PERTURBCELLRL: ALIGNING DISTRIBUTIONS AND GROUNDING BIOLOGY VIA POST-TRAINING PERTURBATION GENERATORS

Anonymous authors
Paper under double-blind review

ABSTRACT

Single-cell perturbation models can reduce costly wet-lab screening by predicting how cells respond transcriptionally to interventions. Recent advances in flow-matching have enabled population-level prediction of cellular responses. However, flow-matching training can fail to recover certain target distributions even within the model family, limiting its ability to capture cellular heterogeneity. We first prove that post-training can recover these distributions, then introduce PerturbCellRL, a reinforcement learning framework that post-trains single-cell perturbation generators using per-cell rewards. The central component is a gene-expression energy witness that translates population-level discrepancies into per-cell feedback. We further prove that this reward's policy gradient points toward better distributional alignment. Two complementary rewards, calibrated on real cells, penalize atypical expression profiles and insufficient pathway-level responses to perturbations. Across genetic and chemical perturbation benchmarks, PerturbCellRL substantially improves distributional alignment and recovers pathway enrichment patterns more faithfully. These results establish reward-guided post-training as an effective strategy for improving both distributional accuracy and biological fidelity in perturbation prediction.

1 INTRODUCTION

Predicting how single-cell gene expression responds to genetic or chemical perturbations is a core capability of the envisioned virtual cell (Bunne et al., 2024; Roohani et al., 2025). In single-cell perturbation screens, a central task is to assess how an intervention changes a cell population relative to a control. These changes can involve both shifts in **average expression** and **cell-to-cell variation**. For example, the same population-average change could arise from a weak response across many cells or a strong response confined to a small subset. A predictive model should therefore capture not only average expression but also the heterogeneity of cellular responses. Since gene-expression measurements are destructive, control and perturbed cells are observed as unpaired populations, making the response distribution a natural prediction target. Recent flow-matching models (Lipman et al., 2023; Liu et al., 2023) provide a well-suited framework for learning these perturbation-specific distributions of single-cell gene expression (Yu et al., 2026; Klein et al., 2025).

However, via a counterexample, we prove that minimizing the flow-matching loss can fail to recover a target distribution that lies within the model family, while post-training recovers the target within the same family (Figure 1). This gap motivates post-training with direct distributional feedback on generated cell populations.

Reinforcement learning provides a practical framework: sample cellular responses, evaluate them with reward functions, and update the generator to favor higher-reward outputs (Black et al., 2023; Fan et al., 2023; Liu et al., 2025; Xue et al., 2025; Li et al., 2025; Zheng et al., 2025; Wu et al., 2026). We seek to assign a reward to each generated cell, yet distributional agreement depends on the population as a whole. The question is how to translate population discrepancies into per-cell rewards that improve distributional alignment while preserving cellular heterogeneity and biological fidelity.

To address this, we introduce **PerturbCellRL**, a reinforcement learning framework for post-training single-cell perturbation generators with per-cell rewards. Its central component is an energy witness

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.

in gene-expression space that scores each generated cell against generated and real target populations. This reward encourages proximity to real responses while discouraging crowding among generated cells. We prove that optimizing this per-cell reward can improve alignment between generated and target distributions under suitable assumptions. This connection provides a theoretical basis for using individual-cell feedback to improve population-level predictions.

To explicitly assess biological fidelity during post-training, we complement distributional feedback with assessments of expression profiles and perturbation-specific pathway responses. Expression support penalizes unusually extreme expression profiles, while pathway discrimination penalizes weak evidence for the requested perturbation relative to the control. We use DiffusionNFT (Zheng et al., 2025) to integrate these rewards and post-train the perturbation generator.

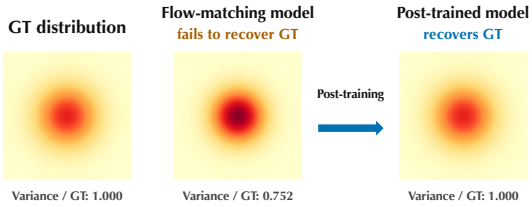


Figure 1: **Post-training resolves a flow-matching mismatch.** Left: the ground-truth (GT) density in a two-dimensional Gaussian example. Middle: the flow-matching optimum has the correct mean but only 75.2% of the GT variance. Right: post-training recovers the GT distribution within the same model family (Proposition 1).

We evaluate PerturbCellRL on the Norman and ComboSciPlex benchmarks for genetic and chemical perturbations. Across four evaluation settings, PerturbCellRL achieves the best distributional alignments among all compared methods, and also recovers pathway enrichment patterns more accurately. Ablations examine how the three rewards jointly shape distributional alignment, expression support, and pathway responses.

In summary, our theoretical analyses identify opportunities to improve distributional alignment beyond flow-matching training alone. Building on these insights, PerturbCellRL uses direct feedback on generated responses to guide post-training toward better distributional alignment and single-cell biological fidelity. These improvements may support downstream applications in drug discovery and personalized medicine by reducing reliance on costly wet-lab experiments.

2 Related Work

Single-cell perturbation prediction with large-scale generative modeling. Single-cell perturbation prediction aims to infer how cellular transcriptomes change under genetic or chemical interventions. This problem has received substantial attention, driven by the increasing availability of large-scale perturbation datasets and advances in generative modeling.

On the data side, datasets such as Norman (Norman et al., 2019), ComboSciPlex (Mathur et al., 2022) and Virtual Cell Challenge (Roohani et al., 2025) provide challenging genetic and chemical perturbation benchmarks. scPerturb (Peidli et al., 2024) highlights the broader availability of harmonized single-cell perturbation data.

On the modeling side, early deep generative approaches, notably scVI (Lopez et al., 2018), established probabilistic representation learning for single-cell transcriptomics. Subsequent perturbation prediction methods have used conditional autoencoders, graph neural networks, transformer architectures, and generative models to predict responses under unseen perturbations or cellular contexts (Lotfollahi et al., 2023; Roohani et al., 2024; Adduri et al., 2025; Bereket & Karaletsos, 2023; Rampášek et al., 2019). More recently, state-of-the-art methods have begun to adopt flow matching (Lipman et al., 2023; 2024; Liu et al., 2023), which provides a natural framework for conditional generation. This is particularly well suited to perturbation prediction, where models generate perturbed cells conditioned on control states and perturbation labels (Yu et al., 2026; Klein et al., 2025; Zhang et al., 2025). In this work, we build on this line of research, using scDFM (Yu et al., 2026) as our base generator and further improving it through post-training.

Reinforcement learning for biological alignment. Although generative models can produce expression-like cell profiles, incorporating biological priors into their predictions remains challenging. This issue is especially important in single-cell perturbation prediction, where a generated cell may

appear realistic at the expression level while its perturbation effect is biologically inconsistent. A trustworthy prediction should match the observed population-level expression distribution, keep individual cells within the range of real expression profiles, and carry the pathway-level response to the applied perturbation. However, these objectives are often non-differentiable, making them difficult to optimize directly.

One promising solution is reinforcement learning (RL), which has recently been used to align diffusion and flow models with human preferences or task-specific rewards (Black et al., 2023; Fan et al., 2023; Liu et al., 2025; Xue et al., 2025; Li et al., 2025; Wu et al., 2026). Flow and diffusion models introduce additional challenges because exact sample likelihoods are often intractable. Prior work such as FlowGRPO and MixGRPO formulate this problem as a Markov decision process, while recent methods such as DiffusionNFT (Zheng et al., 2025) introduce forward-process objectives for online RL, improving both training efficiency and stability. In this work, we adapt this alignment perspective to transcriptomic perturbation prediction, where rewards are derived from distributional feedback and calibrated single-cell biological criteria rather than visual quality or human-preference scores.

3 METHOD

3.1 PROBLEM SETUP AND MOTIVATION

Problem Setup. Let $\mathcal{X} \subseteq \mathbb{R}^G$ be the expression space over G genes. For perturbation $c \in \mathcal{C}$, let P_0 and P_c denote the control and target perturbed populations. We use $\boldsymbol{u}$ and $\boldsymbol{x}_1$ for control and real perturbed profiles, respectively. Training triples $(\boldsymbol{u}, c, \boldsymbol{x}_1)$ are drawn from the observed data; noise $\boldsymbol{x}_0 \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ and time $t \sim \mathcal{U}[0,1]$ are sampled independently of each other and these triples.

Flow-based generators such as scDFM (Yu et al., 2026) learn a conditional velocity field $\boldsymbol{v}_\theta$ by minimizing

$$\mathcal{L}_{\mathrm{FM}}(\theta) = \mathbb{E}_{\boldsymbol{u},c,\boldsymbol{x}_1,\boldsymbol{x}_0,t}\left[\|\boldsymbol{v}_\theta(\boldsymbol{x}_t, t \mid \boldsymbol{u}, c) - (\boldsymbol{x}_1 - \boldsymbol{x}_0)\|_2^2\right], \qquad \boldsymbol{x}_t = (1-t)\boldsymbol{x}_0 + t\boldsymbol{x}_1. \quad (1)$$

The flow $\dot{Z}_t = \boldsymbol{v}_\theta(Z_t, t \mid \boldsymbol{u}, c)$, initialized at $Z_0 = \boldsymbol{x}_0$, generates $\hat{\boldsymbol{x}}_1 = Z_1$ with conditional distribution $\pi_\theta(\cdot \mid \boldsymbol{u}, c)$. Marginalizing over control profiles gives the generated population

$$Q_{\theta,c}(\mathrm{d}\hat{\boldsymbol{x}}_1) = \int \pi_\theta(\mathrm{d}\hat{\boldsymbol{x}}_1 \mid \boldsymbol{u}, c)\, P_0(\mathrm{d}\boldsymbol{u}). \quad (2)$$

Motivation. Flow matching provides a strong starting point. However, we show that flow matching can fail to recover a target distribution even when that distribution lies within the model family (Proposition 1). Appropriate post-training can correct this mismatch.

Proposition 1 (Flow-matching mismatch and recovery by post-training)**.** *There exists a model family whose parameterization includes the target distribution, yet the minimizer of its flow-matching loss generates a different distribution. Starting from this minimizer, witness-based post-training within the same family converges to the target distribution for a sufficiently small constant step size.*

Intuitively, post-training pulls the generated distribution toward the target by rewarding samples that improve population-level agreement. Appendix B gives the construction and proof.

This mismatch motivates post-training with direct feedback on generated cells. Starting from pretrained scDFM parameters θ_0, we aim to improve both population-level alignment and single-cell biological fidelity while staying close to the pretrained generator. We partition perturbations into disjoint sets $\mathcal{C}_{\mathrm{train}}$ and $\mathcal{C}_{\mathrm{test}}$, using only training conditions for post-training and evaluating on held-out test conditions. We optimize the following objective over training perturbations:

$$\begin{aligned} \min_{\theta} \quad & \mathbb{E}_{c\sim\mathrm{Unif}(\mathcal{C}_{\mathrm{train}})}\left[D(Q_{\theta,c}, P_c) - \lambda_{\mathrm{bio}}\, \mathbb{E}_{\hat{\boldsymbol{x}}_1\sim Q_{\theta,c}}\left[r_{\mathrm{bio}}(\hat{\boldsymbol{x}}_1, c)\right]\right] \\ \text{subject to} \quad & \mathbb{E}_{\substack{c\sim\mathrm{Unif}(\mathcal{C}_{\mathrm{train}}) \\ \boldsymbol{u}\sim P_0}}\left[D_{\mathrm{KL}}(\pi_\theta(\cdot \mid \boldsymbol{u}, c) \,\|\, \pi_{\theta_0}(\cdot \mid \boldsymbol{u}, c))\right] \le \epsilon. \end{aligned} \quad (3)$$

Here, D measures population discrepancy (e.g., energy distance), r_{bio} rewards single-cell biological fidelity, $\lambda_{\mathrm{bio}} \ge 0$ balances the two objectives, and ϵ bounds deviation from the pretrained generator.

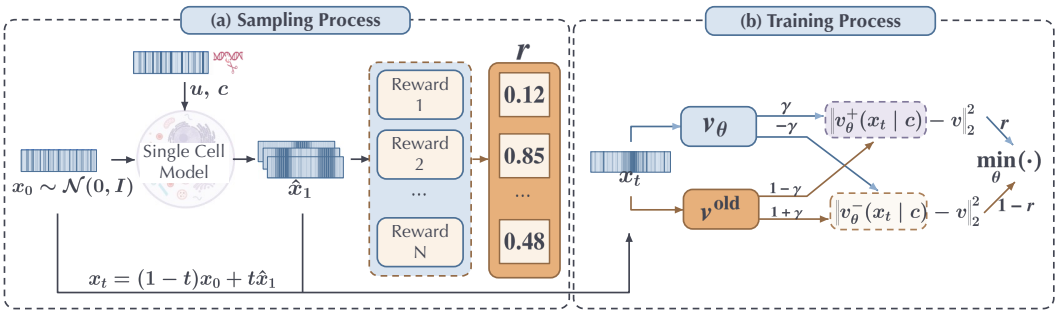


Figure 2: ***PerturbCellRL* algorithm.** RL post-training seeks to increase the likelihood of high-reward samples and decrease the likelihood of low-reward samples. The core training loop consists of interleaved sampling and training phases. (a) Sampling: we generate multiple rollouts from a fixed control expression and perturbation condition, scoring each with the reward suite. (b) Training: because exact likelihoods under flow matching are intractable, we construct positive and negative velocities from the rollout batch and optimize them contrastively, following DiffusionNFT (Zheng et al., 2025). The branch scale γ and preference weight r control the contrastive update.

