Cell2State: Learning Cell State Representations From Barcoded Single-Cell Gene-Expression Transitions

Anonymous authors
Paper under double-blind review

Abstract

Genetic barcoding coupled with single-cell sequencing technology enables direct measurement of cell-to-cell transitions and gene-expression evolution over a long timespan. This new type of data reveals explicit state transitions of cell dynamics. Motivated by dimension reduction methods for dynamical systems, we develop a *cell-to-state* (cell2state) learning method that, through learning from such multi-modal data, maps single-cell gene expression profiles to low-dimensional state vectors that are predictive of cell dynamics. We evaluate the cell2state method using barcoded stem cell dataset (Biddy et al. (2018)) and simulation studies, compared with baseline approaches using features that are not dynamic-aware. We demonstrate the merits of cell2state in challenging downstream tasks including cell state prediction and finding dynamically stable clusters. Further, our method reveals potential latent meta-states of the underlying evolution process. For each of the meta-states, we identify a set of marker genes and development pathways that are biologically meaningful and potentially expand existing knowledge.

1 Introduction

With the explosive amount of data from single-cell genomics studies, one remaining major challenge is the lack of ability to understand cell transition on the individual level. Conventional methods for analyzing single-cell dynamics are mostly based on "ensemble" analysis (Kester & van Oudenaarden, 2018; Tanay & Regev, 2017; Bacher & Kendziorski, 2016; Stegle et al., 2015). Such analysis reveals population-level trends, but cannot reveal behaviors of individual cells.

Recent advances in genomic technology have enabled scientists to *directly measure cell lineages* and connect two cells that are far apart in the time course. This is opposed to inferring lineages from snapshots at nearby time points. The gene barcoding approach works by inserting into each cell a DNA sequence, i.e., a barcode, that randomizes across cells so that no two cells bear the same sequence (Woodworth et al., 2017). As the cell divides and differentiates, its descendants can be identified based on sequencing the label. This concept has been utilized recently in single-cell analysis of embryonic development (Yao et al., 2017; Wagner et al., 2018), stem cell reprogramming (Biddy et al., 2018), and fate determination in hematopoiesis (Weinreb et al., 2020). This genetic barcoding approach enables tracking of evolutionary trajectories across individual cell lineages.

In this paper, we focus on the new type of single-cell data enabled by the single-cell gene barcoding technology. The barcode can directly connect parent cells with their descendants over a long time span. Thus, gene barcoding coupled with RNA-seq generates pairs of gene-expression transitions $\{(X, X')\}$, where each (X, X') is the gene expression profiles for a parent cell at an early time point and one of its descendants at the later time point. One may view the gene-expression profile X as the raw state of a cell, which is a high-dimensional vector. Such state-transition data makes it possible to learn about a cell's law of state transition. In other words, the new data type can let us decode the single-cell transition law in a way similar to system identification for dynamical systems.

We wish to learn mathematical abstractions of cell gene-expression states, i.e., a map $\mathbf{\Psi}$ from gene-expression profile X to a vector $\mathbf{\Psi}(X)$ of lower dimension. A good cell state abstraction should be low-dimensional and predictive, compressing predictive signals from gene-expression levels in a compact vector. In other words, we hope to maximize the mutual information $I(\mathbf{\Psi}(X), X')$ between the embedded parent cell state and its descendant. Ideally, we hope to achieve a nearly lossless encoding of states, i.e., $I(\mathbf{\Psi}(X), X') \approx I(X, X')$. To learn such $\mathbf{\Psi}$ from noisy high-dimensional gene-expression data, we build on ideas from dynamical system theory, kernel machine learning, and low-rank optimization. In the area of molecular dynamics, Schütte et al. (2011) showed that the leading spectrum of transfer operator can be used to identify coresets for faster simulation. In reinforcement learning, computers need to figure out state abstraction of unknown transition systems in order to learn to control quickly (Sutton et al., 1998). Sun et al. (2019) developed a state embedding method to analyze game trajectories to significantly reduce the state dimension of one-player Atari games. See **Appendix A** for more discussion on related works.

A summary of our work:

- Building on the spectral compression ideas from dynamic systems, we develop a cell-to-state (cell2state) representation learning method for analyzing gene-expression transition data made available by the single-cell barcoding technology. The cell2state algorithm is trained using gene-expression transition pairs; it finds a mapping to approximate and embed the transition distributions in a low-dimensional space. The embedding map is learned by first "lifting" the data's dimension by random feature projection, then estimating a large matrix embedding of the transition distributions, and finally "compressing" the dimension by low-rank optimization.
- We provide information-theoretic analysis of the learnt cell2state embedding $\hat{\mathbf{\Psi}}$. In particular, we show that, upon appropriate quantization, the embedding map can be used to encode raw gene-expression data into a small number of bits. We show that this encoding can be nearly lossless, and we establish sample complexity bounds for preserving the mutual information between parent and descendants up to 1 bit.
- We apply cell2state to a published single-cell barcoded RNA-seq dataset for studying stem cells (Biddy et al., 2018). In this analysis, we used cell2state to map early-day cells to state vectors of dimension ≤ 8. Via the cell2state map, the cell populations demonstrated sharp polytope structures, where distinct vertices provide early signals that predict diverse dynamics and cell fates. To evaluate the learned cell state vectors, we test them in three downstream tasks: **(i)** finding dynamically stable cell cluster; **(ii)** early prediction of cell dynamics, such as cell descendants' activities or fates, based on low-dimensional state vectors; **(iii)** subpopulation analysis to identify marker genes that signal distinct cell dynamics. Across these tasks, we observe substantially improved performance using the learned cell states, as compared with baselines that use either the raw data or features that are not dynamics-aware. In particular, our results show that cell2state achieves similar/better level of prediction accuracy using $\leq$7 dimensions, compared to neural networks that use raw gene expressions as input (up to 5000 dimensions).
- Further, we identify and examine subpopulations of cells that have the most representative low-dimensional states (in other words, subsets of cells that are close to likely meta-states, under the assumption of a latent-state model). These subpopulations are used to identify biologically relevant marker genes. These marker genes identified by cell2state are known to relate to stem cell reprogramming and epigenetic regulations.

2 Markov Branching Process Model for Single-Cell Dynamics

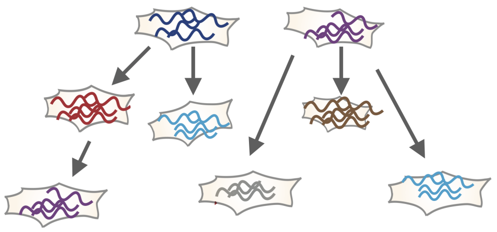
We model the time-course dynamics of gene expressions as a *branching diffusion process* (Edwards (1970); see **Figure 1**). Let X_t denote the gene-expression profile of a cell at a time point t, which is a high-dimensional vector. Each X_t has a random number of descendants X_{t+1}^i, $i = 1, \ldots, N$ with independent and identical distributions.

Definition 1. *Define $p(X_{t+1}|X_t)$ as the transition function for the gene expression profile X_t to evolve to a collection of descendants $\{X_{t+1}^{(i)}\}_{i=1}^N$ in a fixed amount of time, i.e., for any measurable set S,*

$$p(S|X_t) = \mathbb{E}\left[\sum_{i=1}^{N} \mathbf{1}\{X_{t+1}^{(i)} \in S\} \mid X_t\right],$$

where N is also a random variable.

Note that p is not necessarily a probability density function because a cell could have multiple descendants. If the growth rate is such that $\mathbb{E}[N|X_t] = \int p(y|x)dy > 1$, we say the cell is actively growing.



Gene-expression transition distribution: $p(X'|X)$
Growth function: $\lambda(X) = \mathbb{E}[no.descendants|X]$

Figure 1: Cell divides and differentiates, modeled as a Markov branching process.

Definition 2. *Let $\mathbf{P}$ be the transition operator of the branching diffusion process with transition function p, given by*

$$\mathbf{P}f(X) = \mathbb{E}\left[\sum_{i=1}^{N} f\{X_{t+1}^{(i)} \in S\} \mid X_t = X\right].$$

In single-cell analysis, the transition function, p, and operator, $\mathbf{P}$, are infinite-dimensional, thus estimating them is largely intractable from finite noisy data. Like many other dynamic systems, cell dynamics is often driven by a small set of marker genes, and thus, it may admit an intrinsic low-dimension structure. We make the following assumption:

Assumption 1. *Let $\mathcal{H}$ be a space of functions. There exists a r-dimensional embedding map $\Phi^* \subset \mathcal{H}$ such that*

$$\mathbf{P}f \in \mathit{Span}(\Phi^*), \forall f \in \mathcal{H}.$$

Here, $\mathcal{H}$ will be specified later.

Such low-rank structure of Assumption 1 naturally exists in dynamical processes that admit latent states. Suppose that each x can be represented as a mixture over meta-states $\{z\}$ such that

$$p(x'|x) = \sum_z p(x'|z)p_Z(z|x).$$

This is a common latent-state model for stochastic processes; see **Figure 2** for an illustration. In the single-cell context, a meta-state is often referred to as a "cell type", which has a distinct "pathway" (i.e., future dynamics). The "cell type" is defined as a function of the gene-expression profile, but the function is unknown and to be learnt. In this latent-state model , let $\Phi^*(x) = p_Z(\cdot|x)$. Then we can verify that Assumption 1 holds. In this case, finding the embedding map Φ^* would make it possible to recover the set of meta-states $\{z\}$ and aggregation distributions $p_Z(\cdot|x)$.

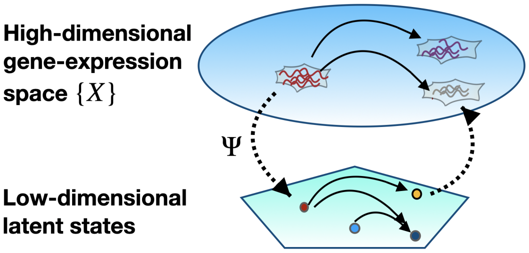

High-dimensional gene-expression space $\{X\}$
Ψ
Low-dimensional latent states

Figure 2: A latent state model, where the optimal embedding $\mathbf{\Psi}^*$ maps raw cell states to latent meta-states.

3 Mapping Gene Expressions To Low-Dimensional Cell States

Recall that our goal is to find mathematical abstractions of high-dimensional expression profiles $\{X_t\}$. We will estimate an embedding map from gene-expression profiles to a low-dimensional vector space: $\mathbf{\Psi} : x \in \mathbb{R}^d \mapsto \mathbf{\Psi}(x) \in \mathbb{R}^r$. Ideally, we hope $\mathbf{\Psi}(X_t)$ to be low-dimensional while still containing as much information about X_{t+1} as possible.

3.1 Embedding the transition operator into a functional space

Consider a kernel mean embedding of the transition operator $\mathbf{P}$: $Q = \Pi_{\mathcal{H}} \mathbf{P} \Pi_{\mathcal{H}}$, where the projections are with respect to appropriate norms. By assumption, we can verify that $\mathrm{rank}(Q) < r$. We seek to estimate Q from cell transition data, perform singular value truncation, and then find the low-dimensional embedding map by transforming the left single functions of Q.

To guarantee the function space $\mathcal{H}$ is sufficiently *expressive*, we adopt a kernel composition approach for "lifting" the dimensions. First, we construct an initial kernel K_0 that best fits the dataset's topology and preserves neighborhood relations. To find such a K_0, we can leverage existing dimension reduction methods for single-cell data analysis, such as PCA, manifold-based, and graph-based methods (see Kester & van Oudenaarden (2018); Tanay & Regev (2017); Bacher & Kendziorski (2016); Stegle et al. (2015) for reviews). Then, we construct the kernel function $K = K_0 \circ K_1$ by taking the composition between K_0 and another kernel function K_1 (e.g., the Gaussian kernel) - this step would further lift the problem's dimension and improve the function space's expressibility. One can also take compositions of multiple kernels to mimic a multi-layer neural network (Cho & Saul, 2009).

3.2 Low-rank compression of cell states via random features

We propose a kernelized state embedding method based on random feature projection for computing an estimator $\hat{\mathbf{\Psi}}$ from transition data $\{(X_i, X_i')\}_{i=1}^N$, which can be obtained from cell trajectories. For analyzing single-cell sequencing data and embedding transition distributions, we will choose the function space $\mathcal{H}$ with the kernel function K tailored to the data's geometry.

Suppose we have chosen a kernel function K (for example, a Gaussian kernel). We perform nonparametric estimation of $\mathbf{\Psi}^*$ by *generating a large number of random features to approximate the kernel space in large finite dimensions.* Then, we downsize the estimator by using spectral decomposition. In the case where each parent cell has a single descendant, the cell2state method works by (informally):

(1) Generate random Fourier functions $\phi(\cdot) = (\phi_1(\cdot), \ldots, \phi_d(\cdot))^\top$ by randomized decomposition of its kernel function K to approximately span $\mathcal{H}$ (Rahimi & Recht, 2008).

(2) Estimate a finite matrix embedding of the scaled condition probability distribution $\frac{1}{\sqrt{p(x')}} p(x'|x)$ by $\hat{P} = \Sigma_0^{-1/2} \left(\frac{1}{N} \sum_{i=1}^N \phi(X_i)\phi(X_i')^\top \right) \Sigma_1^{-1/2}$, where Σ_0, Σ_1 are covariances at the two time points.

(3) Let $\hat{\mathbf{\Psi}}(\cdot) = (\hat{U}_r \hat{\Lambda}_r)^\top \Sigma_0^{-1} \Phi(\cdot)$ where $\hat{U}_r, \hat{\Lambda}_r$ are from top r truncation of the SVD of $\hat{P}$.

