DEMIXPERT: DECOMPOSED RESPONSE MODELING WITH GAUSSIAN MIXTURES FOR OOD SINGLE-CELL PERTURBATION PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Accurate prediction of transcriptome-wide responses to unseen genetic perturbations requires modeling both perturbation-specific transcriptional shifts and heterogeneous cellular responses. Existing methods often entangle deterministic response structure with stochastic population-level variation, causing dominant shared patterns to mask weaker perturbation-specific signals and impair distributional modeling. To address these challenges, we propose **DeMixPert**, an approach for **De**composed response Modeling with Gaussian **Mix**tures for Out-Of-Distribution (OOD) single-cell **Pert**urbation prediction. DeMixPert decomposes perturbation-induced changes into a basal-state-dependent systematic response, a perturbation-specific response, and population-level variation. The systematic component is derived from the basal state encoded from control-cell expression, whereas the perturbation-specific component is inferred from pretrained target embeddings for unseen-target generalization. DeMixPert models population-level variation using a Gaussian prototype Invertible Network and adaptively combines reusable Gaussian prototypes according to the basal state and perturbation condition. The resulting mixture is mapped to a condition-specific variation distribution. Sampled variations are integrated with the systematic and perturbation-specific components, followed by joint decoding with the basal state to reconstruct perturbed-cell gene expression. Experimental results show that DeMixPert effectively captures heterogeneous single-cell perturbation responses and achieves superior performance across unseen-perturbation settings. The source code will be made publicly available upon publication.

1 INTRODUCTION

Single-cell genetic perturbation profiling couples controlled genetic interventions with transcriptome-wide readouts, enabling cell-resolved characterization of transcriptional responses and the regulatory programs underlying them (Adamson et al., 2016; Dixit et al., 2016). Despite its value, the scalability of such profiling is limited by the combinatorial growth of experimental conditions across perturbation targets, cellular states, and biological contexts (Cheng et al., 2026), making exhaustive measurement infeasible. Accordingly, *in silico* perturbation-response prediction offers a scalable means of estimating cellular responses under experimentally profiled conditions.

However, predicting responses under experimentally unobserved conditions involves more than estimating an average perturbation-induced expression shift (Yu et al., 2025). These challenges can be understood through three intertwined components of perturbation responses, as illustrated in Fig. 1. First, the basal cellular state contributes a systematic component that shapes responses across perturbation conditions (Dong et al., 2023; Song et al., 2025). The shared basal-state structure can be learned from observed perturbation conditions and transferred to unseen perturbation settings. Second, each perturbation condition induces a perturbation-specific transcriptional effect that distinguishes individual responses across interventions (Norman et al., 2019; Replogle et al., 2020). Unseen perturbation prediction requires inferring perturbation-specific effects from biological relationships between unseen targets and training targets. Third, cells exposed to the same perturbation may occupy distinct response subpopulations, whose internal structure and relative abundance vary with cellular state and environmental context (Adamson et al., 2016; Frangieh et al., 2021).

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

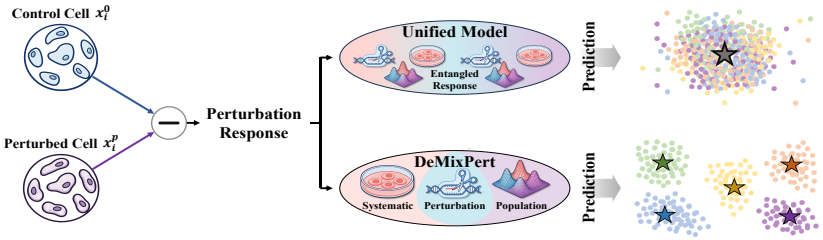


Figure 1: Comparison of unified and decomposed response modeling. By separating entangled response components, DeMixPert better preserves perturbation-specific signals and condition-dependent population structure.

Consequently, Out-Of-Distribution (OOD) prediction imposes three distinct requirements involving systematic-structure transfer, unseen-target effect inference, and condition-specific population-distribution estimation without direct observations.

Existing methods improve perturbation-response prediction from different perspectives. Latent-variable approaches represent perturbation responses through state transformations or factorized perturbation components (Lotfollahi et al., 2023). In parallel, Knowledge-guided methods exploit gene representations or biological priors to generalize from limited perturbation observations (Cui et al., 2024). From another perspective, population-distribution approaches learn transitions from control to perturbed populations using optimal transport, set-level distributional objectives, or flow matching (Bunne et al., 2023; Adduri et al., 2025; Yu et al., 2026). Despite substantial advances in response transfer and distributional modeling, most approaches still formulate the full perturbation response as a monolithic prediction target, leaving shared systematic response, perturbation-specific response, and condition-dependent population-level variation entangled. This limitation is consequential because Systema shows that shared systematic variation can dominate standard evaluation metrics (Viñas Torné et al., 2025). Accordingly, accurate OOD prediction requires disentangling basal-state-dependent systematic responses, perturbation-specific responses, and population-level variation.

In this paper, we propose **DeMixPert**, a **De**composed response framework with Gaussian **Mix**tures for out-of-distribution single-cell **Pert**urbation prediction. Our main contributions are summarized as follows:

- We propose DeMixPert, a decomposed response model for OOD single-cell perturbation prediction. DeMixPert disentangles deterministic response structure from stochastic cellular heterogeneity and captures condition-specific response distributions using adaptive Gaussian mixtures.
- We introduce a Gaussian prototype Invertible Network for population-level variation modeling. The invertible network uses context-adaptive weights to combine reusable Gaussian mixture prototypes, estimating condition-specific population-variation distributions.
- Quantitative analyses demonstrate that DeMixPert improves OOD perturbation prediction by accurately recovering perturbation-specific responses while preserving population-level distributional fidelity.

2 Related Work

2.1 Genetic Perturbation Prediction

Existing approaches formulate cellular responses as latent transformations: scGen applies an average shift, whereas CPA factorizes treatment, dose, and covariate contributions (Lotfollahi et al., 2019; 2023). For unseen targets, GEARS leverages gene-relation graphs, while GenePert draws on external gene embeddings (Roohani et al., 2024; Chen & Zou, 2024). To capture heterogeneity in unpaired data, STATE predicts post-intervention cell sets, whereas scDFM learns full conditional distributions via flow matching (Adduri et al., 2025; Yu et al., 2026). Systema further reveals that shared

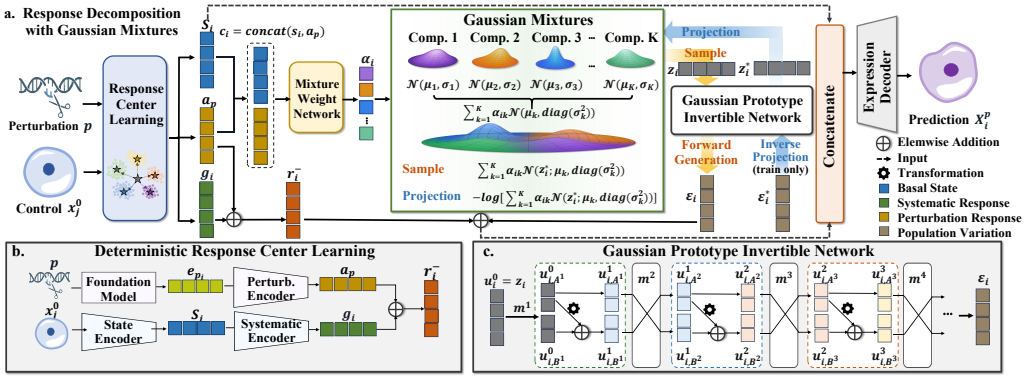


Figure 2: Overview of DeMixPert. **(a)** The overall workflow of DeMixPert. **(b)** The architecture of the Deterministic Response Center Learning module. **(c)** The architecture of the Gaussian prototype Invertible Network.

systematic variation can dominate standard metrics and mask target-specific signals (Viñas Torné et al., 2025). Nevertheless, most prior work does not jointly disentangle basal-state-dependent systematic responses, perturbation-specific responses, and condition-dependent population-level variation.

2.2 Gaussian Mixtures and Distribution Modeling

Gaussian mixture (GM) models provide a flexible representation of complex population structure (McLachlan & Peel, 2000), while normalizing flows transform tractable source distributions through invertible mappings (Dinh et al., 2014). PerturbNet combines perturbation embeddings with a conditional invertible network to generate cell-state distributions (Yu et al., 2025). More recently, MixFlow replaces the conventional unimodal Gaussian source with a descriptor-conditioned Gaussian mixture and transports it using conditional flow matching (Rubbi et al., 2026). Despite improved distributional flexibility, existing approaches still model perturbation responses monolithically, conflating systematic and perturbation-specific effects with residual variation. This entanglement limits component-wise generalization and mechanistic interpretation.

3 Methodology

3.1 Preliminaries

In the context of OOD single-cell perturbation prediction, we decompose responses into systematic, perturbation-specific, and population-level components to capture transferable patterns and condition-specific heterogeneity.

Let $\mathcal{D} = \left\{X^0, \{X^p\}_{p\in\mathcal{P}}\right\}$ denote a single-cell genetic perturbation dataset, where $\mathcal{P}$ is the set of perturbation conditions, each involving one or more target genes. The control population is defined by $X^0 = \{x_j^0 \in \mathbb{R}^G\}_{j=1}^{N_0}$. Here, N_0 is the number of control cells and G denotes the number of selected genes. For each perturbation condition $p \in \mathcal{P}$, the corresponding perturbed population is defined by $X^p = \{x_i^p\}_{i=1}^{N_p}$, where N_p is the number of cells observed under perturbation p.

Since the control and perturbed populations are unpaired, for each perturbed cell x_i^p, we randomly sample a control cell x_j^0 from X^0 as its reference and define the perturbation response as $\Delta x_i^p = x_i^p - x_j^0$. To facilitate subsequent response modeling, the expression-space response Δx_i^p is projected into a latent space using an encoder E_r: $r_i^* = E_r(\Delta x_i^p) \in \mathbb{R}^{d_r}$. The objective of DeMixPert is to predict a perturbation response $\hat{r}_i$ that approximates the target response r_i^*. In DeMixPert, $\hat{r}_i$ is modeled as three components: $\hat{r}_i = g_i + a_p + \epsilon_i$. Here, g_i, a_p, and ϵ_i denote the systematic response, perturbation-specific response, and population-level response variation, respectively.

3.2 OVERVIEW

Fig. 2 summarizes DeMixPert. Control-state and target encoders define a deterministic response center. A context-weighted Gaussian mixture and invertible network generate residual variation; the complete response is decoded jointly with the control state. During training, observed residuals are inverted to evaluate their conditional source likelihood.