3.2 RL Post-Training

Algorithm overview. We adopt DiffusionNFT (Zheng et al., 2025) for reward-guided velocity fitting. Each RL iteration alternates between sampling and optimization: we generate multiple cell profiles per control–perturbation pair, score them using the reward suite in Section 3.3, and update the model based on relative reward rankings within each group (Figure 2).

Algorithm detail. At the start of each iteration, let $\boldsymbol{v}^{\text{old}}$ be a frozen copy of the current velocity field. We use it to generate samples; it remains fixed throughout optimization. For a generated profile $\hat{\boldsymbol{x}}_1$ conditioned on $(\boldsymbol{u}, c)$, define the training state and target velocity as

$$\boldsymbol{x}_t = (1-t)\boldsymbol{x}_0 + t\hat{\boldsymbol{x}}_1, \qquad \boldsymbol{v} = \hat{\boldsymbol{x}}_1 - \boldsymbol{x}_0, \tag{4}$$

where $\boldsymbol{x}_0 \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ and $t \sim \mathcal{U}[0,1]$ are resampled independently at each training step. All velocity fields below are evaluated at $(\boldsymbol{x}_t, t \mid \boldsymbol{u}, c)$, with arguments omitted for brevity. NFT defines positive and negative velocity branches that interpolate toward and away from $\boldsymbol{v}_\theta$ relative to $\boldsymbol{v}^{\text{old}}$:

$$\boldsymbol{v}_\theta^+ = \gamma \boldsymbol{v}_\theta + (1-\gamma)\boldsymbol{v}^{\text{old}}, \qquad \boldsymbol{v}_\theta^- = -\gamma \boldsymbol{v}_\theta + (1+\gamma)\boldsymbol{v}^{\text{old}}, \tag{5}$$

where $\gamma > 0$ controls branch scale. We standardize each cell's reward R across rollouts sharing the same control and perturbation pair to obtain advantage A. Section 3.3.3 specifies reward construction. Clipping and rescaling A to $[-1,1]$ gives $\widetilde{A}$. We define the preference weight and optimize

$$r = \tfrac{1+\widetilde{A}}{2} \in [0,1], \quad \mathcal{L}(\theta) = \mathbb{E}\Big[r\|\boldsymbol{v}_\theta^+ - \boldsymbol{v}\|_2^2 + (1-r)\|\boldsymbol{v}_\theta^- - \boldsymbol{v}\|_2^2 + \lambda_{\text{reg}}\|\boldsymbol{v}_\theta - \boldsymbol{v}_{\theta_0}\|_2^2\Big]. \tag{6}$$

The preference weights r and $1-r$ determine the relative strength of the positive and negative velocity targets. The last term regularizes the model toward the fixed pretrained generator $\boldsymbol{v}_{\theta_0}$. After optimization, we update the frozen sampling policy $\boldsymbol{v}^{\text{old}} \leftarrow \boldsymbol{v}_\theta$ and begin the next iteration. Appendix C provides the full procedure.

3.3 Reward Suite

Under ideal on-policy updates, per-cell witness rewards provide a descent direction for population discrepancy, as shown in Theorem 1. We instantiate this principle in gene-expression space, then combine the population reward with single-cell expression support and perturbation-specific pathway discrimination rewards, as shown in Figure 3.

3.3.1 Population Alignment via an Energy Witness

We first establish a theoretical link between per-cell rewards and population-level distributional alignment. Fix a perturbation c, with generated population $Q = Q_{\theta,c}$ and ground-truth population

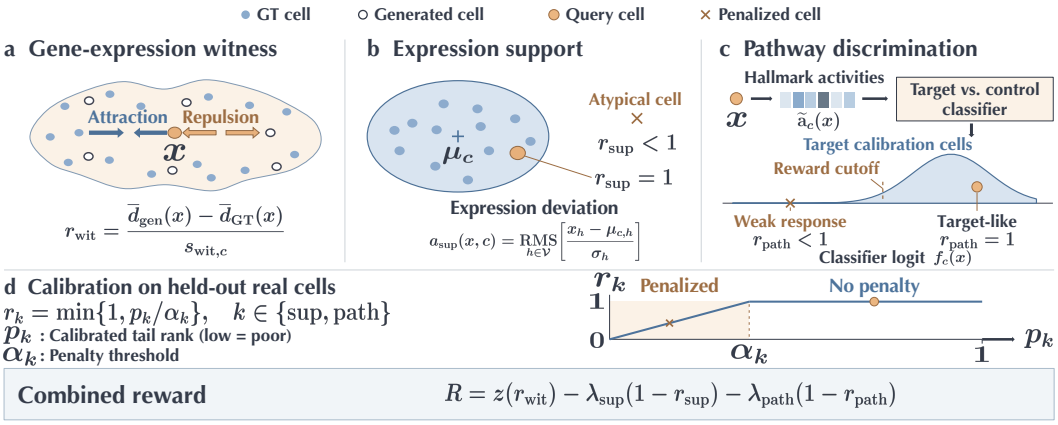


Figure 3: **Complementary population and single-cell rewards.** (a) The gene-expression witness rewards closeness to real cells while penalizing crowding among generated cells. (b) Expression support penalizes unusually large standardized radial deviations. (c) Pathway discrimination penalizes weak target-versus-control evidence from Hallmark activities. Bottom panels show calibration and reward combination: tail ranks p_k and cutoffs α_k define $r_k = \min\{1, p_k/\alpha_k\}$ for $k \in \{\text{sup}, \text{path}\}$.

$P = P_c$. For a fixed feature map $\phi : \mathcal{X} \to \mathbb{R}^d$ (we present the theory for general ϕ and instantiate it with the identity map $\phi(\boldsymbol{x}) = \boldsymbol{x}$ on gene expression below), let $d_\phi(\hat{\boldsymbol{x}}_1, \boldsymbol{x}_1) = \|\phi(\hat{\boldsymbol{x}}_1) - \phi(\boldsymbol{x}_1)\|_2$. The energy distance in this space is

$$D_\phi(Q, P) = 2\mathbb{E}_{Q\otimes P} d_\phi - \mathbb{E}_{Q\otimes Q} d_\phi - \mathbb{E}_{P\otimes P} d_\phi. \tag{7}$$

We assign each cell the corresponding energy witness:

$$w^\phi_{Q,P}(\hat{\boldsymbol{x}}_1) = \mathbb{E}_{\hat{\boldsymbol{x}}'_1 \sim Q} d_\phi(\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}'_1) - \mathbb{E}_{\boldsymbol{x}_1 \sim P} d_\phi(\hat{\boldsymbol{x}}_1, \boldsymbol{x}_1). \tag{8}$$

Intuitively, this reward encourages generated cells to be close to real cells while discouraging them from clustering together.

Theorem 1 (Witness policy-gradient identity)**.** *Suppose P_c and $Q_{\theta,c}$ have finite first feature moments. Assume conditional densities are differentiable with parameter-independent support, and differentiation commutes with the relevant expectations. Treat the current witness $w^\phi_{Q_{\theta,c},P_c}$ as a fixed reward. Its on-policy gradient satisfies*

$$\begin{aligned} g_\phi(\theta) &:= \mathbb{E}_{\substack{\boldsymbol{u}\sim P_0 \\ \hat{\boldsymbol{x}}_1 \sim \pi_\theta(\cdot|\boldsymbol{u},c)}} \left[w^\phi_{Q_{\theta,c},P_c}(\hat{\boldsymbol{x}}_1) \nabla_\theta \log \pi_\theta(\hat{\boldsymbol{x}}_1 \mid \boldsymbol{u}, c) \right] \\ &= -\frac{1}{2} \nabla_\theta D_\phi(Q_{\theta,c}, P_c). \end{aligned} \tag{9}$$

Whenever $g_\phi(\theta) \neq 0$, it is a descent direction for D_ϕ.

The witness thus provides per-cell credit that yields a policy gradient reducing population discrepancy. When discrepancy reaches zero, the generated and target distributions agree in the chosen feature space. Appendix A proves the identity and this characterization.

Energy witness in gene-expression space. The population reward r_{wit} measures distributional agreement in gene-expression space. We use the identity map $\phi_{\text{wit}}(\boldsymbol{x}) = \boldsymbol{x}$ on the current gene panel. Using real responses $\{\boldsymbol{x}_{1,j}\}$ from P_c, we define

$$r_{\text{wit},i} = \frac{\widehat{w}_i^{\phi_{\text{wit}}}}{s_{\text{wit},c}}, \qquad s_{\text{wit},c} = \left(\text{median}_{j<\ell} \|\boldsymbol{x}_{1,j} - \boldsymbol{x}_{1,\ell}\|_2^2\right)^{1/2} > 0, \tag{10}$$

where $\widehat{w}_i^{\phi_{\text{wit}}}$ estimates the energy witness $w^{\phi_{\text{wit}}}_{Q_{\theta,c},P_c}(\hat{\boldsymbol{x}}_{1,i})$ in equation 8 from generated and real cells; Appendix C details the sampling and estimator. Because $s_{\text{wit},c}$ is a positive constant derived from the target population, this scaling preserves the gradient direction established in Theorem 1.

3.3.2 SINGLE-CELL FIDELITY VIA CALIBRATION

The population witness is complemented by two single-cell quality criteria: expression support and pathway response to the applied perturbation. Both use held-out real cells to identify unusually poor scores and convert them into bounded rewards that penalize only these cells.

Expression support. To construct the expression support reward r_{sup}, we measure how far a cell deviates from the typical expression profile of its condition. Let $\mu_{c,h}$ be the reference mean for gene h under condition c, and σ_h its standard deviation across reference cells. For variable genes $\mathcal{V} = \{h : \sigma_h > 0\}$, define the radial nonconformity score

$$a_{\text{sup}}(\boldsymbol{x}, c) = \left[\frac{1}{|\mathcal{V}|} \sum_{h \in \mathcal{V}} \left(\frac{x_h - \mu_{c,h}}{\sigma_h} \right)^2 \right]^{1/2}. \tag{11}$$

Higher values indicate more extreme expression profiles. Let $p_{\text{sup}}(\boldsymbol{x}, c)$ be the upper-tail rank of this score among real calibration scores pooled across conditions. A small p_{sup} indicates an unusually large radial deviation. The expression support reward is

$$r_{\text{sup}}(\boldsymbol{x}, c) = \min \left\{ 1, \frac{p_{\text{sup}}(\boldsymbol{x}, c)}{\alpha_{\text{sup}}} \right\}, \qquad \alpha_{\text{sup}} \in (0, 1). \tag{12}$$

Pathway discrimination. Expression support alone does not assess whether a cell carries the response associated with the requested perturbation. The pathway discrimination reward r_{path} assesses evidence for this response relative to control cells. We represent these responses using Hallmark gene sets (Liberzon et al., 2015). Let $\mathcal{G}_p$ contain pathway p's genes present in the current gene panel. For pathways with $|\mathcal{G}_p| \geq 5$, we define the standardized activity vector directly:

$$\widetilde{\boldsymbol{a}}_c(\boldsymbol{x}) = \left(\frac{|\mathcal{G}_p|^{-1} \sum_{h \in \mathcal{G}_p} x_h - m_{c,p}}{s_{c,p}} \right)_{p : s_{c,p} > 0}, \tag{13}$$

where $m_{c,p}$ and $s_{c,p}$ are pathway reference means and standard deviations.

Following the treatment-versus-control classification approach of Augur (Skinnider et al., 2021), we fit a separate ℓ_1-regularized logistic classifier for each condition. The classifier logit is $f_c(\boldsymbol{x}) = \boldsymbol{\omega}_c^\top \widetilde{\boldsymbol{a}}_c(\boldsymbol{x}) + b_c$. Higher logits indicate stronger target-versus-control evidence. Let $p_{\text{path}}(\boldsymbol{x}, c)$ be the lower-tail rank of this logit among held-out real target cells under condition c. A small p_{path} indicates unusually weak response evidence. The pathway discrimination reward is

$$r_{\text{path}}(\boldsymbol{x}, c) = \min \left\{ 1, \frac{p_{\text{path}}(\boldsymbol{x}, c)}{\alpha_{\text{path}}} \right\}, \qquad \alpha_{\text{path}} \in (0, 1). \tag{14}$$

Both rewards penalize cells whose rank falls below the threshold; above it, the reward saturates at 1, avoiding incentives to minimize deviation or maximize evidence beyond what real cells exhibit. This leaves population matching to the gene-expression witness. Appendix D provides details.

