RANKFUSE: INTEGRATING SINGLE-CELL REPRESENTATIONS THROUGH NEIGHBOUR RANKINGS

Anonymous authors
Paper under double-blind review

ABSTRACT

Single-cell encoders perform differently across datasets, but their embeddings use incompatible coordinate systems. We present RankFuse, an encoder-agnostic method that combines frozen single-cell encoders through each cell's ranked nearest neighbours across batches. Reciprocal rank fusion merges the per-encoder rankings into a single graph, and a one-layer graph convolution learns an integrated representation for batch integration, with a variance term to prevent collapse and a batch adversary to remove batch effects. No encoder is fine-tuned; the 1,640-parameter adapter and a training-only batch discriminator are trained on the dataset being integrated, and a new encoder is added by running one neighbour search and retraining the adapter. On PBMC and an immune cell atlas, fusing scGPT and Geneformer with two encoders pretrained on a 46-study corpus (a batch-adversarial scVI and a DINO encoder) improves unscaled scIB Total over the strongest single encoder by 5.6% and 2.0% relative and gives the strongest batch correction of the four on the atlas; on pancreas, the smallest dataset, it lands 0.003 below the best encoder. Ablations show that fusion exceeds scGPT with its own graph and the same adapter by 0.035 on the atlas and 0.012 on PBMC, and does not exceed it on pancreas. With target-trained encoders on PBMC, changing the graph encoders has a much larger effect than changing the feature encoders, supporting neighbour rankings as a coordinate-free interface for combining representation models.

1 INTRODUCTION

Measurements of single cells carry a technical signature from the experiment that produced them, and cells of one type from two experiments can resemble each other less than cells of two types from one. Removing that signature without removing the biology is the batch correction problem. Harmony aligns cells in a low-dimensional space by iterative clustering (Korsunsky et al., 2019), scVI models the counts generatively with the batch as a covariate (Lopez et al., 2018), and, for a different problem, Seurat's weighted-nearest-neighbour analysis combines modalities measured in the same cells (Hao et al., 2021). Benchmark rankings vary across datasets, and no method leads on all of them (Tran et al., 2020; Luecken et al., 2022). The arrival of self-supervised encoders has not changed this pattern; a recent benchmark of nineteen of them on single-cell data finds the same pattern (Ovcharenko et al., 2025). The advice that follows in practice is to try several encoders and keep whichever scores best on your own data.

Objectives developed for visual representation learning have motivated many of these encoders: contrastive learning as in SimCLR (Chen et al., 2020) and self-distillation as in DINO (Caron et al., 2021) both have single-cell counterparts. They give different representations of the same cells. In visual correspondence, pretrained DINOv2 and Stable Diffusion features are accurate in different respects, semantic matching for the first and spatial layout for the second, and fusing the two after independent normalisation outperforms either alone (Zhang et al., 2023). This suggests that encoders are worth combining when each is accurate where the other is not, which a difference in architecture does not by itself ensure.

Encoder embeddings differ in dimension, scale and coordinate system: two encoders place the same cells in spaces fitted under different objectives, with no correspondence between axes. Concatenating their coordinates yields a representation dominated by whichever block carries the most variance,

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.

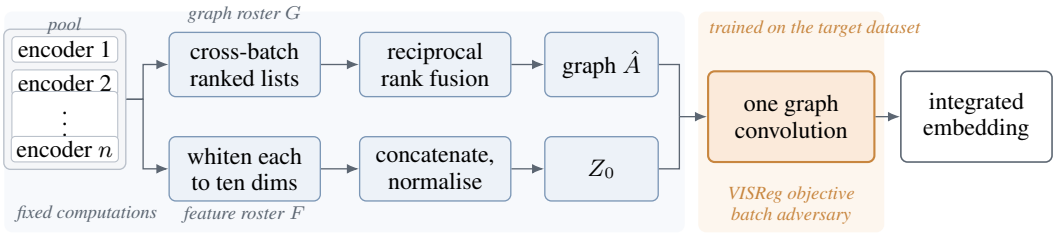

Figure 1: RankFuse. Every encoder in the pool ranks each cell's neighbours in the other batches. The rosters G and F split two roles: the lists of G are merged by reciprocal rank fusion into one neighbourhood graph, and the embeddings of F are whitened and concatenated into the input features Z_0. Shading and the heavy border mark off the trained representation layer, a graph convolution over that graph fitted with a variance-regularised objective and a batch-adversarial term, whose output is the integrated embedding. The two rosters coincide when the pool holds only four encoders.

and averaging them needs an alignment that does not exist. What they do share is an ordering: each can rank the cells most similar to a given cell, and a rank is comparable across encoders where a coordinate is not. In antibody–antigen affinity prediction, AbRank uses relative affinity comparisons to handle noisy measurements collected across assays (Liu et al., 2025), which motivates a related question for single-cell integration: can the ordering of neighbouring cells provide a useful basis for combining encoders whose coordinates are not directly comparable? The concrete aggregation we use is reciprocal rank fusion, developed in document retrieval to combine systems that share no scoring scale (Cormack et al., 2009).

