Distribution-Conditioned Transport: Zero-Shot Transport from Any Distribution to Any Distribution

Anonymous Authors[1]

Abstract

1. Introduction

Learning transport maps between probability distributions is a central challenge across machine learning and the sciences. A wide variety of approaches, including diffusion and flow-based models, have addressed with great success the one-to-one transport problem: learning a map which pushes a source density P_0 to a target density P_1 (Goodfellow et al., 2014; Rezende & Mohamed, 2015; Genevay et al., 2018b; Liu et al., 2022; Lipman et al., 2023; Albergo et al., 2023b). However, a new class of problems is emerging where the goal is not only to model the evolution between a single pair of distributions, but rather to model dynamics in a way that can generalize across a broad range of source and target distributions.

A concrete motivating example of this shift can be seen in the study of cellular dynamics. Experimental techniques such as clonal lineage tracing have enabled the generation of datasets containing snapshots not just of a single cell population, but of thousands of distinct populations (clones) evolving in parallel (Biddy et al., 2018; Weinreb et al., 2020; Wagner & Klein, 2020). These datasets are sparse, in the sense that we do not observe the all time-marginals for all populations. For example, we may observe one population at both an initial time t_0 and a final time t_1, while others are observed only at either t_0 or t_1.

Two lines of literature have begun to approach this problem. Multimarginal stochastic interpolants offer an approach to learn dynamics between any pair of a fixed set of k distributions, solving a k-to-k transport problem (Albergo et al., 2023a), but cannot condition on a continuous space of distributions nor generalize to distributions unseen during training. In another direction, (Fishman et al., 2025) develop approaches to learn "autoencoders" on the space of distributions by simultaneously learning to embed distributions and sample from the distribution conditional on that embedding. Meta flow matching (Atanackovic et al., 2024) is an early approach that leverages a heuristic distribution encoder for a structured transport problem. It embeds the source distribution with a distribution encoder coupled with a flow matching transport map. While effective in settings where many source-target pairs are available, if we observed many pairs of clonal lineages at both t_0 and t_1, the MFM framework cannot ingest unpaired marginal distributions (e.g., cell populations observed at a single timepoint).

In this work we unify these perspectives. We first generalize and formalize the approach in (Atanackovic et al., 2024), demonstrating how distribution encoders developed in (Fishman et al., 2025) can be coupled with a broad class of transport models for source-conditioned transport. This immediately enables us to handle any-to-any transport by conditioning on both source and target distribution embeddings. This formulation enables us to generalize across distributions, predicting transport maps between population pairs unseen during training, as well as allowing us to make use of unstructured, partial observations (such as orphan marginals). We demonstrate the effectiveness of our approach first using a set of synthetic benchmark datasets, then on real-world applications ranging from image style transfer to learning cell population dynamics from lineage traced scRNA-seq data.

2. Methods

In many modern biological assays, we do not observe isolated datapoints in a vacuum. Instead, we repeatedly observe *sets* of measurements drawn from coherent biological entities that change in structured ways. Examples include (i) clonal lineages assayed at multiple timepoints in lineage tracing, (ii) cells from the same donor before and after a perturbation, and (iii) matched tissue microenvironments profiled under two experimental conditions. In many of these cases we are interested in a certain kind of counterfactual: what would these cells look like if they were in that state? This counterfactual problem is most naturally posed as a distributional transport problem, where we take

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

a set of cells in one state and move them to another. Recent years have seen the development of a large number of methods for distributional transport (Goodfellow et al., 2020; Genevay et al., 2018a; Lipman et al., 2022). These methods fundamentally transport one distribution to another; this process can be augmented by conditioning to transport one initial distribution to many different target states (modeling differentiation for example) or many initial states to a corresponding target (like modeling a first time step to a second time step across many cell types). But we do not always have high-quality representations of the distributions we want to transport between.

To motivate this setting we focus on lineage tracing, where we observe many sets of clones of the same cell across different timepoints. The goal is to predict cell fate: to take the initial distribution of a clone and push it forward in time. Here, all cells are usually of the same "type", so there is no natural way to represent the clone other than by the set of cells that define the clone. This requires modeling transport on the space of distributions. In this section we will outline tools for modeling on the space of distributions.

Formally, for an entity i (e.g. a clone), we observe a set of samples

$$S_i = \{x_{ij}\}_{j=1}^{m_i}, \quad x_{ij} \in \mathcal{X},$$

modeled as i.i.d. draws $x_{ij} \overset{\text{iid}}{\sim} P_i$ from some unknown distribution $P_i \in \mathcal{P}(\mathcal{X})$.

The first tool we will develop here is a distribution encoder, which enables principled training of representions of sets of points as vectors. Then we will show how these distribution encoders can be linked to downstream transport models.

2.1. Distribution encoders: representing an empirical population as a point in latent space

From each distribution we observe a finite sample set $S_i = \{x_{ij}\}_{j=1}^{m_i}$. We follow Fishman et al. (2025), defining a *distribution encoder*

$$\mathcal{E} : S_i \mapsto z_i \in \mathbb{R}^d,$$

which produces a fixed-dimensional embedding z_i intended to summarize the entire distribution P_i, rather than any particular cell. The key aim of distribution encoders is that z_i reflects only the underlying distributional signal and not sampling noise in S_i. To enforce this the encoder must be *distributionally invariant*. This imposes two invariances:

Permutation invariance. Reordering the samples in S_i does not change $\mathcal{E}(S_i)$.

