)} = \frac{1}{N_{\text{ctrl}}^{(c)}} \sum_{i \in \mathcal{C}_{\text{ctrl}}^{(c)}} \mathbf{c}_i, \tag{17}$$

and

$$\bar{\mathbf{c}}_{\text{pert}}^{(c)} = \frac{1}{N_{\text{pert}}^{(c)}} \sum_{j \in \mathcal{C}_{\text{pert}}^{(c)}} \mathbf{c}_j. \tag{18}$$

The perturbation shift for cell type c is defined as the difference between these two mean embeddings:

$$\mathbf{s}^{(c)} = \bar{\mathbf{c}}_{\text{pert}}^{(c)} - \bar{\mathbf{c}}_{\text{ctrl}}^{(c)}. \tag{19}$$

This vector represents the average direction and magnitude of the control-to-perturbed transition for cell type c. For each control cell i of that cell type, we construct a target perturbed embedding by applying the same shift to its original embedding:

$$\mathbf{c}_{i,\text{pert}}^{*} = \mathbf{c}_{i,\text{ctrl}} + \mathbf{s}^{(c)}. \tag{20}$$

The neural network is then trained to move each control-cell embedding toward this target. Specifically, we minimize the mean squared error between the predicted perturbed embedding and the constructed target:

$$\mathcal{L}_{\text{shift}} = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \frac{1}{N_{\text{ctrl}}^{(c)}} \sum_{i \in \mathcal{C}_{\text{ctrl}}^{(c)}} \left\| \hat{\mathbf{c}}_{i,\text{pert}} - \mathbf{c}_{i,\text{pert}}^{*} \right\|_2^2, \tag{21}$$

where $\mathcal{C}$ denotes the set of training cell types represented in the batch. In this way, the network learns cell-type-specific perturbation directions without requiring one-to-one control–perturbed cell pairs.

E.5 INFERENCE: PERTURBATION TRAJECTORY

At inference, we repeatedly apply the trained perturbation network f_θ to track how a control-cell embedding moves toward the perturbed state. Starting from the original control-cell embedding, $\mathbf{c}^{(0)} = \mathbf{c}_{\text{ctrl}}$, the current embedding $\mathbf{c}^{(t)}$ is used as the input to the network at each step:

$$f_\theta(\mathbf{c}^{(t)}) = \begin{bmatrix} \Delta\mathbf{c}^{(t)} \\ \log \boldsymbol{\sigma}^{2(t)} \end{bmatrix}. \tag{22}$$

Thus, the network predicts a new perturbation shift $\Delta\mathbf{c}^{(t)}$ from the current position in the embedding space, together with its uncertainty. The current embedding is then updated as

$$\mathbf{c}^{(t+1)} = \mathbf{c}^{(t)} + \alpha\Delta\mathbf{c}^{(t)} + \tau\boldsymbol{\epsilon}^{(t)} \odot \boldsymbol{\sigma}^{(t)}, \qquad \boldsymbol{\epsilon}^{(t)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{23}$$

where α controls the step size along the predicted perturbation direction and τ controls the amount of stochastic variation. The updated embedding $\mathbf{c}^{(t+1)}$ is then used as the input to the same network at the next step.

By repeatedly applying this procedure, we obtain a sequence of predicted embeddings,

$$\mathbf{c}_{\text{ctrl}} \to \mathbf{c}^{(1)} \to \mathbf{c}^{(2)} \to \cdots \to \mathbf{c}^{(T)} \simeq \mathbf{c}_{pert}, \tag{24}$$

which represents the predicted perturbation trajectory. If the cell embedding preserves perturbation-associated biological variation, these predicted states should progressively move from the control-cell region toward the region occupied by observed perturbed cells. We therefore compare the predicted trajectory with the distribution of real perturbed cells in the same embedding space.

E.6 Meta-cell construction: Trajectory-guided control–perturbed pairing

After generating the perturbation trajectory, we use the embedding at the final prediction step, $\mathbf{c}^{(5)}$, to identify an observed perturbed cell corresponding to each control cell. Pairing is performed separately within each cell type so that a control cell can only be matched to a perturbed cell of the same cell type. Specifically, we calculate the cosine distance between all predicted step-5 embeddings and observed perturbed-cell embeddings within each cell type and use the Hungarian algorithm to obtain one-to-one matches that minimize the total cosine distance. Thus, each control cell and each observed perturbed cell can be used in at most one pair.

Because single-cell scATAC-seq profiles are highly sparse, we evaluate accessibility changes at the meta-cell level rather than directly from individual control–perturbed cell pairs. We first construct a control meta-cell around each paired control cell. Using the cell embeddings generated by each scATAC-seq method, the control cell is used as an anchor, and control cells of the same cell type with cosine similarity of at least 0.9 are selected. The anchor cell is always included, and up to 10 control cells are retained.

