SHORTEST-PATH FLOW MATCHING WITH MIXTURE-CONDITIONED BASES FOR UNSEEN-CONDITION GENERALIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Robust generalization under distribution shift remains a key challenge for conditional generative modeling: conditional flow-based methods often fit the training conditions well but fail to extrapolate to unseen ones. This is particularly important for therapeutic prioritization, where the goal is to predict cellular responses to unseen candidate drugs and genetic perturbations in order to select the most effective intervention. We introduce **SP-FM**, a shortest-path flow-matching framework that improves generalization to unseen conditions by conditioning both the base distribution and the flow field on the condition. Specifically, SP-FM learns a condition-dependent base distribution parameterized as a flexible, learnable mixture, together with a condition-dependent vector field trained via shortest-path flow matching. Conditioning the base allows the model to adapt its starting distribution across conditions, enabling smooth interpolation and more reliable extrapolation beyond the observed training range. We provide theoretical insights into the resulting conditional transport and show how mixture-conditioned bases enhance robustness under shift. Empirically, SP-FM is effective across heterogeneous domains, including predicting responses to unseen perturbations in single-cell transcriptomics and modeling treatment effects in microscopy-based drug screening. Overall, SP-FM provides a simple yet effective plug-in strategy for improving conditional generative modeling and generalization to unseen conditions.

1 INTRODUCTION

Conditional generative models have achieved remarkable success across domains ranging from image synthesis to molecular design (Rombach et al., 2022; Ho & Salimans, 2022). A common paradigm underlying many state-of-the-art approaches, including diffusion models and flow matching, is to learn a transport map from a simple base distribution to a complex target distribution, conditioned on some input descriptor (Lipman et al., 2023; Tong et al., 2023; Benamou & Brenier, 2000). However, a critical limitation emerges when these models encounter *unseen conditions* at test time: their performance degrades substantially, limiting their utility in real-world applications where exhaustive enumeration of all possible conditions is infeasible.

Generalization to unseen conditions is particularly challenging in scientific applications because models are routinely asked to generate responses for *conditions that were never observed during training*. Here, we use *condition* to mean the *intervention or experimental setting* that indexes a data-generating distribution—for example, a drug identity, a genetic perturbation, or a specific rotation of a letter—rather than a shift in the input instances of a predictor. Conditional generative models are often brittle in this regime: they may successfully generate samples for the training condition set but degrade sharply on truly held-out conditions, producing overly averaged samples or missing condition-specific effects.[2]

This issue appears across scientific domains (Subramanian et al., 2017; Srivatsan et al., 2020; Norman et al., 2019; Replogle et al., 2022; Ahlmann-Eltze et al., 2025), underscoring the difficulty of

[2]In this paper, *generalization to unseen conditions* refers to generating from an unseen *condition* (intervention/setting) excluded from training, not to test-time covariate shift in predictor inputs. We assume the unseen condition is drawn from the same overall condition distribution.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

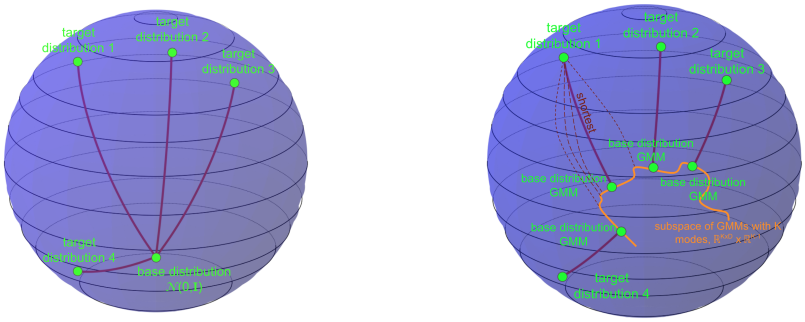


Figure 1: **Standard conditional flow matching versus SP-FM. (a)** Standard CFM fixes the source to $\mathcal{N}(0, I)$ and conditions only the velocity field on y, learning separate flows from the same base to different targets. **(b)** SP-FM also conditions the source: a y-dependent GMM base is chosen to shorten the geodesic to the target, so nearby descriptors induce nearby bases and improve interpolation/extrapolation to unseen conditions.

robust conditional generation under shift. Why do current conditional generative models struggle on unseen conditions? We argue that the root cause lies in an overlooked architectural choice: *the base distribution is fixed*. Standard conditional flow matching and diffusion models use a fixed Gaussian $\mathcal{N}(0, I)$ as the source distribution, conditioning only the decoder (velocity field or score network) on the input descriptor (Lipman et al., 2023; Klein et al., 2025; Adduri et al., 2026). Conditional diffusion/flow objectives can be viewed as learning score (or velocity) fields that behave as *condition-specific* estimators (Song & Ermon, 2019; Ho & Salimans, 2022) independently trained on each condition, and those independent score/density estimators are not expected to generalise to unseen conditions, unless, *e.g.,*, the decoder happens to be smooth with respect to its input descriptor. Given that conditioning the decoder is not effective to achieve generalisation, the base distribution is a natural choice to condition on population descriptor.

Recent work has begun to move beyond fixed base distributions, but these efforts target different goals than ours. Atanackovic et al. (2025) propose Meta Flow Matching, which conditions the dynamics on an *observed initial population* encoded by a graph neural network, effectively assuming access to the starting distribution at test time and focusing on transferring dynamics across contexts.

Other approaches construct application-specific priors to shorten transport paths and improve sampling efficiency (Bunne et al., 2023). However, no method learns a base distribution that varies with the condition descriptor for the purpose of generalization to unseen conditions. SP-FM explicitly *learns* a condition-dependent base distribution jointly with the flow field, with the goal of improving generalization to unseen conditions, and provides theoretical motivation for why conditioning the base can matter. In related work, Chen et al. (2025) also employ Gaussian mixtures in flow matching, but for a different purpose: they model the *velocity field* as a mixture to enable efficient few-step sampling, while retaining a fixed base distribution.

We propose a different perspective: *the base distribution itself should be conditioned on the descriptor*. Our key insight is that when two conditions are similar, whether two rotation angles of images that differ by a small increment, two drugs with related chemical structures (Rogers & Hahn, 2010), or two genetic perturbations affecting the same pathway (Dixit et al., 2016), their corresponding target distributions are also likely to be similar. By learning a base distribution that reflects this similarity structure, we encode prior knowledge about condition relationships directly into the generative process. The flow then needs only to correct for the residual differences, rather than learning the entire mapping from scratch.

We introduce **SP-FM**, a conditional flow-matching framework that jointly learns a descriptor-conditioned Gaussian mixture base distribution and a descriptor-conditioned velocity field trained via shortest-path flow matching. The key architectural change is simple and can be plugged into existing pipelines: instead of sampling from a fixed Gaussian, SP-FM uses learned networks to predict condition-specific mixture parameters that define the base distribution. Samples from this adaptive base are then transported to the target via a learned velocity field. Because similar descriptors produce similar base distributions, the model can smoothly interpolate and extrapolate to unseen conditions, a capability that fixed-base approaches fundamentally lack.

We provide theoretical analysis supporting this design. Building on connections between flow matching and optimal transport (Peyré & Cuturi, 2019; Benamou & Brenier, 2000), we analyze the degrees of freedom in the transport problem as a function of the number of mixture components. We show that when the base is a single Gaussian, the dual optimal transport problem becomes ill-defined, providing a formal explanation for the poor generalization of vanilla conditional flow matching. With multiple mixture components, the problem becomes well-posed, and we derive bounds relating the base complexity, target complexity, and data dimension.

Our main contributions are:

- We identify the fixed base distribution as a key limiting factor for generalization to unseen conditions in conditional flow matching and propose SP-FM, which jointly learns a descriptor-conditioned Gaussian mixture base and a shortest-path velocity field.
- We provide theoretical analysis showing that single-Gaussian bases lead to ill-posed inference, while mixture bases with sufficiently many components yield well-defined problems with bounded estimation error.
- We demonstrate consistent improvements over conditional flow matching as well as application-specific methods across eight datasets, including synthetic benchmarks and perturbation screening in high-content microscopy and single-cell transcriptomics, positioning SP-FM as a flexible framework for conditional generation with improved generalization to unseen conditions.

1.1 Related Work

Flow Matching and Continuous Normalizing Flows. Flow matching has emerged as a scalable framework for training continuous normalizing flows by regressing onto conditional velocity fields along interpolation paths between base and target distributions (Lipman et al., 2023). Unlike score-based diffusion models that require simulating stochastic differential equations, flow matching enables simulation-free training and deterministic sampling via ordinary differential equations. Extensions have incorporated optimal transport couplings to straighten trajectories and improve sample efficiency (Tong et al., 2023; Benamou & Brenier, 2000). A key limitation shared by these approaches is the reliance on a fixed base distribution—typically a standard Gaussian—which we argue fundamentally limits generalization to unseen conditions.

Conditional Generative Models. Conditioning generative models on auxiliary information has enabled applications ranging from class-conditional image synthesis (Ho & Salimans, 2022; Rombach et al., 2022) to text-to-image generation. In scientific domains, conditional generative models are used to predict cellular responses to perturbations (Lotfollahi et al., 2023; Hetzel et al., 2022; Roohani et al., 2024; Klein et al., 2025), model molecular dynamics (Bunne et al., 2023), and generate cell morphologies (Palma et al., 2025; Navidi et al., 2025; Zhang et al., 2025) under unseen drug perturbations. *Generalization to unseen conditions* can limit the usefulness of these models for biological discovery and drug development, especially in *combinatorial* treatment settings where exhaustive experimental screening is infeasible (Bertrand et al., 2025). One contributing factor is that conditional diffusion/flow objectives can be viewed as learning score (or velocity) fields that behave as *condition-specific* estimators (Song & Ermon, 2019; Ho & Salimans, 2022) independently trained on each condition: models may fit the training descriptors well, yet lack guarantees of extrapolation to novel descriptors. Recent work further suggests that failures of generalization can coincide with memorization, raising privacy concerns when training on licensed or sensitive biomedical data (Yoon et al., 2023; Bertrand et al., 2025; Bamberger et al., 2026). Together, these observations motivate methods that impose additional structure across the descriptor space to support systematic extrapolation.

Gaussian Mixtures in Generative Models. Gaussian mixture models (GMMs) have a long history as flexible density estimators with universal approximation properties (Goodfellow et al., 2016). Recent work has revisited GMMs in the context of modern generative models. Chen et al. (2025) propose GMFlow, which models the *velocity distribution* as a Gaussian mixture to capture multimodal flow directions, improving few-step sampling and reducing over-saturation artifacts. Ahamed et al. (2025) use GMMs in latent spaces for molecular generation. These approaches modify how the

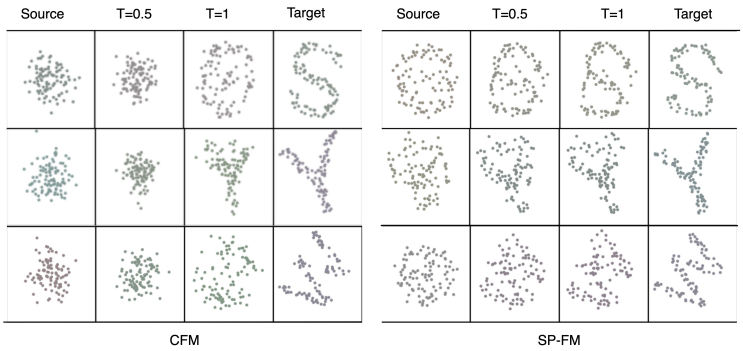


Figure 2: **Synthetic qualitative results.** Samples along the integration trajectory ($t = 0, 0.5, 1$) with the corresponding targets. SP-FM yields smoother trajectories and more faithful reconstructions on unseen rotations than CFM.

velocity or score is represented but do not condition the *base distribution* on input descriptors. In contrast, SP-FM learns a descriptor-conditioned GMM as the source distribution.

Learned and Adaptive Prior Distributions. Several works have explored moving beyond fixed Gaussian priors. In the context of variational autoencoders, learned priors have been used to improve expressiveness (Bereket & Karaletsos, 2023). For flow-based models, Bunne et al. (2023) learn transport maps via neural optimal transport for single-cell perturbation modeling, though with a focus on paired control-treatment settings rather than generalization to unseen conditions. Meta Flow Matching (Atanackovic et al., 2025) conditions on initial populations by embedding them via graph neural networks, enabling generalization across different starting distributions on the Wasserstein manifold. This approach is complementary to ours: while Meta Flow Matching learns to adapt to varying initial populations, SP-FM learns to predict appropriate base distributions from condition descriptors, directly targeting the unseen conditions generalization problem.

2 Problem Setup

We study population-level generative modelling. Let $\{\rho_1, \dots, \rho_N\}$ denote N probability measures on $\mathbb{R}^D$, each corresponding to a distinct population. For each $n \in \{1, \dots, N\}$ we observe a dataset $\boldsymbol{X}_n = \{\boldsymbol{x}_{n,1}, \dots, \boldsymbol{x}_{n,s_n}\} \subset \mathbb{R}^D$, consisting of s_n samples drawn i.i.d. from ρ_n. Each population is associated with a descriptor $\boldsymbol{y}_n \in \mathcal{Y}$ that encodes metadata about the population (e.g., the rotation angle of a letter, perturbation applied to cells, a molecular embedding of a drug, or cell-type information).

Given training data $\{(\boldsymbol{X}_n, \boldsymbol{y}_n)\}_{n=1}^N$, the aim is to learn a conditional generative model that produces samples from ρ_n when provided with $\boldsymbol{y}_n$. At test time, the model is presented with an unseen descriptor $\boldsymbol{y}_{\text{test}}$ and must generate samples approximating the corresponding, unknown distribution ρ_{test}. This setup captures, for example, predicting the distribution of gene expression states under new perturbations, conditioned on information about the perturbation, baseline cells, and biological experimental conditions.

3 Mixture-Conditioned Flow Matching

Standard conditional flow matching fixes the base distribution to $\mathcal{N}(\mathbf{0}_D, \sigma^2 \mathbf{I}_{D\times D})$ and conditions only the decoder, effectively yielding independent estimators for each population and limiting generalization (See, Fig. 1). SP-FM instead conditions both the base distribution and the flow module on perturbations and covariates. (Fig. 1).

Let I denote the number of mixture components. Plausible base distributions are Gaussian mixtures $\mu_{\text{GMM}(\boldsymbol{\Theta},\boldsymbol{p})}$ with $\boldsymbol{\Theta} \in \mathbb{R}^{I\times D}$ specifying mode locations, $\boldsymbol{p} \in \mathbb{S}^{I-1}$ specifying mixture weights, and

fixed component variance σ^2. The family of such mixtures forms the restricted subspace illustrated in Fig. 1 in orange.

