

Supplementary Notes for “Comprehensive benchmarking of RNA velocity methods across single-cell datasets”

Yida Wu^{1†}, Chuihan Kong^{1†}, Xu Liao¹, Zhixiang Lin², Xiaobo Sun^{3*},

and Jin Liu^{1*}

¹School of Data Science, The Chinese University of Hong Kong, Shenzhen, China.

²Department of Statistics, The Chinese University of Hong Kong, Shatin, Hong Kong SAR, China.

³Department of Human Genetics, School of Medicine, Emory University, Atlanta, GA, USA.

[†]These authors contributed equally to this work.

*Corresponding author. Email: xiaobo.sun@emory.edu,
liujinlab@cuhk.edu.cn

Contents

1	Note S1: Rationale for computing CBBDir and ICVCoh using UMAP-projected velocity vectors	2
2	Note S2: Enrichment analysis of method rankings	3
3	Note S3: Evaluation of batch-insensitive embeddings from RNA velocity methods	3
4	Note S4: Implementation of cell-specific latent time for RNA velocity methods	3
5	Note S5: Justification for additional tasks excluded from the overall ranking	4
6	Note S6: Additional benchmarking metrics	5
7	Note S7: Computational environments	5

8	Note S8: Hyperparameter sensitivity analysis for guideline-recommended methods	5
9	Note S9: Shared and disparate findings compared with existing benchmark studies	7
10	Note S10: Benchmarking datasets	8
11	Note S11: Theoretical proof of the effective entropy score	13

1 Note S1: Rationale for computing CBDiR and ICVCoh using UMAP-projected velocity vectors

We computed cross-boundary directional correctness (CBDiR) and in-cluster coherence (ICVCoh) using uniform manifold approximation and projection (UMAP)-projected velocity vectors for two reasons. First, UMAP is widely used for cell annotation and biological interpretation in single-cell analysis [17]. Computing CBDiR and ICVCoh in the same space as cell-type annotations allows velocity directions to be evaluated consistently with the biological structures being interpreted. This choice is also consistent with the original definitions of CBDiR and ICVCoh, which were introduced and computed in UMAP space [35]. Second, although UMAP is nonlinear and may distort global geometry, it is constructed from a k -nearest-neighbor (k NN) graph and is designed to preserve local manifold structure [18]. Because both CBDiR and ICVCoh rely primarily on local neighborhood relationships, the local structures relevant to these metrics are expected to be largely retained. In our pipeline, the k NN graph used for UMAP construction is derived from principal component analysis (PCA) space, further anchoring the embedding in the original high-dimensional structure.

We further evaluated the sensitivity of CBDiR and ICVCoh to the choice of evaluation space using Data 3. For the 20 RNA velocity methods that output gene-space velocities, we compared metric values computed in gene-expression space with those computed in UMAP space (Additional file 2: Fig. S2, top panel). The two spaces showed positive correlations for both CBDiR ($\rho = 0.860$) and ICVCoh ($\rho = 0.719$). The top-performing methods were also largely preserved, with 8 of the top 10 methods overlapping for CBDiR and 9 of the top 10 overlapping for ICVCoh. We also compared PCA-space and UMAP-space evaluations across all 25 RNA velocity methods (Additional file 2: Fig. S2, bottom panel). The correlations were even stronger for both CBDiR ($\rho = 0.992$) and ICVCoh ($\rho = 0.955$), with complete overlap among the top 10 methods for both metrics. These results suggest that although projection can affect absolute velocity directions, the relative ranking of methods is largely robust to the choice of evaluation space.

2 Note S2: Enrichment analysis of method rankings

To evaluate whether specific groups are significantly enriched in the top or bottom n rankings, we performed an enrichment analysis using a one-sided hypergeometric test. The P -value, defined as the probability of observing at least k methods from a specific ranking category within the selected rank cutoff, is calculated as:

$$P(X \geq k) = \sum_{i=k}^{\min(n,K)} \frac{\binom{K}{i} \binom{N-K}{n-i}}{\binom{N}{n}} \quad (1)$$

where N is the total number of evaluated methods, K is the total number of methods in the specific category, n is the cutoff size (e.g., top 10), and k is the observed number of methods from that group within the cutoff. Statistical significance is indicated by asterisks (* : $P < 0.05$, ** : $P < 0.01$, *** : $P < 0.001$).