See **Algorithm 1** in the Appendix for the full description of the cell2state algorithm, which also handles the case where cells have multiple descendants. Given a cell's gene expression profile x, the vector $\hat{\mathbf{\Psi}}(x)$ can be viewed as a low-dimensional mean embedding of the transition function $p(\cdot|x)$. Thus it should be predictive of this cell's future dynamics.

Runtime Complexity of Algorithm 1. The algorithm uses random features and singular value truncation, both designed for maximal computation efficiency. The overall runtime for training is at most $O(n + nD^2 + D^3)$, where n is number of cells, D is number of Fourier features ($\leq n$). This is the same complexity as computing covariances and PCA in the random feature space. After training, querying the embedding map $\hat{\mathbf{\Psi}}$ takes only $O(rD)$ time. In our experiments, Algorithm 1 runs in seconds, while training an MLP (a deep neural network) for cell fate prediction takes 10-15 minutes.

3.3 Information-Theoretical Analysis

In this section, we analyze the information-theoretic property of the cell2state embedding map $\hat{\mathbf{\Psi}}$. Assume without loss of generality that the feature $\phi(x)$ is upper bounded by $\|\phi(x)\|^2 \leq C, \forall x$. Let $P = \mathbb{E}[\hat{P}]$. Using an analysis similar to Sun et al. (2019), it can be shown that the distance distortion of state embedding map satisfies

Theorem 1. *Let Assumption 1 hold. With probability $1-q$, then for all x, y in the dataset,*

$$|\|\hat{\mathbf{\Psi}}(x) - \hat{\mathbf{\Psi}}(y)\| - \|p(\cdot|x) - p(\cdot|y)\|_{\mathcal{H}}| \leq \frac{16C\kappa}{\gamma}\sqrt{\frac{\log\frac{2d}{q}}{N}} + \frac{32C^{\frac{3}{2}}\kappa}{\gamma^{\frac{3}{2}}}\frac{\log\frac{2d}{q}}{N}$$

where d is the dimension of ϕ, and N is the number of sample transitions, $\gamma^{-1} = \max\{\|\Sigma_0^{-1}\|, \|\Sigma_1^{-1}\|\}, \kappa = \frac{\|P\|}{\sigma_r(P)}$.

Further, we will apply quantization to $\hat{\mathbf{\Psi}}$ and encode the parent cell data into finitely many bits. Suppose the state space admits a block structure, i.e., we have a mapping $\Omega^* : \mathcal{X} \to [k]$, such that

$$p(\cdot|x) = p(\cdot|x'), \qquad \forall x, x' \quad s.t. \quad \Omega^*(x) = \Omega^*(x').$$

Suppose the SVD of P is $P = U_r\Lambda_r V_r^\top$ and let $\mathbf{\Psi}(x) = (U_r\Lambda_r)^\top\Sigma_0^{-\frac{1}{2}}\phi(x)$. Let $A = \{A_j\}$ be an arbitrary quantization of $\hat{\mathbf{\Psi}}(\mathcal{X})$ such that $\|x - x'\| \leq \frac{\delta}{8}, \forall x, x' \in A_j, \forall j$, where $\delta = \min_{x,x':\mathbf{\Psi}(x)\neq\mathbf{\Psi}(x')} \|\mathbf{\Psi}(x) - \mathbf{\Psi}(x')\|$. In addition, define the encoding map $\hat{\Omega}_{\hat{\mathbf{\Psi}}}$ such that

$$\hat{\Omega}_{\hat{\mathbf{\Psi}}}(x) = A_j, \qquad \text{iff} \qquad \hat{\mathbf{\Psi}}(x) \in A_j.$$

We can show that the learned encoding map largely preserves the mutual information between a parent cell's raw gene expression and its descendants' gene expression profiles. More precisely, we show that the loss of information can be bounded and estimated, as follows

Theorem 2. *Let Assumption 1 hold with the aforementioned block partition structure. The estimated encoding map $\hat{\Omega}_{\hat{\mathbf{\Psi}}}$ is nearly loss-less in the following sense:*

$$\mathbb{E}[I(\hat{\Omega}_{\hat{\mathbf{\Psi}}}(X), X')] \geq I(X, X') - 2d\exp\left(-\frac{3\gamma^2 N\delta^2}{800C^2\kappa^2}\right)\log k,$$

where the expectation is over the distribution that generates the transition data.

Therefore, in order to ensure $\mathbb{E}_{\hat{\mathbf{\Psi}}}[I(\hat{\Omega}_{\hat{\mathbf{\Psi}}}(X), X')] \geq I(X, X') - \Delta$, we need the sample size $N \geq \mathcal{O}\left(\frac{C^2\kappa^2}{\gamma^2\delta^2}\log\frac{d\log k}{\Delta}\right)$. When $\Delta = 1$, we can quantify the sample size needed to preserve mutual information up to 1 bit difference.

4 Experiment with Cell Reprogramming Data

We use the single-cell gene expression data with genetic barcoding tags from Biddy et al. (2018)to test our cell state embedding method. Biddy et al. (2018) tracked reprogramming of mouse embryonic fibroblasts (MEFs) to induced endoderm progenitors (iEPs) by using *CellTagging* technology. During this dynamic reprogramming process, DNA barcodes in the form of randomized nucleotides were delivered into pools of cells and remained in the cell via lentiviral genome integration. Through identifying cells with the same DNA barcode sequence, we can recover the lineage relationships between parental and descendent cells. Barcoded single-cell gene expressions were collected using scRNA-seq at 8 time points over a course of 28 days (**Figure 3(a)** for a tSNE visualization). A cell's raw gene-expression profile is a vector of dimension 28,001.

4.1 Cell2state maps ancestral cells to a tetrahedron and implies the existence of at least 4 meta-states

For our experiment, we pick those cells profiled on day 12 and day 21, as parent cells and descendants respectively (**Figure 3(b)**), as these time points had a reasonably high number of sampled cells and pairs of genetic tags to establish lineage trajectories. After removing low-quality cells, we obtain a count matrix retaining 6233 cells (1997 on day 12 and 3509 on day 21) and spanning 17845 genes. We then process the single-cell data using their cell tags and yield N =165716 cell-to-cell transition pairs. We pick the kernel K to be a composition between the principal component map (obtained by PCA) and a Gaussian kernel with tunable width γ. Then, we apply the cell2state method and compute the embedding map $\hat{\mathbf{\Psi}}$. We visualize in **Figure 3(c)** the top three features of the parent cells given by $\hat{\mathbf{\Psi}}$. These parent cells visibly form a tetrahedron with four vertices in the embedding space, implying at least four potential meta-states with distinct future pathways. Further, we validate that cells near the vertices and cells at the center have diverse cell fates based on gene expression profiles of their descendants on day 21 (**Figure 3(d)**).

4.2 Evaluation of low-dimensional cell states on downstream tasks

Next we investigate the predictive power and biological relevance of the low-dimensional cell state embeddings $\{\hat{\mathbf{\Psi}}(x)\}$. We consider three downstream tasks: clustering, prediction, and gene marker/pathway analysis.

4.2.1 Dynamics-stable cell clustering

A fundamental task in studying single-cell dynamics is to cluster cells into representative "cell types/states" that are biologically meaningful and stable across time. Many innovative tools have increasingly permitted the identification and classification of cell types/states (Kester & van Oudenaarden (2018); Wagner & Klein (2020)), but most of these tools do not account for cells' temporal dynamics. With new barcoding data, we seek to integrate information from both gene expression and lineage trajectories to find dynamic-stable cell clusters.

Dynamic stability of embedding clusters. We apply k-means to the parent cell embeddings and identify 7 major clusters (**Figure 3(f)**). We evaluate the quality of the cluster assignment via examining their respective descendant distributions (**Figure 3(g)**). We clearly observe that the clusters are dynamically stable, i.e., descendants from the same parental cluster tend to stay near to one another (panel f-g). For comparison, the same cluster assignment in the raw data of parent cells (**Figure 3(e)**) is unstructured and mixed up. The contrast between **Figure 3(e-f)** suggests that these dynamically stable cluster structures were hidden in the raw data, but can be revealed by our low-dimensional embedding.

Comparison via computing the cluster assignment's descendant inertia. We further compare the dynamic stability of cell2state clusters with the widely-used graph-based clustering method in Seurat that integrated Louvain algorithm (Blondel et al. (2008); Butler et al. (2018); Stuart et al. (2019)To quantitatively evaluate the dynamic stability, we compute the inertia of cluster assignments using descendant data. For the cluster assignment $\Omega = \{\Omega_1, ..., \Omega_k\}$, we evaluate the inertia of Ω over all descendant cells, i.e., $inertia(\Omega) = \sum_{i\in[k]} \min_{\mu_i} \sum_{X_1\in\Omega_i} ||X_1 - \mu_i||_2^2$. The inertia measures the level of concentration of the clusters, thus if the clusters with smaller inertia across time are more dynamically stable. **Figure 3(h)** shows the descendant inertia values obtained from different cluster assignments. It shows that the cell2state k-means clusters from parent cells achieved a small inertia on descendant cells, similar to that from directly clustering descendants. Both are significantly smaller than assigning clusters on raw data without doing cell2state, which are not dynamics-stable. This validates that *cell2state* yields dynamics-stable cell clusters, i.e., inertia of the clusters remains small across time.

4.2.2 Early prediction of descendant cells' proliferation activity

Next, we evaluate the predictive power of the learned cell state embeddings. One of the fundamental cell states that connects to the underlying biology of stem cells is a cell's proliferation potential. Hence, we seek to use the cell state embedding learned for parent cells to predict proliferation activity of these cell's descendants. Note that the parent and descendant in our experiments are 9 days apart. This is a relatively long time span, and to the best of our knowledge, there does not exist a prior attempt for such early prediction with mammalian cells.

Prediction targets. We measure the proliferation activities of cells on day 21 using two well-established metrics: (i) the G2M cell cycle score, defined by prior studies and widely used

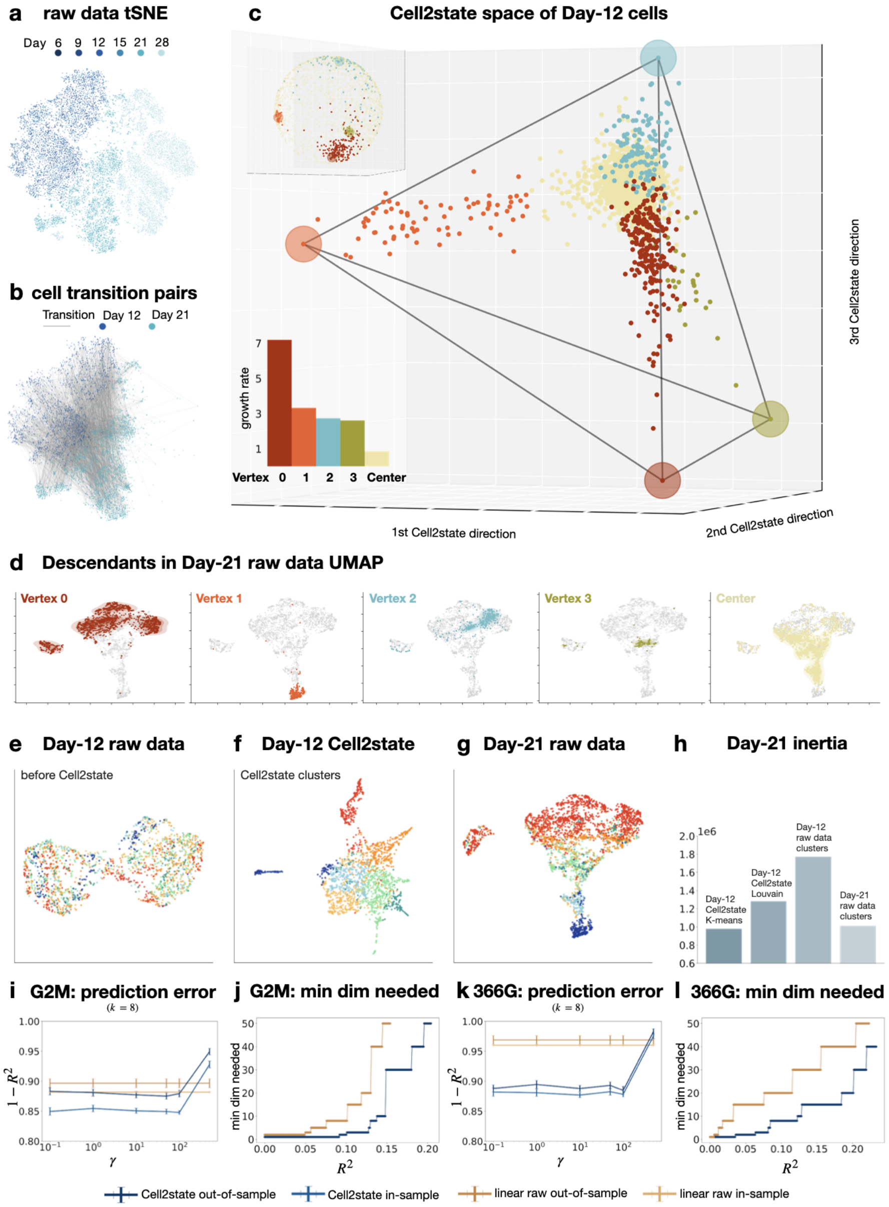


Figure 3: **Cell2state analysis and dimension reduction of of single-cell reprogramming data (Biddy et al., 2018). a.** The single-cell reprogramming data (Biddy et al. (2018)) visualized in tSNE(van der Maaten & Hinton (2008)). **b.** Transition network between day-12 and day-21 cells recovered from cell barcodes. **c.** Cell2state maps day-12 cells to a tetrahedron with 4 visible vertices (top three dimensions visualized). Unit ball projection of top 3 dimensions of cell2state features (upper left) and exhibit diverse growth rates(lower left). **d.** Four vertices identified by day-12 Cell2state features lead to distinct descendent distributions visualized in UMAP(McInnes et al. (2020)). **e-g.** Cluster assignment learnt based on cell2state features, visualized via UMAP with Day-12 raw data (e), Day-12 cell2state feature space (f), and Day-21 descendent data (g). **h.** Comparison of cluster methods for dynamic stability, measured via descendant inertia. **i-l.** Using Cell2state features for early prediction of descendants' proliferation activities (G2M and 366G scores), compared with prediction using raw data. Plots(i,k) give the in-sample and out-of-sample prediction accuracy, where $1/\sqrt{2\gamma}$ is the Gaussian kernel width. Plots (j,l) gives the minimal dimension needed to reach certain prediction accuracy.