3.3.3 REWARD COMBINATION

The population reward r_{wit} is combined with single-cell rewards $r_{\text{sup}}, r_{\text{path}} \in [0, 1]$. Let z denote standardization across rollouts sharing the same control and perturbation pair. Positive weights $\lambda_{\text{sup}}, \lambda_{\text{path}}$ scale the two single-cell penalty terms. The standardized gene-expression witness has unit weight. For cell i, we define the combined reward

$$R_i = z(r_{\text{wit}})_i - \lambda_{\text{sup}}(1 - r_{\text{sup},i}) - \lambda_{\text{path}}(1 - r_{\text{path},i}). \tag{15}$$

The gene-expression witness is standardized before being combined, while the single-cell rewards retain their calibrated scales. We then standardize R within each group to obtain the advantage $A_i = z(R)_i$, which equation 6 maps to the preference weight r_i after clipping and rescaling. The single-cell penalties thus modulate the relative ranking of rollouts within each control–perturbation group, penalizing individually implausible cells even when the population-level witness is favorable.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Datasets. We evaluate on the Norman (Norman et al., 2019) and ComboSciPlex (Mathur et al., 2022) datasets, which cover genetic and chemical perturbations, respectively. Following scDFM (Yu et al., 2026), we evaluate two Norman train-test split protocols, illustrated in Figure 4. In the additive split, all single-gene perturbations and a subset of double-gene perturbations are used for training, and the model predicts held-out double-gene perturbations. In the holdout split, selected double-gene perturbations and their constituent single-gene perturbations are held out for testing, while the remaining perturbations are used for training. We report holdout single and holdout double by averaging over the single-gene and double-gene perturbations in the holdout test set, respectively. For both Norman split protocols, we use four random train–test folds and report fold-averaged results.

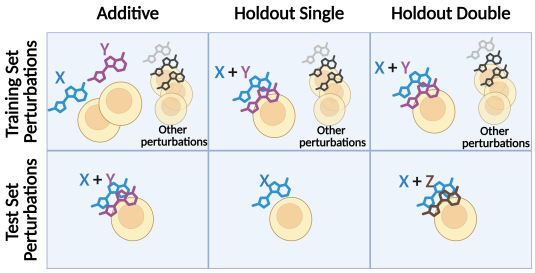


Figure 4: Additive and holdout split protocols. Top rows show training perturbations; bottom rows show held-out test perturbations.

Baselines. We compare against Control (unperturbed cells), Additive (a task-specific baseline for Norman additive that linearly superposes single-gene effects), GEARS (Roohani et al., 2024) (Norman only, as it models genetic perturbations), CPA (Lotfollahi et al., 2023), STATE (Adduri et al., 2025), CellFlow (Klein et al., 2025), and scDFM (Yu et al., 2026). For each test combination, Additive independently samples 128 training cells without replacement from each single perturbation and the control group. It sums the two sampled single-perturbation means, subtracts the sampled control mean, clips negative values to zero, and replicates the resulting vector 128 times for distributional evaluation. Our primary comparison is with scDFM, which serves as both the base model and the strongest flow-matching baseline.

Evaluation metrics. We evaluate distributions, pathway enrichment, and pseudobulk means. Distribution-level metrics are Energy distance, maximum mean discrepancy (MMD), and sliced Wasserstein-1 distance (SW1) (Bonneel et al., 2015). For pathway enrichment, we report Reactome GSEA-NES Spearman (NES-Sp.), following the evaluation approach of PertAdapt (Bai et al., 2026). Pseudobulk-mean metrics are mean absolute error (MAE), Pearson $\rho(\Delta)$ and $\rho(\hat{\Delta})$, DE-Spearman, and discrimination score (DS). We average metrics across test conditions within each setting and, for Norman, across the four folds. Appendix E provides definitions and computational details.

4.2 Main Results

PerturbCellRL substantially improves distributional alignment. As shown in Table 1, PerturbCellRL achieves the lowest Energy distance, MMD, and SW1 among all compared methods in all four settings. Relative to its pretrained counterpart scDFM, PerturbCellRL reduces MMD by 14.7%–20.8% and Energy distance by 6.8%–13.1%. The SW1 distance decreases by 6.4%–10.2% across the four settings. On ComboSciPlex, MMD decreases from 0.0462 to 0.0366, while Energy distance decreases from 0.5177 to 0.4498. These gains are consistent across both Norman split protocols, including held-out single- and double-gene perturbations. These improvements show that reward-guided post-training shifts generated cell populations closer to the target distributions.

PerturbCellRL improves pathway enrichment concordance. PerturbCellRL achieves the highest mean Reactome NES-Sp. among all methods in all four settings. These correlations indicate that PerturbCellRL better recovers pathway enrichment rankings, suggesting improved biological fidelity.

PerturbCellRL remains competitive in pseudobulk-mean prediction. PerturbCellRL improves MAE and both Pearson correlations over scDFM in all four settings, achieving the lowest MAE and highest $\rho(\hat{\Delta})$ throughout. For $\rho(\Delta)$, it ranks first on ComboSciPlex and both holdout settings, and second on Norman additive. It performs comparably to scDFM in DE-Spearman and DS. The results indicate that post-training improves prediction of mean perturbation-induced expression changes.

Table 1: **Population-level performance across Norman and ComboSciPlex settings.** Energy, MMD, and sliced Wasserstein-1 (SW1) compare cell distributions; NES-Sp. measures the concordance of Reactome GSEA normalized enrichment scores; the remaining metrics compare pseudobulk means. Bold indicates best; underline indicates second best. Norman results are averaged over four folds; ComboSciPlex uses the default scDFM split. CSP denotes ComboSciPlex; Add., H-S, and H-D denote Norman additive, holdout single, and holdout double, respectively.

		Distribution			Pathway	Pseudobulk-mean				
Set.	**Model**	**Energy** ↓	**MMD** ↓	**SW1** ↓	**NES-Sp.** ↑	**MAE** ↓	**DE-Sp.** ↑	$\rho(\hat{\Delta})$ ↑	$\rho(\Delta)$ ↑	**DS** ↑
CSP	Control	7.8378	0.5303	0.2100	-	0.0687	-	-0.3725	-	0.5714
	CPA	6.1738	0.4844	0.1863	0.3359	0.0690	0.2490	-0.3994	0.2501	0.5714
	STATE	0.7204	0.0601	0.0661	0.7805	0.0330	0.7317	0.7904	0.7915	0.8571
	CellFlow	2.2834	0.2113	0.1154	0.8868	0.0470	**0.8655**	0.2085	0.8254	0.5714
	scDFM	0.5177	0.0462	0.0654	0.8919	0.0242	0.8363	0.8418	0.8679	0.8776
	PerturbCellRL	**0.4498**	**0.0366**	**0.0587**	**0.8986**	**0.0231**	0.8416	**0.8536**	**0.8770**	**0.8980**
Add.	Control	5.9999	0.3236	0.1988	-	0.0384	-	-0.1285	-	0.5135
	Additive	4.6519	0.2188	0.1818	0.7860	0.0243	0.4971	0.8377	**0.8907**	0.9646
	GEARS	5.0438	0.2522	0.1869	0.6567	0.0381	0.7905	0.5840	0.7092	0.8240
	CPA	4.6570	0.2918	0.1769	0.2509	0.0397	0.2128	-0.0511	0.3171	0.5131
	STATE	0.8992	0.0627	0.0770	0.6430	0.0306	0.6604	0.6641	0.7492	0.7841
	CellFlow	1.4261	0.1156	0.1035	0.7757	0.0223	0.8458	0.7464	0.8264	0.7905
	scDFM	0.5562	0.0459	0.0738	0.7876	0.0232	**0.9017**	0.8333	0.8820	**0.9743**
	PerturbCellRL	**0.4865**	**0.0367**	**0.0672**	**0.7973**	**0.0219**	0.9007	**0.8479**	0.8895	0.9558
H-S	Control	4.6858	0.2611	0.1714	-	0.0247	-	0.2657	-	0.5216
	GEARS	4.2806	0.2314	0.1663	0.5602	0.0259	0.7276	0.5424	0.5986	0.8180
	CPA	3.5537	0.2428	0.1504	0.0400	0.0285	0.2979	0.2441	0.2194	0.5230
	STATE	0.8626	0.0670	0.0743	0.5166	0.0295	0.6624	0.3793	0.5259	0.6716
	CellFlow	1.2079	0.1088	0.0890	0.5910	0.0195	0.7847	0.5401	0.5826	0.6344
	scDFM	0.5642	0.0581	0.0736	0.5947	0.0203	**0.8372**	0.6848	0.7182	**0.8904**
	PerturbCellRL	**0.5021**	**0.0478**	**0.0671**	**0.6153**	**0.0195**	0.8342	**0.7099**	**0.7281**	0.8630
H-D	Control	6.2336	0.3224	0.2066	-	0.0414	-	-0.1412	-	0.5333
	GEARS	5.3218	0.2594	0.1951	0.6901	0.0398	0.8121	0.5473	0.7146	0.8433
	CPA	4.9475	0.2953	0.1853	0.0737	0.0437	0.1827	-0.0840	0.2666	0.5333
	STATE	1.2475	0.0841	0.0934	0.6511	0.0380	0.6859	0.4868	0.6561	0.6933
	CellFlow	1.7584	0.1346	0.1133	0.7639	0.0266	0.8368	0.5088	0.7334	0.7244
	scDFM	0.6398	0.0455	0.0767	0.7602	0.0251	**0.8845**	0.7435	0.8279	**0.9144**
	PerturbCellRL	**0.5961**	**0.0388**	**0.0718**	**0.7657**	**0.0246**	0.8777	**0.7526**	**0.8312**	0.8911

Qualitative comparison. We visualize representative held-out perturbations using target-fitted UMAP projections, which reveal whether predicted single-cell populations occupy the same local manifold as the target cells. As shown in Figure 5, PerturbCellRL predictions exhibit tighter overlap with the target high-density regions, whereas scDFM often spreads into displaced areas of the embedding space. These results confirm that post-training improves distributional alignment with target populations. Appendix G provides visualizations for additional perturbations (Figure 6).

4.3 Single-Reward Ablations

As shown in Table 2, single-reward optimization achieves the best score on its targeted criterion but yields uneven improvements across the remaining ones. In contrast, PerturbCellRL, trained with the combined weighted reward, provides the most balanced performance across the three criteria: it achieves the second-best Energy distance and expression support in both settings while maintaining competitive pathway scores. These results suggest that joint optimization balances distributional alignment and single-cell fidelity.

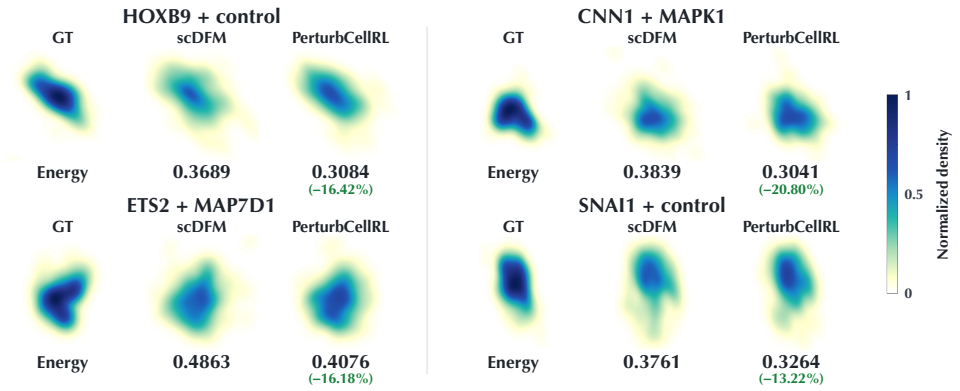


Figure 5: **Cell-density comparisons in UMAP space for four Norman holdout conditions.** Rows show HOXB9 + control, CNN1 + MAPK1, ETS2 + MAP7D1, and SNAI1 + control. Each row compares ground truth (GT), scDFM, and PerturbCellRL. Densities are normalized by the maximum within each row. Annotations report the Energy distance and its relative reduction.

