OPTIMAL TRANSPORT MEETS T-SNE: INTEGRATED VISUALIZATION WITHOUT BATCH EFFECTS IN OT-TSNE

Anonymous authors
Paper under double-blind review

ABSTRACT

Visual exploration is an integral step in understanding the structure of datasets. However, systematic variation between datasets collected under different conditions, known as batch effects, can obscure shared structure and complicate their joint visualization. Existing methods often rely on cross-batch nearest-neighbor relations to integrate the batches. Yet, these local correspondences can be unreliable for challenging cross-batch geometries. We propose optimal-transport t-SNE (OT-tSNE), which replaces nearest-neighbor-based matching with globally coupled affinities derived from entropic optimal transport. Controlled synthetic experiments demonstrated that nearest-neighbor-based methods can deteriorate while optimal-transport affinities and OT-tSNE embeddings remain robust. Across seven real single-cell datasets, we benchmarked OT-tSNE against a broad range of integration and visualization methods and found that it achieves a Pareto-efficient trade-off between batch mixing and biological conservation, and integrates batches particularly well.

1 INTRODUCTION

As many scientific disciplines become more and more data driven, the need for large datasets increases. However, acquiring several smaller datasets is often more feasible than a single large one. Unfortunately, separately recorded datasets regularly show structural differences. Such differences can be technical nuisances like varying measurement techniques or protocols employed, or signals like donors or donor sex, that are undesirable for the task at hand. We collectively call such structural differences *batch effect* and the different smaller datasets *batches*. Many methods have been proposed to remove the batch effect and to integrate batches (Luecken et al., 2022). These typically integrate the data in the original feature space or in a relatively high-dimensional latent space.

As a result, these general-purpose batch removal methods are not directly usable when the aim is to visually explore the data. Visualizing the data as 2D scatter plots, e.g., with neighbor embedding methods like t-SNE (van der Maaten & Hinton, 2008), has become an important step in many exploratory data analysis pipelines (de Bodt et al., 2025). General-purpose batch integration excels at producing good representations for downstream analysis, which then can be visualized with tools like t-SNE. But when visualization is the primary aim, directly producing an integrated visualization is both conceptually simpler and can yield better results. For this goal, we propose *optimal-transport t-SNE (OT-tSNE)*, a novel approach combining optimal transport (Peyré & Cuturi, 2019) and the popular t-SNE algorithm. Vanilla t-SNE lays the k-nearest-neighbor graph (kNN graph) out in 2D. In the presence of a strong batch effect, no kNN edge crosses from one batch to the other, leading to fully separated batches in the visualizations (Fig. 1a, c). The key idea of OT-tSNE is to augment this graph with explicit cross-batch edges before computing the layout. Many existing tools rely on cross-batch k-nearest neighbors, either mutual nearest neighbors, when both points are each others' nearest neighbors (Haghverdi et al., 2018; Stuart et al., 2019; Hie et al., 2019; Barkas et al., 2019) or shared nearest neighbors, when at least one point is the nearest neighbor of the other (Polański et al., 2020; Jain et al., 2021). Nearest neighbors are inherently local as each point selects nearest neighbors separately. This can lead to hubs among the cross-batch edges, especially on the fringes of the batches (Lun, 2019, "kissing problem"), which can deteriorate the integration in the visualization (Fig. 1d,e). Instead, optimal transport computes the overall best association between both

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.

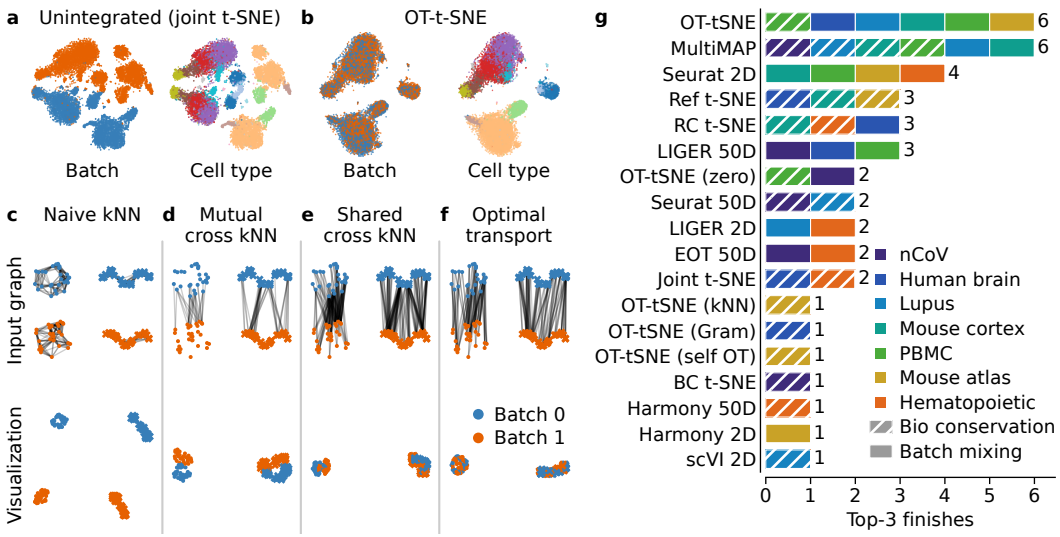

Figure 1: **a.** Unintegrated t-SNE plot of the Lupus (Kang et al., 2018) data shows a clear batch effect. **b.** Our OT-tSNE integrates the batches. **c – f.** t-SNE with various affinity graphs on 2D toy data. **d – f** use the cross-batch edges in addition to the kNN graph. The cross-kNN variants only move the clusters close to each other ("kissing problem"), while optimal transport integrates them. **g.** OT-tSNE and MultiMAP achieved most top-3 ranks across all evaluation metrics and datasets.

batches, while ensuring that each point contributes equally. This allows OT-tSNE to produce very well-integrated visualizations while retaining the structure of individual batches (Fig. 1b,f).

We corroborate the strong performance of OT-tSNE with a large benchmark comparing 23 method configurations for visualizing data with batch effect in 2D across 7 datasets. We found that OT-tSNE matches the state-of-the-art combination of batch mixing and preservation of data structure, particularly outperforming the competitors in terms of batch integration (Fig. 1g).

In sum, our contributions are

1. the first combination of optimal transport and neighbor embedding methods, OT-tSNE, for integrated visualizations of data with batch effect,
2. illustrating that nearest-neighbor-based methods can struggle even if the batch effect is orthogonal to the data, while OT-tSNE remains unaffected,
3. an extensive benchmark for visualizing batched data, highlighting the strong performance of OT-tSNE especially for batch mixing,
4. an open-source implementation of OT-tSNE based on the `pot` and `opentsne` packages.

2 Related work

Our work draws on two main strands of research, visualization and optimal transport. Spectral visualization approaches, such as Laplacian Eigenmaps (Belkin & Niyogi, 2003) or Diffusion Maps (Coifman & Lafon, 2006), lay out a graph approximation of the data via the eigenvectors of the graph's Laplacian. Neighbor embeddings, like t-SNE (van der Maaten & Hinton, 2008) and UMAP (McInnes et al., 2018), often produce visualizations with finer resolution and have thus become the de facto standard for non-linear visualization.

Optimal transport (OT) (Peyré & Cuturi, 2019) computes how to transport mass from one dataset to another at minimal cost. It became popular in machine learning with Cuturi (2013)'s seminal paper on the fast computation of entropically regularized optimal transport with the Sinkhorn-Knopp algorithm. SNEkhorn (Van Assel et al., 2023) used optimal transport maps for visualization by replacing the neighbor-based t-SNE affinities with optimal transport plans. This has some resemblance to our proposed OT-tSNE, but does not address batch integration.

The ability of optimal transport to meaningfully relate different datasets makes it well-suited to align datasets or remove batch effects. Courty et al. (2016) mapped images from a source domain to the average of images in a target domain, weighted by the transport plan. Similarly, Demetci et al. (2022) used optimal transport to align multimodal single-cell data. Landa et al. (2024)'s entropic OT eigenmaps use an entropic OT (EOT) plan to align two datasets, thus extending Laplacian Eigenmaps and Diffusion Maps to data alignment. Like Laplacian Eigenmaps and Diffusion Maps, it struggles to compress rich information into the very low dimensionality needed for visualization.

Batch correction and integration are major topics in the analysis of single-cell RNA-sequencing (scRNA-seq) data (Luecken et al., 2022). Methods like scVI (Lopez et al., 2018) and LIGER (Welch et al., 2019) learn to reconstruct cells from batch-aware encoding models. Most batch correction approaches infer cross-batch associations, often in a dimensionality-reduced space. Harmony (Korsunsky et al., 2019) tries to infer compact cross-batch clusters. MNN (Haghverdi et al., 2018), Seurat (Stuart et al., 2019), Scanorama (Hie et al., 2019) and others leverage mutual nearest neighbors (MNNs) across batches to estimate local batch-effect vectors. A few very recent methods use optimal transport as part of their batch integration pipeline to determine the integration order for multiple batches (Houdjedj et al., 2025), to regularize adversarial integration (Jiang et al., 2026), or as pre-alignment before the main contrastive VAE integration step (Zhang & Qin, 2026). None of them uses optimal transport as the main alignment ingredient, or produces 2D visualizations.

Some batch correction methods output a graph containing both within-batch affinities and cross-batch affinities. These methods are similar in spirit to ours, especially since the resulting graphs are often visualized with UMAP. However, their cross-batch edges are kNN- (Polański et al., 2020, BBKNN) or MNN-based (Barkas et al., 2019, CONOS) and can thus lead to imbalanced cross-batch affinities and the kissing problem (Lun, 2019).

Closest to our work are approaches that directly produce an integrated 2D visualization. Poličar et al. (2023) propose to use the t-SNE embedding of one batch as a reference for the other batch. Compound-SNE (Cess & Haghverdi, 2024) relies on knowledge of shared cell types across different batches and alternates Procrustes alignment to a reference with t-SNE embedding optimization.

Other methods are symmetric in their treatment of batches. Batch-corrected t-SNE (Aliverti et al., 2020, BC t-SNE) first linearly integrates the input data to make it uncorrelated with the nuisance label and then restricts the t-SNE gradient updates to be orthogonal to the nuisance label. Conditional t-SNE (Kang et al., 2021) incorporates the batch information into t-SNE's embedding affinities so that points from different batches move closer together. Revised conditional t-SNE (Heiter et al., 2023, RC t-SNE) instead changes the input affinities to include inter-batch affinities. MultiMAP (Jain et al., 2021) extends UMAP to a multi-dataset visualization method, equipping a graph similar to the BBKNN graph with UMAP-like weights. Deep Visualization (Xu et al., 2023) trains a neural network to produce visualizations with a UMAP-like loss and uses the batch information in the embedding affinities, similar to conditional t-SNE. Our method OT-tSNE also adapts t-SNE for batch-corrected visualizations, but relies on optimal transport to provide cross-batch affinities.

3 METHOD

t-SNE background. t-SNE constructs a low-dimensional representation $Y = (y_1, \dots, y_n)^\top \in \mathbb{R}^{n \times d'}$ of a high-dimensional dataset $X \in \mathbb{R}^{n \times d}$, with $d' \ll d$. It does so by aligning pairwise similarity distributions in the high- and low-dimensional spaces (van der Maaten & Hinton, 2008). The high-dimensional similarities p_{ij} are symmetrized Gaussian affinities with locally adapted bandwidths, whereas the low-dimensional similarities are defined using a normalized Cauchy kernel

$$q_{ij} = (1 + \|y_i - y_j\|^2)^{-1}/Z, \qquad Z = \sum_{kl} (1 + \|y_k - y_l\|^2)^{-1}.$$

Previous work has shown that replacing the Gaussian affinities with a binary shared k-nearest-neighbor graph produces very similar t-SNE embeddings (Artemenkov & Panov, 2020; Böhm et al., 2022; Damrich et al., 2023). To align the similarities and thus compute the embedding coordinates Y, t-SNE minimizes the Kullback-Leibler divergence between q and p

$$\mathcal{L}(Y) = \mathrm{KL}\big(p \| q(Y)\big) = \sum_{ij} p_{ij} \log(p_{ij}/q_{ij}). \tag{1}$$

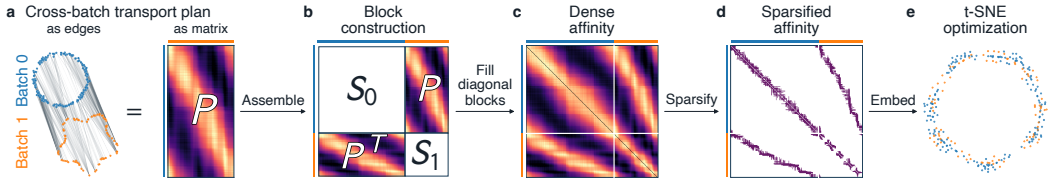

Figure 2: OT-tSNE method overview. **a.** Toy dataset with batch effect and selected optimal transport cross-batch affinities shown as edges (left), with the full plan shown as matrix P (right). **b.** Construction of the four-block joint affinity matrix for the t-SNE backend. The plans $P, P^\top$ form the off-diagonal blocks. **c.** Completing the on-diagonal blocks S_0 and S_1 to reflect similarity of points in the same batch. **d.** Batch-size-aware sparsification of the dense joint affinities to make them suitable for t-SNE optimization. **e.** Integrated OT-tSNE embedding obtained via t-SNE optimization.

