MAPPFN: LEARNING CAUSAL PERTURBATION MAPS IN CONTEXT

Marvin Sextro[1,2,3] **Weronika Kłos**[1,2] **Gabriel Dernbach**[1,2,4,3]
[1]Machine Learning Group, Technische Universität Berlin, Berlin, Germany
[2]Berlin Institute for the Foundations of Learning and Data (BIFOLD)
[3]Aignostics, Berlin, Germany
[4]Institute of Pathology, Charité - Universitätsmedizin Berlin, Berlin, Germany
m.kleine.sextro@tu-berlin.de

ABSTRACT

Planning effective interventions in biological systems requires treatment-effect models that adapt to unseen biological contexts by identifying their specific underlying mechanisms. Yet single-cell perturbation datasets span only a handful of biological contexts, and existing methods cannot leverage new interventional evidence at inference time to adapt beyond their training data. To meta-learn a perturbation effect estimator, we present MapPFN, a prior-data fitted network (PFN) pretrained on synthetic data generated from a prior over causal perturbations. Given a set of experiments, MapPFN uses in-context learning to predict post-perturbation distributions, without gradient-based optimization. Despite being pretrained on *in silico* gene knockouts alone, MapPFN identifies differentially expressed genes, matching the performance of models trained on real single-cell data. Our code and data are available at https://github.com/marvinsxtr/MapPFN.

1 INTRODUCTION

To gain a mechanistic understanding of the behavior of cell populations, single-cell perturbation data has long been the experimental gold standard to identify the causal dependencies that form underlying gene regulatory networks (GRNs) (Sachs et al., 2005). Genetic CRISPR knockout perturbations (Jinek et al., 2012) measured in single cells using Perturb-Seq (Dixit et al., 2016) allow us to measure the outcome of targeted interventions in controlled biological contexts like cell lines (Frangieh et al., 2021). Yet, mapping the whole space of possible cell states and perturbations through experiments alone is infeasible. Even the largest perturbation dataset to date only contains 50 biological contexts (cell lines) (Zhang et al., 2025).

This bottleneck has motivated methods that learn how cells respond to perturbations induced by small molecules or gene knockouts (Bunne et al., 2024; Roohani et al., 2025). Such virtual cell models are aimed at reducing the costs of drug target discovery, as they enable low latency and high throughput evaluation of hypotheses prior to costly and time-consuming validation in the wet lab.

Because sequencing destroys individual cells, perturbation prediction becomes a problem of mapping between unpaired distributions, making optimal transport (OT) a natural approach. These methods learn a transport map between the pre- and post-perturbation cell distributions, conditioned on a treatment or covariates (Bunne et al., 2023; Dong et al., 2023). Lifting the strict assumptions of OT-based methods, recent approaches use generative models to predict the post-perturbation distribution conditioned on covariates (Lotfollahi et al., 2023; Klein et al., 2025) or a learned representation of the initial observational distribution (Atanackovic et al., 2025; Adduri et al., 2025). Yet, they lack test-time, context-based adaptation from a small set of observed interventions, constraining generalization to the biological contexts seen during training.

In this work, we propose to meta-learn perturbation maps using a set of interventional distributions as context conditioning, enabling a diffusion transformer to infer perturbation effects via in-context adaptation. Building on the recent success of prior-data fitted networks (PFNs) (Müller et al., 2022) in tabular prediction (Hollmann et al., 2025; 2023; Qu et al., 2025) and causal inference (Robertson

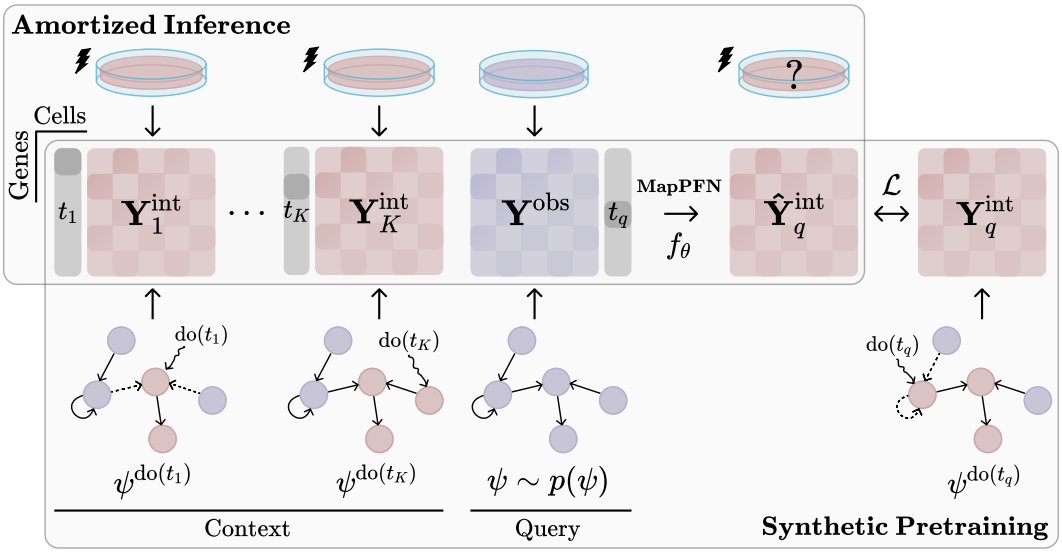

Figure 1: **MapPFN overview.** MapPFN uses in-context learning (ICL) to predict perturbation effects in unseen biological contexts. During pretraining, we draw structural causal models (SCMs) or synthetic gene regulatory networks (GRNs) ψ to generate samples from the observational distribution $\mathbf{Y}^{\text{obs}}$ and a context set of interventional distributions $\{(t_k, \mathbf{Y}^{\text{int}}_k)\}_{k=1}^{K}$, where t_k denotes a perturbation (do-intervention). Given $\mathbf{Y}^{\text{obs}}$ and the context set, MapPFN predicts post-perturbation distributions $\mathbf{Y}^{\text{int}}_q$ arising from unseen interventions t_q. During pretraining, MapPFN meta-learns how to map between pre- and post-perturbation distributions across many causal structures ψ by minimizing $\mathcal{L}(\hat{\mathbf{Y}}^{\text{int}}_q, \mathbf{Y}^{\text{int}}_q)$. At inference time, MapPFN predicts cell-level post-perturbation distributions $\mathbf{Y}^{\text{int}}_q \in \mathbb{R}^{\text{cells}\times\text{genes}}$ in one step through amortized inference, without requiring gradient-based optimization or knowledge of the underlying causal structure ψ.

et al., 2025; Balazadeh et al., 2025; Ma et al., 2025), we introduce PFNs for perturbation prediction with a multi-experiment input to achieve foundational perturbation model pretraining on synthetic data. Different to standard PFN training, our task requires the prediction of a distribution of vectors, for which we adopt the Multimodal Diffusion Transformer (MMDiT) architecture (Esser et al., 2024). We show that conditioning on pre- and multiple post-perturbation distributions improves performance over models that only use a pre-perturbation distribution with a query treatment identifier. Pretrained exclusively on synthetic data, MapPFN identifies differentially expressed genes without any fine-tuning, performing on par with methods trained directly on real single-cell data (Frangieh et al., 2021).

Our Contributions

1. **MapPFN** We frame perturbation prediction as a context-conditioned distribution mapping, and present MapPFN, a prior-data fitted network (PFN) pretrained exclusively on synthetic data, that uses in-context learning (ICL) to predict perturbation effects for unseen biological contexts conditioned on a set of pre- and post-perturbation distributions.
2. **Synthetic Evaluation** In a controlled synthetic benchmark of structural causal models (SCMs), MapPFN achieves competitive few- and zero-shot performance. We quantify identity collapse as a dominant failure mode in existing methods.
3. **Single-cell Evaluation** Pretrained only on a synthetic prior of *in silico* gene knockouts, MapPFN transfers to real single-cell data and identifies differentially expressed genes, achieving similar AUPRC to baselines trained on real single-cell perturbations.
4. **Paired vs. Unpaired Pretraining** We find that pretraining MapPFN on paired interventional distributions improves downstream performance, compared to independent interventional distributions.

2 Problem Statement

We consider the problem of learning how biological systems behave under interventions. In the case of single-cell perturbations, we are given a set of N gene expressions $\mathbf{y}^{\text{obs}} \in \mathbb{R}^d$ measured in a specific cell line and a treatment $t \in \mathcal{T}$ in the form of an intervention on a single gene, resulting in M post-treatment gene expressions $\mathbf{y}^{\text{int}} \in \mathbb{R}^d$. The resulting dataset takes the form $\left\{\left(\mathbf{Y}_\ell^{\text{obs}}, t_\ell, \mathbf{Y}_\ell^{\text{int}}\right)\right\}_{\ell=1}^{L}$, where $\mathbf{Y}^{\text{obs}} \in \mathbb{R}^{N\times d}$, $\mathbf{Y}^{\text{int}} \in \mathbb{R}^{M\times d}$ and L is the number of pairs of biological contexts and treatments. Importantly, there is no direct correspondence between any two pre- and post-treatment cells, rendering this as a problem of learning a map between distributions $p(\mathbf{y}^{\text{obs}})$ and $p(\mathbf{y}^{\text{int}})$.

Given observational samples $\mathbf{Y}^{\text{obs}}$ and an interventional context $\mathcal{C} = \left\{\left(t_k, \mathbf{Y}_k^{\text{int}}\right)\right\}_{k=1}^{K}$ for a subset of treatment conditions $t_k \in \mathcal{T}_\mathcal{C} \subset \mathcal{T}$ in a biological context, we aim to predict the outcome of an unseen query perturbation $t_q \in \mathcal{T} \setminus \mathcal{T}_\mathcal{C}$:

$$p(\mathbf{y}_q^{\text{int}} \mid \text{do}(t_q), \mathbf{Y}^{\text{obs}}, \mathcal{C}) \tag{1}$$

For evaluation, we distinguish between two settings: (i) few-shot, where the model is given observations for a subset of perturbations, and (ii) zero-shot, where no perturbation data is available for the held-out biological context (i.e., $\mathcal{C} = \emptyset$).

3 Related Work

Perturbation Prediction Existing methods differ in their generalization target. Approaches like scGen (Lotfollahi et al., 2019), CPA (Lotfollahi et al., 2023), CellOT (Bunne et al., 2023), and Meta Flow Matching (Atanackovic et al., 2025) aim to generalize across biological contexts by learning conditional maps between pre- and post-perturbation distributions. Methods targeting unseen perturbations instead make assumptions about the causal structure, either through explicit modeling (Schneider et al., 2025) or by incorporating known GRNs (Roohani et al., 2024). Single-cell foundation models (Theodoris et al., 2023; Cui et al., 2024; Hao et al., 2024) have also been applied to perturbation prediction, but perform effect analysis via representation probing rather than generating post-perturbation distributions. Our work targets generalization to unseen biological contexts with a purely data-driven approach that requires no knowledge about the underlying GRN.