Table 2: **Reward-component ablation.** wit, sup, and path denote gene-expression distribution alignment, expression support, and pathway discrimination rewards, respectively. ✓/× indicate whether each training reward is enabled or disabled. The all-disabled row reports the pretrained scDFM. We evaluate distributional alignment using Energy distance (lower is better), since higher mean witness rewards do not necessarily indicate better alignment. sup and path report mean calibrated rewards (higher is better). Norman holdout includes both single- and double-gene test perturbations. Scores are averaged across test conditions within each fold, then equally across the four folds. Bold and underlining indicate the best and second-best scores, respectively.

Training rewards			**Norman additive**			**Norman holdout**		
wit	sup	path	Energy ↓	sup ↑	path ↑	Energy ↓	sup ↑	path ↑
×	×	×	0.5562	0.9088	0.9464	0.5941	0.9152	0.9145
✓	×	×	**0.4794**	0.9282	<u>0.9474</u>	**0.5370**	0.9371	0.9164
×	✓	×	0.5171	**0.9411**	0.9452	0.5713	**0.9508**	0.9135
×	×	✓	0.5608	0.9073	**0.9503**	0.5951	0.9109	**0.9222**
✓	✓	✓	<u>0.4865</u>	<u>0.9395</u>	0.9470	<u>0.5395</u>	<u>0.9407</u>	<u>0.9192</u>

5 Conclusion

We introduced PerturbCellRL, a framework that post-trains single-cell perturbation generators using population-level and single-cell feedback. Our approach addresses a key limitation of flow-matching generators: the learned generator can fail to reproduce a target distribution even when it lies within the model family. We propose three complementary rewards that guide further optimization. A gene-expression energy witness translates population discrepancies into per-cell rewards, while the calibrated expression-support and pathway-discrimination rewards evaluate individual generated cells for biological fidelity. Building on scDFM, PerturbCellRL improves distributional alignment and biological fidelity across genetic and chemical perturbation benchmarks. Reward ablations indicate that joint optimization effectively balances population alignment, expression support, and perturbation-specific pathway responses. Together, these results establish reward-guided post-training as a principled approach to aligning predicted cell populations with observed responses while enforcing single-cell biological fidelity.

Limitations and future work. Our rewards require access to real target cells and are limited by the coverage of the selected gene and pathway features. Moreover, the single-cell biology rewards are manually designed from domain expertise. Future work could explore reference-free rewards with broader biological coverage and prospective validation with domain biologists.

AI USE STATEMENT

In this work, generative AI tools were used to assist with LaTeX template migration, manuscript formatting, the witness theorem and proof, the flow-matching counterexample and post-training convergence proof, and appendix algorithm description, as well as figure-generation code. The authors reviewed all AI-assisted changes and take responsibility for the final content of this work, including text and artifacts produced with their assistance.

REFERENCES

Abhinav K Adduri, Dhruv Gautam, Beatrice Bevilacqua, Alishba Imran, Rohan Shah, Mohsen Naghipourfar, Noam Teyssier, Rajesh Ilango, Sanjay Nagaraj, Mingze Dong, et al. Predicting cellular responses to perturbation across diverse contexts with state. *BioRxiv*, pp. 2025–06, 2025.

Anastasios N Angelopoulos and Stephen Bates. Conformal prediction: A gentle introduction. *Foundations and Trends in Machine Learning*, 16(4):494–591, 2023. doi: 10.1561/2200000101.

Ding Bai, Le Song, and Eric P Xing. Pertadapt: unlocking single-cell foundation models for genetic perturbation prediction via condition-sensitive adaptation. *Bioinformatics*, 42(Supplement_1): btag307, 2026.

Marc G Bellemare, Ivo Danihelka, Will Dabney, Shakir Mohamed, Balaji Lakshminarayanan, Stephan Hoyer, and Rémi Munos. The cramer distance as a solution to biased wasserstein gradients. *arXiv preprint arXiv:1705.10743*, 2017.

Michael Bereket and Theofanis Karaletsos. Modelling cellular perturbations with the sparse additive mechanism shift variational autoencoder. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 1–12, 2023.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. *arXiv preprint arXiv:2305.13301*, 2023.

Nicolas Bonneel, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. Sliced and radon wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, 51(1):22–45, 2015.

Charlotte Bunne, Yusuf Roohani, Yanay Rosen, Ankit Gupta, Xikun Zhang, Marcel Roed, Theo Alexandrov, Mohammed AlQuraishi, Patricia Brennan, Daniel B Burkhardt, et al. How to build the virtual cell with artificial intelligence: Priorities and opportunities. *Cell*, 2024.

Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. Dpok: Reinforcement learning for fine-tuning text-to-image diffusion models. *Advances in Neural Information Processing Systems*, 36:79858–79885, 2023.

Marc Gillespie, Bijay Jassal, Ralf Stephan, Marija Milacic, Karen Rothfels, Andrea Senff-Ribeiro, Johannes Griss, Cristoffer Sevilla, Lisa Matthews, Chuqiao Gong, et al. The reactome pathway knowledgebase 2022. *Nucleic Acids Research*, 50(D1):D687–D692, 2022. doi: 10.1093/nar/gkab1028.

Arthur Gretton, Karsten M Borgwardt, Mathias J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

Dominik Klein, Jonas Simon Fleck, Daniil Bobrovskiy, Lea Zimmermann, Sören Becker, Alessandro Palma, Leander Dony, Alejandro Tejada-Lapuerta, Guillaume Huguet, Hsiu-Chuan Lin, et al. Cellflow enables generative single-cell phenotype modeling with flow matching. *bioRxiv*, pp. 2025–04, 2025.

Junzhe Li, Yutao Cui, Tao Huang, Weijie Kong, Yiming Cheng, Chuxuan Zeng, Yinping Ma, Chun Fan, Miles Yang, Zhao Zhong, et al. Mixgrpo: Unlocking flow-based grpo efficiency with mixed ode-sde. *arXiv preprint arXiv:2507.21802*, 2025.

Arthur Liberzon, Chet Birger, Helga Thorvaldsdóttir, Mahmoud Ghandi, Jill P Mesirov, and Pablo Tamayo. The molecular signatures database hallmark gene set collection. *Cell systems*, 1(6): 417–425, 2015.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *ICLR*, 2023.

Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky TQ Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code. *arXiv preprint arXiv:2412.06264*, 2024.

Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-grpo: Training flow matching models via online rl. *arXiv preprint arXiv:2505.05470*, 2025.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *ICLR*, 2023.

Romain Lopez, Jeffrey Regier, Michael B Cole, Michael I Jordan, and Nir Yosef. Deep generative modeling for single-cell transcriptomics. *Nature methods*, 15(12):1053–1058, 2018.

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, et al. Predicting cellular responses to complex perturbations in high-throughput screens. *Molecular systems biology*, 19(6):MSB202211517, 2023.

Lukas Mathur, B Szalai, NH Du, Ramesh Utharala, Martine Ballinger, JJM Landry, M Ryckelynck, Vladimir Benes, Julio Saez-Rodriguez, and Christoph A Merten. Combi-seq for multiplexed transcriptome-based profiling of drug combinations using deterministic barcoding in single-cell droplets. *Nature communications*, 13(1):4450, 2022.

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.

Stefan Peidli, Tessa D Green, Ciyue Shen, Torsten Gross, Joseph Min, Samuele Garda, Bo Yuan, Linus J Schumacher, Jake P Taylor-King, Debora S Marks, et al. scperturb: harmonized single-cell perturbation data. *Nature Methods*, 21(3):531–540, 2024.

Ladislav Rampášek, Daniel Hidru, Petr Smirnov, Benjamin Haibe-Kains, and Anna Goldenberg. Dr.vae: improving drug response prediction via modeling of drug perturbation effects. *Bioinformatics*, 35(19):3743–3751, 03 2019. ISSN 1367-4803. doi: 10.1093/bioinformatics/btz158.

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, et al. Virtual cell challenge: Toward a turing test for the virtual cell. *Cell*, 188(13):3370–3374, 2025.

Michael A. Skinnider, Jordan W. Squair, Claudia Kathe, Mark A. Anderson, Matthieu Gautier, Kaya J. E. Matson, Marco Milano, Thomas H. Hutson, Quentin Barraud, Aaron A. Phillips, Leonard J. Foster, Gioele La Manno, Ariel J. Levine, and Grégoire Courtine. Cell type prioritization in single-cell data. *Nature Biotechnology*, 39(1):30–34, 2021. doi: 10.1038/s41587-020-0605-1.

Gábor J Székely and Maria L Rizzo. Energy statistics: A class of statistics based on distances. *Journal of Statistical Planning and Inference*, 143(8):1249–1272, 2013. doi: 10.1016/j.jspi.2013.03.018.

Ramon Viñas Torné, Maciej Wiatrak, Zoe Piran, Shuyang Fan, Liangze Jiang, Sarah A Teichmann, Mor Nitzan, and Maria Brbić. Systema: a framework for evaluating genetic perturbation response prediction beyond systematic variation. *Nature Biotechnology*, pp. 1–10, 2025.

Dongxia Wu, Shiye Su, Yuhui Zhang, Elaine Sui, Emma Lundberg, Emily B Fox, and Serena Yeung-Levy. Cellfluxrl: Biologically-constrained virtual cell modeling via reinforcement learning. *arXiv preprint arXiv:2603.21743*, 2026.

Zeyue Xue, Jie Wu, Yu Gao, Fangyuan Kong, Lingting Zhu, Mengzhao Chen, Zhiheng Liu, Wei Liu, Qiushan Guo, Weilin Huang, et al. Dancegrpo: Unleashing grpo on visual generation. *arXiv preprint arXiv:2505.07818*, 2025.

Chenglei Yu, Chuanrui Wang, Bangyan Liao, and Tailin Wu. scdfm: Distributional flow matching model for robust single-cell perturbation prediction. *arXiv preprint arXiv:2602.07103*, 2026.

Yuhui Zhang, Yuchang Su, Chenyu Wang, Tianhong Li, Zoe Wefers, Jeffrey Nirschl, James Burgess, Daisy Ding, Alejandro Lozano, Emma Lundberg, et al. Cellflux: Simulating cellular morphology changes via flow matching. *arXiv preprint arXiv:2502.09775*, 2025.

Kaiwen Zheng, Huayu Chen, Haotian Ye, Haoxiang Wang, Qinsheng Zhang, Kai Jiang, Hang Su, Stefano Ermon, Jun Zhu, and Ming-Yu Liu. Diffusionnft: Online diffusion reinforcement with forward process. *arXiv preprint arXiv:2509.16117*, 2025.

APPENDIX CONTENTS

A Witness Rewards and Distributional Alignment p. 14

We prove the witness policy-gradient identity and its connection to energy-distance minimization.

B Flow-matching Mismatch and Recovery by Witness Post-training p. 17

We derive a flow-matching counterexample and prove convergence of witness post-training within the same model family.

C Algorithm Details p. 19

We provide the post-training algorithm, empirical witness estimator, implementation details, and training hyperparameters.

D Single-Cell Reward Calibration p. 20

We describe scorer fitting and held-out calibration for the expression-support and pathway-discrimination rewards.

E Evaluation Metrics p. 21

We define the distribution-level, pathway-enrichment, and pseudobulk-mean metrics used to evaluate perturbation predictions.

F Dataset Details p. 22

We document the Norman and ComboSciPlex dataset splits and list their held-out perturbation conditions.

G Additional Perturbation Visualizations p. 25

We compare predicted and real cell densities for additional double-gene and drug-combination perturbations.

H Responsible Use p. 25

We discuss intended use, potential prediction failures, and the need for independent experimental validation.

A WITNESS REWARDS AND DISTRIBUTIONAL ALIGNMENT

We restate Theorem 1 and give the full derivation.

Theorem 1 (Witness policy-gradient identity, restated)**.** *Fix a perturbation c, control distribution P_0, target distribution $P = P_c$, and feature map $\phi : \mathcal{X} \to \mathbb{R}^d$. Let $Q_{\theta,c}$ be the marginal distribution obtained by sampling $\boldsymbol{u} \sim P_0$ and then $\hat{\boldsymbol{x}}_1 \sim \pi_\theta(\cdot \mid \boldsymbol{u}, c)$. Define $d_\phi(\hat{\boldsymbol{x}}_1, \boldsymbol{x}_1) = \|\phi(\hat{\boldsymbol{x}}_1) - \phi(\boldsymbol{x}_1)\|_2$ and*

$$\begin{aligned} D_\phi(Q_{\theta,c}, P) &= 2\mathbb{E}_{Q_{\theta,c}\otimes P} d_\phi - \mathbb{E}_{Q_{\theta,c}\otimes Q_{\theta,c}} d_\phi - \mathbb{E}_{P\otimes P} d_\phi, \\ w^\phi_{Q_{\theta,c},P}(\hat{\boldsymbol{x}}_1) &= \mathbb{E}_{\hat{\boldsymbol{x}}'_1 \sim Q_{\theta,c}} d_\phi(\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}'_1) - \mathbb{E}_{\boldsymbol{x}_1 \sim P} d_\phi(\hat{\boldsymbol{x}}_1, \boldsymbol{x}_1). \end{aligned} \tag{16}$$

Suppose P and $Q_{\theta,c}$ have finite first feature moments. Assume conditional densities exist relative to a common dominating measure, are differentiable, and have parameter-independent support. Assume differentiation commutes with the relevant expectations, and the resulting score-weighted integrals are finite. Treat the current witness $w^\phi_{Q_{\theta,c},P}$ as a fixed reward. Its on-policy gradient satisfies

$$\begin{aligned} g_\phi(\theta) &:= \mathbb{E}_{\substack{\boldsymbol{u}\sim P_0 \\ \hat{\boldsymbol{x}}_1 \sim \pi_\theta(\cdot \mid \boldsymbol{u}, c)}} \left[w^\phi_{Q_{\theta,c},P}(\hat{\boldsymbol{x}}_1) \nabla_\theta \log \pi_\theta(\hat{\boldsymbol{x}}_1 \mid \boldsymbol{u}, c) \right] \\ &= -\frac{1}{2} \nabla_\theta D_\phi(Q_{\theta,c}, P). \end{aligned} \tag{17}$$

Whenever $g_\phi(\theta) \neq 0$, it is a descent direction for D_ϕ.