RankFuse builds a single-cell integration method on that interface. It takes a pool of encoders applied to one dataset, has each of them rank every cell's cross-batch neighbours, merges those ranked lists by reciprocal rank fusion, builds one neighbourhood graph from the merged ranking, and trains a single graph convolutional layer (Kipf & Welling, 2017) over that graph to produce the output embedding. Reciprocal rank fusion combines rank position with repeated nomination, so a candidate several encoders rank highly accumulates support. Neither rank nor repeated support guarantees that a proposed neighbour shares the cell's biological identity, and we make no such claim.

The paper makes two contributions. The first is the method: a rank-based route from a heterogeneous pool of encoders to one integrated embedding, with a graph built by a fixed rule and a single trained layer over it. The second is a controlled evaluation that separates where its gains come from, across three datasets and two settings for the encoder pool, one in which the contributing encoders arrive as fixed checkpoints and one in which they are trained on the target dataset. The integration results are useful but uneven: the fusion improves on every encoder in its pool on two of three datasets in the fixed-checkpoint setting, and varies by dataset when the encoders are trained on the target data. The controls bound what that improvement shows. A single encoder given the same graph construction and the same adapter comes close to the four-encoder fusion on PBMC, does as well on pancreas, and falls well short of it on the atlas, so a gain over an unadapted encoder is not by itself a gain from fusion. The settings differ in their encoder pools; in both, the adapter is fitted on the target dataset using its batch labels.

2 RankFuse

RankFuse takes a pool of encoders that have each been applied to the same dataset, together with the batch label of every cell, and returns one embedding of those cells; Figure 1 summarises the pipeline. Given the selected rosters, every stage but the last is a fixed computation over the embeddings and the batch labels, and the last trains a single graph convolutional layer. Selecting the rosters in the larger pool reads cell-type labels, which the adapter's objective never does. Constants named below are held at the same values on all three datasets and their exact values are in Appendix S.2.2.

From rankings to a graph. Each encoder in the graph roster ranks candidate neighbours of every cell, drawn from the other batches. We merge those lists by reciprocal rank fusion (Cormack et al.,

Table 1: The three benchmark datasets. Cell and type counts are the scored population after the shared mask.

Dataset	Cells	Batches	Types	Source
Pancreas	16,382	9	14	six studies, six technologies (Luecken et al., 2022)
PBMC	30,449	8	9	six protocols (Ding et al., 2020)
Immune cell atlas	325,349	12	34	Domínguez Conde et al. (2022)

2009), scoring a candidate c' for a cell c as

$$s(c, c') = \sum_{t \in G} \frac{1}{K_0 + r_t(c, c')}, \qquad K_0 = 60, \tag{1}$$

where r_t is the rank of c' in encoder t's list for c and an encoder contributes nothing when the candidate is absent. High-ranked candidates therefore carry more weight, and no encoder needs a score comparable with any other's. We keep each cell's fifty highest-scoring fused neighbours as its edges and make the graph undirected by retaining an edge proposed by either endpoint. The graph is then fixed for the rest of the pipeline.

Features and the trained layer. The feature roster contributes equally sized blocks: each embedding is principal-component whitened to ten dimensions and normalised before concatenation, so that no block dominates the concatenation through its native scale (Appendices S.2.1 and S.2.4). Four blocks give a 40-dimensional input. A single linear projection followed by one hop of graph aggregation produces the integrated embedding, with no hidden layer and no nonlinearity anywhere in the adapter. We train that layer with the VISReg objective of Wu et al. (2026) on cell pairs sampled from the fused rankings, together with a batch discriminator behind a gradient reversal layer that discourages batch-predictive features. The objective is computed before the output is normalised, and its scale, shape and centring penalties are summed over both members of each pair. Training reads batch labels and no cell-type labels, and neither the graph nor the contributing encoders are learned. Appendix S.2.2 gives the parameter counts, the optimiser settings and the exact terms of the objective.

The two rosters. With four encoders, both rosters contain the whole pool. With thirteen we first designate a reference encoder, the member with the highest solo identity score, and then choose two rosters of four by minimising the mean pairwise overlap of their candidate lists: the graph roster excludes the reference and the feature roster contains it, so the strongest single encoder is kept out of the graph and put into the features. We call this minimum-overlap roster selection. List overlap serves as a proxy for the breadth of candidates retrieved by a roster; candidate coverage is neither optimised nor measured. Comparisons between the two settings therefore differ in the selection step as well as in the encoders. Designating the reference reads cell-type labels, through the solo identity score, and the adapter's objective never does; Section 3 states where labels enter the experiment as a whole, and Appendix S.6.2 gives the selection rule, its enumeration and a worked example.

3 Experimental setup

Datasets. We evaluate on three human single-cell RNA-sequencing datasets, listed in Table 1. The selection follows scSSL-Bench's batch-integration task (Ovcharenko et al., 2025), and PBMC and the immune cell atlas are the datasets that benchmark uses. Pancreas is not. We use scIB's `human_pancreas_norm_complexBatch` (Luecken et al., 2022), a composite of six studies whose nine batches span six sequencing technologies, whereas scSSL-Bench's appendix describes a different pancreas collection, the 14,767-cell five-study release of Tran et al. (2020). The two differ in cells, batches and types, so our pancreas results are not comparable with theirs, and we name the build as well as its constituent studies for that reason. Appendix S.4.1 gives the study-by-study attribution.