The perturbed meta-cell is then constructed directly from the perturbed cells paired with the control cells in the control meta-cell. Thus, no independent neighborhood search is performed among perturbed cells. If the control meta-cell contains

$$\{c_1^{\text{ctrl}}, c_2^{\text{ctrl}}, \ldots, c_k^{\text{ctrl}}\},$$

and the corresponding paired perturbed cells are

$$\{c_1^{\text{pert}}, c_2^{\text{pert}}, \ldots, c_k^{\text{pert}}\},$$

the paired meta-cells are defined as

$$M_m^{\text{ctrl}} = \{c_1^{\text{ctrl}}, c_2^{\text{ctrl}}, \ldots, c_k^{\text{ctrl}}\}, \qquad M_m^{\text{pert}} = \{c_1^{\text{pert}}, c_2^{\text{pert}}, \ldots, c_k^{\text{pert}}\}, \qquad k \le 10. \tag{25}$$

In this way, each perturbed meta-cell is determined by the membership of its paired control meta-cell, preserving the inferred control–perturbed relationships during meta-cell-level accessibility analysis.

E.7 Validation 1: Perturbation-responsive peak identification

We evaluate the quality of the inferred control–perturbed pairs by testing whether they recover consistent perturbation-associated changes in chromatin accessibility. If the inferred pairs reflect biologically meaningful transitions, the same peaks should tend to increase or decrease in accessibility across multiple meta-cell pairs of the same cell type.

We first binarize the original scATAC-seq profile of each cell, such that each peak is represented as accessible or inaccessible. For each control–perturbed meta-cell pair m and peak p, we calculate the fraction of cells in which the peak is accessible:

$$a_{m,p}^{\mathrm{ctrl}} = \frac{1}{|M_m^{\mathrm{ctrl}}|} \sum_{i \in M_m^{\mathrm{ctrl}}} \mathbb{I}(x_{i,p} > 0), \tag{26}$$

$$a_{m,p}^{\mathrm{pert}} = \frac{1}{|M_m^{\mathrm{pert}}|} \sum_{j \in M_m^{\mathrm{pert}}} \mathbb{I}(x_{j,p} > 0). \tag{27}$$

The accessibility change of peak p in meta-cell pair m is then defined as

$$\Delta a_{m,p} = a_{m,p}^{\mathrm{pert}} - a_{m,p}^{\mathrm{ctrl}}. \tag{28}$$

A positive $\Delta a_{m,p}$ indicates increased accessibility after perturbation, whereas a negative value indicates decreased accessibility.

For each cell type and peak, we evaluate whether the accessibility change is consistently observed across meta-cell pairs. We first consider peaks that show a nonzero accessibility change ($\Delta a_{m,p} \neq 0$) in at least 10 meta-cell pairs. For each such peak, we determine the dominant direction of change and require at least 65% of the changed pairs to agree on either increased or decreased accessibility. We further require this dominant direction to be observed in at least 60% of all meta-cell pairs for that cell type. Peaks satisfying these consistency criteria are defined as consistent perturbation-responsive peaks. We use the number of such peaks to evaluate how well each pairing strategy recovers reproducible perturbation-associated changes in chromatin accessibility.

E.8 Validation 2: scATAC accessibility prediction

We further evaluate the similarity between the predicted perturbed cells at the final trajectory step and real perturbed cells by decoding their embeddings into scATAC-seq accessibility profiles and comparing the resulting accessibility patterns. For each representation model, we train a decoder that maps a cell embedding to the accessibility probabilities of 2,000 frequently accessible genomic peaks. The decoder consists of a three-layer MLP with two 512-dimensional hidden layers, ReLU activations, and a dropout rate of 0.1. The decoder is trained using only the training cell types to reconstruct binarized accessibility profiles with binary cross-entropy loss.

Given a cell embedding $\mathbf{c}_i$, the decoder predicts the accessibility probability of the selected peaks as

$$\hat{\mathbf{x}}_i = \sigma\left(D_\theta(\mathbf{c}_i)\right), \tag{29}$$

where D_θ denotes the trained decoder, σ is the sigmoid function, and $\hat{\mathbf{x}}_i \in [0,1]^{2000}$ represents the predicted scATAC-seq accessibility profile.

During validation, we apply the same decoder to the control cells, each intermediate predicted cells along the perturbation trajectory, and the real perturbed cells. This yields decoded scATAC-seq accessibility profiles for the control cells, predicted steps $t = 1, \ldots, 5$, and the real perturbed cells. For each test cell type, the decoded profiles from all states are jointly projected into a 50-dimensional PCA space. We then quantify the distance between each trajectory state and the real perturbed state using Euclidean distance, energy distance, and maximum mean discrepancy (MMD), implemented with the Pertpy library (Heumos et al., 2026).

Specifically, for distance metric d, we evaluate

$$d_t = d\left(\hat{\mathbf{X}}^{(t)}, \hat{\mathbf{X}}^{\mathrm{pert}}\right), \qquad t \in \{\mathrm{ctrl}, 1, \ldots, 5\}, \tag{30}$$

where $\hat{\mathbf{X}}^{(t)}$ denotes the decoded accessibility profiles at trajectory step t and $\hat{\mathbf{X}}^{\mathrm{pert}}$ denotes those obtained from the real perturbed cells. A progressive decrease in d_t indicates that the predicted trajectory moves toward the perturbation-associated accessibility state in the decoded scATAC-seq space.