Proof. We use direct differentiation of the energy-distance expectations, as in Bellemare et al. (2017, Appendix A, Eq. (8)). Write $Q_\theta = Q_{\theta,c}$, $d = d_\phi$, and $w_\theta = w^\phi_{Q_{\theta,c},P}$ throughout the proof. The subscript on w_θ records which population defines the witness. Only the generator depends on θ; P_0, P, and d remain fixed.

Step 1: Set up the distance terms. Let $\mathrm{d}\hat{\boldsymbol{x}}_1$ denote integration against the common dominating measure. The target P need not have a density against this measure. Let q_θ denote the density of Q_θ. The population definition in equation 2 reads

$$q_\theta(\hat{\boldsymbol{x}}_1) = \int \pi_\theta(\hat{\boldsymbol{x}}_1 \mid \boldsymbol{u}, c) P_0(\mathrm{d}\boldsymbol{u}). \tag{18}$$

We distinguish three types of pairwise terms: generated–real, generated–generated, and real–real. The symbols C_θ, S_θ, and R identify them independently of color:

$$\begin{aligned} C_\theta &:= \mathbb{E}_{Q_\theta \otimes P} d = \int\int d(\hat{\boldsymbol{x}}_1, \boldsymbol{x}_1) q_\theta(\hat{\boldsymbol{x}}_1) P(\mathrm{d}\boldsymbol{x}_1)\, \mathrm{d}\hat{\boldsymbol{x}}_1, \\ S_\theta &:= \mathbb{E}_{Q_\theta \otimes Q_\theta} d = \int\int d(\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}'_1) q_\theta(\hat{\boldsymbol{x}}_1) q_\theta(\hat{\boldsymbol{x}}'_1)\, \mathrm{d}\hat{\boldsymbol{x}}'_1\, \mathrm{d}\hat{\boldsymbol{x}}_1, \\ R &:= \mathbb{E}_{P\otimes P} d = \int\int d(\boldsymbol{x}_1, \boldsymbol{x}'_1) P(\mathrm{d}\boldsymbol{x}'_1) P(\mathrm{d}\boldsymbol{x}_1). \end{aligned} \tag{19}$$

The density product in S_θ follows from independent sampling. By definition,

$$D_\phi(Q_\theta, P) = 2C_\theta - S_\theta - R. \tag{20}$$

For a fixed query $\hat{\boldsymbol{x}}_1$, define its mean distances by

$$\begin{aligned} a(\hat{\boldsymbol{x}}_1) &:= \int d(\hat{\boldsymbol{x}}_1, \boldsymbol{x}_1) P(\mathrm{d}\boldsymbol{x}_1), \\ b_\theta(\hat{\boldsymbol{x}}_1) &:= \int d(\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}'_1) q_\theta(\hat{\boldsymbol{x}}'_1)\, \mathrm{d}\hat{\boldsymbol{x}}'_1, \\ w_\theta(\hat{\boldsymbol{x}}_1) &= b_\theta(\hat{\boldsymbol{x}}_1) - a(\hat{\boldsymbol{x}}_1). \end{aligned} \tag{21}$$

Here a is fixed, whereas b_θ depends on the generator. Finite first feature moments make these distance expectations finite, by $d(\hat{\boldsymbol{x}}_1, \boldsymbol{x}_1) \le \|\phi(\hat{\boldsymbol{x}}_1)\|_2 + \|\phi(\boldsymbol{x}_1)\|_2$. All derivative and integration interchanges below use the stated assumptions.

Step 2: Differentiate the generated–real term. In C_θ, only $q_\theta(\hat{\boldsymbol{x}}_1)$ depends on θ. Moving the derivative inside the integrals and collecting terms independent of $\boldsymbol{x}_1$ gives

$$\begin{aligned}\nabla_\theta C_\theta &= \nabla_\theta \int\int d(\hat{\boldsymbol{x}}_1, \boldsymbol{x}_1) q_\theta(\hat{\boldsymbol{x}}_1) P(\mathrm{d}\boldsymbol{x}_1)\, \mathrm{d}\hat{\boldsymbol{x}}_1 \\ &= \int\int d(\hat{\boldsymbol{x}}_1, \boldsymbol{x}_1) \nabla_\theta q_\theta(\hat{\boldsymbol{x}}_1) P(\mathrm{d}\boldsymbol{x}_1)\, \mathrm{d}\hat{\boldsymbol{x}}_1 \\ &= \int \left[\int d(\hat{\boldsymbol{x}}_1, \boldsymbol{x}_1) P(\mathrm{d}\boldsymbol{x}_1)\right] \nabla_\theta q_\theta(\hat{\boldsymbol{x}}_1)\, \mathrm{d}\hat{\boldsymbol{x}}_1 \\ &= \int a(\hat{\boldsymbol{x}}_1) \nabla_\theta q_\theta(\hat{\boldsymbol{x}}_1)\, \mathrm{d}\hat{\boldsymbol{x}}_1.\end{aligned} \tag{22}$$

Step 3: Differentiate both copies of the generated distribution. In S_θ, both density factors depend on θ. The product rule therefore produces two terms:

$$\begin{aligned}\nabla_\theta S_\theta &= \nabla_\theta \int\int d(\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}_1') q_\theta(\hat{\boldsymbol{x}}_1) q_\theta(\hat{\boldsymbol{x}}_1')\, \mathrm{d}\hat{\boldsymbol{x}}_1'\, \mathrm{d}\hat{\boldsymbol{x}}_1 \\ &= \int\int d(\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}_1') \nabla_\theta [q_\theta(\hat{\boldsymbol{x}}_1) q_\theta(\hat{\boldsymbol{x}}_1')]\, \mathrm{d}\hat{\boldsymbol{x}}_1'\, \mathrm{d}\hat{\boldsymbol{x}}_1 \\ &= \underbrace{\int\int d(\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}_1') [\nabla_\theta q_\theta(\hat{\boldsymbol{x}}_1)] q_\theta(\hat{\boldsymbol{x}}_1')\, \mathrm{d}\hat{\boldsymbol{x}}_1'\, \mathrm{d}\hat{\boldsymbol{x}}_1}_{I_1} \\ &\quad + \underbrace{\int\int d(\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}_1') q_\theta(\hat{\boldsymbol{x}}_1) [\nabla_\theta q_\theta(\hat{\boldsymbol{x}}_1')]\, \mathrm{d}\hat{\boldsymbol{x}}_1'\, \mathrm{d}\hat{\boldsymbol{x}}_1}_{I_2}.\end{aligned} \tag{23}$$

For I_1, integrate over $\hat{\boldsymbol{x}}_1'$ first:

$$\begin{aligned}I_1 &= \int \left[\int d(\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}_1') q_\theta(\hat{\boldsymbol{x}}_1')\, \mathrm{d}\hat{\boldsymbol{x}}_1'\right] \nabla_\theta q_\theta(\hat{\boldsymbol{x}}_1)\, \mathrm{d}\hat{\boldsymbol{x}}_1 \\ &= \int b_\theta(\hat{\boldsymbol{x}}_1) \nabla_\theta q_\theta(\hat{\boldsymbol{x}}_1)\, \mathrm{d}\hat{\boldsymbol{x}}_1.\end{aligned} \tag{24}$$

For I_2, first exchange the integration order. Next, swap the dummy names $\hat{\boldsymbol{x}}_1$ and $\hat{\boldsymbol{x}}_1'$. Finally, use the symmetry $d(\hat{\boldsymbol{x}}_1', \hat{\boldsymbol{x}}_1) = d(\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}_1')$:

$$\begin{aligned}I_2 &= \int\int d(\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}_1') q_\theta(\hat{\boldsymbol{x}}_1) \nabla_\theta q_\theta(\hat{\boldsymbol{x}}_1')\, \mathrm{d}\hat{\boldsymbol{x}}_1'\, \mathrm{d}\hat{\boldsymbol{x}}_1 \\ &= \int\int d(\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}_1') q_\theta(\hat{\boldsymbol{x}}_1) \nabla_\theta q_\theta(\hat{\boldsymbol{x}}_1')\, \mathrm{d}\hat{\boldsymbol{x}}_1\, \mathrm{d}\hat{\boldsymbol{x}}_1' \\ &= \int\int d(\hat{\boldsymbol{x}}_1', \hat{\boldsymbol{x}}_1) q_\theta(\hat{\boldsymbol{x}}_1') \nabla_\theta q_\theta(\hat{\boldsymbol{x}}_1)\, \mathrm{d}\hat{\boldsymbol{x}}_1'\, \mathrm{d}\hat{\boldsymbol{x}}_1 \\ &= \int\int d(\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}_1') q_\theta(\hat{\boldsymbol{x}}_1') \nabla_\theta q_\theta(\hat{\boldsymbol{x}}_1)\, \mathrm{d}\hat{\boldsymbol{x}}_1'\, \mathrm{d}\hat{\boldsymbol{x}}_1 \\ &= \int \left[\int d(\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}_1') q_\theta(\hat{\boldsymbol{x}}_1')\, \mathrm{d}\hat{\boldsymbol{x}}_1'\right] \nabla_\theta q_\theta(\hat{\boldsymbol{x}}_1)\, \mathrm{d}\hat{\boldsymbol{x}}_1 \\ &= \int b_\theta(\hat{\boldsymbol{x}}_1) \nabla_\theta q_\theta(\hat{\boldsymbol{x}}_1)\, \mathrm{d}\hat{\boldsymbol{x}}_1 = I_1.\end{aligned} \tag{25}$$

Combining equation 23, equation 24, and equation 25 gives

$$\begin{aligned}\nabla_\theta S_\theta &= I_1 + I_2 \\ &= 2I_1 \\ &= 2 \int b_\theta(\hat{\boldsymbol{x}}_1) \nabla_\theta q_\theta(\hat{\boldsymbol{x}}_1)\, \mathrm{d}\hat{\boldsymbol{x}}_1.\end{aligned} \tag{26}$$

This is the source of the factor 2.

Step 4: Combine all three distance terms. The real–real term depends only on P and d, so it has zero derivative:

$$\begin{aligned}\nabla_\theta R &= \nabla_\theta \int\int d(\boldsymbol{x}_1, \boldsymbol{x}_1') P(\mathrm{d}\boldsymbol{x}_1') P(\mathrm{d}\boldsymbol{x}_1) \\ &= 0.\end{aligned} \tag{27}$$

Differentiating equation 20 and substituting equation 22, equation 26, and equation 27 yields

$$\begin{aligned}\nabla_\theta D_\phi(Q_\theta, P) &= 2\nabla_\theta C_\theta - \nabla_\theta S_\theta - \nabla_\theta R \\ &= 2\int a(\hat{\boldsymbol{x}}_1)\nabla_\theta q_\theta(\hat{\boldsymbol{x}}_1)\,\mathrm{d}\hat{\boldsymbol{x}}_1 \\ &\quad - 2\int b_\theta(\hat{\boldsymbol{x}}_1)\nabla_\theta q_\theta(\hat{\boldsymbol{x}}_1)\,\mathrm{d}\hat{\boldsymbol{x}}_1 - 0 \\ &= 2\int [a(\hat{\boldsymbol{x}}_1) - b_\theta(\hat{\boldsymbol{x}}_1)]\nabla_\theta q_\theta(\hat{\boldsymbol{x}}_1)\,\mathrm{d}\hat{\boldsymbol{x}}_1 \\ &= -2\int [b_\theta(\hat{\boldsymbol{x}}_1) - a(\hat{\boldsymbol{x}}_1)]\nabla_\theta q_\theta(\hat{\boldsymbol{x}}_1)\,\mathrm{d}\hat{\boldsymbol{x}}_1 \\ &= -2\int w_\theta(\hat{\boldsymbol{x}}_1)\nabla_\theta q_\theta(\hat{\boldsymbol{x}}_1)\,\mathrm{d}\hat{\boldsymbol{x}}_1.\end{aligned} \tag{28}$$

The last equality uses the witness definition in equation 21.