Amortized and In-context Learning Rather than optimizing per task, amortized methods learn to perform inference in a single forward pass conditioned on a task context. This context can take the form of the whole dataset for causal structure learning (Lorch et al., 2022; Ke et al., 2023; Dhir et al., 2025) or an input distribution for OT (Amos et al., 2023; Klein et al., 2024) or generative modeling (Atanackovic et al., 2025). Exemplified by large language models (Brown et al., 2020), in-context learning (ICL) achieves amortization by conditioning on example tasks in the input sequence. Recent evidence shows that next-token prediction alone can induce causal discovery and counterfactual reasoning in transformers (Butkus & Kriegeskorte, 2025). Concurrent to our work, Dong et al. (2026) apply ICL to single-cell perturbation prediction. Unlike our approach, they limit the interventional context set to a single experiment and do not use a synthetic prior for pretraining.

Prior-data Fitted Networks Prior-data fitted networks (PFNs) are pretrained on synthetic datasets to perform Bayesian inference in context (Müller et al., 2022). The datasets are generated from a pre-defined generative process, also referred to as the prior. PFNs have recently surpassed classical methods in tabular prediction benchmarks (Hollmann et al., 2025) and were applied to other problems, including causal inference (Balazadeh et al., 2025; Robertson et al., 2025; Ma et al., 2025), full Bayesian inference (Reuter et al., 2025) and optimization (Müller et al., 2023). Yet, contrary to our approach, existing PFNs for causal inference only predict univariate outcomes for individual samples rather than population-level distributions, rendering them incapable of handling perturbation data. In addition, they focus on learning from observational data alone and do not condition predictions on interventional data.

4 METHODS

Primer on Prior-data Fitted Networks In a classical supervised machine learning setting with a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, Bayesian inference assumes a prior $p(\psi)$ representing a space of hypotheses (e.g. structural causal models) that could have generated the data. The aim of PFNs is to approximate the posterior predictive distribution (PPD) $p(y \mid \mathbf{x}, \mathcal{D})$. Given a complete training dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ and an unlabeled query $\mathbf{x}_q$ from the test set, a PFN directly outputs the predicted label y_q. Since the learning process happens in the context of a transformer within a single forward pass, this process is regarded as in-context learning or amortized Bayesian inference. Training PFNs involves sampling a large number of hypotheses $\psi \sim p(\psi)$ and generating synthetic datasets $\mathcal{D} \sim p(\mathcal{D} \mid \psi)$ in an outer loop to meta-learn how to make predictions in context. Therefore, no gradient-based optimization is needed for predictions in unseen datasets. We refer to Müller et al. (2022) and Hollmann et al. (2023) for further details.

Structural Causal Models To study perturbation prediction in a controlled environment, we set up a synthetic experiment where the true data-generating process is known. A structural causal model (SCM) ψ (Pearl, 2009) defines a generative model through a directed acyclic graph (DAG) $\mathcal{G}_\psi$ over variables $\{z_1, z_2, ..., z_d\}$, together with structural assignment $z_k = f_k(z_{\mathrm{PA}(k)}, \epsilon_k)$ for each node z_k, where $z_{\mathrm{PA}(k)}$ denotes the parents of z_k in $\mathcal{G}_\psi$, f_k is a deterministic function, and ϵ_k is an exogenous noise variable. Following the rules of do-calculus (Pearl, 2009), a hard intervention $\mathrm{do}(t)$ on node z_k removes its incoming edges and assigns $z_k := t$, yielding $\psi^{\mathrm{do}(t)}$. Specifically, we consider linear additive noise models (ANMs); a class of SCMs with linear functional relationships f_k and additive noise. In this case, the model is fully determined by a sparse weighted adjacency matrix $\mathbf{W} \in \mathbb{R}^{d \times d}$, where $w_{kj} \neq 0$ only if $j \in \mathrm{PA}(k)$. Given a noise vector $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$, we can sample from linear ANMs by solving the linear system $\mathbf{z} = (\mathbf{I} - \mathbf{W})^{-1}\boldsymbol{\epsilon}$ (Pearl, 2009).

Transductive Perturbation Prediction In practice, the true underlying causal structure of a given biological context is unknown and only partially identifiable from finite data. Hence, we deliberately do not make restrictive assumptions about it or aim to explicitly infer it from data. Inspired by the principle of *transduction* (Vapnik, 2006), we instead directly predict the post-perturbation distribution given observational and interventional data.

Prior We pretrain MapPFN on a large number of synthetic datasets generated from a prior distribution over ψ. Previous work on PFNs has shown that even simple synthetic priors can generalize effectively to real-world data (Hollmann et al., 2023). In addition to SCMs, we sample synthetic GRNs to simulate observational and interventional perturbation data, as we will outline in subsection 5.1.

Modeling Assumptions We assume the observations $\mathbf{Y}^{\mathrm{obs}}$ are generated by a latent SCM ψ, representing the GRN of the given cell population. In this case, a gene knockout perturbation can be modeled as a hard intervention $\mathrm{do}(t)$ on a single node of the underlying causal structure, as it affects a single gene in the underlying GRN. We further assume $\mathbf{Y}^{\mathrm{int}}$ to stem from the intervened-upon latent SCM $\psi^{\mathrm{do}(t)}$. We consider a limited set of atomic interventions $t \in \mathcal{T}$, corresponding to the set of all knocked-out genes in our dataset, and assume all variables of the latent SCM are observed, i.e. the expression levels of all perturbed genes are measured.

Given observational samples $\mathbf{Y}^{\mathrm{obs}}$ and a set of interventional experiments $\mathcal{C} = \left\{\left(t_k, \mathbf{Y}_k^{\mathrm{int}}\right)\right\}_{k=1}^K$ for ψ, we aim to directly predict the post-perturbation distribution of an unseen query treatment $t_q \in \mathcal{T} \setminus \mathcal{T}_\mathcal{C}$. Based on our assumptions, the posterior predictive distribution takes the form

$$p(\mathbf{y}_q^{\mathrm{int}} \mid \mathrm{do}(t_q), \mathbf{Y}^{\mathrm{obs}}, \mathcal{C}) = \int p(\mathbf{y}_q^{\mathrm{int}} \mid \mathrm{do}(t_q), \mathbf{Y}^{\mathrm{obs}}, \psi)\, p(\psi \mid \mathbf{Y}^{\mathrm{obs}}, \mathcal{C})\, d\psi \tag{2}$$

We refer to Robertson et al. (2025) for a theoretical discussion of the sources of uncertainty in this formulation.

Pretraining Process During each pretraining step, we first sample an SCM $\psi \sim p(\psi)$ from our prior. By propagating noise $\mathbf{N} = [\boldsymbol{\epsilon}_1, ..., \boldsymbol{\epsilon}_n]^\top, \boldsymbol{\epsilon}_i \sim \mathcal{N}(0, \mathbf{I})$ through the SCM, we obtain the observational distribution $\mathbf{Y}^{\mathrm{obs}}$. Subsequently, we build the context $\mathcal{C} = \{(t_k, \mathbf{Y}_k^{\mathrm{int}})\}_{k=1}^K$ by sampling SCMs $\psi^{\mathrm{do}(t_k)}$ for a subset of treatments $t_k \in \mathcal{T}_\mathcal{C} \subset \mathcal{T}$. For each intervention in this set, we generate

post-perturbation distributions $\mathbf{Y}_k^{\text{int}}$ by drawing a new noise matrix $\mathbf{N}_k$. Finally, our prediction target is the post-perturbation distribution $\mathbf{Y}_q^{\text{int}}$ arising from an unseen query treatment $t_q \in \mathcal{T} \setminus \mathcal{T}_{\mathcal{C}}$. Figure 1 provides an overview of MapPFN pretraining and inference. The full pretraining process is outlined in Appendix A, Algorithm 1.

Identifiability Perturbation prediction depends on identifiability, i.e. the extent to which the causal graph $\mathcal{G}_\psi$ can be inferred from data, even if it is not explicitly recovered. Interventional data can fully identify the causal graph given sufficient interventions (Eberhardt et al., 2006). Conditioning on an interventional context $\mathcal{C}$ reduces the Markov equivalence class $[\mathcal{G}_\psi]$, as each intervention constrains the set of causal structures consistent with the data (Hauser & Bühlmann, 2012). This provides MapPFN with a theoretical advantage over existing causal PFNs and perturbation models that learn from observational data alone, assuming the true causal graph lies within the support of the prior distribution $p(\psi)$.

4.1 Model

We adopt the Multimodal Diffusion Transformer (MMDiT) architecture (Esser et al., 2024) with minor modifications. We treat cells as tokens, and input noise, cell states, and one-hot encoded treatments are processed as three modality streams with separate parameters. Cross-modal interactions are enabled via joint attention.

Because the inputs are unordered sets of cells, we remove sinusoidal positional encodings and rely on the permutation invariance of attention. Instead, we introduce learnable embeddings to differentiate modalities, query versus context, and observational versus interventional data. We train MapPFN using a conditional flow matching objective with an affine Gaussian probability path (Lipman et al., 2023)

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{\tau, \mathbf{Y}_0, \mathbf{Y}_q^{\text{int}}} \left\| v_\tau^\theta(\mathbf{Y}_\tau \mid t_q, \mathbf{Y}^{\text{obs}}, \mathcal{C}) - (\mathbf{Y}_q^{\text{int}} - \mathbf{Y}_0) \right\|_{\text{F}}^2 \tag{3}$$

where $\tau \sim \text{LogitNormal}(0,1)$, $\mathbf{Y}_0 \sim \mathcal{N}(0, \mathbf{I})$ and $\mathbf{Y}_\tau = (1-\tau)\mathbf{Y}_0 + \tau \mathbf{Y}_q^{\text{int}}$. Please refer to Appendix A and Algorithm 1 for details on the model architecture and pretraining process.

5 Experimental Setup

5.1 Datasets

Linear Structural Causal Models We generate synthetic data from linear structural causal models (SCM) with additive Gaussian noise (Pearl, 2009). We sample directed acyclic graphs (DAGs) from an Erdős–Rényi distribution (Erdős & Rényi, 1960) with $d = 20$ nodes and an edge probability of $p = 0.5$. Additional details on the linear SCM data are provided in Appendix B.1.

Biological Prior To generate our prior dataset of synthetic single-cell data, we use the GRN generator by Aguirre et al. (2025) and simulate expression data from generated regulatory networks using SERGIO (Dibaeinia & Sinha, 2020). We generate directed graphs from a scale-free distribution using the preferential attachment algorithm by Aguirre et al. (2025), allowing to generate networks with similar properties to real GRNs in terms of modularity, sparsity and degree distributions (see Appendix B.2).

