DECOUPLING BASELINE FROM CONDITION IN SINGLE-CELL PERTURBATION DATA WITH SPLITDGD

Anonymous authors
Paper under double-blind review

ABSTRACT

Decoupling the representation of normal cellular states from condition-specific perturbation responses is a central modelling problem in single-cell biology and beyond. Achieving this decoupling without deteriorating the representation's utility for reconstruction and downstream analyses is a non-trivial trade-off. To address this, existing methods build on variational inference with soft regularisers or structural constraints on the latent, and we show that both strategies leak condition information into the invariant component or degrade the utility of the representation. We propose *splitDGD*, an extension of the Deep Generative Decoder framework (Schuster & Krogh, 2023) that enforces decoupling architecturally. This is obtained by constructing two separate latent blocks—an invariant one encoding condition-independent variations (e.g., of cell types) and a condition-specific one capturing the perturbation response—and by masking the latter when the perturbation is absent. Across controlled image benchmarks and single-cell perturbation datasets, in both unsupervised and label-supervised regimes, splitDGD is the only method to mitigate the leakage-vs-utility tradeoff that other models exhibit.

1 INTRODUCTION

Separating invariant structure from condition-specific variation is a fundamental modelling problem across data sources collected under multiple conditions. In perturbation biology, single-cell RNA sequencing measures the transcriptional profiles of thousands of individual cells under both control and perturbed conditions, resolving how a treatment, viral or bacterial infection, or genetic perturbation reshapes gene expression across heterogeneous cell populations (Kang et al., 2018; Haber et al., 2017). The central modelling challenge in this setting is to separate, at the latent level, the stable components of cellular identity—cell type, differentiation state, developmental lineage—from the condition-induced response that overlays them. This decomposition provides the basis for a wide range of downstream tasks: counterfactual prediction at the single-cell level (*what would this baseline cell look like under the perturbation?*) (Lotfollahi et al., 2019; 2023), drug-response analysis (Hetzel et al., 2022; Bunne et al., 2023), transfer to unseen cell types or unseen perturbations (Lotfollahi et al., 2022; Roohani et al., 2024), recovery of condition-specific marker genes for mechanism discovery (Xu et al., 2025; Squair et al., 2021), and integration of perturbation experiments across batches and donors (Zhang et al., 2024). The same scenario arises more broadly whenever data are observed across multiple conditions—drug doses, disease stages, experimental batches, or image transformations—and one wishes to isolate these changes from what remains constant.

Existing perturbation-decoupling methods build on variational inference, comprising an encoder and a decoder, and grow into two distinct families of models. One combines variational inference with a *soft regulariser* that pushes the latent toward condition-invariance (Lotfollahi et al., 2020; 2023; Slavutsky et al., 2025; Zhang et al., 2024; Bereket & Karaletsos, 2023; Piran et al., 2024), in cases also combined with causally-motivated structural priors (Jiang et al., 2025; Gao et al., 2025). The other family, the contrastive analysis lineage (Jones et al., 2022; Weinberger et al., 2023), enforces the separation *structurally* instead of using loss penalty terms, while keeping amortised variational inference: the latent is split and the condition-specific block is zeroed for control samples. We find that, empirically, both families suffer from a ***leakage-versus-utility*** trade-off: what a method gains in suppressing the condition in the invariant latent, it loses in the invariant content that latent must retain. On one hand, methods that leverage a soft regularizer cannot guarantee structural separation, since this is a property of the loss landscape at convergence rather than of the architecture itself (Wu et al.,

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.

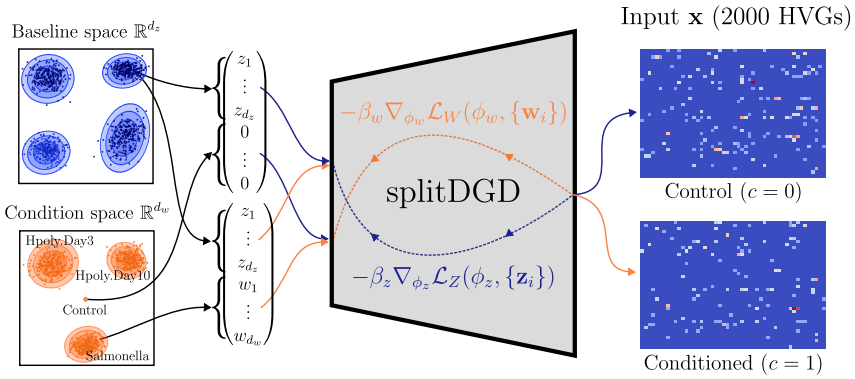


Figure 1: **Overview of *splitDGD*.** The latent decomposes into an invariant block $\mathbf{z}$ and a condition-specific block $\mathbf{w}$, each equipped with a Gaussian mixture fitted alongside the decoder. For baseline samples ($c = 0$), the $\mathbf{w}$-slot of the per-sample latent is hard-masked to zero by construction; perturbed samples ($c = 1$) carry both blocks. The decoder reconstructs the input expression vector from this concatenation. The condition-space clusters correspond to perturbation subtypes (e.g. Hpoly.Day3, Hpoly.Day10, Salmonella, from the `Haber` dataset); control cells, whose $\mathbf{w}$ is zeroed, are forced to the origin. The **blue** (**orange**, resp.) dotted arrow indicates that gradients acts on the weights ϕ_z (ϕ_w) only on samples that present the condition $c = 0$ ($c = 1$).