Proportional invariance. Uniformly duplicating all samples in S_i does not change $\mathcal{E}(S_i)$.

When these hold, the encoder can only depend on the empirical measure

$$\widehat{P}_i \;=\; \frac{1}{m_i}\sum_{j=1}^{m_i} \delta_{x_{ij}}.$$

In particular, there exists a measurable functional ϕ such that

$$\mathcal{E}(S_i) \;=\; \phi\Big(\widehat{P}_i\Big). \tag{1}$$

We refer to $z_i = \phi(\widehat{P}_{i,m})$ as a *distribution embedding*. A key property of such encoders is that they admit a central limit theorem (CLT). Let

$$z_i^\star = \lim_{m_i \to \infty} \phi\Big(\widehat{P}_i\Big) = \phi(P_i)$$

denote the population-level embedding obtained in the infinite-sample limit. Under the invariances above and Hadamard differentiability of the pooling operator defining ϕ, we have

$$\sqrt{m_i}\,\big(\mathcal{E}(S_i) - z_i^\star\big) \;\xrightarrow{d}\; \mathcal{N}(0, \Sigma_i) \tag{2}$$

for a covariance Σ_i depending on P_i and the encoder $\mathcal{E}$.

For our purposes this CLT is the most important feature of distribution encoders because, as proved in (Fishman et al., 2025), it means that for any downstream loss we are interested in we can train models on moderate-sized sets at training time and compute gradients (in expectation) as though we were training using the true population embeddings z_i^*.

2.2. Supervised (one-to-one) transport: source-conditioning

A key setting for distributional transport is when we observe "pairs" of datasets. In our lineage tracing example this corresponds to observing many clones across two timepoints, where we want to learn to transport from timepoint one to timepoint two. Another core example is when we observe many cell types under the same perturbation condition, our goal is to transport the unperturbed cells into their perturbed condition. This transport setting is analogous to supervised learning: we observe many source sets of cells and their corresponding target distribution and we want to learn a one-to-one map from the source to the target. In these settings it turns out by conditioning the the source distribution we can learn well defined transport maps. We will call these source-conditioned transport models.

A number of recent works have developed particular source-conditioned models such as (Atanackovic et al., 2024; Klein et al., 2025) in the context of flow matching or (He et al., 2025) using a direct MMD objective. Our notion of "source-conditioned" models generalizes these architectures under

a common framework and clarifies the underlying assumptions under which they enable distributional transport.

Formally, we will assume throughout this section that we have a supervised set of paired distributions sampled from some meta distribution (e.g. the set of clones sampled from the distribution of clones in the initial cell type)

$$(P_{1_{\text{src}}}, P_{1_{\text{tgt}}}), \dots (P_{n_{\text{src}}}, P_{n_{\text{tgt}}}) \sim Q,$$

and we observe empirical sets

$$S_{i_{\text{src}}} = \{x_{i_{\text{src}}j}\}_{j=1}^{m_{i_{\text{src}}}} \sim P_{i_{\text{src}}} \text{ and } S_{i_{\text{tgt}}} = \left\{x_{i_{\text{tgt}}j}\right\}_{j=1}^{m_{i_{\text{tgt}}}} \sim P_{i_{\text{tgt}}}.$$

We encode the *source* only so $z_{i_{\text{src}}} = \mathcal{E}\left(S_{i_{\text{src}}}\right)$, where $\mathcal{E}$ is a distribution encoder as laid out in Sec. 2.1. We can then learn a source-conditioned transport map acting on individual samples,

$$\mathcal{T} : \mathcal{X} \times \mathbb{R}^d \to \mathcal{X}, \quad x_{i_{\text{src}}} \mapsto \widehat{x}_{i_{\text{tgt}}} = \mathcal{T}\big(x_{i_{\text{src}}} \mid z_{\text{src}}\big),$$

so that transported samples asymptotically follow the target distribution:

$$\mathcal{T}(S_{i_{\text{src}}} \mid \mathcal{E}(S_{i_{\text{src}}})) \xrightarrow[m_{i_{\text{src}}} \to \infty]{d} P_{i_{\text{tgt}}} \tag{3}$$

Here $\mathcal{T}$ can be any conditional distributional transport model, and under a supervised pairing policy, the correct destination $P_{i_{\text{tgt}}}$ is implied once the source is specified.

We jointly fit $\mathcal{E}$ and $\mathcal{T}$ using the native loss of the chosen transport mechanism, written abstractly as

$$\mathcal{L}_{sc} = \ell(S_{\text{tgt}},\ \mathcal{T}(S_{i_{\text{src}}} \mid \mathcal{E}(S_{i_{\text{src}}}))\,, \tag{4}$$

where ℓ may be a flow-matching objective (Atanackovic et al., 2024), a distributional divergence (e.g. Sinkhorn/MMD) (He et al., 2025), or any other generative transport model (see Sec 3.3). Because z_{src} obeys the CLT, we train with minibatches from S_{src} and S_{tgt} without biasing the population-level objective. After training, $\mathcal{T}(\cdot \mid \mathcal{E}(\cdot))$ can be applied to unseen sources to generate counterfactual realizations under their implied targets.

2.3. Unsupervised (any-to-any) transport: source-target conditioning

In many biological settings we do not have clean source-target pairings. Returning to our running lineage tracing case we actually only observe the same clone in both timepoints a small fraction of the time. If we restrict ourselves to supervised transport models we have to throw away 90% of the clones we observe.