In addition, we evaluate whether the final predicted state reproduces the peak-wise accessibility changes observed in the real perturbed state. For each test cell type, we first average the decoded

accessibility profiles across cells separately for the control, final predicted ($t = 5$), and real perturbed states. For each peak p, we then calculate the real and predicted perturbation effects as

$$\Delta a_p^{\mathrm{real}} = \bar{a}_p^{\mathrm{pert}} - \bar{a}_p^{\mathrm{ctrl}}, \tag{31}$$

and

$$\Delta a_p^{\mathrm{pred}} = \bar{a}_p^{(5)} - \bar{a}_p^{\mathrm{ctrl}}, \tag{32}$$

where $\bar{a}_p^{\mathrm{ctrl}}$, $\bar{a}_p^{(5)}$, and $\bar{a}_p^{\mathrm{pert}}$ denote the mean decoded accessibility of peak p across cells in the control, final predicted, and real perturbed states, respectively. We compute the Pearson delta correlation between Δa^{real} and Δa^{pred} across the 2,000 decoded peaks separately for each test cell type. A higher correlation indicates that the final predicted state better reproduces the direction and relative magnitude of perturbation-associated accessibility changes observed in real perturbed cells.

F ADDITIONAL TABLES

Table A1: Hyperparameters used for target-dataset pretraining of scChromHEARTs.

Hyperparameter	**Value**
Number of Transformer layers	1–2
Hidden dimension	512
Number of attention heads	8
Attention dropout	0.0
Number of chromosome tokens	24
RoPE base	10^6
Input genomic regions	500
Batch size	38–42
Learning rate	5×10^{-5}
Number of epochs	1,200

Table A2: Summary of scATAC-seq datasets used for downstream evaluation. The table reports the evaluation task, tissue, number of annotated cell types, and number of cells for each dataset.

Task	**Dataset**	**Tissue**	**# Cell types**	**# Cells**
Cell type annotation	Ameen2022 (Ameen et al., 2022)	Cardiac	20	30,426
	Kanemaru2023 (Kanemaru et al., 2023)	Cardiac	12	139,835
	Granja2019 (Granja et al., 2019)	Bone marrow	22	26,670
	Buenrostro2018 (Buenrostro et al., 2018)	Bone marrow	10	2,034
	Granja2019 (Granja et al., 2019)	PBMC	18	8,368
	Muto2021 (Muto et al., 2021)	Kidney	13	24,205
	Li2023 (Li et al., 2023)	Brain	15	112,762
Batch effect correction	GSE194122 (Luecken et al., 2021)	BMMC	22	69,249
	GSE204682 (Zhu et al., 2023)	Cortex	15	45,549
Perturbation	Lareau2019 (Lareau et al., 2019)	Bone marrow	15	136,425
Cross-dataset transferability	Hocker2021 (Hocker et al., 2021)	Cardiac	27	80,389
	Zhang2021 (Zhang et al., 2021)	Cardiac	25	31,384

Table A3: Cell-type clustering performance of cell embeddings generated by each scATAC-seq method (ARI / NMI; higher is better). Best results are shown in **bold**, and second-best results are underlined.

Dataset	Metric	scChromHEARTs	EpiAgent	ChromFound	LSI	SANGO	PeakVI	PoissonVI
GSE194122	ARI ↑	0.532	**0.664**	0.490	0.583	0.628	0.545	0.551
	NMI ↑	0.674	**0.733**	0.627	0.731	0.716	0.706	0.699
GSE204682	ARI ↑	0.808	0.845	0.809	0.782	**0.851**	0.813	0.794
	NMI ↑	0.824	0.855	0.804	0.834	**0.858**	0.850	0.848

Table A4: Number of consistent perturbation-responsive peaks identified from control and perturbed meta-cell pairs constructed using each pairing strategy.

Pairing	scChromHEARTs	EpiAgent	ChromFound	LSI	SANGO	PeakVI	PoissonVI
Trajectory	**34**	17	0	20	0	17	6
Random	18	20	0	20	0	21	7
OT	19	20	0	18	0	19	6