a **Day-12 Cell2state vertex analysis: top positive marker genes**

Vertex 0	Vertex 1	Vertex 2	Vertex 3	Center
Igfbp2	**Npnt**	H19	**Sdpr**	**S100a7a**
Comp	**Sox11**	**Il6st**	**Sgk1**	Apoa1
Maoa	**Ncam1**	**Fbln2**	**Cpe**	**S100a13**
Csrp1	Col6a2	**Fstl1**	Vgll3	**Clu**
H2afz	Kitl	Csrp2	Fscn1	Crabp1
Scd2	Lxn	Tinagl1	Itga5	**Timp3**
Eva1b	Tuba1a	Fn1	Serf2	Fam129a
H6pd	Col12a1	Ddah1	Rpl37a	Ctgf

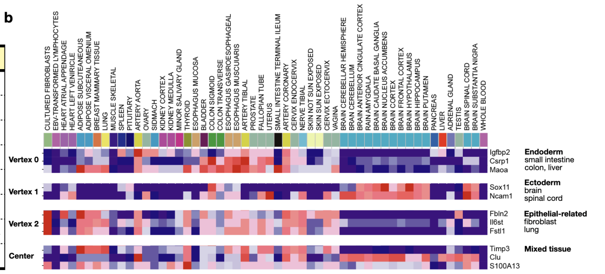


Figure 4: **Biological interpretation of the cell2state map learnt from (Biddy et al., 2018). a.** Top positive marker genes associated with each vertex/center obtained by DESeq2(Love et al. (2014)), ranked by fold-change with positive logFC(>0.5). Those genes marked in **bold** were novel markers identified by cell2state and known to be related to stem cell biology based on recent literature. **b.** Identifying major cell development programs associated with each subpopulation by top ranked marker gene via GTEx(Carithers & Moore (2015)). Vertex 0/1/2 is enriched for endoderm/ectoderm/epithelial-related programs respectively; the center has mixed tissue pattern.

in single-cell data analysis (Tirosh et al. (2016); Butler et al. (2018); Stuart et al. (2019)); (ii) the average expression level of 366 proliferation genes, designated by gene ontology (GO) annotations (Ashburner et al. (2000); Consortium (2019)).

Results and comparison. Figure 3(i,k) visualize the in-sample and out-of-sample prediction error as the model parameter (i.e., γ determines kernel width) varies. For comparison, we also trained predictors that use the principal components of raw data directly for the same prediction task, which we treat as the baselines. Observe that the cell2state-based predictors consistently outperforms the baseline across instances. **Figure 3(j,l)** illustrate the embedding dimension needed as the predication accuracy level varies. The results suggest that our cell embedding approach has substantially reduced dimensionality of the raw data, achieving similar or higher prediction accuracy using a fraction of dimensions.

4.2.3 Finding marker genes and cell development pathways

Finally, we seek to interpret the cell state embeddings and evaluate if these learned low-dimensional structures are biologically relevant. We examine the polytope structure of parent cells in the cell2state embedding space (**Figure 3(c)**). Based on the cluster assignment, we then select sample cells that are close to vertices (top 50% of cells in the cluster) and apply DESeq2 (Love et al. (2014)) to these representative cells from each vertex. DESeq2 is a tool that allows one to identify significantly enriched genes in a subpopulation as compared to the full data. This allows us to find marker genes that distinguish individual subpopulation from one another.

Finding top ranker marker genes Table in **Figure 4(a)** summarizes a list of top positive marker genes associated with each vertex as ranked by fold-change (logFC). We note that our results, from day12 cells only, already recover the majority of the genes/pathways implicated in Biddy et al. (2018), such as Apoa1, Col1a2, Peg3, as well as Wnt and Igf2 pathways (highlighted in bold).

Analysis of the development pathways of each set Additionally, our analysis demonstrates that each subpopulation has distinctive signature tissue programs via Genotype-Tissue Expression (GTEx)(Carithers & Moore (2015)), which reveals their putative cell fates and development pathways (**Figure 4(b)**).

5 Experiment with Synthetic Data

To further test cell2state, we artificially construct highly nonlinear, random cell dynamics for a simulated experiment. We use gene-expression data from Weinreb et al. (2020)but we added artificially nonlinear transitions between day2 and day6 cells. We generate simulated day2-day6 transitions with a Markov branching process as follows: We first partition the 2D UMAP embedding space for day2 cells into 3 regions/clusters using two concentric circles (**Figure 5(a)**). We also partition the 2D UMAP space for day6 cells into the 4 quadrant

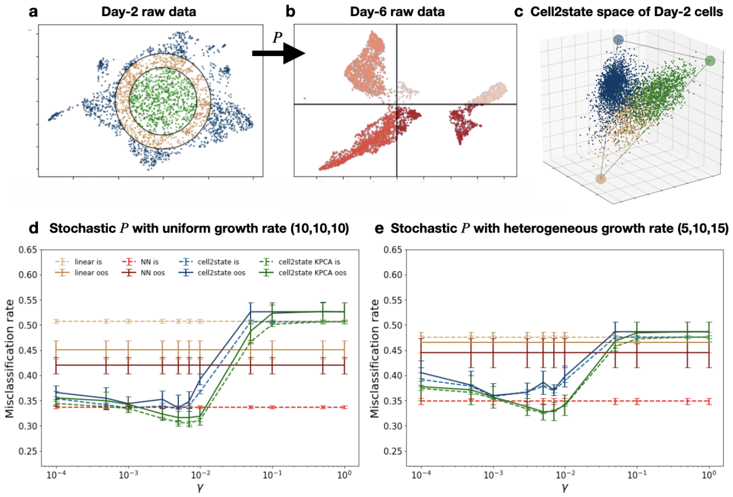


Figure 5: **Simulation experiment with synthetic nonlinear transition dynamics**. **a-b.** A Markov branching process, with the number of descendants sampled from a Poisson distribution, generates transitions from day 2 cells to day 6 cells. **c.** Cell2states maps day-2 cells into a polytope. **d-e.** Early prediction of dominant descendant cell fate for simulated data. Misclassification rate is fate prediction error; γ is the inverse squared length parameter of the Gaussian kernel.

regions, i.e, 4 cell fates (**Figure 5(b)**). For each day2 cell X, let $c(X) \in \{1, 2, 3\}$ denote its cluster assignment. Then the number of descendants is given by $N \sim \text{Poi}(\lambda_{c(X)})$, where λ_i is fixed for $i = 1, 2, 3$. We generate cell transitions using the cluster-to-cluster transition matrix P, by sampling descendants from the corresponding day6 cluster. This Markov branching process has highly nonlinear, discontinuous dynamics due to our artificial construction of the three initial regions and the four cell fates.

Application to cell fate prediction. We apply cell2state to the simulated cell transition data and learn low-dimensional state vectors of all parent cells. **Figure 5(c)** shows that cell2state maps the three parental clusters to a triangle, where each latent parental cluster is mapped to one vertex. We then use these learned cell states to predict the dominant cell fate of each parent cell, using a linear classifier. We consider both the cell2state based on random Fourier features and a variant of cell2state where exact kernel decomposition is used instead of random features. For comparison, we also tested the basic linear classifier and a multi-layer perceptron neural network, both taking raw gene-expression profiles as input for cell fate prediction (**Appendix B** for simulation experiment details).

Comparison with linear and neural network classifiers. Figure 5(d-e) illustrate the simulation results. We see that cell2state together with a basic linear classifier achieves the best out-of-sample accuracy across all experiments. The cell2state predictors using the low-dimensional states have comparable or slightly better performances than neural networks that take as input the raw data. This validates our theory that cell2state has largely compressed the high-dimensional information about the nonlinear cell dynamics into lower dimensions. Also note that, the basic linear classifier using raw data performs poorly in comparison to cell2state and neural networks, as it fails to capture the process's nonlinearity. Overall, this simulation demonstrates cell2state's ability to learn meaningful cell state representations even from highly irregular, nonsmooth dynamics.

6 Summary

We provide a random feature-based cell state embedding method for mapping single-cell gene expression profiles to low-dimensional representations based on barcoded gene-expression trajectories. Application to stem cell dataset and simulation studies suggests the learned cell state embedding carries predictive signals of cell dynamics.

References

Anna Alemany, Maria Florescu, Chloé S Baron, Josi Peterson-Maduro, and Alexander Van Oudenaarden. Whole-organism clone tracing using single-cell sequencing. *Nature*, 556 (7699):108, 2018.

Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25–29, 2000.

Rhonda Bacher and Christina Kendziorski. Design and computational analysis of single-cell rna-sequencing experiments. *Genome biology*, 17(1):63, 2016.

Brent A Biddy, Wenjun Kong, Kenji Kamimoto, Chuner Guo, Sarah E Waye, Tao Sun, and Samantha A Morris. Single-cell mapping of lineage and identity in direct reprogramming. *Nature*, 564(7735):219, 2018.

Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

Andrew Butler, Paul Hoffman, Peter Smibert, Efthymia Papalexi, and Rahul Satija. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nature biotechnology*, 36(5):411–420, 2018.

Latarsha J Carithers and Helen M Moore. The genotype-tissue expression (gtex) project, 2015.

Michelle M Chan, Zachary D Smith, Stefanie Grosswendt, Helene Kretzmer, Thomas M Norman, Britt Adamson, Marco Jost, Jeffrey J Quinn, Dian Yang, Matthew G Jones, et al. Molecular recording of mammalian embryogenesis. *Nature*, 570(7759):77, 2019.

Youngmin Cho and Lawrence Saul. Kernel methods for deep learning. In Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta (eds.), *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc., 2009. URL `https://proceedings.neurips.cc/paper/2009/file/5751ec3e9a4feab575962e78e006250d-Paper.pdf`.

Gene Ontology Consortium. The gene ontology resource: 20 years and still going strong. *Nucleic acids research*, 47(D1):D330–D338, 2019.

Anthony WF Edwards. Estimation of the branch points of a branching diffusion process. *Journal of the Royal Statistical Society: Series B (Methodological)*, 32(2):155–164, 1970.

Zhicheng Ji and Hongkai Ji. Tscan: Pseudo-time reconstruction and evaluation in single-cell rna-seq analysis. *Nucleic acids research*, 44(13):e117–e117, 2016.

Reza Kalhor, Kian Kalhor, Leo Mejia, Kathleen Leeper, Amanda Graveline, Prashant Mali, and George M Church. Developmental barcoding of whole mouse via homing crispr. *Science*, 361(6405):eaat9804, 2018.

Lennart Kester and Alexander van Oudenaarden. Single-cell transcriptomics meets lineage tracing. *Cell Stem Cell*, 23(2):166–179, 2018.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. URL `http://arxiv.org/abs/1412.6980`.

Stefan Klus, Feliks Nüske, Péter Koltai, Hao Wu, Ioannis Kevrekidis, Christof Schütte, and Frank Noé. Data-driven model reduction and transfer operator approximation. *Journal of Nonlinear Science*, 28(3):985–1010, 2018.

Miroslav Kratochvil, Abhishek Koladiya, Jana Balounova, Vendula Novosadova, Radislav Sedlacek, Karel Fivser, Jiri Vondrasek, and Karel Drbal. Som-based embedding improves efficiency of high-dimensional cytometry data analysis. *bioRxiv*, pp. 496869, 2019.

Michael I. Love, Wolfgang Huber, and Simon Anders. Moderated estimation of fold change and dispersion for rna-seq data with deseq2. *Genome Biology*, 15(12):550, 2014. doi: 10.1186/s13059-014-0550-8. URL `https://doi.org/10.1186/s13059-014-0550-8`.

Eugenio Marco, Robert L Karp, Guoji Guo, Paul Robson, Adam H Hart, Lorenzo Trippa, and Guo-Cheng Yuan. Bifurcation analysis of single-cell gene expression data reveals epigenetic landscape. *Proceedings of the National Academy of Sciences*, 111(52):E5643–E5650, 2014.

Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction, 2020.

Aaron McKenna, Gregory M Findlay, James A Gagnon, Marshall S Horwitz, Alexander F Schier, and Jay Shendure. Whole-organism lineage tracing by combinatorial and cumulative genome editing. *Science*, 353(6298):aaf7907, 2016.

Sumit Mukherjee, Yue Zhang, Sreeram Kannan, and Georg Seelig. Prior knowledge and sampling model informed learning with single cell rna-seq data. *bioRxiv*, pp. 142398, 2017.

Xiaojie Qiu, Qi Mao, Ying Tang, Li Wang, Raghav Chawla, Hannah A Pliner, and Cole Trapnell. Reversed graph embedding resolves complex single-cell trajectories. *Nature methods*, 14(10):979, 2017.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis (eds.), *Advances in Neural Information Processing Systems 20*, pp. 1177–1184. Curran Associates, Inc., 2008. URL `http://papers.nips.cc/paper/3182-random-features-for-large-scale-kernel-machines.pdf`.