2024). On the other hand, existing models that impose hard constraints in the condition-specific block can still fail to enforce robust decoupling due to the amortised encoder, which can entangle condition information in the split latent space (Gabbay & Hoshen, 2019; Weinberger et al., 2022).

Motivated by this, we study the trade-off empirically and propose a method to mitigate it. We follow the contrastive analysis structural separation, but in a more accommodating model family. We propose **splitDGD**, an extension of the encoder-free *deep generative decoder* (DGD) framework (Schuster & Krogh, 2023; Schuster et al., 2024), in which per-sample latents are inferred by maximum-a-posteriori estimation jointly with the decoder weights and a latent distribution, rather than produced by an amortised encoder. splitDGD comprises three architectural components that enforce decoupling by construction: (i) a **split latent** with an invariant block $\mathbf{z}$ and a condition-specific block $\mathbf{w}$, each with its own learnable parametric prior fitted alongside the decoder; (ii) a **hard mask** that zeros $\mathbf{w}$ on non-perturbed samples; and (iii) **alternate prior training** that restricts the $\mathbf{z}$-prior to non-perturbed samples while pulling every embedding toward it. The decoupling is asymmetric: only leakage from $\mathbf{w}$ into $\mathbf{z}$ is targeted; the reverse direction is unconstrained by design.

Although our experimental setting is canonical perturbation biology (single-cell expression under a binary control-vs-perturbed indicator) our model is modality-agnostic. We show that, across two image datasets and three single-cell RNA-seq datasets, splitDGD simultaneously reduces condition leakage and preserves the invariant factor of the representations, while the other competitors typically achieve one of the two at the cost of the other. Our contributions can be summarised as follows:

- A **split-latent extension of the DGD framework** (ref. to Fig. 1) with hard masking and alternate prior training, enforcing decoupling by construction and without any regularisation pressure.
- **Comprehensive empirical validation** of our method against CVAE and three state-of-the-art competitors on three single-cell RNA-seq datasets and two image datasets, showing that splitDGD mitigates the leakage-versus-utility trade-off that other methods exhibit.

This paper is organized as follows: in Section 2 we introduce the relevant background before moving in Section 3 to present splitDGD. In Section 4, we conduct the experimental evaluation, then we conclude with related work (Section 5) and a final discussion Section 6.

2 Preliminaries

In this section, we introduce the data setting and notation and summarise the Deep Generative Decoder (DGD) framework that serves as the modelling backbone for splitDGD.

Setting and notation. We consider a dataset $\mathcal{D} = \{(\mathbf{x}_i, c_i)\}_{i=1}^N$ of N observations $\mathbf{x}_i \in \mathbb{R}^D$ together with binary condition labels $c_i \in \{0, 1\}$. Samples with $c_i = 0$ arhe *baseline* (e.g. control cells, unperturbed samples) and samples with $c_i = 1$ are *perturbed* (e.g. stimulated cells, treated samples); let $N_0 = |\{i : c_i = 0\}|$ and $N_1 = N - N_0$ denote the number of samples in each group. Our goal is to learn, for each sample, a representation that separates condition-invariant content, e.g., cellular identity and background structure, from condition-specific content present only in perturbed samples.

Deep Generative Decoders (DGDs). Unlike traditional variational autoencoders (VAEs) (Kingma, 2013), the DGD avoids the amortised-posterior approach entirely: per-sample latent representations $\mathbf{z}_i$ are treated as free parameters of the optimisation and inferred directly by maximum-a-posteriori (MAP) gradient descent alongside the decoder weights and prior parameters. Concretely, a neural decoder $g_\theta : \mathbb{R}^d \to \mathbb{R}^D$ maps each latent to an output-distribution parameter vector, $p_\theta(\mathbf{x} \mid g_\theta(\mathbf{z}))$, and a parametric prior $p_\phi(\mathbf{z})$ with learnable parameters ϕ is placed over the latent space.