G Additional Figures

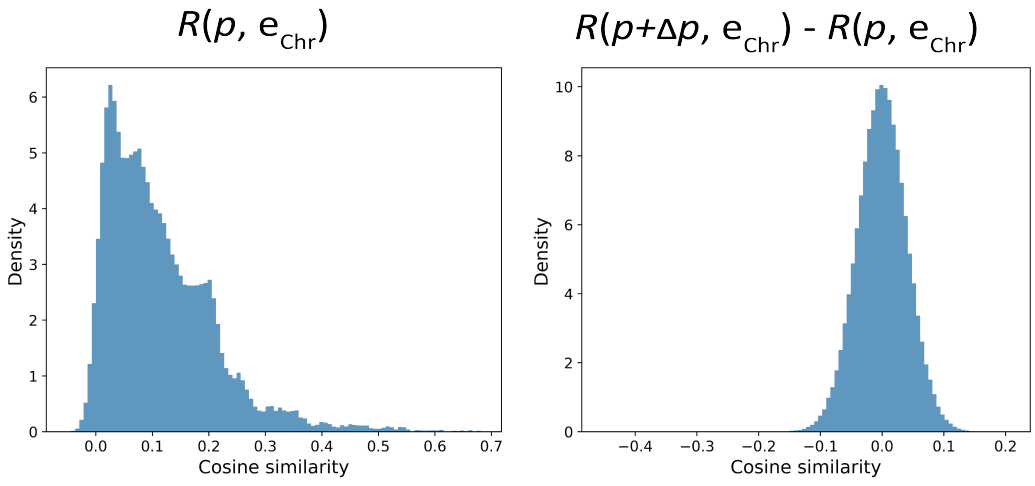

Figure A1: Cosine similarity distributions of chromosome-aware positional embeddings. The left panel shows pairwise cosine similarities between rotated chromosome embeddings $R(p, \mathbf{e}_{\text{Chr}})$, while the right panel shows pairwise cosine similarities between relative positional difference embeddings $R(p + \Delta p, \mathbf{e}_{\text{Chr}}) - R(p, \mathbf{e}_{\text{Chr}})$.