Bushra Raj, Daniel E Wagner, Aaron McKenna, Shristi Pandey, Allon M Klein, Jay Shendure, James A Gagnon, and Alexander F Schier. Simultaneous single-cell profiling of lineages and cell types in the vertebrate brain by scgestalt. *bioRxiv*, pp. 205534, 2017.

Prajit Ramachandran, Barret Zoph, and Quoc V. Le. Searching for activation functions, 2017.

Sabrina Rashid, Darrell N Kotton, and Ziv Bar-Joseph. Tasic: determining branching models from time series single cell data. *Bioinformatics*, 33(16):2504–2512, 2017.

Abbas H Rizvi, Pablo G Camara, Elena K Kandror, Thomas J Roberts, Ira Schieren, Tom Maniatis, and Raul Rabadan. Single-cell topological rna-seq analysis reveals insights into cellular differentiation and development. *Nature biotechnology*, 35(6):551, 2017.

Wouter Saelens, Robrecht Cannoodt, Helena Todorov, and Yvan Saeys. A comparison of single-cell trajectory inference methods. *Nature biotechnology*, 37(5):547, 2019.

Tuhin Sarkar, Alexander Rakhlin, and Munther A Dahleh. Finite-time system identification for partially observed lti systems of unknown order. *arXiv preprint arXiv:1902.01848*, 2019.

Christof Schütte, Frank Noé, Jianfeng Lu, Marco Sarich, and Eric Vanden-Eijnden. Markov state models based on milestoning. *The Journal of chemical physics*, 134(20):05B609, 2011.

Manu Setty, Vaidotas Kiseliovas, Jacob Levine, Adam Gayoso, Linas Mazutis, and Dana Pe'er. Palantir characterizes cell fate continuities in human hematopoiesis. *bioRxiv*, pp. 385328, 2018.

Jaehoon Shin, Daniel A Berg, Yunhua Zhu, Joseph Y Shin, Juan Song, Michael A Bonaguidi, Grigori Enikolopov, David W Nauen, Kimberly M Christian, Guo-li Ming, et al. Single-cell rna-seq with waterfall reveals molecular cascades underlying adult neurogenesis. *Cell stem cell*, 17(3):360–372, 2015.

Satinder P Singh, Tommi Jaakkola, and Michael I Jordan. Reinforcement learning with soft state aggregation. In *Advances in neural information processing systems*, pp. 361–368, 1995.

Bastiaan Spanjaard, Bo Hu, Nina Mitic, Pedro Olivares-Chauvet, Sharan Janjuha, Nikolay Ninov, and Jan Philipp Junker. Simultaneous lineage tracing and cell-type identification using crispr–cas9-induced genetic scars. *Nature biotechnology*, 36(5):469–473, 2018.

Oliver Stegle, Sarah A Teichmann, and John C Marioni. Computational and analytical challenges in single-cell transcriptomics. *Nature Reviews Genetics*, 16(3):133, 2015.

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

Yifan Sun, Yaqi Duan, Hao Gong, and Mengdi Wang. Learning low-dimensional state embeddings and metastable clusters from time series data. *Advances in Neural Information Process Systems*, 2019.

Richard S Sutton, Andrew G Barto, et al. *Introduction to reinforcement learning*, volume 2. MIT press Cambridge, 1998.

Amos Tanay and Aviv Regev. Scaling single-cell genomics from phenomenology to mechanism. *Nature*, 541(7637):331, 2017.

Itay Tirosh, Benjamin Izar, Sanjay M Prakadan, Marc H Wadsworth, Daniel Treacy, John J Trombetta, Asaf Rotem, Christopher Rodman, Christine Lian, George Murphy, et al. Dissecting the multicellular ecosystem of metastatic melanoma by single-cell rna-seq. *Science*, 352(6282):189–196, 2016.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. URL http://jmlr.org/papers/v9/vandermaaten08a.html.

Saligrama R Venkatesh and Munther A Dahleh. On system identification of complex systems from finite data. *IEEE Transactions on Automatic Control*, 46(2):235–257, 2001.

Daniel E Wagner and Allon M Klein. Lineage tracing meets single-cell omics: opportunities and challenges. *Nature Reviews Genetics*, pp. 1–18, 2020.

Daniel E Wagner, Caleb Weinreb, Zach M Collins, James A Briggs, Sean G Megason, and Allon M Klein. Single-cell mapping of gene expression landscapes and lineage in the zebrafish embryo. *Science*, 360(6392):981–987, 2018.

Shuxiong Wang, Adam L MacLean, and Qing Nie. Soptsc: Similarity matrix optimization for clustering, lineage, and signaling inference. *bioRxiv*, pp. 168922, 2018.

Caleb Weinreb, Alejo Rodriguez-Fraticelli, Fernando D Camargo, and Allon M Klein. Lineage tracing on transcriptional landscapes links state to fate during differentiation. *Science*, 367(6479), 2020.

Joshua D Welch, Alexander J Hartemink, and Jan F Prins. Matcher: manifold alignment reveals correspondence between single cell transcriptome and epigenome dynamics. *Genome biology*, 18(1):138, 2017.

F Alexander Wolf, Fiona K Hamey, Mireya Plass, Jordi Solana, Joakim S Dahlin, Berthold Göttgens, Nikolaus Rajewsky, Lukas Simon, and Fabian J Theis. Paga: graph abstraction reconciles clustering with trajectory inference through a topology preserving map of single cells. *Genome biology*, 20(1):59, 2019.

Mollie B Woodworth, Kelly M Girskis, and Christopher A Walsh. Building a lineage from single cells: genetic techniques for cell lineage tracking. *Nature Reviews Genetics*, 18(4): 230, 2017.

Lin F Yang and Mengdi Wang. Sample-optimal parametric q-learning with linear transition models. *Proceedings of International Conference on Machine Learning*, 2019.

Zizhen Yao, John K Mich, Sherman Ku, Vilas Menon, Anne-Rachel Krostag, Refugio A Martinez, Leon Furchtgott, Heather Mulholland, Susan Bort, Margaret A Fuqua, et al. A single-cell roadmap of lineage bifurcation in human esc models of embryonic brain development. *Cell stem cell*, 20(1):120–134, 2017.

A Background and Related Works

In recent years, enabled by our ability to track and measure individual cells, single-cell analysis tools have demonstrated their significance in bringing next-level knowledge about cellular functions and regulations, thereby elucidating the mechanisms of human health and disease. As most biological or disease processes involve cells changing over time, a major need is to identify the lineage or state-transition of individual cells in single-cell analysis (Kester & van Oudenaarden, 2018). Using single-cell genomics (e.g. RNA-seq, proteomics), prior work demonstrated success based on mathematical inference and were broadly useful in understanding fundamental questions in cell evolution, differentiation, and development (Wolf et al., 2019; Kratochvil et al., 2019; Wang et al., 2018; Setty et al., 2018; Mukherjee et al., 2017; Welch et al., 2017; Qiu et al., 2017; Rizvi et al., 2017; Rashid et al., 2017; Ji & Ji, 2016; Shin et al., 2015; Marco et al., 2014). Nonetheless, as benchmarked in a recent meta-analysis on single-cell trajectory inference methods (Saelens et al., 2019), many existing tools have variabilities arising from the fact that, regardless of the underlying mathematics and statistical methodologies employed, inferred trajectories were constructed indirectly from genomics data such as gene-expression profiles. Thus, their performances are often sensitive to the qualities, dimensions, and underlying topologies of the original datasets (Saelens et al., 2019; Kester & van Oudenaarden, 2018). Additionally, most mathematical inference methods assume underlying "smoothness" of cell dynamics, yet discontinuous jumps and existence of "cycling" or "reversible" cell state transitions could confound these methods.

Hence, the latest single-cell studies began to employ new experimental methods to achieve genetic barcoding of individual cells for direct measurement of cell lineages (Woodworth et al., 2017; Kester & van Oudenaarden, 2018; Wagner & Klein, 2020). These new single-cell genomics methods and datasets, while more powerful, present significant challenges for computational and data analytical tools that we aim to take a substantial step to address in this proposal. Several recent single-cell studies leveraged DNA barcoding approaches, where each cell is given a static DNA sequence that randomizes across cells so that no two cells would bear the same sequence (Woodworth et al., 2017; Kester & van Oudenaarden, 2018). This sequence could then serve as a "tag" to identify cell lineage recoverable by single-cell sequencing. For example, this concept has been utilized in single-cell analysis of embryonic development (Yao et al., 2017; Wagner et al., 2018), stem cell reprogramming (Biddy et al., 2018), and fate determination in hematopoiesis (Weinreb et al., 2020). While static DNA barcoding is able to identify "clonal identity" across time, the emerging CRISPR barcoding technology can label cells using barcodes that evolve as the cell branches and reconstruct full lineage tree structures. An evolving barcoding system that gives each cell a unique tag will be useful for studying cell dynamics in a biological or disease process at finer resolution (Alemany et al., 2018; Kalhor et al., 2018; Raj et al., 2017; Spanjaard et al., 2018; Chan et al., 2019). These higher-resolution CRISPR barcoding tools let scientists track individual cell lineages more easily than ever. They present new open questions for single-cell data analysis. Hence,we are motivated to analyze individual cells' gene-expression trajectories made available by these barcoding tools, and delve into the underlying single-cell dynamics. Our goal is to develop a streamlined analysis pipeline, connecting these genetic barcodes and single-cell profiles (e.g. gene-expression), to help uncover the hidden mechanisms of cell evolution and fate determination.

With the explosive amount of data from single-cell genomics studies, one remaining untapped problem is the lack of ability to understand cell transition on the individual level. Even if explicit barcodes are available and can define individual cell transitions, to decode single-cell dynamics from such data, we need to solve a highly complex system identification and state estimation problem whereas the true system dynamics is hidden under limited, noisy, and high-dimensional observations. The complexity of decoding single-cell dynamics is due to three multiplicative factors: (1) the stochastic nature of cellular dynamics, (2) nonlinearity or even discontinuity of single-cell gene expression across time, with possible existence of "cycling" and "reversible" transitions, and (3) high dimensionality of gene expression profiles. Hence, new types of dynamic single-cell datasets call for analytical methods that match such complexity and scale.

Conventional methods for analyzing single-cell data are mostly based on “ensemble” analysis, without barcodes to delineate individual transition paths (Kester & van Oudenaarden, 2018). Our work focuses on data made available by novel genetic barcoding approach that enables tracking evolutionary trajectories across individual cells while simultaneous performing genomics read-out of cell states (Woodworth et al., 2017; Kester & van Oudenaarden, 2018; Wagner & Klein, 2020). In these barcoding experiments, a pool of designed or randomized DNA sequences are synthesized and delivered into cells so that each cell bears a different sequence. These sequences, uniquely tagging individual cells, enable the recovery of cell clonality or lineage using genomics methods such as sequencing (Woodworth et al., 2017; Kester & van Oudenaarden, 2018). Recent new development in genetic barcoding tools hold the promise to enable tracking cells at higher capacity, with finer temporal resolution, and for more complex models (Wagner & Klein, 2020). Taking the latest CRISPR barcoding technology for example, a fixed-length DNA barcode is inserted into individual cells, and the barcode dynamically change as the cell branches by nature of the CRISPR editing process (McKenna et al., 2016; Kalhor et al., 2018; Chan et al., 2019). These CRISPR barcodes could allow the reconstruction of lineage history of cells conveniently with end-point data collection, revealing each cell’s transition paths at scale (Wagner & Klein, 2020). In this case, we have pairs of gene-expression transitions $\{(X, X')\}$, where each (X, X') are the gene expression profiles for a parent cell at one time point and its descent at the next time point. This new type of datasets, particularly datasets from recent CRISPR-based barcoding experiments, calls for continued innovation to invent suitable data methods that we are proposing here.

Our method is related to state representation learning and spectral dimension reduction in reinforcement learning, dynamical systems and scientific computing. In particular, our kernelized state embedding estimator generalizes the notation of diffusion map to reproducing kernel spaces. For molecular dynamics,(Schütte et al., 2011) showed that the leading spectrum of the transition operator contains information on slow dynamics of the system, and it can be used to identify coresets upon which a coarse-grained Markov state model could be built. Please see (Klus et al., 2018) for comprehensive surveys on various spectral dimension methods for dynamical systems. They did not study statistical properties of these methods which motivated our research. In control and reinforcement learning, computers need to figure out state abstraction (e.g, state features, state aggregation) of unknown transition systems in order to learn to control quickly (Singh et al., 1995; Venkatesh & Dahleh, 2001; Sutton et al., 1998; Sarkar et al., 2019). One recent work (Yang & Wang, 2019) showed that the statistical complexity of reinforcement learning scales linearly with respect to dimension of the state representation. (Sun et al., 2019) developed a statistically proven kernelized state learning method for Markov processes. These inspirations from dynamic system and reinforcement learning led us to develop single-cell analysis tools in order to map cell states to low-dimensional embeddings.

B Implementation Details

Algorithm 1 gives the full description of the Cell2state algorithm. It takes as input collections of cell transition pairs between two timepoints $t_0 < t_1$, organized into N clones. Each clone $C_i = \{C_0^i, C_1^i\}$ consists of two sets, where C_0^i is a set of ancestor cells and C_1^i is a set of descendant cells. The ancestor/descendant cell sets, containing single-cell gene-expression profiles Y, are pairwise disjoint: $\forall i \neq j \in [N], C_0^i \cap C_0^j = \emptyset, C_1^i \cap C_1^j = \emptyset$. For a gene-expression profile Y_0 as an element in set C_0^i, we also denote the vector representation X_0 preprocessed from Y_0 with $X_0 \in C_0^i$ for simplification; this notation is applied to all cell sets $C_0^i, C_1^i, i \in [N]$. For every $X_0 \in C_0^i, X_1 \in C_1^i$, we consider them together as a cell-to-cell transition pair (X_0, X_1). In our experiments, we use the Gaussian kernel function, i.e., $K(X, X') = exp(-\gamma||X - X'||^2)$, which is a positive semidefinite shift-invariant kernel.