Preprocessing and the scoring basis. Scoring is carried out on one preprocessing basis throughout, which we write as pp4000: counts normalised to 10,000 per cell, log1p-transformed and reduced

to 4,000 batch-aware highly variable genes. This expression matrix is used for scIB metrics that require one, including principal component regression. It is not a claim about every encoder's input: the two foundation models consume their own gene vocabularies and tokenisations at inference, so what the comparison shares is the scoring basis and the cell population, not the input each encoder reads.

Two settings for the encoder pool. The paper reports two settings and they differ in where the contributing encoders were trained. In the *fixed-checkpoint* setting the pool holds four encoder checkpoints applied without any fine-tuning on the dataset being scored: the published scGPT whole-human checkpoint (Cui et al., 2024), Geneformer V1-10M (Theodoris et al., 2023), and two encoders of our own trained on scREF (Diaz-Mejia et al., 2025), a collection of 46 public studies each identified by PubMed ID, all four producing a cell embedding from a final layer. Pretraining overlap is audited for the scREF half of that pool only: the source studies of PBMC and of the atlas are absent from its training manifest, as are five of the six pancreas studies, and the sixth has not been checked. For scGPT and Geneformer the corpora were not audited at this resolution. Appendix S.4.1 gives the audit and its limits. In the *target-trained* setting the pool holds thirteen encoders, all trained on the dataset they are scored on, which is scSSL-Bench's convention for this task. That benchmark evaluates nineteen self-supervised methods in several families; our pool takes all seven of its generic ones, namely SimCLR (Chen et al., 2020), MoCo (He et al., 2020), SimSiam (Chen & He, 2021), NNCLR (Dwibedi et al., 2021), BYOL (Grill et al., 2020), VICReg (Bardes et al., 2022) and Barlow Twins (Zbontar et al., 2021), together with Concerto (Yang et al., 2022), which the benchmark groups with its domain-specific contrastive methods, and scVI (Lopez et al., 2018). Its remaining methods are not in our pool, so the comparisons below are against these baselines and not against the benchmark's full set. To them we add four batch-aware models we train ourselves, a batch-adversarial scVI written BA-scVI (Diaz-Mejia et al., 2025), the same with an added reconstruction-contrast term, written BA-scVI + RecCon, a self-distillation encoder in the manner of DINO (Caron et al., 2021) whose teacher logits are centred within each study, written DINO (batch-centred), and a variant of BA-scVI whose encoder sums expression-weighted gene embeddings with no attention blocks, written BA-scVI (gene-token). We train those four from scratch on each dataset on pp4000, taking the embedding at the end of training rather than a best-metric checkpoint, with the protocol in Appendix S.2.3.

Where labels enter. Cell-type labels enter in three places: evaluation, the early-stopping rule of the four batch-aware models we train, and selection of the reference encoder. No training objective anywhere in the pipeline fits parameters to them. The stopping criterion for the four models we train reads them: early stopping uses the identity score and cross-batch accuracy, both computed from the labels of the cells the model is later scored on, since the benchmark provides no held-out split. Roster selection reads them a third time, through the solo identity score that designates the reference encoder b. The eight self-supervised baselines carry none of this exposure, training to a fixed epoch count. It can favour the four models we train and the fusion built on them, and we make no claim about the net direction of its effect. The RankFuse adapter is fitted on the target cells and reads their batch labels in both settings, as does every batch-aware method in the comparison.

Metrics. We report the scIB aggregates (Luecken et al., 2022), a biological conservation score, a batch correction score and their weighted total $\text{Total} = 0.6\,\text{Bio} + 0.4\,\text{Batch}$, with the component definitions and the version pin in Appendix S.1.2. Total is the primary endpoint. Alongside them we report the neighbourhood identity score of Diaz-Mejia et al. (2025), which was built so that easy targets cannot inflate accuracy, asking instead whether a cell's neighbours are predictable across batches. Appendix S.1.2 defines it and Appendix S.1.1 reports its two factors separately. We do not compare it across datasets, where the gate rate and the class base rate differ.

Table conventions. All values are raw and unscaled, computed with `scib-metrics` 0.5.1 on the pp4000 basis. All were scored under a patch that seeds the package's Leiden clustering except the PCA rows on pancreas and PBMC, which predate it. Appendix S.4.2 gives the details. The table published with scSSL-Bench is min–max scaled within each dataset across the methods it contains, so none of our values may be differenced against it. Bold marks the best value in each column at full precision, so two cells that print identically may differ beyond the displayed digits. Appendix S.4.2 records the rest.

Table 2: Fixed-checkpoint setting. The two scREF encoders and the two foundation models are applied as fixed checkpoints without target-dataset fine-tuning, and RankFuse fuses all four. Section 3 states what is and is not established about their pretraining overlap. Conventions are those of Section 3: raw, unscaled, pp4000. The RankFuse and control rows are means over three layer seeds; the four encoder rows are single checkpoints with no seed axis. † The matched control gives adapted scGPT, the strongest single-encoder arm on PBMC, its own graph and the identical adapter. On each dataset it was scored in a separate pass together with a rescoring of the fusion arm, so it is not bolded against the rows above; in those passes RankFuse reads 0.592, 0.661 and 0.613, and Appendix S.4.2 gives the same-pass margins.