Given a sampled regulatory network, we simulate single-cell gene expressions using SERGIO (Dibaeinia & Sinha, 2020), which models cell expressions as the steady state of a system of Stochastic Differential Equations (SDEs). Regulatory interactions are parameterized by Hill functions (Gesztelyi et al., 2012), capturing nonlinear and saturation effects. Genetic perturbations are performed in-silico by removing the perturbed gene from the regulatory network and re-simulating the system. To obtain gene expression counts, we apply the technical noise model by Dibaeinia & Sinha (2020) for 10x Chromium single-cell RNA sequencing. Additional details on the single-cell prior and its hyperparameters are provided in Appendix B.2.

Single-cell Data Beyond the SCM setting, we evaluate whether a model pretrained exclusively on synthetic data generalizes to a real single-cell perturbation dataset (Frangieh et al., 2021). This dataset consists of a CRISPR-Cas9 perturbation screen with scRNA-seq readouts in patient-derived melanoma

cells, profiling knockouts of 248 genes involved in an immune evasion program associated with resistance to immunotherapy. In total, 218,000 single-cell expressions were measured across three biological contexts with varying activation of immune response pathways: untreated control, IFN-γ stimulation and a co-culture with tumor-infiltrating lymphocytes (TIL). To reduce dimensionality, we follow Schneider et al. (2025) and restrict analysis to the 50 genes with the strongest perturbation effects. For evaluation, we use the IFN-γ context as the test set. Additional details on the data are provided in Appendix B.3.

Data Split We distinguish two evaluation settings for the held-out biological context. In the *few-shot* setting, only specific perturbations of the test context are withheld from training, but other perturbations of the context can appear in training. The few-shot setting follows the Virtual Cell Challenge (Roohani et al., 2025), where adaptation to a new biological context is based on a limited number of interventional experiments. In the *zero-shot* setting, all perturbations are withheld from training.

For linear SCM experiments, we train all methods including MapPFN on the same synthetic data. For the single-cell experiments, MapPFN is trained exclusively on synthetic data and evaluated on real perturbations, while baselines are trained on real single-cell data, as they do not admit a similar pretraining phase. In the few-shot setting, baselines are trained on perturbations from the holdout context, whereas MapPFN only observes them at inference time without parameter updates. Additional details on the data split are provided in Appendix B.4.

5.2 Baselines

We compare our method against Conditional Optimal Transport (CondOT) (Bunne et al., 2022) and Meta Flow Matching (MetaFM) (Atanackovic et al., 2025). MetaFM conditions not only on the treatment but also on a learned representation of the observational distribution obtained from a graph neural network (GNN). As lower and upper bounds, we report two reference baselines following Bunne et al. (2023): an identity baseline that predicts the observational distribution $\hat{\mathbf{y}}^{\text{int}} \sim p(\mathbf{y}^{\text{obs}})$, and an oracle baseline that uses the observed post-perturbation distribution $\hat{\mathbf{y}}^{\text{int}} \sim p(\mathbf{y}^{\text{int}})$. Additional details on the baselines are provided in Appendix C.

5.3 Metrics

We evaluate perturbation prediction by comparing the predicted post-perturbation distribution $\hat{\mathbf{Y}}^{\text{int}}$ to the ground-truth distribution $\mathbf{Y}^{\text{int}}$ in terms of distributional similarity, moment-level accuracy and biological signal recovery. Distributional similarity is quantified using the entropy-regularized Wasserstein distance (W_2) (Cuturi, 2013) and the Maximum Mean Discrepancy (MMD) (Gretton et al., 2012). Moment-level accuracy is measured by the root mean squared error (RMSE) between the predicted and ground-truth distribution means. To assess whether predictions are distinguishable across perturbations, we report the transposed rank (Rank$^\top$) (Wu et al., 2025b). We introduce the *magnitude ratio* (MagRatio), which normalizes the predicted effect size by the true intervention effect to quantify effect recovery independent of scale (see Appendix D).

Biological relevance is evaluated using the area under the precision-recall curve (AUPRC) (Zhu et al., 2025). This metric compares differentially expressed genes (DEGs) inferred *in silico* from the predicted post-perturbation distribution with those observed in the ground-truth data. Additional details on the metrics are provided in Appendix D.

6 Results

Magnitude ratio uncovers identity collapse as common failure mode in baselines Table 1 evaluates perturbation prediction performance in linear SCMs under the few-shot setting. Across all metrics (MMD, RMSE, Rank$^\top$ and MagRatio), MapPFN most accurately recovers the post-perturbation distribution compared to CondOT and MetaFM. Notably, MapPFN achieves the lowest transposed rank, indicating that it is less prone to mode collapse. All methods outperform the identity baseline.

Table 1: **Evaluation within a prior of linear SCMs in the few-shot setting.** Test metrics for the SCM dataset in the few-shot setting, measuring similarity between the predicted and ground-truth post-perturbation distribution. We aggregate all metrics over three random seeds (mean ± std). Bold indicates results within one standard deviation of the best. MapPFN shows strong performance across metrics.

Method	W_2 (↓)	MMD (↓)	RMSE (↓)	Rank$^\top$ (↓)	MagRatio
CondOT	13.85 ± 0.12	5.14 ± 0.01	0.15 ± 0.00	0.11 ± 0.02	0.09 ± 0.01
MetaFM	**13.73** ± 0.16	4.81 ± 0.19	0.15 ± 0.01	0.09 ± 0.03	0.12 ± 0.00
MapPFN (ours)	**13.69** ± 0.05	**4.28** ± 0.06	**0.14** ± 0.00	**0.01** ± 0.01	**0.99** ± 0.02
Identity	17.61 ± 0.14	12.98 ± 0.35	0.28 ± 0.01	0.49 ± 0.01	0.00 ± 0.00
Observed	9.82 ± 0.08	3.66 ± 0.06	0.07 ± 0.00	0.00 ± 0.00	1.00 ± 0.00

Table 2: **Comparison of MapPFN pretrained on synthetic prior to baselines trained on real single-cell data in the few-shot setting.** Test metrics were aggregated over three random seeds (mean ± std). Bold indicates results within one standard deviation of the best. MapPFN identifies differentially expressed genes, matching performance of baselines trained on real single-cell data.

Method	W_2 (↓)	MMD (↓)	RMSE (↓)	Rank$^\top$ (↓)	MagRatio	AUPRC (↑)
CondOT	22.15 ± 0.33	**7.01** ± 0.28	**0.09** ± 0.01	**0.06** ± 0.03	0.05 ± 0.00	**0.34** ± 0.03
MetaFM	**21.12** ± 0.94	**7.27** ± 0.28	**0.09** ± 0.00	**0.08** ± 0.01	0.12 ± 0.00	0.29 ± 0.03
MapPFN (ours)	**21.94** ± 0.80	11.38 ± 1.37	0.14 ± 0.01	0.13 ± 0.03	**0.99** ± 0.02	**0.35** ± 0.02
Identity	22.57 ± 0.08	6.63 ± 0.10	0.11 ± 0.00	0.51 ± 0.02	0.00 ± 0.00	0.05 ± 0.01
Observed	7.30 ± 0.08	1.40 ± 0.02	0.03 ± 0.00	0.00 ± 0.00	1.00 ± 0.00	0.78 ± 0.09
MapPFN (zero-shot)	23.40 ± 0.23	16.72 ± 0.80	0.20 ± 0.01	0.26 ± 0.02	1.08 ± 0.01	0.17 ± 0.01
MapPFN (unpaired)	27.89 ± 6.36	31.24 ± 12.10	0.32 ± 0.10	0.29 ± 0.09	1.31 ± 0.29	0.12 ± 0.06

We observe a similar pattern in the zero-shot setting (Appendix G, Table 6). Across datasets and evaluation settings, MapPFN is the only method with a magnitude ratio close to one, indicating that the predicted causal effect matches the observed effect in scale. In contrast, CondOT and MetaFM yield magnitude ratios around 0.1, suggesting little deviation from the observational distribution. We attribute this behavior to both baselines either initializing the generative flow to the observational distribution or initializing the model weights as an identity map.

Synthetic pretraining recovers differentially expressed genes in real perturbation data Table 2 reports perturbation prediction performance for baselines trained on real single-cell data and for MapPFN, which was trained exclusively on a synthetic prior. MapPFN achieves performance comparable to MetaFM in terms of Wasserstein distance, while CondOT and MetaFM outperform MapPFN on MMD, RMSE and Rank$^\top$. This performance gap is consistent with a residual mismatch between the synthetic prior and real single-cell distributions, though MapPFN substantially outperforms the identity baseline on transposed rank. Beyond distributional similarity metrics, a key question for biological applications is whether MapPFN can identify which genes are differentially expressed. This information is critical for understanding treatment mechanisms and planning interventions.

Figure 2 reports the precision-recall curves for recovering differentially expressed genes (DEGs) from the predicted post-perturbation distribution (see Appendix D). Besides a random predictor, we include a *target-only* baseline, always predicting that only the knocked-out gene is differentially expressed. MapPFN achieves the highest AUPRC and is the only method that consistently outperforms the *target-only* baseline. While CondOT and MapPFN achieve better precision than MetaFM at recall above 0.25, both CondOT and MetaFM achieve lower precision than MapPFN at recall levels below 0.25. Notably, MetaFM achieves strong distributional metrics but the worst AUPRC, suggesting that distributional similarity does not necessarily reflect performance on biologically relevant downstream tasks. These results indicate that MapPFN is capable of identifying differentially expressed genes, on par with baselines trained on real-world data. At the same time, MapPFN only requires sampling from the pretrained generative model, while existing methods need to be trained from scratch. Additional results for the zero-shot setting can be found in Appendix G, Table 7.

Learning from interventional experiments in context improves performance over observational data alone We evaluate whether MapPFN benefits from improved identifiability by conditioning on interventional data. Specifically, we ablate the effect of providing a set of interventional distributions $\mathcal{C}$ versus the zero-shot setting, where $\mathcal{C} = \emptyset$. As shown in Table 2, conditioning on interventional distributions consistently improves performance over using only observational data and a treatment identifier alone. Since the model architecture is held fixed, this gain can be attributed to the interventional context rather than architectural differences. We ablate the effect of the context size $K = |\mathcal{C}|$ in Appendix G, Figure 4, finding that performance improves monotonically with additional perturbation experiments, with diminishing returns beyond four. Because the zero-shot setting requires training baselines on a reduced dataset that excludes all perturbations of the hold-out context, we report their zero-shot performance separately in Appendix G, Table 7.

7 Discussion

We introduced MapPFN, a prior-data fitted network that frames perturbation prediction as an in-context learning problem. By pretraining exclusively on synthetic perturbation data, MapPFN is not limited by the number of experimentally measured biological contexts, adapting to real-world data without gradient-based optimization. In the following, we will recapitulate our findings, discuss limitations of our approach and give a brief outlook.