3 Note S3: Evaluation of batch-insensitive embeddings from RNA velocity methods

Among the 25 RNA-only velocity methods evaluated in this study, LatentVelo (std) and all the scTour variants were the only methods whose original studies reported batch-effect mitigation during dynamic inference. LatentVelo (std) incorporates batch IDs into the encoder and decoder while maintaining batch-independent latent dynamics [33], whereas scTour learns developmental time, vector fields, and latent representations jointly through continuous latent-state transformations [30]. To assess whether these methods produce batch-insensitive embeddings, we used `scib-metrics` (v0.5.6) [43] to evaluate biological conservation and batch-correction performance of their low-dimensional embeddings. PCA was used as the uncorrected baseline, and Harmony [44] was used as the batch-corrected baseline. On Data 5, LatentVelo (std) and all scTour variants outperformed PCA, with the scTour variants achieving performance comparable to Harmony (Additional file 2: Fig. S4b, top panel). On Data 6, only LatentVelo (std) and scTour (ZINB) outperformed PCA, and LatentVelo (std) achieved performance comparable to Harmony (Additional file 2: Fig. S4b, bottom panel). These results indicate that LatentVelo (std) and scTour (MSE/NB/ZINB) can mitigate batch effects to some extent during velocity inference, although their performance varies across datasets.

4 Note S4: Implementation of cell-specific latent time for RNA velocity methods

In this section, we describe the implementation details used to obtain cell-specific latent time (CLT) for each RNA velocity method according to its temporal modeling strategy. As introduced in the manuscript, we categorized the 25 RNA velocity methods into four groups based on how they model temporal information. Specifically, Group 1 methods explicitly estimate a unified CLT, whereas Group 2 methods estimate gene-specific latent times. Group 3 methods either infer CLT indirectly through a graph-based approach or do not provide temporal outputs at

all. Lastly, the Group 4 method (SDEvelo) estimates CLT using an SDE model combined with optimal transport alignment.

We obtained CLT estimates for each group as follows:

- **Groups 1 and 4:** We directly used the CLT estimates generated simultaneously with the velocity estimates.
- **Group 2:** We followed the original implementations and applied `scvelo.tl.latent_time()` to obtain the CLT. This function heuristically aggregates gene-specific latent times by assuming that early cells tend to be early across most genes, while terminal cells tend to be late.
- **Group 3:** For methods that provide their own CLT estimation procedures, we followed their original implementations (detailed below). For the remaining Group 3 methods, which do not produce native temporal outputs, we ensured a fair comparison by applying `scvelo.tl.velocity_pseudotime()`, following the approach used in scVelo (stc). This function estimates CLT based on a graph constructed from the velocity outputs of the corresponding methods.

Method-specific implementations for Group 3:

- **cellDancer:** cellDancer uses `celldancer.pseudo_time()` following kinetic parameter estimation. This function estimates CLT by simulating RNA velocity streamlines in low-dimensional embeddings and ordering cells along dominant trajectories via graph-based time alignment.
- **Dynamo (m1):** Dynamo (m1) applies `dynamo.ext.ddhodge()` following velocity inference. This function infers a single-cell potential, which serves as a proxy for CLT, by building a velocity-informed diffusion graph and computing the scalar potential from the divergence of the graph-based vector field via Hodge decomposition.

5 Note S5: Justification for additional tasks excluded from the overall ranking

To provide a more systematic assessment, our benchmarking framework includes four additional tasks: quantification stability, simulation experiments, multimodal integration, and computational scalability. However, these were excluded from the overall rankings to ensure a fair and robust comparison focused on core performance. The specific rationale for excluding each of these tasks is detailed below.

Quantification stability was evaluated separately, as quantification is an independent upstream process whose variability could confound the direct assessment of the velocity estimation methods themselves. Therefore, this metric mainly provides practical guidance for real-world usage rather than serving as a measure of a model’s intrinsic modeling capacity or biological interpretability. Simulation experiments were omitted due to the significant discrepancy between synthetic and real-world datasets. Specifically, simulations tended to bias

results against deep learning (DL)-based methods while favoring those built on generative assumptions similar to the simulation. Multimodal integration was excluded because it is a specialized feature limited to a small subset of tools, making it unsuitable as a universal metric. Finally, as computational scalability reflects technical capacity rather than biological accuracy, we provide separate scalability tiers for cells and genes, respectively, to guide researchers without conflating performance with resource requirements.