Steps 1-3 of Algorithm 1 corresponds to solving a regression formulation:

$$\min_P RSS = \sum_{i=1}^{N} \sum_{X_0 \in C_0^i} ||\Phi(X_0)^\top \Sigma_0^{-1/2} P - \frac{1}{|C_0^i|} \sum_{X_1 \in C_1^i} \Phi(X_1)^\top \Sigma_1^{-1/2}||^2.$$

In Steps 4-5 of Algorithm 1, we approximate $\hat{P}$ by $\hat{U}\hat{S}_{[1,..,r]}\hat{V}$ and construct the Cell2state map $\hat{\mathbf{\Psi}}(\cdot)$.

Algorithm 1 Cell2state Algorithm (cell-to-cell)

```
Input: Collection of cell-transition data $\{C^i = \{C_0^i, C_1^i\}\}_{i=1}^N$, Rank $r$, Kernel function $K$
Parameter: $\tilde{\lambda}$,$\gamma$,$D$,$d$,$\lambda_0$,$\lambda_1$
Notation: Let $A_0 = \bigcup_{i=1}^N C_0^i$, $A_1 = \bigcup_{i=1}^N C_1^i$.
1: Preprocess all gene-expression profiles $Y \in A_0 \cup A_1$ by a selected function $g(\cdot)$ and normalize: $X := \left(\tilde{\Sigma} + \tilde{\lambda} I\right)^{-1/2} [g(Y) - \tilde{\mu}_g] \in \mathbb{R}^{\tilde{d}}$, where $\tilde{\Sigma} = \sum_{Y \in A_0 \cup A_1} [g(Y) - \tilde{\mu}_g] [g(Y) - \tilde{\mu}_g]^\top$ and $\tilde{\mu}_g = \frac{1}{|A_0 \cup A_1|} \sum_{Y \in A_0 \cup A_1} g(Y)$.   // $\tilde{d} = 50$, $\tilde{\lambda} = 1$
2: Generate random Fourier functions and keep top $d$ PCs out of $D$ random Fourier features: $\Phi(\cdot) = [\Phi_1(\cdot), \Phi_2(\cdot), ..., \Phi_d(\cdot)]^\top$   // $D = 2000, d = 500$
    a: Draw $D$ iid samples $w_1, ...w_D \in \mathbb{R}^{\tilde{d}}$ from Fourier transform of chosen kernel $K$: $p(w) = \int e^{-jw'\delta} K(\delta) d\Delta$.   // $p(w)$ gives $Normal(0, 2\gamma I_{\tilde{d}})$ when $K(\delta) = exp(-\gamma||\delta||^2)$;$\gamma = 0.4$
    b: Draw $D$ iid samples $b_1, ...b_D \in \mathbb{R}$ from $Uniform(0, 2\pi)$;
    c: Generate a randomized feature map $\tilde{\Phi}(X) = \sqrt{\frac{2}{D}}[cos(w_1^\top X + b_1), ..., cos(w_D^\top X + b_D)]^\top$;
    d: Run PCA on $\tilde{\Phi}(X)$ to generate map $\Phi(\cdot)$ by top $d$ leading eigenvectors $\xi_1, ..., \xi_d$: $\Phi(X) = [\xi_1^\top \tilde{\Phi}(X), ..., \xi_d^\top \tilde{\Phi}(X)]^\top$. Delete mean.
3: Estimate $\hat{P} = \Sigma_0^{-1/2} \left[\sum_{i=1}^N \frac{1}{|C_0^i|} \sum_{X_0 \in C_0^i} \sum_{X_1 \in C_1^i} \Phi(X_0)\Phi(X_1)^\top\right] \Sigma_1^{-1/2}$,
   where $\Sigma_0 = \left[\sum_{X_0 \in A_0} \Phi(X_0)\Phi(X_0)^\top\right] + \lambda_0 I$ and $\Sigma_1 = \left[\sum_{X_1 \in A_1} \Phi(X_1)\Phi(X_1)^\top\right] + \lambda_1 I$.
   // $\lambda_0 = 0.1$, $\lambda_1 = 0$
4: SVD $\hat{U}\hat{S}\hat{V} = \hat{P}$
5: Compute Cell2state features using top $r$ singular values $\hat{\mathbf{\Psi}}(x) = (\Phi(x)^\top \Sigma_0^{-1/2} \hat{U}\hat{S}_{[1,..,r]})$
Output: $x \mapsto \hat{\mathbf{\Psi}}(x)$
```

The dataset we used (Biddy et al. (2018)) measured reprogramming activities of mouse embryonic fibroblasts to induced endoderm progenitors, containing 48,515 single-cell gene-expression profiles from day 0 to day 28, out of which 17,803 cells has transition information visualized on tSNE coordinates given by the original paper and colored by time points (**Figure 3(a)**). In particular, we input 1,997 day-12 ancestor cells and 3,509 day-21 descendent cells with 165,716 transition pairs in-between, colored on the same tSNE coordinates with downsampled transitions in **Figure 3(b)**; these single-cell expression profiles measure over 28,001 genes. We choose the input K to be the composition of the principal component map and Gaussian kernel with rank $r = 8$. The parameter values producing cell2state features with low inertia, distinct descendants structure, and good prediction performance are preferred. The experiment results demonstrated in **Figure 3(c-h)** of Section 4 are associated with parameters $\tilde{\lambda} = 0$, $\gamma = 50$, $D = 2000$, $d = 900$, $\lambda_0 = 0.1$, $\lambda_1 = 0$.

Data preprocessing. Our raw data is a collection of barcoded single-cell gene-expression profiles, with expression measurements of over 20000 genes. We use a standard single-cell data preprocessing pipeline, Seurat workflow(Butler et al. (2018); Stuart et al. (2019)). Firstly, we keep only genes expressed as non-zero in at least one cell and only cells expressed as non-zero in more than 200 genes. Then we filter out low-quality cells according to commonly used quality control metrics, such as the percentage of mitochondrial gene counts, and further log-transform the counts. Finally, we compare the barcode associated with each cell and arrange the cells into clones of transition pairs.

Predictions and Data splits. When testing the cell2state features of prediction of descendant activities, we use the ridge-regularized linear regressor/classifier from the 'sklearn' package. For computing out-of-sample errors, we repeated a random half-half data split 10 times and reported results with 95% confidence intervals.

Marker gene identification. Differential gene-expression analysis is used to find marker genes in each clusters. Marker genes are those genes that have significantly different levels of expressions in the cluster and could distinguish it from others. Specifically, we input the identity file of ancestor cells into R and run DESeq2(Love et al. (2014)) by the 'FindMarkers' function of Seurat(Butler et al. (2018); Stuart et al. (2019)) with parameter 'min.pct' = 0.25 and 'logfc.threshold' = 0.25.

Simulation experiment details. In Section 5, we highlighted a simulation setting where we take gene-expression data from Weinreb et al. (2020) and generate nonlinear transitions. In this section of the appendix, we provide further details as to how we go about our simulation experiments. First, we reduce the raw gene expression data to 50 dimensions via PCA as a preprocessing step. We then compare various embedding methods by providing these 50 PCs as input:

- **Linear**: Keep top 50 principal components of PCA.
- **Cell2state**: Run *cell2state* algorithm to get top 8 features.
- **Cell2state KPCA**: Compute the exact Gaussian kernel(not RFFs) in *cell2state* to get top 8 features.
- **Neural network (NN)**: A multilayer perceptron (MLP) with Swish activations (Ramachandran et al., 2017) and cross entropy loss.

With the exception of **NN**, all other embedding methods were trained with logistic regression. We train **NN** with the ADAM optimizer (Kingma & Ba, 2015) for 400 epochs and batch size 128.

We also conduct a grid-search over hyperparameters/architecture settings: $\lambda \in \{$1e-4, 1e-3, 1e-2$\}$, layers $\in \{$[128, 64], [128, 64, 32], [256, 128], [256, 128, 64]$\}$, initial learning rate $\in \{$1e-2, 5e-3, 1e-3, 5e-4, 1e-4$\}$. We use 5-fold cross validation to train all models. Furthermore, in Table 1, we see that there is very little variation of validation accuracy across different architectural settings as long as we choose the best hyperparameters for each architecture.

Table 1: Grid search results over the various hyperparameter/architecture settings in the stochastic setting. The third column represents the best validation accuracy among the various values of λ and initial learning rates for a given architecture. The second column is the training accuracy associated with the value of λ and initial learning rate that gave the best validation accuracy.

Hidden layers	Training Acc. (best val)	Validation Acc. (best)
[64, 32]	60.3 ± 0.3	58.9 ± 2.3
[64, 32, 16]	60.1 ± 0.3	58.9 ± 2.1
[128, 64]	59.6 ± 0.5	58.7 ± 2.6
[128, 64, 32]	60.3 ± 0.4	58.8 ± 2.6
[256, 128]	59.7 ± 0.5	59.0 ± 2.7
[256, 128, 64]	60.2 ± 0.5	59.0 ± 2.9

C Proof of Theorems

C.1 Assumptions

We first restate the assumptions needed for proofs of Theorems 1 and 2.

- $\|\phi(x)\|^2 \le C, \forall x$;
- The samples X_i are generated i.i.d. from some unknown distribution;
- In the encoding map analysis (Theorem 2), we assume there exists a lossless encoding map $\Omega^* : X \mapsto [k]$, i.e., $I(\Omega^*(X); Y) = I(X; Y)$. It means that
$$p(\cdot|x) = p(\cdot|x'), \qquad \forall x, x' \quad s.t. \quad \Omega^*(x) = \Omega^*(x').$$
Assume w.l.o.g that $r = \mathrm{rank}(\Phi^*)$ is no less than k. In this case, we may ensure $\Pi_{\mathcal{H}} p(\cdot|x) \neq \Pi_{\mathcal{H}} p(\cdot|x')$ when $\Omega^*(x) \neq \Omega^*(x')$. Otherwise we get $|\Pi_{\mathcal{H}} p(\cdot|\mathcal{X})| < k$, which implies a contradiction that $\mathrm{rank}(\Phi^*) = \mathrm{rank}(\boldsymbol{P}\mathcal{H}) < k$. Then we are able to recover the block structure Ω^* via estimating Φ^* by $\hat{\boldsymbol{\Psi}}$.

In subsequent proofs, we focus solely on the case that each state generates exactly one descendant. Our proof can be easily extended to the case of a branching process where a state has a random number of independent descendants. When a state x has N_x descendants $x' := \{x^{(1)'}, x^{(2)'}, \dots, x^{(N_x)'}\}$, the only difference is to let $\Phi(x') = \sum_{j=1}^{N_x} \Phi(x^{(j)'})$ and $p(\cdot|x) = \mathbb{E}[N_x] p'(\cdot|x)$ where $p'(\cdot|x)$ is the distribution of a single descendant. In this case, we would still be able to establish concentration bounds for $\hat{P}$ and the rest of the proof follows similarly.

C.2 Lemmas

First, we provide a matrix concentration bound for the estimated embedding matrix.

Lemma 1. *Let*
$$P = \Sigma_0^{-\frac{1}{2}} \mathbb{E}[\phi(X_0)\phi(X_0')^\top] \Sigma_1^{-\frac{1}{2}}.$$
For any $\varepsilon > 0$, we have
$$\mathbb{P}(\|P - \hat{P}\| > \varepsilon) \le 2d \exp\left(-\frac{3\gamma N \varepsilon^2}{2C^2 d(3+\varepsilon)}\right).$$
where $\gamma^{-1} = \max\{\|\Sigma_0^{-1}\|$.