Joint distribution and MAP objective of DGDs. Given model parameters (θ, ϕ) and per-sample latents $\mathbf{Z} = \{\mathbf{z}_i\}_{i=1}^N$, the DGD assumes the joint distribution factorization as:

$$P(\mathbf{X}, \mathbf{Z}, \theta, \phi) \;=\; \prod_{i=1}^N p_\theta(\mathbf{x}_i \,|\, g_\theta(\mathbf{z}_i))\; p_\phi(\mathbf{z}_i)\; p(\theta)\, p(\phi), \tag{1}$$

where $p(\theta)$ and $p(\phi)$ are hyperpriors on the decoder and prior parameters. We follow VAE convention in calling $p_\phi(\mathbf{z})$ a prior even though ϕ is learned alongside θ; only $p(\theta)$ and $p(\phi)$ are priors in a strict Bayesian sense. Training maximises model parameters *and* per-sample representations according to

$$\max_{(\theta,\,\phi,\,\mathbf{Z})} \; \sum_{i=1}^N [\log p_\theta(\mathbf{x}_i \mid g_\theta(\mathbf{z}_i)) + \log p_\phi(\mathbf{z}_i)] + \log p(\theta) + \log p(\phi). \tag{2}$$

This is maximum a posteriori (MAP) estimation applied jointly to parameters and representations. Here, the classic variational inference step is avoided, the posterior $p_\theta(\mathbf{z} \mid \mathbf{x})$ is not modeled. Training proceeds by standard stochastic gradient descent, typically with separate Adam optimisers for θ, ϕ, and $\mathbf{Z}$ (different parameter groups benefit from different learning rates).

Test-time inference. At test time, the DGD processes a new observation $\mathbf{x}^{\text{new}}$ by freezing the parameters (θ, ϕ), instantiating a fresh latent $\mathbf{z}^{\text{new}}$, and optimising $\mathbf{z}^{\text{new}}$ against the log-joint (Eq. 2) for a small number of Adam steps; all other parameters stay fixed. Because the model parameters are frozen, this is a well-posed MAP problem in the latent alone, and the returned $\mathbf{z}^{\text{new}}$ is the actual MAP estimate up to local optimiser error. The cost is a small amount of compute at inference (seconds to minutes on the dataset sizes we consider), in exchange for the absence of an amortisation gap.

3 SplitDGD: A split-latent extension for architectural decoupling

To prevent condition leakage into the invariant latent without sacrificing the latent's utility for downstream analyses—the trade-off motivated in Section 1—we extend the DGD framework with a split latent and two architectural mechanisms. Section 3.1 introduces the split-latent generative model along with its two core mechanisms; Section 3.2 describes the training objective. The parametrisation of the learnable Gaussian mixture priors and Bayesian hyperpriors is given in Appendix G.

3.1 Hard masking and alternate prior training

We split the latent space into an invariant block $\mathbf{z} \in \mathbb{R}^{d_z}$ and a condition-specific block $\mathbf{w} \in \mathbb{R}^{d_w}$, with a gating mechanism that makes $\mathbf{w}$ available to the decoder only for perturbed samples. Together with an alternate training rule for the $\mathbf{z}$-distribution (**Mechanism 2** below), these choices enforce the separation between $\mathbf{z}$ and $\mathbf{w}$ *by construction* rather than by soft objectives. An overview is in Fig. 1.

Split-latent generative model. The generative process factorises as

$$\mathbf{z}_i \sim p_{\phi_z}(\mathbf{z}) \;=\; \sum_{k=1}^{K_z} \pi_k^z\, \mathcal{N}(\mathbf{z} \,|\, \boldsymbol{\mu}_k^z,\, \boldsymbol{\Sigma}_k^z)\,, \quad \mathbf{w}_i \sim p_{\phi_w}(\mathbf{w}) \;=\; \sum_{k=1}^{K_w} \pi_k^w\, \mathcal{N}(\mathbf{w} \,|\, \boldsymbol{\mu}_k^w,\, \boldsymbol{\Sigma}_k^w)\,, \tag{3}$$

$$\mathbf{x}_i \sim p_\theta(\mathbf{x} \,|\, g_\theta([\mathbf{z}_i\,;\, c_i \cdot \mathbf{w}_i]))\,, \tag{4}$$

where $g_\theta : \mathbb{R}^{d_z + d_w} \to \mathbb{R}^D$ is a neural decoder, $[\,\cdot\,;\,\cdot\,]$ denotes concatenation, and all component covariances $\boldsymbol{\Sigma}_k^z, \boldsymbol{\Sigma}_k^w$ are diagonal. Both p_{ϕ_z} and p_{ϕ_w} are learnable Gaussian mixtures whose parameters ϕ_z, ϕ_w are fit jointly with the decoder and the per-sample representations (Appendix G). The

multiplicative mask $c_i \cdot \mathbf{w}_i$ in Eq. 4 is our gating mechanism: for baseline samples ($c_i = 0$) it zeroes the $\mathbf{w}$-slot before it reaches the decoder, so although $\mathbf{w}_i$ is drawn from the same mixture prior for every sample, baseline $\mathbf{w}_i$ plays no role in generating $\mathbf{x}_i$ (see **Mechanism 1** below). The observation likelihood $p_\theta(\mathbf{x} \mid \cdot)$ can be Gaussian, Bernoulli, Poisson, or negative binomial depending on the data modality; per-dataset choices are given in Section 4. Together, the model defines the joint distribution