6 Note S6: Additional benchmarking metrics

Distance correlation

For each simulated dataset, ground-truth RNA velocity was available in gene space. However, for methods that output low-dimensional embedded velocities (e.g., VeloAE, scTour, and LatentVelo (std)), direct comparison with the high-dimensional ground truth is not feasible. Distance correlation (dCor) [51, 52] was therefore used to quantify nonlinear associations between inferred and true velocity structures based on pairwise cell-cell similarities. Formally,

$$\text{dCor}(\mathbf{V}^{\text{true}}, \mathbf{V}^{\text{inf}}) = \frac{\sum_{i,j=1}^N A_{ij} B_{ij}}{\sqrt{\left(\sum_{i,j=1}^N A_{ij}^2\right) \left(\sum_{i,j=1}^N B_{ij}^2\right)}}, \quad (2)$$

where A and B are double-centered pairwise Euclidean distance matrices of $\mathbf{V}^{\text{true}}$ and $\mathbf{V}^{\text{inf}}$, respectively:

$$A_{ij} = a_{ij} - \bar{a}_{i\cdot} - \bar{a}_{\cdot j} + \bar{a}_{\cdot\cdot}, \quad B_{ij} = b_{ij} - \bar{b}_{i\cdot} - \bar{b}_{\cdot j} + \bar{b}_{\cdot\cdot}, \quad (3)$$

with $a_{ij} = \|\mathbf{v}_i^{\text{true}} - \mathbf{v}_j^{\text{true}}\|_2$ and $b_{ij} = \|\mathbf{v}_i^{\text{inf}} - \mathbf{v}_j^{\text{inf}}\|_2$. Higher dCor values indicate stronger nonlinear concordance between inferred and true velocity structures.

Pearson’s correlation

For each simulated dataset, Pearson’s correlation between true and inferred CLT values was used to assess temporal accuracy. Higher correlation values indicate more accurate CLT inference.

7 Note S7: Computational environments

All analyses were performed on a Linux server running Ubuntu 22.04 LTS (Kernel 5.15), equipped with Intel Xeon Platinum 8352V CPUs (128 threads), 1 TB of RAM, and a single NVIDIA vGPU (32 GB of memory).

8 Note S8: Hyperparameter sensitivity analysis for guideline-recommended methods

We performed hyperparameter sensitivity analysis for the guideline-recommended methods, including UniTVelo (uni), Pyro-Velocity (m2), veloVI, scTour (ZINB), SDEvelo, DeepVelo,

Velocityto, Dynamo (m1), and SvelvetVAE. Data 3 was used for both Task I and Task II. For each method, we selected key hyperparameters highlighted in the original papers or tutorial documentation (Additional file 2: Fig. S20). Specifically, we considered the following parameters:

- **Pyro-Velocity (m2):** the number of training epochs, `max_epochs`. The default value is 4000, and we tested `max_epochs` values of 1000, 2000, 3000, and 4000.
- **veloVI:** the penalty scale, `penalty_scale`, which controls the relative contribution of the switch-time regularization term to the overall objective, together with the evidence lower bound (ELBO). The default value is 0.2; here, we evaluated `penalty_scale` values of 0.05, 0.1, 0.2, 0.4, and 0.8.
- **scTour (ZINB):** the reconstruction loss weight `alpha_recon_lec`, which balances the reconstruction loss from the encoder-derived latent representation z and that from the ODE-derived latent representation z_t . Specifically, we varied `alpha_recon_lec` and set `alpha_recon_lode` to $1 - \text{alpha_recon_lec}$. The default value is 0.5, and we tested `alpha_recon_lec` values of 0.3, 0.4, 0.5, 0.6, and 0.7.
- **UniTVelo (uni):** the threshold `AGENES_R2`, which determines whether additional genes are included during the later stage of unified-time fitting. The default value is 1.0, corresponding to no additional gene inclusion, and we tested `AGENES_R2` values of 0.2, 0.4, 0.6, 0.8, and 1.0.
- **SDEvelo:** the number of cells used for model fitting, `n_cell`. The default value is 5000, and we tested `n_cell` values of 500, 1000, 2000, 5000, and 10000.
- **DeepVelo:** the neighbor parameter `topC`, which specifies the number of candidate cells used in the continuity loss, including the cell itself. Since the default setting corresponds to $k = 30$ future neighbors after excluding the cell itself, we tested k values of 10, 20, 30, 50, and 100, implemented by setting `topC` to $k + 1$.
- **Velocityto:** the quantile range `perc`, which determines the lower and upper quantiles used to fit the steady-state regression. The default setting is [5, 95], and we evaluated `perc` values of [1, 99], [2, 98], [5, 95], and [10, 90].
- **Dynamo (m1):** the model and parameter-estimation setting, which determines whether kinetic parameters are estimated under a stochastic moment-based model or a deterministic regression-based model. We evaluated four configurations: stochastic models using the generalized method of moments (`gmm`) or a negative binomial likelihood (`negbin`), and deterministic models using ordinary least squares regression (`ols`) or the RANSAC (RANdom Sample Consensus) algorithm (`ransac`).
- **SvelvetVAE:** the loss weight `velocity_lambda`, which directly controls the contribution of the unspliced/velocity matching loss in the training objective. The default value is 10.0, and we tested `velocity_lambda` values of 1, 3, 10, and 30.