Proof. We have
$$\hat{P} - P = \Sigma_0^{-\frac{1}{2}} \left[\frac{1}{N}\sum_{i=1}^{N} (\phi(X_i)\phi(X_i')^\top - \mathbb{E}[\phi(X_i)\phi(X_i')^\top])\right] \Sigma_1^{-\frac{1}{2}}.$$
Let $\boldsymbol{M}_i = \Sigma_0^{-\frac{1}{2}} \phi(X_i)\phi(X_i')^\top \Sigma_1^{-\frac{1}{2}} - \mathbb{E}[\Sigma_0^{-\frac{1}{2}} \phi(X_i)\phi(X_i')^\top \Sigma_1^{-\frac{1}{2}}]$, we have
$$\|\boldsymbol{M}_i\| = \|\Sigma_0^{-\frac{1}{2}} \phi(X_i)\phi(X_i')^\top \Sigma_1^{-\frac{1}{2}} - \mathbb{E}[\Sigma_0^{-\frac{1}{2}} \phi(X_i)\phi(X_i')^\top \Sigma_1^{-\frac{1}{2}}]\| \le \frac{2C}{\gamma},$$
and
$$\begin{aligned}
\|\mathbb{E}[\boldsymbol{M}_i^\top \boldsymbol{M}_i]\| =& \|\Sigma_1^{-\frac{1}{2}} \mathbb{E}[(\phi(X_i)\phi(X_i')^\top - \mathbb{E}[\phi(X_i)\phi(X_i')]^\top)^\top \Sigma_0^{-1} (\phi(X_i)\phi(X_i')^\top - \mathbb{E}[\phi(X_i)\phi(X_i')^\top]) \Sigma_1^{-\frac{1}{2}}]\| \\
\le& \|\mathbb{E}[\Sigma_1^{-\frac{1}{2}} (\phi(X_i)\phi(X_i')^\top)^\top \Sigma_0^{-1} \phi(X_i)\phi(X_i')^\top \Sigma_1^{-\frac{1}{2}}]\| \\
=& \|\mathbb{E}[[\phi(X_i)^\top \Sigma_0^{-1} \phi(X_i)] \Sigma_1^{-\frac{1}{2}} \phi(X_i')\phi(X_i')^\top \Sigma_1^{-\frac{1}{2}}]\| \\
\le& \frac{C}{\gamma}, \\
\|\mathbb{E}[\boldsymbol{M}_i \boldsymbol{M}_i^\top]\| =& \|\mathbb{E}[\Sigma_0^{-\frac{1}{2}} (\phi(X_i)\phi(X_i')^\top - \mathbb{E}[\phi(X_i)\phi(X_i')^\top]) \Sigma_1^{-1} (\phi(X_i)\phi(X_i')^\top - \mathbb{E}[\phi(X_i)\phi(X_i')^\top])^\top \Sigma_0^{-\frac{1}{2}}]\| \\
\le& \|\mathbb{E}[\Sigma_0^{-\frac{1}{2}} \phi(X_i)\phi(X_i')^\top \Sigma_1^{-1} (\phi(X_i)\phi(X_i')^\top)^\top \Sigma_0^{-\frac{1}{2}}]\| \\
=& \|\mathbb{E}[\phi(X_i')^\top \Sigma_1^{-1} \phi(X_i') \Sigma_0^{-\frac{1}{2}} \phi(X_i')\phi(X_i)^\top \Sigma_0^{-\frac{1}{2}}]\| \\
\le& \frac{C}{\gamma}.
\end{aligned}$$

By matrix Bernstein's inequality, we have

$$\mathbb{P}(\|P-\hat{P}\| > \varepsilon) \le 2d \exp\left(-\frac{3\gamma N \varepsilon^2}{2C(3+\varepsilon)}\right).$$

□

Next we show that the estimated embedding map preserves the correct feature space with high probability.

Lemma 2. *Let $\hat{\mathbf{\Psi}}$ be the estimated mapping, we have*

$$\mathbb{P}(\inf_{\boldsymbol{O}:\boldsymbol{O}^\top \boldsymbol{O}=I} \sup_{x\in\mathcal{X}} \|\boldsymbol{O}\hat{\mathbf{\Psi}}(x) - \mathbf{\Psi}(x)\| > \varepsilon) \le 2d\exp\left(-\frac{3\gamma^2 N\varepsilon^2}{8C^{\frac{3}{2}}\kappa(12\sqrt{C}\kappa+\sqrt{\gamma}\varepsilon)}\right),$$

where $\kappa = \frac{\|P\|}{\sigma_r(P)}$.

Proof. Suppose the SVD of P is $P = U_r\Lambda_r V_r^\top$. Let $\tilde{P} = \hat{U}_r\hat{\Lambda}_r\hat{V}_r^\top$. Let $\mathbf{\Psi}(x) = (U_r\Lambda_r)^\top \Sigma_0^{-\frac{1}{2}}\phi(x)$. Let $\boldsymbol{O}$ be an arbitrary orthonormal matrix. For any $x\in\mathcal{X}$, we have

$$\begin{aligned}\|\mathbf{\Psi}(x) - \boldsymbol{O}\hat{\mathbf{\Psi}}(x)\| &= \|(U_r\Lambda_r)^\top\Sigma_0^{-\frac{1}{2}}\phi(x) - \boldsymbol{O}(\hat{U}_r\hat{\Lambda}_r)^\top\Sigma_0^{-\frac{1}{2}}\phi(x)\| \\ &\le \frac{\sqrt{C}}{\sqrt{\gamma}}\|(U_r\Lambda_r)^\top - \boldsymbol{O}(\hat{U}_r\hat{\Lambda}_r)^\top\| \\ &= \frac{\sqrt{C}}{\sqrt{\gamma}}\|(PV_r)^\top - \boldsymbol{O}(\tilde{P}\hat{V}_r)^\top\| \\ &= \frac{\sqrt{C}}{\sqrt{\gamma}}\|(PV_r)^\top - \boldsymbol{O}(P'\hat{V}_r)^\top + \boldsymbol{O}(P\hat{V}_r)^\top - \boldsymbol{O}(\tilde{P}\hat{V}_r)^\top\| \\ &\le \frac{\sqrt{C}}{\sqrt{\gamma}}\left(\|P\|\|V_r - \hat{V}_r\boldsymbol{O}^\top\| + \|P-\tilde{P}\|\right).\end{aligned}$$

Taking infinum over $\boldsymbol{O}$, we get

$$\begin{aligned}\inf_{\boldsymbol{O}}\|\mathbf{\Psi}(x) - \boldsymbol{O}\hat{\mathbf{\Psi}}(x)\| &\le \inf_{\boldsymbol{O}}\frac{\sqrt{C}}{\sqrt{\gamma}}\left(\|P\|\|V_r - \hat{V}_r\boldsymbol{O}^\top\| + \|P-\tilde{P}\|\right) \\ &\le \frac{\sqrt{C}}{\sqrt{\gamma}}\left(\sqrt{2}\|P\|\|\sin\Theta(V_r,\hat{V}_r)\| + \|P-\tilde{P}\|\right) \\ &\le \frac{\sqrt{C}}{\sqrt{\gamma}}\left(\sqrt{2}\|P\|\frac{\|P-\hat{P}\|}{\sigma_r(P)} + \|P-\tilde{P}\|\right).\end{aligned}$$

By Weyl's inequality, we have

$$\|P-\tilde{P}\| \le \|P-\hat{P}\| + \|\hat{P}-\tilde{P}\| = \|P-\hat{P}\| + \sigma_{r+1}(P) \le 2\|P-\hat{P}\|.$$

In conclusion, we get

$$\inf_{\boldsymbol{O}}\|\mathbf{\Psi}(x) - \boldsymbol{O}\hat{\mathbf{\Psi}}(x)\| \le \frac{\sqrt{C}}{\sqrt{\gamma}}\left(\frac{\sqrt{2}\|P\|}{\sigma_r(P)}+2\right)\|P-\hat{P}\|, \forall x,$$

which implies

$$\inf_{\boldsymbol{O}}\sup_{x\in\mathcal{X}}\|\mathbf{\Psi}(x) - \boldsymbol{O}\hat{\mathbf{\Psi}}(x)\| \le \frac{2\sqrt{C}}{\sqrt{\gamma}}\left(\frac{\|P\|}{\sigma_r(P)}+1\right)\|P-\hat{P}\|.$$

According to the result of Lemma 1, we get

$$\begin{aligned}\mathbb{P}\left(\inf_{\boldsymbol{O}}\sup_{x\in\mathcal{X}}\|\boldsymbol{\Psi}(x)-\boldsymbol{O}\hat{\boldsymbol{\Psi}}(x)\|>\varepsilon\right)&\leq\mathbb{P}\left(\|P-\hat{P}\|>\frac{\sqrt{\gamma}\varepsilon}{2\sqrt{C}(1+\kappa)}\right)\\&\leq 2d\exp\left(-\frac{3\gamma^2N\varepsilon^2}{4C^{\frac{3}{2}}(1+\kappa)(6\sqrt{C}(1+\kappa)+\sqrt{\gamma}\varepsilon)}\right)\\&\leq 2d\exp\left(-\frac{3\gamma^2N\varepsilon^2}{8C^{\frac{3}{2}}\kappa(12\sqrt{C}\kappa+\sqrt{\gamma}\varepsilon)}\right).\end{aligned}$$

□

C.3 Proof of Theorem 1

Proof. For any orthonormal matrix $\boldsymbol{O}$, we have

$$\begin{aligned}\|\boldsymbol{\Psi}(x)-\boldsymbol{\Psi}(y)\|&=\|\boldsymbol{\Psi}(x)-\boldsymbol{O}\hat{\boldsymbol{\Psi}}(x)+\boldsymbol{O}\hat{\boldsymbol{\Psi}}(x)-\boldsymbol{O}\hat{\boldsymbol{\Psi}}(y)+\boldsymbol{O}\hat{\boldsymbol{\Psi}}(y)-\hat{\boldsymbol{\Psi}}(y)\|\\&\leq\|\boldsymbol{\Psi}(x)-\boldsymbol{O}\hat{\boldsymbol{\Psi}}(x)\|+\|\boldsymbol{O}\hat{\boldsymbol{\Psi}}(x)-\boldsymbol{O}\hat{\boldsymbol{\Psi}}(y)\|+\|\boldsymbol{O}\hat{\boldsymbol{\Psi}}(y)-\hat{\boldsymbol{\Psi}}(y)\|\\&=\|\boldsymbol{\Psi}(x)-\boldsymbol{O}\hat{\boldsymbol{\Psi}}(x)\|+\|\hat{\boldsymbol{\Psi}}(x)-\hat{\boldsymbol{\Psi}}(y)\|+\|\boldsymbol{O}\hat{\boldsymbol{\Psi}}(y)-\hat{\boldsymbol{\Psi}}(y)\|.\end{aligned}$$

Similarly we have

$$\|\hat{\boldsymbol{\Psi}}(x)-\hat{\boldsymbol{\Psi}}(y)\|\leq\|\boldsymbol{\Psi}(x)-\boldsymbol{O}\hat{\boldsymbol{\Psi}}(x)\|+\|\boldsymbol{\Psi}(x)-\boldsymbol{\Psi}(y)\|+\|\boldsymbol{O}\hat{\boldsymbol{\Psi}}(y)-\hat{\boldsymbol{\Psi}}(y)\|.$$

Therefore, we have

$$|\|\boldsymbol{\Psi}(x)-\boldsymbol{\Psi}(y)\|-\|\hat{\boldsymbol{\Psi}}(x)-\hat{\boldsymbol{\Psi}}(y)\||\leq 2\sup_{x\in\mathcal{X}}\|\boldsymbol{\Psi}(x)-\boldsymbol{O}\hat{\boldsymbol{\Psi}}(x)\|.$$

Taking infimum over $\boldsymbol{O}$, and notice that according to the result of Lemma 2, we have w.p. $1-q$,

$$\inf_{\boldsymbol{O}}\sup_{x\in\mathcal{X}}\|\boldsymbol{\Psi}(x)-\boldsymbol{O}\hat{\boldsymbol{\Psi}}(x)\|\leq\frac{8C\kappa}{\gamma}\sqrt{\frac{\log\frac{2d}{q}}{N}}+\frac{16C^{\frac{3}{2}}\kappa}{\gamma^{\frac{3}{2}}}\frac{\log\frac{2d}{q}}{N}.$$

We get w.p. $1-q$,

$$|\|\boldsymbol{\Psi}(x)-\boldsymbol{\Psi}(y)\|-\|\hat{\boldsymbol{\Psi}}(x)-\hat{\boldsymbol{\Psi}}(y)\||\leq\frac{16C\kappa}{\gamma}\sqrt{\frac{\log\frac{2d}{q}}{N}}+\frac{32C^{\frac{3}{2}}\kappa}{\gamma^{\frac{3}{2}}}\frac{\log\frac{2d}{q}}{N}.$$

Now, define inner products

$$\langle f,g\rangle_{\mathcal{H}}=\left[\int_y f(y)\Sigma_1^{-\frac{1}{2}}\phi(y)dy\right]^\top\left[\int_y g(y)\Sigma_1^{-\frac{1}{2}}\phi(y)dy\right].$$

we have

$$\int_y\Sigma_1^{-\frac{1}{2}}\phi(y)[\Sigma_1^{\frac{1}{2}}C^{-1}\phi(y)]^\top dy=\int_y\Sigma_1^{-\frac{1}{2}}\phi(y)\phi(y)^\top C^{-1}\Sigma_1^{\frac{1}{2}}dy=I.$$

Therefore, $e(y)=\Sigma_1^{\frac{1}{2}}C^{-1}\phi(y)$ is a set of orthonormal basis w.r.t. $\langle\cdot,\cdot\rangle_{\mathcal{H}}$. And for any fixed x, we have

$$\Pi_{\mathcal{H}}(p(\cdot|x))=\left[\int p(y|x)(\Sigma_1^{-\frac{1}{2}}\phi(y))^\top dy\right]\Sigma_1^{\frac{1}{2}}C^{-1}\phi(\cdot).$$

According to assumption 1, we know for any function $f\in\mathcal{H}$, we have

$$\boldsymbol{P}f(x)=\int p(y|x)f(y)dy\in\mathcal{H}.$$

Therefore, we know that each entry of $\int p(y|x)(\Sigma_1^{-\frac{1}{2}}\phi(y))^\top dy$ is in $\mathcal{H}$. Denote $\int p(y|x)(\Sigma_1^{-\frac{1}{2}}\phi(y))^\top dy = \phi(x)^\top V$, we have

$$V = \Sigma_0^{-1}\int \pi(x)\phi(x)p(y|x)(\Sigma_1^{-\frac{1}{2}}\phi(y))^\top dxdy.$$

Therefore, we get

$$\begin{aligned}\Pi_{\mathcal{H}}(p(\cdot|x)) &= (\Sigma_0^{-1}\Phi(x))^\top\left[\int \pi(x)\Phi(x)p(y|x)(\Sigma_1^{-\frac{1}{2}}\Phi(y))^\top dxdy\right]\Sigma_1^{\frac{1}{2}}C^{-1}\Phi(\cdot)\\ &= (\Sigma_0^{-1}\Phi(x))^\top\left[\Sigma_0^{-\frac{1}{2}}P\right]\Sigma_1^{\frac{1}{2}}C^{-1}\Phi(\cdot)\\ &= \boldsymbol{\Psi}(x)^\top\Sigma_1^{\frac{1}{2}}C^{-1}\phi(\cdot)\\ &= \boldsymbol{\Psi}(x)^\top e(\cdot).\end{aligned}$$

