Accurate, scalable, and unified single-cell atlas integration with scBIOT

Haihui Zhang[1,2*], Peiwu Qin[3]

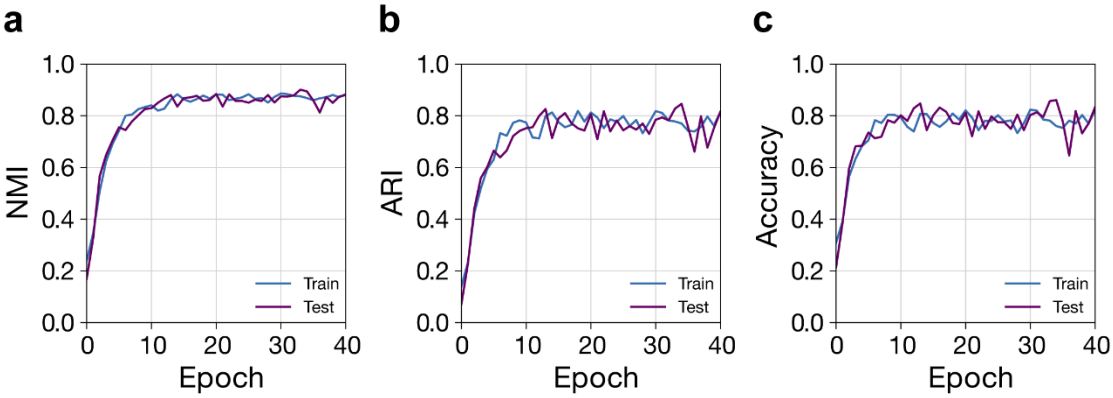
[1]Department of Orofacial Sciences, School of Dentistry, University of California San Francisco, San Francisco, CA
[2]Cancer Biology and Immunology Laboratory, University of California San Francisco, San Francisco, CA
[3]Institute of Biopharmaceutical and Health Engineering, Shenzhen International Graduate School, Tsinghua University, Shenzhen, Guangdong, China

[*]Correspondence should be addressed to Haihui Zhang (Haihui.Zhang@ucsf.edu).

Keywords: scBIOT, Optimal transport, Generative model, Transformer, VAE, Self-supervised learning, Contrastive learning, Multi-omics, Single-cell integration