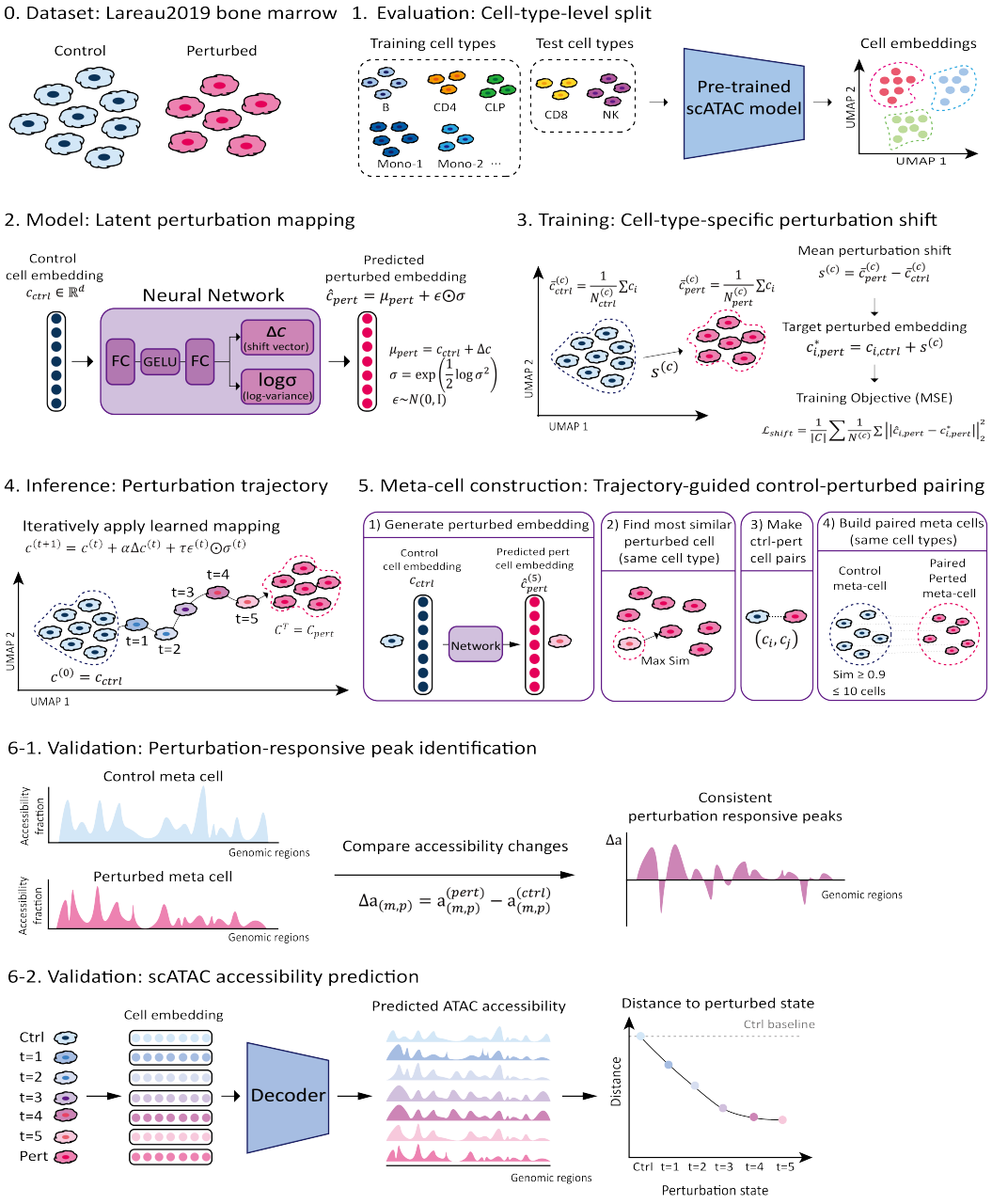

Figure A2: **Overview of trajectory-guided perturbation analysis.** (0) CD8 and NK cells are held out for testing, while the remaining cell types are used for training. (1) Cell embeddings are generated using each scATAC-seq model. (2) A neural network learns a latent perturbation mapping from control to perturbed states. (3) Cell-type-specific perturbation shifts are used as training targets. (4) The learned mapping is iteratively applied to generate perturbation trajectories. (5) The final predicted states are used for control–perturbed cell pairing and paired meta-cell construction. (6-1) Perturbation-responsive peaks are identified based on consistent accessibility changes between control and perturbed meta-cells. (6-2) All states are decoded into scATAC accessibility profiles, and the resulting predicted profiles are compared based on their distances to the predicted perturbed profile.

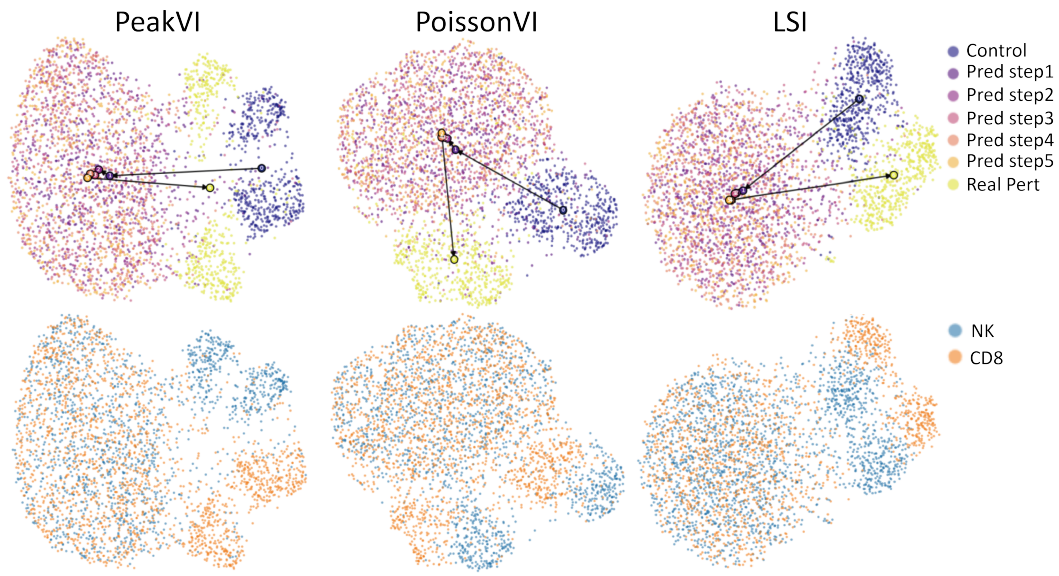

Figure A3: UMAP visualization of cell-type-specific perturbed cell generation along the predicted trajectory.