We find that conditioning on interventional experiments improves prediction of unseen perturbations compared to methods that condition only on treatment identity or the observational distribution. We attribute this to improved identifiability, as the interventional context helps MapPFN to reduce the Markov equivalence class of the underlying causal mechanisms. Despite being trained exclusively on synthetic perturbations, MapPFN recovers differentially expressed genes in real single-cell data on par with baselines trained on real perturbations, suggesting that domain-specific simulators can provide sufficient inductive bias for transfer to real-world settings. We identify identity collapse as a common failure mode in existing methods, evidenced by magnitude ratios near zero.

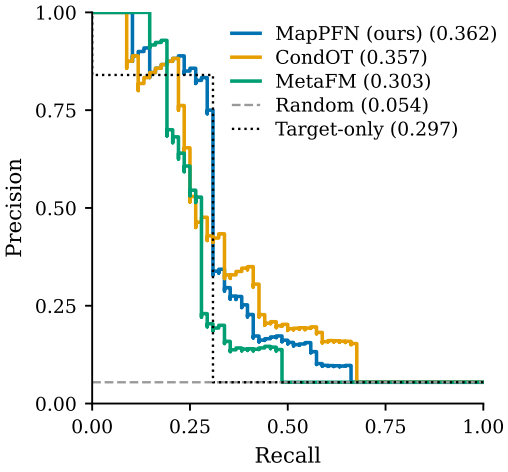

Figure 2: **Recovery of differentially expressed genes in real single-cell data (Frangieh et al., 2021).** Precision-recall curves and AUPRC for identification of $n = 68$ differentially expressed genes in the held-out IFN-γ context. For each method, we report the model with the median AUPRC across three seeds. Despite being pretrained exclusively on synthetic data, MapPFN achieves competitive AUPRC compared to baselines trained on real perturbations from the held-out context.

Limitations Despite encouraging generalization to real-world single-cell data, we note that the ability of MapPFN to generalize to unseen biological contexts largely depends on our synthetic biological prior. Future work should investigate to which extent counterfactual priors may be preferable and how to systematically evaluate them. While MapPFN can adapt to any set of genes, scaling to larger input dimensions offers a promising direction (Kolberg et al., 2025), as the current version only permits focusing on a single regulatory mechanism. While MapPFN adapts to new datasets at inference time in minutes, pretraining on synthetic data requires 36 GPU hours upfront. Finally, extending MapPFN to support non-atomic drug-based or chemical perturbations remains an open challenge (Schneider et al., 2025; Dong et al., 2026; Wu et al., 2025a).

Outlook Given the success of PFNs in tabular prediction and causal inference, we anticipate that scaling MapPFN in terms of parameters and synthetic data will yield further improvements. Next to generating larger perturbation datasets experimentally, our findings point toward scaling pretraining on synthetic biological priors as a complementary path toward context-adaptive virtual cell foundation models.

ACKNOWLEDGEMENTS

The authors would like to thank Michael Plainer, Jonas Loos and Alexander Möllers for the fruitful discussions and helpful input.

REFERENCES

Abhinav K. Adduri, Dhruv Gautam, Beatrice Bevilacqua, Alishba Imran, Rohan Shah, Mohsen Naghipourfar, Noam Teyssier, Rajesh Ilango, Sanjay Nagaraj, Mingze Dong, Chiara Ricci-Tam, Christopher Carpenter, Vishvak Subramanyam, Aidan Winters, Sravya Tirukkovular, Jeremy Sullivan, Brian S. Plosky, Basak Eraslan, 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. *bioRxiv:10.1101/2025.06.26.661135*, 2025.

Matthew Aguirre, Jeffrey P. Spence, Guy Sella, and Jonathan K. Pritchard. Gene regulatory network structure informs the distribution of perturbation effects. *PLOS Computational Biology*, 21(9): 1–31, 2025.

Brandon Amos, Lei Xu, and J. Zico Kolter. Input convex neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 146–155, 2017.

Brandon Amos, Giulia Luise, Samuel Cohen, and Ievgen Redko. Meta optimal transport. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 791–813, 2023.

Lazar Atanackovic, Xi (Nicole) 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*, volume 2025, pp. 94586–94610, 2025.

Vahid Balazadeh, Hamidreza Kamkari, Valentin Thomas, Benson Li, Junwei Ma, Jesse C. Cresswell, and Rahul G. Krishnan. CausalPFN: Amortized causal effect estimation via in-context learning. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Yoav Benjamini and Yosef Hochberg. On the adaptive control of the false discovery rate in multiple testing with independent statistics. *Journal of Educational and Behavioral Statistics*, 25(1):60–83, 2000.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901, 2020.

Charlotte Bunne, Andreas Krause, and Marco Cuturi. Supervised Training of Conditional Monge Maps. In *Advances in Neural Information Processing Systems*, volume 35, pp. 6859–6872, 2022.

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(11):1759–1768, 2023.

Charlotte Bunne, Yusuf Roohani, Yanay Rosen, Ankit Gupta, Xikun Zhang, Marcel Roed, Theo Alexandrov, Mohammed AlQuraishi, Patricia Brennan, Daniel B. Burkhardt, Andrea Califano, Jonah Cool, Abby F. Dernburg, Kirsty Ewing, Emily B. Fox, Matthias Haury, Amy E. Herr, Eric

Horvitz, Patrick D. Hsu, Viren Jain, Gregory R. Johnson, Thomas Kalil, David R. Kelley, Shana O. Kelley, Anna Kreshuk, Tim Mitchison, Stephani Otte, Jay Shendure, Nicholas J. Sofroniew, Fabian Theis, Christina V. Theodoris, Srigokul Upadhyayula, Marc Valer, Bo Wang, Eric Xing, Serena Yeung-Levy, Marinka Zitnik, Theofanis Karaletsos, Aviv Regev, Emma Lundberg, Jure Leskovec, and Stephen R. Quake. How to build the virtual cell with artificial intelligence: Priorities and opportunities. *Cell*, 187(25):7045–7063, 2024.

Eivinas Butkus and Nikolaus Kriegeskorte. Causal discovery and inference through next-token prediction. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

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.

Marco Cuturi. Sinkhorn Distances: Lightspeed Computation of Optimal Transport. In *Advances in Neural Information Processing Systems*, volume 26, 2013.

Marco Cuturi, Laetitia Meng-Papaxanthos, Yingtao Tian, Charlotte Bunne, Geoff Davis, and Olivier Teboul. Optimal Transport Tools (OTT): A JAX Toolbox for all things Wasserstein. *arXiv:2201.12324*, 2022.

Anish Dhir, Matthew Ashman, James Requeima, and Mark van der Wilk. A meta-learning approach to bayesian causal discovery. In *The Thirteenth International Conference on Learning Representations*, 2025.

Payam Dibaeinia and Saurabh Sinha. SERGIO: A Single-Cell Expression Simulator Guided by Gene Regulatory Networks. *Cell Systems*, 11(3):252–271, 2020.

Atray Dixit, Oren Parnas, Biyu Li, Jenny Chen, Charles P. Fulco, Livnat Jerby-Arnon, Nemanja D. Marjanovic, Danielle Dionne, Tyler Burks, Raktima Raychndhury, 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, 2016.

Mingze Dong, Bao Wang, Jessica Wei, Antonio H. de O. Fonseca, Curtis J. Perry, Alexander Frey, Feriel Ouerghi, Ellen F. Foxman, Jeffrey J. Ishizuka, Rahul M. Dhodapkar, and David van Dijk. Causal identification of single-cell experimental perturbation effects with CINEMA-OT. *Nature Methods*, 20(11):1769–1779, 2023.

Mingze Dong, Abhinav Adduri, Dhruv Gautam, Christopher Carpenter, Rohan Shah, Chiara Ricci-Tam, Yuval Kluger, Dave P. Burke, and Yusuf Husein Roohani. Stack: In-Context Learning of Single-Cell Biology. *bioRxiv:10.64898/2026.01.09.698608*, 2026.

J. R. Dormand and P. J. Prince. A family of embedded Runge-Kutta formulae. *Journal of Computational and Applied Mathematics*, 6(1):19–26, 1980.

Aleksandr Dremov, Alexander Hägele, Atli Kosson, and Martin Jaggi. Training dynamics of the cooldown stage in warmup-stable-decay learning rate scheduler. *Transactions on Machine Learning Research*, 2025.

Frederick Eberhardt, Clark Glymour, and Richard Scheines. N-1 Experiments Suffice to Determine the Causal Relations Among N Variables. In *Innovations in Machine Learning: Theory and Applications*, pp. 97–112. Springer Berlin Heidelberg, 2006.

Pál Erdős and Alfréd Rényi. On the evolution of random graphs. *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, 5(1):17–60, 1960.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12606–12633, 2024.

Jean Feydy, Thibault Séjourné, François-Xavier Vialard, Shun-ichi Amari, Alain Trouve, and Gabriel Peyré. Interpolating between optimal transport and mmd using sinkhorn divergences. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pp. 2681–2690, 2019.

Chris J. Frangieh, Johannes C. Melms, Pratiksha I. Thakore, Kathryn R. Geiger-Schuller, Patricia Ho, Adrienne M. Luoma, Brian Cleary, Livnat Jerby-Arnon, Shruti Malu, Michael S. Cuoco, Maryann Zhao, Casey R. Ager, Meri Rogava, Lila Hovey, Asaf Rotem, Chantale Bernatchez, Kai W. Wucherpfennig, Bruce E. Johnson, Orit Rozenblatt-Rosen, Dirk Schadendorf, Aviv Regev, and Benjamin Izar. Multimodal pooled Perturb-CITE-seq screens in patient models define mechanisms of cancer immune evasion. *Nature Genetics*, 53(3):332–341, 2021.

Aude Genevay, Gabriel Peyre, and Marco Cuturi. Learning generative models with sinkhorn divergences. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pp. 1608–1617, 09–11 Apr 2018.

Rudolf Gesztelyi, Judit Zsuga, Adam Kemeny-Beke, Balazs Varga, Bela Juhasz, and Arpad Tosaki. The Hill equation and the origin of quantitative pharmacology. *Archive for History of Exact Sciences*, 66(4):427–438, 2012.

Daniel T. Gillespie. The chemical Langevin equation. *The Journal of Chemical Physics*, 113(1): 297–306, 2000.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A Kernel Two-Sample Test. *Journal of Machine Learning Research*, 13(25):723–773, 2012.

Minsheng Hao, Jing Gong, Xin Zeng, Chiming Liu, Yucheng Guo, Xingyi Cheng, Taifeng Wang, Jianzhu Ma, Xuegong Zhang, and Le Song. Large-scale foundation model on single-cell transcriptomics. *Nature Methods*, 21(8):1481–1491, 2024.