Overall, most methods showed relatively stable performance across the tested parameter settings, but the default settings were not always optimal. These results suggest that users may tune key method-specific hyperparameters according to dataset characteristics and analysis goals.

9 Note S9: Shared and disparate findings compared with existing benchmark studies

Several conclusions from our benchmark are consistent with previous RNA velocity benchmarking studies. Despite differences in evaluation task design, our results are consistent with previous benchmarking studies by Ancheta et al. [24], Luo et al. [25], and Huang et al. [26] in showing that RNA velocity method performance is highly context-dependent, and that no single method is uniformly optimal across evaluation settings. We also observed that method performance fluctuates substantially across datasets in directional consistency, consistent with Luo et al. and Huang et al.

Beyond these shared findings, our benchmark differs from prior studies in both scope and evaluation design. Existing benchmarks mainly focus on general performance dimensions, such as accuracy, stability, and usability, and often assess accuracy using directional correctness and local coherence. In contrast, we adopted a task-aware framework covering directional consistency, temporal precision, negative-control robustness, sequencing-depth stability, quantification stability, and simulation-based evaluation. In particular, we included mature steady-state systems as negative controls to quantify false-positive dynamic signals, a vulnerability highlighted by Bergen et al. [21] and Gorin et al. [22] but largely not evaluated in previous benchmarks. We also treated temporal precision as a core task by using metabolic-labeling, time-course, and lineage-tracing datasets with experimentally measured ground truth to evaluate inferred cell-specific latent time. Another distinction in evaluation design is that Luo et al. and Huang et al. did not separately evaluate different variants of the same method, such as UniTVelo [27] and Pyro-Velocity [28]. Since these variants rely on different model assumptions and can perform differently across tasks, representing each method by a single default variant may obscure practical differences in method selection.

Finally, we acknowledge that several prior benchmarks cover aspects that are not included in our study, but these differences mainly reflect different evaluation scopes and design goals. Huang et al. extended the benchmark framework to spatial velocity methods, whereas our benchmark focuses on RNA velocity inference in single-cell datasets. Ancheta et al. further evaluated downstream driver-gene identification, whereas our benchmark focuses on the direct outputs of RNA velocity methods, namely velocity vectors and latent time. By focusing on these direct outputs across biologically motivated tasks, our framework provides a detailed characterization of method behavior and highlights methodological gaps, including the need for improved gene-dependence modeling, more accurate temporal inference, and better-designed multimodal architectures. We also acknowledge that our benchmark does not evaluate gene-selection sensitivity, an aspect assessed by Luo et al., and we have noted this limitation in the Discussion section.

10 Note S10: Benchmarking datasets

Data 1

This dataset captures transcriptional dynamics in human bone marrow using single-cell RNA sequencing [7]. It profiles key bone marrow cell populations and is designed to capture major transitions in hematopoietic differentiation, including the progressive loss of cellular plasticity. For the CBBir evaluation, directional correctness was assessed across established differentiation trajectories: (Hematopoietic stem cells 1 $\rightarrow$ Hematopoietic stem cells 2), (Hematopoietic stem cells 1 $\rightarrow$ Erythroids 1), and (Erythroids 1 $\rightarrow$ Erythroids 2). Three-fold cross-validation was performed on this dataset.