$$P(\mathbf{X}, \mathbf{Z}, \mathbf{W}, \theta, \phi_z, \phi_w \mid \mathbf{c}) \;=\; \prod_{i=1}^{N} p_\theta(\mathbf{x}_i \mid g_\theta([\mathbf{z}_i;\, c_i \cdot \mathbf{w}_i]))\; p_{\phi_z}(\mathbf{z}_i)\, p_{\phi_w}(\mathbf{w}_i \mid c_i) \cdot p(\theta)\, p(\phi_z)\, p(\phi_w), \tag{5}$$

which we maximise in log space, jointly over (θ, ϕ_z, ϕ_w) and per-sample latents $(\mathbf{Z}, \mathbf{W})$, as in Eq. 2.

Mechanism 1: Hard masking. The decoder input in Eq. 4 is $[\mathbf{z}_i;\, c_i \cdot \mathbf{w}_i]$, not $[\mathbf{z}_i;\, \mathbf{w}_i]$. For baseline samples ($c_i = 0$), the multiplication $c_i \cdot \mathbf{w}_i$ produces an exact zero vector, so the decoder receives only $[\mathbf{z}_i;\, \mathbf{0}]$. This is a *structural* constraint: the model cannot use $\mathbf{w}_i$ to explain any aspect of baseline samples, because $\mathbf{w}_i$ does not reach the decoder. Consequently, $\mathbf{z}_i$ is the only way through which baseline-sample variations, e.g., the cell type identity, the digit class, or the background structure, reach the decoder. In implementation terms, the representation layer stores one trainable pair $(\mathbf{z}_i, \mathbf{w}_i)$ per training sample and returns $c_i \cdot \mathbf{w}_i$ at each forward pass. A direct consequence is that the stored $\mathbf{w}_i$ for any baseline sample receives a zero gradient at every step of training and therefore stays at its initial value under any gradient-based optimiser; initialised at $\mathbf{w}_i^{(0)} = \mathbf{0}$, it stays at $\mathbf{0}$ for the duration of training. Importantly, our goal is asymmetric: only $\mathbf{z}$'s independence from condition information is targeted; the reverse direction is not pursued.

Mechanism 2: Alternate prior training. The parameters ϕ_z of the $\mathbf{z}$-mixture are updated from baseline samples only, while the $\mathbf{z}_i$ embeddings of *all* samples are regularised toward the baseline-fitted $\mathbf{z}$-mixture. We achieve both objectives in a single backward pass by *alternating* the role of the stop-gradient operator $\mathrm{sg}(\cdot)$ across two terms. The stop-gradient acts as the identity in the forward pass and zeroes the backward gradient with respect to its argument: for any differentiable f, $\nabla_x f(\mathrm{sg}(x)) = \mathbf{0}$. We accordingly write the $\mathbf{z}$-latent loss as a sum of two terms, with $\mathrm{sg}(\cdot)$ placed so that each term's gradient flows to a different parameter set:

$$\mathcal{L}_Z(\phi_z, \{\mathbf{z}_i\}) \;=\; \underbrace{-\frac{1}{N}\sum_{i=1}^{N} \log p_{\mathrm{sg}(\phi_z)}(\mathbf{z}_i)}_{\text{updates } \{\mathbf{z}_i\}} \;\underbrace{-\frac{1}{N_0}\sum_{i:c_i=0} \log p_{\phi_z}(\mathrm{sg}(\mathbf{z}_i))}_{\text{updates } \phi_z} \;-\; \log p(\phi_z). \tag{6}$$

The first term pulls every $\mathbf{z}_i$ (baseline or perturbed) toward the current GMM modes while ϕ_z is held fixed; the second term fits ϕ_z to the baseline $\mathbf{z}$-embeddings alone while the embeddings are held fixed, under the hyperprior $p(\phi_z)$ (Appendix G). The $\frac{1}{N}$ and $\frac{1}{N_0}$ averages are estimated by per-batch averages in SGD. Only ϕ_z requires this alternation. The corresponding $\mathbf{w}$-latent loss combines the $\mathbf{w}$-embedding log-density with its prior,

$$\mathcal{L}_W(\phi_w, \{\mathbf{w}_i\}) \;=\; -\frac{1}{N_1}\sum_{i:c_i=1} \log p_{\phi_w}(\mathbf{w}_i) \;-\; \log p(\phi_w), \tag{7}$$

and, unlike $\mathcal{L}_Z$, needs no stop-gradient routing: the per-sample sum excludes baselines by construction (**Mechanism 1** freezes $\mathbf{w}_i = \mathbf{0}$ for them), so both terms back-propagate directly into ϕ_w and into $\{\mathbf{w}_i : c_i = 1\}$ in a single pass.