Given populations $\{\rho_1, \ldots, \rho_N\}$ with descriptors $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N\}$, each ρ_n is projected to its closest element $\mu_{\mathrm{GMM}(\boldsymbol{\Theta}^{(n)}, \boldsymbol{p}^{(n)})}$ in this subspace. The optimal transport to ρ_n from its projection defines a velocity field $v(., t; \boldsymbol{y}_n)$. These velocity fields form geodesics on the 2-Wasserstein manifold of probability measures, which are shown by the red curves in Fig. 1. Predictors $h_{\boldsymbol{\Theta}} : \mathcal{Y} \to \mathbb{R}^{I \times D}$ and $h_{\boldsymbol{p}} : \mathcal{Y} \to \mathbb{S}^{I-1}$ are trained so that $h_{\boldsymbol{\Theta}}(\boldsymbol{y}_n) \approx \boldsymbol{\Theta}^{(n)}$ and $h_{\boldsymbol{p}}(\boldsymbol{y}_n) \approx \boldsymbol{p}^{(n)}$. At test time, the predicted base $\mu_{\mathrm{GMM}(h_{\boldsymbol{\Theta}}(\boldsymbol{y}_{test}), h_{\boldsymbol{p}}(\boldsymbol{y}_{test}))}$ is evolved by $v(., t; \boldsymbol{y}_{test})$ to approximate ρ_{test}.

3.1 Training objectives

SP-FM is trained with two complementary losses.

Objective 1. Align the learned velocity field $v_\theta(., t; \boldsymbol{y}_n)$ with the OT path between ρ_n and its projected base:

$$\mathcal{L}_{\mathrm{OT}} = \mathbb{E}\left[\sum_{s=1}^{S} \left\| v\Big(t, (1-t)\boldsymbol{x}_s^{(0)} + t\boldsymbol{x}_{\tau(s)}, \boldsymbol{y}_n\Big) - (\boldsymbol{x}_{\tau(s)} - \boldsymbol{x}_s^{(0)}) \right\|^2\right] \tag{1}$$

, where $\{\boldsymbol{x}_s\} \sim \rho_n$, $\{\boldsymbol{x}_s^{(0)}\} \sim \mu_{GMM(h_{\boldsymbol{\Theta}}(\boldsymbol{y}_n), h_{\boldsymbol{p}}(\boldsymbol{y}_n))}$, $t \sim \mathrm{Unif}(0, 1)$, and τ denotes mini-batch OT pairing (Tong et al., 2023).

Objective 2. Encourage each projection $\mu_{GMM(h_{\boldsymbol{\Theta}}(\boldsymbol{y}_n), h_{\boldsymbol{p}}(\boldsymbol{y}_n))}$ to remain close to ρ_n by minimizing geodesic length:

$$\mathcal{L}_{\mathrm{geo}} = \mathbb{E}\left[\sum_{s=1}^{S} \left\| \boldsymbol{x}_{\tau(s)} - \boldsymbol{x}_s^{(0)} \right\|^2\right], \tag{2}$$

with the Gumbel–softmax trick ensuring differentiability with respect to mixture weights and mode locations. The training schedule is described in Appendix A. Training alternates between updates to the flow model and updates to the mixture-conditioned base distribution, coordinated by a training planner P. Specifically, P splits training into three phases: (i) a warm-up phase in which only the flow model is optimized; (ii) an alternating phase in which multiple flow updates are interleaved with occasional base updates; and (iii) a cool-down phase in which the base is frozen and training continues by refining only the flow model. During warm-up and alternating phases, dropout in $H_{\boldsymbol{\Theta}}$ and $H_{\boldsymbol{p}}$ is enabled, while it is disabled during cool-down.

3.2 Inference

At test time, given a novel descriptor $\boldsymbol{y}_{test}$, SP-FM predicts the base distribution $\mu_{\mathrm{GMM}(h_{\boldsymbol{\Theta}}(\boldsymbol{y}_{test}), h_{\boldsymbol{p}}(\boldsymbol{y}_{test}))}$ and evolves it with $v(., t; \boldsymbol{y}_{test}),\ t \in [0, 1]$ to generate samples approximating ρ_{test}.

4 Generalization Analysis

The Wasserstein manifold of probability measures is infinite-dimensional, hindering the adoption of theoretical generalization guarantees for finite-dimensional spaces. To circumvent this, we assume that the target distribution ρ is itself a Gaussian mixture model (GMM) with J components, $\rho = \rho_{GMM(\boldsymbol{\Gamma}, \boldsymbol{q})}$, where $\boldsymbol{\Gamma} \in \mathbb{R}^{J \times D}$ are fixed mode locations, $\boldsymbol{q} \in \mathbb{S}^{J-1}$ are mixture weights, and all components share a fixed variance. This setting still captures highly general distributions: as $J \to \infty$ and $\boldsymbol{\Gamma}$ forms a sufficiently fine grid covering $\mathbb{R}^D$, the family $GMM(\boldsymbol{\Gamma}, \boldsymbol{q})$ is a universal approximator of smooth densities (Goodfellow et al., 2016). Thus, assuming ρ is GMM-induced is a broad and flexible modelling assumption. In this formulation, prediction reduces to estimating the mixture weights $\boldsymbol{q} \in \mathbb{S}^{J-1}$ from the population descriptor $\boldsymbol{y}$.

A naive solution is to directly learn a predictor mapping $\boldsymbol{y} \mapsto \boldsymbol{q}$. However, this approach suffers from poor generalization as J grows, since the dimensionality of $\boldsymbol{q}$ directly affects the generalization. SP-FM instead takes a structured approach: rather than predicting $\boldsymbol{q}$ directly, it (i) predicts the projection of the target distribution onto the subspace of GMMs with I components (possibly, $I \ll J$), and (ii) learns the velocity field that transports this projection to the target distribution.

Besides the aforementioned assumptions, in the theoretical analysis we assume that this 2-step algorithm uses mixture Wasserstein distance (Delon & Desolneux, 2020; Bing et al., 2022) and the flow model induced by it, as opposed to the commonly used 2-Wasserstein distance. In the latter case, the pushforward function of the flow model can map its input points to any arbitrary point in $\mathbb{R}^D$. While in the former case, a discrete optimal transport problem is solved between the source and target GMMs to determine, intuitively, how the modes of the source GMM should be moved or split to create the target GMM, and that fully determines the flow. In this sense, the latter metric and flow model are a generalisation of the former. Nonetheless, this analysis highlights the importance of conditioning the base distribution on population descriptor $\boldsymbol{y}$, which we experimentally validate in Sec. 6.

In this section we only summarize the key theoretical findings. Details are provided in Appendix B.

Statement 4.1 (Summary of theoretical findings). Recall that $I, J, D \in \mathbb{N}$ are respectively the number of modes in the base and target GMM distributions, and the data dimension.

- §1 In the testing phase, the velocity field $\boldsymbol{V} \in \mathbb{R}^{I\times J}$ is identified up to $J - ID$ degrees of freedom. In particular, if $I \geq \lceil J/D \rceil$, then $\boldsymbol{V}$ is uniquely identified.
- §2 If $I = 1$, the problem reduces to predicting the J mixture weights $\boldsymbol{q}$ directly from $\boldsymbol{y}$, corresponding to the common baseline of fixing the base distribution to $\mathcal{N}(\mathbf{0}_D, \boldsymbol{I}_{D\times D})$ and explaining its poor generalization.

Essentially, Statement 4.1 suggests that increasing the number I of modes of the base distribution by one, reduces degrees of freedom of the flow model by the data dimension D, thereby facilitating the prediction and identification of the flow model in the testing phase. In Appendix H the discrepancy between mixture Wasserstein and Wasserstein distances is analysed and quantified.

5 Experiments

We evaluate on one synthetic benchmark and six real perturbation screening datasets, spanning four single-cell transcriptomics datasets (Norman, Combo–SciPlex, iAstrocytes, Replogle–Nadig) and two image-based morphology datasets (BBBC021, RxRx1). In these screens, cells are exposed to an external perturbation (e.g., a small-molecule drug or drug combination, or a genetic perturbation such as CRISPR/CRISPRi), inducing a condition-specific shift in cellular state. In transcriptomic datasets, each condition corresponds to a population of single-cell gene-expression profiles represented as a cell-by-gene matrix. In imaging datasets, each condition corresponds to a population of microscopy images of cells after intervention. Dataset statistics (cells/images, perturbations, and cell types) are provided in Appendix Table 7.

5.1 Synthetic Data

Table 1: **Synthetic letter–rotation benchmark.** Models are trained on even-indexed rotations and evaluated on held-out odd-indexed rotations (validation: `S`; test: unseen letters `W` and `Y`). We report both Generated→Target and Source→Target distances. The EM-GMM baseline has no transport model, so its Source→Target distance equals its Generated→Target distance. SP-FM (I=10) predicts mode locations via $h_{\boldsymbol{\Theta}}$, while SP-FM* (I=60) uses free learnable mode locations initialized from rendered training samples.

Dataset	Model	Generated vs. Target				Source vs. Target			
		MMD ($\times 10^{-4}$) ↓	W1 ↓	W2 ↓	ED ($\times 10^{-2}$) ↓	MMD ($\times 10^{-4}$) ↓	W1 ↓	W2 ↓	ED ($\times 10^{-2}$) ↓
Letter S (Val)	CFM	162.7 ± 6.9	0.587 ± 0.020	0.707 ± 0.027	3.732 ± 0.271	824.9 ± 13.8	1.067 ± 0.008	1.190 ± 0.007	15.341 ± 0.464
	CFM (I=1)	149.9 ± 3.3	0.569 ± 0.012	0.689 ± 0.017	3.543 ± 0.331	3330.5 ± 16.7	1.803 ± 0.005	1.921 ± 0.005	48.597 ± 0.273
	SP-FM* (I=60)	98.6 ± 1.4	0.474 ± 0.003	0.574 ± 0.001	2.183 ± 0.061	198.6 ± 0.8	0.646 ± 0.001	0.739 ± 0.001	3.471 ± 0.025
	SP-FM (I=10)	**95.9 ± 3.9**	**0.467 ± 0.004**	**0.567 ± 0.004**	**2.052 ± 0.069**	**111.4 ± 8.3**	**0.500 ± 0.016**	**0.593 ± 0.018**	**2.225 ± 0.167**
Letter W (Test)	CFM	180.9 ± 8.1	0.618 ± 0.014	0.728 ± 0.015	3.770 ± 0.139	608.8 ± 4.9	0.897 ± 0.006	1.015 ± 0.007	10.775 ± 0.076
	CFM (I=1)	168.4 ± 3.4	0.594 ± 0.007	0.700 ± 0.007	3.365 ± 0.218	3109.3 ± 19.9	1.606 ± 0.010	1.720 ± 0.011	41.252 ± 0.327
	GMM (I=600)	893.1 ± 38.7	1.131 ± 0.030	1.320 ± 0.038	12.205 ± 0.096	–	–	–	–
	SP-FM* (I=60)	127.9 ± 13.1	0.532 ± 0.020	0.631 ± 0.022	2.592 ± 0.358	182.5 ± 6.2	0.610 ± 0.007	0.713 ± 0.006	3.224 ± 0.144
	SP-FM (I=10)	**123.9 ± 7.4**	**0.523 ± 0.015**	**0.623 ± 0.018**	**2.310 ± 0.146**	**134.1 ± 7.0**	**0.542 ± 0.017**	**0.640 ± 0.021**	**2.439 ± 0.072**
Letter Y (Test)	CFM	233.3 ± 19.7	0.628 ± 0.016	0.742 ± 0.018	4.603 ± 0.385	602 ± 12.8	1.061 ± 0.005	1.241 ± 0.005	14.122 ± 0.083
	CFM (I=1)	193.9 ± 9.3	0.570 ± 0.009	0.672 ± 0.010	3.681 ± 0.130	2555.3 ± 10.6	1.546 ± 0.009	1.775 ± 0.009	38.991 ± 0.320
	GMM (I=600)	742.3 ± 52.8	0.899 ± 0.032	1.051 ± 0.042	9.601 ± 0.340	–	–	–	–
	SP-FM* (I=60)	**179.9 ± 1.5**	**0.543 ± 0.004**	**0.640 ± 0.006**	**3.201 ± 0.046**	**407.9 ± 22.5**	**0.934 ± 0.030**	**1.086 ± 0.031**	**8.385 ± 0.669**

Table 2: Single-cell perturbation prediction benchmarks (mean ± std over four runs; lower is better).

Dataset	Model	Metrics (↓)			
		MMD	W_1	W_2	E-dist
Norman	CFM	.199 ± .009	30.45 ± .24	467.8 ± 8.1	16.2 ± .3
	SP-FM	**.192** ± .004	**30.08**** ± .28	**456.3***** ± 9.1	**15.9** ± .3
Combo	CFM	.141 ± .005	26.01 ± .46	344.8 ± 12	9.0 ± .3
	SP-FM	**.129***** ± .006	**25.64** ± .18	**335.3** ± 4.8	**8.5**** ± .2
iAstros	CFM	.163 ± .003	27.12 ± .11	369.7 ± 3.0	13.2 ± .1
	SP-FM	**.135***** ± .004	**27.06** ± .23	**367.6** ± 6.5	**12.6*** ± .1
Jurkat	CFM	.116 ± .006	52.92 ± .20	1408 ± 11	28.0 ± .3
	SP-FM	**.095**** ± .003	**52.87*** ± .22	**1405*** ± 12	**27.3**** ± .4

Table 3: Image-based morphology benchmarks on unseen drugs (mean ± std; lower is better).

Dataset	Method	Metrics (↓)			
		MMD	W_1	W_2	E-dist
BBBC021	M-Diff	.100 ± .000	32.3 ± .01	492 ± 7.2	3.27 ± .30
	CFM	.250 ± .001	108 ± .29	5904 ± 32	18.9 ± .09
	GMFlow	.262 ± .015	111 ± 5.6	6512 ± 974	23.3 ± 6.9
	PriorGrad	.060 ± .001	32.7 ± .1	551 ± 4	3.53 ± .08
	SP-FM	**.044***** ± .002	**29.5***** ± .73	**443***** ± 24	**2.40***** ± .09
RxRx1	M-Diff	.060 ± .006	31.3 ± .64	503 ± 13	2.21 ± .49
	CFM	.253 ± .001	119 ± 1.2	7078 ± 43	20.6 ± .31
	GMFlow	.252 ± .000	118 ± .03	6995 ± 4	20.4 ± .01
	PriorGrad	.043 ± .002	32.7 ± .2	542 ± 6	2.47 ± .12
	SP-FM	**.031***** ± .001	**30.9***** ± .12	**483***** ± 4.7	**1.67***** ± .03

Following Meta Flow Matching (Atanackovic et al., 2025), we construct a synthetic benchmark where each condition corresponds to a rotated letter silhouette. For each alphabet character we generate $R{=}20$ rotations; each (letter, rotation) pair defines one condition with descriptor $\boldsymbol{y}_i$, formed by concatenating a one-hot letter code with the normalized rotation angle. The target population ρ_i is obtained by sampling uniformly from the foreground pixels of the silhouette.