To leverage this additional data we develop an "unsupervised" analogue to the "supervised" transport setting. In the unsupervised world we simply have a set of unstructured distributions: $P_1, \dots P_n \sim Q$ and associated samples:

$$S_i = \{x_{ij}\}_{j=1}^{m_i} \sim P_i$$

Our goal is to develop a model capable of learning to transport between any two distributions i and i'. Multimarginal stochastic interpolants (Albergo et al., 2023a) developed a class of models that can flow between any pair of distributions from a fixed initial set of K distributions by effectively embedding each distribution as a corner of the K-simplex and conditioning on the source and target corners. Our notion of source-target conditioned transport models leverage distribution encoders to learn distribution embeddings, enabling us to generalize to a new $(K+1)^{\text{th}}$ distribution, and to embed a continuous number of distributions.

$$\mathcal{T} : \mathcal{X} \times \mathbb{R}^d \times \mathbb{R}^d \to \mathcal{X},\ x_i \mapsto \widehat{x}_{i \to i'} = \mathcal{T}\big(x_i \mid z_i, z_{i'}\big),$$

to satisfy

$$\mathcal{T}(S_i \mid \mathcal{E}(S_i), \mathcal{E}(S_{i'})) \xrightarrow[m_i, m_{i'} \to \infty]{d} P_{i'} \tag{5}$$

Then using the same logic of the CLT we can train on samples using the objective conditioning on the encoder for the transport map of choice

$$\mathcal{L}_{\text{stc}} = \ell\left(S_{i'}, \mathcal{T}(S_i \mid \mathcal{E}(S_i), \mathcal{E}(S_{i'}))\right) \tag{6}$$

2.4. Semi-supervised transport: specializing source-target transport for supervised tasks

The goal for these unsupervised transport maps is really partial supervision: we want to train a transport map that can use all our data to learn a high-quality generative model, but at the end of the day we still care about prediction in the "paired" setting. There are two major points that are relevant for adapting source-target conditioned models to supervised tasks. First we outline how we can make latent predictions to enable supervised prediction from an unsupervised model; second we will comment on how we can choose pairings between source and target to leverage existing structure in the underlying distributions that "align" with the supervised task we are interested in.

To make this concrete, we can return to the lineage tracing case where we want to leverage all the clones, but the goal really is cell fate prediction. To actually convert the any-to-any model into a "supervised" model we can train a lightweight model to predict $z_{\text{src}} \mapsto z_{\text{tgt}}$ using the subset of paired (src, tgt) examples available for the task of interest; we then evaluate $\mathcal{T}(\cdot \mid z_{\text{src}}, \widehat{z}_{\text{tgt}})$ to generate counterfactual samples "as if" drawn from P_{tgt} in a semi-supervised manner. For additional performance, we can actually co-train the "supervised" latent predictor along with the transport model by adding a predictor loss term

$$\mathcal{L}_{\text{cotrain}} = \mathcal{L}_{\text{stc}} + \lambda \cdot \ell'\left(z_{\text{tgt}}, \widehat{z}_{\text{tgt}}\right), \tag{7}$$

with λ a weighing hyperparameter. This additional loss term regularizes the latent space learned by the encoder for downstream training of the lightweight model.

A second important practical point here is that we may not actually need nor want to learn to transport between any two distributions. Once more using the lineage tracing case for inspiration, it may make more sense to learn to transport any clone from timepoint one to any clone at timepoint two, instead of also learning allowing within timepoint and reverse time transport. The source-target conditioned model supports any-to-any inference but that does not mean we have to actually train on all pairs of datasets. We can instead *choose* a meta-distribution Q that defines a distribution over source-target pairs which we will train on. This allows analysts to encode relevant domain-specific structure into the underlying model through the design of Q. For example we can encourage the model to learn transport forward in time rather than the less structured any-to-any setting.

3. Related work

3.1. Distribution Embedding

Several lines of literature have tried to learn distribution embeddings or summary statistics. Kernel methods, such as kernel mean embedding (KME) and set kernels, provide nonparametric approaches to represent probability measures as points in a reproducing kernel Hilbert space, enabling tasks like distributional regression and classification (Smola et al., 2007; Muandet et al., 2012; Oliva et al., 2013; Szabo et al., 2015; Muandet et al., 2017). Our distribution embedding framework follows the formal framework in (Fishman et al., 2025) to learn distribution invariant-embeddings leveraging permutation-invariant architectures with a particular class of pooling operators (Zaheer et al., 2017; Wagstaff et al., 2021; Zhang et al., 2022).

3.2. Distribution Transport

Meta Flow Matching (Atanackovic et al., 2024) develops the idea of source-conditioning for learning transport maps across contexts. Our notion of source-conditioned models generalizes their approach to a broader class of transport models and demonstrates the conditions under which we can expect it to achieve (3). In single cell RNA-seq modeling there are a number of recent works that are effectively source-conditioned models (Klein et al., 2025; He et al., 2025; Adduri et al., 2025) as it is a useful technique for generalizing across cell types.

Multimarginal stochastic interpolants (MMSI) are the closest analogue to our any-to-any transport, except they require a fixed set of K distributions (Albergo et al., 2023a) and enforce transport paths that go through "all" distributions to learn a barycenter, while we adaptively learn our transport paths through the distribution space. MMSI models "time" on the simplex, where each point in the simplex corresponds to an interpolant taking a convex combination of a sample from each of the K distributions proportional to their simplicial weight. The key choice here is what paths through the simplex to choose. If we consider the "edge path" where we flow directly between two corners of the simplex with no weight on other distributions then MMSI is a stochastic interpolant (or equivalently flow matching) transport model paired with a one-hot distribution encoder, but instead of flexibly combining $z_i, z_{i'}$ and t the model operates on $tz_i + (1-t)z_{i'}$, forcing the model to combine its understanding of time and the distribution embedding.