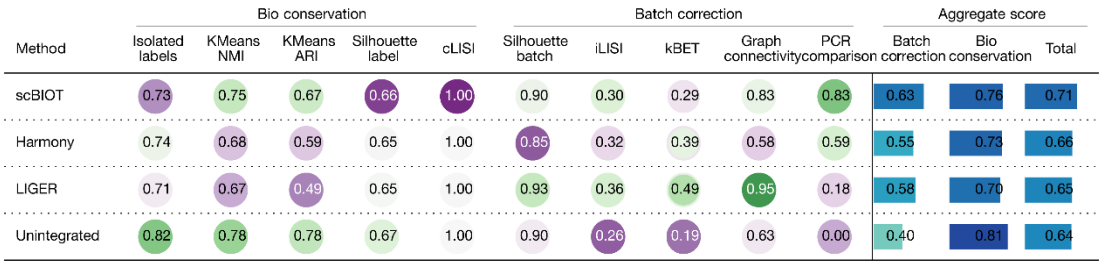
Extended Data

Extended Data Fig. 1



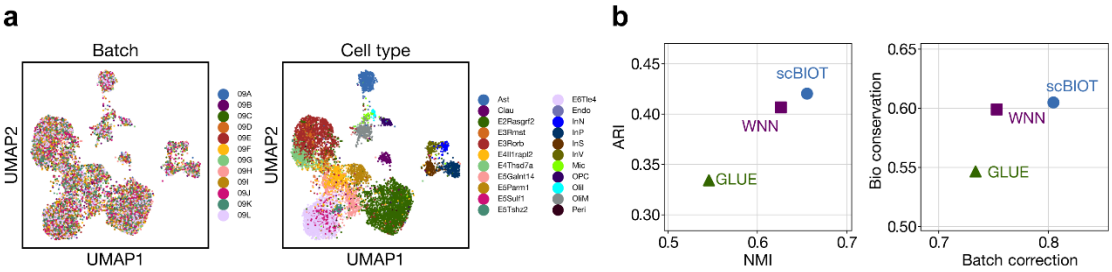
Extended Data Fig. 1. Benchmarking clustering metrics against OT-derived pseudo-labels. **a,** Normalized mutual information (NMI) between model-inferred clusters and pseudo-labels generated via optimal transport (OT). **b,** Adjusted Rand index (ARI), measuring similarity between pseudo-labels and predicted labels. **c,** Classification accuracy, the fraction of cells whose predicted labels match the OT pseudo-labels. Across panels, higher values indicate greater concordance with the OT reference.

Extended Data Fig. 2



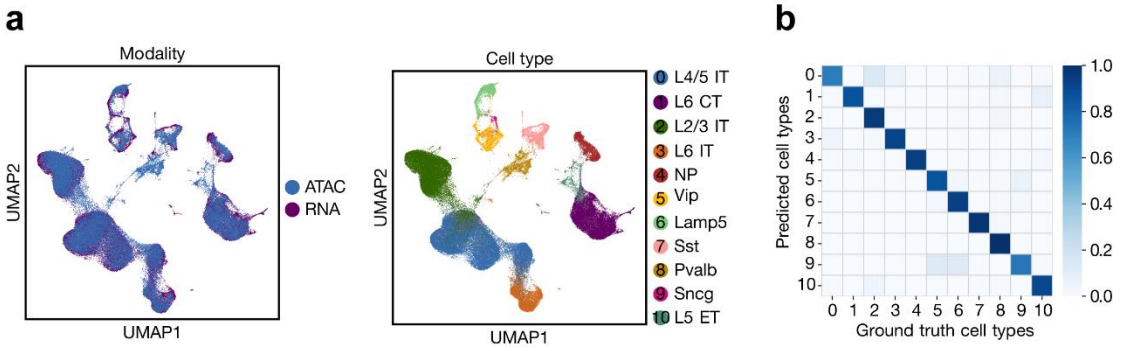
Extended Data Fig. 2. Quantitative comparison of biological conservation and batch-effect correction across iterative LSI (Unintegrated), LIGER, Harmony, and scBIOT for scATAC-seq brain dataset with large-window peaks[1].

Extended Data Fig. 3



Extended Data Fig. 3. a, UMAP embeddings colored by batch and scBIOT-defined clusters in the SNARE-seq datasets. **b,** ARI and NMI scores comparing three integration methods across the datasets[2].

Extended Data Fig. 4



Extended Data Fig. 4. a, UMAP of the unpaired scRNA-seq and snATAC-seq datasets integrated with scBIOT, colored by batch (left) and by scBIOT-inferred clusters (right). **b,** Confusion matrix quantifying agreement between reference cell-type annotations and scBIOT-transferred labels from scRNA-seq to snATAC-seq[3].

Reference

1. Luecken, M. D. *et al.* Benchmarking atlas-level data integration in single-cell genomics. *Nat Methods* **19**, 41–50 (2022).

2. Chen, S., Lake, B. B. & Zhang, K. High-throughput sequencing of the transcriptome and chromatin accessibility in the same cell. *Nat Biotechnol* **37**, 1452–1457 (2019).

3. Yao, Z. *et al.* A transcriptomic and epigenomic cell atlas of the mouse primary motor cortex. *Nature* **598**, 103–110 (2021).

Extended Methods

Batch-aware optimal-transport (OT) integration with rare-type protection

Data and notation

Let $X_0 \in R^{N\times d}$ be the baseline embedding (e.g., PCA, LSI). Each cell i has a batch label $b_i \in \{1,\dots,B\}$. The algorithm iteratively updates the embedding $X^{(t)}$ and returns the best iterate X^*.