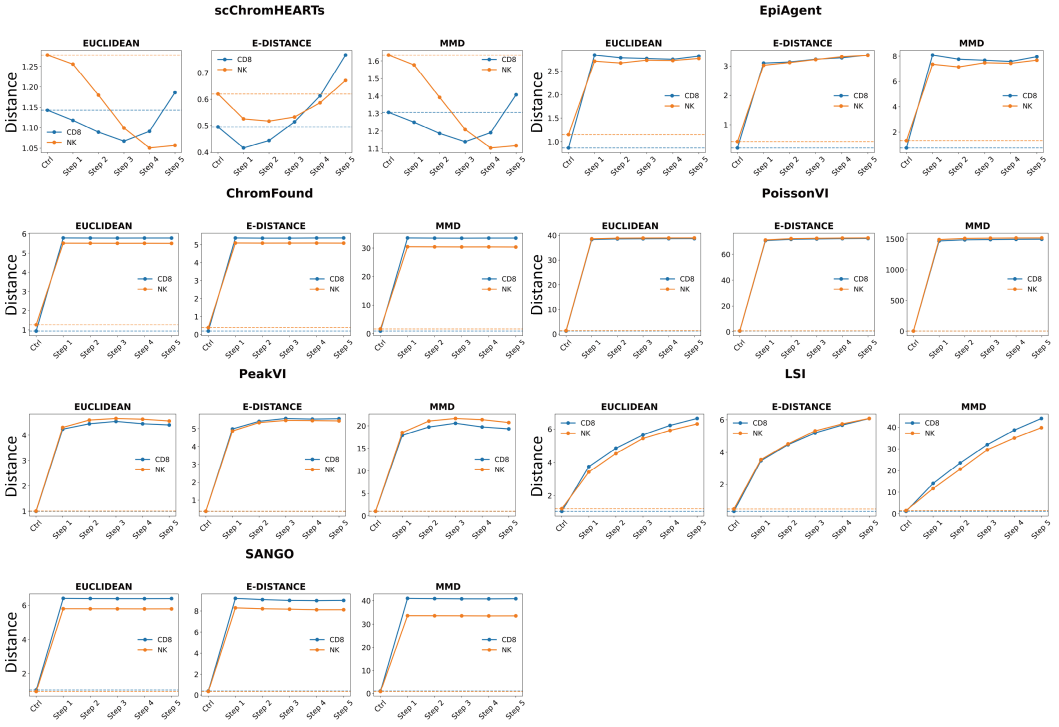

Figure A4: Line plots of distances between decoded scATAC accessibility profiles and the predicted perturbed state for each cell type, where distances were quantified using Euclidean distance, energy distance (E-distance), and maximum mean discrepancy (MMD). The dashed line denotes the distance between the control and predicted perturbed profiles (Ctrl baseline).

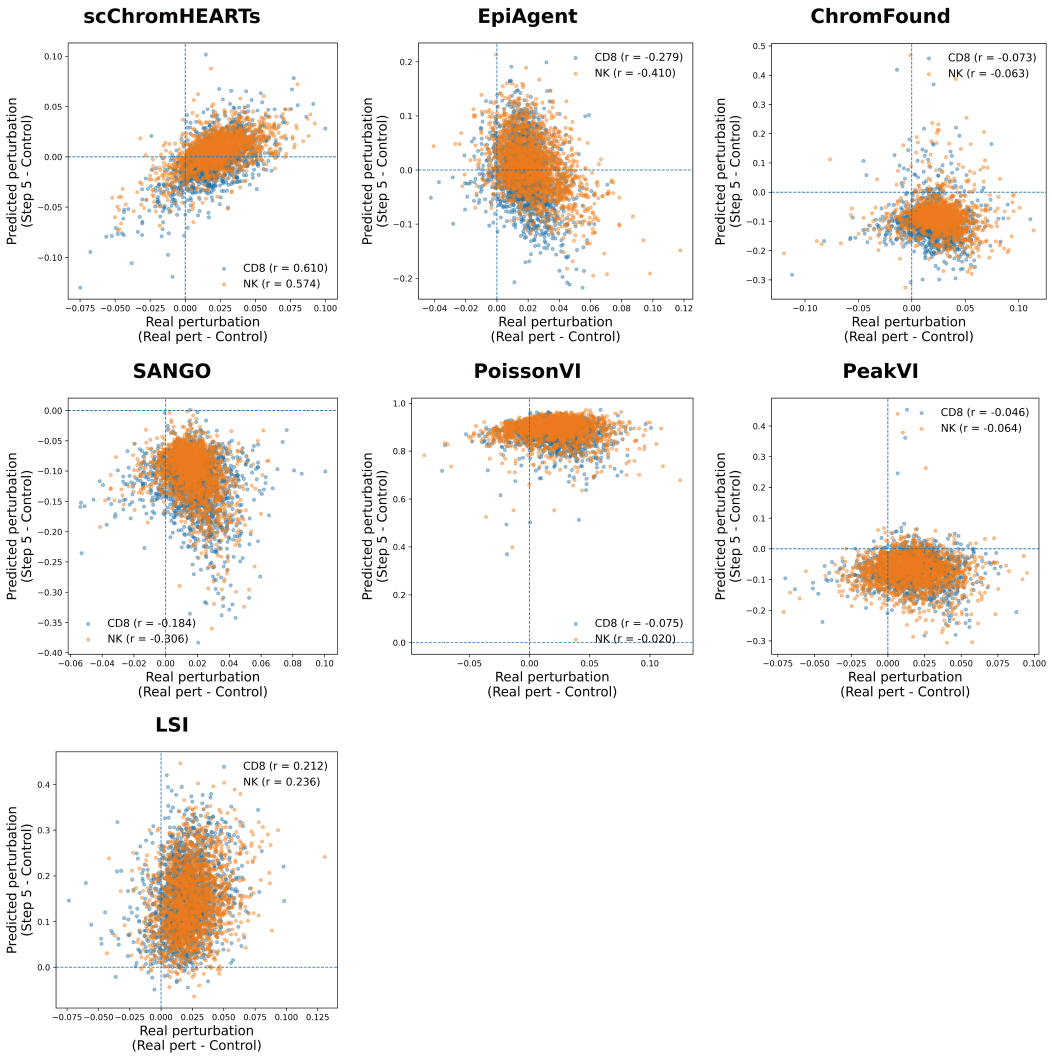

Figure A5: Scatter plots comparing real and predicted peak-wise perturbation effects. Peak-wise accessibility changes were calculated from profiles averaged across cells within each cell type. Each point represents a genomic peak, and Pearson delta correlation coefficients indicate the agreement between real and predicted perturbation effects.