	Pancreas				**PBMC**				**Immune cell atlas**			
Model	Total	Bio	Batch	Identity	Total	Bio	Batch	Identity	Total	Bio	Batch	Identity
Encoders, applied as fixed checkpoints												
scREF BA-scVI	0.495	0.557	0.404	0.486	0.625	0.668	0.561	**0.836**	0.597	0.629	0.548	**0.627**
scREF DINO	**0.595**	0.629	**0.544**	0.609	0.591	0.631	0.532	0.817	0.579	0.613	0.529	0.582
scGPT	0.510	0.597	0.380	0.403	0.537	0.643	0.378	0.430	0.566	0.625	0.478	0.433
Geneformer	0.402	0.473	0.295	0.290	0.411	0.474	0.318	0.293	0.505	0.541	0.451	0.317
Fusion over those encoders												
RankFuse, all four	0.592	**0.630**	0.535	**0.654**	**0.660**	**0.711**	**0.583**	0.800	**0.609**	**0.643**	**0.558**	0.581
Matched single-encoder control												
Adapted scGPT†	0.591	0.616	0.553	0.712	0.650	0.682	0.600	0.796	0.578	0.573	0.586	0.550

Seeds and repeatability. The two settings replicate on different axes. In the target-trained setting every dataset is run at three encoder seeds, 20 to 22, and every value we report for a model we train is the unweighted mean over the three; the deterministic and fixed-checkpoint rows have no seed axis. In the fixed-checkpoint setting the encoders are single checkpoints and the replicated axis is the adapter, at three layer seeds. As a guide to repeatability, two PBMC configurations that were trained twice under identical settings differed by 0.0028 and 0.0033 in Total. That observation is specific to those runs and is neither a significance threshold nor an uncertainty bound for other comparisons. Where a comparison had a decision threshold registered in advance, the threshold is stated with that comparison and applies to it alone.

4 Results

4.1 Integration across three datasets

Fixed checkpoints. Fusing the four fixed checkpoints of Section 3 raises the weighted total over the strongest encoder they contain on two of three datasets, from 0.625 to 0.660 on PBMC and from 0.597 to 0.609 on the atlas, while on pancreas the two differ by 0.003 (Table 2). Biological conservation and batch correction both improve on PBMC and on the atlas, with the biological term contributing the larger part of the weighted gain in each case. On the atlas the strongest single encoder still leads on identity while the fusion leads on Total. It does not reach a model trained on the target data, 0.660 against 0.690 on PBMC. What is reused across datasets here is the encoder checkpoints. The graph and the layer over it are built and fitted on the target dataset in this setting as in every other, and the layer reads its batch labels.

Encoders trained on the target data. Table 3 repeats the comparison with every method trained on the dataset it is scored on, which is scSSL-Bench's convention, and Figure 2 shows the PBMC embeddings. The outcome depends on the dataset. RankFuse takes the highest total on PBMC, 0.711 against 0.690 for the strongest single model, a margin that holds on two of three encoder seeds; it ranks fourth on the atlas, 0.029 behind DINO (batch-centred), with the strongest batch correction in that column; and on pancreas it sits 0.030 behind the token-encoder BA-scVI, a shortfall that is almost all biological conservation, 0.765 against 0.813. It is above every self-supervised baseline in our comparison on all three. Appendix S.4.3 gives the per-seed values and each baseline in turn.

Table 3: Target-trained setting. Every model above the final rule is trained on the dataset it is scored on; the two foundation models below it are the fixed checkpoints of Table 2, repeated for orientation. Values for a model we train are means over encoder seeds 20 to 22, in a single pass per dataset; PCA is deterministic and the two foundation models are single checkpoints, so none of the three has a seed axis. `tok-m-noaug` is a further gene-token model, with attention, that is not part of the thirteen-model fusion pool; Appendix S.4.2 gives its configuration and says which records are missing. It is the exception: its PBMC row comes from a later pass, and its atlas cells are seed 20 alone and are not bolded. The two BA-scVI variants differ in their encoder: BA-scVI uses a four-layer multilayer perceptron, and BA-scVI (gene-token) an attention-free gene-token encoder with one layer where that perceptron has four, before the stage both share. Appendix S.4.2 maps both to their checkpoint identifiers and records the seed variability of individual rows.