OT-tSNE. Let $X_b = (x_1^{(b)}, \dots, x_{n_b}^{(b)})^\top \in \mathbb{R}^{n_b \times d}$ for $b = 0, 1$ denote two batches from the same dataset. OT-tSNE modifies only t-SNE's high-dimensional similarities, while leaving the low-dimensional Cauchy similarities and the objective unchanged. To define these similarities jointly across the two batches, we construct an affinity matrix $A \in \mathbb{R}^{(n_0+n_1)\times(n_0+n_1)}$ and set $p_{ij} = A_{ij} / \sum_{kl} A_{kl}$. The matrix A has a four-block structure, with within-batch affinities on the diagonal and cross-batch affinities in the off-diagonal blocks. Fig. 2 shows the schematic of OT-tSNE.

Cross-batch affinities. The cross-batch affinity $P \in \mathbb{R}^{n_0 \times n_1}$ is given by the entropic optimal transport (EOT) plan between X_0 and X_1 (Cuturi, 2013; Peyré & Cuturi, 2019). Let $C_{ij} = \|x_i^{(0)} - x_j^{(1)}\|_2^2$ denote the Euclidean cross-batch cost matrix. The EOT plan is defined as

$$P = \arg\min_{\Gamma \geq 0} \sum_{ij} C_{ij}\Gamma_{ij} + \epsilon \sum_{ij} \Gamma_{ij} \log \Gamma_{ij} \quad \text{s.t.} \quad \Gamma \mathbf{1} = \frac{1}{n_0}\mathbf{1},\ \Gamma^\top \mathbf{1} = \frac{1}{n_1}\mathbf{1}. \tag{2}$$

The resulting plan forms the off-diagonal blocks of the joint affinity matrix in Fig. 2. The marginal constraints ensure that the cross-batch coupling P is globally balanced, so that all points are equally associated with the other batch. This is in contrast to local nearest-neighbor approaches that can lead to imbalanced associations with hubs and barely connected regions (Fig. 1d,e).

The hyperparameter ϵ controls the sparsity of the transport plan, with larger (smaller) ϵ yielding more uniform (concentrated) plans. We set $\epsilon = \text{median}_{ij}\, C_{ij}$, following Landa et al. (2024).

Within-batch affinities. Our within-batch affinities, S_0, S_1, are inspired by EOT eigenmaps (Landa et al., 2024), which leverage singular value decomposition of the cross-batch transport plan P

$$P = U\Sigma V^\top, \qquad U = [u_1, \dots, u_q], \quad V = [v_1, \dots, v_q], \quad \Sigma = \text{diag}(\sigma_1, \dots, \sigma_q), \tag{3}$$

where $q = \min(n_0, n_1)$, $u_i \in \mathbb{R}^{n_0}$, $v_i \in \mathbb{R}^{n_1}$, and σ_i are the left / right singular vectors, and singular values of P, respectively. The EOT eigenmap coordinates are then given by the first columns of

$$E = \begin{pmatrix} \sqrt{n_0}U \\ \sqrt{n_1}V \end{pmatrix}. \tag{4}$$

We construct a symmetric joint affinity matrix A whose eigenvectors are EOT eigenmap coordinates, while retaining P as its cross-batch affinities. Specifically, we set

$$A = \frac{1}{\sqrt{n_0 n_1}} E\Sigma E^\top = \begin{pmatrix} S_0 & P \\ P^\top & S_1 \end{pmatrix} \text{ with } S_0 = \sqrt{n_0/n_1}\, U\Sigma U^\top, S_1 = \sqrt{n_1/n_0}\, V\Sigma V^\top. \tag{5}$$

We use this EOT eigenmap-based completion as our default.

We consider four additional constructions of the within-batch affinities. **(i)** Setting the within-batch affinities to zero. **(ii)** Gram matrices $S_0 = PP^\top$ and $S_1 = P^\top P$ relate points within one batch by how many cross-batch assignments they share. From the singular value decomposition, they satisfy

$$PP^\top = U\Sigma^2 U^\top, \qquad P^\top P = V\Sigma^2 V^\top. \tag{6}$$

Therefore, unlike the default completion above, this gives different powers of the singular values in the within- and cross-batch blocks. We refer to this method as OT-tSNE with Gram diagonals. **(iii)** Another optimal transport-based affinity matrix is the entropic optimal transport plan from each batch to itself. We refer to this variant as self OT. **(iv)** Finally, we represent within-batch affinities using a shared k-nearest-neighbor graph. This is closest to neighbor embedding methods like t-SNE and UMAP. We compare the alternative within-batch constructions empirically in Section 5.

Batch-size-aware sparsification. The EOT plan, and consequently A, is dense, whereas the high-dimensional t-SNE affinities are sparse in practice. We therefore sparsify A by removing self-edges and retaining a neighborhood of size $k = 15$ for every observation. To avoid overrepresenting the larger batch, the neighbor budget is split according to the sizes of batches $b = \{0, 1\}$,

$$k_b \approx k\frac{n_b}{n_0 + n_1}, \qquad k_0 + k_1 = k. \tag{7}$$

Thus, for every observation, we keep its strongest k_0 affinities to batch 0 and strongest k_1 affinities to batch 1. We symmetrize the selected edges by retaining an edge whenever either endpoint selects the other. We then binarize the resulting graph and normalize its total affinity mass to one. The sparsification can introduce mild imbalance in the resulting affinity.

t-SNE. Finally, we supply the sparse affinity matrix to t-SNE via `openTSNE` (Poličar et al., 2024).

4 Experimental Setup

4.1 Datasets

In addition to a synthetic version of MNIST with batch effect, described in Section 5.1, we benchmark OT-tSNE and competitors on seven real-world single-cell datasets with batch effect. We collected these from prior works on batch effect removal (Tran et al., 2020; Landa et al., 2024) and added a subset of Yao et al. (2021)'s mouse brain atlas. The datasets vary in size from ≈ 4000 to $\approx 30\,000$ points. The batch effect ranges from replicated identical experimental conditions over varying biological signal (donor sex, treatment perturbation) to different single-cell measurement techniques. More details on the datasets can be found in Appendix A and Tab. S1.

4.2 Evaluation

We evaluate each joint embedding along two complementary objectives: batch mixing and conservation of biological structure, following Luecken et al. (2022). For each objective, we use two metrics, all scaled to $[0, 1]$ such that higher values indicate better performance.

Batch mixing. We quantify batch mixing with the kNN batch-effect test (Büttner et al., 2019, kBET) and the integration local inverse Simpson's index (Korsunsky et al., 2019, iLISI).
Biological conservation. We measure preservation of biological structure using neighborhood purity and the silhouette score (Rousseeuw, 1987).
We provide the precise metric definitions and implementation details in Appendix D.
Ranking. Following established practices for benchmark aggregation (Maier-Hein et al., 2018; Luecken et al., 2022), we normalize and aggregate metric values before ranking methods. Then, we averaged biological-conservation and batch-mixing metrics to obtain an overall score. We ranked methods within each dataset, and then averaged the resulting ranks across datasets. Additionally, we count how often a method obtains a top-3 rank per metric category and dataset, see Appendix D.1.

4.3 Compared methods

We compare OT-tSNE against prominent methods for batch integration and joint visualization, grouped according to how the final two-dimensional embedding is obtained.

Direct integration and visualization. We first consider methods dedicated to batch-corrected visualization. They directly construct a joint two-dimensional representation and therefore provide the closest comparison to OT-tSNE. These include MultiMAP (Jain et al., 2021), BC t-SNE (Aliverti et al., 2020), RC t-SNE (Heiter et al., 2023), and Ref t-SNE (Poličar et al., 2023).

General-purpose integration methods. We additionally evaluate prominent general-purpose integration and alignment methods: Scanorama (Hie et al., 2019), Seurat (Stuart et al., 2019), scVI (Lopez et al., 2018), LIGER (Welch et al., 2019), Harmony (Korsunsky et al., 2019), and EOT eigenmaps (Landa et al., 2024). We evaluate these methods in two regimes. First, we use each method to construct a higher-dimensional integrated representation, as is common for downstream analyses. We choose 50 dimensions in our experiments before visualizing them with t-SNE. Second, we construct the integrated representation directly in 2D, a more restrictive setting that tests whether it can serve as the final visualization. We denote these two regimes by the method name followed by 50D or 2D, respectively, and refer to EOT eigenmaps simply as EOT throughout the experiments.

No integration. Finally, we include joint PCA and joint t-SNE (van der Maaten & Hinton, 2008) as reference baselines, simply by concatenating the two batches and embedding them.

5 Results

5.1 Synthetic experiments

We first highlight the strength of optimal-transport-based cross-batch affinities. Nearest-neighbor affinities can become unreliable when faced with challenging batch effects. For instance, a well-known failure case is the "kissing problem", which is most immediate when the batch effect acts along signal directions (Fig. 1c–f). In high-dimensional data, however, the structure of interest often lies on a much lower-dimensional manifold, such that batch effects are often orthogonal to the biological signal. This assumption underlies several batch correction techniques (Haghverdi et al., 2018; Stuart et al., 2019; Hie et al., 2019). Here, we demonstrate that even in this setting, nearest-neighbor cross-batch affinities can struggle, while OT-tSNE is provably unaffected.

Balanced OT is invariant to orthogonal batch effects. Consider perturbing two datasets by nuisance vectors, $\widetilde{x}_i^{(0)} = x_i^{(0)} + n_i^{(0)}$ and $\widetilde{x}_j^{(1)} = x_j^{(1)} + n_i^{(1)}$. Assume that the nuisances are orthogonal to the data and to nuisances of the other dataset, i.e., all $n_i^{(0)}, n_j^{(1)} \in \mathbb{R}^d$ are orthogonal to all $x_i^{(0)}$ and $x_j^{(1)}$ and also each n_i^0 is orthogonal to each $n_j^{(1)}$. Then, the squared Euclidean cost used in Eq. 2 only changes by row- and column-wise offsets

$$\widetilde{C}_{ij} = C_{ij} + \|n_i^{(0)}\|_2^2 + \|n_j^{(1)}\|_2^2 + 2\left\langle x_i^{(0)} - x_j^{(1)},\, n_i^{(0)} - n_j^{(1)} \right\rangle - 2\left\langle n_i^{(0)}, n_j^{(1)} \right\rangle =: C_{ij} + \nu_i^{(0)} + \nu_j^{(1)}, \tag{8}$$

because all other terms are zero due to orthogonality. For any feasible coupling Γ in Eq. 2, the perturbed and unperturbed cost functions differ by a constant,

$$\sum_{ij} \widetilde{C}_{ij}\Gamma_{ij} = \sum_{ij} C_{ij}\Gamma_{ij} + \frac{1}{n_0}\sum_i \nu_i^{(0)} + \frac{1}{n_1}\sum_j \nu_j^{(1)}, \tag{9}$$

since the additional terms are fixed by the marginals. Hence, for fixed ϵ, the optimal EOT plan P is unchanged under this perturbation model. More general orthogonal nuisance models have been theoretically analyzed by Kuang & Tabak (2017) and Landa et al. (2024).