Alain Hauser and Peter Bühlmann. Characterization and Greedy Learning of Interventional Markov Equivalence Classes of Directed Acyclic Graphs. *Journal of Machine Learning Research*, 13(79): 2409–2464, 2012.

Lukas Heumos, Yuge Ji, Lilly May, Tessa D. Green, Stefan Peidli, Xinyue Zhang, Xichen Wu, Johannes Ostner, Antonia Schumacher, Karin Hrovatin, Michaela Müller, Faye Chong, Gregor Sturm, Alejandro Tejada, Emma Dann, Mingze Dong, Gonçalo Pinto, Mojtaba Bahrami, Ilan Gold, Sergei Rybakov, Altana Namsaraeva, Amir Ali Moinfar, Zihe Zheng, Eljas Roellin, Isra Mekki, Chris Sander, Mohammad Lotfollahi, Herbert B. Schiller, and Fabian J. Theis. Pertpy: an end-to-end framework for perturbation analysis. *Nature Methods*, 2025.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv:2207.12598*, 2021.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. TabPFN: A transformer that solves small tabular classification problems in a second. In *The Eleventh International Conference on Learning Representations*, 2023.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, 2025.

Martin Jinek, Krzysztof Chylinski, Ines Fonfara, Michael Hauer, Jennifer A. Doudna, and Emmanuelle Charpentier. A Programmable Dual-RNA–Guided DNA Endonuclease in Adaptive Bacterial Immunity. *Science*, 337(6096):816–821, 2012.

Nan Rosemary Ke, Silvia Chiappa, Jane X Wang, Jorg Bornschein, Anirudh Goyal, Melanie Rey, Theophane Weber, Matthew Botvinick, Michael Curtis Mozer, and Danilo Jimenez Rezende. Learning to induce causal structure. In *International Conference on Learning Representations*, 2023.

Patrick Kidger. On Neural Differential Equations. *arXiv:2202.02435*, 2022.

Patrick Kidger and Cristian Garcia. Equinox: neural networks in JAX via callable PyTrees and filtered transformations. *Differentiable Programming workshop at Neural Information Processing Systems 2021*, 2021.

Dominik Klein, Théo Uscidda, Fabian Theis, and Marco Cuturi. GENOT: Entropic (Gromov) Wasserstein Flow Matching with Applications to Single-Cell Genomics. In *Advances in Neural Information Processing Systems*, volume 37, pp. 103897–103944, 2024.

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:10.1101/2025.04.11.648220*, 2025.

Christopher Kolberg, Katharina Eggensperger, and Nico Pfeifer. TabPFN-Wide: Continued Pre-Training for Extreme Feature Counts. *arXiv:2510.06162*, 2025.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023.

Lars Lorch, Scott Sussex, Jonas Rothfuss, Andreas Krause, and Bernhard Schölkopf. Amortized Inference for Causal Structure Learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 13104–13118, 2022.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Mohammad Lotfollahi, F. Alexander Wolf, and Fabian J. Theis. scGen predicts single-cell perturbation responses. *Nature Methods*, 16(8):715–721, 2019.

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, Jose 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), 2023.

Malte D. Luecken and Fabian J. Theis. Current best practices in single-cell RNA-seq analysis: a tutorial. *Molecular Systems Biology*, 15(6), 2019.

Yuchen Ma, Dennis Frauen, Emil Javurek, and Stefan Feuerriegel. Foundation Models for Causal Inference via Prior-Data Fitted Networks. *arXiv:2506.10914*, 2025.

Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do bayesian inference. In *International Conference on Learning Representations*, 2022.

Samuel Müller, Matthias Feurer, Noah Hollmann, and Frank Hutter. PFNs4BO: In-context learning for Bayesian optimization. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 25444–25470, 2023.

Judea Pearl. *Causality*. Cambridge University Press, 2009.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual Reasoning with a General Conditioning Layer. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), 2018.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. TabICL: A tabular foundation model for in-context learning on large data. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 50817–50847. PMLR, 2025.

Alexander Reisach, Christof Seiler, and Sebastian Weichwald. Beware of the Simulated DAG! Causal Discovery Benchmarks May Be Easy to Game. In *Advances in Neural Information Processing Systems*, volume 34, pp. 27772–27784, 2021.

Arik Reuter, Tim G. J. Rudner, Vincent Fortuin, and David Rügamer. Can transformers learn full Bayesian inference in context? In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 51531–51582, 2025.

Jake Robertson, Arik Reuter, Siyuan Guo, Noah Hollmann, Frank Hutter, and Bernhard Schölkopf. Do-PFN: In-Context Learning for Causal Effect Estimation. In *Advances in Neural Information Processing Systems*, volume 39, 2025.

Yusuf Roohani, Kexin Huang, and Jure Leskovec. Predicting transcriptional outcomes of novel multigene perturbations with GEARS. *Nature Biotechnology*, 42(6):927–935, 2024.

Yusuf H. Roohani, Tony J. Hua, Po-Yuan Tung, Lexi R. Bounds, Feiqiao B. Yu, Alexander Dobin, Noam Teyssier, Abhinav Adduri, Alden Woodrow, Brian S. Plosky, Reshma Mehta, Benjamin Hsu, Jeremy Sullivan, Chiara Ricci-Tam, Nianzhen Li, Julia Kazaks, Luke A. Gilbert, Silvana Konermann, Patrick D. Hsu, Hani Goodarzi, and Dave P. Burke. Virtual Cell Challenge: Toward a Turing test for the virtual cell. *Cell*, 188(13):3370–3374, 2025.

Karen Sachs, Omar Perez, Dana Pe'er, Douglas A. Lauffenburger, and Garry P. Nolan. Causal Protein-Signaling Networks Derived from Multiparameter Single-Cell Data. *Science*, 308(5721): 523–529, 2005.

Nora Schneider, Lars Lorch, Niki Kilbertus, Bernhard Schölkopf, and Andreas Krause. Generative intervention models for causal perturbation modeling. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 53388–53412, 2025.

Ryan Soklaski, Justin Goodwin, Olivia Brown, Michael Yee, and Jason Matterer. Tools and Practices for Responsible AI Engineering. *arXiv:2201.05647*, 2022.

Christina V. Theodoris, Ling Xiao, Anant Chopra, Mark D. Chaffin, Zeina R. Al Sayed, Matthew C. Hill, Helene Mantineo, Elizabeth M. Brydon, Zexian Zeng, X. Shirley Liu, and Patrick T. Ellinor. Transfer learning enables predictions in network biology. *Nature*, 618(7965):616–624, 2023.

Vladimir Vapnik. *Estimation of Dependences Based on Empirical Data*. Springer, 2006.

Isaac Virshup, Sergei Rybakov, Fabian J. Theis, Philipp Angerer, and F. Alexander Wolf. anndata: Access and store annotated data matrices. *Journal of Open Source Software*, 9(101):4371, 2024.

Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945.

F. Alexander Wolf, Philipp Angerer, and Fabian J. Theis. SCANPY: large-scale single-cell gene expression data analysis. *Genome Biology*, 19(1):15, 2018.

Menghua Wu, Umesh Padia, Sean H. Murphy, Regina Barzilay, and Tommi Jaakkola. Identifying biological perturbation targets through causal differential networks. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 67537–67561, 2025a.

Yan Wu, Esther Wershof, Sebastian M. Schmon, Marcel Nassar, Błażej Osiński, Ridvan Eksi, Zichao Yan, Rory Stark, Kun Zhang, and Thore Graepel. PerturBench: Benchmarking Machine Learning Models for Cellular Perturbation Analysis. In *Advances in Neural Information Processing Systems*, volume 39, 2025b.

Jesse Zhang, Airol A. Ubas, Richard de Borja, Valentine Svensson, Nicole Thomas, Neha Thakar, Ian Lai, Aidan Winters, Umair Khan, Matthew G. Jones, John D. Thompson, Vuong Tran, Joseph Pangallo, Efthymia Papalexi, Ajay Sapre, Hoai Nguyen, Oliver Sanderson, Maria Nigos, Olivia Kaplan, Sarah Schroeder, Bryan Hariadi, Simone Marrujo, Crina Curca Alec Salvino, Guillermo Gallareta Olivares, Ryan Koehler, Gary Geiss, Alexander Rosenberg, Charles Roco, Daniele Merico, Nima Alidoust, Hani Goodarzi, and Johnny Yu. Tahoe-100M: A Giga-Scale Single-Cell Perturbation Atlas for Context-Dependent Gene Function and Cellular Modeling. *bioRxiv:10.1101/2025.02.20.639398*, 2025.

Hongxu Zhu, Amir Asiaee, Leila Azinfar, Jun Li, Han Liang, Ehsan Irajizad, Kim-Anh Do, and James P Long. AUPRC: a metric for evaluating the performance of in-silico perturbation methods in identifying differentially expressed genes. *Briefings in Bioinformatics*, 26(5):bbaf426, 2025.

A Model

A.1 Architecture

We build on the Multi-modal Diffusion Transformer (MMDiT) architecture from the Stable Diffusion 3 family (Esser et al., 2024). Instead of text and image modalities, we keep the denoising process, the pre- and post-treatment data as well as the treatment in three modality streams. With this setup, each modality has separate weights and information flows between modalities via joint attention. As we are working with sets of cells, we use the permutation invariance of the attention mechanism by removing the sinusoid positional encoding. Instead, we add learnable embeddings (a) for each treatment in the context to tell apart different conditions, (b) to tell apart observational and interventional data, and (c) to tell apart the query condition from the context. Our model has 8 layers with an embedding dimension of 256 and a $2\times$ expansion to 512 in the feed-forward layers. We append 8 register tokens to the noise stream and use 4 multi-head attention heads of size 64 each. Time conditioning is implemented by Feature-wise Linear Modulation (FiLM) (Perez et al., 2018). Overall, this configuration amounts to approximately 25M trainable parameters.

A.2 Training

We train our model using a flow matching (Lipman et al., 2023) objective with an affine Gaussian probability path. During training, we randomly drop the condition by replacing it with a learnable `null` embedding with probability $p = 0.2$. Following Esser et al. (2024), we sample $t \sim \text{LogitNormal}(0, 1)$. We use the AdamW optimizer (Loshchilov & Hutter, 2019) with a warmup-stable-decay learning rate schedule (Dremov et al., 2025) using 1% of the total number of steps for warmup to a peak learning rate of 0.0001 and 20% for a square root decay. We maintain an exponential moving average (EMA) of model weights with a decay of 0.999 and use these weights for inference (Esser et al., 2024).