Reference and batch prototypes (rare-aware)

For a chosen reference mode (largest batch or union), we compute prototypes by MiniBatch *k*-means with sparsity weights:

$$w_i \;=\; \frac{\bar{d}_i}{\frac{1}{N}\sum_{j=1}^{N}\bar{d}_j}, \qquad \bar{d}_i \;=\; \frac{1}{k}\sum_{j\in\mathcal{N}_k(i)} \| x_i - x_j \|_2, \tag{1}$$

Where $\mathcal{N}_k(i)$ are *k*-nearest neighbors (kNN) in $X^{(t)}$.

Unbalanced entropic OT and barycentric projection

Between a batch prototype set (B_ℓ) and the reference (R), we solve unbalanced entropic OT with squared-Euclidean cost:

$$C_{pq} \;=\; \frac{\| b_p - r_q \|_2^2}{\sigma_C^2}, \qquad \sigma_C \;=\; std(C), \tag{2}$$

optionally clipping large entries to a chosen quantile to stabilize tails.

Let $a \in \Delta^{K_{\text{batch}}\text{-}1}$ and $b \in \Delta^{K_{\text{ref}}\text{-}1}$ be uniform masses. We compute a coupling $T \in R_+^{K_{batch}\times K_{ref}}$ that minimizes

$$\mathcal{L}_{\mathrm{UOT}}(T) = \langle C, T\rangle + \varepsilon\, KL(T \parallel \mathbf{1}) + \lambda\, KL(T\mathbf{1} \parallel a) + \lambda\, KL(T^{\top}\mathbf{1} \parallel b), \tag{3}$$

with entropic regularization e (reg) and marginal relaxation λ (reg_m). We use Sinkhorn scaling in log-space (GPU) with updates

$$K = e^{-C/\varepsilon}, \quad u^{(t+1)} = \left(\frac{a}{Kv^{(t)}}\right)^{\tau}, \quad v^{(t+1)} = \left(\frac{b}{K^{\top}u^{(t+1)}}\right)^{\tau}, \qquad \tau = \frac{\lambda}{\lambda+\varepsilon}, \tag{4}$$

until convergence in $\left\|\log u^{(t+1)} - \log u^{(t)}\right\|_{\infty}$ and $\left\|\log v^{(t+1)} - \log v^{(t)}\right\|_{\infty}$.

Each prototype $b_p$is projected to the reference via the OT barycenter:

$$\tilde{b}_p \;=\; \frac{\sum_q T_{pq}\; r_q}{\sum_q T_{pq}}. \tag{5}$$

Prototype-to-cell displacement and bridge weighting

Cells inherit the prototype displacement with adaptive weights:

$$\Delta_p^{\mathrm{proto}} = \tilde{b}_p - b_p, \qquad s_p = \frac{1}{1+\parallel \Delta_p^{\mathrm{proto}} \parallel_2/\sigma_{\Delta}}, \qquad \alpha_i = s_{n(i)}(1 - 0.35\,\beta_i), \tag{6}$$

where $\sigma_{\Delta} = std\left(\left\|\Delta_p^{\mathrm{proto}}\right\|_2\right)$ and β_i is the bridge score. The initial per-cell shift is

$$\mathbf{s}_i = \alpha_i\, \Delta_{n(i)}^{\mathrm{proto}}. \tag{7}$$

Bridge score from batch mixing and sparsity

Let $p_{i,c}$ be the fraction of batch (c) among the kNN of cell i in $X^{(t)}$. Define the neighbor batch entropy

$$H_i = -\sum_{c=1}^{B} p_{i,c}\, log\left(p_{i,c} + 10^{-12}\right), \qquad \widehat{H}_i = \frac{H_i}{\log B}, \tag{8}$$

is combined with a normalized local sparsity score $\hat{\rho}_i \in [0,1]$ (from Eq. 1) to form

$$\beta_i = 0.5\,\widehat{H}_i + 0.5\,\hat{\rho}_i. \tag{9}$$

Larger β_i down-weights the move (update) near inter-batch bridges and in very sparse regions.