3.3 DETERMINISTIC RESPONSE CENTER LEARNING

To prevent systematic variation from obscuring perturbation-specific effects, DeMixPert constructs a deterministic response center by combining separately learned systematic and perturbation-specific responses. The three encoders E_s, E_{sys}, and E_{pert} produce the basal-state representation, systematic response, and perturbation-specific response, respectively: $s_i = E_s(x_j^0) \in \mathbb{R}^{d_s}$, $g_i = E_{\text{sys}}(s_i) \in \mathbb{R}^{d_r}$, and $a_p = E_{\text{pert}}(e_p) \in \mathbb{R}^{d_r}$. Here, s_i summarizes the unperturbed cellular context and is retained for subsequent variation modeling. The systematic response g_i captures basal-state-dependent patterns shared across perturbation conditions, whereas a_p captures the perturbation-specific response derived from the perturbation embedding e_p. We derive e_p from frozen pretrained scGPT gene representations (Cui et al., 2024). Let $\mathcal{M}_p$ denote the set of target genes in perturbation condition p, and let $\mathbf{e}_m \in \mathbb{R}^{512}$ denote the embedding of gene m. The condition embedding is $e_p = \frac{1}{|\mathcal{M}_p|} \sum_{m \in \mathcal{M}_p} \mathbf{e}_m$. For a single-gene perturbation, this reduces to the embedding of that gene; for a combinatorial perturbation, the target-gene embeddings are averaged with equal weight. Further details are provided in Appendix A.

Finally, the resulting systematic and perturbation-specific responses are combined to form the deterministic response center: $r_i^- = g_i + a_p \in \mathbb{R}^{d_r}$. The r_i^- denotes the deterministic response pattern jointly specified by the basal cellular state s_i and perturbation target e_p in the response latent space $\mathbb{R}^{d_r}$.

3.4 GAUSSIAN PROTOTYPE INVERTIBLE NETWORK

The deterministic response center captures systematic and perturbation-specific responses but not the conditional population-level variation distribution. Accordingly, DeMixPert models the remaining population-level variation using a Gaussian prototype Invertible Network. Using the target response r_i^*, the target variation from the deterministic response center is given by $\epsilon_i^* = r_i^* - r_i^- \in \mathbb{R}^{d_r}$.

For each perturbation condition, the target variations of its observed cells form samples from the corresponding population-variation distribution. To capture potential variation patterns shared across perturbations, the target variations ϵ^* pooled from all training conditions are modeled using a K-component diagonal Gaussian mixtures:

$$q_{\text{GM}}(\epsilon) = \sum_{k=1}^{K} \pi_k \mathcal{N}\left(\epsilon; \mu_k, \text{diag}(\sigma_k^2)\right), \tag{1}$$

where π_k, μ_k, and $\text{diag}(\sigma_k^2)$ denote the weight, mean, and diagonal variance of the k-th Gaussian prototype, respectively. These parameters are periodically re-estimated from the target variations using the expectation-maximization algorithm (Dempster et al., 1977). Further details are provided in Supplementary Appendix F.

The global Gaussian Mixtures provide reusable Gaussian prototypes shared across perturbations. However, the contribution of each prototype depends jointly on the basal state and perturbation condition. Accordingly, s_i and a_p are concatenated to predicts Gaussian prototype weights:

$$\alpha_i = \text{softmax}\left(\text{MLP}(\text{concat}(s_i, a_p))\right) \in \mathbb{R}^K, \tag{2}$$

where $\text{concat}(\cdot,\cdot)$ denotes concatenation, $\text{MLP}(\cdot)$ denotes a multilayer perceptron, and $\text{softmax}(\cdot)$ normalizes the predicted prototype weights. The conditional Gaussian mixture is then defined as:

$$P(z \mid \alpha_i) = \sum_{k=1}^{K} \alpha_{ik} \mathcal{N}\left(z; \mu_k, \text{diag}(\sigma_k^2)\right). \tag{3}$$

The prototype parameters are shared globally, while their contributions are adapted to the condition representation c_i. Then, a source variable $z_i \sim P(z \mid \alpha_i)$ is sampled from the conditional mixture and passed through the invertible network T_θ, producing a sample ϵ_i from the final population-variation distribution: $\epsilon_i = T_\theta(z_i; c_i)$.

Let $u_i^{(0)} = z_i$ and $u_i^l = F_l\left(u_i^{(l-1)}; c_i\right) \in \mathbb{R}^{d_r}$ denote the output of the l-th coupling layer. A binary mask $m_l \in \{0,1\}^{d_r}$ partitions the feature dimensions into two complementary index sets, A^l and B^l, corresponding to mask values of 1 and 0, respectively, such that $A^\ell \cap B^\ell = \emptyset$, $A^\ell \cup B^\ell = \{1, \ldots, d_r\}$. Accordingly, the two input subvectors are defined as $u_{i,A^l}^{(l-1)} := \left(u_{ij}^{(l-1)}\right)_{\in A^l} \in \mathbb{R}^{|A^l|}$, $u_{i,B^l}^{(l-1)} := \left(u_{ij}^{(l-1)}\right)_{j \in B^l} \in \mathbb{R}^{|B^l|}$. The $u_i^{(l-1)}$ can be recovered by restoring the two subvectors to the positions specified by m_l: $u_i^{(l-1)} = \mathrm{merge}_{m_l}\left(u_{i,A^l}^{(l-1)}, u_{i,B^l}^{(l-1)}\right)$. Here, merge_{m^l} restores the two subsets to their original feature positions specified by m^l. The mask is fixed within each layer but varies across layers, allowing all feature dimensions to be updated.

Within the l-th coupling layer, $u_{i,A^l}^{(l-1)}$ remain unchanged and is used to determine the transformation applied to $u_{i,B^l}^{(l-1)}$. DeMixPert first measures how closely the unchanged features match each Gaussian prototype. The normalized matching weight associated with prototype k is

$$\gamma_{ik}^l = \frac{\pi_k \mathcal{N}\left(u_{i,A^l}^{(l-1)}; \mu_{k,A^l}, \mathrm{diag}\left(\sigma_{k,A^l}^2\right)\right)}{\sum_{q=1}^{K} \pi_q \mathcal{N}\left(u_{i,A^l}^{(l-1)}; \mu_{q,A^l}, \mathrm{diag}\left(\sigma_{q,A^l}^2\right)\right)}. \tag{4}$$

In parallel, a layer-specific MLP h^l produces prototype modulation coefficients from c_i: $b_i^l = h^l(c_i) \in \mathbb{R}^K$. Here, γ_i^l captures feature-level prototype matching, whereas b_i^l captures the contribution of each prototype under the current basal-state and perturbation context. The two coefficients are combined to generate the additive shift applied to B^l:

$$f^l(u_{i,A^l}, c_i) = W^l\left(\gamma_i^l \odot b_i^l\right) \in \mathbb{R}^{|B^l|}, \tag{5}$$

where $W^l \in \mathbb{R}^{|B^l| \times K}$ is a learnable projection matrix and $\odot$ denotes element-wise multiplication. The additive coupling transformation is defined as follows: $u_{i,A^l}^l = u_{i,A^l}^{l-1}$, $u_{i,B^l}^l = u_{i,B^l}^{l-1} + f_l\left(u_{i,A^l}^{l-1}, c_i\right)$. The two subsets are then reassembled to obtain the complete output of the l-th coupling layer:

$$F_l\left(u_i^{l-1}; c_i\right) = u_i^l = \mathrm{merge}_{m^l}\left(u_{i,A^l}^l, u_{i,B^l}^l\right). \tag{6}$$

Since the A^l subset remains unchanged, the same shift function can be evaluated during inversion (Dinh et al., 2016). The inverse transformation is defined by $u_{i,A^l}^{l-1} = u_{i,A^l}^l$ and $u_{i,B^l}^{l-1} = u_{i,B^l}^l - f_l\left(u_{i,A^l}^l, c_i\right)$.

The recovered subsets are reassembled to obtain the complete output of the inverse layer:

$$F_l^{-1}\left(u_i^l; c_i\right) = u_i^{l-1} = \mathrm{merge}_{m^l}\left(u_{i,A^l}^{l-1}, u_{i,B^l}^{l-1}\right). \tag{7}$$

Each coupling layer is bijective and has a triangular Jacobian with unit determinant (Dinh et al., 2014). This invertibility maps each target variation back to the Gaussian mixtures. Its likelihood under the source distribution supervises the context-dependent prototype weights and the invertible transformation. The corresponding likelihood objective is introduced in the training loss.

After L coupling layers, the generated variation sample is

$$\epsilon_i = T_\theta(z_i; c_i) = F_L \circ \cdots \circ F_1(z_i; c_i) \in \mathbb{R}^{d_r}. \tag{8}$$

For perturbation condition p, the collection $\{\epsilon_i : p_i = p\}$ forms samples from its predicted population-level variation distribution. Each sample perturbs the deterministic response center to yield the response latent: $\hat{r}_i = r_i^- + \epsilon_i$.

3.5 EXPRESSION DECODER

The post-perturbation expression is jointly determined by the original cellular context and the perturbation-induced response. The basal-state representation s_i captures the cellular context, whereas the predicted response latent $\hat{r}_i$ encodes the perturbation effect. The concatenated representation is then decoded to reconstruct post-perturbation gene expression: $\hat{x}_i^p = D_\psi\left(\mathrm{concat}(s_i, \hat{r}_i)\right)$. Therein, D_ψ denotes the expression decoder.

3.6 TRAINING OBJECTIVE

DeMixPert is trained end-to-end to jointly learn the deterministic response center and condition-specific population-level variation. The total training objective consists of four loss components that supervise response alignment, population-variation likelihood, gene-expression reconstruction, and population-distribution matching.

Response Alignment Loss. A mean squared error loss is used to align the predicted response latent with its target: $\mathcal{L}_{\mathrm{align}} = \frac{1}{N}\sum_{i=1}^{N} \|\hat{r}_i - r_i^*\|_2^2$, where N denotes the number of cells in the training batch.

Population-Variation Likelihood Loss. To learn the conditional population-level variation, each target variation is first mapped back to the Gaussian mixtures: $z_i^* = T_\theta^{-1}(\epsilon_i^*; c_i)$. We then minimize its negative log-likelihood under the conditional Gaussian mixtures:

$$\mathcal{L}_{\mathrm{gm}} = -\frac{1}{N}\sum_{i=1}^{N} \log\left[\sum_{k=1}^{K} \alpha_{ik}\mathcal{N}\left(z_i^*; \mu_k, \mathrm{diag}(\sigma_k^2)\right)\right], \tag{9}$$

where, the $\mathcal{L}_{\mathrm{gm}}$ jointly supervises the context-dependent prototype weights and the invertible transformation.