For SP-FM, we consider two implementations. In the first ($I{=}10$), both the mode locations $h_{\boldsymbol{\Theta}}(\boldsymbol{y}_i)$ and the mixture weights $h_{\boldsymbol{p}}(\boldsymbol{y}_i)$ are predicted from the descriptor. In the second ($I{=}60$), I is set equal to the number of conditions and the GMM mode locations are implemented as free learnable parameters, each initialized from a single rendered sample of its corresponding condition, while the mixture weights are still predicted by $h_{\boldsymbol{p}}(\boldsymbol{y}_i)$. The velocity field $v(\cdot, t; \boldsymbol{y}_i)$ is then trained using the flow-matching objectives in Eqs. 1–2 to transport the resulting base distribution toward ρ_i. As a baseline, vanilla conditional flow matching (CFM) uses a fixed isotropic Gaussian base $\mathcal{N}(\mathbf{0}, \mathbf{I})$.

We study rotation-interpolation generalization by training on the 10 even-indexed rotation angles (uniformly spaced in $[0, 2\pi]$) and evaluating on the 10 held-out odd-indexed angles. We validate on one letter (reported for `S`) and test on two letters (`W`, `Y`); implementation details and additional validation-letter results are in Appendix C.2. Results of the 1-Wasserstein (W1), 2-Wasserstein (W2), Energy Distance (ED), and the maximum-mean discrepancy (MMD) distributional distances are reported in Table 1 with qualitative samples in Figure 2. We further verify our method on a harder blocked-rotation split with larger gaps between seen and unseen rotations, where SP-FM continues to outperform CFM across all metrics (Appendix C.2).

We compare SP-FM to: (i) **CFM** (fixed Gaussian base); (ii) a **conditional GMM (EM)** baseline with condition-dependent mixture weights but no learned transport; and (iii) **SP-FM (**$I{=}1$), restricting the base to a single learnable Gaussian. For flow models, we select checkpoints by validation W2 and report all metrics at the selected checkpoint.

SP-FM outperforms all baselines on unseen rotations in both validation and test. The CFM $I{=}1$ ablation underperforms SP-FM, indicating that a learnable unimodal base alone is insufficient, while the EM-GMM baseline underperforms SP-FM, indicating that adapting the base without learning transport is insufficient. SP-FM also improves *source vs. target* distances, providing direct evidence that the learned base reduces mismatch prior to transport. We further compare against GMFlow (Chen et al., 2025) and PriorGrad (Lee et al., 2022). SP-FM achieves the lowest distances on all four metrics (Table 10).

SP-FM remains effective with a less informative descriptor: with the letter identity removed and only the rotation retained, it still outperforms CFM (MMD ($\times 10^{-4}$): 215.0 vs. 270.8; W_1: 0.694 vs. 0.773; W_2: 0.799 vs. 0.887; ED ($\times 10^{-2}$): 3.564 vs. 5.441). On the harder blocked-rotation split (Table 8), SP-FM also outperforms CFM, but by a reduced margin. As expected, a less informative descriptor or a more distant unseen condition makes the predicted source less well matched to the target.

5.2 Predicting single-cell perturbation responses to unseen perturbations

We evaluate whether SP-FM can predict gene-expression responses for perturbations not observed during training. Table 7 (Appendix) summarizes the datasets, including those with single and combinatorial perturbations where cells are treated with two perturbations at the same time (Combo-Sciplex (Lotfollahi et al., 2023), Norman (Norman et al., 2019)).

Table 4: Single cell transcriptomics benchmarks. Distributional metrics are computed in PCA-50 space; lower is better. Runtime is reported in minutes.

Dataset	Method	Distribution Metrics (↓)				Runtime (↓)
		E-dist	MMD	W_1	W_2	(min)
Adamson	CellFlow+SP-FM	**0.052*** ± .052	**0.066*** ± .037	**0.328*** ± .139	**0.139*** ± .178	34 ± 1
	CellFlow (Van.)	0.056 ± .053	0.069 ± .037	0.344 ± .145	0.159 ± .183	**27** ± 1
	GEARS	0.138 ± .015	0.504 ± .125	0.439 ± .027	0.160 ± .022	62 ± 2
	scGPT	0.118 ± .016	0.463 ± .095	0.412 ± .030	0.141 ± .023	120 ± 49
Norman	CellFlow+SP-FM	**0.039** ± .036	**0.056*** ± .030	**0.257*** ± .108	**0.085** ± .119	47 ± 1
	CellFlow (Van.)	0.043 ± .040	0.064 ± .031	0.274 ± .111	0.102 ± .137	54 ± 3
	GMFlow	**0.039** ± .030	0.059 ± .040	0.262 ± .060	0.088 ± .080	–
	PriorGrad	0.069 ± .010	0.074 ± .070	0.326 ± .090	0.167 ± .270	–
	SP-FM+Meta FM	0.100 ± .107	0.067 ± .065	0.612 ± .162	0.224 ± .118	–
	GEARS	0.107 ± .018	0.706 ± .149	0.328 ± .038	0.094 ± .024	**29** ± 1
	scGPT	0.091 ± .015	0.645 ± .119	0.310 ± .035	**0.085** ± .022	111 ± 52
ComboSciplex	CellFlow+SP-FM	**0.104*** ± .111	**0.103*** ± .076	**0.458*** ± .259	**0.271*** ± .378	40 ± 8
	CellFlow (Van.)	0.119 ± .103	0.139 ± .070	0.519 ± .226	0.301 ± .363	**6** ± 0
	GMFlow	0.125 ± .042	0.140 ± .028	0.525 ± .069	0.316 ± .116	–
	PriorGrad	0.162 ± .035	0.141 ± .024	0.542 ± .057	0.431 ± .100	–
	SP-FM+Meta FM	0.243 ± .065	0.141 ± .029	0.784 ± .069	0.414 ± .117	–
Replogle Jurkat	CellFlow+SP-FM	**0.808** ± .044	**0.452** ± .039	**1.469** ± .095	**2.975*** ± .303	43 ± 4
	CellFlow (Van.)	0.825 ± .043	0.464 ± .029	1.503 ± .061	3.112 ± .327	**11** ± 0
Replogle K562	CellFlow+SP-FM	**0.039*** ± .029	**0.068*** ± .030	**0.264*** ± .103	**0.091*** ± .105	55 ± 2
	CellFlow (Van.)	0.045 ± .035	0.079 ± .033	0.283 ± .110	0.111 ± .119	**26** ± 14
	GEARS	0.197 ± .037	0.558 ± .104	0.608 ± .096	0.441 ± .148	117 ± 43
	scGPT	0.169 ± .033	0.478 ± .081	0.564 ± .096	0.378 ± .137	80 ± 20

Bold indicates the best mean performance. * indicates statistical significance after Benjamini–Hochberg correction ($q < 0.05$).

Table 5: Relative distance between source and target distributions for SP-FM and OT-CFM. Positive values indicate that the SP-FM base is closer to the target.

Dataset	Train		Test	
	W_1	MMD	W_1	MMD
ComboSciplex	+62.4%	+52.5%	+63.4%	+53.8%
Norman	+69.8%	+28.5%	+67.5%	+18.9%
Jurkat	+63.5%	+44.9%	+63.1%	+45.7%
BBBC021	+80.8%	+83.3%	+80.8%	+81.8%
RxRx1	+81.9%	+85.9%	+81.9%	+85.5%

Table 6: Inference-time ablation of the learned base. The SP-FM velocity field is fixed while the learned GMM is replaced by $\mathcal{N}(0, I)$.

Base	MMD	W_1	W_2	E-dist
BBBC021				
GMM (SP-FM)	**.044**	**29.5**	**444**	**2.42**
$\mathcal{N}(0, I)$	.306	164.6	13688	36.4
CFM	.257	111.0	6208	20.5
RxRx1				
GMM (SP-FM)	**.031**	**30.9**	**483**	**1.67**
$\mathcal{N}(0, I)$	.288	149.8	11355	30.5
CFM	.253	118.7	7079	20.6

For each dataset, we create four random splits holding out 70% of perturbations from training and evaluating on the held-out perturbations. For iAstrocytes and Replogle–Nadig (Replogle et al., 2022), we filter to conditions with at least 300 cells to ensure reliable empirical targets. The iAstrocytes dataset is not currently publicly available. We report distributional distances between generated and empirical populations (W1, W2, MMD, and Energy Distance) computed on gene reconstructed space. Across all single-cell perturbation benchmarks, SP-FM consistently improves alignment over CFM (Table 2).

5.3 SP-FM IMPROVES CELLFLOW AND OUTPERFORMS BASELINES

CellFlow (Klein et al., 2025) is an ot-cfm (Tong et al., 2023) model that maps source distributions to the perturbed cell distribution conditioned on the perturbation descriptor, where the source distribution is the empirical distribution of unperturbed control cells. We implement an SP-FM variant that replaces the fixed control-cell source with a learnable, condition-dependent GMM base distribution, allowing the source to adapt to the perturbation context. The details of the implementation and parameters are in Appendix D. We benchmark against GEARS (Roohani et al., 2024), which uses a gene–gene interaction graph together with a graph neural network to predict post-perturbation expression, scGPT (Cui et al., 2024), a large foundation model pre-trained on the CELLxGENE corpus, and the aforementioned Meta-FM (Atanackovic et al., 2025), PriorGrad (Lee et al., 2022), and GMFlow (Chen et al., 2025). Since MFM cannot generate output from pure noise, we implement it as an addition to SP-FM, where the MFM base distribution encoder GNN provides an encoding of the base distribution. PriorGrad needs to set the prior's mean, which in our application is not obtainable in the testing phase for an unseen population. To do so, given test population descriptor, we find the closest training set population, and set the prior's mean as the mean of that population.

Combosciplex only contains chemical perturbations. Jurkat's gene assay panel is far smaller than either model's vocabulary; very few perturbed targets are represented. We therefore did not use GEARS and scGPT baselines on these datasets. Distributional metrics are computed in the 50-dimensional PCA space fitted on the full dataset, following the standards proposed by CellFlow's original paper. Extending CellFlow with SPFM consistently improves performance, despite not relying on control cells for the generation process. This combination, as shown in Table 4, also beats every other baseline.

5.4 PREDICTING IMAGE-BASED MORPHOLOGICAL RESPONSES FOR UNSEEN DRUG PERTURBATIONS

We also evaluate SP-FM on high-content imaging benchmarks, training on a subset of compounds/siRNAs and predicting phenotypes for unseen interventions from their descriptors at test time.

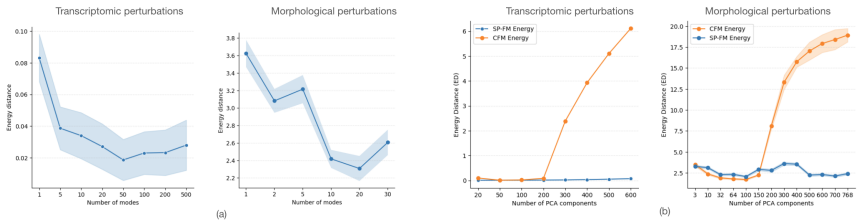


Figure 3: **Ablations.** (a) Energy distance as a function of the number of GMM modes. (b) Energy distance as a function of PCA dimensionality.

We use BBBC021 (Caie et al., 2010) (25 train compounds, 9 test) following prior work (Palma et al., 2025), and RxRx1 (Sypetkowski et al., 2023) (853 train siRNAs, 213 test). We compare against vanilla CFM. We generate in feature space by encoding images with DINOv2 (Oquab et al., 2024) into 768-D embeddings, and compute all metrics (MMD, W1, W2, ED) in the same embedding space for SP-FM, CFM and MorphoDiff. SP-FM achieves consistently lower distances in both datasets (Table 3), indicating improved recovery of unseen phenotypic responses.

6 Ablations

We analyze key design choices in SP-FM on transcriptomic (iAstrocytes) and morphological (BBBC021) benchmarks. **(i)Number of mixture components.** We vary the number of modes I in the Gaussian-mixture base and measure Energy Distance on held-out perturbations. Increasing I improves performance up to a dataset-dependent optimum, after which gains saturate or slightly degrade (Figure 3a), consistent with over-parameterization and harder optimization. Overall, the results support I as an effective complexity knob for SP-FM, as anticipated by our theoretical analysis summarized in Statement 4.1. **(ii)Robustness to data dimensionality.** We train SP-FM and CFM on PCA embeddings with increasing dimensionality and evaluate Energy Distance on held-out perturbations. CFM degrades substantially as dimensionality increases, whereas SP-FM remains stable across the tested range (Figure 3b), indicating that conditioning the base distribution mitigates high-dimensional mismatch. We hypothesize that this robustness is due to the fact that, since in SP-FM the base distribution is close to the target, the decoder flow field respects the dimension of target data manifold, while when the base distribution equals $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$, the flow field is inevitably dispersed over $\mathbb{R}^D$ near $t = 0$.

To further isolate the improvement induced by the learned conditional base distribution itself, we measure the distance of learned condition-dependent base distributions from the empirical target distributions. We see that SP-FM's base distribution is generally closer to the target distribution than a simple Gaussian base distribution, as reported in Tab.5. We also substitute the trained SP-FM's GMM with a standard-normal isotropic Gaussian distribution at inference time, while keeping the learned velocity field unchanged. We then re-evaluate the model. Tab.6 reports the results averaged over three inference seeds (the standard deviation is negligible and therefore omitted). Replacing the learned base with a standard Gaussian substantially degrades performance across all metrics and both datasets, highlighting the importance of learning the conditional base distribution in SP-FM.

7 Limitations and Future Work

SP-FM relies on informative condition descriptors (e.g., gene embeddings or chemical fingerprints); noisy or incomplete metadata may limit performance. Our experiments focus on single perturbations and small combinations, and extending to higher-order combinations, dose–response, and temporal dynamics remains open. Finally, our generalization analysis assumes smooth transport fields and may not capture highly heterogeneous biological settings. Future work will address richer multimodal descriptors and extensions to interaction- and time-dependent responses.

AI USE STATEMENT

In this work, we did not use generative AI tools for tasks requiring disclosure, and the remaining required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools to summarize and analyse existing literature, edit the manuscript for grammar and spelling, and assist with software debugging. We have reviewed and verified all AI-assisted work. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work uses publicly available benchmark datasets and does not involve the collection of new human-subject data. We are unaware of any additional ethical concerns specific to this work beyond those associated with the responsible use of machine learning methods and the original datasets.

REPRODUCIBILITY STATEMENT

We provide implementation details, experimental settings, and evaluation procedures in the main text and appendices to facilitate reproduction of our results. All datasets are described and appropriately referenced; all except iAstrocytes are publicly available, with their original sources appropriately referenced. Additional implementation details, hyperparameters, and experimental configurations are provided in the appendix and supplementary material. The source code used to produce the results will be made publicly available upon publication.

REFERENCES

Abhinav K. Adduri, Dhruv Gautam, Beatrice Bevilacqua, Mohsen Naghipourfar, Alishba Imran, Rohan Shah, Noam Teyssier, Rishi Verma, Christopher Carpenter, Basak Eraslan, Francis Chalissery, Rajesh Ilango, Vishvak Subramanyam, Chiara Ricci-Tam, Sanjay Nagaraj, Aidan Winters, Mingze Dong, Stefanie Fellinger, Adam Krejci, Tilmann Burckstummer, Sravya Tirukkovular, Jeremy Sullivan, Brian S. Plosky, Nicholas D. Youngblut, Jure Leskovec, Luke A. Gilbert, Silvana Konermann, Patrick D. Hsu, Alexander Dobin, Dave P. Burke, Hani Goodarzi, and Yusuf H. Roohani. Predicting cellular responses to perturbation across diverse contexts with STATE. *Cell*, 189(19):5914–5931.e20, 2026. doi: 10.1016/j.cell.2026.07.052.