Data 2

This dataset captures transcriptional dynamics in the developing dentate gyrus using droplet-based single-cell RNA sequencing (10x Genomics Chromium) [65]. Cells were profiled at two developmental time points, P12 and P35, capturing maturation of the granule cell lineage as well as other differentiated cell populations within the developing tissue. For the CBBir evaluation, directional correctness was assessed across the following differentiation trajectories: (nIPC $\rightarrow$ Neuroblast), (Neuroblast $\rightarrow$ Granule immature), (Granule immature $\rightarrow$ Granule mature), (Radial Glia-like $\rightarrow$ Astrocytes), and (OPC $\rightarrow$ OL). Two-fold cross-validation was performed on this dataset.

Data 3

This dataset captures transcriptional dynamics during mouse gastrulation using single-cell RNA sequencing and was specifically subset to the erythroid lineage [66]. It contains cells collected at nine developmental time points between embryonic day 6.5 and day 8.5, providing a focused view of erythroid differentiation within the broader gastrulation program. For the CBBir evaluation, directional correctness was assessed across known differentiation trajectories: (Blood progenitors 1 $\rightarrow$ Blood progenitors 2), (Blood progenitors 2 $\rightarrow$ Erythroid 1), (Erythroid 1 $\rightarrow$ Erythroid 2), and (Erythroid 2 $\rightarrow$ Erythroid 3). Three-fold cross-validation was performed on this dataset.

Data 4

This dataset captures transcriptional dynamics in the developing pancreas using single-cell RNA sequencing, focusing on epithelial cells and Ngn3-Venus fusion cells at embryonic day 15.5 [67]. It provides a snapshot of endocrine differentiation as progenitor cells within the pancreatic epithelium progress toward four major endocrine fates: alpha, beta, delta, and epsilon cells. For the CBBir evaluation, directional correctness was assessed across established differentiation trajectories: (Ngn3 high EP $\rightarrow$ Pre-endocrine), (Pre-endocrine $\rightarrow$ Alpha), (Pre-endocrine $\rightarrow$ Beta), (Pre-endocrine $\rightarrow$ Delta), and (Pre-endocrine $\rightarrow$ Epsilon). Three-fold cross-validation was performed on this dataset.

Data 5

This dataset contains embryonic pancreatic epithelial cells profiled by four 10x Genomics single-cell RNA-seq experiments from a Neurogenin3 (Ngn3)-Venus fusion (NVF) reporter mouse line, covering embryos from E12.5 to E15.5 [67]. Our analysis focused on four cell types: Ngn3 low EP, Ngn3 high EP, Fev+, and Endocrine cells. For the CBBDir evaluation, directional correctness was assessed across established differentiation trajectories: (Ngn3 low EP $\rightarrow$ Ngn3 high EP), (Ngn3 high EP $\rightarrow$ Fev+), and (Fev+ $\rightarrow$ Endocrine). Three-fold cross-validation was performed on this dataset.

Data 6

This dataset contains two 10x Multiome datasets generated from human hematopoietic stem and progenitor cells (HSPCs) [14, 69]. Both samples were isolated from human bone marrow and cultured for seven days, but they came from different donors, and their library preparation and sequencing were performed months apart. For the CBBDir evaluation, directional correctness was assessed along established differentiation trajectories: (LMPP $\rightarrow$ GMP), (HSC $\rightarrow$ MEP), (MEP $\rightarrow$ Erythrocyte), (MEP $\rightarrow$ Prog MK), and (Prog MK $\rightarrow$ Platelet). Three-fold cross-validation was performed on this dataset.

Data 7

This dataset captures gene-regulatory dynamics in the developing human cerebral cortex using single-cell multiomic profiling [72]. It integrates single-cell RNA-seq and ATAC-seq data from primary cortical tissue collected at post-conceptual weeks 16, 20, 21, and 24, profiling major neuronal and glial populations across key stages of corticogenesis. The data span continuous differentiation trajectories from radial glia and intermediate progenitors to excitatory neurons and interneurons, with well-characterized developmental markers. For the CBBDir evaluation, directional correctness was assessed across established differentiation trajectories: (Cyc. Prog. $\rightarrow$ Early RG), (Early RG $\rightarrow$ Late RG), (Cyc. Prog. $\rightarrow$ nIPC), and (nIPC $\rightarrow$ GluN). Five-fold cross-validation was performed on this dataset.