Style-transfer and unpaired image-translation methods provide a complementary line of work. Classical neural style transfer (Gatys et al., 2015; 2016) encodes "style" implicitly via Gram matrices of CNN features, but does not define an explicit or generalizable latent representation of the style domain. Multi-domain translation frameworks such as CycleGAN, StarGAN, and MUNIT (Zhu et al., 2017; Choi et al., 2018; Huang et al., 2018) condition the generator on a domain identifier, typically a one-hot categorical label or an instance-level style code. These approaches are restricted to a fixed, finite set of domains and cannot generalize transport to a new domain without retraining. In contrast, our source–target–conditioned transport learns *distribution embeddings* directly from sets of samples, enabling conditioning on a continuous family of contexts and permitting transport to previously unseen distributions.

3.3. Transport Models

A wide range of generative transport mechanisms can be used as the conditional map $\mathcal{T}(\cdot \mid z)$ or $\mathcal{T}(\cdot \mid z, z')$ in our framework. Classical adversarial approaches learn source–target mappings via discriminators, including GANs (Goodfellow et al., 2014) and Wasserstein GANs (Arjovsky et al., 2017). Kernel-based methods transport distributions by matching maximum mean discrepancy (MMD), as in generative moment matching networks (Li et al., 2015). Optimal-transport–based generative models construct maps using entropic or Sinkhorn divergences (Genevay et al., 2018b). Normalizing flows provide invertible transport parameterizations through continuous change-of-variable models (Rezende & Mohamed, 2015). Continuous-time flow-based models, such as flow matching, rectified flows, and stochastic interpolants (Lipman et al., 2023; Liu et al., 2022; Albergo et al., 2023b), define transport by learning a velocity field along a path interpolating between source and target.

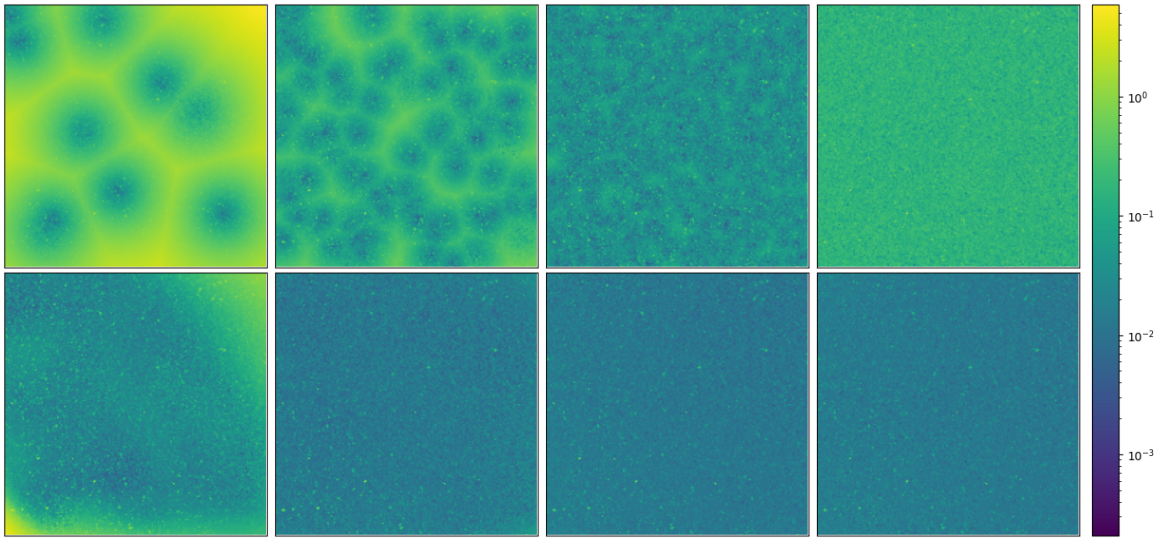

Figure 1. We compare a K-to-K transport model to the source-target conditioned any-to-any model for learning to transport between Gaussian distributions. We plot the W_2 distance between true and transported datapoints for all values of $\mu \in [0, 5]^2$. The top row in the K-to-K model, the bottom is the any-to-any model. Each column corresponds to models trained on different numbers of distributions $K = 10,\ 100,\ 1{,}000,\ 10{,}000$, but we fix the number of sample sets at $n = 50{,}000$. For the K-to-K model we compute the closest dataset in its set of K and use that as the prediction, which gives rise to the Voronoi-like strucutre in the first several plots; for the any-to-any model we can simply embed the target. For all K the any-to-any model achieves much better and more uniform transport. For $K = 10{,}000$ we also see the complete breakdown of the K-to-K model as it fails to learn high-quality embeddings for every distribution.

On discrete sequences, unsupervised machine translation methods (Lample et al., 2018a; Artetxe et al., 2018; Lample et al., 2018b) learn bidirectional maps between languages using only monolingual corpora, providing a discrete analogue of our unpaired any-to-any transport between distributions. More recently, flow-matching ideas have been extended to discrete domains via discrete flow matching and related categorical flow-matching frameworks that operate directly on probability simplices (Gat & Lipman, 2024; Cheng et al., 2024; Davis et al., 2024).