Supervised-w variant. Following the supervised covariate scheme of multiDGD (Schuster et al., 2024), the $\mathbf{w}$-block admits subtype labels without any architectural addition: given labels $\ell_i \in \{1, \ldots, K_w\}$ for each perturbed sample, the mixture marginal in the $\mathbf{w}$-prior loss is replaced with the single-component density

$$\log p_{\phi_w}(\mathbf{w}_i \mid \ell_i) \;=\; \log \mathcal{N}\big(\mathbf{w}_i \,\big|\, \boldsymbol{\mu}^w_{\ell_i},\, \boldsymbol{\Sigma}^w_{\ell_i}\big)\,, \tag{8}$$

so that each perturbed sample contributes only to its assigned component (hard assignment) and the mixture weights $\boldsymbol{\pi}^w$ are regularised solely by their Dirichlet hyperprior (Appendix G). No auxiliary classifier head and no additional loss term are needed. When labels are absent, we use the full mixture density $\log p_{\phi_w}(\mathbf{w}_i)$ (Eq. 7); the two regimes share the same architecture, loss structure, and optimisation scheme. In our experiments, the unlabelled regime is the default and is denoted simply *splitDGD*; the labelled regime is denoted *splitDGD (sup-*$\mathbf{w}$*)*.

3.2 TRAINING

In the DGD framework, per-sample latents are free optimisation parameters rather than encoder outputs (Section 2). Concretely, we store training-sample representations in two tables $\mathbf{Z}_{\text{train}} \in \mathbb{R}^{N \times d_z}$ and $\mathbf{W}_{\text{train}} \in \mathbb{R}^{N \times d_w}$, indexed by sample identity, and treat their entries as trainable parameters alongside (θ, ϕ_z, ϕ_w). The full per-sample loss is

$$\mathcal{L}(\theta, \phi_z, \phi_w, \{\mathbf{z}_i, \mathbf{w}_i\}) = \underbrace{\mathcal{L}_{\text{recon}}(\theta, \{\mathbf{z}_i, \mathbf{w}_i\})}_{\text{reconstruction NLL}} + \beta_z \mathcal{L}_Z(\phi_z, \{\mathbf{z}_i\}) + \beta_w \mathcal{L}_W(\phi_w, \{\mathbf{w}_i\}), \quad (9)$$

where $\mathcal{L}_{\text{recon}} = -\frac{1}{N}\sum_i \log p_\theta(\mathbf{x}_i \mid g_\theta([\mathbf{z}_i; c_i \mathbf{w}_i]))$ is the reconstruction negative log-likelihood under the output module, $\mathcal{L}_Z$ is defined in Eq. 6, and $\mathcal{L}_W$ in Eq. 7. The coefficients $\beta_z, \beta_w > 0$ weight $\mathcal{L}_Z$ and $\mathcal{L}_W$; we fix $\beta_z = 1$ so that $\mathcal{L}_Z$ carries the same per-sample weight as $\mathcal{L}_{\text{recon}}$, and treat β_w as a hyperparameter. The four parameter groups—decoder weights θ, prior parameters ϕ, train representations, and test representations—are optimised by separate Adam optimisers with their own learning rates; full details are given in Appendix H.

4 EXPERIMENTS

As mentioned in Section 1, suppressing condition leakage from the invariant latent may come at the expense of the representation's utility. In order to evaluate this leakage-versus-utility trade-off, we use four evaluation criteria: a leakage criterion (**C1**), and three utility criteria (**C2–C4**):

- **(C1) Decoupling**. The invariant latent $\mathbf{z}$ carries no information about the condition label c, in the sense that no classifier can recover c from $\mathbf{z}$ above chance. **C1** defines the decoupling task.
- **(C2) Preservation**. The same $\mathbf{z}$ retains the invariant factor it is meant to encode—cell type, digit identity, or baseline class—recoverable by a supervised classifier.
- **(C3) Coverage**. The condition-specific latent $\mathbf{w}$ captures the variation induced by the perturbation. When the perturbation has internal structure (multiple drugs, doses, or subtypes), $\mathbf{w}$ encodes this structure rather than collapsing to a single mode.
- **(C4) Fidelity**. Observed samples are reconstructed faithfully and counterfactual predictions (*"what would this baseline sample look like under the perturbation?"*) are accurate.