Md Atik Ahamed, Qiang Ye, and Qiang Cheng. MolSnap: Snap-fast molecular generation with latent variational mean flow. *arXiv preprint arXiv:2508.05411*, 2025. doi: 10.48550/arXiv.2508.05411. URL https://arxiv.org/abs/2508.05411.

Constantin Ahlmann-Eltze, Wolfgang Huber, and Simon Anders. Deep-learning-based gene perturbation effect prediction does not yet outperform simple linear baselines. *Nature Methods*, 22: 1657–1661, 2025. doi: 10.1038/s41592-025-02772-6.

Lazar Atanackovic, Xi Zhang, Brandon Amos, Mathieu Blanchette, Leo J. Lee, Yoshua Bengio, Alexander Tong, and Kirill Neklyudov. Meta flow matching: Integrating vector fields on the wasserstein manifold. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/ebdb990471f653dffb425eff03c7c980-Abstract-Conference.html.

Jacob Bamberger, Iolo Jones, Dennis Duncan, Michael M. Bronstein, Pierre Vandergheynst, and Adam Gosztolai. Carré du champ flow matching: better quality-generalisation tradeoff in generative models. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/dbc9f9d486bd1714896111b0fcd78cd1-Abstract-Conference.html.

Jean-David Benamou and Yann Brenier. A computational fluid mechanics solution to the Monge–Kantorovich mass transfer problem. *Numerische Mathematik*, 84(3):375–393, 2000.

Michael Bereket and Theofanis Karaletsos. Modelling cellular perturbations with the sparse additive mechanism shift variational autoencoder. In *Advances in Neural Information Processing Systems*, 2023.

Quentin Bertrand, Anne Gagneux, Mathurin Massias, and Rémi Emonet. On the closed-form of flow matching: Generalization does not arise from target stochasticity. In *Advances in Neural Information Processing Systems*, 2025. doi: 10.52202/085713-0289.

Xin Bing, Florentina Bunea, and Jonathan Niles-Weed. Estimation and inference for the wasserstein distance between mixing measures in topic models. *arXiv preprint arXiv:2206.12768*, 2022. URL https://arxiv.org/abs/2206.12768.

Charlotte Bunne, Stefan G. Stark, Gabriele Gut, Jacobo Sarabia del Castillo, Mitch Levesque, Kjong-Van Lehmann, Lucas Pelkmans, Andreas Krause, and Gunnar Rätsch. Learning single-cell perturbation responses using neural optimal transport. *Nature Methods*, 20:1759–1768, 2023. doi: 10.1038/s41592-023-01969-x.

Peter D. Caie, Rebecca E. Walls, Alexandra Ingleston-Orme, Sandeep Daya, Tom Houslay, Rob Eagle, Mark E. Roberts, and Neil O. Carragher. High-content phenotypic profiling of drug response signatures across distinct cancer cells. *Molecular Cancer Therapeutics*, 9(6):1913–1926, 2010. doi: 10.1158/1535-7163.MCT-09-1148.

Hansheng Chen, Kai Zhang, Hao Tan, Zexiang Xu, Fujun Luan, Leonidas Guibas, Gordon Wetzstein, and Sai Bi. Gaussian mixture flow matching models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 9783–9802. PMLR, 2025. URL https://proceedings.mlr.press/v267/chen25cl.html.

Haotian Cui, Chloe Wang, Hassaan Maan, Kuan Pang, Fengning Luo, Nan Duan, and Bo Wang. scGPT: toward building a foundation model for single-cell multi-omics using generative AI. *Nature Methods*, 21(8):1470–1480, 2024. doi: 10.1038/s41592-024-02201-0.

Julie Delon and Agnès Desolneux. A wasserstein-type distance in the space of gaussian mixture models. *SIAM Journal on Imaging Sciences*, 13(2):936–970, 2020. doi: 10.1137/19M1301047.

Atray Dixit, Oren Parnas, Biyu Li, Jenny Chen, Charles P. Fulco, Livnat Jerby-Arnon, Nemanja D. Marjanovic, Danielle Dionne, Tyler Burks, Raktima Raychowdhury, Britt Adamson, Thomas M. Norman, Eric S. Lander, Jonathan S. Weissman, Nir Friedman, and Aviv Regev. Perturb-Seq: Dissecting molecular circuits with scalable single-cell RNA profiling of pooled genetic screens. *Cell*, 167(7):1853–1866.e17, 2016. doi: 10.1016/j.cell.2016.11.038.

Rémi Flamary, Cédric Vincent-Cuaz, Nicolas Courty, Alexandre Gramfort, Oleksii Kachaiev, Huy Quang Tran, Laurène David, Clément Bonet, Nathan Cassereau, Théo Gnassounou, Eloi Tanguy, Julie Delon, Antoine Collas, Sonia Mazelet, Laetitia Chapel, Tanguy Kerdoncuff, Xizheng Yu, Matthew Feickert, Paul Krzakala, Tianlin Liu, Eduardo Fernandes Montesuma, Nathan Neike, Baptiste Genest, David Coeurjolly, Thibaut Germain, Sienna O'Shea, Marco Corneli, and Ferdinand Genans. Pot python optimal transport, 7 2026. URL https://github.com/PythonOT/POT.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

Leon Hetzel, Simon Boehm, Niki Kilbertus, Stephan Günnemann, Mohammad Lotfollahi, and Fabian J. Theis. Predicting cellular responses to novel drug perturbations at a single-cell resolution. In *Advances in Neural Information Processing Systems*, volume 35, pp. 26711–26722, 2022.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv*, 2022. URL https://doi.org/10.48550/arXiv.2207.12598. Preprint.

Dominik Klein, Jonas Simon Fleck, Daniil Bobrovskiy, Lea Zimmermann, Sören Becker, Alessandro Palma, Leander Dony, Alejandro Tejada-Lapuerta, Guillaume Huguet, Hsiu-Chuan Lin, Nadezhda Azbukina, Fátima Sanchís-Calleja, Theo Uscidda, Artur Szalata, Manuel Gander, Aviv Regev, Barbara Treutlein, J. Gray Camp, and Fabian J. Theis. CellFlow enables generative single-cell phenotype modeling with flow matching. *bioRxiv*, 2025. doi: 10.1101/2025.04.11.648220. URL https://doi.org/10.1101/2025.04.11.648220. Preprint.

Sang-gil Lee, Heeseung Kim, Chaehun Shin, Xu Tan, Chang Liu, Qi Meng, Tao Qin, Wei Chen, Sungroh Yoon, and Tie-Yan Liu. PriorGrad: Improving conditional denoising diffusion models with data-dependent adaptive prior. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=_BNiN4IjC5.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Mohammad Lotfollahi, Anna Klimovskaia Susmelj, Carlo De Donno, Leon Hetzel, Yuge Ji, Ignacio L. Ibarra, Sanjay R. Srivatsan, Mohsen Naghipourfar, Riza M. Daza, Beth Martin, Jay Shendure, José L. McFaline-Figueroa, Pierre Boyeau, F. Alexander Wolf, Nafissa Yakubova, Stephan Günnemann, Cole Trapnell, David Lopez-Paz, and Fabian J. Theis. Predicting cellular responses to complex perturbations in high-throughput screens. *Molecular Systems Biology*, 19(6): e11517, 2023. doi: 10.15252/msb.202211517.

Zeinab Navidi, Jun Ma, Esteban A. Miglietta, Le Liu, Anne E. Carpenter, Beth A. Cimini, Benjamin Haibe-Kains, and Bo Wang. MorphoDiff: Cellular morphology painting with diffusion models. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/1ebccf36966b44873ca65f7644a34c33-Abstract-Conference.html.

Thomas M. Norman, Max A. Horlbeck, Joseph M. Replogle, Alex Y. Ge, Albert Xu, Marco Jost, Luke A. Gilbert, and Jonathan S. Weissman. Exploring genetic interaction manifolds constructed from rich single-cell phenotypes. *Science*, 365(6455):786–793, 2019. doi: 10.1126/science.aax4438.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. URL https://mlanthology.org/tmlr/2024/oquab2024tmlr-dinov2/.

Alessandro Palma, Fabian J. Theis, and Mohammad Lotfollahi. Predicting cell morphological responses to perturbations using generative modeling. *Nature Communications*, 16:505, 2025. doi: 10.1038/s41467-024-55707-8.

Gabriel Peyré and Marco Cuturi. Computational optimal transport: With applications to data science. *Foundations and Trends in Machine Learning*, 11(5–6):355–607, 2019. doi: 10.1561/2200000073.

Joseph M. Replogle, Reuben A. Saunders, Angela N. Pogson, Jeffrey A. Hussmann, Alexander Lenail, Alina Guna, Lauren Mascibroda, Eric J. Wagner, Karen Adelman, Gila Lithwick-Yanai, Nika Iremadze, Florian Oberstrass, Doron Lipson, Jessica L. Bonnar, Marco Jost, Thomas M. Norman, and Jonathan S. Weissman. Mapping information-rich genotype-phenotype landscapes with genome-scale Perturb-seq. *Cell*, 185(14):2559–2575.e28, 2022. doi: 10.1016/j.cell.2022.05.013.

David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, 50(5):742–754, 2010. doi: 10.1021/ci100050t.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022. doi: 10.1109/CVPR52688.2022.01042.

Yusuf Roohani, Kexin Huang, and Jure Leskovec. Predicting transcriptional outcomes of novel multigene perturbations with GEARS. *Nature Biotechnology*, 42:927–935, 2024. doi: 10.1038/s41587-023-01905-6.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, 2019.

Sanjay R. Srivatsan, José L. McFaline-Figueroa, Vijay Ramani, Lauren Saunders, Junyue Cao, Jonathan Packer, Hannah A. Pliner, Dana L. Jackson, Riza M. Daza, Lena Christiansen, Fan Zhang, Frank Steemers, Jay Shendure, and Cole Trapnell. Massively multiplex chemical transcriptomics at single-cell resolution. *Science*, 367(6473):45–51, 2020. doi: 10.1126/science.aax6234. URL https://www.science.org/doi/abs/10.1126/science.aax6234.

Aravind Subramanian, Rajiv Narayan, Steven M. Corsello, David D. Peck, Ted E. Natoli, Xiaodong Lu, Joshua Gould, John F. Davis, Andrew A. Tubelli, Jacob K. Asiedu, David L. Lahr, Jodi E. Hirschman, Zihan Liu, Melanie Donahue, Bina Julian, Mariya Khan, David Wadden, Ian C. Smith, Daniel Lam, Arthur Liberzon, Courtney Toder, Mukta Bagul, Marek Orzechowski, Oana M. Enache, Federica Piccioni, Sarah A. Johnson, Nicholas J. Lyons, Alice H. Berger, Alykhan F. Shamji, Angela N. Brooks, Anita Vrcic, Corey Flynn, Jacqueline Rosains, David Y. Takeda, Roger Hu, Desiree Davison, Justin Lamb, Kristin Ardlie, Larson Hogstrom, Peyton Greenside, Nathanael S. Gray, Paul A. Clemons, Serena Silver, Xiaoyun Wu, Wen-Ning Zhao, Willis Read-Button, Xiaohua Wu, Stephen J. Haggarty, Lucienne V. Ronco, Jesse S. Boehm, Stuart L. Schreiber, John G. Doench, Joshua A. Bittker, David E. Root, Bang Wong, and Todd R. Golub. A next generation connectivity map: L1000 platform and the first 1,000,000 profiles. *Cell*, 171(6): 1437–1452.e17, 2017. doi: 10.1016/j.cell.2017.10.049.

Maciej Sypetkowski, Morteza Rezanejad, Saber Saberian, Oren Kraus, John Urbanik, James Taylor, Ben Mabey, Mason Victors, Jason Yosinski, Alborz Rezazadeh Sereshkeh, Imran Haque, and Berton Earnshaw. RxRx1: A dataset for evaluating experimental batch correction methods. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 4285–4294, 2023.

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, 2023.

TaeHo Yoon, Joo Young Choi, Sehyun Kwon, and Ernest K. Ryu. Diffusion probabilistic models generalize when they fail to memorize. In *ICML 2023 Workshop on Structured Probabilistic Inference and Generative Modeling*, 2023.

Yuhui Zhang, Yuchang Su, Chenyu Wang, Tianhong Li, Zoe Wefers, Jeffrey J. Nirschl, James Burgess, Daisy Ding, Alejandro Lozano, Emma Lundberg, and Serena Yeung-Levy. CellFlux: Simulating cellular morphology changes via flow matching. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 76111–76138. PMLR, 2025. URL https://proceedings.mlr.press/v267/zhang25bt.html.

A TRAINING SCHEDULE AND PSEUDOCODE

Algorithm 1 provides pseudocode for SP-FM training. Training alternates between updating the velocity field and updating the mixture-conditioned base distribution according to a planner P, which also controls whether dropout in $(H_{\boldsymbol{\Theta}}, H_{\boldsymbol{p}})$ is enabled.

Algorithm 1 Training SP-FM

Require: Datasets $\{(\boldsymbol{X}_n, \boldsymbol{y}_n)\}_{n=1}^N$, number of GMM modes I, fixed GMM variance σ^2, training planner P, learning rate ℓ.

Initialize parameters: $params(v(\cdot,\cdot,\cdot))$, $params(h_{\boldsymbol{\Theta}}(\cdot))$, $params(h_{\boldsymbol{p}}(\cdot))$.
$cnt_epoch \leftarrow 0, \quad cnt_iteration \leftarrow 0.$
for each training epoch **do**
  while epoch not finished **do**
    $mode_train,\ flag_settoeval_H \leftarrow P.next(cnt_epoch,\ cnt_iteration)$
    Sample a population $(\boldsymbol{X}_n, \boldsymbol{y}_n)$
    Sample $\{\boldsymbol{x}_s\}_{s=1}^S \sim \rho_n$ {S target samples}
    Sample $\{\boldsymbol{x}_s^{(0)}\}_{s=1}^S \sim \mu_{\mathrm{GMM}\left(h_{\boldsymbol{\Theta}}(\boldsymbol{y}_n),\, h_{\boldsymbol{p}}(\boldsymbol{y}_n)\right)}$ {S base samples}
    $\tau \leftarrow$ minibatch OT pairing between $\{\boldsymbol{x}_s\}_{s=1}^S$ and $\{\boldsymbol{x}_s^{(0)}\}_{s=1}^S$
    if $flag_settoeval_H$ **then**
      $H_{\boldsymbol{\Theta}}.eval();\ H_{\boldsymbol{p}}.eval()$ {disable dropout}
    else
      $H_{\boldsymbol{\Theta}}.train();\ H_{\boldsymbol{p}}.train()$ {enable dropout}
    end if
    if $mode_train$ equals 'velocity field' **then**
      $t \leftarrow \mathrm{Unif}(0,1)$
      $\mathcal{L}_{OT} \leftarrow$ compute using Eq. 1
      $params(v) \leftarrow params(v) - \ell \cdot \nabla \mathcal{L}_{OT}$
    else if $mode_train$ equals 'base distribution' **then**
      $\mathcal{L}_{geo} \leftarrow$ compute using Eq. 2
      $params(H_{\boldsymbol{\Theta}}, H_{\boldsymbol{p}}) \leftarrow params(H_{\boldsymbol{\Theta}}, H_{\boldsymbol{p}}) - \ell \cdot \nabla \mathcal{L}_{geo}$
    else
      ERROR: unknown value for $mode_train$
    end if
    $cnt_iteration \leftarrow cnt_iteration + 1$
  end while
  $cnt_epoch \leftarrow cnt_epoch + 1$
end for

B GENERALISATION ANALYSIS, DETAILS AND PROOFS

As outlined in Sec. 4, to circumvent the infinite-dimensional Wasserstein manifold of measures and to enable the adoption of theoretical generalization guarantees for finite-dimensional spaces, we assume that the target distribution ρ is itself a Gaussian mixture model (GMM) with J components, fixed mode locations that cover $\mathbb{R}^D$ as $J \to \infty$, and varying mixture weights. Moreover, recall that μ and ρ denote the base and target distributions, respectively.