	Pancreas				**PBMC**				**Immune cell atlas**			
Model	Total	Bio	Batch	Identity	Total	Bio	Batch	Identity	Total	Bio	Batch	Identity
Baselines												
PCA	0.545	0.661	0.371	0.226	0.459	0.547	0.327	0.464	0.508	0.605	0.362	0.268
scVI	0.681	0.747	0.583	0.904	0.657	0.670	0.638	0.854	0.645	0.700	0.562	**0.713**
Self-supervised baselines												
SimCLR	0.561	0.581	0.530	0.516	0.522	0.502	0.553	0.518	0.596	0.618	0.564	0.291
MoCo	0.567	0.585	0.541	0.557	0.519	0.492	0.560	0.525	0.599	0.624	0.563	0.290
VICReg	0.588	0.614	0.550	0.539	0.539	0.513	0.577	0.532	0.622	0.659	0.566	0.329
Barlow Twins	0.545	0.556	0.529	0.482	0.502	0.485	0.528	0.493	0.575	0.600	0.537	0.210
BYOL	0.551	0.541	0.564	0.575	0.470	0.400	0.575	0.443	0.578	0.589	0.562	0.345
NNCLR	0.548	0.564	0.523	0.505	0.507	0.486	0.538	0.495	0.582	0.602	0.552	0.223
SimSiam	0.530	0.529	0.532	0.516	0.477	0.456	0.508	0.439	0.563	0.578	0.541	0.200
Concerto	0.472	0.409	0.567	0.322	0.426	0.329	0.572	0.296	0.442	0.400	0.504	0.214
Batch-aware models, trained on the dataset												
BA-scVI	0.734	0.758	**0.699**	0.935	0.667	0.652	0.689	0.860	0.622	0.662	0.561	0.627
BA-scVI + RecCon	0.705	0.724	0.676	0.894	0.647	0.647	0.647	0.835	0.624	0.659	0.571	0.591
DINO (batch-centred)	0.604	0.640	0.550	0.831	0.602	0.635	0.552	0.782	**0.665**	**0.728**	0.570	0.606
BA-scVI (gene-token)	**0.758**	**0.813**	0.676	**0.956**	0.690	**0.718**	0.649	**0.879**	0.643	0.678	0.589	0.703
tok-m-noaug	0.735	0.789	0.654	0.953	0.658	0.676	0.631	0.861	0.630	0.668	0.573	0.741
Fusion												
RankFuse	0.729	0.765	0.675	0.907	**0.711**	0.711	**0.710**	0.848	0.636	0.644	**0.624**	0.673
Foundation models, no target-dataset fine-tuning												
scGPT	0.510	0.597	0.380	0.403	0.537	0.643	0.378	0.430	0.566	0.625	0.478	0.433
Geneformer	0.402	0.473	0.295	0.290	0.411	0.474	0.318	0.293	0.505	0.541	0.451	0.317

4.2 Contributions of graph construction and adaptation

One encoder given the same treatment. Table 2 compares a pipeline against encoders as they arrive, so it measures fusion and adaptation together. To separate them we gave a single encoder its own cross-batch graph, built by the identical procedure, and the identical adapter, and scored it in the same pass as the fusion. On the atlas the fusion leads adapted scGPT by 0.035, on every layer seed, which clears the margin a prespecified rule required for support. On PBMC, where adapted scGPT is the strongest such arm, the fusion leads by 0.012, in the band the rule set aside as inconclusive, and on pancreas, the smallest dataset, the two score alike. On pancreas and the atlas a raw scREF checkpoint scores above adapted scGPT, so there the control compares fusion with adapted scGPT alone (Appendix S.5).

On PBMC, adaptation helps the four encoders very unevenly: scGPT rises from 0.537 to 0.650 given its own graph and the adapter, scREF BA-scVI moves only from 0.625 to 0.626, and the four arms span 0.480 to 0.650. Shuffling one encoder's graph while preserving every degree drops it from 0.626 to 0.469, so the adapter does not lift an arbitrary checkpoint over an arbitrary graph (Appendix S.5).

Which encoders build the graph. The two rosters need not hold the same encoders, so we crossed them against the two roles on PBMC over three encoder seeds in one pass. Swapping the graph roster moved Total by 0.108 and swapping the feature roster by 0.019, with the same sign on all

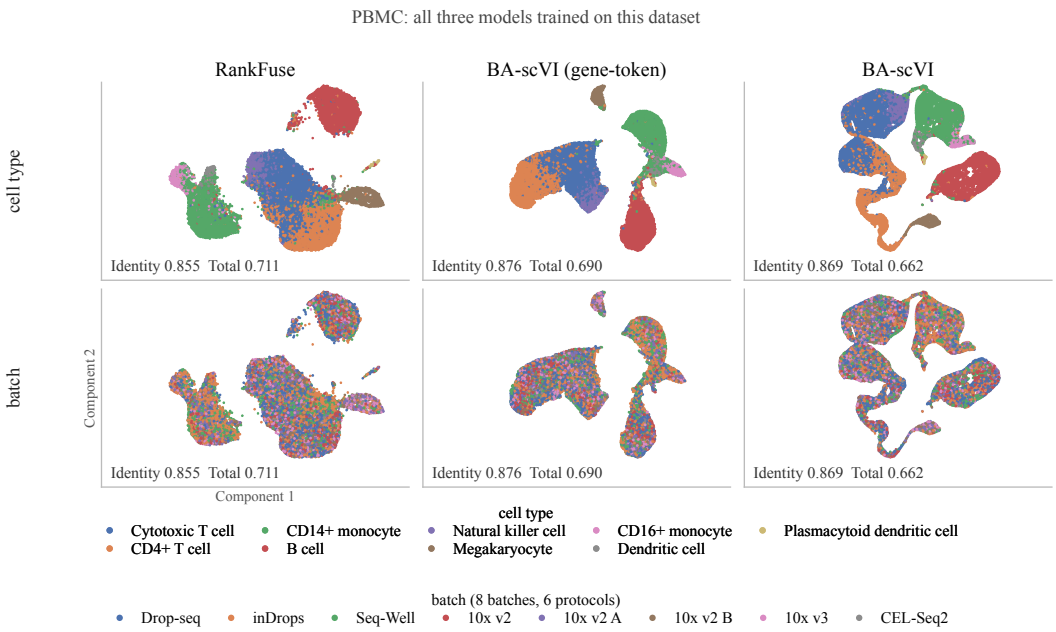

Figure 2: PBMC in the target-trained setting. RankFuse is shown against the two strongest baselines of Table 3, with the upper row coloured by cell type and the lower by batch, so a column shows the same embedding under the two things integration has to reconcile. Each panel carries its own identity score and Total at the lower left. These are seed-20 embeddings while the tables report three-seed means, so a panel figure and the corresponding table cell are different quantities. Figure S2 has the remaining models, including Concerto, and the other two datasets.