Data 8

This dataset contains single-nucleus RNA-seq profiles of fetal brain samples collected during mid-gestation, from post-conception weeks 12 to 19 [40]. For the CBBDir evaluation, directional correctness was assessed along established differentiation trajectories: (Radial glia $\rightarrow$ Intermediate progenitor), (Intermediate progenitor $\rightarrow$ Newborn excitatory neuron), (Newborn excitatory neuron $\rightarrow$ Immature excitatory neuron), and (Immature excitatory neuron $\rightarrow$ Mature excitatory neuron). Three-fold cross-validation was performed on this dataset.

Data 9

This dataset captures transcriptional dynamics in mouse intestinal organoids using single-cell 5-ethynyl uridine (EU)-labeled RNA sequencing (scEU-seq), which simultaneously quantifies newly transcribed and pre-existing mRNAs [74]. For the CBBDir evaluation, directional correctness

was assessed across known differentiation trajectories: (Stem cells $\rightarrow$ TA cells), (TA cells $\rightarrow$ Enterocytes), (Stem cells $\rightarrow$ Enteroendocrine progenitor), and (Enteroendocrine progenitor $\rightarrow$ Enteroendocrine). Three-fold cross-validation was performed on this dataset.

Data 10

This dataset profiles transcriptional dynamics in primary mouse cortical neurons using single-cell metabolically labeled new RNA tagging sequencing (scNT-seq) [75]. Neurons were stimulated with potassium chloride (KCl) and sampled at 0, 15, 30, 60, and 120 minutes. For the CBBir evaluation, directional correctness was assessed across sequential time intervals: (0 $\rightarrow$ 15), (15 $\rightarrow$ 30), (30 $\rightarrow$ 60), and (60 $\rightarrow$ 120). Three-fold cross-validation was performed on this dataset.

Data 11

This dataset characterizes gene expression in fluorescence ubiquitination cell cycle indicator (FUCCI) U2OS cells [76]. To establish discrete ground-truth temporal labels, the continuous FUCCI pseudotime was discretized into five bins following the published tutorial: https://yoseflab.github.io/velovi_reproducibility/estimation_comparison/fucci.html. Three-fold cross-validation was performed on this dataset.

Data 12

This dataset comprises single-cell transcriptomic profiles from the primary tumor sample 3726_NT_T1 of a genetically engineered mouse model (GEMM) of lung adenocarcinoma (LUAD) [78]. The single-cell fitness signature was obtained from the original study, where the expression of a specific gene module was associated with cellular proliferative fitness estimated from the phylogenetic tree. To establish discrete ground-truth temporal labels, we discretized this continuous fitness signature into four quantile-based bins, with higher fitness scores indicating later temporal states. Three-fold cross-validation was performed on this dataset.

Data 13

This dataset profiles the developing murine hindlimb across four embryonic time points: E11.5, E13.5, E15.5, and E18.5 [81]. Data processing was performed using **cellranger** (v8.0.0) for read alignment and **velocity** (v0.17.17) for quantification of spliced and unspliced mRNA abundances. Three-fold cross-validation was performed on this dataset.

Data 14

This dataset captures transcriptomic diversification and maturation in the mouse primary visual cortex (V1) during postnatal development using droplet-based single-nucleus RNA sequencing. Nuclear transcriptomes were profiled at six postnatal time points (P8, P14, P17, P21, P28, and P38), spanning stages before, during, and after the critical period of ocular dominance plasticity [83]. We focused on the Gluta_NR subset, corresponding to glutamatergic neurons in the non-responsive lineage. Five-fold cross-validation was performed on this dataset.

Data 15

This dataset captures the reprogramming of mouse embryonic fibroblasts (MEFs) into induced endoderm progenitors (iEPs), with samples collected at eight time points [85]. Five-fold cross-validation was performed on this dataset.

Data 16

This dataset profiles the reprogramming of MEFs into induced pluripotent stem cells (iPSCs) under serum and 2i conditions, with samples collected at 39 time points over 18 days [86]. For the present analysis, only the serum condition was considered. Five-fold cross-validation was performed on this dataset.

Data 17

This dataset consists of PBMCs from a 10x Genomics experiment containing 65,877 cells. It profiles major immune cell types derived from hematopoietic stem cells in the bone marrow and includes representatives of both myeloid and lymphoid lineages [88]. Five-fold cross-validation was performed on this dataset.

Data 18 & 20

These two datasets consist of PBMCs from a healthy female donor aged 25, obtained by 10x Genomics from AllCells. Data 18 [89] consists of 10,194 cells, and Data 20 [91] consists of 10,497 cells. Three-fold cross-validation was performed on both datasets.