All of these models define mappings that take samples from a source distribution and produce samples matching a target distribution. Our framework is orthogonal to the specific transport mechanism: any such model can be conditioned on distribution embeddings to yield a source-conditioned or source–target–conditioned transport map that satisfies the population-level limits in (3) and (5).

4. Unsupervised Learning Enables Zero-Shot Transport

4.1. Gaussian Transport

As a first illustration, we study transport between simple Gaussian distributions, where we can control the data-generating process and directly visualize performance. This experiment introduces the main components of our framework—distribution encoders and source–target–conditioned transport maps—in a setting where the ground-truth structure is transparent.

We construct a dataset of bivariate normal distributions by sampling parameters $(\mu_i, \Sigma_i),\ i = 1, \ldots, n$, with means $\mu_i \in [0, 5]^2$ drawn from a uniform prior and covariances $\Sigma_i \in \mathbb{R}^{2\times 2}$ drawn from a simple inverse–Wishart prior (see App. **??** for details). We then choose $K \in \{10, 100, 1{,}000, 10{,}000\}$ distinct parameter sets and draw $n = 50{,}000$ sample sets so that each distribution contributes n/K sample sets to the dataset.

For each value of K we train two models on the same dataset. The first is a conventional *K-to-K* architecture that treats each of the K distinct Gaussians as a discrete label (a "corner") and learns transport conditional on the distribution-specific labels. At test time this model has no way to condition on a new target distribution, so we follow the natural baseline of assigning each target to its nearest training distribution and using that as the target condition. The second is our *source–target conditioned* model from Sec. 2.3, which encodes both source and target sets via the distribution encoder $z_i = \mathcal{E}(S_i)$ and learns a transport map $\mathcal{T}(x \mid z_{\text{src}}, z_{\text{tgt}})$ that can operate between arbitrary pairs of distributions, including unseen targets. For these experiments we use a mean-pooled DeepSets encoder as $\mathcal{E}$ and a flow matching model as the transport map. The flow matching architectures are identical across models, with

K	K-to-K W_2^2	Any-to-any W_2^2
10	0.7779	0.0870
100	0.1369	0.0166
1,000	0.0435	0.0150
10,000	0.1683	0.0157

Table 1. Quantitative averages of the results visualized in Fig. 1. The any-to-any model performs substantially better overall and, unlike the K-to-K baseline, does not degrade when the number of distinct distributions K becomes very large.

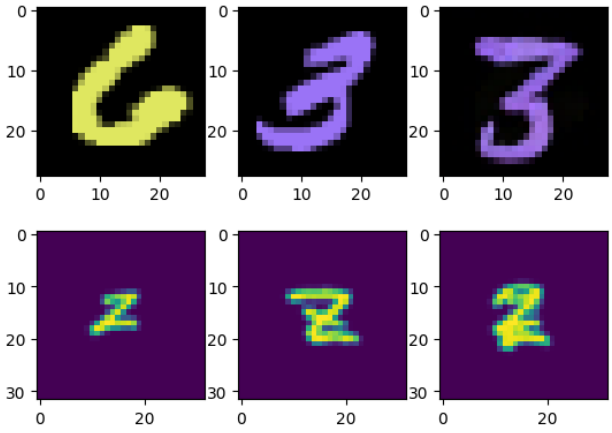

Figure 2. Here we present two examples of image style transfer. The first image is the sample from the source distribution, the second is a sample from the target distribution, and the third image is the first image transported to the target distribution conditioning on the source and target embeddings. Both tasks use completely heldout source and target distributions illustrating the model's capacity for zero-shot generalization.

embeddings of the same dimension.

To visualize zero-shot performance we fix a single source distribution from the training set and evaluate transport to a dense grid of target parameters $\mu \in [0, 5]^2$ with corresponding random covariances Σ. The results are visualized in Fig. 1, and a quantitative comparison is given in Table 1.

4.2. Unsupervised Style Transfer

We can also leverage our any-to-any transport model for style transfer on images without language embeddings when the data are grouped into coherent distributions. Our goal here is not to propose a competitive method for classical image style transfer, where styles are typically specified by language tags and each tag may correspond to only a handful of images. In such settings, high-quality language embeddings already provide a natural representation of "style," so there is little need to learn separate distribution encoders. Instead, this experiment serves as a sanity check demonstrating that, when images are partitioned into clean sets, our approach can learn transport maps between these image distributions—a situation that more closely mirrors our biological applications, where we often lack high-quality side information to condition on.

MNIST Colors We first use the MNIST dataset to illustrate style transfer in this distributional setting. Each distribution is defined by a particular RGB color and digit class, e.g., "yellow sixes" or "purple threes." By training on a large number of colors we can teach the model to zero-shot transport between digit-color distributions. We illustrate this in Fig. 2 where we use heldout colors to demonstrate how the model can map from yellow sixes to purple threes.

EMNIST Handwriting We next consider the EMNIST dataset, where we treat each writer as a "style" and each character as content. We form one distribution per (writer, character) pair by collecting all images of that character written by a given writer, and train the any-to-any model to transport between these distributions. At test time we evaluate zero-shot style transfer between unseen writers: the model is given source and target sets from writers that were never observed during training and must map individual characters from the source style to the target style. In Fig. 2 we show a representative example for two heldout authors and the letter "z," where our model successfully transfers from a writer who draws a simple diagonal "z" with no crossbar to a writer whose "z" has a horizontal crossbar.

5. Improving Semi-supervised Learning

In the following we present a wide range of applications of DCTs that demonstrate how our method can be applied to multiple very different domains and perform well across the board.