In contrast, nearest-neighbor affinities are sensitive to the nuisance magnitudes in Eq. 8, which can reorder the pairwise distances and hence change the local neighborhoods.

MNIST with orthogonal class-dependent batch effects. We apply the above noise setting to a subsample of MNIST and construct two batches using the leading PCA subspace to represent the dominant signal variation (details in Appendix B). The batches are perturbed in disjoint 30-dimensional trailing-PCA subspaces that are orthogonal to the signal subspace and to each other. Specifically, each digit class receives its own perturbation direction and magnitude,

$$\widetilde{x}_i^{(0)} = x_i + \lambda n_{c(i)}^{(0)}, \qquad \widetilde{x}_j^{(1)} = x_j + \lambda n_{c(j)}^{(1)}, \tag{10}$$

where $c(i)$ denotes the digit class of sample i and λ controls the batch-effect strength (Fig. 3a). Thus, different regions of the data experience batch effects of different directions and magnitudes. This is different from the usual orthogonality notion, where one batch is rigidly shifted relative to the other by some shared batch vector orthogonal to the signal subspace. Nevertheless, in our setup the batch effect is still confined to dimensions orthogonal to the main signal. Since we choose the perturbation from the trailing PCs, the orthogonal decomposition in Eq. 8 is only approximate.

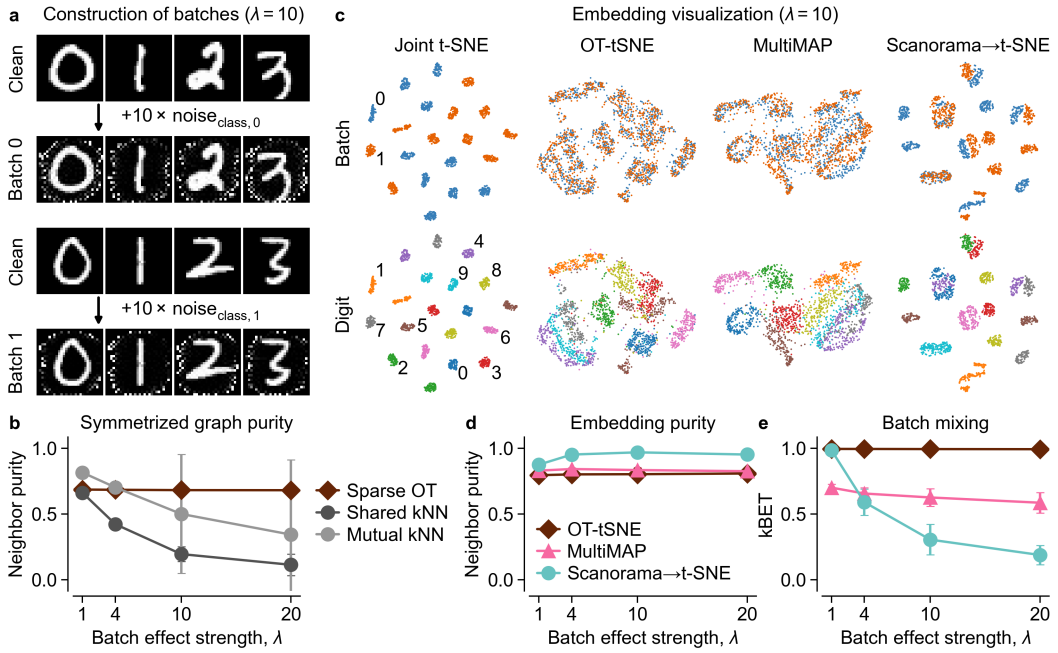

Figure 3: **a.** Construction of class-dependent orthogonal batch effects. **b.** Cross-batch graph purity for OT, shared kNN, and mutual kNN affinities. **c.** Example embeddings at $\lambda = 10$, colored by batch (top) and digit class (bottom). **d–e.** Embedding purity and batch mixing for the select integrated visualization methods across corruption strengths.

Cross-batch affinity stability. We compared sparsified EOT affinities with shared and mutual cross-batch kNN affinities using the same neighborhood size $k = 15$. For EOT affinities, we retained the top-15 cross-batch affinities per point, symmetrized the resulting graph and binarized it. We evaluated the affinities by the neighbor purity, defined as the fraction of retained neighbors with the same digit label, averaged over points with at least one retained edge (Fig. 3c). As the strength of the batch effect increases, both shared and mutual kNN affinities progressively deteriorated. For mutual kNN, this deterioration persisted despite conservative matching, as the fraction of points retaining a reciprocal cross-batch edge fell from 98% to 11% between $\lambda = 0$ and $\lambda = 20$. In contrast, the EOT affinities remained stable over the same corruption range.

Two-dimensional embeddings. We finally ask whether this difference in cross-batch affinities affects the resulting two-dimensional embeddings. We compare OT-tSNE with MultiMAP (Jain et al., 2021), whose integration graph uses shared cross-batch neighborhood information, and Scanorama (Hie et al., 2019) followed by t-SNE, which relies on mutual nearest-neighbor correspondences. OT-tSNE maintains nearly unchanged batch mixing and digit purity across the corruption range (Fig. 3d–e). Scanorama progressively deteriorates in batch mixing, while MultiMAP is more robust but also deteriorates as the batch effect increases. Digit purity remains high for all three methods, indicating that the dominant effect is a loss of cross-batch alignment. The visualization of embeddings at $\lambda = 10$ in Fig. 3b qualitatively reflects these trends.

Together, these experiments show that under heterogeneous orthogonal batch effects, balanced transport yields more stable cross-batch affinities than local neighbor matching, and that this difference persists in the resulting two-dimensional visualizations.

5.2 Benchmarking on real datasets

To test whether the results in the controlled setting extend to real data, we extensively benchmarked OT-tSNE on seven single-cell datasets with diverse batch effects and biological compositions.

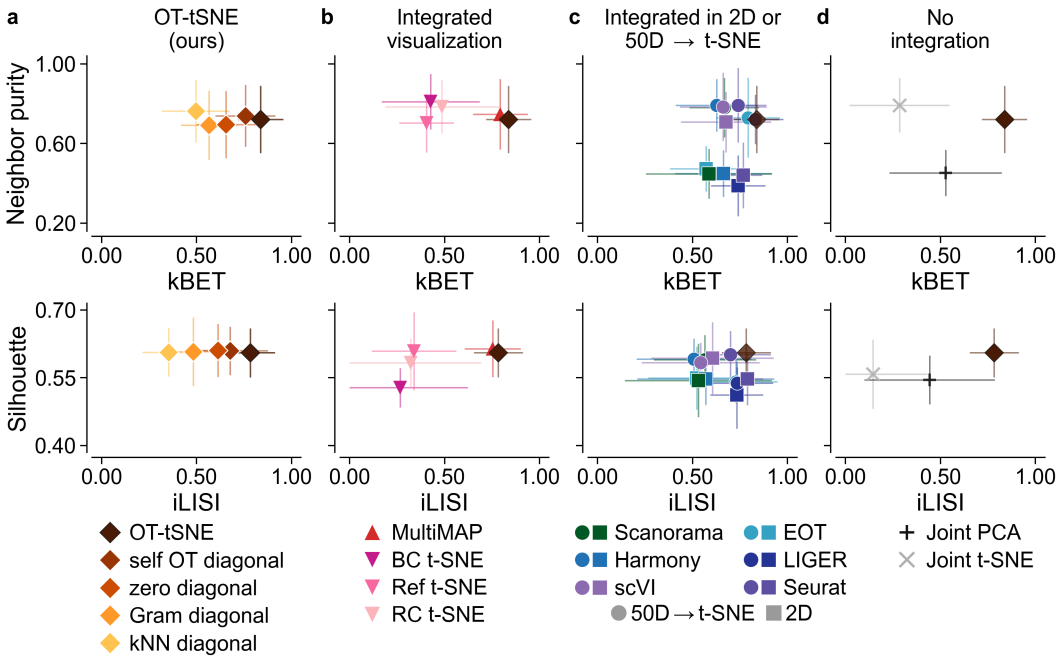

Figure 4: Pooled batch-mixing scores (x-axis; kBET or iLISI) versus biological-conservation scores (y-axis; purity or silhouette) across datasets for **a.** OT-tSNE variants, **b.** integrated visualization methods, **c.** general-purpose integration methods either with 50D latent spaces visualized with t-SNE (dots) or with 2D latent spaces directly used for visualization (squares), and **d.** baselines without integration. Reported means and standard deviations over datasets. OT-tSNE is repeated in all panels for better comparability.

Across the seven datasets, OT-tSNE achieved a strong balance between batch mixing and biological conservation. Its pooled scores were among the best methods for all metrics with particularly strong batch mixing scores, kBET and iLISI (Fig. 4). This performance was also consistent across datasets and metrics: OT-tSNE appears among the top three ranked methods more often than any competing method (Fig. 1g). The aggregate ranking over all four metrics likewise placed OT-tSNE first (Tab. 1).

The construction of the within-batch affinities mostly affected batch mixing. The five OT-tSNE variants had similar pooled purity and silhouette scores, whereas the default spectral construction attains the highest kBET and iLISI scores (Fig. 4a). We therefore use this construction as the default throughout the paper.

MultiMAP was the strongest competitor (Fig. 4b). It was slightly better in biological conservation, while OT-tSNE mixed batches better (Fig. S3). The other integrated visualization tools, BC t-SNE, RC t-SNE, and Ref t-SNE, integrated batches much worse.

General-purpose integration methods followed by t-SNE were competitive but less consistent across datasets (Fig. 4c). Some 50-dimensional integrations matched or exceeded OT-tSNE on individual biological-conservation metrics, but their performance varied substantially across metrics and datasets (Figs. S5 and S6). Adapting the methods to output integrated embeddings in 2D generally reduced biological conservation relative to integrating in 50 dimensions followed by t-SNE, with scVI-2D as a notable exception.

Table 1: Aggregated ranks across all four metrics and all datasets, lower is better.

Method	Rank
OT-tSNE	2.6
MultiMAP	3.6
OT-tSNE (self OT)	4.9
Seurat 50D	5.0
LIGER 50D	7.6
OT-tSNE (zero)	8.4
Scanorama 50D	8.6
Harmony 50D	9.0
EOT 50D	9.1
scVI 50D	10.0
scVI 2D	11.1
OT-tSNE (Gram)	12.1
RC t-SNE	12.4
OT-tSNE (kNN)	13.1
Seurat 2D	14.0
Ref t-SNE	16.9
Harmony 2D	16.9
LIGER 2D	17.1
BC t-SNE	17.7
Joint t-SNE	17.9
Scanorama 2D	18.3
EOT 2D	19.6
Joint PCA	20.1

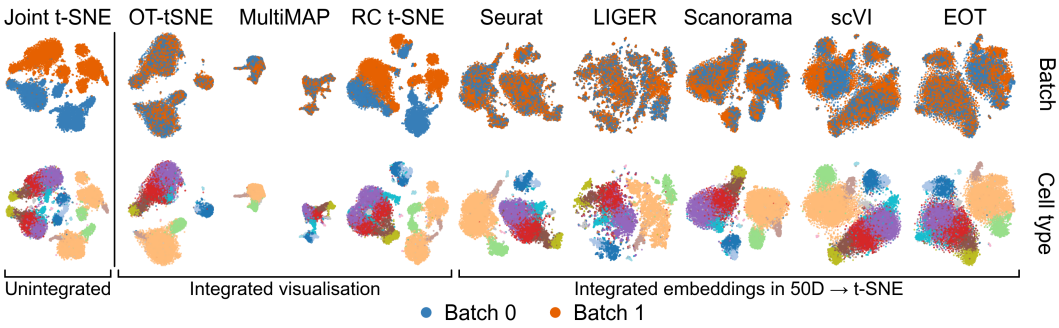

Figure 5: Lupus embeddings across selected methods. Top: colored by batch. Bottom: the same embeddings colored by annotated cell type. For visualizations of the other datasets, see Fig. S7.

Without batch integration, joint t-SNE largely preserved biological structure but left the batches separated. In contrast, joint PCA mixed the batches at the expense of biological conservation (Fig. 4d).