Therefore,

$$\|p(\cdot|x) - p(\cdot|y)\|_{\mathcal{H}} = \|\boldsymbol{\Psi}(x) - \boldsymbol{\Psi}(y)\|,$$

which implies

$$|\|p(\cdot|x) - p(\cdot|y)\|_{\mathcal{H}} - \|\hat{\boldsymbol{\Psi}}(x) - \hat{\boldsymbol{\Psi}}(y)\|| \le \frac{16C\kappa}{\gamma}\sqrt{\frac{\log\frac{2d}{q}}{N}} + \frac{32C^{\frac{3}{2}}\kappa}{\gamma^{\frac{3}{2}}}\frac{\log\frac{2d}{q}}{N}.$$

□

C.4 Proof of Theorem 2

Proof. According to the proof of Theorem 1, we know

$$\|p(\cdot|x) - p(\cdot|y)\|_{\mathcal{H}} = \|\boldsymbol{\Psi}(x) - \boldsymbol{\Psi}(y)\|.$$

Therefore, when $\Omega^*(x) = \Omega^*(x')$, we must have

$$0 = \|p(\cdot|x) - p(\cdot|x')\|_{\mathcal{H}} = \|\boldsymbol{\Psi}(x) - \boldsymbol{\Psi}(x')\|,$$

i.e., $\boldsymbol{\Psi}(x) = \boldsymbol{\Psi}(x')$. Therefore, $\boldsymbol{\Psi}$ is the linear mapping such that $|\boldsymbol{\Psi}(\mathcal{X})| \le k$. For any given $\hat{\psi}$, define $\boldsymbol{O}_{\hat{\psi}}$ as an orthonormal matrix such that

$$\sup_{x\in\mathcal{X}}\|\boldsymbol{O}_{\hat{\psi}}\hat{\psi}(x) - \boldsymbol{\Psi}(x)\| \le \inf_{\boldsymbol{O}:\boldsymbol{O}^\top\boldsymbol{O}=I}\sup_{x\in\mathcal{X}}\|\boldsymbol{O}\hat{\psi}(x) - \boldsymbol{\Psi}(x)\| + \frac{\delta}{8}.$$

According to Lemma 2, we have

$$\begin{aligned}\mathbb{P}(\exists x, \|\boldsymbol{O}_{\hat{\boldsymbol{\Psi}}}\hat{\Psi}(x) - \boldsymbol{\Psi}(x)\| > \delta/4) &\le \mathbb{P}(\inf_{\boldsymbol{O}:\boldsymbol{O}^\top\boldsymbol{O}=I}\sup_{x\in\mathcal{X}}\|\boldsymbol{O}\hat{\Psi}(x) - \boldsymbol{\Psi}(x)\| > \frac{\delta}{8})\\ &\le 2d\exp\left(-\frac{3\gamma^2N\delta^2}{8\times 64C^{\frac{3}{2}}\kappa(12\sqrt{C}\kappa + \sqrt{\gamma}\delta/8)}\right).\end{aligned}$$

According to our setting, for any x_1, x_2 such that $\boldsymbol{\Psi}(x_1) = \boldsymbol{\Psi}(x_2) = \boldsymbol{\Psi}_i$, we have

$$p(\cdot|x_1) = p(\cdot|x_2) = p_i(\cdot).$$

Therefore, by definition, we have

$$\begin{aligned}I(X, X') &= \int p(y|x)\pi(x)\log\frac{p(y|x)}{p_Y(y)}dxdy\\ &= \sum_i\int p(y|x)\pi(x)\mathbf{1}_{\boldsymbol{\Psi}(x)=\boldsymbol{\Psi}_i}\log\frac{p(y|x)}{p_Y(y)}dxdy\\ &= \sum_i\int p_i(y)\mathbb{P}(\boldsymbol{\Psi}(X) = \boldsymbol{\Psi}_i)\log\frac{p_i(y)}{p_Y(y)}dy\\ &= \sum_i\sum_j\int p_i(y)\mathbb{P}(\boldsymbol{\Psi}(X) = \boldsymbol{\Psi}_i, \Omega_{\hat{\psi}}(X) = A_j)\log\frac{p_i(y)}{p_Y(y)}dy.\end{aligned}$$

On the other hand, we have

$$
\begin{aligned}
I(\Omega_{\hat{\psi}}(X), X') = \sum_j \int p(y|x)\pi(x)\mathbf{1}_{\Omega_{\hat{\psi}}(x)=A_j} \log \frac{p(y|\Omega_{\hat{\psi}}(x)=A_j)}{p_Y(y)} dxdy \\
= & \sum_i \sum_j \int p(y|x)\pi(x)\mathbf{1}_{\mathbf{\Psi}(x)=\mathbf{\Psi}_i, \Omega_{\hat{\psi}}(x)=A_j} \log \frac{p(y|\Omega_{\hat{\psi}}(x)=A_j)}{p_Y(y)} dxdy \\
= & \sum_i \sum_j \int p(y|x)\pi(x)\mathbf{1}_{\mathbf{\Psi}(x)=\mathbf{\Psi}_i, \Omega_{\hat{\psi}}(x)=A_j} \log \frac{\int p(y|x)\pi(x)\mathbf{1}_{\Omega_{\hat{\psi}}(x)=A_j} dx}{p_Y(y)\mathbb{P}(\Omega_{\hat{\psi}}(X)=A_j)} dxdy \\
= & \sum_i \sum_j \int p(y|x)\pi(x)\mathbf{1}_{\mathbf{\Psi}(x)=\mathbf{\Psi}_i, \Omega_{\hat{\psi}}(x)=A_j} \log \frac{\int p(y|x)\pi(x)\mathbf{1}_{\Omega_{\hat{\psi}}(x)=A_j} dx}{\int p(y|x)\pi(x)\mathbf{1}_{\mathbf{\Psi}(x)=\mathbf{\Psi}_i, \Omega_{\hat{\psi}}(x)=A_j} dx} dxdy \\
+ & \sum_i \sum_j \int p(y|x)\pi(x)\mathbf{1}_{\mathbf{\Psi}(x)=\mathbf{\Psi}_i, \Omega_{\hat{\psi}}(x)=A_j} \log \frac{\int p(y|x)\pi(x)\mathbf{1}_{\mathbf{\Psi}(x)=\mathbf{\Psi}_i, \Omega_{\hat{\psi}}(x)=A_j} dx}{p_Y(y)\mathbb{P}(\Omega_{\hat{\psi}}(X)=A_j)} dxdy \\
= & \sum_j \int p(y|x)\pi(x)\mathbf{1}_{\Omega_{\hat{\psi}}(x)=A_j} \sum_i \mathbb{P}(\mathbf{\Psi}(X)=\mathbf{\Psi}_i|Y=y, \hat{\Omega}_{\hat{\psi}}(X)=A_j) \log \frac{1}{\mathbb{P}(\mathbf{\Psi}(X)=\mathbf{\Psi}_i|Y=y, \hat{\Omega}_{\hat{\psi}}(X)=A_j)} dxdy \\
+ & \sum_i \sum_j \int p(y|x)\pi(x)\mathbf{1}_{\mathbf{\Psi}(x)=\mathbf{\Psi}_i, \Omega_{\hat{\psi}}(x)=A_j} \log \frac{\int p(y|x)\pi(x)\mathbf{1}_{\mathbf{\Psi}(x)=\mathbf{\Psi}_i, \Omega_{\hat{\psi}}(x)=A_j} dx}{p_Y(y)\mathbb{P}(\Omega_{\hat{\psi}}(X)=A_j)} dxdy \\
\geq & \sum_i \sum_j \int p(y|x)\pi(x)\mathbf{1}_{\mathbf{\Psi}(x)=\mathbf{\Psi}_i, \Omega_{\hat{\psi}}(x)=A_j} \log \frac{\int p(y|x)\pi(x)\mathbf{1}_{\mathbf{\Psi}(x)=\mathbf{\Psi}_i, \Omega_{\hat{\psi}}(x)=A_j} dx}{p_Y(y)\mathbb{P}(\Omega_{\hat{\psi}}(X)=A_j)} dxdy.
\end{aligned}
$$

And we have

$$
\begin{aligned}
& \sum_i \sum_j \int p(y|x)\pi(x)\mathbf{1}_{\mathbf{\Psi}(x)=\mathbf{\Psi}_i, \Omega_{\hat{\psi}}(x)=A_j} \log \frac{\int p(y|x)\pi(x)\mathbf{1}_{\mathbf{\Psi}(x)=\mathbf{\Psi}_i, \Omega_{\hat{\psi}}(x)=A_j} dx}{p_Y(y)\mathbb{P}(\Omega_{\hat{\psi}}(X)=A_j)} dxdy \\
= & \sum_i \sum_j \int p_i(y)\pi(x)\mathbf{1}_{\mathbf{\Psi}(x)=\mathbf{\Psi}_i, \Omega_{\hat{\psi}}(x)=A_j} \log \frac{\int p_i(y)\pi(x)\mathbf{1}_{\mathbf{\Psi}(x)=\mathbf{\Psi}_i, \Omega_{\hat{\psi}}(x)=A_j} dx}{p_Y(y)\mathbb{P}(\Omega_{\hat{\psi}}(X)=A_j)} dxdy \\
= & \sum_i \sum_j \int p_i(y)\mathbb{P}(\mathbf{\Psi}(x)=\mathbf{\Psi}_i, \Omega_{\hat{\psi}}(x)=A_j) \log \frac{p_i(y)\mathbb{P}(\mathbf{\Psi}(x)=\mathbf{\Psi}_i, \Omega_{\hat{\psi}}(x)=A_j)}{p_Y(y)\mathbb{P}(\Omega_{\hat{\psi}}(X)=A_j)} dy.
\end{aligned}
$$

which implies

$$
I(\Omega_{\hat{\psi}}(X), X') \geq \sum_i \sum_j \int p_i(y)\mathbb{P}(\mathbf{\Psi}(x)=\mathbf{\Psi}_i, \Omega_{\hat{\psi}}(x)=A_j) \log \frac{p_i(y)\mathbb{P}(\mathbf{\Psi}(x)=\mathbf{\Psi}_i, \Omega_{\hat{\psi}}(x)=A_j)}{p_Y(y)\mathbb{P}(\Omega_{\hat{\psi}}(X)=A_j)} dy.
$$

Combining the results so far, we get

$$
\begin{aligned}
& I(X, X') - I(\hat{\Omega}_{\hat{\psi}}(X), X') \\
\leq & \sum_i \sum_j \int p_i(y)\mathbb{P}(\mathbf{\Psi}(X)=\mathbf{\Psi}_i, \Omega_{\hat{\psi}}(X)=A_j) \log \frac{p_i(y)}{p_Y(y)} dy \\
& - \sum_i \sum_j \int p_i(y)\mathbb{P}(\mathbf{\Psi}(x)=\mathbf{\Psi}_i, \Omega_{\hat{\psi}}(x)=A_j) \log \frac{p_i(y)\mathbb{P}(\mathbf{\Psi}(x)=\mathbf{\Psi}_i, \Omega_{\hat{\psi}}(x)=A_j)}{p_Y(y)\mathbb{P}(\Omega_{\hat{\psi}}(X)=A_j)} dy \\
= & \sum_i \sum_j \int p_i(y)\mathbb{P}(\mathbf{\Psi}(X)=\mathbf{\Psi}_i, \Omega_{\hat{\psi}}(X)=A_j) \log \frac{\mathbb{P}(\Omega_{\hat{\psi}}(X)=A_j)}{\mathbb{P}(\mathbf{\Psi}(x)=\mathbf{\Psi}_i, \Omega_{\hat{\psi}}(x)=A_j)} dy \\
= & \sum_i \sum_j \mathbb{P}(\mathbf{\Psi}(X)=\mathbf{\Psi}_i, \hat{\Omega}_{\hat{\psi}}(X)=A_j) \log \frac{\mathbb{P}(\hat{\Omega}_{\hat{\psi}}(X)=A_j)}{\mathbb{P}(\mathbf{\Psi}(X)=\mathbf{\Psi}_i, \hat{\Omega}_{\hat{\psi}}(X)=A_j)}.
\end{aligned}
$$

Taking expectation on both sides, we get

$$\begin{aligned}\mathbb{E}_{\hat{\boldsymbol{\Psi}}}[I(Y,\hat{\Omega}_{\hat{\boldsymbol{\Psi}}}(X))] \geq & I(Y,X) \\ & - \int \left(\sum_i \sum_j \mathbb{P}(\boldsymbol{\Psi}(X)=\boldsymbol{\Psi}_i, \hat{\Omega}_{\hat{\psi}}(X)=A_j) \log \frac{\mathbb{P}(\hat{\Omega}_{\hat{\psi}}(X)=A_j)}{\mathbb{P}(\boldsymbol{\Psi}(X)=\boldsymbol{\Psi}_i, \hat{\Omega}_{\hat{\psi}}(X)=A_j)} \right) d\mathbb{P}(\hat{\psi}).\end{aligned}$$