Algorithm 1 MapPFN Pretraining

Input: prior $p(\psi)$, treatments $\mathcal{T}$, context size K
for $i = 1, 2, \ldots, N$ **do**
  Draw SCM $\psi \sim p(\psi)$
  Draw observational samples $\mathbf{Y}^{\text{obs}} \sim p(\mathbf{y}^{\text{obs}} \mid \psi)$
  Draw context treatments $\mathcal{T}_\mathcal{C} \subset \mathcal{T}$ with $|\mathcal{T}_\mathcal{C}| = K$
  for $k = 1, \ldots, K$ **do**
    Draw $\mathbf{Y}_k^{\text{int}} \sim p(\mathbf{y}^{\text{int}} \mid \text{do}(t_k), \psi)$
  end for
  Set context $\mathcal{C} \leftarrow \{(t_k, \mathbf{Y}_k^{\text{int}})\}_{k=1}^K$
  Draw query treatment $t_q \sim \mathcal{T} \setminus \mathcal{T}_\mathcal{C}$
  Draw target $\mathbf{Y}_q^{\text{int}} \sim p(\mathbf{y}^{\text{int}} \mid \text{do}(t_q), \psi)$
  Draw time $\tau \sim \text{LogitNormal}(0, 1)$, $\mathbf{Y}_0 \sim \mathcal{N}(0, \mathbf{I})$
  Compute $\mathcal{L}_{\text{CFM}}(\theta; \mathbf{Y}_0, \tau, \mathbf{Y}_q^{\text{int}}, t_q, \mathbf{Y}^{\text{obs}}, \mathcal{C})$
  Update $\theta \leftarrow \theta - \alpha \nabla \mathcal{L}_{\text{CFM}}(\theta)$
end for

Note: $\mathbf{Y} \sim p(\mathbf{y} \mid \cdot, \psi)$ implies first sampling noise $\mathbf{N} \in \mathbb{R}^{n \times d}$ and stacking n i.i.d. samples.

A.3 Inference

To generate samples, we integrate the learned flow by solving its ordinary differential equation (ODE) using the Dopri5 (Dormand & Prince, 1980) solver, as implemented in `diffrax` (Kidger, 2022). We use classifier-free guidance (Ho & Salimans, 2021) for conditional generation with a guidance weight $\omega = 2.0$ by default.

B Datasets

B.1 Linear Additive Noise Models

Structural Causal Model We generate synthetic observational and interventional data using a linear additive noise model (ANM) with Gaussian noise of the form $\mathbf{z} = \mathbf{W}\mathbf{z} + \boldsymbol{\epsilon}$, where $\mathbf{W} \in \mathbb{R}^{d \times d}$ is a weighted adjacency matrix encoding the causal graph and $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$ represents independent additive noise. The underlying directed acyclic graph (DAG) is sampled from an Erdős-Rényi (Erdős & Rényi, 1960) model $\mathcal{G}(d, p)$ with $d = 20$ nodes and an edge probability of $p = 0.5$, restricted to the upper triangular structure under a random node permutation to ensure acyclicity. Edge weights are sampled uniformly from $[-2, -0.5] \cup [0.5, 2]$, ensuring coefficients are bounded away from zero to exclude negligible causal effects. To ensure observations have approximately unit variance and fall within the $[-2, 2]$ range, we normalize the weight matrix by rescaling $\mathbf{W} \leftarrow \mathbf{D}^{-1/2}\mathbf{W}$ where

$\mathbf{D} = \mathrm{diag}(\mathbf{T}\mathbf{T}^\top)$ and $\mathbf{T} = (\mathbf{I} - \mathbf{W})^{-1}$ denotes the transfer matrix. To avoid varsortability (Reisach et al., 2021), we scale all variables of the generated data to unit variance.

Atomic Interventions Interventional data is generated following Pearl's do-calculus: for an intervention $\mathrm{do}(t)$, we remove all incoming edges to the intervened node and set its value to $c \sim \mathrm{Unif}([0.5, 1.5])$, simulating a gene perturbation experiment where the treated genes have varying perturbation efficiencies. To condition the model on the treatment, we use a d-dimensional one-hot-encoding, where the element at the hot index contains the intervention value c.

Experiment Design We intervene on each of the 20 nodes in 1000 randomly generated DAGs to generate all 20k possible context/treatment conditions. Per treatment condition, we sample $n = 500$ pre-perturbation observations, resulting in 10M interventional vector-valued samples. Additionally, we generate 500 untreated observations per DAG, adding to a total of 10.5M samples. On the linear SCM dataset, all models are trained for 50k steps. For MapPFN, we use a context $\mathcal{C}$ with $K = 4$ perturbation experiments.

B.2 Synthetic Single-Cell Data

To generate synthetic perturbation datasets across diverse contexts, we combine a preferential attachment algorithm for sampling graphs with properties close to real GRNs (Aguirre et al., 2025) and SERGIO (Dibaeinia & Sinha, 2020) for simulating observations from these graphs using Hill functions and adding technical noise.

Gene Regulatory Networks GRNs have unique properties that we want our prior to replicate. As summarized by Aguirre et al. (2025), these properties are (1) sparsity, (2) directed edges and cycles, (3) asymmetry of in- and out-degree distributions and (4) modularity. To ensure our dataset captures the diversity of GRNs, we sample the hyperparameters uniformly from ranges suggested by Aguirre et al. (2025), as summarized in Table 3.

Since SERGIO requires GRNs that are acyclic, we remove cycles by removing the edge with the smallest absolute weight in each cycle. Additionally, SERGIO requires at least one master regulator (MR), i.e. genes with no incoming edges but at least one outgoing edge. If no MRs exist after cycle removal, we select the top 5% of genes with the lowest in-degree among all genes with outgoing edges and remove all incoming edges, forcing them to become MRs.

Simulation Given a regulatory network sampled in the previous step, we simulate single-cell expressions using SERGIO (Dibaeinia & Sinha, 2020). SERGIO models the expression level of each gene as a function of its regulators using Hill functions (Gesztelyi et al., 2012). It then models the gene interaction dynamics by solving a Stochastic Differential Equation (SDE) called chemical Langevin equation (CLE) (Gillespie, 2000). Single-cell expression values are generated by applying technical noise to the steady state of this system. We sample the hyperparameters for the simulation and technical noise uniformly from the ranges summarized in Table 4. For improved simulation speed, we use a reimplementation of SERGIO in Rust[1].

Experiment Design We sample single-cell data in 6000 synthetic GRNs of 50 genes and simulate $n = 200$ single-cells expressions per treatment condition. We train MapPFN for a total of 400k steps and use a context $\mathcal{C}$ containing $K = 8$ perturbation experiments.

B.3 Single-cell Data

For validation on a real-world data, we choose a single-cell RNA-seq dataset containing approximately 218,000 cells measured using Perturb-Seq under 248 CRISPR gene knockout perturbations (Frangieh et al., 2021). Perturbed genes were selected by their membership in an immune evasion program. We obtain the data from the version provided by the `pertpy` (Heumos et al., 2025) package[2]. The knockout perturbations were measured in three patient-derived melanoma cell lines, comprising

[1] https://github.com/rainx0r/sergio_rs

[2] https://pertpy.readthedocs.io/en/stable/api/data/pertpy.data.frangieh_2021_rna.html

Table 3: GRN structure parameters for the graph generator.

Symbol	Description	Range
k	Number of gene groups/modules	$\{1, 2, 3\}$
p	Sparsity term (avg. regulators per gene)	$[1.5, 3.0]$
δ_{in}	In-degree uniformity term	$[10, 300]$
δ_{out}	Out-degree uniformity term	$[1, 30]$
w	Modularity term (within-group connectivity)	$[1, 900]$

Table 4: SERGIO simulation and technical noise parameters.

Symbol	Description	Range
Simulation parameters		
k	Interaction strengths	$[1.0, 5.0]$
b	Master regulator production rates	$[0.5, 2.0] \cup [3.0, 5.0]$
γ	Hill function coefficients (nonlinearity)	$[1.5, 2.5]$
λ	Decay rates per gene	$[0.5, 1.0]$
ζ	Stochastic process noise scale	$[0.5, 1.5]$
Technical noise parameters		
μ_{outlier}	Log-normal outlier mean	$[0.8, 5.0]$
μ_{lib}	Log-normal library size mean	$[4.5, 6.0]$
σ_{lib}	Log-normal library size std	$[0.3, 0.7]$
δ	Dropout percentile	$[8.0, 8.0]$
ξ	Dropout temperature	$[45.0, 82.0]$

one control, one treated with interferon-γ (IFN-γ) to put the cells into an alarmed state and a co-culture treated with tumor infiltrating lymphocytes (TIL) to simulate an immune response. For our experiments, we use the cell line treated with IFN-γ as the hold-out context.

Experiment Design We filter our gene and perturbation set to obtain a complete experiment design including single-cell measurements for all combinations of cell lines and knockout perturbations. To achieve this, we only select perturbations whose target gene was in the set of genes whose expression was measured. Conversely, we only selected those genes that occur in at least one perturbation. This results in a dataset of ∼212,000 cells and 239 genes, significantly reducing the dimensionality from the initial ∼23,700 measured genes. Out of this gene set, we select 50 marker genes by performing differential expression analysis between each perturbation and control using `sc.tl.rank_genes_groups`. To identify genes strongly responsive to perturbation, we select the 50 genes with the highest absolute score across all perturbations. At the median, each (cell line, perturbation) condition contains 216 single cells. Accordingly, we sample $n = 200$ i.i.d. cells per condition to train our models.

Preprocessing Following best practice for single-cell RNA sequencing preprocessing (Luecken & Theis, 2019), we first normalize the total counts per cell to be equal to the median total count across all cells, followed by a `log1p` transform

$$\tilde{\mathbf{x}} = \log_2 \left(1 + \frac{m \cdot \mathbf{x}}{\|\mathbf{x}\|_1} \right) \tag{4}$$

where $m = \text{median}_i(\|\mathbf{x}_i\|_1)$ is the median total count across all cells. We use the implementation of `sc.pp.normalize_total` and `sc.pp.log1p` provided by `scanpy` (Wolf et al., 2018).

B.4 Data Split

We split the data at the condition level, where each condition corresponds to a context-treatment pair (ψ_i, t_j). Each pair is assigned independently to the train, validation, or test split, ensuring that the samples of a particular context/treatment condition are only contained in a single split.

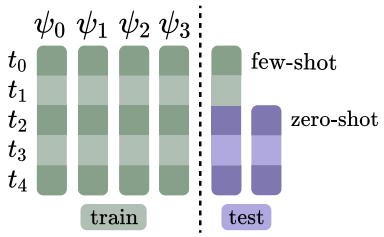

Figure 3: Data split in the few-shot and zero-shot setting. Each box represents a dataset $\mathbf{Y}_{ij}^{\text{int}} \in \mathbb{R}^{N \times d}$ sampled from the SCM ψ_i under treatment t_j. Green boxes are part of the training data and purple boxes are withheld for evaluation. In the few-shot setting, the training data includes interventional distributions from a subset of perturbations in the test context. In the zero-shot setting, no perturbations on the test context are available in the training data. In this setting the model has to recover perturbation effects from observational data alone. The few-shot setting is of practical interest as highlighted by the Virtual Cell Challenge (Roohani et al., 2025).