Definition B.1 (Mixture Wasserstein Distance)**.** Let $\mu_{GMM(\boldsymbol{\Theta},\boldsymbol{p})}$ and $\rho_{GMM(\boldsymbol{\Gamma},\boldsymbol{q})}$ be two measures induced by their corresponding GMMs with I and J modes, mode positions $\boldsymbol{\Theta} \in \mathbb{R}^{I\times D}$ and $\boldsymbol{\Gamma} \in \mathbb{R}^{J\times D}$, and mode weights $\boldsymbol{p} \in \mathbb{S}^{I-1}$ and $\boldsymbol{q} \in \mathbb{S}^{J-1}$. The mixture Wasserstein distance between $\mu_{GMM(\boldsymbol{\Theta},\boldsymbol{p})}$ and $\mu_{GMM(\boldsymbol{\Gamma},\boldsymbol{q})}$ is defined as follows:

$$\mathcal{MW}_2^2(\mu,\rho) = \inf_{\boldsymbol{V}\in\mathbb{R}^{I\times J}} \sum_{i=1}^{I}\sum_{j=1}^{J} ||\boldsymbol{\Theta}_i - \boldsymbol{\Gamma}_j||_2^2 \; v_{ij} \; p_i \tag{3}$$

$$\text{s.t.} \sum_{j=1}^{J} v_{ij} = 1 \;\; , \;\; \forall i \in \{1,...,I\} \tag{4}$$

$$q_j = \sum_{i=1}^{I} v_{ij} \; p_i \;\; , \;\; \forall j \in \{1,....,J\} \quad \text{(boundary condition)} \tag{5}$$

$$v_{ij} \geq 0 \;\; , \;\; \forall i \in \{1,...,I\}, \; \forall j \in \{1,...,J\} \tag{6}$$

This is equivalent to discrete optimal transport between the GMMs' modes with the transportation cost determined by their mode locations. Intuitively, v_{ij} is the portion of mass of the i-th mode in the 1st GMM μ which is moved to the j-th mode in the 2nd GMM ρ.

Definition B.2 (Mixture Wasserstein Flow)**.** The matrix $\boldsymbol{V}^{I\times J}$ in Eq. 3 induces a time-dependant flow $v : \mathbb{R}^D \times [0,1] \to \mathbb{R}^D$ referred to as mixture Wasserstein flow.

Lemma B.3 (Dual of Mixture Wasserstein Distance)**.** *The dual form of the constrained optimisation of Eq. 3 is as follows:*

$$\sup_{\boldsymbol{z}\in\mathbb{R}^{I+J}} \sum_{i=1}^{I} z_i + \sum_{j=1}^{J} q_j \; z_{j+I} \tag{7}$$

$$\textit{s.t. } z_i + p_i \cdot z_{j+I} \; \leq p_i ||\boldsymbol{\Theta}_i - \boldsymbol{\Gamma}_j||_2^2 \;\; , \;\; \forall i \in \{1,...,I\}, \; j \in \{1,...,J\} \tag{8}$$

$$\{z_1, z_2, ..., z_{I+J}\} \quad \textit{unrestricted in sign , i.e. } \in \mathbb{R} \tag{9}$$

Definition B.4 (Subset sum condition)**.** In the discrete optimal transport problem of Eq. 3, the marginal measures $[p_1, ..., p_I]$ and $[q_1, ..., q_J]$ are said to meet the subset sum condition if for any proper non-empty subsets of them - denoted by $\{p_{i1}, ..., p_{i_{I'}}\}$ and $\{q_{j1}, ..., q_{j_{J'}}\}$- we have that

$$(p_{i1} + ... + p_{i_{I'}}) \neq (q_{j1} + ... + q_{j_{J'}}). \tag{10}$$

Statement B.5. In the discrete optimal transport problem of Eq. 3 assume that the marginal distributions meet the subset sum condition. Then we have that:

1. Any optimal solution of Eq. 3, $\boldsymbol{V}^* \in \mathbb{R}^{I\times J}$, has exactly $I + J - 1$ non-zero elements.
2. A dual constraint in Eq. 8 is binding if and only if the corresponding $\boldsymbol{V}^*[i,j] = v_{ij}^*$ is non-zero.
3. The dual problem of Eq. 7 has a unique optimum $\boldsymbol{z}^* \in \mathbb{R}^{I+J}$.

The importance of Statement B.5 is that - as we will prove in Statement B.7 - for any μ and ρ the subset sum condition, and hence 1, 2, and 3 explained above hold almost surely. Therefore, we can generally assume 1, 2, and 3 in Statement B.5 hold for any μ and ρ.

Statement B.6. In the discrete optimal transport problem of Eq. 3 assume that the marginal distributions meet the subset sum condition. Moreover, assume that $\boldsymbol{\Theta}$, $\boldsymbol{\Gamma}$, $\boldsymbol{p}$, and $[z_1^*, ..., z_I^*]$ in the dual form are known and fixed. Then $\boldsymbol{V}_{\geq 0}^*$ is uniquely determined, where $\boldsymbol{V}^* \in \mathbb{R}^{I\times J}$ is an optimal solution for Eq. 3, and $\boldsymbol{V}_{\geq 0}^* \in \mathbb{R}^{I\times J}$ is a matrix where the element in its i-th row and j-th column is 1 if $v_{ij}^* > 0$ and is 0 otherwise.

Proof: The constraint of Eq. 7 rewrites as

$$z_{j+I} \leq \Big[||\boldsymbol{\Theta}_i - \boldsymbol{\Gamma}_j||_2^2 - \frac{z_i}{p_i} \Big] \tag{11}$$

To maximise the objective of Eq. 7, the variables z_{j+I} should take their maximum possible value. Therefore we have that

$$z_{j+I}^* = \max_{1\leq i\leq I} \Big[||\boldsymbol{\Theta}_i - \boldsymbol{\Gamma}_j||_2^2 - \frac{z_i^*}{p_i} \Big]. \tag{12}$$

In other words, given $[z_1^*, ..., z_I^*]$ the above equation uniquely specifies the values for $[z_{I+1}^*, ..., z_{I+J}^*]$ and regardless of the probability values of the target distribution $\boldsymbol{q} \in \mathbb{R}^J$. According to Statement B.5 , in Eq. 12 equality holds if and only if $v_{ij}^* > 0$, and this determines $\boldsymbol{V}_{\geq 0}^*$. ■

The importance of Statement B.6 is that, intuitively, having predicted the closest GMM to ρ (i.e. $\boldsymbol{\Theta}$ and $\boldsymbol{p}$), having predicted the first I optimal dual values $[z_1^*, .., z_I^*] \in \mathbb{R}^I$, and having fixed the GMM modes that cover the feature space (i.e. $\boldsymbol{\Gamma}$), $\boldsymbol{V}_{\geq 0}^*$ is uniquely identified. This is important, because to obtain a predictor $\boldsymbol{V}_{\geq 0}^* \approx h_{\boldsymbol{V}_{\geq 0}^*}(\boldsymbol{y})$ where $h_{\boldsymbol{V}_{\geq 0}^*} : \mathbb{R}^{dim(\boldsymbol{y})} \rightarrow \{0, 1\}^{I \times J}$, one can instead train the following three predictors, and $h_{\boldsymbol{V}_{\geq 0}^*}(.)$ will be uniquely determined:

1. $\boldsymbol{\Theta} \approx h_{\boldsymbol{\Theta}}(\boldsymbol{y})$, where $h_{\boldsymbol{\Theta}} : \mathbb{R}^{dim(\boldsymbol{y})} \rightarrow \mathbb{R}^{I \times D}$.
2. $\boldsymbol{p} \approx h_{\boldsymbol{p}}(\boldsymbol{y})$, where $h_{\boldsymbol{p}} : \mathbb{R}^{dim(\boldsymbol{y})} \rightarrow \mathbb{S}^{I-1}$.
3. $[z_1^*, ..., z_I^*] \approx h_z(\boldsymbol{y})$, where $h_z : \mathbb{R}^{dim(\boldsymbol{y})} \rightarrow \mathbb{R}^I$

Statement B.7. Let $\mu_{GMM(\boldsymbol{\Theta}, \boldsymbol{p})}$ and $\rho_{GMM(\boldsymbol{\Gamma}, \boldsymbol{q})}$ be two measures induced by their corresponding GMMs. Then the marginal distributions $\boldsymbol{p}$ and $\boldsymbol{q}$ meet the subset sum condition almost surely. Therefore, Statements B.5 and B.6 hold almost surely.

Proof: There are $(2^I - 2) \times (2^J - 2)$ pairs of proper non-empty subsets from the marginal measures $[p_1, ..., p_I]$ and $[q_1, ..., q_J]$. Each such a pair like $\{p_{i1}, ..., p_{i_{I'}}\}$ and $\{q_{j1}, ..., q_{j_{J'}}\}$ specify the following subspace of $\mathbb{S}^{I-1} \times \mathbb{S}^{J-1}$ where the subset sum condition is violated:

$$p_{i1} + ... + p_{i_{I'}} = q_{j1} + ... + q_{j_{J'}} \tag{13}$$

Each such subspace has zero measure, so the union of these $(2^I - 2) \times (2^J - 2)$ subspaces has also zero measure. ■

Definition B.8 (Projection to subspace of GMMs with I modes)**.** Let $\rho_{GMM(\boldsymbol{\Gamma}, \boldsymbol{q})}$ be a measure induced by a GMM with modes $\boldsymbol{\Gamma} \in \mathbb{R}^{J \times D}$ and mode probabilities $\boldsymbol{q} \in \mathbb{S}^{J-1}$. The goal is to find the closest measure to $\rho_{GMM(\boldsymbol{\Gamma}, \boldsymbol{q})}$ which is induced by a GMM with I modes, where $I << J$ and in terms of mixture Wasserstein distance. This GMM $\mu_{GMM(\boldsymbol{\Theta}*, \boldsymbol{p}*)}$ is found by:

$$\boldsymbol{\Theta}^*, \boldsymbol{p}^* = \underset{\substack{\boldsymbol{\Theta} \in \mathbb{R}^{I \times D} \\ \boldsymbol{p} \in \mathbb{R}^I}}{argmin} \ \mathcal{MW}_2^2(\mu_{GMM(\boldsymbol{\Theta}, \boldsymbol{p})}, \rho_{GMM(\boldsymbol{\Gamma}, \boldsymbol{q})}) \tag{14}$$

Let's assume we have the 3 predictors $h_{\boldsymbol{\Theta}}(.)$, $h_{\boldsymbol{p}}(.)$, and $h_z(.)$ explained above, that take in the population descriptor $\boldsymbol{y}$ and predict the projection of the unseen ρ to the space of GMMs (*i.e.* $\boldsymbol{\Theta}^*$ and $\boldsymbol{p}^*$) as well the I dual solution values $[z_1^*, ..., z_I^*]$. Given these, according to Statement B.6 we can uniquely determine $I + J - 1$ elements of $\boldsymbol{V}^* \in \mathbb{R}^{I \times J}$ which are non-zero, and the rest are uniquely set to zero. Recall that the goal is to predict the flow $\boldsymbol{V}^* \in \mathbb{R}^{I \times J}$, because when it is applied to the measure $\mu_{GMM(\boldsymbol{\Theta}*, \boldsymbol{p}*)}$ it gives us the prediction for ρ. Having identified non-zero elements of $\boldsymbol{V}^*$, Statements B.9 and B.10 impose linear constraints on the identified elements, thereby improving the idenfication and prediction of $\boldsymbol{V}^*$.

Statement B.9 (Barycentre condition). For the matrix $\boldsymbol{\Theta}^* \in \mathbb{R}^{I \times D}$ in Eq. 14 we have that

$$\boldsymbol{\Theta}_i^* = \sum_{j=1}^{J} v_{ij}^* \boldsymbol{\Gamma}_j. \tag{15}$$

Proof: We compute the derivative of Eq. 3 with respect to $\boldsymbol{\Theta}_i$ and set it to zero:

$$\frac{\partial}{\partial \boldsymbol{\Theta}_i} \sum_{j=1}^{J} ||\boldsymbol{\Theta}_i - \boldsymbol{\Gamma}_j||_2^2 \; v_{ij}^* \; p_i = \tag{16}$$

$$\frac{\partial}{\partial \boldsymbol{\Theta}_i} \sum_{j=1}^{J} \left[\boldsymbol{\Theta}_i^T \boldsymbol{\Theta}_i - 2\boldsymbol{\Theta}_i^T \boldsymbol{\Gamma}_j\right] \; v_{ij}^* \; p_i = \tag{17}$$

$$\sum_{j=1}^{J} [2\boldsymbol{\Theta}_i - 2\boldsymbol{\Gamma}_j] v_{ij}^* p_i = \tag{18}$$

$$2p_i \; [\boldsymbol{\Theta}_i \cancelto{1}{\sum_{j=1}^{J} v_{ij}^*} - \sum_{j=1}^{J} v_{ij}^* \boldsymbol{\Gamma}_j] = 0 \tag{19}$$

$$\Rightarrow \boldsymbol{\Theta}_i^* = \sum_{j=1}^{J} v_{ij}^* \boldsymbol{\Gamma}_j \tag{20}$$

■

Statement B.10 (Optimality condition for mode probabilities). The following optimality condition

$$\frac{\partial}{\partial \boldsymbol{p}} \mathcal{MW}_2^2(\mu_{GMM(\boldsymbol{\Theta}^*, \boldsymbol{p})}, \rho_{GMM(\boldsymbol{\Gamma}, \boldsymbol{q})})|_{\boldsymbol{p}=\boldsymbol{p}^*} = 0 \tag{21}$$

implies that

$$\sum_{j=1}^{J} v_{ij}^* ||\boldsymbol{\Theta}_i^* - \boldsymbol{\Gamma}_j||_2^2 = const, \quad \forall i \in \{1, ..., I\}. \tag{22}$$

Proof: At the optimal point $[\boldsymbol{\Theta}^*, \boldsymbol{p}^*]$, if two mode probabilities like $p_{i_1}^*$ and $p_{i_2}^*$ are changed by $+\epsilon$ and $-\epsilon$ (where ϵ is a small positive scalar), we have that

$$\frac{change \; in \; \mathcal{MW}_2^2}{change \; in \; \boldsymbol{p}} \tag{23}$$

$$= \frac{1}{\epsilon}\Big[\sum_{j=1}^{J} ||\boldsymbol{\Theta}_{i_1}^* - \boldsymbol{\Gamma}_j||_2^2 \; v_{i_1 j}^* \; (+\epsilon)\Big] + \frac{1}{\epsilon}\Big[\sum_{j=1}^{J} ||\boldsymbol{\Theta}_{i_2}^* - \boldsymbol{\Gamma}_j||_2^2 \; v_{i_2 j}^* \; (-\epsilon)\Big] \tag{24}$$

$$= \Big[\sum_{j=1}^{J} ||\boldsymbol{\Theta}_{i_1}^* - \boldsymbol{\Gamma}_j||_2^2 \; v_{i_1 j}^*\Big] - \Big[\sum_{j=1}^{J} ||\boldsymbol{\Theta}_{i_2}^* - \boldsymbol{\Gamma}_j||_2^2 \; v_{i_2 j}^*\Big] = 0 \tag{25}$$

Therefore

$$\sum_{j=1}^{J} ||\boldsymbol{\Theta}_{i_1}^* - \boldsymbol{\Gamma}_j||_2^2 \; v_{i_1 j}^* = \sum_{j=1}^{J} ||\boldsymbol{\Theta}_{i_2}^* - \boldsymbol{\Gamma}_j||_2^2 \; v_{i_2 j}^* \qquad \forall i_1, i_2 \in \{1, ..., I\}, \;\; i_1 \neq i_2. \tag{26}$$

■

Now we are ready to state the training phase and testing phase of the learning algorithm.