Data 19 & 21

These two datasets consist of PBMCs from two different healthy donors. Data 19 [90] contains 8,381 cells, and Data 21 [92] contains 11,769 cells. Three-fold cross-validation was performed on both datasets.

Data 22

This dataset profiles human hematopoietic stem and progenitor cells (HSPCs) and their early erythroid and myeloid progeny [93]. The original study used Cap Analysis of Gene Expression, ChIP-seq for histone modifications, and Moloney leukemia virus integration site mapping to characterize transcriptional activity and regulatory element usage during early hematopoietic differentiation. These data capture promoter, enhancer, and super-enhancer landscapes across multipotent stem cells and lineage-committed progenitors, providing a detailed view of transcriptional states along the myeloid and erythroid lineages. For the CBDir evaluation, directional correctness was assessed across established differentiation trajectories: (HSC $\rightarrow$ MPP), (MPP $\rightarrow$ LMPP), (LMPP $\rightarrow$ GMP), (HSC $\rightarrow$ MEP), (MEP $\rightarrow$ Erythrocyte), (MEP $\rightarrow$ Prog MK), and (Prog MK $\rightarrow$ Platelet). Three-fold cross-validation was performed on this dataset.

Data 23

This dataset contains single-cell ATAC-seq and single-cell RNA-seq profiles from the developing human cerebral cortex, generated using the Chromium platform (10x Genomics) across four fetal developmental stages: post-conceptual weeks (PCWs) 16, 20, 21, and 24 [72]. Our analysis used the two PCW21 copies, which capture cellular heterogeneity during mid-gestational cortical development. For the CDBir evaluation, directional correctness was assessed across known cortical differentiation trajectories: (Cyc. $\rightarrow$ RG/Astro), (Cyc. $\rightarrow$ nIPC/ExN), (nIPC/ExN $\rightarrow$ ExUp), and (ExUp $\rightarrow$ ExDp). Three-fold cross-validation was performed on this dataset.

Data 24

This dataset contains joint single-nucleus ATAC-seq and gene expression profiles from embryonic mouse brain at E18. It provides paired chromatin accessibility and transcriptional measurements that capture cellular heterogeneity across major embryonic brain regions [97]. For the CDBir evaluation, directional correctness was assessed across known differentiation trajectories: (RG, Astro, OPC $\rightarrow$ IPC), (IPC $\rightarrow$ V-SVZ), (V-SVZ $\rightarrow$ Upper Layer), and (Upper Layer $\rightarrow$ Deeper Layer). Three-fold cross-validation was performed on this dataset.

Data 25

This dataset was generated from human RPE1-FUCCI cells using scEU-seq, with the FUCCI reporter system providing cell-cycle position information through Geminin-GFP and Cdt1-RFP signals and enabling cells to be ordered along cell-cycle progression [74]. For this study, data from the pulse experiment were selected and processed following the guidelines provided at: https://yoseflab.github.io/velovi_reproducibility/estimation_comparison/sceu.html. To establish discrete ground-truth temporal labels, continuous cell cycle scores were discretized into five bins. Three-fold cross-validation was performed on this dataset.

Data 26

This dataset profiles the dynamics of human hematopoiesis using scNT-seq to capture metabolic labeling information [15]. For the CDBir evaluation, directional correctness was assessed across established differentiation trajectories: (HSC $\rightarrow$ MEP-like), (HSC $\rightarrow$ GMP-like), (MEP-like $\rightarrow$ Meg), (MEP-like $\rightarrow$ Ery), (MEP-like $\rightarrow$ Bas), (GMP-like $\rightarrow$ Bas), (GMP-like $\rightarrow$ Mon), and (GMP-like $\rightarrow$ Neu). Three-fold cross-validation was performed on this dataset.

Data 27 & 28

These two datasets were generated using ODE-based dynamics with `scvelo.datasets.simulation()` in scVelo [11]. Each dataset contains 1,500 cells, with Data 27 comprising 100 genes and Data 28 comprising 500 genes. Transcription, splicing, and degradation rates were fixed at $\alpha = 5$, $\beta = 0.3$, and $\gamma = 0.5$, respectively. Three-fold cross-validation was performed on both datasets.