Step 5: Convert the density derivative into a policy gradient. On the support of the conditional density, the derivative of the logarithm gives

$$\begin{aligned}\nabla_\theta \log \pi_\theta(\hat{\boldsymbol{x}}_1 \mid \boldsymbol{u}, c) &= \frac{\nabla_\theta \pi_\theta(\hat{\boldsymbol{x}}_1 \mid \boldsymbol{u}, c)}{\pi_\theta(\hat{\boldsymbol{x}}_1 \mid \boldsymbol{u}, c)}, \\ \nabla_\theta \pi_\theta(\hat{\boldsymbol{x}}_1 \mid \boldsymbol{u}, c) &= \pi_\theta(\hat{\boldsymbol{x}}_1 \mid \boldsymbol{u}, c)\nabla_\theta \log \pi_\theta(\hat{\boldsymbol{x}}_1 \mid \boldsymbol{u}, c).\end{aligned} \tag{29}$$

These identities are used almost everywhere under the conditional law. Since P_0 is fixed, differentiating equation 18 gives

$$\begin{aligned}\nabla_\theta q_\theta(\hat{\boldsymbol{x}}_1) &= \nabla_\theta \int \pi_\theta(\hat{\boldsymbol{x}}_1 \mid \boldsymbol{u}, c) P_0(\mathrm{d}\boldsymbol{u}) \\ &= \int \nabla_\theta \pi_\theta(\hat{\boldsymbol{x}}_1 \mid \boldsymbol{u}, c) P_0(\mathrm{d}\boldsymbol{u}) \\ &= \int \pi_\theta(\hat{\boldsymbol{x}}_1 \mid \boldsymbol{u}, c)\nabla_\theta \log \pi_\theta(\hat{\boldsymbol{x}}_1 \mid \boldsymbol{u}, c) P_0(\mathrm{d}\boldsymbol{u}).\end{aligned} \tag{30}$$

Substitute this into equation 28, exchange the integrals over $\hat{\boldsymbol{x}}_1$ and $\boldsymbol{u}$, and recognize the expectation under the sampling policy:

$$\begin{aligned}\nabla_\theta D_\phi(Q_\theta, P) &= -2\int w_\theta(\hat{\boldsymbol{x}}_1)\nabla_\theta q_\theta(\hat{\boldsymbol{x}}_1)\,\mathrm{d}\hat{\boldsymbol{x}}_1 \\ &= -2\int w_\theta(\hat{\boldsymbol{x}}_1)\left[\int \pi_\theta(\hat{\boldsymbol{x}}_1 \mid \boldsymbol{u}, c)\nabla_\theta \log \pi_\theta(\hat{\boldsymbol{x}}_1 \mid \boldsymbol{u}, c) P_0(\mathrm{d}\boldsymbol{u})\right]\mathrm{d}\hat{\boldsymbol{x}}_1 \\ &= -2\int P_0(\mathrm{d}\boldsymbol{u})\left[\int w_\theta(\hat{\boldsymbol{x}}_1)\pi_\theta(\hat{\boldsymbol{x}}_1 \mid \boldsymbol{u}, c)\nabla_\theta \log \pi_\theta(\hat{\boldsymbol{x}}_1 \mid \boldsymbol{u}, c)\,\mathrm{d}\hat{\boldsymbol{x}}_1\right] \\ &= -2\mathbb{E}_{\boldsymbol{u}\sim P_0}\mathbb{E}_{\hat{\boldsymbol{x}}_1\sim\pi_\theta(\cdot\mid\boldsymbol{u},c)}\left[w_\theta(\hat{\boldsymbol{x}}_1)\nabla_\theta \log \pi_\theta(\hat{\boldsymbol{x}}_1 \mid \boldsymbol{u}, c)\right] \\ &= -2g_\phi(\theta).\end{aligned} \tag{31}$$

Dividing both sides by -2 proves equation 17.

Finally, whenever $g_\phi(\theta) \neq 0$,

$$\nabla_\theta D_\phi(Q_\theta, P)^\top g_\phi(\theta) = -\tfrac{1}{2}\|\nabla_\theta D_\phi(Q_\theta, P)\|_2^2 < 0, \tag{32}$$

so $g_\phi(\theta)$ is a descent direction for D_ϕ. □

Corollary 1 (Witness gap and distributional agreement)**.** *For any P, Q with finite first feature moments, the witness $w = w^{\phi}_{Q,P}$ satisfies*

$$\mathbb{E}_P w - \mathbb{E}_Q w = D_\phi(Q, P) \geq 0. \tag{33}$$

Equality holds exactly when $Q_\phi = P_\phi$, where Q_ϕ and P_ϕ are the corresponding feature distributions.

Proof. Expanding both witness expectations and using symmetry yields

$$\begin{aligned} \mathbb{E}_P w - \mathbb{E}_Q w &= 2\mathbb{E}_{Q\otimes P} d_\phi - \mathbb{E}_{Q\otimes Q} d_\phi - \mathbb{E}_{P\otimes P} d_\phi \\ &= D_\phi(Q, P). \end{aligned} \tag{34}$$

Finite feature moments ensure all expectations are finite. Since $D_\phi(Q, P) = \mathrm{ED}(Q_\phi, P_\phi)$, Euclidean energy distance is nonnegative and vanishes exactly when these distributions agree (Székely & Rizzo, 2013). □

B Flow-matching Mismatch and Recovery by Witness Post-training

We prove Proposition 1 using the two-dimensional Gaussian example shown in Figure 1. Both training stages optimize the same scalar parameter. Let $Z \sim \mathcal{N}(0, I_2)$ and consider

$$v_b(x, t) = -bx, \qquad b \geq 0. \tag{35}$$

Its exact time-one map and endpoint distribution are

$$T_b(z) = e^{-b} z, \qquad Q_b = \mathcal{N}(0, e^{-2b} I_2). \tag{36}$$

Set $\tau = e^{-1}$ and $P = Q_1 = \mathcal{N}(0, \tau^2 I_2)$. The target is therefore exactly representable at $b = 1$. All endpoint distributions have the correct mean, zero. Let $X_1 \sim P$ and $t \sim \mathcal{U}[0, 1]$ be independent of each other and of Z. Training uses $X_t = (1 - t)Z + tX_1$ and target velocity $X_1 - Z$.

Proof. **The population FM minimizer misses the target.** The velocity-regression residual is

$$-bX_t - (X_1 - Z) = [1 - b(1 - t)]Z - (1 + bt)X_1. \tag{37}$$

Independence and zero means eliminate the cross term. Using $\mathbb{E}\|Z\|^2 = 2$ and $\mathbb{E}\|X_1\|^2 = 2\tau^2$ gives

$$\begin{aligned} L_{\mathrm{FM}}(b) &= 2 \int_0^1 \left([1 - b(1 - t)]^2 + \tau^2 (1 + bt)^2 \right) \mathrm{d}t \\ &= \frac{2(1 + \tau^2)}{3} b^2 - 2(1 - \tau^2) b + 2(1 + \tau^2). \end{aligned} \tag{38}$$

This quadratic is strictly convex. Its unique minimizer is positive and therefore feasible:

$$b^\star_{\mathrm{FM}} = \frac{3(1 - e^{-2})}{2(1 + e^{-2})} = \frac{3}{2} \tanh(1) \approx 1.142391. \tag{39}$$

Since $e^2 > 5$, this minimizer exceeds 1. Consequently, $Q_{b^\star_{\mathrm{FM}}} \neq P$: its mean is correct, but its variance is too small. The mismatch holds under population training and exact flow integration.

Closed-form terminal energy distance. Write $s = e^{-b}$ and $c_2 = \mathbb{E}\|Z\| = \sqrt{\pi/2}$. For independent $\hat{X}_1, \hat{X}_1' \sim Q_b$ and $X_1, X_1' \sim P$,

$$\begin{aligned} \hat{X}_1 - X_1 &\sim \mathcal{N}(0, (s^2 + \tau^2) I_2), \\ \hat{X}_1 - \hat{X}_1' &\sim \mathcal{N}(0, 2s^2 I_2), \\ X_1 - X_1' &\sim \mathcal{N}(0, 2\tau^2 I_2). \end{aligned} \tag{40}$$

Thus the Euclidean energy distance in equation 7 is

$$D(b) := D_{\mathrm{id}}(Q_b, P) = c_2 \left[2\sqrt{s^2 + \tau^2} - \sqrt{2}(s + \tau) \right]. \tag{41}$$

The inequality $2(s^2+\tau^2) \geq (s+\tau)^2$ shows that $D(b) \geq 0$, with equality exactly when $s = \tau$. Therefore $b = 1$ uniquely minimizes terminal energy distance.

The witness gradient points toward the correct parameter. At each iteration, form the current energy witness

$$w_b(x) = \mathbb{E}_{\hat{X}_1' \sim Q_b} \|x - \hat{X}_1'\| - \mathbb{E}_{X_1 \sim P} \|x - X_1\|. \tag{42}$$

Hold this witness fixed while differentiating the sampling policy. The Gaussian density has common support $\mathbb{R}^2$, with score

$$\partial_b \log q_b(x) = 2 - e^{2b}\|x\|^2. \tag{43}$$

On compact parameter intervals, Gaussian tails justify differentiation under expectations. Theorem 1, with identity features and no conditioning, gives

$$\begin{aligned} G(b) &:= \mathbb{E}_{\hat{X}_1 \sim Q_b} \left[w_b(\hat{X}_1) \partial_b \log q_b(\hat{X}_1) \right] \\ &= -\frac{1}{2} D'(b) = c_2 s \left[\frac{s}{\sqrt{s^2+\tau^2}} - \frac{1}{\sqrt{2}} \right]. \end{aligned} \tag{44}$$

In particular,

$$G(b) > 0 \ (b < 1), \qquad G(1) = 0, \qquad G(b) < 0 \ (b > 1). \tag{45}$$

There are no other stationary points at finite $b \geq 0$. The continuous parameter flow $\dot{b} = G(b)$ stays between its initialization and 1 and converges to 1. Indeed, the trajectory is monotone and bounded; any limit different from 1 would have nonzero derivative.

Discrete parameter updates also converge. We give an explicit sufficient constant step size. Let $u = s/\sqrt{s^2+\tau^2} \in [0,1]$. Differentiation of equation 44 yields

$$G'(b) = -c_2 s \left(2u - u^3 - \frac{1}{\sqrt{2}} \right). \tag{46}$$

Since $0 \leq 2u - u^3 \leq \sqrt{2}$ and $s \leq 1$,

$$|G'(b)| \leq \frac{c_2}{\sqrt{2}} = \frac{\sqrt{\pi}}{2} =: L, \qquad b \geq 0. \tag{47}$$

For any finite initialization $b_0 \geq 0$ different from 1, let $I = [\min\{b_0, 1\}, \max\{b_0, 1\}]$. Define

$$k(b) = \begin{cases} G(b)/(1-b), & b \neq 1, \\ -G'(1) = c_2\tau/(2\sqrt{2}), & b = 1. \end{cases} \tag{48}$$

This function is continuous and strictly positive on I. The derivative bound and $G(1) = 0$ imply $k(b) \leq L$. Compactness gives $k_{\min} := \min_{b \in I} k(b) > 0$. For any fixed $0 < \eta \leq 1/L = 2/\sqrt{\pi}$, consider

$$b_{n+1} = b_n + \eta G(b_n). \tag{49}$$

Then

$$b_{n+1} - 1 = [1 - \eta k(b_n)](b_n - 1). \tag{50}$$

Each iterate lies between the previous iterate and 1. Inductively, all iterates remain in I, and

$$|b_n - 1| \leq (1 - \eta k_{\min})^n |b_0 - 1| \longrightarrow 0. \tag{51}$$

Thus $Q_{b_n} \to P$ and $D(b_n) \to 0$. Taking $b_0 = b^\star_{\mathrm{FM}}$ proves the post-training claim. Every update stays within the original parameterized model family. □

The mismatch also holds for unweighted data-prediction loss. For the same velocity family, define the clean-data prediction

$$\widehat{X}_{1,b}(x,t) = x + (1-t)v_b(x,t) = [1 - b(1-t)]x. \tag{52}$$

This is a denoising prediction, not an integrated ODE endpoint. Because $X_1 = X_t + (1-t)(X_1 - Z)$, its residual satisfies

$$\widehat{X}_{1,b}(X_t,t) - X_1 = (1-t)[v_b(X_t,t) - (X_1 - Z)]. \tag{53}$$

With uniform time sampling, the unweighted data loss becomes

$$\begin{aligned} L_x(b) &= \mathbb{E}\|\widehat{X}_{1,b}(X_t, t) - X_1\|^2 \\ &= \frac{6+\tau^2}{15}b^2 - \frac{3-\tau^2}{3}b + \frac{2(1+\tau^2)}{3}. \end{aligned} \tag{54}$$

Its unique minimizer is

$$b_x^\star = \frac{5(3-e^{-2})}{2(6+e^{-2})} \approx 1.167281 > 1. \tag{55}$$

The strict inequality follows from $e^2 > 7/3$. The same witness updates converge from $b_x^\star$ to 1.