Gene-Expression Reconstruction Loss. The reconstruction loss preserves gene expression information by reconstructing the perturbed expression from both the predicted and target responses, while recovering the control expression from a zero-response vector $\mathbf{0} \in \mathbb{R}^{d_r}$:

$$\begin{aligned}\mathcal{L}_{\mathrm{rec}} = \mathcal{L}_{mse}(D_\psi([s_i, \hat{r}_i]), x_i^{p_i}) + \mathcal{L}_{mse}(D_\psi([s_i, r_i^*]), x_i^{p_i}) \\ + \mathcal{L}_{mse}\big(D_\psi([s_i, \mathbf{0}]), x_j^0\big)\,,\end{aligned} \tag{10}$$

where $\mathcal{L}_{mse}$ denotes the mean square error loss.

Distribution Loss. At the population level, we use energy distance to align the generated and observed expression distributions under each perturbation condition:

$$\mathcal{L}_{\mathrm{dist}} = \frac{1}{|\mathcal{P}|}\sum_{p\in\mathcal{P}} \mathrm{ED}\left(\{\hat{x}_i^p\}_{i:p_i=p}, \{x_i^p\}_{i:p_i=p}\right), \tag{11}$$

where $\mathcal{P}$ denotes the set of perturbation conditions and $\mathrm{ED}(\cdot)$ is the energy distance (Székely & Rizzo, 2013). The overall training objective is $\mathcal{L} = \mathcal{L}_{\mathrm{align}} + \lambda_{\mathrm{gm}}\mathcal{L}_{\mathrm{gm}} + \mathcal{L}_{\mathrm{rec}} + \mathcal{L}_{\mathrm{dist}}$. The λ_{gm} balances the numerical scale of the negative log-likelihood against the other loss components. All trainable parameters are jointly optimized by minimizing $\mathcal{L}$.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Datasets and preprocessing. We benchmark DeMixPert on four widely used single-cell genetic perturbation datasets. Adamson (Adamson et al., 2016) and Papalexi (Papalexi et al., 2021) contain single-gene perturbations, whereas Norman (Norman et al., 2019) and Replogle (Replogle et al., 2020) involve combinatorial perturbations. All datasets are processed following a standard single-cell RNA-seq preprocessing pipeline. For each dataset, we select 2,048 highly variable genes (HVGs) and additionally retain all perturbation-target genes to define the input gene space. Detailed dataset descriptions and statistics are provided in Appendix C.

Table 1: Comprehensive evaluation results on four genetic perturbation datasets, reported as mean ± standard deviation. ↑ (↓) indicates that higher (lower) values are better. The best result for each metric is marked in **bold**.

Dataset	Method	Top 100 DEGs Level				All Gene Level			
		C-DEGs↑	E-Dist↓	W-Dist↓	MSE↓	DES↑	MSE↓	Centroid Acc↑	PDS↑
Papalexi	GEARS	2.280 ± 0.460	0.617 ± 0.218	12.720 ± 0.263	0.065 ± 0.032	0.194 ± 0.115	0.024 ± 0.015	0.240 ± 0.089	0.640 ± 0.049
	scGPT	1.040 ± 0.862	0.208 ± 0.327	**3.468 ± 1.159**	0.056 ± 0.059	0.077 ± 0.045	0.010 ± 0.011	0.500 ± 0.000	0.623 ± 0.024
	GenePert	5.427 ± 0.889	0.337 ± 0.136	7.011 ± 0.190	0.027 ± 0.012	0.127 ± 0.067	0.004 ± 0.002	0.500 ± 0.000	0.600 ± 0.025
	scDFM	5.560 ± 1.506	1.971 ± 0.582	14.053 ± 0.324	0.274 ± 0.104	**0.212 ± 0.099**	0.099 ± 0.037	0.200 ± 0.141	0.568 ± 0.095
	STATE	4.760 ± 1.499	0.760 ± 0.214	13.364 ± 0.222	0.072 ± 0.033	0.198 ± 0.107	0.025 ± 0.014	0.160 ± 0.089	0.600 ± 0.000
	DeMixPert	**22.000 ± 4.854**	**0.171 ± 0.088**	6.369 ± 0.403	**0.008 ± 0.006**	0.200 ± 0.129	**0.002 ± 0.001**	**0.720 ± 0.110**	**0.904 ± 0.046**
Adamson	GEARS	2.280 ± 0.145	0.563 ± 0.125	9.654 ± 0.217	0.046 ± 0.015	0.313 ± 0.082	0.012 ± 0.003	0.133 ± 0.067	0.605 ± 0.026
	scGPT	4.151 ± 0.131	2.997 ± 0.310	**4.936 ± 0.326**	0.037 ± 0.014	**0.378 ± 0.078**	0.005 ± 0.002	**0.586 ± 0.079**	0.567 ± 0.037
	GenePert	0.063 ± 0.010	0.468 ± 0.174	4.938 ± 0.301	0.031 ± 0.013	0.105 ± 0.038	**0.002 ± 0.001**	0.533 ± 0.014	0.570 ± 0.025
	scDFM	3.400 ± 1.381	1.355 ± 0.216	10.771 ± 0.628	0.132 ± 0.027	0.278 ± 0.082	0.048 ± 0.007	0.067 ± 0.047	0.539 ± 0.040
	STATE	2.947 ± 0.357	0.820 ± 0.138	10.989 ± 0.169	0.052 ± 0.017	0.280 ± 0.091	0.015 ± 0.003	0.080 ± 0.030	0.534 ± 0.004
	DeMixPert	**10.733 ± 5.198**	**0.448 ± 0.052**	5.481 ± 0.113	**0.021 ± 0.003**	0.153 ± 0.078	0.006 ± 0.001	0.560 ± 0.076	**0.947 ± 0.020**
Norman	GEARS	3.850 ± 0.557	0.855 ± 0.220	6.380 ± 0.234	0.049 ± 0.019	0.334 ± 0.029	0.008 ± 0.002	0.237 ± 0.112	0.802 ± 0.090
	scGPT	6.644 ± 0.627	3.174 ± 0.157	**5.627 ± 0.239**	0.028 ± 0.007	**0.540 ± 0.031**	0.004 ± 0.001	**0.940 ± 0.013**	0.886 ± 0.025
	GenePert	19.181 ± 0.658	1.113 ± 0.154	6.305 ± 0.256	0.078 ± 0.013	0.404 ± 0.022	0.008 ± 0.011	0.639 ± 0.034	0.646 ± 0.033
	scDFM	15.350 ± 3.548	0.711 ± 0.148	7.607 ± 0.464	0.046 ± 0.011	0.389 ± 0.029	0.009 ± 0.003	0.463 ± 0.116	0.890 ± 0.037
	STATE	21.006 ± 5.872	1.341 ± 0.217	7.175 ± 0.177	0.105 ± 0.021	0.354 ± 0.013	0.011 ± 0.002	0.031 ± 0.000	0.524 ± 0.014
	DeMixPert	**31.650 ± 11.739**	**0.374 ± 0.056**	6.606 ± 0.269	**0.003 ± 0.0002**	0.506 ± 0.063	**0.003 ± 0.0002**	0.750 ± 0.163	**0.981 ± 0.005**
Replogle	GEARS	2.111 ± 0.176	0.448 ± 0.060	12.743 ± 0.114	0.038 ± 0.009	0.189 ± 0.047	0.012 ± 0.002	0.267 ± 0.127	0.669 ± 0.058
	scGPT	6.089 ± 0.755	2.263 ± 0.280	**4.302 ± 0.483**	0.011 ± 0.004	**0.269 ± 0.060**	0.002 ± 0.001	**0.778 ± 0.125**	0.654 ± 0.087
	GenePert	12.533 ± 2.881	0.200 ± 0.047	4.600 ± 0.550	**0.002 ± 0.001**	0.200 ± 0.047	0.002 ± 0.001	0.558 ± 0.048	0.580 ± 0.008
	scDFM	3.422 ± 1.630	0.802 ± 0.091	13.234 ± 0.272	0.092 ± 0.015	0.202 ± 0.053	0.026 ± 0.004	0.178 ± 0.061	0.588 ± 0.052
	STATE	**12.711 ± 3.278**	0.510 ± 0.075	13.008 ± 0.107	0.040 ± 0.012	0.207 ± 0.041	0.011 ± 0.002	0.111 ± 0.000	0.548 ± 0.014
	DeMixPert	12.378 ± 4.071	**0.155 ± 0.040**	5.614 ± 0.156	0.005 ± 0.001	0.227 ± 0.066	**0.002 ± 0.000**	0.778 ± 0.193	**0.938 ± 0.039**

Compared methods. We compare DeMixPert against five representative perturbation-response prediction methods: GEARS (Roohani et al., 2024), scGPT (Cui et al., 2024), GenePert (Chen & Zou, 2024), scDFM (Yu et al., 2026), and STATE (Adduri et al., 2025). These methods span graph learning, pretrained foundation models, ridge regression, conditional flow matching, and set-based Transformers. Accordingly, the comparative experiments enable DeMixPert to be evaluated against methods with substantially different assumptions and predictive mechanisms.

Metrics and implementation. Model performance is evaluated at both the top-100 differentially expressed genes (DEGs) level and the all-gene level. At the top-100 DEG level, we report Common Differentially Expressed Genes (C-DEGs), Energy Distance (E-Dist), Wasserstein Distance (W-Dist), and Mean Squared Error (MSE). At the all-gene level, we report the Differential Expression Score (DES), MSE, Centroid Accuracy (Centroid Acc), and the Perturbation Discrimination Score (PDS) (Wei et al., 2026; Viñas Torné et al., 2025; Roohani et al., 2025). Together, these metrics assess differential-expression recovery, distributional agreement, expression error, and perturbation discrimination. MSE is computed between the predicted and observed population centroids. Formal metric definitions are provided in Appendix J.2.

Condition-disjoint splits evaluate unseen perturbations. The combinatorial setting evaluates unseen combinations of targets observed individually during training (0/2 unseen), whereas the single-gene setting holds out target conditions. Single-gene datasets use 70%/10%/20% training/validation/test splits. For combinatorial datasets, training includes all single-gene conditions and half of the combinatorial conditions, while the remainder is divided equally between validation and test sets. Validation selects checkpoints, and results are reported as mean ± standard deviation across random seeds. DeMixPert uses PyTorch and AdamW. Default and dataset-specific configurations are provided in AppendiX B and H, respectively.

4.2 Quantitative Comparison

Table 1 shows that DeMixPert achieves the highest mean PDS and lowest mean E-Dist across all four datasets among the learned methods evaluated. It also leads C-DEGs on Papalexi, Adamson, and Norman, reaching 31.650 versus STATE's 21.006 on Norman and remaining competitive on Replogle at 12.378 versus 12.711.

Table 2: Results of the ablation study (mean ± sample standard deviation over five seeds).