Embeddings of the selected methods on the Lupus dataset illustrate the corresponding trade-off between batch mixing and cell-type structure (Fig. 5).

The main exception is the hematopoietic dataset. Here, OT-tSNE retained comparatively strong batch mixing but lost substantially more biological structure than on the other datasets (Figs. S5 and S6). However, this dataset was generally more challenging to integrate for all the methods.

6 Limitations and Future Work

OT-tSNE relies on balanced optimal transport, which forces every observation in each batch to be equally associated with the other batch. This can be problematic when batches contain strongly different cell-type proportions or non-overlapping cell types. We assume that this is the main reason for weaker performance of OT-tSNE on the hematopoietic dataset, whose batches differ substantially in cell-type composition (see Fig. S1b for batch composition, Fig S5-S6). Unbalanced optimal transport (Chizat et al., 2018) provides a possible extension by relaxing the marginal constraints and thereby avoiding forced correspondences between populations that are absent from the other batch.

OT-tSNE is currently formulated for two batches. An easy multi-batch extension would compute pairwise transport plans between batches as off-diagonal blocks of a larger joint affinity matrix.

Runtime is another limitation. In our CPU-only benchmark, OT-tSNE was comparable to several competing methods on smaller datasets, but the affinity matrix creation increasingly dominated as dataset size grows (Fig. S8). Preliminary investigations pointed towards the SVD decomposition and reconstruction as worst offenders, which can be easily sped up with approximate versions. Beyond that, EOT's Sinkhorn iterations (Cuturi, 2013) can be accelerated on GPU. Landmark approximations (Altschuler et al., 2019) could further reduce the size of the transport problem.

Finally, unlike general-purpose integration methods, OT-tSNE produces a two-dimensional visualization rather than a reusable integrated representation for downstream analysis. A promising extension is to use OT-derived cross-batch correspondences as supervision for high-dimensional representation learning, for example through a contrastive objective.

7 Conclusion

We presented OT-tSNE for visualization of datasets with batch effects. Rather than using local cross-batch nearest-neighbor relations, OT-tSNE uses entropic optimal transport to construct high-dimensional cross-batch affinities directly visualized by t-SNE. Our controlled experiments showed that these affinities were robust under heterogeneous orthogonal batch effects, where nearest-neighbor affinities deteriorate. In a benchmark on seven real single-cell datasets, OT-tSNE consistently achieved strong batch mixing while also preserving biological structure well.

AI USE STATEMENT

In this work, we used generative AI tools for generating synthetic datasets (Appendix 5.1), designing or providing feedback on research methodology and experiments, implementing methods, cleaning and reformatting datasets (esp. joint preprocessing in Appendix A.1). We did not use generative AI tools to develop theoretical models or conceptual frameworks, formulate mathematical claims, propose or refine hypotheses, interpret results, or provide critical ingredients for proving mathematical claims. Assistance with writing proofs, qualitative or thematic data analysis, and translation were not applicable to this work.

Additionally, we used generative AI tools for creating and editing software code, drafting and editing parts of the manuscript, and formatting references. We manually edited all text afterwards. Similarly, all AI-assisted code and analyses were reviewed by the authors. We take responsibility for the final content of this work, including text, claims, code, analyses, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We ensure reproducibility by clearly describing our method (Section 3) and experiment setup, including datasets, metrics and competitors (Section 4) in the main paper. Additionally, we give further details on the single-cell datasets and preprocessing in Appendix A and describe the synthetic dataset construction in Appendix B. We dedicate Appendix C to implementation details of the benchmarked methods. Overall, we stuck to default hyperparameters as far as possible. We give full details on the evaluation metrics in Appendix D. Finally, we include our code as supplementary and will make it open source upon acceptance.

REFERENCES

Emanuele Aliverti, Jeffrey L Tilson, Dayne L Filer, Benjamin Babcock, Alejandro Colaneri, Jennifer Ocasio, Timothy R Gershon, Kirk C Wilhelmsen, and David B Dunson. Projected t-SNE for batch correction. *Bioinformatics*, 36(11):3522–3527, 2020.

Jason Altschuler, Francis Bach, Alessandro Rudi, and Jonathan Niles-Weed. Massively scalable sinkhorn distances via the nyström method. *Advances in neural information processing systems*, 32, 2019.

Seth A Ament, Rianne R Campbell, Mary Kay Lobo, Joseph P Receveur, Kriti Agrawal, Alejandra Borjabad, Siddappa N Byrareddy, Linda Chang, Declan Clarke, Prashant Emani, et al. The single-cell opioid responses in the context of hiv (scorch) consortium. *Molecular psychiatry*, 29(12): 3950–3961, 2024.

Aleksandr Artemenkov and Maxim Panov. NCVis: noise contrastive approach for scalable visualization. In *Proceedings of The Web Conference 2020*, pp. 2941–2947, 2020.

Nikolas Barkas, Viktor Petukhov, Daria Nikolaeva, Yaroslav Lozinsky, Samuel Demharter, Konstantin Khodosevich, and Peter V Kharchenko. Joint analysis of heterogeneous single-cell rna-seq dataset collections. *Nature methods*, 16(8):695–698, 2019.

Mikhail Belkin and Partha Niyogi. Laplacian Eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

Jan Niklas Böhm, Philipp Berens, and Dmitry Kobak. Attraction–repulsion spectrum in neighbor embeddings. *Journal of Machine Learning Research*, 23(95):1–32, 2022.

Maren Büttner, Zhichao Miao, F Alexander Wolf, Sarah A Teichmann, and Fabian J Theis. A test metric for assessing single-cell rna-seq batch correction. *Nature Methods*, 16(1): 43–49, 2019. doi: 10.1038/s41592-018-0254-1. URL https://doi.org/10.1038/s41592-018-0254-1.

Colin G Cess and Laleh Haghverdi. Compound-SNE: comparative alignment of t-snes for multiple single-cell omics data visualization. *Bioinformatics*, 40(7), 2024. doi: 10.1093/bioinformatics/btae471.

Lenaic Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. Scaling algorithms for unbalanced optimal transport problems. *Mathematics of computation*, 87(314):2563–2609, 2018.

Ronald R Coifman and Stéphane Lafon. Diffusion Maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30, 2006.

Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal transport for domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 39(9):1853–1865, 2016.

Marco Cuturi. Sinkhorn Distances: Lightspeed Computation of Optimal Transport. In *Advances in Neural Information Processing Systems*, volume 26, pp. 2292–2300, 2013.

Sebastian Damrich, Niklas Böhm, Fred A Hamprecht, and Dmitry Kobak. From t-SNE to UMAP with contrastive learning. In *The Eleventh International Conference on Learning Representations*, 2023.

Cyril de Bodt, Alex Diaz-Papkovich, Michael Bleher, Kerstin Bunte, Corinna Coupette, Sebastian Damrich, Enrique Fita Sanmartin, Fred A Hamprecht, Emőke-Ágnes Horvát, Dhruv Kohli, et al. Low-dimensional embeddings of high-dimensional data. *arXiv preprint arXiv:2508.15929*, 2025.

Pinar Demetci, Rebecca Santorella, Björn Sandstede, William S. Noble, and Ritambhara Singh. SCOT: Single-cell multi-omics alignment with optimal transport. *Journal of Computational Biology*, 2022.

Laleh Haghverdi, Aaron TL Lun, Michael D Morgan, and John C Marioni. Batch effects in single-cell rna-sequencing data are corrected by matching mutual nearest neighbors. *Nature biotechnology*, 36(5):421–427, 2018.

Xiaoping Han, Renying Wang, Yincong Zhou, Lijiang Fei, Huiyu Sun, Shujing Lai, Assieh Saadatpour, Ziming Zhou, Haide Chen, Fang Ye, et al. Mapping the mouse cell atlas by microwell-seq. *Cell*, 172(5):1091–1107, 2018.

Edith Heiter, Bo Kang, Ruth Seurinck, and Jefrey Lijffijt. Revised conditional t-sne: Looking beyond the nearest neighbors. In *Advances in Intelligent Data Analysis XXI*, pp. 169–181. Springer Nature Switzerland, 2023.

Brian Hie, Bryan Bryson, and Bonnie Berger. Efficient integration of heterogeneous single-cell transcriptomes using scanorama. *Nature biotechnology*, 37(6):685–691, 2019.

Aissa Houdjedj, Yacine Marouf, Mekan Myradov, Süleyman Onur Doğan, Burak Onur Erten, Oznur Tastan, Cesim Erten, and Hilal Kazan. Scituna: single-cell data integration tool using network alignment. *BMC bioinformatics*, 26(1):92, 2025.

Mika Sarkin Jain, Krzysztof Polanski, Cecilia Dominguez Conde, Xi Chen, Jongeun Park, Lira Mamanova, Andrew Knights, Rachel A Botting, Emily Stephenson, Muzlifah Haniffa, et al. Multimap: dimensionality reduction and integration of multimodal data. *Genome biology*, 22(1):346, 2021.

Chenghan Jiang, Ren Zheng, Yongle Ji, Shubin Cao, Yanhua Fang, Zhe Wang, Ruoyu Wang, Shanshan Liang, and Shuai Tao. Geometric-aware and interpretable deep learning for single-cell batch correction via explicit disentanglement and optimal transport. *bioRxiv*, pp. 2026–02, 2026.

Bo Kang, Dario Garcia Garcia, Jefrey Lijffijt, Raúl Santos-Rodríguez, and Tijl De Bie. Conditional t-SNE: More informative t-SNE embeddings. *Machine Learning*, 2021.

Hyun Min Kang, Meena Subramaniam, Sasha Targ, Michelle Nguyen, Lenka Maliskova, Elizabeth McCarthy, Eunice Wan, Simon Wong, Lauren Byrnes, Cristina M Lanata, et al. Multiplexed droplet single-cell rna-sequencing using natural genetic variation. *Nature biotechnology*, 36(1): 89–94, 2018.

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

Max Kuang and Esteban G Tabak. Preconditioning of optimal transport. *SIAM Journal on Scientific Computing*, 39(4):A1793–A1810, 2017.

Boris Landa, Yuval Kluger, and Rong Ma. Entropic optimal transport eigenmaps for nonlinear alignment and joint embedding of high-dimensional datasets. *arXiv preprint arXiv:2407.01718*, 2024.

Romain Lopez, Jeffrey Regier, Michael B Cole, Michael I Jordan, and Nir Yosef. Deep generative modeling for single-cell transcriptomics. *Nature methods*, 15(12):1053–1058, 2018.

Malte D Luecken, Maren Büttner, Kridsadakorn Chaichoompu, Anna Danese, Marta Interlandi, Michaela F Müller, Daniel C Strobl, Luke Zappia, Martin Dugas, Maria Colomé-Tatché, et al. Benchmarking atlas-level data integration in single-cell genomics. *Nature methods*, 19(1):41–50, 2022.

Aaron Lun. A description of the theory behind the fastmnn algorithm. `https://marionilab.github.io/FurtherMNN2018/theory/description.html`, 2019.

Lena Maier-Hein, Matthias Eisenmann, Annika Reinke, Sinan Onogur, Marko Stankovic, Patrick Scholz, Tal Arbel, Hrvoje Bogunovic, Andrew P. Bradley, Aaron Carass, et al. Why rankings of biomedical image analysis competitions should be interpreted with care. *Nature Communications*, 9(1):5217, 2018. doi: 10.1038/s41467-018-07619-7.

Leland McInnes, John Healy, and James Melville. UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

Sonia Nestorowa, Fiona K Hamey, Blanca Pijuan Sala, Evangelia Diamanti, Mairi Shepherd, Elisa Laurenti, Nicola K Wilson, David G Kent, and Berthold Göttgens. A single-cell resolution map of mouse hematopoietic stem and progenitor cell differentiation. *Blood, The Journal of the American Society of Hematology*, 128(8):e20–e31, 2016.

Franziska Paul, Ya'ara Arkin, Amir Giladi, Diego Adhemar Jaitin, Ephraim Kenigsberg, Hadas Keren-Shaul, Deborah Winter, David Lara-Astiaso, Meital Gury, Assaf Weiner, et al. Transcriptional heterogeneity and lineage commitment in myeloid progenitors. *Cell*, 163(7):1663–1677, 2015.