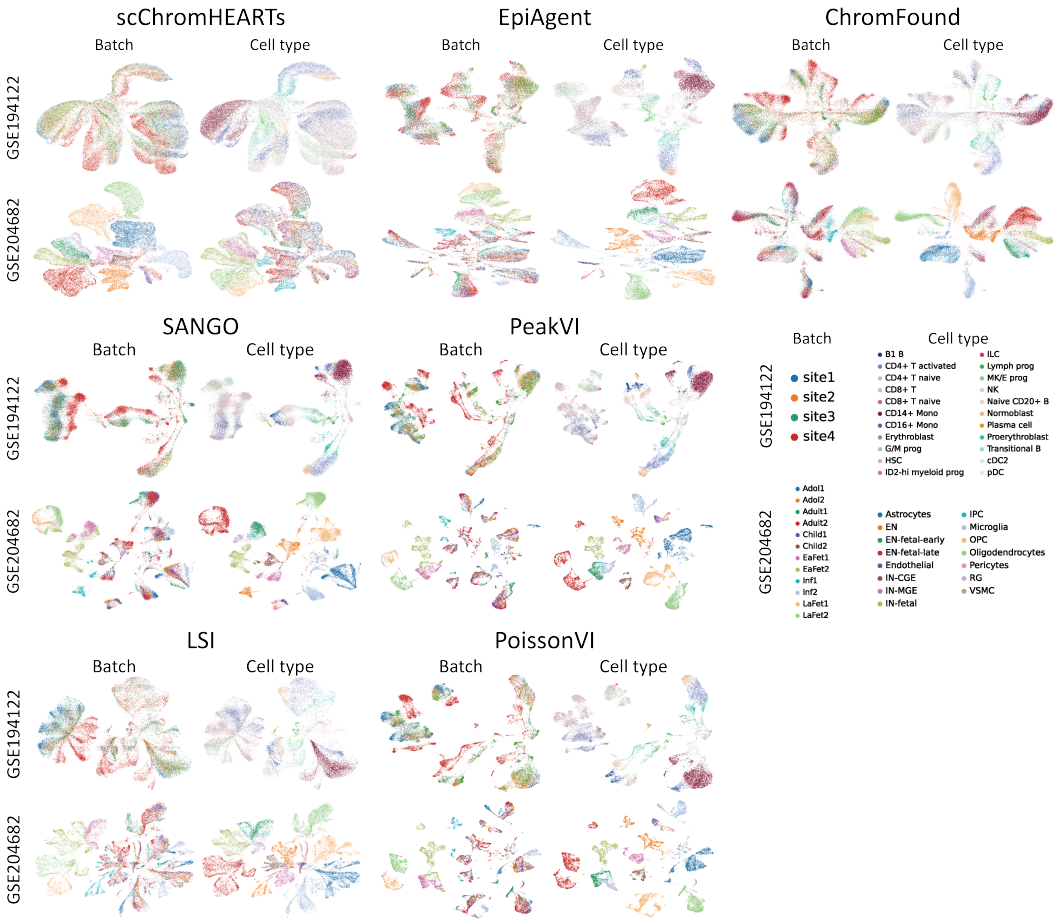

Figure A6: UMAP visualization of cell embeddings colored by batch and cell type across two scATAC-seq datasets and different representation methods.