Adamson					Replogle				
Variant	Top 100 DEGs		All Genes		Variant	Top 100 DEGs		All Genes	
	C-DEGs↑	W-Dist↓	Centroid Acc↑	PDS↑		C-DEGs↑	W-Dist↓	Centroid Acc↑	PDS↑
w/o Decomp	10.520 ± 6.923	5.927 ± 0.130	0.240 ± 0.101	0.733 ± 0.077	w/o Decomp	12.022 ± 3.480	5.905 ± 0.139	0.244 ± 0.093	0.736 ± 0.082
w/o Gene	8.773 ± 4.474	5.772 ± 0.155	0.267 ± 0.141	0.787 ± 0.039	w/o Gene	9.733 ± 3.068	5.852 ± 0.226	0.333 ± 0.157	0.760 ± 0.091
w/o Sys	10.400 ± 6.285	5.746 ± 0.164	0.360 ± 0.138	0.846 ± 0.043	w/o Sys	11.556 ± 5.719	5.832 ± 0.184	0.333 ± 0.079	0.800 ± 0.058
w/o GM	**11.107** ± 6.234	5.781 ± 0.121	0.280 ± 0.110	0.797 ± 0.054	w/o GM	**14.356** ± 6.217	5.833 ± 0.286	0.289 ± 0.099	0.773 ± 0.097
DeMixPert	10.733 ± 5.198	**5.481** ± 0.113	**0.560** ± 0.076	**0.947** ± 0.020	**DeMixPert**	12.378 ± 4.552	**5.614** ± 0.156	**0.778** ± 0.192	**0.938** ± 0.039

To assess gains beyond simple response approximations, additional comparisons use Training-mean, based on average training responses, and Additive, based on summed single-gene effects. DeMixPert outperforms Training-mean in mean performance on seven of eight metrics on Papalexi and all eight on Adamson. On both combinatorial datasets, it achieves higher C-DEGs and DES and lower MSE at both evaluation levels than Additive, with trade-offs on other metrics (Appendix N).

Together, these results demonstrate DeMixPert's effectiveness for predicting unseen single-gene and combinatorial perturbations, with gains in DEG overlap, reconstruction, and multiple distributional and discrimination metrics.

4.3 Ablation Study

Table 2 compares four independently retrained variants on Adamson and Replogle: unified response modeling (w/o Decomp), removal of the target-specific branch (w/o Gene), removal of the systematic branch (w/o Sys), and removal of both the Gaussian mixture and invertible network (w/o GM); details are in Appendix J.1. The full model has the lowest W-Dist and highest Centroid Accuracy/PDS among these variants. Removing the target branch reduces C-DEGs, whereas w/o GM increases C-DEGs but worsens the other reported metrics, revealing a trade-off between DEG overlap and distributional/discrimination performance.

4.4 Sensitivity Analysis

Fig. 3 shows performance changes under different Gaussian prototype numbers K and GM-loss weights λ_{gm} relative to the reference setting ($K = 8$, $\lambda_{\text{gm}} = 0.01$). Deviating from $K = 8$ generally degrades DEG recovery, distribution modeling, and expression reconstruction, whereas larger K improves perturbation identification, indicating a fidelity–separability trade-off. Similarly, increasing λ_{gm} improves identification but slightly affects other objectives. Overall, the reference setting achieves the best balance.

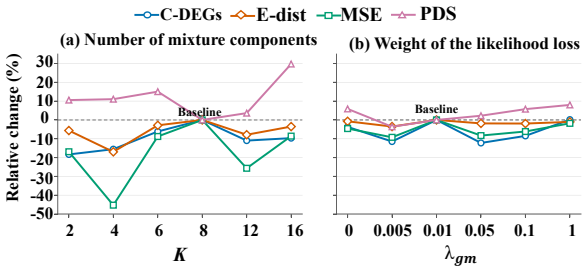


Figure 3: Sensitivity analysis on the Papalexi dataset relative to $K = 8$ and $\lambda_{\text{gm}} = 0.01$. Positive values indicate improvement and negative values indicate degradation after aligning metric directions. The score construction is described in Appendix M.

4.5 Interpretability Analysis

We further examine two aspects of DeMixPert: whether response decomposition separates basal-state-dependent variation from perturbation-associated structure, and whether the learned Gaussian prototype weights capture condition-specific population heterogeneity.

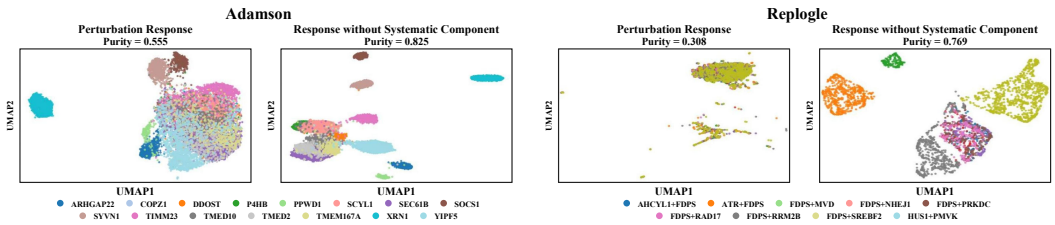


Figure 4: UMAP visualizations of the observed-response encoding r_i^* (left) and $r_i^* - g_i$ (right) on Adamson and Replogle. Colors indicate perturbation conditions. Purity is computed in the standardized response-latent space; the clustering and evaluation protocol is given in Appendix L.1.

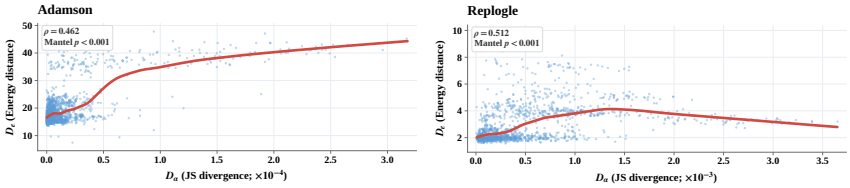


Figure 5: Association between condition-level prototype-weight divergence D_α and observed residual-population distance D_ϵ on training conditions from Adamson and Replogle. Red curves show LOWESS fits. Aggregation, sampling, and testing details are provided in Appendix L.2.

Fig. 4 compares the encoded perturbation response before and after removing the systematic component g_i. After removing g_i, the response representations form more compact and clearly separated perturbation-specific clusters. Cluster purity increases from 0.555 to 0.825 on Adamson and from 0.308 to 0.769 on Replogle, supporting the role of the decomposition in separating basal-state-associated variation from perturbation-specific response structure.

Fig. 5 relates condition-level prototype composition to observed-response residual differences. Here, D_α is the Jensen–Shannon divergence between mean prototype-weight profiles (Lin, 2002), and D_ϵ is the energy distance between observed-response residual populations. The mean profiles are computed over the same set of control states. Formal definitions and the Spearman and Mantel testing procedures (Spearman, 1961; Dietz, 1983) are provided in Appendix L.2.

Across all pairs of perturbation conditions, significant positive rank associations are observed on both Adamson ($\rho = 0.462$) and Replogle ($\rho = 0.512$), with Mantel $p < 0.001$ in both cases. These results indicate that differences in the learned prototype weights are associated with condition-specific population-distribution differences.

5 Conclusion

In this study, we present DeMixPert, a Gaussian-mixture-based response decomposition framework for OOD single-cell perturbation prediction. DeMixPert disentangles perturbation responses into systematic, perturbation-specific, and population-level components, each modeled through a dedicated mechanism. A Gaussian prototype Invertible Network further characterizes cellular heterogeneity through context-adaptive prototype composition. Extensive experiments demonstrate accurate recovery of perturbation-specific responses under OOD settings. The adaptive prototype composition also captures condition-specific population structures, offering a structured interpretation of heterogeneous cellular responses.

AI USE STATEMENT

During manuscript preparation, we used generative AI tools for language editing, LaTeX formatting assistance, and debugging of document-generation code. We did not use generative AI tools to generate experimental data, numerical results, or unsupported scientific claims. All AI-assisted edits were reviewed and verified by the authors, and the authors take responsibility for the final content of this paper.

REPRODUCIBILITY STATEMENT

The model architecture, data preprocessing, dataset splits, hyperparameters, and evaluation metrics are documented in the main text and appendices. Experiments use predefined random seeds, with checkpoint selection based exclusively on validation data. Anonymized source code is provided in the supplementary materials to facilitate reproduction of the reported results.

REFERENCES

Britt Adamson, Thomas M Norman, Marco Jost, Min Y Cho, James K Nuñez, Yuwen Chen, Jacqueline E Villalta, Luke A Gilbert, Max A Horlbeck, Marco Y Hein, et al. A multiplexed single-cell crispr screening platform enables systematic dissection of the unfolded protein response. *Cell*, 167(7):1867–1882, 2016.

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.

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.

Yiqun Chen and James Zou. Genepert: Leveraging genept embeddings for gene perturbation prediction. *bioRxiv*, pp. 2024–10, 2024.

Jiabei Cheng, Changxi Chi, Jingbo Zhou, Hongyi Xin, and Jun Xia. Prescribe: Predicting single-cell responses with bayesian estimation. *Advances in Neural Information Processing Systems*, 38: 145457–145489, 2026.

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.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society: series B (methodological)*, 39(1): 1–22, 1977.

E Jacquelin Dietz. Permutation tests for association between two distance matrices. *Systematic Biology*, 32(1):21–26, 1983.

Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

Atray Dixit, Oren Parnas, Biyu Li, Jenny Chen, Charles P Fulco, Livnat Jerby-Arnon, Nemanja D Marjanovic, Danielle Dionne, Tyler Burks, Raktima Raychowdhury, et al. 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, et al. Causal identification of single-cell experimental perturbation effects with cinema-ot. *Nature methods*, 20(11): 1769–1779, 2023.

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, et al. Multimodal pooled perturb-cite-seq screens in patient models define mechanisms of cancer immune evasion. *Nature genetics*, 53(3):332–341, 2021.

Jianhua Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 37(1):145–151, 2002.

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

Geoffrey J McLachlan and David Peel. *Finite mixture models*. John Wiley & Sons, 2000.

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.

Efthymia Papalexi, Eleni P Mimitou, Andrew W Butler, Samantha Foster, Bernadette Bracken, William M Mauck III, Hans-Hermann Wessels, Yuhan Hao, Bertrand Z Yeung, Peter Smibert, et al. Characterizing the molecular regulation of inhibitory immune checkpoints with multimodal single-cell screens. *Nature genetics*, 53(3):322–331, 2021.

Joseph M Replogle, Thomas M Norman, Albert Xu, Jeffrey A Hussmann, Jin Chen, J Zachery Cogan, Elliott J Meer, Jessica M Terry, Daniel P Riordan, Niranjan Srinivas, et al. Combinatorial single-cell crispr screens by direct guide rna capture and targeted sequencing. *Nature biotechnology*, 38(8):954–961, 2020.

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.

Andrea Rubbi, Amir Akbarnejad, Mohammad Vali Sanian, Aryan Yazdan Parast, Hesam Asadollahzadeh, Arian Amani, Naveed Akhtar, Sarah Cooper, Andrew Bassett, Pietro Liò, et al. Mixflow: Mixture-conditioned flow matching for out-of-distribution generalization. *arXiv preprint arXiv:2601.11827*, 2026.