Gabriel Peyré and Marco Cuturi. Computational optimal transport with applications to data sciences. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

Krzysztof Polański, Matthew D Young, Zhichao Miao, Kerstin B Meyer, Sarah A Teichmann, and Jong-Eun Park. Bbknn: fast batch alignment of single cell transcriptomes. *Bioinformatics*, 36(3): 964–965, 2020.

Pavlin G. Poličar, Martin Stražar, and Blaž Zupan. Embedding to reference t-SNE space addresses batch effects in single-cell classification. *Machine Learning*, 112(2):721–740, 2023.

Pavlin G Poličar, Martin Stražar, and Blaž Zupan. openTSNE: a modular Python library for t-SNE dimensionality reduction and embedding. *Journal of Statistical Software*, 109:1–30, 2024.

Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

Tim Stuart, Andrew Butler, Paul Hoffman, Christoph Hafemeister, Efthymia Papalexi, William M Mauck, Yuhan Hao, Marlon Stoeckius, Peter Smibert, and Rahul Satija. Comprehensive integration of single-cell data. *cell*, 177(7):1888–1902, 2019.

Tabula Muris Consortium. Single-cell transcriptomics of 20 mouse organs creates a Tabula Muris. *Nature*, 562(7727):367–372, 2018.

Hoa Thi Nhu Tran, Kok Siong Ang, Marion Chevrier, Xiaomeng Zhang, Nicole Yee Shin Lee, Michelle Goh, and Jinmiao Chen. A benchmark of batch-effect correction methods for single-cell rna sequencing data. *Genome biology*, 21(1):12, 2020.

Hugues Van Assel, Titouan Vayer, and Rémi Flamaryand Nicolas Courty. SNEkhorn: Dimension Reduction with Symmetric Entropic Affinities. In *Advances in Neural Information Processing Systems*, volume 37, 2023.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

Joshua D Welch, Velina Kozareva, Ashley Ferreira, Charles Vanderburg, Carly Martin, and Evan Z Macosko. Single-cell multi-omic integration compares and contrasts features of brain cell identity. *Cell*, 177(7):1873–1887, 2019.

Yongjie Xu, Zelin Zang, Jun Xia, Cheng Tan, Yulan Geng, and Stan Z. Li. Structure-preserving visualization for single-cell RNA-Seq profiles using deep manifold transformation with batch-correction. *Communications Biology*, 2023.

Zizhen Yao, Cindy TJ Van Velthoven, Thuc Nghi Nguyen, Jeff Goldy, Adriana E Sedeno-Cortes, Fahimeh Baftizadeh, Darren Bertagnolli, Tamara Casper, Megan Chiang, Kirsten Crichton, et al. A taxonomy of transcriptomic cell types across the isocortex and hippocampal formation. *Cell*, 184(12):3222–3241, 2021.

Haihui Zhang and Peiwu Qin. Accurate, scalable, and unified single-cell atlas integration with scbiot. *bioRxiv*, pp. 2026–01, 2026.

Grace XY Zheng, Jessica M Terry, Phillip Belgrader, Paul Ryvkin, Zachary W Bent, Ryan Wilson, Solongo B Ziraldo, Tobias D Wheeler, Geoff P McDermott, Junjie Zhu, et al. Massively parallel digital transcriptional profiling of single cells. *Nature communications*, 8(1):14049, 2017.

Linnan Zhu, Penghui Yang, Yingze Zhao, Zhenkun Zhuang, Zhifeng Wang, Rui Song, Jie Zhang, Chuanyu Liu, Qianqian Gao, Qumiao Xu, et al. Single-cell sequencing of peripheral mononuclear cells reveals distinct immune response landscapes of covid-19 and influenza patients. *Immunity*, 53(3):685–696, 2020.

APPENDIX

A SINGLE-CELL DATASET DESCRIPTIONS

All but the Human brain datasets contain single-cell RNA-sequencing data.

Human brain contains single-nucleus RNA-seq data for $n = 9215$ cells from the same human postmortem insular-cortex sample, profiled using 10x Chromium v3 snRNA-seq (Ament et al., 2024). The two batches correspond to replicates 1 and 4 and the batch effect is only due to technical artifacts of measuring both batches independently with the same protocol. It is therefore quite small. We downloaded the data from `https://zenodo.org/records/17167270`.

nCov contains $n = 4117$ peripheral blood mononuclear cells (PBMCs) from a longitudinal study of a COVID-19 patient 5 of Zhu et al. (2020). The batches were sampled at days 1 and 13 after hospitalization and so that the batch effect is mainly due to the disease progression.

Lupus contains $n = 13\,000$ PBMCs from lupus patients (Kang et al., 2018). The batches correspond to the control and treatment cohorts.

Mouse Cortex contains the $n = 26\,588$ layer 2/3 intratelencephalic-projecting glutamatergic neurons of the entorhinal mouse cortex in Yao et al. (2021), i.e., subclasses L2 IT ENTl, L2 IT ENTm, L3 IT ENT, L2/3 IT ENTl. The batch effect corresponds to the sex of the mouse.

PBMC contains $n = 15\,476$ peripheral blood mononuclear cells of healthy donors recorded either with the 10x Chromium 3' or 5' chemistry from Zheng et al. (2017). The batch effect is thus due to measurement technology.

Mouse atlas contains $n = 6954$ cells from the two whole mouse atlases of Han et al. (2018) and Tabula Muris Consortium (2018). Only 11 shared cell types are used following Tran et al. (2020). This dataset contains a strong batch effect since Han et al. (2018) acquired the data via the Microwell-seq technology, while the Tabula Muris Consortium (2018) used SMART-seq2.

Hematopoeitic contains $n = 4649$ differentiating hematopoietic stem and progenitor cells from mouse bone marrow. Again the two batches come from different studies. The first batch is from Paul et al. (2015) who used the MARS-seq protocol. The second is from Nestorowa et al. (2016) who used SMART-seq2. We used the same subsets as Tran et al. (2020). In addition to the different measurement technologies, the two batches contain different cell types and even the shared ones have very different proportions (Fig. S1), making this a challenging dataset with strong batch effect.

The last three datasets (PBMC, mouse atlas, and hematopoetic) were used in Tran et al. (2020) and we downloaded them from the author's Docker image `https://hub.docker.com/r/jinmiaochenlab/batch-effect-removal-benchmarking`.

A.1 PREPROCESSING

We used `scanpy` to apply a unified preprocessing pipeline to all datasets. Our aim was to standardize the input across methods as much as possible and thereby disentangle the effects of preprocessing from those of the integration methods themselves. First, we concatenated the batches and removed cells with fewer than 10 detected genes and genes detected in fewer than three cells. We selected

Table S1: Single-cell datasets overview

Name	Description	Main batch effect	n_0	n_1	classes
Human brain	human insular cortex	technical replicates	3145	6070	9
nCov	PBMCs of COVID patient	days in hospital	2877	1240	13
Lupus	PBMCs of Lupus patients	treatment vs control	7451	6548	13
Mouse cortex	mouse entorhinal cortex	donor sex	11352	15236	22
PBMC	PBMCs	sc-technology	8098	7378	9
Mouse atlas	mouse atlas	sc-technology	4239	2715	11
Hematopoeitic	mouse hematopoiesis	sc-technology	2729	1920	7

1 000 highly variable genes using the `seurat_v3` flavor. Feature selection was performed on raw counts.

We retained the raw counts for the selected genes as input to methods that require count data: scVI (Lopez et al., 2018), Seurat (Stuart et al., 2019), LIGER (Welch et al., 2019), and Ref tSNE (Poličar et al., 2023). For these methods, internal feature selection was disabled so that all methods used the same 1,000-gene panel, while other method-specific preprocessing of the input was retained.

For the remaining methods, we normalized each cell to a total count of $10\,000$ and applied the natural-log transformation $\log(1 + x)$. Scanorama (Hie et al., 2019), MultiMAP (Jain et al., 2021), and BC tSNE (Aliverti et al., 2020) operated directly on this common log-normalized $1\,000$-gene representation, as these methods apply their own preprocessing in feature space. Finally, we fitted a single 50-dimensional PCA jointly to the concatenated batches to obtain a common PCA representation. OT-tSNE, EOT eigenmaps (Landa et al., 2024), joint t-SNE, Harmony (Korsunsky et al., 2019), and RC tSNE (Heiter et al., 2023) operated on the corresponding batches in this 50-dimensional representation. Overall, all methods used the same cells and selected genes while receiving the representation appropriate for their input requirements.

A.2 Batch compositions

The datasets vary in their cell-type composition across batches (Fig. S1–S2). All datasets except Hematopoietic contain shared cell types across batches, whereas Hematopoietic additionally contains batch-specific cell types. Cell-type proportions are relatively balanced across batches in some datasets (Lupus, Human Brain, Mouse cortex, PBMC), but are rather imbalanced in the other datasets (nCov, Hematopoietic, Mouse atlas). Consequently, batch effects can be entangled with biological composition, making integration particularly challenging for datasets with imbalanced or non-overlapping cell types.

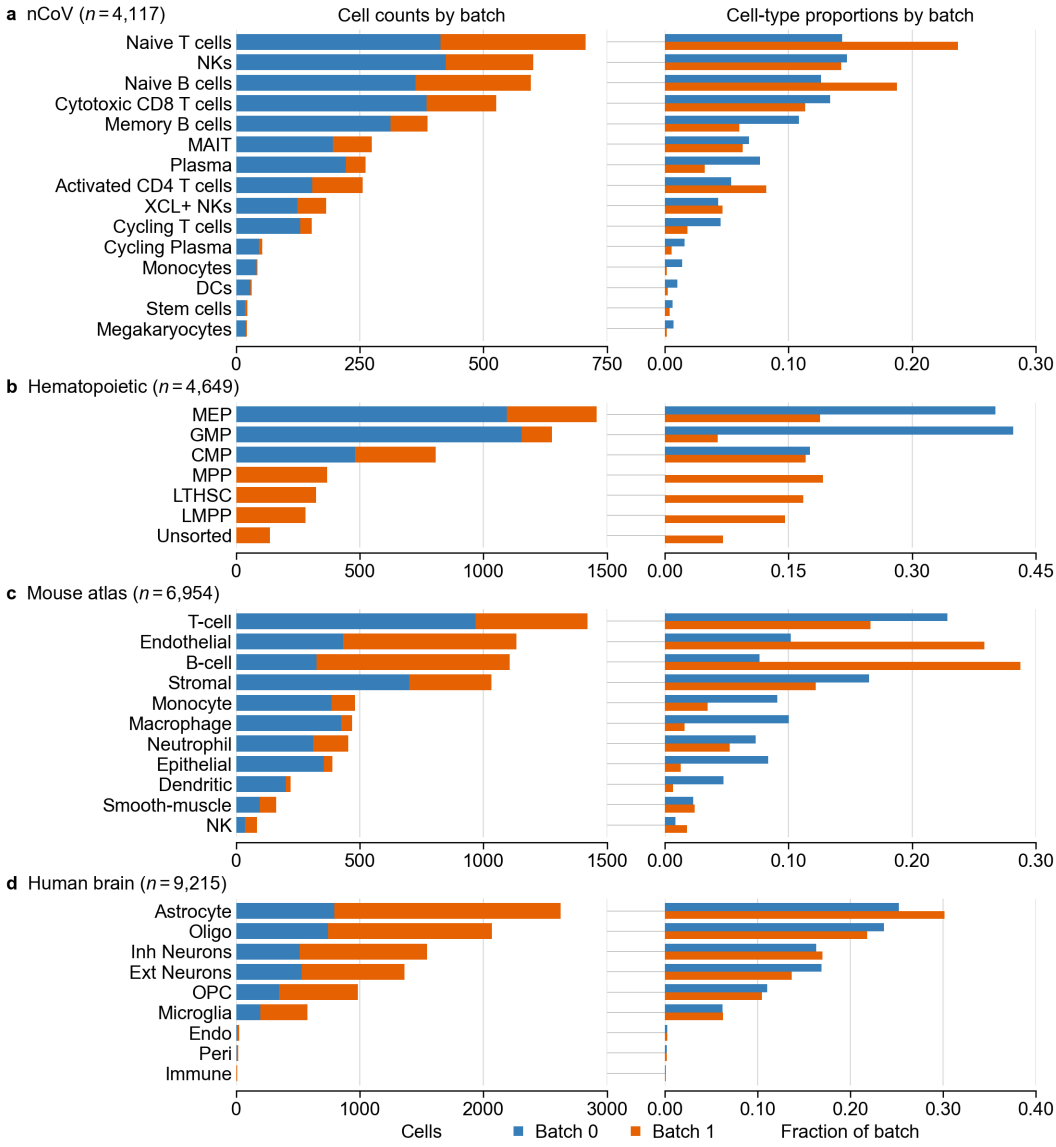

Figure S1: Batch compositions of the subset of single-cell datasets: nCoV, Hematopoietic, Mouse atlas, Human brain. Cell counts per cell type (left) and corresponding within-batch cell-type proportions (right).