For both the few-shot and the zero-shot setting, we select a single holdout context. Half of the treatments of this context are assigned to the test split, while the other half is either ignored (zero-shot) or included in the train split (few-shot). This ensures that we can reuse the same test set for both settings. Figure 3 shows a visualization of the two settings. To obtain a validation set, randomly select 10% of the remaining train conditions.

C Baselines

CondOT Conditional Optimal Transport (CondOT) trains a partially input-convex neural network (PICNN) (Amos et al., 2017) to learn a global conditional OT map for different treatment conditions or subpopulations (Bunne et al., 2022). We use the identity initialization, as the Gaussian initialization requires target distribution statistics that are unavailable for unseen contexts. Since CondOT conditions on fixed identifiers that cannot generalize beyond training, we condition only on the one-hot encoded treatment.

Meta Flow Matching Meta Flow Matching (MetaFM) (Atanackovic et al., 2025) proposes to integrate the vector fields on the Wasserstein manifold by conditioning the flow on a learned representation of the observational distribution. With the aim of modeling interactions between individual cells, MetaFM separately trains a graph neural network (GNN) yielding population embeddings. In addition to the initial distribution, MetaFM is conditioned on the one-hot encoded treatment.

D Metrics

We measure the discrepancy between the distribution of predicted samples and the distribution of ground-truth samples. We evaluate our models in terms of distributional, correlation and ranking-based metrics.

Wasserstein Distance The entropy-regularized Wasserstein distance (Cuturi, 2013) between ground-truth samples $\mathbf{Y} \in \mathbb{R}^{n \times d}$ and predicted samples $\hat{\mathbf{Y}} \in \mathbb{R}^{m \times d}$ is computed as

$$W_2(\mathbf{Y}, \hat{\mathbf{Y}}) := \left(\min_{\mathbf{P} \in \mathcal{U}(\mathbf{Y}, \hat{\mathbf{Y}})} \sum_{i=1}^{n} \sum_{j=1}^{m} \mathbf{P}_{ij} ||\mathbf{y}_i - \hat{\mathbf{y}}_j||_2^2 - \epsilon H(\mathbf{P}) \right)^{1/2} \tag{5}$$

where ϵ is the regularization parameter, $\mathcal{U}(\mathbf{Y}, \hat{\mathbf{Y}})$ is the set of transport matrices of shape $n \times m$ given by

$$\mathcal{U} = \left\{ \mathbf{P} \in \mathbb{R}_{\geq 0}^{n \times m} \colon \mathbf{P}\mathbf{1}_m = \frac{1}{n} \cdot \mathbf{1}_n \text{ and } \mathbf{P}^\top \mathbf{1}_n = \frac{1}{m} \cdot \mathbf{1}_m \right\} \tag{6}$$

and H is the entropy computed as $H(\mathbf{P}) = -\sum_{ij} \mathbf{P}_{ij} \log \mathbf{P}_{ij} - 1$. To obtain a valid distance that becomes zero if and only if the compared distributions are equal, we use the Sinkhorn divergence (Feydy et al., 2019; Genevay et al., 2018) given by

$$S_2(\mathbf{Y}, \hat{\mathbf{Y}}) = W_2(\mathbf{Y}, \hat{\mathbf{Y}}) - \frac{1}{2} W_2(\mathbf{Y}, \mathbf{Y}) - \frac{1}{2} W_2(\hat{\mathbf{Y}}, \hat{\mathbf{Y}}) \tag{7}$$

We use the implementation provided by the optimal transport tools (OTT) package (Cuturi et al., 2022) with the regularization parameter $\epsilon = 0.1$.

Maximum Mean Discrepancy The squared Maximum Mean Discrepancy (MMD) (Gretton et al., 2012) between ground truth and predicted samples $\mathbf{Y}$ and $\hat{\mathbf{Y}}$ for a conditionally positive definite kernel k is defined as

$$\text{MMD}^2(\mathbf{Y}, \hat{\mathbf{Y}}) = \mathbb{E}_{\mathbf{y},\mathbf{y}'}[k(\mathbf{y}, \mathbf{y}')] + \mathbb{E}_{\hat{\mathbf{y}},\hat{\mathbf{y}}'}[k(\hat{\mathbf{y}}, \hat{\mathbf{y}}')] - 2\mathbb{E}_{\mathbf{y},\hat{\mathbf{y}}}[k(\mathbf{y}, \hat{\mathbf{y}})] \tag{8}$$

We compute the MMD for the Gaussian radial basis function (RBF) kernel

$$k_{\text{RBF}}(\mathbf{x}, \mathbf{y}) = \exp\left(-\gamma ||\mathbf{x} - \mathbf{y}||_2^2\right) \tag{9}$$

and report the mean over multiple length scales $\gamma \in \{10, 1, 0.1, 0.01, 0.001\}$.

Root Mean Squared Error We follow Wu et al. (2025b) in computing the Root Mean Squared Error (RMSE)

$$\text{RMSE}(\mathbf{Y}, \hat{\mathbf{Y}}) = \sqrt{\frac{1}{n} \sum_{i}^{n} (\hat{\mu}_i - \mu_i)^2} \tag{10}$$

between the mean of the predicted and ground-truth post-perturbation distributions $\boldsymbol{\mu} = \mathbb{E}[\mathbf{y}^{\text{int}}]$ and $\hat{\boldsymbol{\mu}} = \mathbb{E}[\hat{\mathbf{y}}^{\text{int}}]$.

Transposed Rank To evaluate whether model predictions are distinguishable across perturbations, we adopt the transposed rank ($\text{Rank}^\top$) metric from Wu et al. (2025b). Let $\boldsymbol{\mu}_i = \mathbb{E}[\mathbf{y}_i^{\text{int}}]$ and $\hat{\boldsymbol{\mu}}_i = \mathbb{E}[\hat{\mathbf{y}}_i^{\text{int}}]$ denote the mean observed and predicted expression for perturbation i, respectively. The transposed rank metric measures, for each perturbation i, what fraction of other observations $\boldsymbol{\mu}_j$ are closer to $\hat{\boldsymbol{\mu}}_i$ than the matched observation $\boldsymbol{\mu}_i$:

$$\text{Rank}^\top_{\text{avg}} = \frac{1}{p} \sum_{i=1}^{p} \text{Rank}^\top(\hat{\boldsymbol{\mu}}_i), \quad \text{Rank}^\top(\hat{\boldsymbol{\mu}}_i) = \frac{1}{p-1} \sum_{\substack{1 \le j \le p \\ j \neq i}} \mathbb{I}\left(d(\hat{\boldsymbol{\mu}}_i, \boldsymbol{\mu}_j) \le d(\hat{\boldsymbol{\mu}}_i, \boldsymbol{\mu}_i)\right) \tag{11}$$

where p is the number of perturbations and d is the Euclidean distance. This metric ranges from 0 (perfect) to 1 (worst), with 0.5 corresponding to random predictions. The transposed rank is particularly sensitive to mode collapse, as a model generating similar predictions for all perturbations will have many ground-truth observations closer than the matched one.

Magnitude Ratio Causal effects can occur on different scales across biological contexts, making absolute distributional distances difficult to interpret. In particular, a small distance does not imply a weak causal effect, nor does a large distance imply a strong one. To normalize for effect scale, we introduce the *magnitude ratio*, which measures how much of the true intervention effect is recovered by the prediction. Let d denote a distributional distance (e.g. Wasserstein distance). The *magnitude ratio* is defined as

$$\text{MagRatio}(\mathbf{Y}^{\text{obs}}, \mathbf{Y}^{\text{int}}, \hat{\mathbf{Y}}^{\text{int}}) = \frac{d(\mathbf{Y}^{\text{obs}}, \hat{\mathbf{Y}}^{\text{int}})}{d(\mathbf{Y}^{\text{obs}}, \mathbf{Y}^{\text{int}})} \tag{12}$$

A perfect prediction corresponds to a magnitude ratio of 1.0 and an identity collapse ($\hat{\mathbf{Y}}^{\text{int}} = \mathbf{Y}^{\text{obs}}$) results in a magnitude ratio of 0.0. The magnitude ratio is invariant to the absolute effect scale and quantifies effect size recovery but not directionality. We report it using the Wasserstein distance.

Table 5: Hyperparameter search ranges for each method.

Method	Hyperparameter	Search Range
MapPFN	Classifier-free guidance ω	$\{1.0, 1.5, 2.0, 2.5, 3.0\}$
CondOT	Hidden dimensions	$\{64, 128, 256\}$
	Hidden layers	$\{2, 3, 4\}$
MetaFM	k-nearest neighbors	$\{0, 10, 50, 100\}$
	GNN embedding dimensions	$\{64, 128, 256\}$

Area Under the Precision Recall Curve To evaluate whether model predictions reliably imply identification of differentially expressed genes (DEGs), we adopt the AUPRC metric from Zhu et al. (2025). For a given perturbation, ground-truth DEGs are identified using a per-gene Wilcoxon rank-sum test comparing single-cell expression values before and after intervention, under the null hypothesis of identical distributions (Wilcoxon, 1945). Benjamini-Hochberg (Benjamini & Hochberg, 2000) correction is applied across genes, and DEGs are defined by jointly thresholding on effect size and statistical certainty, using the absolute $\log_2$ fold-change ($\tau_l = 0.2$) and the negative $\log_{10}$ p-value ($\tau_p = 2$).

$$Z_g = \mathbb{I}\left(\tilde{p}_g > \tau_p \wedge |\tilde{l}_g| > \tau_l\right) \tag{13}$$

where $\tilde{p}_g = -\log_{10}(p_g)$ and $\tilde{l}_g = \log_2(\tilde{\mu}_g^{\text{int}}/\tilde{\mu}_g^{\text{obs}})$ denote the negative log p-value and log fold-change for gene g, respectively. For in silico predictions, we compute a ranking score $R_g = |\hat{l}_g| \cdot \mathbb{I}(\hat{p}_g > \tau_p)$ that combines the magnitude of predicted expression change with statistical significance. By varying a threshold r on this score, we generate a family of classifiers $\hat{Z}_g(r) = \mathbb{I}(R_g > r)$ and construct precision-recall curves against the ground-truth labels Z_g. The AUPRC summarizes model performance, with the baseline AUPRC given by $\pi = (\text{number of DEGs})/(\text{total genes})$, corresponding to random ranking. As an additional baseline for gene knockout perturbations, we consider a predictor that assigns a positive score only to the perturbed gene. Differential expression analysis was performed using `scanpy.tl.rank_genes_groups` (Wolf et al., 2018).