Bicna Song, Dingyu Liu, Weiwei Dai, Natalie F Mcmyn, Qingyang Wang, Dapeng Yang, Adam Krejci, Anatoly Vasilyev, Nicole Untermoser, Anke Loregger, et al. Decoding heterogeneous single-cell perturbation responses. *Nature cell biology*, 27(3):493–504, 2025.

Charles Spearman. The proof and measurement of association between two things. 1961.

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.

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.

Zhiting Wei, Yiheng Wang, Yicheng Gao, Shuguang Wang, Ping Li, Duanmiao Si, Yuli Gao, Siqi Wu, Danlu Li, Kejing Dong, et al. Benchmarking algorithms for generalizable single-cell perturbation response prediction. *Nature Methods*, 23(2):451–464, 2026.

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.

Hengshi Yu, Weizhou Qian, Yuxuan Song, and Joshua D Welch. Perturbnet predicts single-cell responses to unseen chemical and genetic perturbations. *Molecular Systems Biology*, 21(8):960, 2025.

Table 3: Summary of the notation used in DeMixPert.

Symbol	Definition
Problem formulation	
$\mathcal{D}$	A single-cell genetic perturbation dataset.
X^0	The population of unperturbed control cells.
X^p	The observed cell population under perturbation condition p.
x_j^0, x_i^p	Gene-expression profiles of a control cell and a cell under perturbation p, respectively.
$\mathcal{P}$, p	The set of perturbation conditions and an individual perturbation condition, respectively.
N_0, N_p	The numbers of control cells and cells under perturbation p.
G	The number of selected genes in the expression space.
π_0	The distribution of unperturbed control cells.
$\pi(\cdot \mid p)$	The condition-specific distribution of cells under perturbation p.
f_θ	The stochastic conditional generator parameterized by θ.
Δx_i^p	The observed expression change defined in the main text, $\Delta x_i^p = x_i^p - x_j^0$, where x_j^0 is a randomly sampled control cell.
Response decomposition	
E_s, E_r	The basal-state encoder and response encoder, respectively.
$s_i \in \mathbb{R}^{d_s}$	The basal cellular-state representation of cell i.
$r_i^* \in \mathbb{R}^{d_r}$	The target response latent encoded from the observed expression change Δx_i^p.
d_s, d_r	The dimensions of the basal-state latent and response latent spaces.
g_i	The basal-state-dependent systematic response component.
$e_{p_i} \in \mathbb{R}^{d_e}$	The pretrained embedding of perturbation condition p_i.
d_e	The dimension of the pretrained perturbation embedding.
a_{p_i}	The perturbation-level average response effect derived from e_{p_i}.
$\bar{r}_i$	The deterministic response center, $\bar{r}_i = g_i + a_{p_i}$.
ϵ_i^*	The target cell-specific deviation, $\epsilon_i^* = r_i^* - \bar{r}_i$.
Gaussian-prototype residual modeling	
K	The number of diagonal Gaussian response prototypes.
π_k, μ_k, σ_k^2	The global mixture weight, mean, and diagonal variance of the k-th Gaussian prototype.
c_i	The joint condition representation, $c_i = \mathrm{concat}(s_i, a_{p_i})$.
α_{ik}	The condition-dependent mixture weight of prototype k for cell i.
z_i	A source variable sampled from the condition-dependent Gaussian mixture.
T_θ	The conditional invertible transformation that maps z_i to a cell-specific deviation.
L	The number of additive coupling layers in T_θ.
$\gamma_{ik}^{(\ell)}$	The prototype-matching responsibility of prototype k at coupling layer ℓ.
ϵ_i	The generated stochastic cell-specific deviation.
$\hat{r}_i$	The predicted response latent, $\hat{r}_i = \bar{r}_i + \epsilon_i$.
D_ψ	The expression decoder parameterized by ψ.
$\hat{x}_i^{p_i}$	The predicted post-perturbation expression profile.
Training objectives	
$\mathcal{L}_{\mathrm{align}}$	The response-latent alignment loss.
$\mathcal{L}_{\mathrm{gm}}$	The Gaussian-mixture negative log-likelihood loss.
$\mathcal{L}_{\mathrm{rec}}$	The expression reconstruction loss.
$\mathcal{L}_{\mathrm{dist}}$	The perturbation-condition-specific distribution matching loss.
$\mathcal{L}$	The overall training objective.

A Perturbation Embedding Construction

As stated in the main paper, the perturbation embedding is constructed from pretrained scGPT gene representations Cui et al. (2024). Let $\mathcal{M}_p$ denote the set of target genes named by perturbation condition p, and let $\mathbf{e}_m \in \mathbb{R}^{512}$ be the pretrained scGPT embedding of target gene m. We define

$$e_p = \frac{1}{|\mathcal{M}_p|} \sum_{m \in \mathcal{M}_p} \mathbf{e}_m \in \mathbb{R}^{512}. \tag{12}$$

For a single-gene perturbation, this reduces to the embedding of that gene. For a combinatorial perturbation, the target-gene embeddings are averaged with equal weight. Conditions whose named target is unavailable in the frozen embedding table are removed before split generation.

B Default Model Configuration

This section reports the default architecture and optimization configuration shared across all datasets. A summary of the model architecture is provided in Table 4, and dataset-specific training settings are reported in Supplementary Appendix H.

Table 4: Main-text-aligned DeMixPert component summary. Hyphens denote successive linear-layer widths; residual-block repetition is written explicitly.

Module	Layer dimensions	Activation	Dropout	Norm.
State encoder	G–1024–512–256	SiLU after hidden layers	0.05	LayerNorm after hidden linear layers
Response encoder	G–1024–512–128	SiLU after hidden layers	0.05	LayerNorm after hidden linear layers
Systematic predictor	256–512; 3×(512–1024–512); 512–128	SiLU in blocks	0.05	LayerNorm at each block and output head
Anchor predictor	512–512; 3×(512–1024–512); 512–128	SiLU in blocks	0.05	LayerNorm at each block and output head
Condition network	384–256–256; heads 8 and 32	SiLU	0	None
Coupling flow	$4\times$ additive half-coupling; $8 \to 64$ shift map	None	0	None
Expression decoder	$384 \to G$	MLP-based mapping	–	–

Hyperparameter	Value
State latent dimension	256
Response latent dimension	128
Perturbation embedding dimension	512
Gaussian components K	8
Coupling layers L	4
Condition-network hidden width	256
Prototype temperature	1.0
Optimizer	AdamW
Learning rate	5×10^{-5}
Weight decay	10^{-6}
Gradient clipping	global norm 5.0
Learning-rate scheduler	none (constant learning rate)

Default configuration shared across datasets. Dataset-specific batch sizes, evaluation intervals, Gaussian-mixture refresh settings, and training schedules are listed in Supplementary Appendix H.

C Dataset Descriptions

We evaluate DeMixPert on four single-cell genetic perturbation datasets, comprising two single-gene perturbation datasets, Adamson and Papalexi, and two combinatorial perturbation datasets, Norman and Replogle. These datasets cover different perturbation technologies, cellular contexts, and prediction settings. Dataset-specific preprocessing and feature construction are described in Supplementary Appendix D, while the perturbation-level splitting protocol is described in Supplementary Appendix E.

Adamson. The Adamson dataset was generated using Perturb-seq in K562 cells to characterize transcriptional responses associated with the unfolded protein response Adamson et al. (2016). Each non-control condition corresponds to a single targeted genetic perturbation.

Papalexi. The Papalexi dataset was generated using ECCITE-seq in THP-1 cells to measure the transcriptional consequences of CRISPR perturbations targeting immune-regulatory genes Papalexi et al. (2021). The dataset contains single-gene perturbation conditions.

Norman. The Norman dataset contains CRISPR activation measurements in K562 cells and was designed to study transcriptional responses and genetic interactions induced by single-gene and

Table 5: Dataset sources and post-filter statistics used in the experiments. "Single" and "Dual" denote non-control perturbation conditions; every dataset additionally contains one shared control condition.

Dataset	Source	Species	Cell line	Sequencing	Perturbation	Cells	Control cells	Genes	Single	Dual
Adamson	Adamson et al. (2016)	Human	K562	Perturb-seq	CRISPRi	56,998	7,629	2,061	76	0
Papalexi	Papalexi et al. (2021)	Human	THP-1	ECCITE-seq	CRISPR knockout	19,340	2,000	2,050	24	0
Norman	Norman et al. (2019)	Human	K562	Perturb-seq	CRISPRa	96,994	2,000	2,096	101	125
Replogle	Replogle et al. (2020)	Human	K562	direct-capture Perturb-seq	CRISPRi	26,777	2,000	2,050	33	35

combinatorial perturbations Norman et al. (2019). The formal input uses one shared gene space for all cells and perturbation conditions.

Replogle. The Replogle dataset contains both single-gene and dual-gene perturbations measured using direct guide-RNA capture technology Replogle et al. (2020).

D Data Preprocessing

This section describes data quality control, normalization, gene selection, perturbation embedding construction, and the response reference used in the main-text formulation.

All models and metrics operate on the frozen `logNor` expression layer. The feature space is the union of the 2,048 genes selected by expression variability and the perturbation target genes that are available in the dataset. The resulting gene counts are reported in Table 5. Gene selection is performed once when constructing each frozen input, before the condition-level split is generated; it is therefore shared by all five seeds and all compared methods.

Perturbation embeddings are constructed as defined in Supplementary Appendix A.

Following the main text, a control cell x_j^0 is randomly sampled from the control population for each perturbed cell x_i^p, and the response is defined as

$$\Delta x_i^p = x_i^p - x_j^0. \tag{13}$$

E Out-of-Distribution Data Splitting Protocol

We perform perturbation-level splitting to evaluate generalization to unseen perturbation conditions. The non-control perturbations assigned to the training, validation, and test sets are mutually disjoint. Control cells are not treated as prediction targets in the split; instead, they are retained as a shared reference population for constructing perturbation responses and evaluating differential expression.

E.1 Single-Gene Perturbation Datasets

For Adamson and Papalexi, all unique non-control perturbation conditions are first sorted and then randomly permuted using a seed-specific NumPy random number generator. Given N non-control conditions, the numbers assigned to the training and validation sets are computed as

$$N_{\text{train}} = \text{round}(0.7N), \qquad N_{\text{val}} = \text{round}(0.1N), \tag{14}$$

and the remaining conditions are assigned to the test set. For datasets with at least three non-control conditions, the implementation ensures that the validation and test sets are non-empty. The resulting conditions within each split are stored in lexicographic order.