three seeds, and a seven-roster series varying the graph roster alone spans 0.168. The graph roster has the larger effect in these target-trained PBMC comparisons. Changing it changes both the graph and the training positives, which are sampled from the same fused rankings. A separate preregistered hypothesis, relating the quality gap between a roster's two best encoders to fusion performance, was not supported (Appendix S.6.1).

Removing the graph but keeping the encoders can be tested in both settings. With target-trained encoders, concatenating the four whitened feature-roster embeddings and scoring them directly gives 0.670 against 0.758 on pancreas and 0.640 against 0.690 on PBMC, below the best single member on all six encoder seeds, and the graph and layer recover 0.058 and 0.071 of that. Compressing the best member alone to ten dimensions moves it by at most 0.014, so compression does not by itself account for the drop, though this control does not isolate the effect of compressing the other blocks. With fixed checkpoints the corresponding control, a lossless rotation of the four whitened blocks with nothing trained, reaches 0.531 on PBMC against 0.625 for the strongest encoder in it. Concatenation is below the best member in both settings (Appendix S.6.1).

4.3 What the layer adds and what it costs

Table 4 varies the layer and the objective in turn on a single-seed basis, so both sweeps are exploratory. They show no clear advantage from greater depth or from attention over a single graph convolution, and VISReg has the highest observed Total.

What the layer contributes depends on the setting. Where the encoders are already trained on the target data, replacing fixed propagation with the trained layer changes Total little and in both directions across the six pancreas and PBMC seed pairs, and lowers it on all three atlas pairs, by 0.004 to 0.012. On PBMC with fixed checkpoints, the layer delivers the advantage: removing it takes the fusion from 0.660 to 0.623, below the 0.625 of the strongest encoder it fuses. It improves Total in that setting by 0.037 on PBMC and 0.038 on pancreas; on the atlas it changes Total by 0.001, smaller than the difference between two repeat trainings of one configuration, and the untrained fusion al-

Table 4: Layer and objective sweeps. Each varies one component with the other two held fixed, on PBMC at a single seed on an earlier design's roster. The two are separate runs on that basis, which is why the shared arms differ slightly, and the sweep does not establish a reliable ordering among the closely scoring configurations. These rows are not three-seed means and are not comparable cell for cell with Table 3. Bio and Batch are in Appendix S.7.

Layer	Total	Objective	Total
GCN depth 1	**.7164**	VISReg	**.7210**
GAT depth 2	.7136	VICReg	.7160
GCN depth 2	.7131	VISReg, shape wt. 2.0	.7154
GAT depth 1	.7120	InfoNCE	.7141
Fixed diffusion	.7116	Fixed diffusion	.7091
Union graph, unlearned	.7081	*Best single model*	*.6873*
EdgeConv depth 2	.7007		
EdgeConv depth 1	.6989		
Best single model	*.6874*		

ready stands above the contributing encoders, 0.608 against 0.597. In the target-trained setting the layer raises batch correction and lowers biological conservation on nine of nine seed pairs; with transferred encoders it raises both on PBMC and pancreas. Neighbourhood identity falls on every arm we have run, 0.761 to 0.673 on the atlas, and that is the column where Table 3 puts us below three models. The per-seed arms are in Appendix S.7.

5 Discussion

RankFuse provides a way to combine single-cell encoders through the relationships they propose between cells. Rather than requiring their embedding coordinates to agree, it merges neighbour rankings into one integration graph and adapts a compact representation over that graph. Our controls separate two sources of improvement, the relationships the encoders supply and adaptation to the target dataset, and their contributions differ across datasets and pools. That is why a gain over an unadapted encoder need not be a gain from fusion. The balance between batch correction and biological conservation depends on the setting. With target-trained encoders the adapter raises batch correction and lowers biological conservation in every measured pair; with transferred encoders it raises biological conservation as well, on PBMC and pancreas. Neighbourhood identity falls in both settings, so a reader who cares most about fine-grained type structure should weigh the identity column against the total.

Integration and generic representation learning ask for related but different invariances, which is one reason performance here is task-dependent. Single-cell measurements are sparse and shaped by sampling and technical variation, and batches can contain different cell populations in different proportions (Lopez et al., 2018; Luecken et al., 2022). Agreement between two augmented views of a cell does not by itself identify which part of that variation should be removed: an encoder can preserve what is needed to type a cell while keeping the structure of the study it came from, and stronger mixing can remove biological differences along with technical ones. Those considerations motivate building on cross-batch relationships and training against a batch adversary. They are plausible readings of why methods reorder across tasks, not causes our experiments isolate. Nor do they suggest that objectives from visual representation learning are unsuited to expression data: scSSL-Bench finds generic self-supervised methods competitive on several of its tasks (Ovcharenko et al., 2025), and the scREF DINO checkpoint is the strongest fixed encoder on pancreas in our own comparison.