Statement B.11 (Training phase). Given is a set of N GMM-induced measures $\{\rho_{GMM(\boldsymbol{\Gamma}, \boldsymbol{q}^{(n)})}\}_{n=1}^{N}$ and their corresponding population descriptors $\{\boldsymbol{y}_1, .., \boldsymbol{y}_N\}$. Note that the matrix $\boldsymbol{\Gamma} \in \mathbb{R}^{J \times D}$ are the GMM modes in the target distribution that carpet the feature space, and are kept fixed during trianing and testing, hence to superscript n for $\boldsymbol{\Gamma}$. For each dataset instance, the following vectors and matrices are obtained

1. The parameters of the closest GMM to $\rho_{GMM(\boldsymbol{\Gamma}, \boldsymbol{q}^{(n)})}$ and with I modes:

$$\boldsymbol{\Theta}^{*(n)}, \boldsymbol{p}^{*(n)} = \underset{\substack{\boldsymbol{\Theta} \in \mathbb{R}^{I \times D} \\ \boldsymbol{p} \in \mathbb{R}^{I}}}{argmin} \; \mathcal{MW}_2^2(\mu_{GMM(\boldsymbol{\Theta}, \boldsymbol{p})}, \rho_{GMM(\boldsymbol{\Gamma}, \boldsymbol{q}^{(n)})}) \tag{27}$$

2. Having obtained $\boldsymbol{\Theta}^{*(n)}$ and $\boldsymbol{p}^{*(n)}$, solve the dual form of Eq. 7 for the distributons $\mu_{GMM(\boldsymbol{\Theta}^{*(n)},\boldsymbol{p}^{*(n)})}$ and $\rho_{GMM(\boldsymbol{\Gamma}^{(n)},\boldsymbol{q}^{(n)})}$. Save the first I elements of the dual solution $[z_1^{*(n)}, ..., z_I^{*(n)}]$.

Now having obtained $\{\boldsymbol{\Theta}^{*(n)}, \boldsymbol{p}^{*(n)}, [z_1^{*(n)}, ..., z_I^{*(n)}]\}_{n=1}^N$, the following predictors are trained

1. $\boldsymbol{\Theta}^{*(n)} \approx h_{\boldsymbol{\Theta}}(\boldsymbol{y}_n)$, where $h_{\boldsymbol{\Theta}} : \mathbb{R}^{dim(\boldsymbol{y})} \to \mathbb{R}^{I\times D}$.
2. $\boldsymbol{p}^{*(n)} \approx h_{\boldsymbol{p}}(\boldsymbol{y}_n)$, where $h_{\boldsymbol{p}} : \mathbb{R}^{dim(\boldsymbol{y})} \to \mathbb{S}^{I-1}$.
3. $[z_1^{*(n)}, ..., z_I^{*(n)}] \approx h_z(\boldsymbol{y}_n)$, where $h_z : \mathbb{R}^{dim(\boldsymbol{y})} \to \mathbb{R}^I$

Statement B.12 (Testing phase). In the testing phase, a population descriptor $\boldsymbol{y}_{test}$ is given. Firstly, the following vectors/matrices are predicted:

1. $\boldsymbol{\Theta}^{(test)} \approx h_{\boldsymbol{\Theta}}(\boldsymbol{y}_{test})$.
2. $\boldsymbol{p}^{(test)} \approx h_{\boldsymbol{p}}(\boldsymbol{y}_{test})$
3. $[z_1^{(test)}, ..., z_I^{(test)}] \approx h_z(\boldsymbol{y}_{test})$

Now having obtained $\left[\boldsymbol{\Theta}^{(test)}, \boldsymbol{p}^{(test)}, [z_1^{(test)}, ..., z_I^{(test)}]\right]$, according to Statement B.6 $\boldsymbol{V}_{\geq 0}^{test}$ is uniquely identified and according to Statement B.7 almost surely has exactly $I + J - 1$ non-zero elements. In other words, at this point we are left with $I + J - 1$ variables (i.e. v_{ij}^{test}-s) that we want to identify as closely as possible. For these $I + J - 1$ variables, we have the following constraints

1. Each barycentre condition in Eq. 15 provides D linear constraints on non-zero v_{ij}^{test}-s, resulting in $I \times D$ linear constraints. Note that the matrix $\boldsymbol{\Gamma} \in \mathbb{R}^{J\times D}$ are the GMM modes in the target distribution that carpet the feature space, and are kept fixed throughout.
2. Each constraint of Eq. 22 provides a linear constraint, resulting in I more linear constraints. We add the constant scalar in Eq. 22 to our variable list, so we have $I + J$ variables to identify.

In sum, we have $I + J$ variables, $ID + I$ constraints, and hence $J - ID$ degrees of freedom. Having estimated $\boldsymbol{\Theta}^{(test)}$ and the flow $\boldsymbol{V}^{(test)}$, we can apply the latter on the former to arrive at an estimate for $\rho^{(test)}$

Statement B.13. If $I = 1$ the dual problem of Eq. 7 is ill-defined. **Proof**: If $I = 1$, according to Eq. 12, we have that

$$z_{j+I}^* = |||\boldsymbol{\Theta}_1 - \boldsymbol{\Gamma}_j||_2^2 - \frac{z_1^*}{\cancel{p_1}}\cdot{}^{1} \tag{28}$$

If we substitute this in Eq. 7

$$\sup_{\boldsymbol{z}\in\mathbb{R}^{1+J}} z_1 + \sum_{j=1}^{J} q_j \; z_{j+I} = \tag{29}$$

$$\sup_{\boldsymbol{z}\in\mathbb{R}^{1+J}} z_1 + \sum_{j=1}^{J} q_j \; \left[|||\boldsymbol{\Theta}_1 - \boldsymbol{\Gamma}_j||_2^2 - \frac{z_1}{\cancel{p_1}}{}_{1}\right] = \tag{30}$$

$$\sup_{\boldsymbol{z}\in\mathbb{R}^{1+J}} z_1 - z_1 \cancel{\sum_{j=1}^{J} q_j}^{1} + \sum_{j=1}^{J} q_j \; \left[|||\boldsymbol{\Theta}_1 - \boldsymbol{\Gamma}_j||_2^2\right] \tag{31}$$

In the last equation z_1 is cancelled out from the objective. ■

Essentially, according to Statement B.13 the described training and testing procedure fail, and this discourages the use of uni-modal Gaussian distribution as the base distribution.

Table 7: Single-cell perturbation datasets used to evaluate SP-FM.

Dataset	Cells	Perturbations	Cell type
Norman (Norman et al., 2019)	∼90k	276 CRISPR*	K562
Combo-Sciplex (Lotfollahi et al., 2023)	∼65k	31 compounds*	A549
iAstrocytes	∼150k	148 CRISPR	iAstrocytes
Replogle (Replogle et al., 2022)	∼70k	164 CRISPRi	Jurkat
Replogle–Nadig (Replogle et al., 2022)	∼75k	153 CRISPRi	K562
BBBC021 (Caie et al., 2010)	97,504 images	35 compounds	MCF7
RxRx1 (Sypetkowski et al., 2023)	170,193 images	1066 siRNA	U2OS

*single or paired perturbations.

C Datasets

C.1 Perturbation datasets summary

Table 7 provides dataset statistics for all perturbation screening benchmarks used in the main paper, including the number of cells/images, the number of perturbation conditions, and the cell type.

C.2 Synthetic Letters Dataset Details

Data generation. We construct a synthetic benchmark by rendering uppercase letters as binary silhouettes and sampling point clouds from the foreground pixels. Each condition corresponds to a (letter, rotation) pair. For a given letter, we generate $R = 20$ rotations uniformly spaced in $[0, 2\pi]$. For each condition, we create multiple replicas by rendering the same silhouette with different RGB colours; these replicas are used only to provide multiple samples from the same conditional distribution and are not included in the conditioning vector.

Point-cloud representation. Each sample is a point cloud in $\mathbb{R}^D$ with $D = 5$, consisting of 2D coordinates and RGB channels: $\boldsymbol{x} = (x, y, r, g, b)$. We extract foreground-pixel coordinates, normalize them to $[-1, 1]^2$, rescale to correct aspect ratio using the glyph's bounding box, optionally add Gaussian coordinate noise (set to 0 in our experiments), and finally rotate the point cloud by the condition's angle.

Colour replicas. For each (letter, rotation) condition, we generate C colour replicas to better approximate the underlying conditional distribution. In our experiments we use $C = 100$. To ensure that colour does not dominate the evaluation metrics, we restrict the colour variation to a narrow range by sampling only the red channel $r \sim \mathrm{Unif}(r_{\min}, r_{\max})$ with $(r_{\min}, r_{\max}) = (0.5, 0.6)$ and setting $g = b = 0$; thus colour provides mild diversity without strongly affecting distance metrics.

Condition descriptor. The conditioning vector $\boldsymbol{y}$ concatenates (i) a one-hot encoding of the letter identity and (ii) a continuous rotation scalar normalized to $[0, 1]$:

$$\boldsymbol{y} = [\mathrm{onehot}(\text{letter});\ \theta/(2\pi)].$$

Colour is not included in $\boldsymbol{y}$.

Source distribution. For conditional flow-matching training we sample the source point cloud from an isotropic Gaussian. In the our setting, each source point cloud uses a shared RGB value across all points (sampled uniformly within the same restricted colour range) while the 2D coordinates are sampled from $\mathcal{N}(0, I)$, matching the data format of the target clouds.

Train/validation/test splits (rotation interpolation). We study rotation interpolation by splitting the rotation indices: training uses even-indexed rotations, while validation and test use odd-indexed rotations. All letters appear in training (but only at even rotations). The validation set uses a designated validation letter (`S` in the main paper), evaluated on odd rotations. Test sets are formed

Table 8: Blocked-rotation synthetic benchmark. We evaluate a harder interpolation setting in which rotations are split into alternating blocks of five seen and five unseen rotations. We report both Generated/Target and Source/Target distances. Lower is better for all metrics.

Comparison	Split	Method	W1 ↓	W2 ↓	ED ↓	MMD ↓
Generated vs. Target	Val	SP-FM	**0.5929**	**0.7058**	**0.0308**	**0.0165**
	Val	CFM	0.6700	0.7907	0.0408	0.0227
	Test	SP-FM	**0.6510**	**0.7689**	**0.0329**	**0.0194**
	Test	CFM	0.7016	0.8167	0.0433	0.0246
Source vs. Target	Val	SP-FM	**0.6326**	**0.7394**	**0.0308**	**0.0176**
	Val	CFM	1.0550	1.1810	0.1497	0.0803
	Test	SP-FM	**0.6475**	**0.7648**	**0.0320**	**0.0178**
	Test	CFM	0.8800	0.9997	0.1033	0.0586

from unseen test letters (`W` and `Y` in the main paper), also evaluated on odd rotations. Additional results with alternative validation/test letters are reported in the Table 9.

Subsampling for OT coupling. Each rendered silhouette contains many foreground pixels; to make mini-batch OT pairing tractable, we subsample a fixed number of points per cloud. We use $n = 100$ points per cloud because the discrete OT coupling has super-cubic complexity in n in our implementation, making larger clouds prohibitively expensive. This subsampling is applied consistently across all methods.

Key hyperparameters. Unless otherwise stated, we use: batch size 10, point subsampling $n = 100$, number of letters 4, number of rotations $R = 20$, number of colour replicas per condition $C = 100$, $(r_{\min}, r_{\max}) = (0.5, 0.6)$, and zero coordinate noise.

Blocked rotation interpolation. To test whether the synthetic generalization gains rely mainly on smooth interpolation between neighboring rotations, we additionally evaluate a more challenging blocked-rotation split. In this setting, we generate $R = 25$ rotations and divide them into alternating blocks of five seen and five unseen rotations: rotations 0–4 are used for training, rotations 5–9 are held out, rotations 10–14 are used for training, and so on. This produces larger gaps between seen and unseen rotations than the even/odd split, reducing the advantage of local neighboring-rotation interpolation. For this experiment, we use h_θ to predict the GMM modes from condition descriptors with $I = 10$ mixture components. We use `S` as the validation letter and `W` as the unseen test letter. As shown in Table 8, although all distances increase relative to the smoother even/odd interpolation setting, SP-FM consistently outperforms vanilla CFM across all metrics, indicating that the generalization benefit persists even under larger rotation gaps.

C.3 SP-FM IMPLEMENTATION

Dual-flow architecture (geometry vs. color). We use a dual-MLP conditional flow architecture: (i) a geometry flow v_{xy} acting on $(x, y) \in \mathbb{R}^2$ per point, and (ii) a color flow v_{rgb} acting on an RGB vector $(r, g, b) \in \mathbb{R}^3$ per cloud. Concretely, we decompose each cloud as

$$\mathbf{X}_{\text{xy}} \in \mathbb{R}^{N\times 2}, \qquad \mathbf{c} \in \mathbb{R}^{1\times 3},$$

where $\mathbf{c}$ denotes the (shared) RGB color of the cloud (in our setup, all points in a cloud share the same color). Both velocity fields are implemented as MLPs that take time t and the concatenation of state and condition:

$$v_{\text{xy}}(t, \mathbf{x}_{\text{xy}}; \mathbf{y}) \;=\; \text{MLP}_{\text{xy}}\big(t,\, [\mathbf{x}_{\text{xy}}, \mathbf{y}]\big), \qquad v_{\text{rgb}}(t, \mathbf{c}; \mathbf{y}) \;=\; \text{MLP}_{\text{rgb}}\big(t,\, [\mathbf{c}, \mathbf{y}]\big).$$

Conditional base distribution with random per-condition mode initialization. To sample the initial state $\mathbf{X}_0$ we use a condition-dependent mixture base. In the `use_kmeans=True` setting, mixture centers are initialized by **randomly sampling** K training point clouds for each condition $\mathbf{y}$ (i.e., for each letter-rotation pair), rather than running K-means. These sampled clouds serve as fixed centers, while the mixture weights are predicted by a small network $h_p(\mathbf{y})$. At sampling time we draw

a (soft) mixture over centers via Gumbel–Softmax (temperature τ) and add Gaussian perturbation with variance σ^2 to obtain $\mathbf{X}_{0,\text{xy}}$.