E.2 Combinatorial Perturbation Datasets

For Norman and Replogle, we follow the combinatorial perturbation protocol of Wei et al. (2026). All single-gene perturbation conditions are included in the training set. The dual-gene conditions are randomly permuted, and

$$N_{\text{dual,train}} = \text{round}\left(0.5 N_{\text{dual}}\right) \tag{15}$$

dual-gene conditions are assigned to training. If N_{remain} denotes the number of remaining dual-gene conditions, the validation set receives

$$N_{\text{dual,val}} = \left\lfloor \frac{N_{\text{remain}}}{2} \right\rfloor, \tag{16}$$

and all other remaining dual-gene conditions are assigned to the test set. Therefore, when the number of remaining dual-gene conditions is odd, the test set contains one more condition than the validation set.

E.3 Random Seeds and Split Validation

All methods are evaluated using the same five data-splitting seeds:

$$\mathcal{S} = \{17, 23, 29, 31, 37\}. \tag{17}$$

For each dataset and random seed, the generated split is saved before model training and reused by DeMixPert and all comparative methods. The implementation explicitly verifies

$$\mathcal{P}_{\text{train}} \cap \mathcal{P}_{\text{val}} = \mathcal{P}_{\text{train}} \cap \mathcal{P}_{\text{test}} = \mathcal{P}_{\text{val}} \cap \mathcal{P}_{\text{test}} = \emptyset. \tag{18}$$

This condition-level separation ensures that the reported performance measures generalization to unseen perturbation conditions rather than memorization of cells from perturbations observed during training.

F Gaussian-Prototype EM Updates

This section describes the initialization, expectation-maximization procedure, and periodic refresh strategy of the Gaussian response prototypes.

Residual population. The Gaussian mixture model is fitted to training residuals only. With model parameters fixed in evaluation mode, the implementation computes

$$\epsilon_i^* = E_r(\Delta x_i^p) - \bar{r}_i \tag{19}$$

for every non-control training cell. At most the dataset-specific number of residuals in Table 7 is sampled uniformly without replacement. Residual extraction and mixture fitting use `no_grad`; consequently, the EM updates are not differentiated through.

Initialization. For $K = 8$, a seeded random permutation of the N residuals is generated and the first K residual vectors initialize the component means. The mixture weights are initialized uniformly, $\pi_k = 1/K$. Every component receives the same feature-wise population variance, with $\epsilon_{\text{cov}} = 10^{-5}$:

$$\sigma_{kj}^{2(0)} = \max\left\{ \frac{1}{N} \sum_{i=1}^{N} \left(\epsilon_{ij}^* - \bar{\epsilon}_j^*\right)^2, \epsilon_{\text{cov}} \right\}. \tag{20}$$

The covariance is diagonal; no full-covariance or tied-covariance parameter is estimated.

E step. At EM iteration t, let $v_{kj}^{(t)} = \max(\sigma_{kj}^{2(t)}, \epsilon_{\text{cov}})$. Responsibilities are computed in log space:

$$\ell_{ik}^{(t)} = \log \max(\pi_k^{(t)}, 10^{-12}) - \frac{1}{2}\sum_{j=1}^{d_r} \frac{(\epsilon_{ij}^* - \mu_{kj}^{(t)})^2}{v_{kj}^{(t)}} - \frac{1}{2}\sum_{j=1}^{d_r} \left[\log v_{kj}^{(t)} + \log(2\pi)\right], \tag{21}$$

$$\gamma_{ik}^{(t)} = \exp\left[\ell_{ik}^{(t)} - \text{LSE}_{m=1}^{K}\, \ell_{im}^{(t)}\right], \tag{22}$$

where LSE denotes log-sum-exp.

M step. Let $N_k^{(t)} = \max(\sum_i \gamma_{ik}^{(t)}, 10^{-8})$. The update is

$$\pi_k^{(t+1)} = \frac{N_k^{(t)}}{N}, \tag{23}$$

$$\mu_k^{(t+1)} = \frac{1}{N_k^{(t)}} \sum_i \gamma_{ik}^{(t)} \epsilon_i^*, \tag{24}$$

$$\sigma_{kj}^{2(t+1)} = \max\left\{ \frac{1}{N_k^{(t)}} \sum_i \gamma_{ik}^{(t)} \left(\epsilon_{ij}^* - \mu_{kj}^{(t+1)}\right)^2, \epsilon_{\text{cov}} \right\}. \tag{25}$$

The variance floor is the covariance regularizer; no additional Wishart, entropy, or Dirichlet penalty is used in EM.

Iteration and convergence. The mean log-likelihood computed in the E step,

$$\mathcal{Q}^{(t)} = \frac{1}{N} \sum_i \text{LSE}_{k=1}^{K}\, \ell_{ik}^{(t)} \tag{26}$$

is monitored once per EM iteration. EM stops when $|\mathcal{Q}^{(t)} - \mathcal{Q}^{(t-1)}| < 10^{-4}$ or after the dataset-specific maximum number of iterations in Table 7, whichever occurs first.

Empty and near-empty components. Responsibilities are soft, and the effective count is lower-bounded by 10^{-8}. Therefore, a near-empty component remains finite and is not deleted, merged, or randomly reinitialized. If fewer than K residual samples are available, the refresh routine leaves the previous mixture parameters unchanged rather than silently reducing K.

Refresh schedule and numerical stability. The Gaussian mixture model is initialized once at epoch 0 and refreshed after every dataset-specific number of training epochs using newly recomputed residuals. The refresh seed is the run seed at epoch 0 and `run_seed+epoch` thereafter. EM is performed in float32 on CPU and the resulting weights, means, and variances replace non-trainable model buffers. Besides the variance and effective-count floors, all mixture normalization uses log-sum-exp, Gaussian log-probabilities use a minimum variance of 10^{-6} at training/inference, and sampling applies the same 10^{-6} floor before the square root. Conditional mixture weights are produced with softmax; fixed global weights are clamped to 10^{-8} before taking logarithms.

G Baseline References and Code

Table 6 lists the paper and official code repository for each comparison method used in the main paper. We omit repository-specific implementation details here and refer readers to the corresponding public releases.

H Dataset-Specific Hyperparameters

This section reports the dataset-specific training, validation, and Gaussian-prototype refresh settings used in the experiments.

Table 6: Paper and official code links for the five comparison methods used in the main experiments.

Method	Paper	Official code
GEARS	https://www.nature.com/articles/s41587-023-01905-6	https://github.com/snap-stanford/GEARS
scGPT	https://www.nature.com/articles/s41592-024-02201-0	https://github.com/bowang-lab/scGPT
GenePert	https://www.biorxiv.org/content/10.1101/2024.10.27.620513v1	https://github.com/zou-group/GenePert
scDFM	https://openreview.net/forum?id=QSGanMEcUV	https://github.com/AI4Science-WestlakeU/scDFM
STATE	https://www.biorxiv.org/content/10.1101/2025.06.26.661135v1	https://github.com/ArcInstitute/state

Table 7: Dataset-specific optimization and Gaussian-mixture refresh configuration. "Cell batch" is the number of non-control cells in one optimizer update; gradient accumulation is not used. "Int." is measured in training epochs.

Dataset	LR	Cell batch	Max epochs	Eval int.	EM int.	EM samples	EM iter.	K
Adamson	5×10^{-5}	1024	5000	1000	1000	5000	15	8
Papalexi	5×10^{-5}	4096	5000	100	100	50,000	50	8
Norman	5×10^{-5}	16,384	5000	200	200	20,000	20	8
Replogle	5×10^{-5}	1024	5000	200	100	30,000	50	8

For every dataset, $\lambda_{\text{align}} = \lambda_{\text{rec}} = \lambda_{\text{dist}} = 1$ and $\lambda_{\text{gm}} = 0.01$. The Gaussian mixture model is also refreshed at epoch 0, so an EM interval of 100 denotes updates at epochs $0, 100, 200, \ldots$.

I Additional Experimental Results

This section reports the sensitivity analysis of DeMixPert. The complete ablation results are already reported in the main paper; the module-level definitions and forward-path changes are provided in Supplementary Appendix J and are not duplicated as a second figure here.

J Ablation Details and Evaluation Metrics

J.1 Ablation Implementation Details

The main paper defines four ablated variants:

- **w/o Decomp** replaces the explicit three-part decomposition $\hat{r}_i = g(s_i) + a(e_{p_i}) + \epsilon_i$ with a unified response mapping. The response is no longer represented by separately parameterized systematic, target-specific, and population components.
- **w/o Gene** removes the embedding-derived target-specific response by setting $a(e_{p_i}) = \mathbf{0}$. Hence $\bar{r}_i = g(s_i)$; the pretrained embedding-derived deterministic response branch is absent.
- **w/o Sys** removes the basal-state-dependent systematic response by setting $g(s_i) = \mathbf{0}$, yielding $\bar{r}_i = a(e_{p_i})$. The state encoder remains active because s_i is still used by the condition network and expression decoder.
- **w/o GM** removes the entire Gaussian-Prototype Invertible Network, including both the Gaussian mixture and the invertible network. No GMM is fitted, no EM refresh is performed, no coupling transform is applied, and $\epsilon_i = \mathbf{0}$. The prediction therefore reduces to the deterministic response center, $\hat{r}_i = g(s_i) + a(e_{p_i})$. The inapplicable likelihood term $\lambda_{\text{gm}}\mathcal{L}_{\text{gm}}$ is disabled.

The ablation comparison is conducted on Adamson and Replogle. Each variant is independently initialized and retrained rather than produced by post-hoc masking of a trained complete model.

Architecture-specific operations and their associated losses are disabled only when inapplicable to the corresponding variant.

J.2 Evaluation Protocol and Metric Definitions

We evaluate each test perturbation condition independently and then macro-average the resulting scores across test conditions. Let $\mathbf{X}^p \in \mathbb{R}^{N_p \times G}$ and $\widehat{\mathbf{X}}^p \in \mathbb{R}^{\widehat{N}_p \times G}$ denote the observed and predicted expression matrices under perturbation condition p, respectively. Their population-level mean expression vectors are

$$\boldsymbol{\mu}^p = \frac{1}{N_p} \sum_{i=1}^{N_p} \mathbf{x}_i^p, \qquad \widehat{\boldsymbol{\mu}}^p = \frac{1}{\widehat{N}_p} \sum_{i=1}^{\widehat{N}_p} \widehat{\mathbf{x}}_i^p. \tag{27}$$

We denote the mean expression vector of the observed control population by $\boldsymbol{\mu}^0$.

J.2.1 Top-100 DEG Metrics

Top-100 DEG selection. For each perturbation condition p, genes are ranked by comparing the observed perturbed cells with the observed control cells using the Wilcoxon rank-sum test. Let $\mathcal{S}_p^{\text{true}}$ denote the 100 highest-ranked genes. The same procedure is applied to the predicted perturbed cells and the observed control cells to obtain $\mathcal{S}_p^{\text{pred}}$. The observed set $\mathcal{S}_p^{\text{true}}$ is used as the evaluation gene space for MSE, E-Dist, and W-Dist at the top-100 DEG level.