Cluster-sharpening field

To encourage crisper clusters, we compute *K* pseudo-centers $\{c_k\}$. For each cell i, let c_1 and c_2 be the nearest and second-nearest centers with distances $d1 < d2$. Define a margin

$$m_i = \frac{d_2 - d_1}{\text{median}(d_2)}, \qquad g_i = \frac{1}{1 + \exp\big((m_i - 1)/0.8\big)}, \tag{10}$$

and a sharpening displacement

$$\mathbf{q}_i = pull\,(c_1 - x_i) + g_i\,push\,(x_i - c_2). \tag{11}$$

The total proposed shift is

$$\tilde{\mathbf{s}}_i = \mathbf{s}_i + \gamma_{\text{sharp}}\,\mathbf{q}_i. \tag{12}$$

Rare-aware kNN smoothing

On the fixed kNN graph of X_0 with neighbor set $\mathcal{N}_0(i)$, we smooth shifts but skip rare cells (top 15% sparsest by $\hat{\rho}_i$):

$$\tilde{\mathbf{s}}_i \leftarrow (1-\lambda)\,\tilde{\mathbf{s}}_i + \lambda\,\frac{1}{|\mathcal{N}_0(i)|}\sum_{j\in\mathcal{N}_0(i)}\tilde{\mathbf{s}}_j\,. \tag{13}$$

Step capping and edge-stretch guards

We cap each step size by the local baseline scale:

$$\boldsymbol{m_i} = clip\Big(\eta\, \tilde{\boldsymbol{s}}_i, \|.\|_2 \le \kappa\, \bar{d}_i^{(0)}\Big), \tag{14}$$

with η the global step (step_lo→step_hi), κ = max_step_local, and $\bar{d}_i^{(0)}$ from Eq. 1 computed in X_0.

To prevent graph over-stretch/compression relative to the original geometry, we enforce for all (i, j) edges of the kNN graph in X_0:

$$s_i^{\min} \le \frac{\| (x_i + \mathbf{m}_i) - \big(x_j + \mathbf{m}_j\big) \|_2}{d_{ij}^{(0)}} \le s_i^{\max}, \tag{15}$$

with $s_i^{\min}$, $s_i^{\max}$ interpolated by β_i between bulk and bridge limits. If violated, moves are uniformly scaled to satisfy (15).

The candidate embedding is updated as

$$X^{\text{cand}} = X^{(t)} + M, \quad M = [\mathbf{m}_1, \dots, \mathbf{m}_N]^\top, \qquad X^{(t+1)} = PostScale\big(X^{\text{cand}}; \mu_0, \sigma_0\big), \tag{16}$$

where () recenters and rescales each dimension to match (μ_0, σ_0) of X_0.

Metrics and selection objective

At each iteration we evaluate:

Batch-mixing (higher is better): mean neighbor-entropy

$$mix(X) = \frac{1}{N}\sum_{i=1}^{N} H_i\,(X), \tag{17}$$

computed as in Eq. 8 with neighbors in X.

Neighborhood overlap with the baseline (higher is better):

$$ovl(X_0, X) = \mathbb{E}_i\left[\frac{\left|\mathcal{N}_k^{X_0}(i) \cap \mathcal{N}_k^{X}(i)\right|}{k}\right], \tag{18}$$

Graph strain (lower is better): mean squared relative change of edge lengths,

$$\mathrm{str}(X) = E_{(i,j)}\left[\left(clip\left(\frac{\left\|x_i - x_j\right\|_2}{d_{ij}^{(0)}} - 1, -c, c\right)\right)^2\right]. \tag{19}$$