For color, we initialize a single RGB value per cloud by sampling uniformly from a predefined range $[a, b] \subset [0, 1]$ and broadcasting it to all points:

$$\mathbf{c}_0 \sim \mathcal{U}([a, b]^3), \qquad \mathbf{X}_{0,\text{rgb}} = \mathbf{1}_N \mathbf{c}_0,$$

and finally concatenate geometry and color to obtain $\mathbf{X}_0 = [\mathbf{X}_{0,\text{xy}}, \mathbf{X}_{0,\text{rgb}}] \in \mathbb{R}^{N\times 5}$.

OT pairing. Before computing training losses, we align points between the sampled source cloud $\mathbf{X}_0$ and the target cloud $\mathbf{X}_1$ using point-wise optimal transport (OT) within each sample.

Flow matching objective. Given an OT-aligned pair $(\mathbf{X}_0, \mathbf{X}_1)$ under condition $\mathbf{y}$, we sample $t \sim \mathcal{U}[0, 1]$ and define

$$\mathbf{X}_t = (1 - t)\mathbf{X}_0 + t\mathbf{X}_1, \qquad \mathbf{U} = \mathbf{X}_1 - \mathbf{X}_0.$$

We use the standard flow-matching regression form

$$\mathcal{L}_{\text{FM}} = \mathbb{E}\Big[\|v(t, \mathbf{X}_t; \mathbf{y})\|_2^2 - 2\langle v(t, \mathbf{X}_t; \mathbf{y}),\, \mathbf{U}\rangle\Big].$$

With the dual architecture, we compute this loss separately for geometry and color and sum them:

$$\mathcal{L}_{\text{FM}} = \mathcal{L}_{\text{FM}}^{\text{xy}} + \mathcal{L}_{\text{FM}}^{\text{rgb}},$$

where $\mathcal{L}_{\text{FM}}^{\text{xy}}$ is evaluated over all N points, and $\mathcal{L}_{\text{FM}}^{\text{rgb}}$ is evaluated on the single RGB vector per cloud.

Geodesic-length loss (default endpoint loss). In addition, we employ a shortest-path regularizer using the default endpoint loss:

$$\mathcal{L}_{\text{geo}} = \mathbb{E}\big[\|\mathbf{X}_1 - \mathbf{X}_0\|_2^2\big].$$

In the dual setting, this decomposes as

$$\mathcal{L}_{\text{geo}} = \mathcal{L}_{\text{geo}}^{\text{xy}} + \mathcal{L}_{\text{geo}}^{\text{rgb}}, \qquad \mathcal{L}_{\text{geo}}^{\text{xy}} = \mathbb{E}\big[\|\mathbf{X}_{1,\text{xy}} - \mathbf{X}_{0,\text{xy}}\|_2^2\big], \quad \mathcal{L}_{\text{geo}}^{\text{rgb}} = \mathbb{E}\big[\|\mathbf{c}_1 - \mathbf{c}_0\|_2^2\big].$$

Training schedule: warmup and alternating updates. We use manual optimization with two optimizers: one for the flow parameters (geometry and color flows) and one for the base components (mixture predictors). Training proceeds in two stages.

Warmup (flow-only). For the first K_{warm} steps, we update only the flow parameters by minimizing the flow-matching loss:

$$K_{\text{warm}} = 1000, \qquad \theta \leftarrow \theta - \eta_\theta \nabla_\theta \mathcal{L}_{\text{FM}},$$

where θ denotes the parameters of v_{xy} and v_{rgb}. During warmup, the base model is not updated.

Alternating phase (base vs. flow). After warmup, we alternate updates between the base model and the flow. Let s be the global step counter and define $s' = s - K_{\text{warm}}$. Every $(M + 1)$ steps we perform a base update; otherwise we perform a flow update:

$$\text{update base at step } s \iff s' \bmod (M + 1) = 0, \qquad M = 10.$$

Therefore, after warmup, we perform one base update followed by M flow updates repeatedly (we used $M = 10$).

Detailed hyperparameters are available in Table 12.

C.4 GMM IMPLEMENTATION

Conditional GMM baseline (no binning; exact-condition table with interpolation). In the GMM baseline, each target point cloud is flattened into a single vector

$$x \in \mathbb{R}^{D_{\text{cloud}}}, \qquad D_{\text{cloud}} = N \cdot d,$$

where N is the number of points per cloud and d is the per-point feature dimension (in our setup $d = 5$). Because the dataloader may return points in a random order, we optionally sort points by the x-coordinate before flattening to make the representation consistent across batches.

Table 9: Results on the synthetic experiment based on letter H silhouettes. This experiment assesses the model's ability to generalize to rotations of the corresponding shape that are not observed during training.

Dataset	Model	Generated vs. Target			Source vs. Target		
		MMD ↓	W1 ↓	W2 ↓	MMD ↓	W1 ↓	W2 ↓
Letter H (Validation)	CFM	0.02021 ± 0.00115	0.7044 ± 0.0222	0.8626 ± 0.0229	0.09731 ± 0.00101	1.2101 ± 0.0056	1.3342 ± 0.0066
	SP-FM	**0.01387 ± 0.00092**	**0.5939 ± 0.0190**	**0.7372 ± 0.0198**	**0.02773 ± 0.00042**	**0.7631 ± 0.0053**	**0.9048 ± 0.0047**
Letter W (Test)	CFM	0.01778 ± 0.00101	0.6118 ± 0.0187	0.7219 ± 0.0203	0.06088 ± 0.00049	0.8972 ± 0.0062	1.0151 ± 0.0071
	SP-FM	**0.01352 ± 0.00143**	**0.5429 ± 0.0196**	**0.6441 ± 0.0167**	**0.01872 ± 0.00062**	**0.6140 ± 0.0091**	**0.7173 ± 0.0099**
Letter Y (Test)	CFM	0.02648 ± 0.00230	0.6669 ± 0.0295	0.7779 ± 0.0390	0.06020 ± 0.00128	1.0614 ± 0.0048	1.2407 ± 0.0055
	SP-FM	**0.01735 ± 0.00167**	**0.5406 ± 0.0259**	**0.6358 ± 0.0295**	**0.04134 ± 0.00221**	**0.9525 ± 0.0399**	**1.1096 ± 0.0448**

Model. We fit a K-component diagonal-covariance GMM with condition-dependent mixture weights:

$$p(x \mid c) \;=\; \sum_{k=1}^{K} \pi_{c,k}\, \mathcal{N}\big(x;\, \mu_k,\, \mathrm{diag}(\sigma_k^2)\big),$$

where the component parameters $\{\mu_k, \sigma_k^2\}_{k=1}^{K}$ are shared across all conditions, and the mixture weights $\pi_c \in \Delta^{K-1}$ are stored in a table Π indexed by a discrete condition id.

Condition indexing. We treat each training condition vector c (e.g., one-hot letter identity concatenated with a continuous rotation coordinate) as a distinct key:

$$c \;\mapsto\; \text{cond_id}(c),$$

implemented by rounding the condition values to a fixed precision (e.g., 10^{-6}) to obtain stable dictionary keys. This produces a table $\Pi \in \mathbb{R}^{C\times K}$ where C is the number of unique training conditions.

Interpolation for unseen rotations (generalization_type = interpolation). At test time, if a condition $c^\star$ was not seen during training (e.g., an unseen rotation), we compute $\pi(c^\star)$ by linearly interpolating mixture-weight vectors from the two closest seen rotations. Let r denote the rotation coordinate inside the condition vector (the last dimension in our setting), and let $\mathcal{S}$ be the set of seen conditions restricted to the same letter identity (i.e., same one-hot dimensions, different r). We find two neighbors $(c_0, c_1) \in \mathcal{S}$ such that $r_0 \le r^\star \le r_1$, and set

$$a \;=\; \frac{r^\star - r_0}{r_1 - r_0 + \epsilon}, \qquad \pi(c^\star) \;=\; (1-a)\,\pi(c_0) \;+\; a\,\pi(c_1),$$

with ϵ a small constant for numerical stability. If $r^\star$ lies outside the seen range, we clamp to the nearest endpoint (i.e., use $\pi(c_0)$ or $\pi(c_1)$). Optionally, we can sharpen $\pi(c)$ at inference using

$$\pi'(c) \;=\; \mathrm{softmax}\!\left(\frac{\log(\pi(c))}{T_\pi}\right),$$

where $T_\pi = 1$ leaves the distribution unchanged and $T_\pi < 1$ makes it peakier.

EM fitting with per-condition π_c. We learn (μ_k, σ_k^2) jointly with the table $\{\pi_c\}_{c=1}^{C}$ using EM on training pairs $\{(x_n, c_n)\}_{n=1}^{N}$.

C.5 Comparison with GMFlow and PriorGrad

We further compare SP-FM with GMFlow (Chen et al., 2025) and PriorGrad (Lee et al., 2022) on the letter–rotation benchmark

(Table 10).

D CellFlow + SPFM

CellFlow + SPFM replaces the fixed control-cell source distribution of vanilla CellFlow with a jointly trained, condition-dependent GMM base distribution, following the SP-FM framework.

Table 10: Comparison with GMFlow and PriorGrad on the synthetic letter–rotation benchmark (Letter W, test; held-out odd-indexed rotations). All distances are Generated→Target; lower is better. GMFlow models the velocity field as a Gaussian mixture with a fixed $\mathcal{N}(0, I)$ base. For PriorGrad, the prior mean for an unseen condition is set to the mean of the closest training condition. CFM and SP-FM (I=10) are reproduced from Table 1.

Model	MMD ($\times 10^{-4}$) ↓	W1 ↓	W2 ↓	ED ($\times 10^{-2}$) ↓
CFM	180.9 ± 8.1	0.618 ± 0.014	0.728 ± 0.015	3.770 ± 0.139
GMFlow	170	0.585	0.711	3.39
PriorGrad	156	0.571	0.679	2.45
SP-FM (I=10)	$\mathbf{123.9 \pm 7.4}$	$\mathbf{0.523 \pm 0.015}$	$\mathbf{0.623 \pm 0.018}$	$\mathbf{2.310 \pm 0.146}$

Input representation. Raw gene expression profiles (post-normalisation and log-transformation) are projected into a $d = 50$-dimensional PCA space fitted on the cells using highly variable genes. All subsequent computations operate in this PCA-50 space.

Condition encoder. Each perturbation condition c is represented as a set of one or more covariate tokens. A shared embedding layer maps each covariate to a learned vector; a set encoder with *attention-token pooling* then aggregates variable-size sets into a fixed-size condition embedding $e_c \in \mathbb{R}^{256}$. Multi-covariate combinations are handled naturally by pooling over all set elements.

Conditional velocity The velocity field is a neural network with three sub-networks:

1. **Time encoder:** the scalar flow time $t \in [0, 1]$ is embedded via a sinusoidal encoding (1 024 frequencies, max period 10^4) and passed through a 3-layer MLP with hidden width 2 048 and SiLU activations, producing a time embedding.
2. **Cell encoder:** the cell state $x \in \mathbb{R}^{50}$ is passed through an independent 3-layer MLP with hidden width 2 048 and SiLU activations.
3. **Decoder:** the time embedding, cell embedding, and condition embedding e_c are concatenated and fed through a 3-layer MLP with hidden width 4 096, producing the predicted velocity $\hat{v} \in \mathbb{R}^{50}$.

GMM base distribution. The GMM module predicts $K = 16$ Gaussian components in PCA-50 space from the condition embedding e_c:

- $H_\theta(e_c) \in \mathbb{R}^{K \times d}$: a single hidden-layer MLP (width 128, SiLU) predicting the K component means.
- $H_p(e_c) \in \mathbb{R}^K$: a single hidden-layer MLP (width 128, SiLU) predicting the (unnormalised) mixture logits.

A sample x_0 is drawn from the GMM via Gumbel-Softmax (temperature $\tau = 0.5$, soft relaxation):

$$x_0 = \sum_{k=1}^{K} w_k \, \mu_k + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \, \sigma^2 I), \tag{32}$$

where $w_k = \mathrm{GumbelSoftmax}(H_p(e_c))_k$ and $\mu_k = H_\theta(e_c)_k$. The variance σ^2 is automatically calibrated per dataset: the k-means clustering of the PCA-projected training cells is fitted first ($K = 16$ clusters), and σ^2 is set to the mean within-cluster squared residual:

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^{N} \|x_i - \mu_{k(i)}\|^2, \tag{33}$$

where $k(i)$ is the cluster assignment of control cell i.

k-means initialisation. The $K = 16$ cluster centres from the above k-means fit are used to initialise the bias of the final linear layer of H_θ, with the kernel initialised near zero ($\sigma_{\text{init}} = 10^{-3}$). This ensures that, at initialisation, $H_\theta(e_c) \approx \{\mu_k\}_{k=1}^K$ for every condition, grounding the GMM in the geometry of the training-cells manifold.

E MODEL HYPERPARAMETERS

Table 11: Model, optimization, and training hyperparameters for SP-FM.

Category	Setting	Role / explanation
Base distribution (conditioned GMM)		
Number of mixture components	$I = 20$	Controls expressivity of the condition-dependent base distribution. Larger I allows better alignment with multimodal targets.
Component variance	$\sigma^2 = 0.01$	Fixed isotropic variance for each GMM component to stabilize training and avoid degenerate modes.
Conditioned base	enabled (disabled for vanilla CFM)	Base distribution parameters are predicted from perturbation descriptors via a learned encoder.
Gumbel–softmax temperature	$\tau = 0.5$	Softens mixture assignments during training to ensure differentiability of base distribution learning.
Hard mixture assignments	disabled	Uses soft (probabilistic) mixture weights rather than hard argmax selection.
Flow / velocity field architecture		
Hidden layer widths	$[200, 200, 200]$	Three-layer MLP used to parameterize the conditional velocity field.
Main network dropout	0.0	Dropout disabled in the velocity field to preserve deterministic dynamics.
Condition encoder		
Encoder dropout	0.5	Dropout applied to the base distribution encoder to regularize conditioning on perturbation descriptors.
Data representation		
PCA dimensionality	$d = 20$	Gene expression profiles are projected into a low-dimensional latent space prior to modeling.
Optimization		
Flow optimizer	Adam, lr $= 10^{-3}$, betas:(0.0, 0.99)	Optimizer and learning rate for training the conditional velocity field.
Base distribution optimizer	Adam, lr $= 10^{-3}$	Optimizer and learning rate for training the base distribution encoder.
Training epochs	1000	Total number of training epochs.
Training schedule		
Warmup phase	5% of training	Only the velocity field is trained while the base distribution is frozen.
Alternating phase	85% of training	Alternates between updating the velocity field and the base distribution.
Cooldown phase	10% of training	Base distribution is frozen; training refines the flow only.
Base update frequency	every 10 iterations	Controls how often the base distribution is updated during the alternating phase.
Optimal transport and evaluation		
OT coupling method	Exact minibatch OT	Constructs shortest-path interpolants using exact optimal transport couplings.
ODE solver steps (eval)	100	Number of integration steps used to generate samples at evaluation time.