Common differentially expressed genes (C-DEGs). C-DEGs measures the number of overlapping genes between the top-100 DEG sets obtained from the observed and predicted populations:

$$\text{C-DEGs}_p = \left| \mathcal{S}_p^{\text{true}} \cap \mathcal{S}_p^{\text{pred}} \right|. \tag{28}$$

The score ranges from 0 to 100, with a higher value indicating better recovery of the perturbation-responsive genes. The values reported in the main table may be non-integers because the condition-level counts are macro-averaged.

Mean squared error (MSE). For a gene set $\mathcal{S}$, the population-level MSE is defined as

$$\text{MSE}_p(\mathcal{S}) = \frac{1}{|\mathcal{S}|} \sum_{g \in \mathcal{S}} \left(\widehat{\mu}_g^p - \mu_g^p \right)^2. \tag{29}$$

At the top-100 DEG level, we use $\mathcal{S} = \mathcal{S}_p^{\text{true}}$. A lower value indicates more accurate recovery of the mean perturbation response.

Energy distance (E-Dist). Energy distance evaluates the discrepancy between the complete predicted and observed cell populations. Let $\widehat{\mathbf{x}}, \widehat{\mathbf{x}}'$ be independent samples from the predicted distribution and $\mathbf{x}, \mathbf{x}'$ be independent samples from the observed distribution. Energy distance is defined as

$$\begin{aligned} \text{E-Dist}_p = 2\mathbb{E}\left[\|\widehat{\mathbf{x}} - \mathbf{x}\|_2 \right] - \mathbb{E}\left[\|\widehat{\mathbf{x}} - \widehat{\mathbf{x}}'\|_2 \right] \\ - \mathbb{E}\left[\|\mathbf{x} - \mathbf{x}'\|_2 \right]. \end{aligned} \tag{30}$$

For the top-100 DEG result, all expression vectors are restricted to $\mathcal{S}_p^{\text{true}}$. Lower E-Dist indicates better agreement between the predicted and observed distributions. The code uses the empirical V-statistic

$$2\overline{d}(\widehat{X}^p, X^p) - \overline{d}(\widehat{X}^p, \widehat{X}^p) - \overline{d}(X^p, X^p), \tag{31}$$

where each bar is the mean of the complete pairwise Euclidean-distance matrix, including its zero diagonal for within-population terms. The result is clamped below at zero. Each population is subsampled without replacement to at most 2,000 cells with a fixed condition-index seed.

Wasserstein distance (W-Dist). W-Dist measures the optimal-transport cost between the empirical predicted and observed cell distributions. Using the squared Euclidean ground cost, it is calculated as

$$\text{W-Dist}_p = \left[\sum_{i,j} \gamma_{ij}^\star \left\| \widehat{\mathbf{x}}_i^p - \mathbf{x}_j^p \right\|_2^2 \right]^{1/2}, \tag{32}$$

where $\gamma^\star$ denotes the transport coupling returned by the shared optimal-transport implementation. W-Dist is evaluated on $\mathcal{S}_p^{\text{true}}$, and a lower value indicates greater distributional similarity. The primary backend is `Pertpy Distance` with the Wasserstein metric; no additional cell subsampling is applied. If Pertpy is unavailable, the released fallback computes an entropically regularized Sinkhorn cost with $\varepsilon = 0.08$ and 35 iterations and reports the square root of the non-negative cost.

J.2.2 ALL-GENE METRICS

Differential Expression Score (DES). For each perturbation p, significant DEGs are identified separately from the observed and predicted populations relative to the same observed control population. We use the Wilcoxon rank-sum test with Benjamini–Hochberg correction and an adjusted p-value threshold of 0.05. Let $\mathcal{D}_p^{\text{true}}$ and $\mathcal{D}_p^{\text{pred}}$ denote the resulting significant gene sets.

When the predicted set contains more genes than the observed set, it is truncated to the $|\mathcal{D}_p^{\text{true}}|$ genes with the largest absolute log-fold changes. Denoting the resulting predicted set by $\widetilde{\mathcal{D}}_p^{\text{pred}}$, DES is

$$\text{DES}_p = \frac{\left|\mathcal{D}_p^{\text{true}} \cap \widetilde{\mathcal{D}}_p^{\text{pred}}\right|}{\left|\mathcal{D}_p^{\text{true}}\right|}. \tag{33}$$

Conditions with no significant observed DEGs are excluded from the DES average. A higher DES indicates more accurate recovery of statistically significant perturbation-responsive genes.

All-gene MSE. All-gene MSE uses the same definition as above, but is calculated over the complete frozen gene space rather than $\mathcal{S}_p^{\text{true}}$.

Centroid accuracy. For every test perturbation, we compare its predicted population centroid with the observed centroids of all test perturbations:

$$\text{CA}_p = \mathbf{1}\left[\arg\min_{q \in \mathcal{T}} \left\|\widehat{\boldsymbol{\mu}}^p - \boldsymbol{\mu}^q\right\|_2 = p\right], \tag{34}$$

where $\mathcal{T}$ is the set of test perturbation conditions. The final centroid accuracy is the mean of CA_p across conditions. A higher value means that predicted populations preserve condition-specific identities more accurately.

Perturbation Discrimination Score (PDS). The main comparison table denotes this metric as PDS. The L1-based implementation below evaluates whether the predicted response of condition p is most similar to the corresponding observed response rather than to another test perturbation. We first define the predicted and observed perturbation-effect vectors as

$$\widehat{\boldsymbol{\delta}}^p = \widehat{\boldsymbol{\mu}}^p - \boldsymbol{\mu}^0, \qquad \boldsymbol{\delta}^q = \boldsymbol{\mu}^q - \boldsymbol{\mu}^0. \tag{35}$$

For every pair (p, q), the directly perturbed target genes of both conditions are excluded, producing the comparison gene set $\mathcal{G}_{p,q}$. The L1 distance is

$$d_{p,q} = \sum_{g \in \mathcal{G}_{p,q}} \left|\widehat{\delta}_g^p - \delta_g^q\right|. \tag{36}$$

Let r_p be the rank of $d_{p,p}$ among $\{d_{p,q} : q \in \mathcal{T}\}$ in ascending order. The condition-level score is

$$\text{PDS}_p = 1 - \frac{r_p - 1}{|\mathcal{T}|}. \tag{37}$$

The final score is averaged across all test perturbations. A higher PDS indicates better discrimination between different perturbation responses. Ties are resolved deterministically by the lexicographic condition name. If target-gene removal would eliminate the entire feature space, the implementation falls back to all genes. This L1-based quantity may be named `PDS-L1` in evaluator artifacts, but it is reported as PDS in the main paper.

Aggregation protocol. Across the five seeds defined in Supplementary Appendix E, the main table reports the arithmetic mean and sample standard deviation ($\mathrm{ddof} = 1$) across these five trials. DES conditions with no significant observed genes are represented by NaN and omitted from the DES macro-average. The per-condition evaluation caps are 64, 512, 256, and 256 cells for Adamson, Papalexi, Norman, and Replogle, respectively. All methods are evaluated using the same frozen evaluation implementation.

K Reproducibility Details

Software environment. DeMixPert is implemented in Python and PyTorch. Data processing and evaluation use AnnData, Scanpy, NumPy, pandas, and SciPy.

Random seeds. The five split seeds defined in Supplementary Appendix E are also used for model initialization. For every run, the Python, NumPy, PyTorch, and CUDA random-number generators are initialized using the corresponding seed.

Model selection. Model selection is performed exclusively on the validation set. Specifically, the checkpoint with the lowest macro-averaged top-100 DEG MSE on the validation perturbation conditions is selected. The test set is evaluated only after checkpoint selection and is never used for model selection or hyperparameter tuning. The same checkpoint-selection rule is applied to every random seed.

L Details of the Interpretability Experiments

L.1 Separability of the Decomposed Response Representation

We first examined whether the response decomposition isolates perturbation-specific information from variation driven by the basal cell state. For a perturbed cell i, the response representation was obtained from the expression difference between the perturbed cell x_i and its randomly sampled control reference x_j^0:

$$r_i^* = E_r(x_i^p - x_j^0). \tag{38}$$

The model decomposes this representation as

$$r_i^* = g_i + a_p + \epsilon_i^* \tag{39}$$

where g_i denotes the systematic response associated with the basal cell state, a_p is the perturbation-specific anchor for perturbation p, and $\epsilon_i^* = r_i^* - (g_i + a_p)$ is the target residual. The systematic component is predicted from the latent state of the paired control cell:

$$s_i = E_s(x_j^0), \qquad g_i = E_{\mathrm{sys}}(s_i). \tag{40}$$

To assess the effect of removing state-driven variation, we compared the following two response representations:

$$r_i^{\mathrm{raw}} = r_i^*, \tag{41}$$

$$r_i^{\mathrm{specific}} = r_i^* - g_i. \tag{42}$$

The second representation removes the systematic component predicted from the basal cell state while retaining the perturbation-specific anchor and cell-level residual variation.

Experimental protocol. All analyses used the most recent baseline checkpoint selected according to validation performance (`best.pt`). Checkpoints selected using test performance (`best_test.pt`), as well as checkpoints from sensitivity or ablation experiments, were not used.

We evaluated the Adamson and Replogle datasets using five random seeds:

$$\mathcal{S} = \{17, 23, 29, 31, 37\}\,. \tag{43}$$

For each dataset–seed combination, the analysis was restricted to perturbations in the test split. To prevent perturbations with larger cell populations from disproportionately influencing the clustering results, we randomly sampled an equal number of cells from every test perturbation.

For every sampled cell, we computed the raw response latent r_i^*, the control-state latent s_i, and the corresponding systematic response g_i. The specific response was subsequently obtained as $r_i^* - g_i$.

Within each dataset–seed combination, the raw and specific response representations were concatenated when fitting a `StandardScaler`. The resulting common scaling transformation was then applied to both representations. This ensured that the raw and specific responses were evaluated using the same feature scaling.

K-means clustering was performed separately on the standardized raw and specific representations. The number of clusters was set equal to the number of held-out perturbations:

$$K = \#\{\text{test perturbations}\}. \tag{44}$$

The resulting cluster assignments were evaluated against the ground-truth perturbation labels using the following metrics:

- **Adjusted Rand Index (ARI):** agreement between the inferred clustering and the true perturbation partition, adjusted for chance.
- **Normalized Mutual Information (NMI):** normalized information shared by the inferred cluster assignments and the ground-truth perturbation labels.
- **Purity:** the proportion of correctly assigned cells when each cluster is labeled according to its most frequent ground-truth perturbation.
- **Silhouette score:** the compactness and separation of the inferred clusters in the response latent space.

Let $\mathcal{C}$ denote the inferred clusters, I_p the set of evaluated cells with perturbation label p, and N the total number of evaluated cells. Cluster purity is defined as $\mathrm{Purity} = \frac{1}{N}\sum_{C\in\mathcal{C}} \max_p |C \cap I_p|$.

All quantitative metrics were computed directly in the standardized, high-dimensional response latent space. UMAP projections were used only for visualization and were not used for clustering or metric calculation.