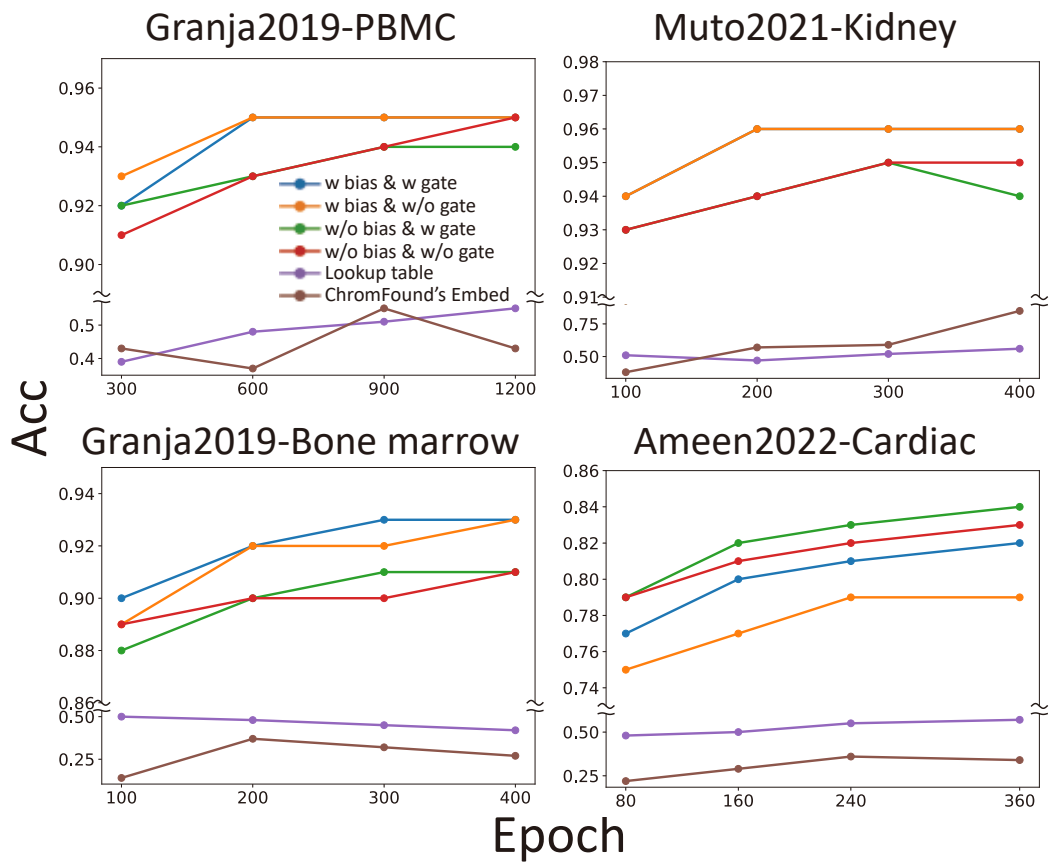

Figure A7: Accuracy of genomic region embedding variants across training epochs on four scATAC-seq datasets.

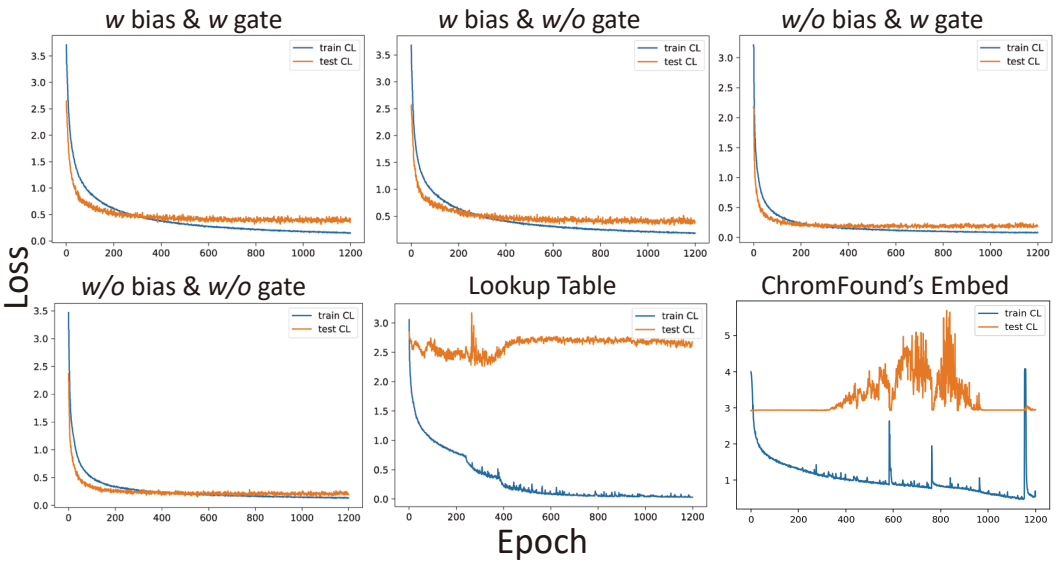

Figure A8: Training and test contrastive losses on Granja2019-PBMC across different genomic region embedding designs.

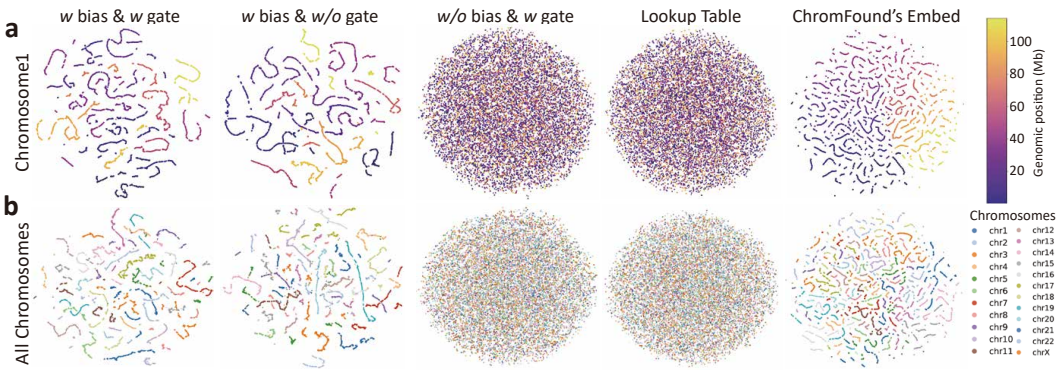

Figure A9: UMAP visualization of genomic region embeddings under different embedding designs before pre-training on Granja2019-PBMC. Genomic intervals were embedded using each randomly initialized region embedding design. (a) chromosome 1 embeddings colored by genomic position and (b) embeddings from all chromosomes colored by chromosome identity.