5.1. Forecasting

Forecasting time-series data is a natural application of distribution transport models. Recent work (Berlinghieri et al., 2025) introduced two forecasting benchmarks defined on point clouds: ocean current trajectories and PBMC developmental trajectories (see App. A for details).

In a purely supervised approach such as MFM, a model is trained only to transport from one time point to the next. When we train DCTs in this adjacent-pair fashion—pairing each time t only with $t{+}1$—we obtain reasonable performance, but we do not surpass the state of the art.

We therefore adopt a semi-supervised training scheme in which we pair between any two time points during training. As a first attempt, we train the encoder–generator pair in a fully unsupervised any-to-any fashion, and only after

Table 2. Gulf of Mexico Vortex Forecast

Method	GoM	PBMC
snapMMD	0.66±0.031	0.0042±0.0017
SBIRR-ref	0.35±0.032	0.097±0.060
SBIRR-forward	0.62±0.054	0.51±0.16
DCTs		
paired (MFM)	0.249±0.105	0.014±0.006
any-to-any	0.61±0.112	0.0062±0.0035
any-to-any, cotrained	**0.15±0.113**	**0.0039±0.0019**
any-to-any, unlabeled	0.34±0.078	0.0046±0.0031

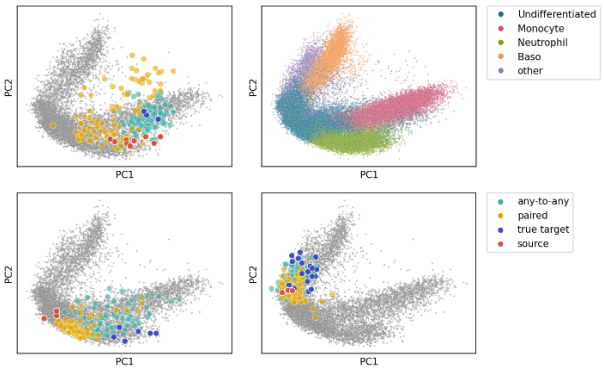

Figure 3. Any-to-any vs. paired pushforward prediction on (Weinreb et al., 2020). Visualizing first 2 PCs of true and generated data. Each subplot shows a single clone, chosen based on difference between energy distance of predictions between models.

Method	Energy Distance ↓
Paired (supervised)	3.43 ± 0.10
Any-to-any (semi-supervised)	$\mathbf{3.17} \pm 0.09$

Table 3. Comparison of pushforward performance on the Weinreb et al. (2020). We report the mean±s.e.m of energy distance (lower is better) across 628 held-out test clones.

convergence do we train a latent predictor in a supervised way on the frozen encoder embeddings. This strict separation between unsupervised and supervised phases yields performance comparable to the purely supervised baseline. Intuitively, the encoder has no incentive to structure the latent space in a way that is convenient for forecasting, and because it collapses each distribution at time t into a single latent vector, time-series with only 10–20 time points provide very few supervised examples for the predictor. To address this, we instead co-train a ridge-regression predictor together with the encoder–generator pair, and in this setting we obtain state-of-the-art performance on both benchmarks.

We further extend the any-to-any pairing strategy by incorporating data that lack explicit time labels, which cannot be used for standard supervised training. Including these unlabeled distributions increases the amount of data available for the encoder–generator and improves performance compared to the strictly labeled any-to-any setting, though it does not quite match the MMD scores of the fully co-trained any-to-any model. The likely reason is that, while the additional unlabeled data help learn better distribution embeddings, the absence of time labels prevents co-training the latent predictor on a large fraction of the data, leading to a less tightly structured latent space for forecasting.

5.2. Clonal population dynamics

We next apply DCTs to learn clonal population dynamics using lineage-traced single-cell RNA-sequencing (scRNA-seq) data from Weinreb et al. (2020). A critical challenge in this dataset is sparsity: there are about $6 \cdot 10^3$ clones measured in total, however only about $2 \cdot 10^3$ of these clones are profiled at multiple timepoints. The majority of clones are "orphans" observed only at a single timepoint.

While source-conditioned approaches can only be trained on clones observed at multiple timepoints, training DCTs with any-to-any pairings allow us to make use of all available marginals. To assess the benefit of moving to the semi-supervised paradigm, we compare two training approaches. First is a baseline which is exclusively trained on paired clones, akin to meta-flow matching. Second is an any-to-any model trained on all pairs of observed marginals.

Both settings use an identical architecture operating on the first 50 principal components (PCs) of the gene expression profiles. The model consists of a mean-pooled deep sets encoder and an energy-based model generator. For latent prediction, we use a post-hoc ridge regression. As shown in Table 5.2, the any-to-any training objective outperforms the source-conditioned baseline, demonstrating the benefit of incorporating unpaired clones into the learning process. In Fig. 3, we visualize the predictions of three clones which have large differences between the two models.

6. Conclusion

We introduced distribution-conditioned transport, a general framework that couples distribution encoders with conditional transport models to enable zero-shot transport between arbitrary pairs of distributions. By viewing sets of samples as first-class objects and conditioning transport maps on learned source and target embeddings, our any-to-any formulation can exploit unpaired marginals, interpolate across contexts, and generalize to distributions unseen during training. Empirically, DCTs outperform K-to-K baselines on controlled Gaussian benchmarks, support zero-shot image style transfer, and achieve state-of-the-art perfor-

mance on recent distributional forecasting tasks, while also improving pushforward prediction in sparse lineage-tracing scRNA-seq.