Data 29 & 30

These two datasets were generated using SDE-based dynamics with `sdevelo.SimData()` in SDEvelo [39]. Each dataset contains 1,600 cells, with Data 29 comprising 100 genes and Data 30 comprising 500 genes. The number of trajectories K was fixed at 8. Three-fold cross-validation was performed on both datasets.

Data 31 & 32

These two datasets were generated using `dyngen` [50], a multimodal simulation framework that models gene regulatory networks and transcriptional kinetics. Each dataset contains 1,500 cells with 100 target genes, 100 housekeeping genes, and a set of transcription factors determined by the specified trajectory backbone. Data 31 uses a linear backbone, whereas Data 32 uses a bifurcating backbone. Key simulation parameters were `tau = 0.01`, `census_interval = 1`, and `num_simulations = 100`. Three-fold cross-validation was performed on both datasets.

Data 33 & 34

These two datasets were generated using ODE-based dynamics with `scvelo.datasets.simulation()` in `scVelo`. Data 33 consists of seven datasets with cell counts ranging from $n = 1,000$ to $1,000,000$, with fixed gene counts at 1,000. Data 34 consists of six datasets with gene counts ranging from $p = 1,000$ to $20,000$, with fixed cell counts at 10,000. Transcription, splicing, and degradation rates were fixed at $\alpha = 5$, $\beta = 0.3$, and $\gamma = 0.5$, respectively.

11 Note S11: Theoretical proof of the effective entropy score

Given a velocity graph computed using `scvelo.tl.velocity_graph()`, we construct a cell-cell transition matrix with `scvelo.tl.transition_matrix()` by setting the scale parameter to 30. For a given cell c , we sort its outgoing transition probabilities in descending order and denote them by $P_{(i)}$, where $i = 1, \dots, N$ and N is the total number of cells. We only consider the largest probabilities whose cumulative sum reaches at least 0.95. This truncation prevents infinitesimal probabilities arising from floating-point computation from artificially inflating the score. We define the effective number of these top probabilities as:

$$m_c^{(0.95)} = \min \left\{ m : \sum_{i=1}^m P_{(i)} \geq 0.95 \right\}. \quad (4)$$

Let $\tilde{p}_c$ denote the renormalized probability distribution over these top $m_c^{(0.95)}$ states. Then, we define the EES of cell c as

$$\text{EES}_c = \exp(H(\tilde{p}_c)), \quad (5)$$

where

$$H(\tilde{p}_c) = - \sum_{i=1}^{m_c^{(0.95)}} \tilde{p}_{c,i} \log \tilde{p}_{c,i} \quad (6)$$

is the Shannon entropy [64]. The overall EES is then defined as

$$\begin{aligned} \text{EES} &= \frac{1}{N} \sum_{c=1}^N \text{EES}_c \\ &= \frac{1}{N} \sum_{c=1}^N \exp \left(- \sum_{i=1}^{m_c^{(0.95)}} \tilde{p}_{c,i} \log \tilde{p}_{c,i} \right). \end{aligned} \quad (7)$$

We next demonstrate from a theoretical perspective that this metric accounts for both sparsity and uniformity. Let $u_c = (1/n_c, \dots, 1/n_c)$ denote the uniform distribution over n_c states, where $n_c = m_c^{(0.95)}$. Since $\tilde{p}_c$ is a probability distribution over these n_c states, we have

$$\begin{aligned} D_{\text{KL}}(\tilde{p}_c \parallel u_c) &= \sum_{i=1}^{n_c} \tilde{p}_{c,i} \log \frac{\tilde{p}_{c,i}}{1/n_c} \\ &= \sum_{i=1}^{n_c} \tilde{p}_{c,i} \log \tilde{p}_{c,i} + \log n_c. \end{aligned} \quad (8)$$

Therefore,

$$- \sum_{i=1}^{n_c} \tilde{p}_{c,i} \log \tilde{p}_{c,i} = \log n_c - D_{\text{KL}}(\tilde{p}_c \parallel u_c). \quad (9)$$

Thus, EES_c can be rewritten as

$$\begin{aligned} \text{EES}_c &= \exp(\log n_c - D_{\text{KL}}(\tilde{p}_c \parallel u_c)) \\ &= \underbrace{n_c}_{\text{Sparsity}} \underbrace{\exp(-D_{\text{KL}}(\tilde{p}_c \parallel u_c))}_{\text{Uniformity}}. \end{aligned} \quad (10)$$

This theoretical derivation demonstrates that the proposed metric explicitly accounts for both transition sparsity and uniformity.