E Hyperparameters

By default, we use the hyperparameters recommended by the authors of each baseline. Additionally, we perform a grid search over a limited set of hyperparameters, summarized in Table 5. We select the hyperparameters with the lowest Wasserstein distance measured on the validation set.

F Implementation

We use `JAX` (Bradbury et al., 2018) to implement our experiments. Our model is implemented using `equinox` (Kidger & Garcia, 2021) and `diffrax` (Kidger, 2022) for ODE solving. We also make use of Optimal Transport Tools (`OTT`) (Cuturi et al., 2022) to compute the Sinkhorn distance. We use `hydra-zen` (Soklaski et al., 2022) to configure our experiments. For single-cell data processing, we build upon the `scverse` ecosystem, including `anndata` (Virshup et al., 2024), `scanpy` (Wolf et al., 2018) and `pertpy` (Heumos et al., 2025).

We run our experiments on a high-performance cluster, using a single NVIDIA A100 or H100 GPU with 80 GB of VRAM for training. For the linear SCM dataset, each experiment ran for 2-8h depending on the method and configuration. Pretraining MapPFN on synthetic single-cell data took approximately 10-36h, depending on the setting and corresponding context size.

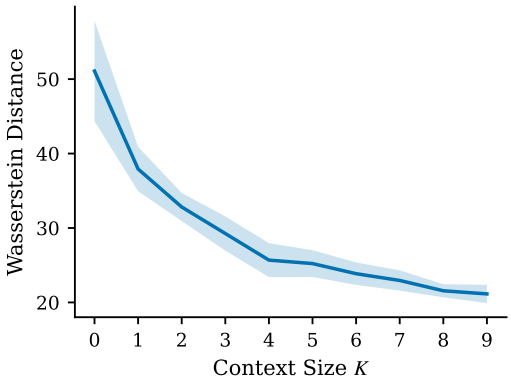

Figure 4: **Increasing the number of perturbation experiments in context improves performance.** Wasserstein distance measured on the test context of the single-cell dataset for varying numbers of perturbation experiments in the context set $\mathcal{C}$. Shaded regions indicate standard deviation over three model seeds. Increasing the context size $K = |\mathcal{C}|$ improves performance of MapPFN, with diminishing returns for more than four perturbation experiments.

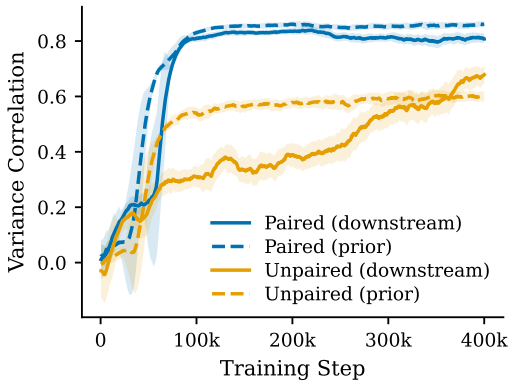

Figure 5: **Counterfactual paired prior improves downstream performance.** Counterfactual interventional distributions using shared noise across treatments accelerate convergence and improve final performance both within the prior (dashed) and on real single-cell data (solid). Variance correlation measures the Pearson correlation between feature variances of predicted and ground-truth samples. Shaded regions indicate rolling standard deviation; curves show EMA ($\alpha = 0.95$).

G Additional Experiments

G.1 Scaling Interventional Context

To evaluate how the performance of MapPFN scales with the amount of interventional experiments provided in context, we measure the Wasserstein distance for varying context sizes $K = |\mathcal{C}|$. As shown in Figure 4, test performance improves monotonically as additional perturbation experiments are provided in context, with diminishing returns beyond four interventional experiments. This indicates that conditioning on interventional data enables the model to learn perturbation-specific mappings that are not accessible to approaches conditioning only on the treatment identifier or the observational distribution.

G.2 Zero-shot Evaluation

In Table 6, we compare MapPFN against baselines in the zero-shot setting within our prior of linear SCMs. For the real-world single-cell data, we show this comparison in Table 7. As we have shown in Table 2, MapPFN achieves competitive performance in the few-shot setting, underscoring that access to interventional distributions helps MapPFN generalize to unseen biological contexts.

Table 6: **Evaluation within a prior of linear SCMs in the zero-shot setting.** Test metrics aggregated over random seeds (mean $\pm$ std) for the SCM dataset in the zero-shot setting. Bold indicates results within one standard deviation of the best.

Method	W_2 (↓)	MMD (↓)	RMSE (↓)	Rank$^\top$ (↓)	MagRatio
CondOT	14.08 ± 0.43	5.17 ± 0.28	0.15 ± 0.00	0.10 ± 0.04	0.10 ± 0.00
MetaFM	**13.72** ± 0.12	4.80 ± 0.11	0.15 ± 0.01	0.09 ± 0.02	0.12 ± 0.00
MapPFN (ours)	**13.67** ± 0.22	**3.99** ± 0.41	**0.13** ± 0.01	**0.00** ± 0.00	**0.98** ± 0.01
Identity	17.61 ± 0.14	12.98 ± 0.35	0.28 ± 0.01	0.49 ± 0.01	0.00 ± 0.00
Observed	9.82 ± 0.08	3.66 ± 0.06	0.07 ± 0.00	0.00 ± 0.00	1.00 ± 0.00

Table 7: **Comparison of MapPFN trained on synthetic prior to baselines trained on real single-cell data in the zero-shot setting.** Test metrics aggregated over random seeds (mean ± std) for the single-cell dataset in the zero-shot setting. While the baselines were trained on the real dataset, MapPFN was pretrained exclusively on synthetic data from our prior. Bold indicates results within one standard deviation of the best.

Method	W_2 (↓)	MMD (↓)	RMSE (↓)	Rank$^\top$ (↓)	MagRatio (↑)
CondOT	22.21 ± 0.38	$\mathbf{7.21} \pm 0.45$	$\mathbf{0.10} \pm 0.01$	$\mathbf{0.07} \pm 0.02$	0.05 ± 0.01
MetaFM	$\mathbf{20.78} \pm 0.50$	9.86 ± 1.00	0.13 ± 0.02	0.15 ± 0.01	0.23 ± 0.02
MapPFN (ours)	23.40 ± 0.23	16.72 ± 0.80	0.20 ± 0.01	0.26 ± 0.02	$\mathbf{1.08} \pm 0.01$
Identity	22.57 ± 0.08	6.63 ± 0.10	0.11 ± 0.00	0.51 ± 0.02	0.00 ± 0.00
Observed	7.30 ± 0.08	1.40 ± 0.02	0.03 ± 0.00	0.00 ± 0.00	1.00 ± 0.00

G.3 Paired Interventional Distributions

Single-cell perturbation prediction is typically framed as a mapping between unpaired distributions, since individual cells are destroyed during measurement. To isolate the task of causal inference from the additional difficulty introduced by unpaired data, we follow Robertson et al. (2025) in pretraining MapPFN on counterfactual interventional data. In our biological prior, pairing is achieved by fixing the random seed of SERGIO across treatments, ensuring that the differences between interventional distributions are not driven by a difference in initial condition to the stochastic differential equation, but only by the differences in underlying mechanism and perturbation effects.

Figure 5 shows the Pearson correlation between the feature-wise variances of the predicted and ground-truth post-perturbation distribution on the validation set, evaluated separately for the paired and unpaired datasets. The paired prior converges to a correlation of approximately 0.8 within 50k training steps. In contrast, the unpaired prior saturates around 0.6 even after 400k steps. Downstream performance increases only gradually with additional pretraining, indicating that longer training is required for the unpaired prior to approach the performance of the paired setting. After pretraining for 400k steps, this results in a dramatic performance drop across metrics compared to the paired prior, as shown in Table 2. These findings show that a prior of counterfactual pairs can improve downstream performance in real single-cell data. We suspect that counterfactual interventional distributions provide stronger signal by isolating causal effects from the added variability of unpaired samples.

Having shown that pretraining MapPFN on counterfactual interventional distributions improves downstream performance (Table 2), we further investigate this effect in our synthetic prior of linear SCMs. MapPFN benefits from paired interventional data in the few-shot setting (Table 8) in terms of Wasserstein distance and MMD, but not in the zero-shot setting (Table 9). This observation is consistent with our findings in real single-cell data.

Table 8: **Evaluation of *paired* interventional distributions within a prior of linear SCMs in the few-shot setting.** Test metrics aggregated over three random seeds (mean ± std) for the SCM dataset in the few-shot setting. To generate counterfactual interventional distributions, we use the same noise across treatments. Bold indicates results within one standard deviation of the best.

Method	W_2 (↓)	MMD (↓)	RMSE (↓)	Rank$^\top$ (↓)	MagRatio
CondOT	14.01 ± 0.41	4.61 ± 0.31	0.14 ± 0.01	0.06 ± 0.01	0.10 ± 0.01
MetaFM	13.87 ± 0.28	4.32 ± 0.02	$\mathbf{0.13} \pm 0.01$	0.07 ± 0.02	0.12 ± 0.00
MapPFN (ours)	$\mathbf{4.22} \pm 0.47$	$\mathbf{3.14} \pm 0.28$	$\mathbf{0.12} \pm 0.01$	$\mathbf{0.01} \pm 0.02$	$\mathbf{1.02} \pm 0.01$
Identity	17.35 ± 0.11	11.86 ± 0.22	0.26 ± 0.00	0.48 ± 0.02	0.00 ± 0.00
Observed	9.89 ± 0.06	3.64 ± 0.05	0.07 ± 0.00	0.00 ± 0.00	1.00 ± 0.00

Table 9: **Evaluation of *paired* interventional distributions within a prior of linear SCMs in the zero-shot setting.** Test metrics aggregated over random seeds (mean $\pm$ std) for the SCM dataset in the zero-shot setting. To generate counterfactual interventional distributions, we use the same noise across treatments. Bold indicates results within one standard deviation of the best.

Method	W_2 (↓)	MMD (↓)	RMSE (↓)	Rank$^\top$ (↓)	MagRatio
CondOT	13.94 ± 0.33	4.45 ± 0.03	0.13 ± 0.00	0.06 ± 0.01	0.09 ± 0.01
MetaFM	**13.67** ± 0.12	4.29 ± 0.12	0.13 ± 0.01	0.09 ± 0.01	0.12 ± 0.00
MapPFN (ours)	**13.66** ± 0.24	**3.81** ± 0.25	**0.12** ± 0.00	**0.00** ± 0.00	**0.98** ± 0.01
Identity	17.35 ± 0.11	11.86 ± 0.22	0.26 ± 0.00	0.48 ± 0.02	0.00 ± 0.00
Observed	9.89 ± 0.06	3.64 ± 0.05	0.07 ± 0.00	0.00 ± 0.00	1.00 ± 0.00