Our results highlight that unsupervised structure in the space of distributions can be harnessed to strengthen semi-supervised learning, particularly when paired with lightweight predictors in latent space. Looking ahead, we see opportunities to refine the design of pairing policies, develop more expressive yet data-efficient latent predictors, and extend distribution-conditioned transport to broader domains where rich collections of related empirical distributions arise naturally, from large-scale biological assays to complex dynamical systems.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Adduri, A. K., Gautam, D., Bevilacqua, B., Imran, A., Shah, R., Naghipourfar, M., Teyssier, N., Ilango, R., Nagaraj, S., Dong, M., et al. Predicting cellular responses to perturbation across diverse contexts with state. *bioRxiv*, pp. 2025–06, 2025.

Albergo, M. S., Boffi, N. M., Lindsey, M., and Vanden-Eijnden, E. Multimarginal generative modeling with stochastic interpolants. *arXiv preprint arXiv:2310.03695*, 2023a.

Albergo, M. S., Boffi, N. M., and Vanden-Eijnden, E. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 24:1–63, 2023b.

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein GAN. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pp. 214–223. PMLR, 2017.

Artetxe, M., Labaka, G., Agirre, E., and Cho, K. Unsupervised neural machine translation. In *International Conference on Learning Representations*, 2018.

Atanackovic, L., Zhang, X., Amos, B., Blanchette, M., Lee, L. J., Bengio, Y., Tong, A., and Neklyudov, K. Meta Flow Matching: Integrating Vector Fields on the Wasserstein Manifold. In *The Thirteenth International Conference on Learning Representations*, 4 October 2024.

Berlinghieri, R., Shen, Y., Jiang, J., and Broderick, T. Oh snapmmd! forecasting stochastic dynamics beyond the schr\" odinger bridge's end. *arXiv preprint arXiv:2505.16082*, 2025.

Biddy, B. A., Kong, W., Kamimoto, K., Guo, C., Waye, S. E., Sun, T., and Morris, S. A. Single-cell mapping of lineage and identity in direct reprogramming. *Nature*, 564(7735):219–224, 2018.

Cheng, X., Chen, T., Li, K., Gao, R., Zhu, J., and Liu, Q. Categorical flow matching. In *International Conference on Learning Representations*, 2024.

Choi, Y., Choi, M., Kim, M., Ha, J.-W., Kim, S., and Choo, J. StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8789–8797, 2018.

Davis, N., Chen, R. T. Q., Hu, Y., and Rezende, D. J. Fisher flow matching: Transport in the probability simplex. In *International Conference on Machine Learning*, 2024.

Fishman, N., Gowri, G., Yin, P., Gootenberg, J., and Abudayyeh, O. Generative distribution embeddings. *arXiv preprint arXiv:2505.18150*, 2025.

Gat, I. and Lipman, Y. Discrete flow matching. In *Advances in Neural Information Processing Systems*, 2024.

Gatys, L. A., Ecker, A. S., and Bethge, M. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015.

Gatys, L. A., Ecker, A. S., and Bethge, M. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2414–2423, 2016.

Genevay, A., Peyre, G., and Cuturi, M. Learning Generative Models with Sinkhorn Divergences. In *International Conference on Artificial Intelligence and Statistics*, pp. 1608–1617. PMLR, 31 March 2018a.

Genevay, A., Peyré, G., and Cuturi, M. Learning generative models with sinkhorn divergences. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84, pp. 1608–1617. PMLR, 2018b.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, volume 27, 2014.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 22 October 2020. ISSN 0001-0782,1557-7317. doi: 10.1145/3422622.

He, C., Zhang, J., Dahleh, M., and Uhler, C. Morph predicts the single-cell outcome of genetic perturbations across conditions and data modalities. *bioRxiv*, 2025.

Huang, X., Liu, M.-Y., Belongie, S., and Kautz, J. Multimodal unsupervised image-to-image translation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 179–196, 2018.

Klein, D., Fleck, J. S., Bobrovskiy, D., Zimmermann, L., Becker, S., Palma, A., Dony, L., Tejada-Lapuerta, A., Huguet, G., Lin, H.-C., et al. Cellflow enables generative single-cell phenotype modeling with flow matching. *bioRxiv*, pp. 2025–04, 2025.

Lample, G., Conneau, A., Denoyer, L., and Ranzato, M. Unsupervised machine translation using monolingual corpora only. In *International Conference on Learning Representations*, 2018a.

Lample, G., Ott, M., Conneau, A., Denoyer, L., and Ranzato, M. Phrase-based & neural unsupervised machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 5039–5049. Association for Computational Linguistics, 2018b.

Li, Y., Swersky, K., and Zemel, R. S. Generative moment matching networks. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, pp. 1718–1727. PMLR, 2015.

Lipman, Y., Chen, R. T. Q., Ben-Hamu, H., Nickel, M., and Le, M. Flow Matching for generative modeling. *International Conference on Learning Representations*, abs/2210.02747, 6 October 2022.

Lipman, Y., Chen, R. T. Q., Ben-Hamu, H., Nickel, M., and Le, M. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Liu, X., Gong, C., and Liu, Q. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Muandet, K., Fukumizu, K., Dinuzzo, F., and Scholkopf, B. Learning from distributions via support measure machines. *Neural Information Processing Systems*, 25:10–18, 29 February 2012.