We select the best iterate by maximizing

$$J(X) = [\mathrm{mix}(X) - \mathrm{mix}(X_0)] + w_{\mathrm{ovl}} ovl(X_0, X) - w_{\mathrm{str}}[\mathrm{str}(X) - \mathrm{str}(X_0)] - \Gamma(X), \tag{20}$$

with soft overlap penalties

$$\Gamma(X) = \gamma\,[max\{0, \underline{o}(t) - ovl\}]^2 + 0.45\, max\{0, \mathrm{ovl}^\star - ovl\}, \tag{21}$$

where $\underline{\boldsymbol{o}}(\boldsymbol{t})$is an annealed floor, $\mathbf{ovl}^\star$ is the best-so-far overlap, and $\boldsymbol{w}_{\mathbf{ovl}}$, $\boldsymbol{w}_{\mathbf{str}}$, γ correspond to w_overlap, w_strain, penalty_gamma. The best $\boldsymbol{X}$ across iterations (patience-based early stopping) is returned.

Trustworthiness

We also report trustworthiness *TW* on a subsample using the standard definition:

$$\mathrm{TW}(k) = 1 - \frac{2}{nk(2n - 3k - 1)} \sum_{i=1}^{n} \sum_{j \in \mathcal{U}_k(i)} \left(r_{X_0}(i,j) - k\right), \tag{22}$$

where $\mathcal{U}_k(i)$ are points that appear in the kNN of *i* in *X* but not in X_0; and $r_{X_0}(i,j)$ is one-based rank of *j* w.r.t. *i* in X_0 (1 = nearest).

Schedules and implementation

We linearly anneal λ (graph smoothing), the global step η, the cost-clipping quantile, and the overlap floor across at most T iterations (default $T = 15$; max_iter). kNN uses FAISS on GPU when available; OT runs on GPU via the custom unbalanced Sinkhorn (ot_backend="torch"; POT fallback provided on CPU).

Evaluation

We report the final iteration index, the metrics in Eqs. 17–19, and trustworthiness TW on a random subsample using the standard definition implemented in scikit-learn (_trustworthiness_score).

Typical hyperparameters

$\varepsilon = 0.028$, $\lambda/(\lambda + \varepsilon) = \tau = 0.40/(0.40 + 0.028)$, $K_{ref} \leq 1024$, $K_{batch} \leq 512$, $pull = 0.78, push = 0.34$; graph smoothing λ∈[0.38,0.52] (annealed); step η∈[0.78,0.96] (annealed); rare cutoff 85th percentile of $\hat{\rho}_i$; guard limits s^{min}∈[0.72,0.88], s^{max}∈[1.24,1.65] (interpolated by β).

Prototype-aware objective for dual-view VAE clustering

We optimized dual-view VAEs augmented with prototype-aware clustering objectives. The models minimize the negative evidence lower bound (ELBO) for the β-VAE, extended with unsupervised and semi-supervised prototype losses that promote cluster consistency and separation.

The basic VAE objective

We minimized the negative evidence lower bound (ELBO) for the β-VAE loss:

$$\mathcal{L}_{vae} \; = \; = E_{q_{\theta(z|x)}}[-\log p_{\emptyset}(x|z)] + \beta \cdot D_{KL}\big(\, q_{\theta}(z|x)||p(z)\big), \qquad (1)$$

where the first term is the reconstruction loss and the second term is the Kullback–Leibler (KL) divergence. For isotropic Gaussian priors, the KL term simplifies to a closed form.

Prototype-aware contrastive learning

Let $z_1, z_2 \in R^{B\times d}$ denote the embeddings of two complementary data views and $\tilde{z} = z \,/\, \|z\|_2$ their normalized forms. A set of learnable prototypes $C = \{C_k\}_{k=1}^{K}$ defines latent cluster centers. The prototype temperature τ controls the sharpness of cosine similarities. . For clustering we use the averaged embedding $z_c = 1/2\,\big(\text{flatten}(z_1) + \text{flatten}(z_2)\big)$.