We summarize the main hyperparameters used across all single-cell perturbation experiments in Table 11. We fix the number of mixture components in the condition-dependent base distribution to $I = 20$, use a three-layer MLP for the conditional velocity field, and train for 1000 epochs with Adam. Base distribution and flow parameters are optimized using an alternating schedule with an initial warmup and final cooldown phase. We also report the hyperparameters for the other two experiments

the synthetic letter-rotation benchmark and the image-based morphology study in Tables 12 and 13, respectively.

Table 12: Model, optimization, and training hyperparameters for SP-FM (letters).

Category	**Setting**	**Role / explanation**
Base distribution (conditioned GMM)		
Number of modes	60	Number of mixture modes used in the conditioned base.
Variance	0.25	Fixed isotropic variance used for sampling from the base.
Temperature	1.0	Controls softness of mixture weighting during sampling.
Mode initialization	Random sample	Initializes mixture modes using random samples per condition on training samples.
Dropout in base weights	0.2	Regularization in the base weight predictor.
Flow / velocity field		
Architecture	Dual MLP	Separate networks for coordinates and color.
Layers	4	Number of MLP layers.
Hidden width	512	Hidden layer size of the MLP(s).
Integration steps	500	Number of ODE solver steps at evaluation.
Data and conditioning		
Feature dimensionality	5	Coordinates plus RGB color features.
Points per sample	100	Number of points per point-cloud sample.
Training batch size	10	Batch size used for training iterations.
Generalization	Interpolation	Train/test split uses interpolation across conditions.
RGB init range	[0.5, 0.6]	Range for initializing RGB per sample.
Optimal transport and loss		
OT pairing	Enabled	Uses OT-based matching between source and target samples.
Geodesic loss	Squared endpoint mismatch	Choice of geodesic/base loss variant used in training.
Optimization and schedule		
Flow optimizer	Adam	Optimizer for flow network parameters.
Flow learning rate	1×10^{-4}	Learning rate for flow updates.
Flow betas	(0.0, 0.999)	Adam momentum parameters for flow.
Base optimizer	Adam	Optimizer for base distribution parameters.
Base learning rate	1×10^{-4}	Learning rate for base updates.
Warmup steps	1000	Only flow is updated during the warmup phase.
Base update interval	10	Base update happens every 11 steps after warmup (alternating schedule).
Epochs	2500	Maximum training epochs.
Validation frequency	250	Run validation every 250 epochs.

Table 13: Model, optimization, and training hyperparameters for SP-FM(morphological).

Category	Setting	Role / explanation
Base distribution (conditioned GMM)		
Number of mixture components	$I = 20$	Controls expressivity of the condition-dependent base distribution. Larger I allows better alignment with multimodal targets.
Component variance	$\sigma^2 = 0.01$	Fixed isotropic variance for each GMM component to stabilize training and avoid degenerate modes.
Conditioned base	enabled (disabled for vanilla CFM)	Base distribution parameters are predicted from perturbation descriptors via a learned encoder.
Gumbel–softmax temperature	$\tau = 0.5$	Softens mixture assignments during training to ensure differentiability of base distribution learning.
Hard mixture assignments	disabled	Uses soft (probabilistic) mixture weights rather than hard argmax selection.
Flow / velocity field architecture		
Hidden layer widths	$[200, 200, 200]$	Three-layer MLP used to parameterize the conditional velocity field.
Main network dropout	0.0	Dropout disabled in the velocity field to preserve deterministic dynamics.
Condition encoder		
Encoder dropout	0.5	Dropout applied to the base distribution encoder to regularize conditioning on perturbation descriptors.
Data representation		
Embedding dimensionality	$d = 768$	For the original setup image embeddings extracted using the DINOv2 model.
Optimization		
Flow optimizer	Adam, lr $= 10^{-3}$, betas:(0.0, 0.99)	Optimizer and learning rate for training the conditional velocity field.
Base distribution optimizer	Adam, lr $= 10^{-3}$	Optimizer and learning rate for training the base distribution encoder.
Training epochs	100	Total number of training epochs.
Training schedule		
Warmup phase	10% of training	Only the velocity field is trained while the base distribution is frozen.
Alternating phase	80% of training	Alternates between updating the velocity field and the base distribution.
Cooldown phase	10% of training	Base distribution is frozen; training refines the flow only.
Base update frequency	every 10 iterations	Controls how often the base distribution is updated during the alternating phase.
Optimal transport and evaluation		
OT coupling method	Exact minibatch OT	Constructs shortest-path interpolants using exact optimal transport couplings.
ODE solver steps (eval)	100	Number of integration steps used to generate samples at evaluation time.

F COMPUTE SPECIFICATIONS

All our experiments are performed on an NVIDIA A100 80GB GPU.

Table 14: Hyperparameter analysis on BBBC021 dataset.

Group	Setting	W1 ↓	W2 ↓	MMD ↓	Energy ↓
Condition encoder	[128]	35.71±1.10	679.78±24.5	0.1001±0.0050	6.07±0.30
	[256]	30.87±0.65	489.41±12.0	0.0434±0.0020	2.39±0.10
	[512]	31.48±0.70	515.49±13.5	0.0479±0.0020	2.66±0.11
	[128,128]	28.82±0.55	420.54±9.8	0.0395±0.0015	2.02±0.08
	[256,256]	28.76±0.50*	419.10±9.5*	0.0368±0.0014	1.92±0.07*
	[512,512]	28.90±0.55	424.34±10.0	0.0366±0.0013*	1.93±0.07
Flow module	[50,50,50]	28.79±0.55	425.72±10.0	0.0550±0.0025	2.86±0.12
	[100,100,100]	35.84±1.05	718.54±26.0	0.0987±0.0045	5.82±0.27
	[200,200,200]	29.48±0.60*	442.99±10.5*	0.0422±0.017*	2.25±0.09*
GMM variance	0.0001	27.47±0.50*	383.32±8.5*	0.0596±0.0028	2.71±0.11
	0.0025	30.73±0.65	488.21±12.0	0.0813±0.0040	4.13±0.20
	0.01	33.03±0.80	628.01±18.0	0.0525±0.0023	3.12±0.14
	0.04	32.68±0.75	553.40±14.0	0.0405±0.0016*	2.31±0.09*
	0.25	60.67±3.30	1891.74±110	0.1478±0.0110	8.11±0.55
	1.0	109.67±8.00	6043.92±480	0.2558±0.0250	19.62±1.65
Scheduler (w/j/c)	0.1 / 0.8 / 0.1	29.98±0.60	466.02±11.5	0.0553±0.024	2.89±0.12
	0.2 / 0.7 / 0.1	29.03±0.55*	432.55±10.0	0.0441±0.0019	2.16±0.09
	0.3 / 0.6 / 0.1	30.30±0.65	466.28±11.5	0.0459±0.0020	2.48±0.10
	0.1 / 0.7 / 0.2	29.20±0.55	432.01±10.0*	0.0380±0.0015*	1.98±0.08*
	0.1 / 0.6 / 0.3	29.77±0.60	461.10±11.0	0.0439±0.0018	2.30±0.09
Update freq.	10	30.72±0.65	480.85±11.5	0.0480±0.0021	2.59±0.10
	15	32.20±0.75	529.25±13.0	0.0679±0.0032	3.73±0.17
	20	30.33±0.65*	468.23±11.0*	0.0465±0.0020*	2.59±0.10*

G Hyperparameter Analysis

We performed extensive hyper-parameter analysis on BBBC021 dataset, varying the architecture of condition encoder, the architecture of flow module, GMM variance, scheduler, and base distribution update frequency. The results are provided in Tab. 14. This analysis show that the algorithm is robust to different scheduler settings and update frequencies to the base distributions (Groups "Scheduler" and "Update freq." in Tab. 14). Nonetheless, the fixed variance of GMM modes has a significant effect, and the value 0.0001 performs the best, which is the default value that we used throughout our experiments. Importantly, Tab. 14 shows that increasing the number of parameters in condition encoder and flow module doesn't necessarily improve the performance, therefore the improvement obtained in SP-FM is not associated with increasing the number of parameters.

H Upper bound on the discrepancy between the distribution distance metrics

The theoretical version of the proposed SP-FM method uses mixture Wasserstein distance, while the practical version uses the normal Wasserstein distance. In this section we provide a bound to quantify this gap. Afterwards, we calculate the bound on 4 real datasets and across different settings.

We start by introducing a lemma from (Delon & Desolneux, 2020).

Lemma H.1. *Let p and q be two GMMs with mode variances σ^2, $\mathcal{W}(.,.)$ be the 2-Wasserstein distance, $\mathcal{MW}(.,.)$ be the mixture Wasserstein distance, and D be the data dimension. We have that:*

$$\mathcal{W}(p,q) \leq \mathcal{MW}(p,q) \leq \mathcal{W}(p,q) + 2\sqrt{2D\sigma^2}. \tag{34}$$

This is the Proposition 6 in (Delon & Desolneux, 2020) (arXiv preprint, V4), when the GMM modes have diagonal covariance matrices with equal variance σ^2.

let μ be the base GMM that the SP-FM algorithm keeps updating for the target distribution ρ, with the theoretical and practical versions using the mixture Wasserstein and Wasserstein distances, denote by $\mathcal{MW}(.,.)$ and $\mathcal{W}(.)$, respectively.

By Lemma H.1:

$$\mathcal{MW}(\mu,\rho) \leq \mathcal{W}(\mu,\rho) + 2\sqrt{2D\sigma^2}. \tag{35}$$

After dividing both sides by $\mathcal{MW}(\mu,\rho)$ we get:

$$1 \leq \frac{\mathcal{W}(\mu,\rho)}{\mathcal{MW}(\mu,\rho)} + \frac{2\sqrt{2D\sigma^2}}{\mathcal{MW}(\mu,\rho)} \tag{36}$$

Therefore:

$$1 - \frac{\mathcal{W}(\mu,\rho)}{\mathcal{MW}(\mu,\rho)} \leq \frac{2\sqrt{2D\sigma^2}}{\mathcal{MW}(\mu,\rho)} \tag{37}$$

In the equation above, the left hand side is a number between zero and one, and measures the amount of discrepancy between the 2 versions of the proposed SP-FM algorithm (the lower the better). To have an upper bound for it, we use the following equation:

$$1 - \frac{\mathcal{W}(\mu,\rho)}{\mathcal{MW}(\mu,\rho)} \leq \frac{2\sqrt{2D\sigma^2}}{\mathcal{MW}(\mu,\rho)} \leq \frac{2\sqrt{2D\sigma^2}}{\min_\nu \mathcal{MW}(\nu,\rho)} \tag{38}$$

For each population ρ in the training set, we first estimate $\min_\nu \mathcal{MW}(\nu,\rho)$ via the procedure explained next. Afterwards, we plug it to Eq. 38 to get an upper bound on the error.

For a population ρ, we estimate $\min_\nu \mathcal{MW}(\nu,\rho)$ as follows. Note that ν runs over GMMs with I modes, but ρ may have $J >> I$ modes. Given N training samples from ρ, we approximate it as a N-mode GMM with constant variance and mode probability values (kernel density estimation). To arrive at the optimal ν, we start by a randomly-initialised GMM ν and repeatedly update its mode positions and probabilities. To update mode positions, we firstly calculate the discrete optimal transport between μ and ν. We then update the modes of ν via the barycentre condition in Statement. B.9. To update the mode probabilities of ν, we use unbalanced Sinkhorn algorithm implemented in POT (Flamary et al., 2026) (`ot.sinkhorn_unbalanced` with `reg_m=(float('inf'), 0.001)`) and set ν's mode probabilities to the target marginal of the optimal transport plan. The model positions and probabilities of ν are updated until the maximum shift to mode positions are below $1e-5$ or a limit of 100 iterations is reached.

The estimated values for the bound in Eq. 38 are provided in the figures below, across 4 datasets, number of base modes I, and number of PCs (i.e. data dimension D).

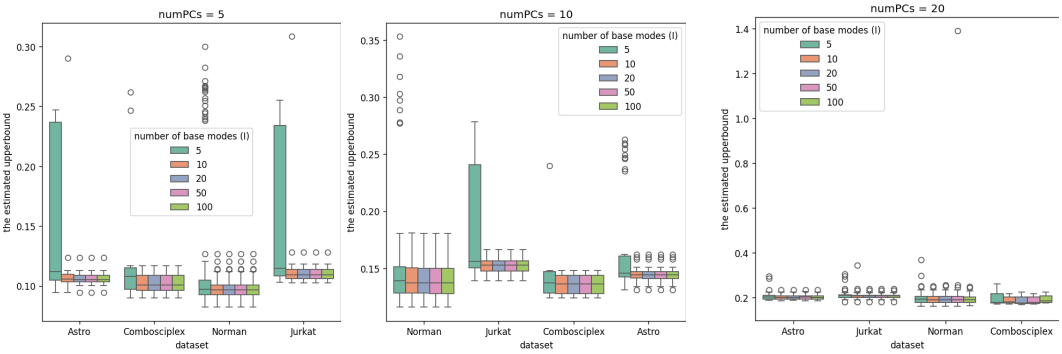


Figure 4: The estimated value for the upper-bound of Eq. 38.

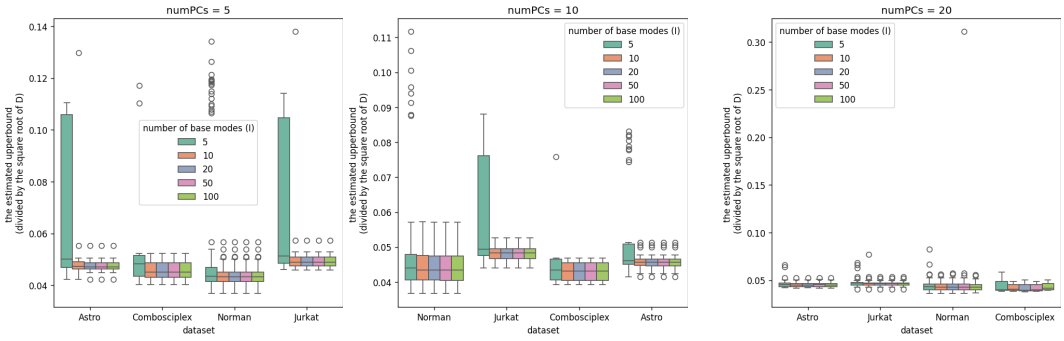


Figure 5: The estimated value for the upper-bound of Eq. 38, divided by $\sqrt{D}$.