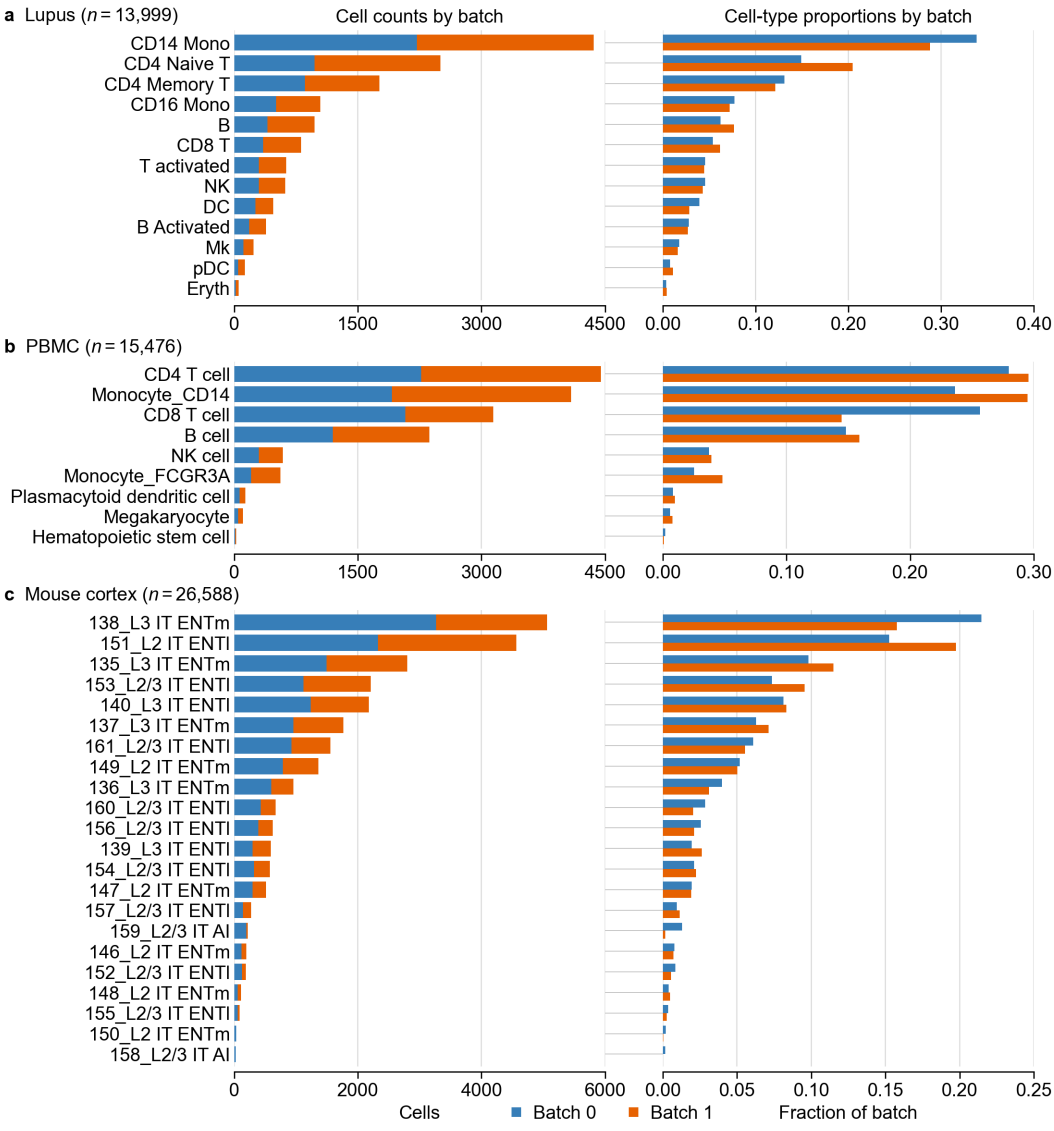

Figure S2: Batch compositions of the subset of single-cell datasets: Lupus, PBMC, Mouse cortex. Cell counts per cell type (left) and corresponding within-batch cell-type proportions (right).

B SYNTHETIC ORTHOGONAL BATCH EFFECT CONSTRUCTION

We sample two disjoint, class-balanced MNIST batches with 100 images per digit and fit PCA jointly to both batches. We take the last 60 numerically resolved PCs (explained variance $> 10^{-10}$ times that of the first PC) and split them into two disjoint 30-dimensional nuisance subspaces, one per batch.

For each class c and batch b, we sample a unit nuisance direction

$$q_c^{(b)} = \frac{\sum_{k=1}^{30} z_{ck}^{(b)} u_k^{(b)}}{\left\| \sum_{k=1}^{30} z_{ck}^{(b)} u_k^{(b)} \right\|_2}, \qquad z_{ck}^{(b)} \sim \mathcal{N}(0,1), \tag{11}$$

where $u_k^{(b)} \in \mathbb{R}^{784}$ are the corresponding PCA basis vectors. The class scales $s_c^{(b)}$ are independent permutations, for each batch, of ten values equally spaced between 0.35 and 2.60. We define

$$n_c^{(b)} = s_c^{(b)} q_c^{(b)} - \frac{1}{10} \sum_{r=0}^{9} s_r^{(b)} q_r^{(b)}, \tag{12}$$

so that the average nuisance within each batch is zero. The corrupted samples are $\widetilde{x}_i^{(b)} = x_i^{(b)} + \lambda n_{c(i)}^{(b)}$, with $\lambda \in \{1, 4, 10, 20\}$.

C IMPLEMENTATION DETAILS

OT-tSNE. We used k=15 neighbors, a commonly used neighborhood size in nearest-neighbor graph methods and the default in UMAP (McInnes et al., 2018). We set the entropic regularization hyperparameter to the median of the Euclidean cross-batch cost, following Landa et al. (2024). We computed the transport plan using Sinkhorn iterations with a convergence tolerance of 10^{-12} and a maximum of 10^4 iterations. For the openTSNE optimization of the constructed affinity, we used random initialization, and 200 iterations in total, comprising 66 early-exaggeration iterations followed by 134 post-exaggeration iterations.

t-SNE parameters for two-step embedding methods. For general-purpose integration methods that first construct a 50-dimensional representation and subsequently use t-SNE for visualization, we applied the same openTSNE configuration. Specifically, EOT 50D, PCA 50D, Harmony 50D, Scanorama 50D, scVI 50D, Seurat 50D, and LIGER 50D were embedded with perplexity 30, PCA initialization, 250 early-exaggeration iterations, and 200 post-exaggeration iterations.

Benchmarked methods. All competitors (EOT, Harmony, Scanorama, scVI, BC t-SNE, RC t-SNE, MultiMAP, Seurat, LIGER, and Ref t-SNE) used their default hyperparameters, except for the specified target dimensionalities (2 and 50) and random seed. Additionally, Seurat, LIGER, and Ref t-SNE required modifications to their standard input preprocessing to ensure consistent feature selection across methods, as detailed below.

Input formats. Different methods require different input representations. However, to minimize differences due to preprocessing and enable a more direct comparison of the integration methods, all methods used the same cells and the same set of $1\,000$ selected genes.

- OT-tSNE, EOT, joint t-SNE, Harmony, and RC t-SNE received the batches in the 50-dimensional PCA representation computed from the log-normalized $1\,000$-gene features.
- PCA, Scanorama, MultiMAP, and BC t-SNE received the log-normalized $1\,000$-gene representation, as these methods work directly in feature space.
- scVI, Seurat, LIGER, and Ref t-SNE normally receive raw counts and perform additional preprocessing internally. To maintain the same feature set across methods, we provided raw counts restricted to the common $1\,000$-gene panel. For Seurat, LIGER, and Ref t-SNE, we therefore disabled their internal gene-selection procedures while retaining the remainder of their standard workflows.

D EVALUATION METRICS

kBET. k-nearest neighbor batch effect test (Büttner et al., 2019, kBET) tests whether a cell's neighborhood has the expected batch composition . We evaluate each cell type separately and retain types with at least 45 cells in each batch. For every retained cell, we find its 15 nearest neighbors within the same type, excluding the query cell, and count from which batches they are. The expected count for each batch is 15 times its proportion within that cell type. We compare the observed batch counts with the expected counts using a Pearson chi-square test with one degree of freedom, since our datasets contain two batches. We calculate the fraction of tests with $p \geq 0.05$ within each type, then average these fractions equally across types. A higher score means fewer neighborhoods show a significant difference from the expected batch composition.

iLISI. Integration local inverse Simpson's index (Korsunsky et al., 2019, iLISI) measures the effective number of batches in a cell's neighborhood. We use $K = 90$ nearest neighbors in the joint embedding with a perplexity of 30. Let $\mathcal{N}_K(i)$ denote these neighbors, excluding cell i, and let d_{ij} be the Euclidean distance between cells i and j. We compute neighbor weights and local batch proportions as

$$w_{ij} = \frac{\exp(-\beta_i d_{ij})}{\sum_{r \in \mathcal{N}_K(i)} \exp(-\beta_i d_{ir})}, \qquad q_{ib} = \sum_{j \in \mathcal{N}_K(i)} w_{ij} \mathbf{1}\{b_j = b\}, \tag{13}$$

where b_j is the batch of cell j, and β_i is adjusted to achieve the target perplexity. With $B = 2$ batches, we compute

$$L_i = \left(\sum_{b=1}^{B} q_{ib}^2 \right)^{-1}, \qquad S_{\text{iLISI}} = \frac{\text{median}_i \, L_i - 1}{B - 1}. \tag{14}$$

As in the original iLISI implementation, the weights use unsquared Euclidean distances.

Neighborhood purity. Neighborhood purity measures the fraction of neighbors that share a cell's type. Let I_c be the set of n_c cells of type c. We evaluate types with at least five cells, $\mathcal{C} = \{c : n_c \geq 5\}$, using $k_c = \min(15, n_c - 1)$ neighbors:

$$S_{\text{purity}} = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \frac{1}{n_c} \sum_{i \in I_c} \frac{1}{k_c} \sum_{j \in \mathcal{N}_{k_c}(i)} \mathbf{1}\{c_j = c\}. \tag{15}$$

Here, c_j is the type of cell j. Neighbors are selected from the entire embedding, excluding the query cell. Cells from types with fewer than five cells remain candidate neighbors. We average first within each evaluated type and then across types, giving each type equal weight.

Silhouette score. The silhouette score measures cell-type separation (Rousseeuw, 1987). For cell i, let a_i be its mean Euclidean distance to other cells of the same type. Let h_i be the smallest of its mean Euclidean distances to each of the other types. We use scikit-learn to compute silhouette widths $s_i \in [-1, 1]$ for all n cells and rescale their mean to $[0, 1]$:

$$s_i = \frac{h_i - a_i}{\max(a_i, h_i)}, \qquad S_{\text{ASW}} = \frac{1}{2} \left(\frac{1}{n} \sum_{i=1}^{n} s_i + 1 \right). \tag{16}$$

Each cell has equal weight, so larger cell types contribute more to the final score.

D.1 RANKING PROCEDURE

For each dataset and regularization setting, we first average every metric across random seeds. Higher values indicate better performance for all four scores. We min–max normalize each metric across methods,

$$\tilde{m}_i = \frac{m_i - \min_j m_j}{\max_j m_j - \min_j m_j}.$$

Within each dataset, each method gets an overall score S_{overall}, defined as the average over all four normalized metrics. Then we rank the methods by S_{overall} within each dataset, with rank 1 denoting the best method. The final aggregate rank of a method is the mean of its dataset-level ranks (Tab. 1).