Our graph is built by a fixed rule from ranked lists, with a single trained layer over it, so the contribution is the combination rather than a learned graph structure. The closest existing work fuses the graphs themselves: similarity network fusion (Wang et al., 2014) builds one network by iteratively diffusing several into agreement, and Conos (Barkas et al., 2019) joins datasets through a graph assembled from many pairwise alignments, while weighted-nearest-neighbour analysis (Hao et al., 2021) learns per-cell weights over modalities that already share a cell. RankFuse combines the encoders one step earlier, at the level of ranked neighbour lists: each contributes an ordering rather than a similarity on its own scale, and one graph is built from the merged ordering. This combines infor-

mation from encoders with incompatible coordinates before constructing the graph. Appendix S.10 places the method against those literatures and against unsupervised rank aggregation in full.

Several limits bound what these results establish. The pancreas deficit is real and we cannot account for it; Appendix S.9 reports where the neighbourhood classification errors fall across cell types, which describes the errors rather than explaining the deficit. Cell-type labels enter reference-encoder selection and early stopping of the four batch-aware models (Section 3), although the adapter's training objective does not use them. Equation S1 ranks candidate rosters on disagreement alone and never on quality, and our pool is not homogeneous in quality: identity across the thirteen target-trained encoders on PBMC spans 0.296 to 0.879, and still 0.439 to 0.879 once the collapsed member is excluded. A quality-blind criterion in a pool of that spread can select an encoder for disagreeing with everyone when the reason is that it carries little signal, which is what happens with Concerto. We therefore report the rule as a heuristic with a measured limitation, and we do not measure the candidate coverage it stands in for. Finally the crossing, the roster series and the two sweeps are on PBMC alone, the assignment they select is carried to the other datasets rather than retested there, and on pancreas, the smallest of the three datasets, fusion does not exceed the strongest single model in either setting. The open problem this leaves is to select complementary neighbours reliably while preserving the useful structure of individual encoders.

AI USE STATEMENT

This work was carried out with assistance from an AI coding and writing system under the direction of the authors, who are responsible for its contents. The division of labour was as follows. The project's intuition, its research questions and the design of its experiments are principally the authors' work, as are the paper's core ideas and its framework. The system was used principally for execution: it implemented the method and the evaluation code, prepared the datasets onto the scoring basis, ran and scored the experiments, and produced figures and tables. The authors drafted the paper's structure and argument, and the system helped fill it out and check it against the experimental record.

Of the categories the conference asks authors to disclose, the following also apply. The system contributed to methodology within the authors' design, including the form of the roster criterion and the choice of several controls, and it derived an analytic correction during an exploratory aggregation experiment. It refined hypotheses, several of which were pre-registered and later withdrawn. It generated synthetic data in two places: the degree-preserving edge shuffle used as a null control in Appendix S.5, and the planted-community graphs used by the determinism verifier in the code release. It surveyed the literature for the related-work discussion, edited for readability and formatted references. This paper states and proves no theorem.

Numbers in the paper are traced in our working notes to the file and field they were read from, and decision thresholds were recorded before the controls they govern were run. The authors have checked the reported results against those sources.

REPRODUCIBILITY STATEMENT

The method is specified in Section 2 and its constants in Appendix S.2.2; the datasets, the preprocessing basis, the metric definitions and the version pin are in Section 3 and Appendices S.1.2 and S.4; the training protocol for the models we train is in Appendix S.2.3. Our layer accumulates each cell's incoming edges with a scatter-add whose floating-point summation order the device does not fix, so two runs of one configuration at one seed on one device do not return the same embedding. The observed size of that effect, the clustering patch we apply to the metric suite, the agreement between independent scoring passes and a cross-device comparison are in Appendix S.3. The implementation, the determinism flag and the patched scorer accompany this submission as anonymised supplementary material.

Supplementary material. The appendix sections referenced above, S.1 to S.10, together with Tables S1 to S6 and Figures S1 and S2, are in the supplementary material, which carries its own copy of the reference list.

REFERENCES

Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations (ICLR)*, 2022.

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. doi: 10.1038/s41592-019-0466-z.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9650–9660, 2021.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1597–1607. PMLR, 2020.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15750–15758, 2021.

Gordon V. Cormack, Charles L. A. Clarke, and Stefan Buettcher. Reciprocal rank fusion outperforms Condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 758–759. ACM, 2009. doi: 10.1145/1571941.1572114.

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. doi: 10.1038/s41592-024-02201-0.

Juan Javier Diaz-Mejia, Elias Williams, Octavian Focsa, Dylan Mendonca, Swechha Singh, Brendan Innes, and Sam Cooper. Benchmarking and optimizing organism wide single-cell RNA alignment methods. In *Workshop on Learning Meaningful Representations of Life (LMRL), International Conference on Learning Representations (ICLR)*, 2025. arXiv:2503.20730.