Muandet, K., Fukumizu, K., Sriperumbudur, B., and Schölkopf, B. Kernel mean embedding of distributions: A review and beyond. *Foundations and Trends® in Machine Learning*, 10(1-2):1–141, 2017. ISSN 1935-8237,1935-8245. doi: 10.1561/2200000060.

Oliva, J. B., Póczos, B., and Schneider, J. Distribution to Distribution Regression. *International Conference on Machine Learning*, 28(3):1049–1057, 16 June 2013.

Rezende, D. J. and Mohamed, S. Variational inference with normalizing flows. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, pp. 1530–1538. PMLR, 2015.

Smola, A., Gretton, A., Song, L., and Schölkopf, B. A Hilbert space embedding for distributions. In *Lecture Notes in Computer Science*, Lecture notes in computer science, pp. 13–31. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007. ISBN 9783540752240,9783540752257. doi: 10.1007/978-3-540-75225-7_5.

Szabo, Z., Gretton, A., Poczos, B., and Sriperumbudur, B. Two-stage sampled learning theory on distributions. In *Artificial Intelligence and Statistics*, pp. 948–957. PMLR, 21 February 2015.

Wagner, D. E. and Klein, A. M. Lineage tracing meets single-cell omics: opportunities and challenges. *Nature Reviews Genetics*, 21(7):410–427, 2020.

Wagstaff, E., Fuchs, F., Engelcke, M., Osborne, M. A., and Posner, I. Universal approximation of functions on sets. *Journal of machine learning research: JMLR*, 23(151): 151:1–151:56, 5 July 2021. ISSN 1532-4435,1533-7928.

Weinreb, C., Rodriguez-Fraticelli, A., Camargo, F. D., and Klein, A. M. Lineage tracing on transcriptional landscapes links state to fate during differentiation. *Science*, 367(6479), 14 February 2020. ISSN 0036-8075,1095-9203. doi: 10.1126/science.aaw3381.

Zaheer, M., Kottur, S., Ravanbakhsh, S., Póczos, B., Salakhutdinov, R., and Smola, A. Deep Sets. *Advances in neural information processing systems*, 30, 10 March 2017.

Zhang, L. H., Tozzo, V., Higgins, J., and Ranganath, R. Set norm and equivariant skip connections: Putting the deep in Deep Sets. *International Conference on Machine Learning*, 162:26559–26574, 23 June 2022. doi: 10.48550/arXiv.2206.11925.

Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2223–2232, 2017.

A. Forecasting

A.1. Gulf of Mexico

A.1.1. Dataset preprocessing details

This dataset consists of the trajectories of 400 particles flowing along an ocean current. It was generated by integrating the velocity field of a vortex observed in the Gulf of Mexico on June 1st, 2024 at 5pm. Initial positions were sampled withing a small radius and the positions across 11 time points make up the dataset. The final time point is held out for the forecasting benchmark.

A.1.2. TDE architecture and training details

We train a mean-pooled fully connected MLP as the distribution encoder and an energy-based model (EBM) as the generator (need to add details of EBM). Models were trained over 1,000 epochs with the ADAM optimizer and learning rate $\eta = ...$ We train the predictor model after the training of encoder and generator has completed in a supervised manner, training it to map the latent representation between consecutive time points. Due to the compressed nature of our distribution embeddings, we train a simple ridge regressor with $\alpha = ...$ The dataset for the predictor is generated by encoding $N = ...$ subsets sampled at random for each time point. To better structure the latent space for the downstream training of the predictor model, we cotrain a similar ridge regression model along the encoder and generator models and add a small loss termn (cosine distance) to the training loss.

A.2. PBMC

A.2.1. Dataset preprocessing details

This dataset consists of a 30-dimensional projection of single-cell RNA-sequencing data of peripheral blood mononuclear cells (PBMCs) that were collected every hour for 21 hours. Here we hold out the last hour for forecasting validation.

A.2.2. TDE architecture and training details

TDE architecture follows the same structure as our implementation for the GoM task. Need to add hyperaprameters used.

B. GISAID spike protein sequences

B.1. Dataset preprocessing details

This dataset consists of SARS-CoV2 spike protein sequences over time and location. We used data from the Global Initiative on Sharing All Influenza Data (GISAID) up to and including January 2025. Sequences were grouped by sampling month and location (country and state), and each group was treated as an empirical distribution of protein sequences. Entries with invalid date fields and ambiguous or undetermined residues were removed. For tokenization, sequences were truncated to 1000 amino acids.

B.2. TDE architecture and training details

The encoder consists of multi-head self-attention blocks stacked on mean-pooled embeddings generated by the ESM2-8M model. For the generator we finetune another instance of the ESM2-8M model with a discrete flow matching objective to mutate source sequences into target sequences. Both for the encoder and the generator, we load the pretrained weights but do not freeze them.

(Predictor still needs to be trained, will probably be similar to the other datasets, so just ridge regression of some sort).

C. Hematopoiesis lineage-tracing scRNA-seq

C.1. Dataset preprocessing details

C.2. TDE architecture and training details

D. Implementations of baseline methods

D.1. Meta flow matching

For the implementation of MFM we follow the details of the original paper (): the encoder architecture consists of a mean-pooled GNN that receives a kNN graph with k = ... of the source samples as input. For the generator, we use a flow-matching archtiecture. In contrast to TDEs, we condition the generator only on the source latent embeddings, so with no conditioning on the target latent.

D.2. SnapMMD

Just using the results from the paper (and when I fiddled around with it I just used the code they uploaded to github together with their implementation).

D.3. other baselines you may have tried

None