We also average the normalized metrics within the biological-conservation and batch-mixing groups to obtain scores S_{bio} and S_{mix}, respectively. For the top-3 finish ranking (Fig. 1g), we rank the methods separately by S_{bio} and S_{mix} within each dataset, and we count how often each method attains a top-3 rank across the seven datasets and these two scores.

We calculate the final aggregate ranks for S_{bio} and S_{mix} as described above. In contrast to overall ranking, these ranks reflect each method's relative biological-conservation or batch mixing quality.

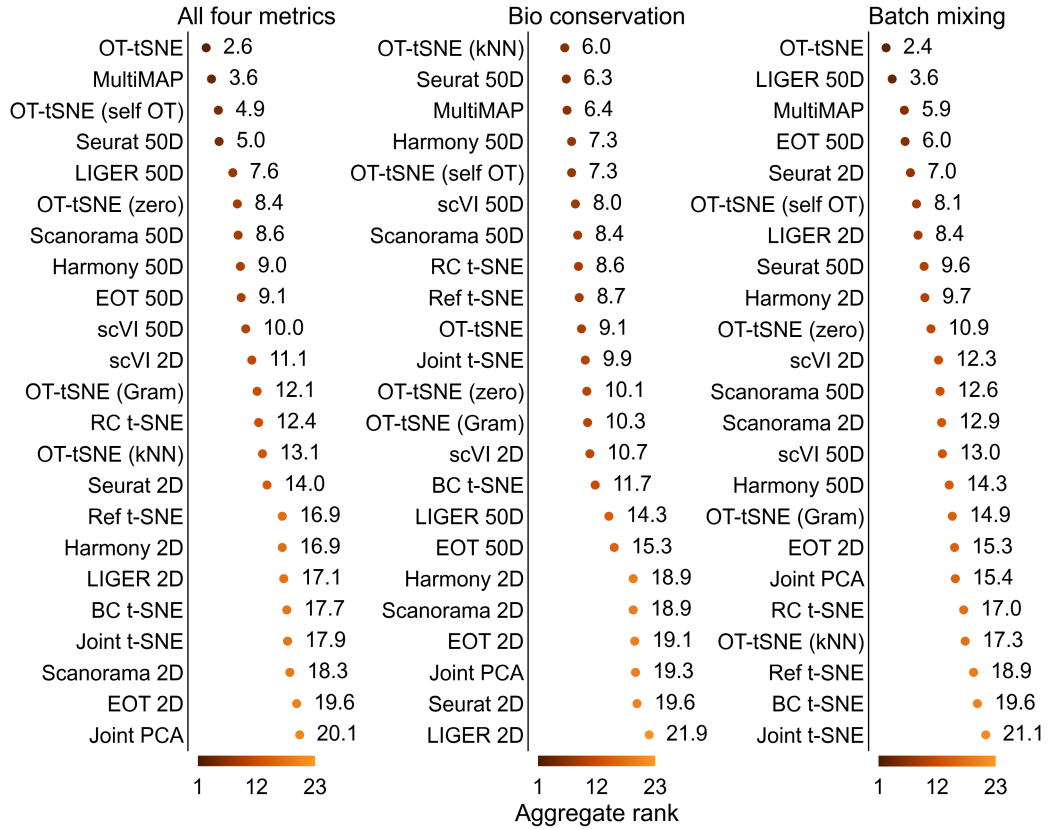

Figure S3: Ranks for all four metrics aggregated, only the aggregated bio-conservation metrics and only the aggregated batch mixing metrics, always across all datasets. Lower is better.

E ADDITIONAL FIGURES

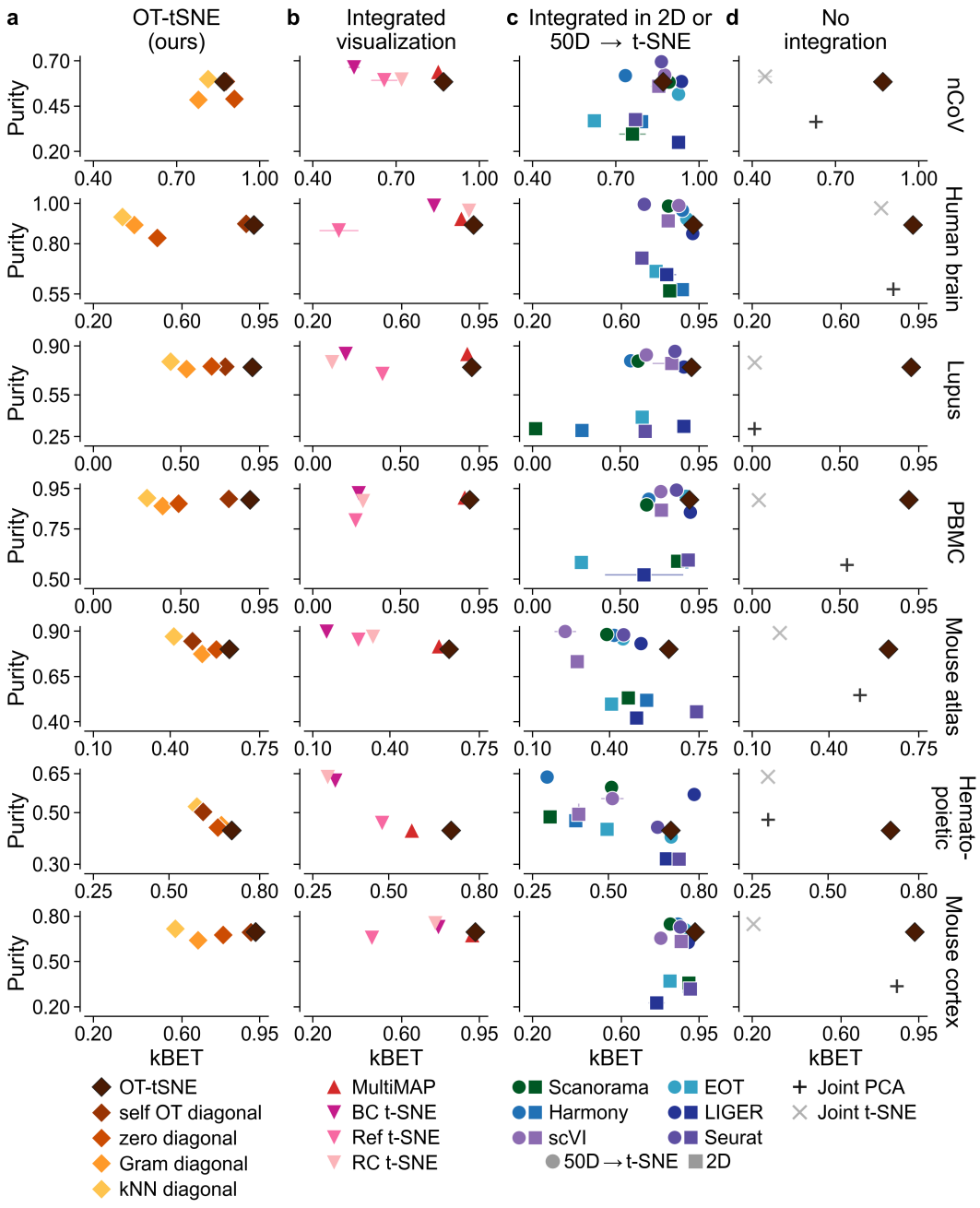

Figure S4: Benchmark results for all methods and datasets for the purity and kBET metrics for **a.** OT-tSNE variants, **b.** integrated visualization methods, **c.** general-purpose integration methods either with 50D latent spaces visualized with t-SNE (dots) or with 2D latent spaces directly used for visualization (squares), and **d.** baselines without integration. Reported means and standard deviations over datasets. OT-tSNE is repeated in all panels for better comparability.

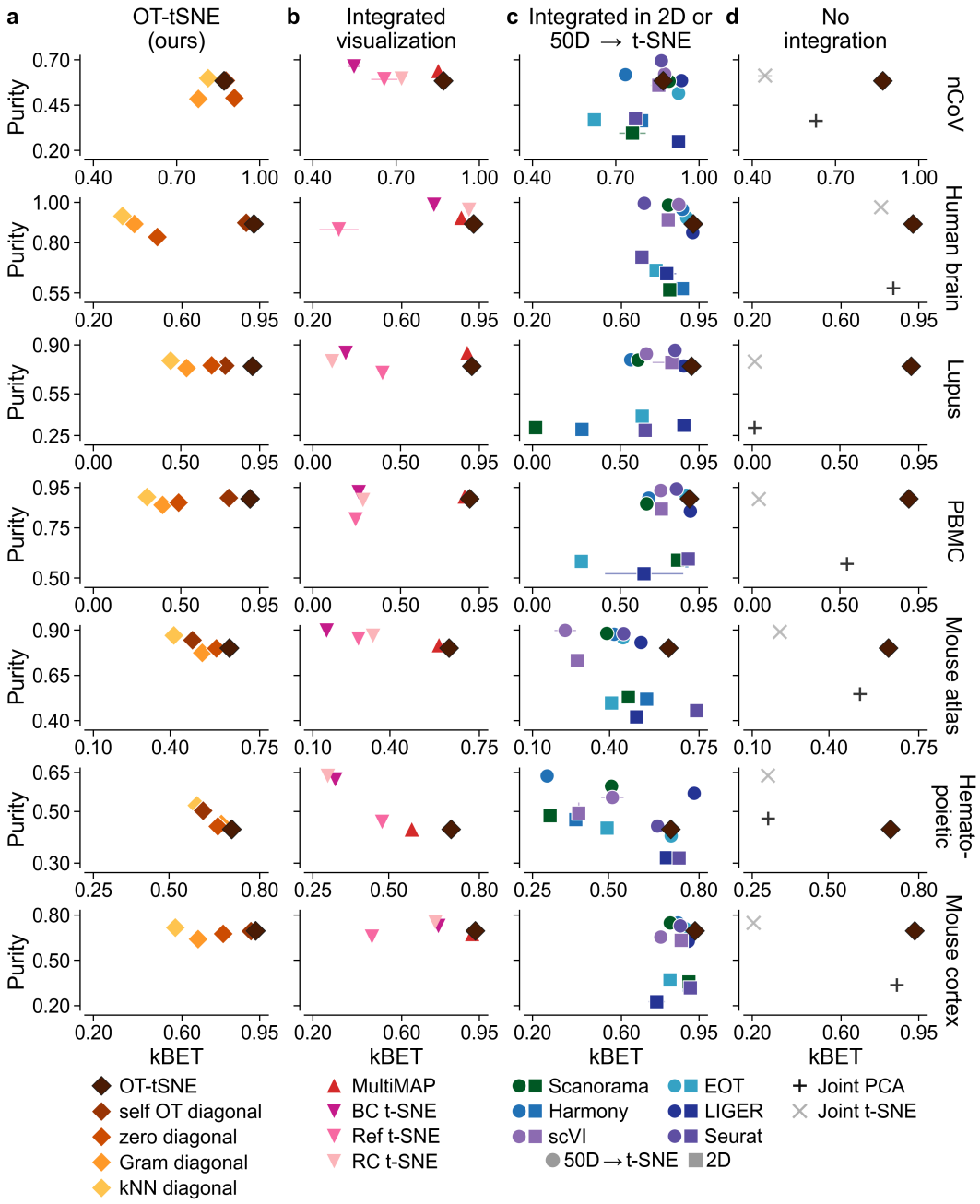

Figure S5: Benchmark results for all methods and datasets for the purity and kBET metrics for **a.** OT-tSNE variants, **b.** integrated visualization methods, **c.** general-purpose integration methods either with 50D latent spaces visualized with t-SNE (dots) or with 2D latent spaces directly used for visualization (squares), and **d.** baselines without integration. Reported means and standard deviations over datasets. OT-tSNE is repeated in all panels for better comparability.