Jiarui Ding, Xian Adiconis, Sean K. Simmons, Monika S. Kowalczyk, Cynthia C. Hession, Nemanja D. Marjanovic, Travis K. Hughes, Marc H. Wadsworth, Tyler Burks, Lan T. Nguyen, John Y. H. Kwon, Boaz Barak, William Ge, Amanda J. Kedaigle, Shaina Carroll, Shuqiang Li, Nir Hacohen, Orit Rozenblatt-Rosen, Alex K. Shalek, Alexandra-Chloé Villani, Aviv Regev, and Joshua Z. Levin. Systematic comparison of single-cell and single-nucleus RNA-sequencing methods. *Nature Biotechnology*, 38(6):737–746, 2020. doi: 10.1038/s41587-020-0465-8.

C. Domínguez Conde, C. Xu, L. B. Jarvis, D. B. Rainbow, S. B. Wells, T. Gomes, S. K. Howlett, O. Suchanek, K. Polanski, H. W. King, L. Mamanova, N. Huang, P. A. Szabo, L. Richardson, L. Bolt, E. S. Fasouli, K. T. Mahbubani, M. Prete, L. Tuck, N. Richoz, Z. K. Tuong, L. Campos, H. S. Mousa, E. J. Needham, S. Pritchard, T. Li, R. Elmentaite, J. Park, E. Rahmani, D. Chen, D. K. Menon, O. A. Bayraktar, L. K. James, K. B. Meyer, N. Yosef, M. R. Clatworthy, P. A. Sims, D. L. Farber, K. Saeb-Parsy, J. L. Jones, and S. A. Teichmann. Cross-tissue immune cell analysis reveals tissue-specific features in humans. *Science*, 376(6594):eabl5197, 2022. doi: 10.1126/science.abl5197.

Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9588–9597, 2021.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, 2020.

Yuhan Hao, Stephanie Hao, Erica Andersen-Nissen, William M. Mauck, Shiwei Zheng, Andrew Butler, Maddie J. Lee, Aaron J. Wilk, Charlotte Darby, Michael Zager, Paul Hoffman, Marlon Stoeckius, Efthymia Papalexi, Eleni P. Mimitou, Jaison Jain, Avi Srivastava, Tim Stuart, Lamar M. Fleming, Bertrand Yeung, Angela J. Rogers, Juliana M. McElrath, Catherine A. Blish, Raphael Gottardo, Peter Smibert, and Rahul Satija. Integrated analysis of multimodal single-cell data. *Cell*, 184(13):3573–3587.e29, 2021. doi: 10.1016/j.cell.2021.04.048.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017.

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. doi: 10.1038/s41592-019-0619-0.

Chunan Liu, Aurelien Pelissier, Yanjun Shao, Lilian Denzler, Andrew C. R. Martin, Brooks Paige, and María Rodríguez Martínez. AbRank: A benchmark dataset and metric-learning framework for antibody-antigen affinity ranking. *arXiv preprint arXiv:2506.17857*, 2025.

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. doi: 10.1038/s41592-018-0229-2.

Malte D. Luecken, M. Büttner, K. Chaichoompu, A. Danese, M. Interlandi, M. F. Mueller, D. C. Strobl, L. Zappia, M. Dugas, M. Colomé-Tatché, and Fabian J. Theis. Benchmarking atlas-level data integration in single-cell genomics. *Nature Methods*, 19(1):41–50, 2022. doi: 10.1038/s41592-021-01336-8.

Olga Ovcharenko, Florian Barkmann, Philip Toma, Imant Daunhawer, Julia E. Vogt, Sebastian Schelter, and Valentina Boeva. scSSL-Bench: Benchmarking self-supervised learning for single-cell data. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 47416–47442. PMLR, 2025.

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, and Patrick T. Ellinor. Transfer learning enables predictions in network biology. *Nature*, 618:616–624, 2023. doi: 10.1038/s41586-023-06139-9.

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. doi: 10.1186/s13059-019-1850-9.

Bo Wang, Aziz M. Mezlini, Feyyaz Demir, Marc Fiume, Zhuowen Tu, Michael Brudno, Benjamin Haibe-Kains, and Anna Goldenberg. Similarity network fusion for aggregating data types on a genomic scale. *Nature Methods*, 11(3):333–337, 2014. doi: 10.1038/nmeth.2810.

Haiyu Wu, Randall Balestriero, and Morgan Levine. VISReg: Variance-invariance-sketching regularization for JEPA training. *arXiv preprint arXiv:2606.02572*, 2026.

Meng Yang, Yueyuxiao Yang, Chenxi Xie, Ming Ni, Jian Liu, Huanming Yang, Feng Mu, and Jian Wang. Contrastive learning enables rapid mapping to multimodal single-cell atlas of multimillion scale. *Nature Machine Intelligence*, 4(8):696–709, 2022. doi: 10.1038/s42256-022-00518-z.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow Twins: Self-supervised learning via redundancy reduction. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12310–12320. PMLR, 2021.

Junyi Zhang, Charles Herrmann, Junhwa Hur, Luisa Polania Cabrera, Varun Jampani, Deqing Sun, and Ming-Hsuan Yang. A tale of two features: Stable Diffusion complements DINO for zero-shot semantic correspondence. In *Advances in Neural Information Processing Systems 36 (NeurIPS)*, 2023.