The ***leakage-versus-utility*** trade-off concerns whether enforcing **C1** costs on any of the three utility axes, and vice versa. **C1** is the primary constraint: it defines what decoupling is, and a method failing **C1** has not decoupled. **C2**, **C3**, and **C4** are utility criteria: they determine whether the decoupled representation remains useful. We remark that **C2** excludes degenerate solutions that satisfy **C1** by collapsing $\mathbf{z}$ to a constant or a random vector. Each criterion maps to a concrete metric, defined together with the experimental setup in Section 4.2. These four criteria define the three questions we address in Sections 4.3, 4.4, and 4.5 before analyzing the leakage-*vs*-utility trade-off (Section 4.6):

- **Q1** [C1] Does splitDGD reduce condition leakage into $\mathbf{z}$ better than soft-objective baselines?
- **Q2** [C2, C3] Does the reduced leakage preserve $\mathbf{z}$'s recovery of the invariant factor, and does $\mathbf{w}$ recover the condition-specific factor when present?
- **Q3** [C4] Is reconstruction and counterfactual fidelity preserved under the architectural mask?

4.1 DATASETS

We use two image and three single-cell RNA-seq datasets, summarised in Table 1. Each dataset provides a binary condition $c \in \{0, 1\}$, a ground-truth $\mathbf{z}$-factor (the invariant of interest, a class or cell type), and, where available, a ground-truth $\mathbf{w}$-subtype factor. Only the binary condition is used during training; the $\mathbf{z}$- and $\mathbf{w}$-factor labels are reserved for evaluation, except in supervised-$\mathbf{w}$ experiments where the $\mathbf{w}$-subtype label is supplied at training time. Additional dataset details are in Appendix B.

4.2 EXPERIMENTAL SETUP

Baselines. We evaluate splitDGD against CVAE (Sohn et al., 2015) and three state-of-the-art perturbation-disentanglement baselines: contrastiveVI (Weinberger et al., 2023), DisCoVR (Slavutsky

Table 1: **Datasets.** N is the total sample count. K_z and K_w are the cardinalities of the $\mathbf{z}$- and $\mathbf{w}$-subtype factors used for evaluation, respectively. `Kang` has no ground-truth $\mathbf{w}$-subtypes.

Dataset	Modality	N	Features	K_z	K_w	$\mathbf{w}$ factor
`DF-MNIST`	Image, 28×28	60 000	784	10	10	MNIST digit overlay
`Morpho-MNIST`	Image, 28×28	180 000	784	10	2	stroke style (thick / thin)
`Kang` (PBMC IFN-β)	scRNA-seq, log-norm	13 999	2000 HVGs	10	—	binary: IFN-β response
`Haber` (H.poly)	scRNA-seq, log-norm	9 842	2000 HVGs	8	3	Infection type
`Sci-Plex3`	scRNA-seq, log-norm	160 329	8630 HVG+DEG	3	188	compound identity

et al., 2025), and scDisInFact (Zhang et al., 2024), covering both families of Section 1: DisCoVR and scDisInFact are *soft* methods, separating the blocks through an adversarial classifier and a classification head respectively; contrastiveVI is the *hard* method, zeroing the condition-specific block of control samples under amortised inference; CVAE has a single latent space and serves as a reference with no decoupling mechanism. Per-method details and configurations are in Appendix D.

Training protocol. We distinguish two *supervision regimes*: (i) The *unsupervised*-$\mathbf{w}$ regime contains methods that consume only c: CVAE, splitDGD, contrastiveVI, and DisCoVR; (ii) The *supervised*-$\mathbf{w}$ regime contains methods that additionally use a subtype label. Here we test: splitDGD (sup-$\mathbf{w}$) via the label-conditional of Eq. 8 and scDisInFact via its classification term on $\mathbf{z}$. The supervised regime applies to all datasets except `Kang`, which has no $\mathbf{w}$-subtype. scDisInFact and contrastiveVI run on single-cell data only and therefore do not appear on the image datasets. All the models' hyperparameters are selected per seed, using an held-out validation fold (details in Appendix C).

Evaluation. To evaluate the performance we use the following six metrics:

- **AUC (linear / nonlinear)** (↓, **C1**). Test AUC of logistic-regression (linear) and Random Forest (RF) nonlinear probes predicting c from $\mathbf{z}$; 0.5 is chance. The nonlinear probe catches higher-order leakage that methods may fail to suppress.
- **$\mathbf{z}$-MCC and $\mathbf{w}$-MCC** (↑, **C2**, **C3**). Multiclass Matthews Correlation Coefficient (MCC) of an RF probe recovering the ground-truth invariant factor (cell type or class) from $\mathbf{z}$ on all samples ($\mathbf{z}$-MCC, **C2**), and the ground-truth subtype factor from $\mathbf{w}$ on $c = 1$ samples only ($\mathbf{w}$-MCC, **C3**).
- **RMSE** (↓, **C4**). Joint-latent reconstruction RMSE on log-normalised expression (single-cell) or $[0, 1]$ pixels (image).
- **Counterfactual R^2** (↑, **C4**). For each control cell and each perturbation p, the model produces a counterfactual prediction of the cell's expression had it received p. The metric, following the in protocol (Lotfollahi et al., 2019; 2020), is the Pearson R^2 between predicted and true per-cell-type mean expression, averaged across perturbations. This is reported only for single-cell datasets as image datasets do not admit this evaluation, full procedure in Appendix I.