Cosine assignments

$$\text{logits}_{*}(i,k) = \frac{\tilde{z}_{*}(i)\cdot \tilde{C}_k}{\tau}, \qquad (2)$$

$$q_*(i) = \text{softmax}_k\left(\text{logits}_*(i,k)\right), \qquad q(i) = softmax\left(\text{logits}\left(z_c(i)\right)\right). \tag{3}$$

Deep embedding clustering (DEC) sharpening

$$f_k = \sum_i q\,(i,k), \quad p(i,k) = \frac{q(i,k)^2/f_k}{\sum_j q\,(i,j)^2/f_j}. \tag{4}$$

Prototype losses

1) **DEC clustering**. Aligns predictions with sharpened targets.

$$\mathcal{L}_{\text{clust}} = KL\left(p \parallel \text{softmax}\left(\text{logits}(\text{z}_\text{c})\right)\right). \tag{5}$$

2) **Soft prototype pull.** Pulls embeddings toward prototypes using p-weights; ramp $\text{r}_{\text{center}} = \min(1, \text{epoch}/8)$.

$$\mathcal{L}_{\text{center}} = r_{\text{center}}\mathbb{E}_i \sum_k p\,(i,k)\ \parallel \tilde{z}_c(i) - \tilde{C}_k\ \parallel_2^2. \tag{6}$$

3) **Prototype repulsion.** Separates prototypes with a hinge on cosine similarity (target t).

$$\mathcal{L}_{\text{repulse}} = \mathbb{E}_{i\neq j}[max\{0, cos\left(\tilde{C}_i, \tilde{C}_j\right) - t\}]^2. \tag{7}$$

4) **CosFace margin on confident self-labels.** For samples with $max_k q(i,k) \geq \text{conf\_thr}$, shift the target logit by m; ramp $\text{r\_sup} = \min(1, \text{epoch}/5)$.

$$\mathcal{L}_{\text{margin}} = CE(\text{softmax}(\text{logits}^m), \hat{\text{y}}), \text{logits}^m(\text{i},\text{y}) = logits(\text{i},\hat{\text{y}}) - m. \tag{8}$$

5) **Label-smoothed supervised CE.** Apply to available pseudo-labels $y_{pseudo} \geq 0$ with smoothing ε.

$$\mathcal{L}_{\text{sup}} = \text{CE}_{\text{smoothed }\varepsilon}\left(\text{logits}(\text{z}_\text{c}), y_{\text{pseudo}}\right). \tag{9}$$

6) **Cross-view assignment consistency.** Stabilizes assignments across views.

$$\mathcal{L}_{\text{cons}} = 1/2\,[\mathrm{KL}(q_1 \parallel q_2) + \mathrm{KL}(q_2 \parallel q_1)]. \tag{10}$$

Total objective

The full training objective is the sum of the VAE term and the prototype-aware terms:

$$\mathcal{L}_{total} = \mathcal{L}_{vae} + \lambda_{\text{clust}}\mathcal{L}_{\text{clust}} + \lambda_{\text{center}}\mathcal{L}_{\text{center}} + \lambda_{\text{repulse}}\mathcal{L}_{\text{repulse}} +$$

$$\left(r_{\text{sup}}\lambda_{\text{margin}}\right)\mathcal{L}_{\text{margin}} + \left(r_{\text{sup}}\lambda_{\text{sup}}\right)\mathcal{L}_{\text{sup}} + \lambda_{\text{cons}}\mathcal{L}_{\text{cons}}. \tag{11}$$

Hyperparameters

τ (proto_tau), m (cosface_m), t (repulse_target), confidence threshold (conf_thr), and label smoothing ε (label_eps) govern the prototype dynamics. Weights λ∗ correspond to the respective terms; prototypes are updated by exponential moving average with momentum proto_ema_m.