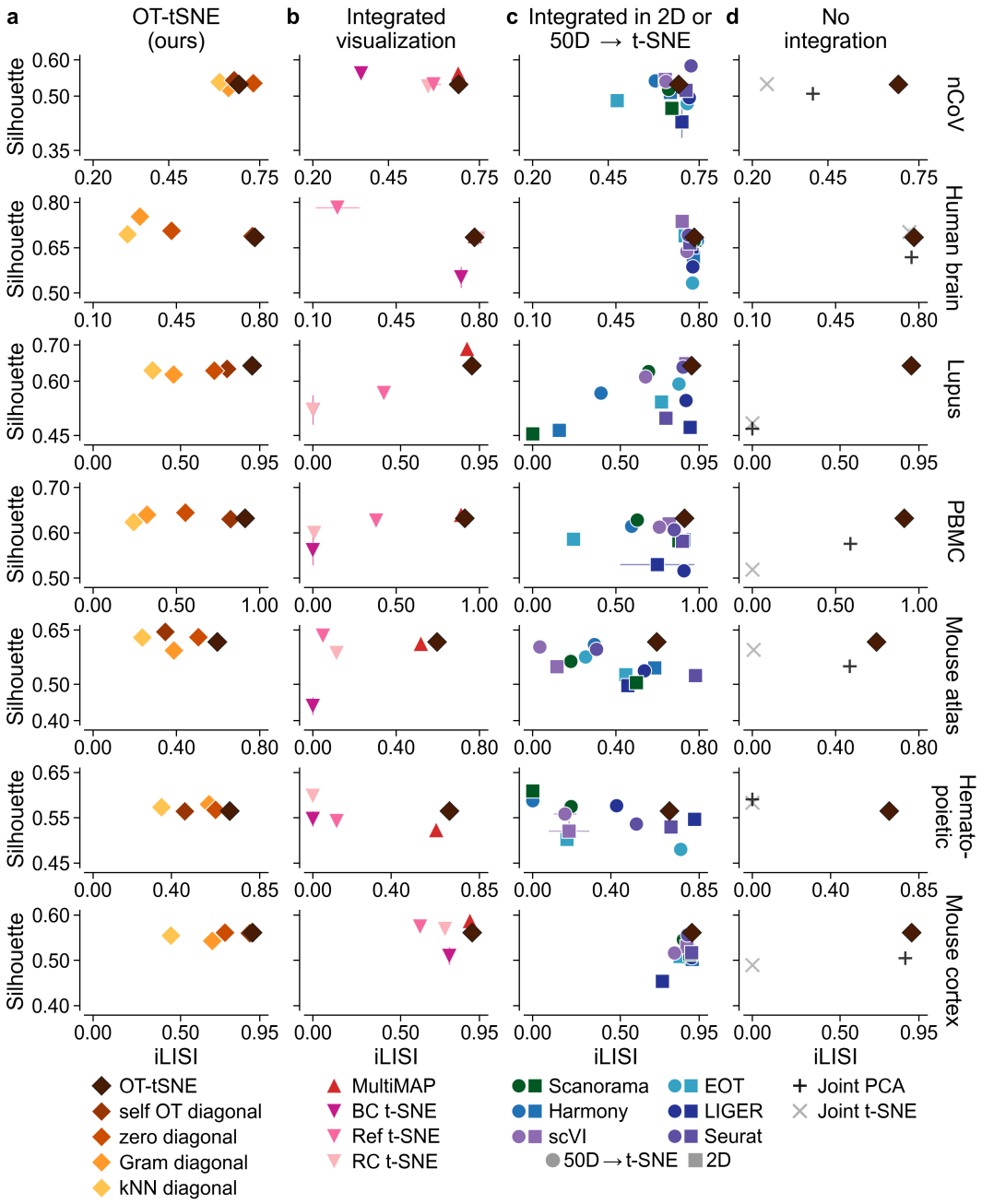

Figure S6: Benchmark results for all methods and datasets for the silhouette and iLISI metrics for **a.** OT-tSNE variants, **b.** integrated visualization methods, **c.** general-purpose integration methods either with 50D latent spaces visulized with t-SNE (dots) or with 2D latent spaces directly used for visualization (squares), and **d.** baselines without integration. Reported means and standard deviations over datasets. OT-tSNE is repeated in all panels for better comparability.

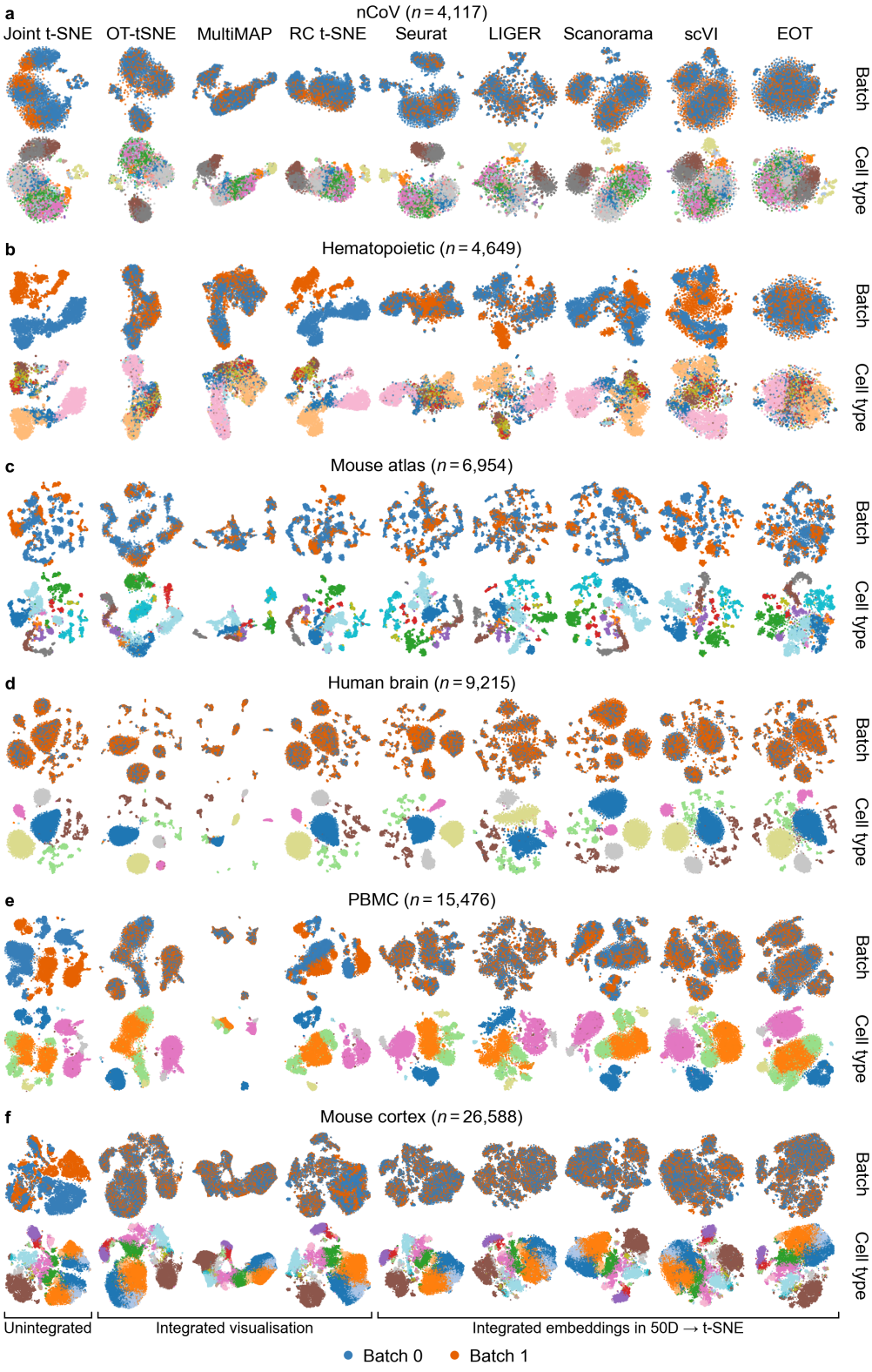

Figure S7: Visualizations of selected methods on the remaining 6 datasets, not shown in Fig. 5. Each embedding is shown twice, in the top row colored by batch, in the bottom row colored by cell type.

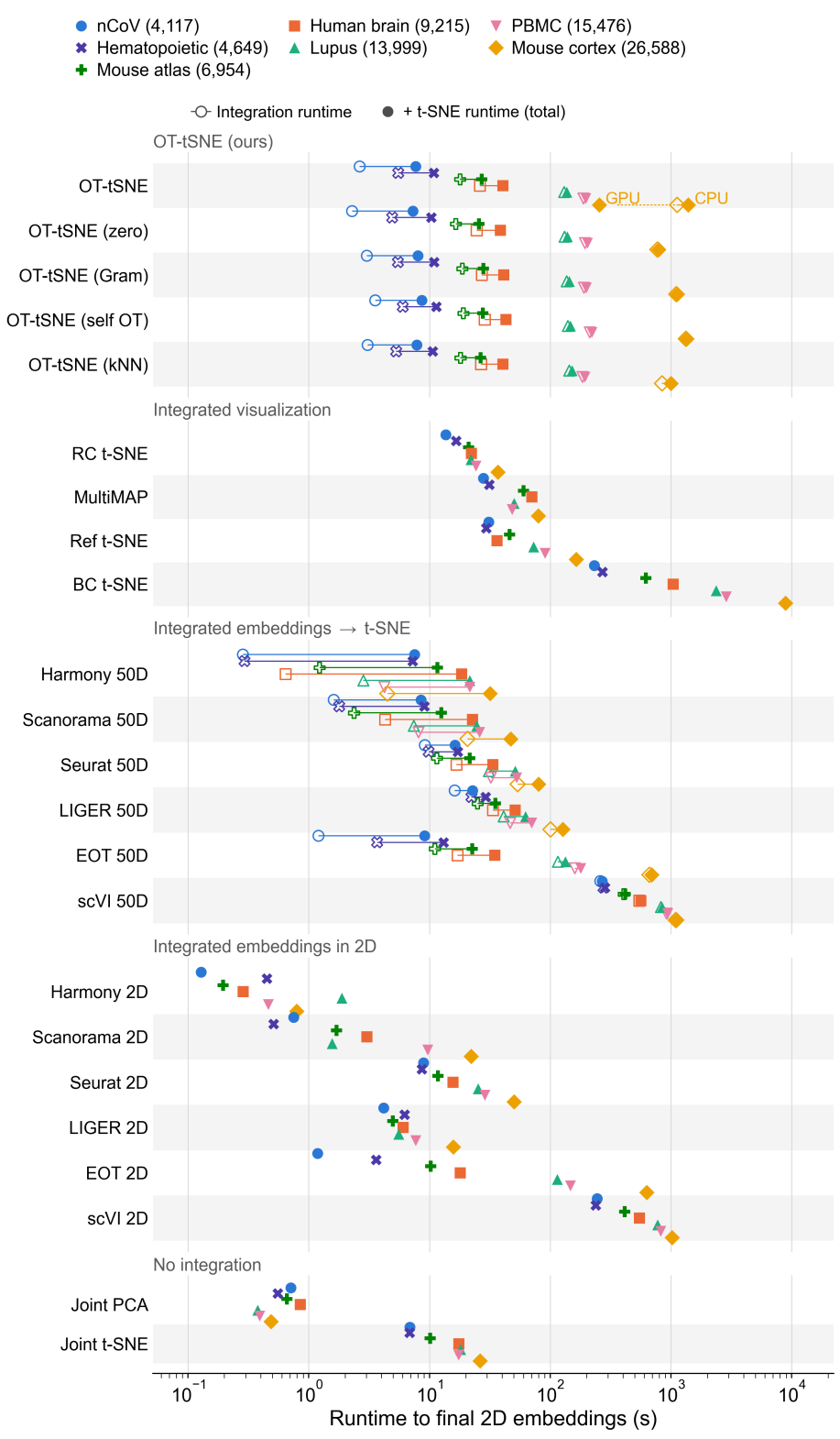

Figure S8: Median runtimes of all methods on all datasets over three random seeds. For OT-tSNE variants, we separate the creation of the affinity matrix from the t-SNE optimization. Similarly, for the general-purpose methods with subsequent t-SNE, we separate both phases. For OT-tSNE, GPU and CPU runtimes were compared on the largest Mouse cortex dataset.