Figure reproduction. Figure 1 uses equation 49 with $\eta = 1$ for 100 updates from $b_{\mathrm{FM}}^\star$. All three panels display exact Gaussian densities on a shared grid with a common linear color scale. The final parameter differs from 1 by less than 10^{-8}. Reported variance ratios are e^{-2b}/e^{-2} for either coordinate. Training updates use exact population gradients.

C Algorithm Details

Algorithm 1 summarizes the training procedure. The implicit branches in equation 5 express reward-guided updates through a single trainable velocity field. The data-collection policy is frozen during each iteration and synchronized with the current model at its end. The pretrained reference remains fixed throughout training. We separately maintain an exponential moving average (EMA) for checkpointing and evaluation.

Each batch fixes a perturbation c and independently samples K controls $\boldsymbol{u}_g \sim P_0$, with m rollouts per control. The rollouts from each sampled control form a sampling group g. We index generated endpoints by $i = (g, j)$. Equation 6 maps their advantages A_i to preference weights r_i before optimization, as shown in Algorithm 1. Training noise is sampled independently of the rollout starts. All velocities in equation 6 are evaluated at the same state $\boldsymbol{x}_{t_i}$, time t_i, and conditioning $(\boldsymbol{u}_g, c)$. The expectation covers batches, endpoints, training noise, and times.

Empirical witness estimation. Let $g(i)$ identify the sampling group of generated cell $\hat{\boldsymbol{x}}_{1,i}$. Each group corresponds to one independent control sampling event. Repeated draws of the same real cell remain separate groups. To estimate the expectation over $Q_{\theta,c}$, we use $\mathcal{I}_i = \{k : c_k = c,\ g(k) \neq g(i)\}$. This excludes $\hat{\boldsymbol{x}}_{1,i}$ and rollouts sharing its sampled control, so reference groups use independently sampled controls. Given M_c real responses $\{\boldsymbol{x}_{1,j}\}_{j=1}^{M_c}$ from P_c, the empirical witness is

$$\widehat{w}_i^\phi = \frac{1}{|\mathcal{I}_i|}\sum_{k\in\mathcal{I}_i} d_\phi(\hat{\boldsymbol{x}}_{1,i}, \hat{\boldsymbol{x}}_{1,k}) - \frac{1}{M_c}\sum_{j=1}^{M_c} d_\phi(\hat{\boldsymbol{x}}_{1,i}, \boldsymbol{x}_{1,j}). \tag{56}$$

Implementation. Rewards use nonnegative-clipped endpoints, while the NFT target uses the original, unclipped model endpoint. Generated endpoints, advantages, and old/reference predictions are detached. We use bf16 mixed precision for the model and combine NFT loss operands in FP32. Each endpoint receives one fresh noise draw and one training time. The operator z subtracts the within-group mean and divides by the within-group standard deviation, with variance computed using divisor m. For joint training, the implementation raises an error if the gene-expression witness or combined reward has zero within-group standard deviation; constant single-cell rewards are permitted. Raw witness credits are not clipped; clipping applies to the final advantages before NFT weighting. Specifically, $\widetilde{A}_i = \mathrm{clip}(A_i, -C, C)/C$ in equation 6, where $C > 0$ is the clipping threshold. The implementation averages squared errors over d velocity features and multiplies the policy loss by C:

$$\mathcal{L}_{\mathrm{impl}}(\theta) = \mathbb{E}\Big[\frac{C}{d}\big(r_i\|\boldsymbol{v}_\theta^+ - \boldsymbol{v}\|_2^2 + (1-r_i)\|\boldsymbol{v}_\theta^- - \boldsymbol{v}\|_2^2\big) + \frac{\lambda_{\mathrm{ref}}}{d}\|\boldsymbol{v}_\theta - \boldsymbol{v}_{\theta_0}\|_2^2\Big]. \tag{57}$$

The effective main-text weight is $\lambda_{\mathrm{reg}} = \lambda_{\mathrm{ref}}/C$, so $\mathcal{L}_{\mathrm{impl}} = (C/d)\mathcal{L}$. Thus, equation 6 preserves the relative policy and reference weights while omitting a positive global scale. Table 3 reports the implementation weight λ_{ref}. Velocity regularization implements the proximity goal in equation 3.

Algorithm 1 *PerturbCellRL*: energy-witness RL for scDFM

Require: Pretrained parameters θ_0; training perturbations; control distribution P_0; K groups with m rollouts each; reward weights; γ, C, λ_{ref}

Initialize $\theta, \theta^{\text{old}}, \theta^{\text{EMA}} \leftarrow \theta_0$; keep θ_0 fixed
for each RL iteration **do**
  Sample $c \sim \text{Unif}(\mathcal{C}_{\text{train}})$ and $\{\boldsymbol{u}_g\}_{g=1}^{K} \sim P_0^{\otimes K}$
  for each group g and rollout j **do**
    Draw rollout noise $\boldsymbol{x}_{0,g,j}^{\text{roll}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$; integrate $\boldsymbol{v}^{\text{old}}$ conditioned on $(\boldsymbol{u}_g, c)$ to generate $\hat{\boldsymbol{x}}_{1,g,j}$
  end for
  Score the complete batch at $\max(\hat{\boldsymbol{x}}_{1,g,j}, 0)$; compute R_i using equation 15 and standardize within groups to obtain $A_i = z(R)_i$
  Set $r_i = \frac{1}{2} + \text{clip}(A_i, -C, C)/(2C)$; detach endpoints and weights
  Shuffle endpoints and split into training microbatches; zero accumulated gradients
  for each microbatch **do**
    Independently draw fresh training noise $\boldsymbol{x}_{0,i} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ and $t_i \sim \mathcal{U}[0,1]$ for each endpoint
    Construct $\boldsymbol{x}_{t_i} = (1-t_i)\boldsymbol{x}_{0,i} + t_i\hat{\boldsymbol{x}}_{1,i}$ and $\boldsymbol{v}_i = \hat{\boldsymbol{x}}_{1,i} - \boldsymbol{x}_{0,i}$
    Compute equation 57; accumulate gradients with respect to θ only
    if the accumulation window is complete **then**
      Clip gradients; update θ; zero gradients; update θ^{EMA}
    end if
  end for
  Synchronize $\theta^{\text{old}} \leftarrow \theta$
end for
return EMA checkpoint for evaluation

Table 3: NFT post-training settings.

Parameter	Additive	Holdout	ComboSciPlex
$\lambda_{\text{sup}}, \lambda_{\text{path}}$	$3, 1$	$1, 3$	$3, 1$
Groups × rollouts	8×8	16×8	8×8
Microbatch / accumulation	$32 \,/\, 2$	$32 \,/\, 4$	$32 \,/\, 2$
Reference weight λ_{ref}	2	2	1
Training α_{path}	0.05	0.20	0.05

Training settings. Table 3 lists setting-specific parameters. All runs use 200 optimizer updates, a learning rate of 5×10^{-7}, $\gamma = 1$, $C = 2$, and a maximum gradient norm of 0.25. Each iteration collects one sampling batch and trains for one inner epoch, producing one optimizer update. Rollouts use RK4 with 20 time points (19 intervals). Both the global seed and training-noise seed are 0. The old-policy decay is 0, corresponding to direct synchronization. The evaluation EMA uses decay $\min\{(1+s)/(10+s), 0.9\}$ at optimizer step s, following the implementation's initial decay schedule. **Setting-specific hyperparameters in Table 3 were selected on a validation set of 5% of the training data; test conditions were used only for final evaluation.**

The standardized gene-expression witness has unit weight in all settings. Support uses pooled calibration with $\alpha_{\text{sup}} = 0.05$. Fixed reward evaluation uses $\alpha_{\text{path}} = 0.05$ for all settings, including holdout, whose training threshold is 0.20. The witness excludes the query's entire sampling group; additional mean and detection energy terms have zero weight.

D Single-Cell Reward Calibration

We separate scorer fitting from calibration following the sample-splitting principle of conformal outlier detection (Angelopoulos & Bates, 2023). For each scorer, we shuffle real cells within each condition using seed 42 and reserve the final half for calibration. The first $\lfloor n_c/2 \rfloor$ cells form the reference subset, where n_c is the condition's real-cell count. The two scorers construct their splits

separately. Post-training scorers use training data only; calibration cells are held out from scorer fitting, but remain available as real targets for the population witness. Once fitted and calibrated, both scorers remain fixed during generator post-training.

For expression support, each reference subset determines its condition mean. Gene scales are sample standard deviations across pooled reference cells, using divisor $n_{\mathrm{ref}} - 1$. Zero-variance genes are excluded, and calibration scores are pooled across conditions with one entry per cell. Specifically, we evaluate equation 11 for each calibration cell under its own condition. Let $b_1, \ldots, b_{n_{\mathrm{sup}}}$ denote these scores, where n_{sup} is the total calibration-cell count across conditions. The upper-tail rank used in equation 12 is

$$p_{\mathrm{sup}}(\boldsymbol{x}, c) = \frac{1 + \sum_{j=1}^{n_{\mathrm{sup}}} \mathbf{1}\{b_j \geq a_{\mathrm{sup}}(\boldsymbol{x}, c)\}}{n_{\mathrm{sup}} + 1}. \tag{58}$$

A small p_{sup} identifies cells outside the calibrated radial envelope. Pooling provides a shared reference across conditions.

For pathway discrimination, we subsample the target-reference subset when necessary to match the available number of controls. Each classifier uses equal numbers of target and control cells. For equation 13, the pathway means $m_{c,p}$ and sample standard deviations $s_{c,p}$ are fitted on these combined reference cells; zero-variance pathways are excluded. Thus, both standardization and pathway selection depend on condition c. We use logistic regression with an ℓ_1 penalty, inverse regularization strength $C_{\mathrm{logit}} = 1$, the `liblinear` solver, seed 42, and 1000 maximum iterations. Only held-out target cells calibrate the resulting logits. Let $n_{\mathrm{path},c}$ be their number under condition c, and let $t_{c,j} = f_c(\boldsymbol{x}^{\mathrm{cal}}_{1,c,j})$ denote the logit of calibration cell $\boldsymbol{x}^{\mathrm{cal}}_{1,c,j}$. The lower-tail rank used in equation 14 is

$$p_{\mathrm{path}}(\boldsymbol{x}, c) = \frac{1 + \sum_{j=1}^{n_{\mathrm{path},c}} \mathbf{1}\{t_{c,j} \leq f_c(\boldsymbol{x})\}}{n_{\mathrm{path},c} + 1}. \tag{59}$$

A small p_{path} indicates classifier evidence in the lower tail of real target responses under the same condition.

Both tail ranks include ties and the finite-sample $+1$ correction. The thresholds $\alpha_{\mathrm{sup}}, \alpha_{\mathrm{path}}$ are specified in Appendix C. These ranks define empirical reward thresholds. Conformal false-positive guarantees require exchangeability with calibration scores.

E Evaluation Metrics

Distribution-level metrics. We compare predicted and real cell populations using three metrics. Energy distance (Székely & Rizzo, 2013) measures distributional differences using Euclidean distances in expression space. We use the empirical form of equation 7, including diagonal pairs in the within-population averages. Maximum mean discrepancy (MMD) (Gretton et al., 2012) provides a complementary kernel-based comparison. We report its unbiased squared estimator with an RBF kernel, setting σ^2 to half the median pairwise squared distance among real target cells for each perturbation. Sliced Wasserstein-1 distance (SW1) (Bonneel et al., 2015) averages exact one-dimensional Wasserstein-1 distances over 1,024 random unit directions. We project all 1,000 expression features without additional transformation. Directions are normalized isotropic Gaussian vectors, generated in four blocks of 256 using NumPy seeds 0–3. All methods and folds share these directions. Each condition uses 128 predicted cells and all real target cells. Exact quantile integration accommodates unequal population sizes. We average conditions equally, then average Norman folds equally. Lower values are better for all three distributional metrics.