`DF-MNIST`, `Morpho-MNIST`, `Kang` and `Haber` are each partitioned via an 80/20 stratified split on the condition label, repeated for 10 independent seeds; probes share the same partition as the representation, and all metrics are reported as mean ± std across the 10 seeds. On `Haber`, we additionally break the counterfactual R^2 down by perturbation subtype (Hpoly.Day3, Hpoly.Day10, Salmonella), as pooled R^2 can mask differential subtype behaviour. `Sci-Plex3` follows a different protocol, designed to test generalisation to *unseen* (cell line, compound) pairs: we adopt PerturBench's covariate-transfer split strategy (Wu et al., 2024), which holds out 30% of the 188 compounds within each cell line, so every test pair is absent from training while the same compound is observed in the other two lines; validation and test compounds are disjoint, and each held-out compound appears in exactly one line. We first analyse each criterion individually (Sections 4.3–4.5); then we combine the results to investigate for all methods the leakage-vs-utility trade-off in Section 4.6.

4.3 Condition leakage (Q1)

splitDGD achieves the intended decoupling: condition information is largely removed from $\mathbf{z}$, as the **C1** columns of Tables 2 and 3 report and as illustrated visually in Appendix Figs. 4 and 5. On both image datasets (Appendix A), splitDGD attains the lowest linear and nonlinear AUC for the condition prediction from $\mathbf{z}$. Gaps to baselines are large; CVAE reaches 0.91 linear AUC on `DF-MNIST` against splitDGD's 0.519; DisCoVR, despite achieving moderate decoupling (0.65 and 0.59 on linear AUC

Table 2: **Results on single-cell datasets (`Kang`, `Haber`, `Sci-Plex3`)**. On `Kang` and `Haber`, mean ± std over 10 split seeds; methods are grouped by **w**-supervision, and **bold** and underline mark the best and second-best within each subgroup (second-best omitted in two-method subgroups). On `Sci-Plex3`, mean ± std over 5 split seeds of the covariate-transfer split; **bold** and underline mark the best and second-best across all six methods, Dashes (—) mark cases where metrics do not apply.

			Cond. **z**-leakage ↓		**z**-recovery ↑	**w**-recovery ↑	Recon. ↓	CF R^2 ↑
		Method	AUC (lin)	AUC (NL)	**z**-MCC	**w**-MCC	RMSE	all genes
`Kang`		CVAE	0.679 ±.038	0.775 ±.020	0.757 ±.027	—	0.166 ±.001	0.944 ±.013
		DisCoVR	0.670 ±.100	0.740 ±.066	0.776 ±.024	—	0.174 ±.002	0.891 ±.015
		scDisInFact	0.564 ±.021	**0.669** ±.027	**0.838** ±.030	—	0.171 ±.002	0.871 ±.012
		ContrastiveVI	0.697 ±.020	0.717 ±.028	0.783 ±.028	—	0.173 ±.004	0.893 ±.012
		splitDGD	**0.521** ±.031	0.713 ±.047	0.820 ±.039	—	**0.157** ±.001	**0.960** ±.010
`Haber`	*unsup-***w**	CVAE	0.679 ±.049	0.739 ±.023	0.738 ±.023	—	0.205 ±.002	**0.897** ±.006
		DisCoVR	0.658 ±.024	0.702 ±.022	0.675 ±.037	0.588 ±.022	0.228 ±.003	0.831 ±.020
		ContrastiveVI	0.684 ±.043	0.774 ±.028	**0.755** ±.011	**0.843** ±.035	0.224 ±.001	0.819 ±.015
		splitDGD	**0.550** ±.021	**0.694** ±.029	0.730 ±.044	0.599 ±.065	**0.198** ±.007	0.885 ±.044
	*sup-***w**	scDisInFact	0.585 ±.050	0.688 ±.013	**0.778** ±.015	0.833 ±.066	0.229 ±.003	0.824 ±.013
		splitDGD (sup-**w**)	**0.532** ±.026	**0.687** ±.032	0.740 ±.023	**1.000** ±.001	**0.196** ±.005	**0.926** ±.009
`Sci-Plex3`		CVAE	0.613 ±.012	0.615 ±.012	0.962 ±.017	—	0.150 ±.001	**0.954** ±.008
		ContrastiveVI	0.598 ±.018	**0.577** ±.008	0.978 ±.004	0.008 ±.003	0.155 ±.001	0.917 ±.007
		scDisInFact*	0.600 ±.014	0.590 ±.011	**0.983** ±.002	−0.002 ±.003	0.168 ±.002	0.872 ±.011
		DisCoVR	0.639 ±.006	0.639 ±.007	0.982 ±.003	−0.006 ±.002	0.205 ±.000	0.847 ±.019
		splitDGD	0.591 ±.077	0.633 ±.014	0.957 ±.011	0.003 ±.003	**0.149** ±.001	0.953 ±.007
		splitDGD (sup-**w**)	**0.555** ±.038	0.587 ±.033	0.974 ±.007	**0.905** ±.179	**0.149** ±.001	0.939 ±.022