Higher ARI, NMI, purity, and silhouette scores for r_i^{specific} than for r_i^{raw} indicate that subtracting g_i removes basal-state-driven variation and makes the perturbation-specific response structure more separable.

L.2 Association Between Prototype Weights and Residual Population Differences

We next investigated whether the Gaussian prototype weights learned by the model reflect genuine differences between perturbation-specific residual cell populations. Specifically, we tested whether perturbation pairs with more dissimilar prototype-weight profiles also exhibit more dissimilar residual-response distributions.

For perturbation p under control-cell state s_i, the mixture weights over the eight Gaussian prototypes are given by

$$\alpha_{i,p} = \mathrm{softmax}\left(h_{\mathrm{mix}}\left([s_i, a_p]\right)\right), \tag{45}$$

where a_p is the embedding-derived perturbation-specific response and $\alpha_{i,p} \in \mathbb{R}^8$ contains the corresponding prototype weights.

For each perturbation, we computed its average prototype-weight profile:

$$\bar{\alpha}_p = \frac{1}{M} \sum_{i=1}^{M} \alpha_{i,p} = \frac{1}{M} \sum_{i=1}^{M} \mathrm{softmax}\left(h_{\mathrm{mix}}\left([s_i, a_p]\right)\right), \tag{46}$$

where M is the number of control cells used to calculate the profile.

Importantly, the same set of M control cells was used for every perturbation within a dataset–seed combination. This control ensures that differences between the profiles $\bar{\alpha}_p$ are attributable to the perturbations rather than to differences in basal cell-state composition.

Experimental protocol. We used the validation-selected baseline checkpoint (`best.pt`). The main visualization was generated using the training split for Adamson seed 23 and Replogle seed 29.

For each pair of perturbations (p, q), prototype-weight dissimilarity was quantified using Jensen–Shannon divergence:

$$D_\alpha(p, q) = \mathrm{JS}\left(\bar{\alpha}_p, \bar{\alpha}_q\right). \tag{47}$$

We then derived the observed residual response for every perturbed cell. The encoded response was

$$r_i^* = E_r(x_i^p - x_j^0), \tag{48}$$

and the residual response was calculated by subtracting the systematic response and perturbation anchor:

$$\epsilon_i^* = r_i^* - (g_i + a_p). \tag{49}$$

Let E_p denote the collection of residual-response vectors associated with perturbation p. To prevent differences in cell counts from biasing the population-distance estimates, we sampled the same number of residual cells for every perturbation. Specifically, we used 30 cells per perturbation for Adamson seed 23 and 51 cells per perturbation for Replogle seed 29.

The balanced sampling procedure was independently repeated 20 times. For each repetition t, the distance between the residual populations of perturbations p and q was quantified using Energy Distance:

$$\mathrm{ED}\left(E_p^{(t)}, E_q^{(t)}\right). \tag{50}$$

The final residual-population distance was calculated by averaging over the 20 sampling repetitions:

$$D_\epsilon(p, q) = \frac{1}{20} \sum_{t=1}^{20} \mathrm{ED}\left(E_p^{(t)}, E_q^{(t)}\right). \tag{51}$$

The Adamson training split contained 53 perturbations and therefore yielded

$$\binom{53}{2} = \frac{53 \times 52}{2} = 1{,}378 \tag{52}$$

unique perturbation pairs. The Replogle training split contained 51 perturbations and yielded

$$\binom{51}{2} = \frac{51 \times 50}{2} = 1{,}275 \tag{53}$$

unique pairs. Each perturbation pair contributed one observation of the form

$$(D_\alpha(p,q), D_\epsilon(p,q)) \,. \tag{54}$$

We quantified the association between the two distance matrices by computing Spearman's rank correlation over their upper-triangular elements:

$$\rho = \mathrm{Spearman}\left(\mathrm{upper}(D_\alpha), \mathrm{upper}(D_\epsilon)\right). \tag{55}$$

Because pairwise distances are not statistically independent, significance was assessed using a Mantel-style permutation test rather than the conventional p-value returned by a standard Spearman correlation test. In each permutation, the perturbation labels of D_α were randomly reordered by jointly permuting its rows and columns, while D_ϵ was kept fixed. Spearman's ρ was then recomputed. This procedure was repeated 5,000 times, and the resulting permutation distribution was used to calculate the empirical p-value.

The resulting associations were

$$\text{Adamson, seed 23:} \quad \rho = 0.462, \qquad p_{\mathrm{Mantel}} < 0.001, \tag{56}$$

$$\text{Replogle, seed 29:} \quad \rho = 0.512, \qquad p_{\mathrm{Mantel}} < 0.001. \tag{57}$$

Visualization. Each scatter point represents a perturbation pair, with $D_\alpha(p,q)$ on the horizontal axis and $D_\epsilon(p,q)$ on the vertical axis. The LOWESS curve provides a visual summary and is not used in the significance test. The figure reports Spearman's ρ and the Mantel permutation p-value.

M Construction of Sensitivity Scores

To summarize metrics with different scales and optimization directions in the sensitivity analysis on the Papalexi dataset presented in Fig. 3, we convert each metric into a direction-aligned percentage change relative to the selected configuration, $K = 8$ and $\lambda_{\mathrm{gm}} = 0.01$. Let $x_m(p)$ denote the five-seed mean of metric m under parameter setting p, and let x_m^{ref} denote its value under the selected configuration. The transformed score is computed as

$$\Delta_m(p) = \begin{cases} 100\left(\dfrac{x_m(p)}{x_m^{\mathrm{ref}}} - 1\right), & \text{if metric } m \text{ is higher-is-better,} \\ 100\left(1 - \dfrac{x_m(p)}{x_m^{\mathrm{ref}}}\right), & \text{if metric } m \text{ is lower-is-better.} \end{cases} \tag{58}$$

The higher-is-better metrics are C-DEGs, DES, Centroid Accuracy, and PDS-L_1, whereas E-Dist, W-Dist, and MSE are lower-is-better. We then obtain each curve by taking the unweighted arithmetic mean of its constituent transformed metrics. Specifically, DEG recovery averages C-DEGs and DES; Distribution averages top-100 E-Dist, top-100 W-Dist, and all-gene E-Dist; Expression error averages top-100 MSE and all-gene MSE; and Perturbation ID averages Centroid Accuracy and PDS-L_1. All-gene E-Dist is used only as an auxiliary distributional measure in the sensitivity analysis and is not included in the primary benchmark table. It is computed using the same energy-distance definition over the complete input gene space. Consequently, positive values indicate improvement over the selected configuration, negative values indicate degradation, and the selected configuration corresponds to 0%. During each sensitivity analysis, one hyperparameter is varied while all remaining settings are held fixed.

Table 8: Comparison of DeMixPert with training-mean and additive baselines. Values are mean ± sample standard deviation across five split seeds. Bold indicates the best displayed mean within each dataset, including ties at the reported precision. Additive is not applicable to single-gene perturbation datasets.

Dataset	Method	Top 100 DEGs				All Genes			
		C-DEGs↑	E-Dist↓	W-Dist↓	MSE↓	DES↑	MSE↓	Centroid Acc↑	PDS↑
Papalexi	Training-mean	2.440 ± 0.623	0.224 ± 0.119	6.839 ± 0.447	0.015 ± 0.010	**0.261** ± 0.153	0.004 ± 0.002	0.560 ± 0.167	0.840 ± 0.075
	Additive	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	DeMixPert	**22.000** ± 4.854	**0.171** ± 0.088	**6.369** ± 0.403	**0.008** ± 0.006	0.200 ± 0.129	**0.002** ± 0.001	**0.720** ± 0.110	**0.904** ± 0.046
Adamson	Training-mean	2.920 ± 0.794	0.519 ± 0.056	5.872 ± 0.149	0.029 ± 0.005	0.084 ± 0.054	0.007 ± 0.001	0.360 ± 0.089	0.843 ± 0.044
	Additive	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	DeMixPert	**10.733** ± 5.198	**0.448** ± 0.052	**5.481** ± 0.113	**0.021** ± 0.003	**0.153** ± 0.078	**0.006** ± 0.001	**0.560** ± 0.076	**0.947** ± 0.020
Norman	Training-mean	2.984 ± 0.343	1.945 ± 0.112	6.206 ± 0.158	0.085 ± 0.009	0.349 ± 0.019	0.010 ± 0.001	0.125 ± 0.026	0.676 ± 0.010
	Additive	10.586 ± 0.866	1.282 ± 0.069	**5.922** ± 0.138	0.028 ± 0.003	0.388 ± 0.022	0.014 ± 0.001	**0.852** ± 0.039	**0.992** ± 0.001
	DeMixPert	**31.650** ± 11.739	**0.374** ± 0.056	6.606 ± 0.269	**0.003** ± 0.0002	**0.506** ± 0.063	**0.003** ± 0.0002	0.750 ± 0.163	0.981 ± 0.005
Replogle	Training-mean	1.311 ± 0.288	0.202 ± 0.049	5.829 ± 0.165	0.011 ± 0.004	0.144 ± 0.045	**0.002** ± 0.001	0.311 ± 0.183	0.731 ± 0.087
	Additive	1.444 ± 0.261	**0.153** ± 0.009	5.787 ± 0.142	0.006 ± 0.000	0.207 ± 0.057	0.003 ± 0.000	**0.800** ± 0.122	0.931 ± 0.040
	DeMixPert	**12.378** ± 4.552	0.155 ± 0.040	**5.614** ± 0.156	**0.005** ± 0.001	**0.227** ± 0.066	**0.002** ± 0.000	0.778 ± 0.192	**0.938** ± 0.039

N Comparison with Simple Baselines

Table 8 compares DeMixPert with training-mean and additive baselines. Training-mean constructs predictions from expression averages estimated from training perturbations. Additive predicts combinatorial responses by summing the constituent single-gene effects relative to control and adding the combined effect to the control profile. It is evaluated only on the combinatorial datasets, with non-applicable entries marked as N/A.

For single-gene perturbations, DeMixPert achieves better mean performance than training-mean on all eight metrics on Adamson and seven of eight on Papalexi. These gains cover DEG overlap, distributional agreement, expression reconstruction, and perturbation discrimination, with DES on Papalexi being the only exception.

For combinatorial perturbations, DeMixPert achieves higher C-DEGs and DES and lower MSE at both evaluation levels than Additive on Norman and Replogle. It additionally reduces E-Dist from 1.282 to 0.374 on Norman and achieves lower W-Dist and higher mean PDS on Replogle.

Together, these results highlight the effectiveness of DeMixPert's decomposed response modeling for unseen perturbations. Its explicit separation of systematic and perturbation-specific responses, combined with Gaussian-mixture modeling of residual population variation, provides a structured representation of both mean expression shifts and cellular heterogeneity. The observed gains in DEG overlap, response reconstruction, and multiple distributional metrics support the value of this design across single-gene and combinatorial perturbation settings.