Pseudobulk-mean metrics. We average expression within each perturbation and report MAE, two Pearson correlations, DE-Spearman, and DS. Mean absolute error (MAE) measures differences between predicted and real means. Pearson $\rho(\Delta)$ correlates these means after subtracting the control mean. Following Systema (Viñas Torné et al., 2025), $\rho(\hat{\Delta})$ instead subtracts the equally weighted mean of training perturbation centroids, emphasizing perturbation-specific effects beyond shared responses. We use Cell-Eval (Adduri et al., 2025) for DE-Spearman and discrimination score (DS). DE-Spearman correlates predicted and real log fold changes on genes significantly differentially expressed in real data at a false discovery rate of 0.05. DS ranks the correct perturbation using L_1 distances between absolute control-relative pseudobulk effects and reports $1 - r/N$, where r is its

zero-based rank among N evaluated perturbations. Lower MAE is better; higher correlations and DS are better.

Pathway enrichment concordance. Following PertAdapt (Bai et al., 2026), we compare GSEA normalized enrichment scores (NES) across Reactome pathways (Gillespie et al., 2022) using Spearman correlation. For each condition, we subtract the shared real control mean from predicted and real pseudobulk means of log-normalized expression. We rank all 1,000 measured genes by these signed effects. Pathways must contain at least five measured genes; the eligible pathways and gene background are shared across methods. We run weighted preranked GSEA with GSEApy 1.3.1, weight 1, 1,000 gene-set permutations, and random seed 42. For each condition, NES-Sp. correlates predicted and real NES vectors over all eligible pathways, without filtering by enrichment significance. We average conditions equally within each setting and fold, then average the four Norman folds equally. Higher values indicate better agreement in pathway enrichment rankings.

F Dataset Details

We use the Norman and ComboSciPlex splits from scDFM. For each split, training conditions are all benchmark conditions not listed as test conditions. We further hold out 5% of these training data as validation set.

Norman additive split. Each additive fold holds out 37 double-gene perturbations. All corresponding single-gene perturbations remain in training. Each fold therefore has 189 train conditions. Table 4 lists test conditions.

Table 4: Norman additive held-out double-gene conditions.

Fold	Test conditions
0	AHR+FEV, BPGM+SAMD1, CBL+UBASH3A, CBL+UBASH3B, CDKN1B+CDKN1A, CEBPB+CEBPA, CEBPB+PTPN12, CEBPE+CEBPA, CEBPE+KLF1, CEBPE+RUNX1T1, CNN1+MAPK1, CNN1+UBASH3A, DUSP9+MAPK1, ETS2+IGDCC3, ETS2+PRTG, FEV+ISL2, FOSB+CEBPB, FOSB+CEBPE, FOSB+OSR2, FOXA3+FOXA1, FOXA3+HOXB9, IRF1+SET, KIF18B+KIF2C, KLF1+MAP2K6, LYL1+IER5L, MAP2K3+IKZF3, MAP2K3+MAP2K6, PTPN12+OSR2, PTPN12+SNAI1, PTPN12+ZBTB25, SAMD1+PTPN12, SET+KLF1, TGFBR2+ETS2, UBASH3B+CNN1, UBASH3B+PTPN12, UBASH3B+UBASH3A, ZC3HAV1+HOXC13
1	AHR+FEV, AHR+KLF1, BCL2L11+BAK1, BPGM+ZBTB1, CBL+PTPN9, CBL+TGFBR2, CBL+UBASH3A, CBL+UBASH3B, CDKN1B+CDKN1A, CDKN1C+CDKN1A, CEBPB+CEBPA, CEBPE+CEBPB, CEBPE+KLF1, DUSP9+SNAI1, ETS2+IKZF3, ETS2+MAP7D1, FOSB+CEBPE, FOXA3+FOXF1, FOXA3+FOXL2, FOXA3+HOXB9, IGDCC3+PRTG, KIF18B+KIF2C, KLF1+CEBPA, KLF1+CLDN6, MAP2K3+IKZF3, MAP2K3+MAP2K6, MAP2K6+SPI1, MAPK1+PRTG, PTPN12+PTPN9, SAMD1+UBASH3B, SET+CEBPE, SGK1+S1PR2, SGK1+TBX2, TGFBR2+IGDCC3, UBASH3B+CNN1, UBASH3B+OSR2, ZC3HAV1+CEBPE
2	BPGM+ZBTB1, CBL+CNN1, CBL+PTPN12, CBL+TGFBR2, CBL+UBASH3B, CEBPE+CEBPA, CEBPE+RUNX1T1, CNN1+MAPK1, CNN1+UBASH3A, DUSP9+KLF1, DUSP9+SNAI1, ETS2+MAP7D1, ETS2+MAPK1, FEV+CBFA2T3, FOSB+IKZF3, FOSB+OSR2, FOSB+PTPN12, FOXA1+HOXB9, FOXL2+MEIS1, IGDCC3+PRTG, JUN+CEBPA, KLF1+CEBPA, LYL1+IER5L, MAP2K3+IKZF3, MAP2K3+MAP2K6, MAP2K6+IKZF3, MAP2K6+SPI1, MAPK1+IKZF3, PTPN12+PTPN9, SAMD1+UBASH3B, TGFBR2+ETS2, UBASH3B+CNN1, UBASH3B+PTPN9, UBASH3B+UBASH3A, UBASH3B+ZBTB25, ZC3HAV1+CEBPE, ZNF318+FOXL2

Fold	Test conditions
3	AHR+FEV, AHR+KLF1, CBL+UBASH3A, CDKN1C+CDKN1A, CEBPE+CNN1, CEBPE+SPI1, CNN1+MAPK1, DUSP9+ETS2, DUSP9+SNAI1, ETS2+MAP7D1, ETS2+PRTG, FEV+CBFA2T3, FEV+MAP7D1, FOSB+CEBPB, FOSB+CEBPE, FOSB+OSR2, FOSB+PTPN12, FOXA1+FOXL2, IGDCC3+PRTG, IRF1+SET, KIF18B+KIF2C, KLF1+CLDN6, LYL1+IER5L, MAPK1+IKZF3, MAPK1+PRTG, PTPN12+SNAI1, PTPN12+UBASH3A, PTPN12+ZBTB25, SAMD1+UBASH3B, SAMD1+ZBTB1, SET+CEBPE, SGK1+S1PR2, SGK1+TBX3, TGFBR2+ETS2, UBASH3B+ZBTB25, ZBTB10+DLX2, ZBTB10+SNAI1

Norman holdout split. Each holdout fold contains 15 held-out double-gene conditions. Their constituent single-gene perturbations are also held out. The train condition counts are 188, 185, 190, and 188. Table 5 lists test conditions.

Table 5: Norman holdout held-out conditions.

Fold	Single-gene test conditions	Double-gene test conditions
0	BPGM, CEBPA, CEBPB, CEBPE, CNN1, ETS2, FEV, FOSB, FOXA1, FOXA3, HOXB9, HOXC13, IER5L, IGDCC3, ISL2, LYL1, PTPN12, SAMD1, SNAI1, UBASH3A, UBASH3B, ZBTB25, ZC3HAV1	BPGM+SAMD1, CEBPB+PTPN12, CEBPE+CEBPA, CNN1+UBASH3A, ETS2+IGDCC3, FEV+ISL2, FOSB+CEBPB, FOXA3+FOXA1, FOXA3+HOXB9, LYL1+IER5L, PTPN12+SNAI1, PTPN12+ZBTB25, SAMD1+PTPN12, UBASH3B+UBASH3A, ZC3HAV1+HOXC13
1	BAK1, BCL2L11, CBL, CEBPA, CEBPB, CEBPE, CLDN6, ETS2, FOSB, IGDCC3, IKZF3, KIF18B, KIF2C, KLF1, MAP2K3, MAP2K6, MAP7D1, PRTG, S1PR2, SAMD1, SGK1, SPI1, TBX2, TGFBR2, UBASH3A, UBASH3B	BCL2L11+BAK1, CBL+UBASH3A, CEBPB+CEBPA, ETS2+MAP7D1, FOSB+CEBPE, IGDCC3+PRTG, KIF18B+KIF2C, KLF1+CEBPA, KLF1+CLDN6, MAP2K3+IKZF3, MAP2K6+SPI1, SAMD1+UBASH3B, SGK1+S1PR2, SGK1+TBX2, TGFBR2+IGDCC3
2	CBL, CEBPA, CEBPE, CNN1, DUSP9, FOSB, FOXA1, HOXB9, IER5L, JUN, KLF1, LYL1, MAP2K3, MAP2K6, MAPK1, PTPN12, RUNX1T1, SNAI1, SPI1, TGFBR2, UBASH3B	CBL+CNN1, CBL+PTPN12, CBL+TGFBR2, CEBPE+CEBPA, CEBPE+RUNX1T1, CNN1+MAPK1, DUSP9+SNAI1, FOSB+PTPN12, FOXA1+HOXB9, JUN+CEBPA, KLF1+CEBPA, LYL1+IER5L, MAP2K3+MAP2K6, MAP2K6+SPI1, UBASH3B+CNN1
3	AHR, CEBPE, CLDN6, CNN1, DLX2, ETS2, FEV, IER5L, IGDCC3, IKZF3, KLF1, LYL1, MAP7D1, MAPK1, PRTG, PTPN12, S1PR2, SGK1, SNAI1, SPI1, TBX3, UBASH3A, ZBTB10	AHR+FEV, CEBPE+CNN1, CEBPE+SPI1, CNN1+MAPK1, ETS2+PRTG, FEV+MAP7D1, IGDCC3+PRTG, KLF1+CLDN6, LYL1+IER5L, MAPK1+IKZF3, PTPN12+UBASH3A, SGK1+S1PR2, SGK1+TBX3, ZBTB10+DLX2, ZBTB10+SNAI1

ComboSciPlex split. ComboSciPlex uses the default scDFM split. The seven held-out conditions are listed in Table 6. All other ComboSciPlex conditions are used for training. This gives 25 train conditions.

Table 6: ComboSciPlex held-out test conditions.

Condition 1	Condition 2
Panobinostat	Crizotinib
Panobinostat	Curcumin
Panobinostat	SRT1720
Panobinostat	Sorafenib
SRT2104	Alvespimycin
control	Alvespimycin
control	Dacinostat

G Additional Perturbation Visualizations

Figure 6 extends the qualitative comparison in Figure 5 to three additional perturbations, including double-gene and drug-combination conditions.

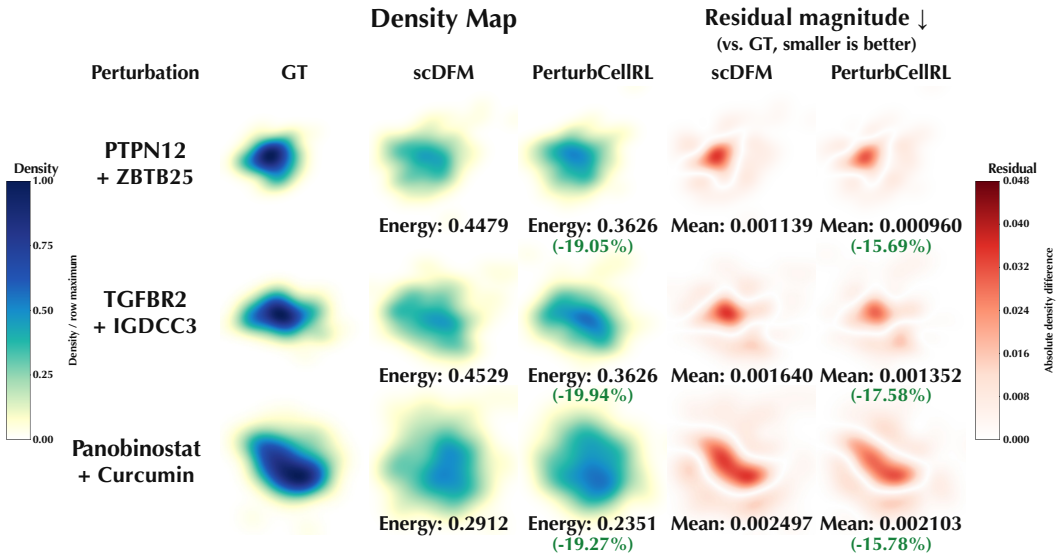


Figure 6: **Cell-density comparisons for additional perturbations.** Rows show PTPN12 + ZBTB25, TGFBR2 + IGDCC3, and Panobinostat + Curcumin. The first three columns compare ground truth (GT), scDFM, and PerturbCellRL. Densities are normalized by the maximum within each row. The last two columns show each model's absolute density difference from GT, on a shared residual scale; smaller values are better. Annotations report Energy distance and its relative reduction.

H Responsible Use

PerturbCellRL is intended as a decision-support method for prioritizing biological hypotheses, not as a replacement for experimental validation. Predictions can be wrong when perturbations are outside the training distribution, when cell states are poorly represented, or when reward models encode incomplete biological knowledge. Any candidate therapeutic or biological conclusion suggested by the model should be validated with independent experiments.