Note that

$$\begin{aligned}&\int \left[\sum_i \sum_j \mathbb{P}(\boldsymbol{\Psi}(X)=\boldsymbol{\Psi}_i, \hat{\Omega}_{\hat{\psi}}(X)=A_j) \log \frac{\mathbb{P}(\hat{\Omega}_{\hat{\psi}}(X)=A_j)}{\mathbb{P}(\boldsymbol{\Psi}(X)=\boldsymbol{\Psi}_i, \hat{\Omega}_{\hat{\psi}}(X)=A_j)} d\mathbb{P}(\hat{\psi}) \right] \\
=&\int_{\hat{\psi}:\forall x, \|\boldsymbol{O}_{\hat{\psi}}\hat{\psi}(x)-\boldsymbol{\Psi}(x)\| \leq \frac{\delta}{4}} \left[\sum_i \sum_j \mathbb{P}(\boldsymbol{\Psi}(X)=\boldsymbol{\Psi}_i, \hat{\Omega}_{\hat{\psi}}(X)=A_j) \log \frac{\mathbb{P}(\hat{\Omega}_{\hat{\psi}}(X)=A_j)}{\mathbb{P}(\boldsymbol{\Psi}(X)=\boldsymbol{\Psi}_i, \hat{\Omega}_{\hat{\psi}}(X)=A_j)} \right] d\mathbb{P}(\hat{\psi}) \\
&+ \int_{\hat{\psi}:\exists x, \|\boldsymbol{O}_{\hat{\psi}}\hat{\psi}(x)-\boldsymbol{\Psi}(x)\| > \frac{\delta}{4}} \left[\sum_i \sum_j \mathbb{P}(\boldsymbol{\Psi}(X)=\boldsymbol{\Psi}_i, \hat{\Omega}_{\hat{\psi}}(X)=A_j) \log \frac{\mathbb{P}(\hat{\Omega}_{\hat{\psi}}(X)=A_j)}{\mathbb{P}(\boldsymbol{\Psi}(X)=\boldsymbol{\Psi}_i, \hat{\Omega}_{\hat{\psi}}(X)=A_j)} \right] d\mathbb{P}(\hat{\psi}) \\
\leq &\int_{\hat{\psi}:\forall x, \|\boldsymbol{O}_{\hat{\psi}}\hat{\psi}(x)-\boldsymbol{\Psi}(x)\| \leq \frac{\delta}{4}} \left[\sum_i \sum_j \mathbb{P}(\boldsymbol{\Psi}(X)=\boldsymbol{\Psi}_i, \hat{\Omega}_{\hat{\psi}}(X)=A_j) \log \frac{\mathbb{P}(\hat{\Omega}_{\hat{\psi}}(X)=A_j)}{\mathbb{P}(\boldsymbol{\Psi}(X)=\boldsymbol{\Psi}_i, \hat{\Omega}_{\hat{\psi}}(X)=A_j)} \right] d\mathbb{P}(\hat{\psi}) \\
&+ \int_{\hat{\psi}:\exists x, \|\boldsymbol{O}_{\hat{\psi}}\hat{\psi}(x)-\boldsymbol{\Psi}(x)\| > \frac{\delta}{4}} \log k \; d\mathbb{P}(\hat{\psi}) \\
\leq &\int_{\hat{\psi}:\forall x, \|\boldsymbol{O}_{\hat{\psi}}\hat{\psi}(x)-\boldsymbol{\Psi}(x)\| \leq \frac{\delta}{4}} \left[\sum_i \sum_j \mathbb{P}(\boldsymbol{\Psi}(X)=\boldsymbol{\Psi}_i, \hat{\Omega}_{\hat{\psi}}(X)=A_j) \log \frac{\mathbb{P}(\hat{\Omega}_{\hat{\psi}}(X)=A_j)}{\mathbb{P}(\boldsymbol{\Psi}(X)=\boldsymbol{\Psi}_i, \hat{\Omega}_{\hat{\psi}}(X)=A_j)} \right] d\mathbb{P}(\hat{\psi}) \\
&+ 2d \exp\left(-\frac{3\gamma^2 N \delta^2}{8 \times 64 C^{\frac{3}{2}} \kappa (12\sqrt{C}\kappa + \sqrt{\gamma}\delta/8)} \right) \log k.\end{aligned}$$

It remains to consider the first term. Note that

$$\begin{aligned}&\sum_i \sum_j \mathbb{P}(\boldsymbol{\Psi}(X)=\boldsymbol{\Psi}_i, \hat{\Omega}_{\hat{\psi}}(X)=A_j) \log \frac{\mathbb{P}(\hat{\Omega}_{\hat{\psi}}(X)=A_j)}{\mathbb{P}(\boldsymbol{\Psi}(X)=\boldsymbol{\Psi}_i, \hat{\Omega}_{\hat{\psi}}(X)=A_j)} \\
=&\sum_i \sum_{j:\exists z \in A_j, \|\boldsymbol{O}_{\hat{\psi}}z-\boldsymbol{\Psi}_i\| \leq \frac{\delta}{4}} \mathbb{P}(\boldsymbol{\Psi}(X)=\boldsymbol{\Psi}_i, \hat{\Omega}_{\hat{\psi}}(X)=A_j) \log \frac{\mathbb{P}(\hat{\Omega}_{\hat{\psi}}(X)=A_j)}{\mathbb{P}(\boldsymbol{\Psi}(X)=\boldsymbol{\Psi}_i, \hat{\Omega}_{\hat{\psi}}(X)=A_j)} \\
&+\sum_i \sum_{j:\forall z \in A_j, \|\boldsymbol{O}_{\hat{\psi}}z-\boldsymbol{\Psi}_i\| > \frac{\delta}{4}} \mathbb{P}(\boldsymbol{\Psi}(X)=\boldsymbol{\Psi}_i, \hat{\Omega}_{\hat{\psi}}(X)=A_j) \log \frac{\mathbb{P}(\hat{\Omega}_{\hat{\psi}}(X)=A_j)}{\mathbb{P}(\boldsymbol{\Psi}(X)=\boldsymbol{\Psi}_i, \hat{\Omega}_{\hat{\psi}}(X)=A_j)}.\end{aligned}$$

Therefore, when $\hat{\psi}$ satisfies $\forall x, \|\boldsymbol{O}_{\hat{\psi}}\hat{\psi}(x)-\boldsymbol{\Psi}(x)\| \leq \frac{\delta}{4}$, then for any j such that $\forall z \in A_j, \|\boldsymbol{O}_{\hat{\psi}}z-\boldsymbol{\Psi}_i\| > \frac{\delta}{4}$, we have

$$\mathbb{P}(\boldsymbol{\Psi}(X)=\boldsymbol{\Psi}_i, \hat{\Omega}_{\hat{\psi}}(X)=A_j) = \mathbb{P}(\boldsymbol{\Psi}(X)=\boldsymbol{\Psi}_i, \hat{\psi}(X)\in A_j) \leq \mathbb{P}(\|\boldsymbol{\Psi}(X)-\boldsymbol{O}_{\hat{\psi}}\hat{\psi}(X)\| > \frac{\delta}{4}) = 0.$$

And for any j such that $\exists z \in A_j, \|\boldsymbol{O}_{\hat{\psi}}z-\boldsymbol{\Psi}_i\| \leq \frac{\delta}{4}$ and any $l \neq i$, since we have $\delta \leq \min_{l\neq i}\|\boldsymbol{\Psi}_i-\boldsymbol{\Psi}_l\|$, we get $\forall z' \in A_j$,

$$\begin{aligned}\|\boldsymbol{\Psi}_l-\boldsymbol{O}_{\hat{\psi}}z'\| = \|\boldsymbol{\Psi}_l-\boldsymbol{\Psi}_i+\boldsymbol{\Psi}_i-\boldsymbol{O}_{\hat{\psi}}z'\| &\geq \|\boldsymbol{\Psi}_l-\boldsymbol{\Psi}_i\| - \|\boldsymbol{\Psi}_i-\boldsymbol{O}_{\hat{\psi}}z'\| \\
&\geq \delta - \|\boldsymbol{\Psi}_i-\boldsymbol{O}_{\hat{\psi}}z+\boldsymbol{O}_{\hat{\psi}}z-\boldsymbol{O}_{\hat{\psi}}z'\| \\
&\geq \delta - \|\boldsymbol{\Psi}_i-\boldsymbol{O}_{\hat{\psi}}z\| - \|\boldsymbol{O}_{\hat{\psi}}z-\boldsymbol{O}_{\hat{\psi}}z'\| \\
&\geq \delta - \frac{\delta}{4} - \frac{\delta}{4} = \frac{\delta}{2}.\end{aligned}$$

Therefore,

$$\mathbb{P}(\boldsymbol{\Psi}(X) = \boldsymbol{\Psi}_l, \hat{\Omega}_{\hat{\psi}}(X) = A_j) = \mathbb{P}(\boldsymbol{\Psi}(X) = \boldsymbol{\Psi}_l, \hat{\psi}(X) \in A_j) \le \mathbb{P}(\|\boldsymbol{\Psi}(X) - \boldsymbol{O}_{\hat{\psi}}\hat{\psi}(X)\| > \frac{\delta}{2}) = 0, \forall l \neq i,$$

which implies

$$\begin{aligned}\mathbb{P}(\boldsymbol{\Psi}(X) = \boldsymbol{\Psi}_i, \hat{\Omega}_{\hat{\psi}}(X) = A_j) &= \mathbb{P}(\hat{\Omega}_{\hat{\psi}}(X) = A_j) - \sum_{l \neq i} \mathbb{P}(\boldsymbol{\Psi}(X) = \boldsymbol{\Psi}_l, \hat{\Omega}_{\hat{\psi}}(X) = A_j) \\ &= \mathbb{P}(\hat{\Omega}_{\hat{\psi}}(X) = A_j).\end{aligned}$$

Combining the results so far, we get

$$\begin{aligned}&\sum_i \sum_j \mathbb{P}(\boldsymbol{\Psi}(X) = \boldsymbol{\Psi}_i, \hat{\Omega}_{\hat{\psi}}(X) = A_j) \log \frac{\mathbb{P}(\hat{\Omega}_{\hat{\psi}}(X) = A_j)}{\mathbb{P}(\boldsymbol{\Psi}(X) = \boldsymbol{\Psi}_i, \hat{\Omega}_{\hat{\psi}}(X) = A_j)} \\ =&\sum_i \sum_{j: \exists z \in A_j, \|\boldsymbol{O}_{\hat{\psi}} z - \boldsymbol{\Psi}_i\| \le \frac{\delta}{4}} \mathbb{P}(\boldsymbol{\Psi}(X) = \boldsymbol{\Psi}_i, \hat{\Omega}_{\hat{\psi}}(X) = A_j) \log \frac{\mathbb{P}(\hat{\Omega}_{\hat{\psi}}(X) = A_j)}{\mathbb{P}(\boldsymbol{\Psi}(X) = \boldsymbol{\Psi}_i, \hat{\Omega}_{\hat{\psi}}(X) = A_j)} \\ &+ \sum_i \sum_{j: \forall z \in A_j, \|\boldsymbol{O}_{\hat{\psi}} z - \boldsymbol{\Psi}_i\| > \frac{\delta}{4}} \mathbb{P}(\boldsymbol{\Psi}(X) = \boldsymbol{\Psi}_i, \hat{\Omega}_{\hat{\psi}}(X) = A_j) \log \frac{\mathbb{P}(\hat{\Omega}_{\hat{\psi}}(X) = A_j)}{\mathbb{P}(\Psi(X) = \boldsymbol{\Psi}_i, \hat{\Omega}_{\hat{\psi}}(X) = A_j)} \\ =&\sum_i \sum_{j: \exists z \in A_j, \|\boldsymbol{O}_{\hat{\psi}} z - \boldsymbol{\Psi}_i\| \le \frac{\delta}{4}} \mathbb{P}(\hat{\Omega}_{\hat{\psi}}(X) = A_j) \log \frac{\mathbb{P}(\hat{\Omega}_{\hat{\psi}}(X) = A_j)}{\mathbb{P}(\hat{\Omega}_{\hat{\psi}}(X) = A_j)} + \sum_i \sum_{j: \forall z \in A_j, \|\boldsymbol{O}_{\hat{\psi}} z - \boldsymbol{\Psi}_i\| > \frac{\delta}{4}} 0 \\ =&0.\end{aligned}$$

which implies

$$\begin{aligned}&\int \left[\sum_i \sum_j \mathbb{P}(\boldsymbol{\Psi}(X) = \boldsymbol{\Psi}_i, \hat{\Omega}_{\hat{\psi}}(X) = A_j) \log \frac{\mathbb{P}(\hat{\Omega}_{\hat{\psi}}(X) = A_j)}{\mathbb{P}(\boldsymbol{\Psi}(X) = \boldsymbol{\Psi}_i, \hat{\Omega}_{\hat{\psi}}(X) = A_j)} d\mathbb{P}(\hat{\psi}) \right] \\ \le& 2d \exp\left(-\frac{3\gamma^2 N \delta^2}{8 \times 64 C^{\frac{3}{2}} \kappa (12\sqrt{C}\kappa + \sqrt{\gamma}\delta/8)} \right) \log k,\end{aligned}$$

i.e.,

$$\mathbb{E}_{\hat{\boldsymbol{\Psi}}}[I(\hat{\Omega}_{\hat{\boldsymbol{\Psi}}}(X), X')] \ge I(X, X') - 2d \exp\left(-\frac{3\gamma^2 N \delta^2}{8 \times 64 C^{\frac{3}{2}} \kappa (12\sqrt{C}\kappa + \sqrt{\gamma}\delta/8)} \right) \log k.$$

Finally, noticing that

$$\delta \le 2 \sup_x \|\Psi(x)\| \le 2\sqrt{C}\|P\|\gamma^{-\frac{1}{2}} \le 2\sqrt{C}\kappa\gamma^{-\frac{1}{2}},$$

we have finished the proof. □