* On `Sci-Plex3` scDisInFact is trained on the binary condition, as the 189-way encoding is infeasible at that scale, so splitDGD (sup-**w**) is the only supervised-**w** method there.

on `DF-MNIST` and `Morpho-MNIST`, respectively), is still significantly worse than both versions of splitDGD, whose largest linear AUC is 0.52. On all single-cell datasets (Table 2), splitDGD is the best or second-best on every linear and nonlinear AUC entry; scDisInFact achieves the lowest nonlinear AUC on `Kang` (0.669 vs. splitDGD 0.713) and contrastiveVI on `Sci-Plex3` (0.577 vs. 0.587), but both differences stay within splitDGD standard deviation. On `Haber` nonlinear AUC, the scores of splitDGD (sup-**w**, 0.687) and scDisInFact (0.688) are statistically tied, with the gap of only 0.001 smaller than the standard deviation of either methods.

Overall, splitDGD models excel at reducing condition leakage (**C1**), with scDisInFact being the runner-up on the leakage metrics; CVAE ranks always among the worst models, as expected for a method without any explicit decoupling mechanism. ContrastiveVI and DisCoVR decouple only partially, remaining close to CVAE on every single-cell dataset.

4.4 Latent factors recovery (Q2)

C2 requires that **z** encode the invariant factors; **C3** that **w** encode the condition-specific structure. Supplementary Figure 4 qualitatively shows both for the single cell `Haber` data: **z** represents the eight epithelial cell types as distinct clusters and **w** separates the three perturbation subtypes (even without subtype supervision). The results on the two metrics for **Q2** are summarised as follows:

Invariant factors recovery (z-MCC): No method leads on **z**-MCC throughout: CVAE is best on both image datasets (0.664, 0.766), scDisInFact on `Kang`, `Haber` among supervised models, and on `Sci-Plex3` (0.838, 0.778, 0.983), and contrastiveVI on `Haber` among unsupervised models (0.755). On the single-cell datasets splitDGD falls always within seed variance of the best model, ranking second best on `Kang` (0.820 vs. 0.838); on the image datasets it is within seed variance of DisCoVR, the only other perturbation-decoupling method evaluated there (0.620 vs. 0.632 on `DF-MNIST`, 0.722 vs. 0.747 on `Morpho-MNIST`). We notice that, since on `Sci-Plex3` all three cell lines are easily separable, all methods score above 0.95, with the best scorer, scDisInFact, increasing by less than 1% over splitDGD (sup-**w**).

Perturbation-structure recovery (w-MCC): splitDGD is the leader (or essentially tied) on **w**-MCC on most applicable case. With subtype label supervision, splitDGD (sup-**w**) saturates the

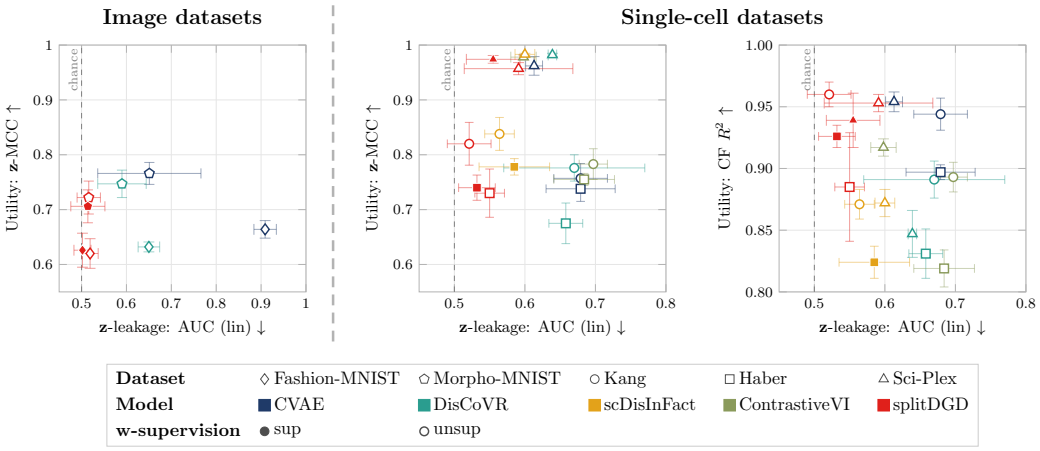


Figure 2: **Leakage-versus-utility trade-off**. Each point is one (method, dataset) pair, with condition leakage from $\mathbf{z}$ on the x-axis (test AUC of a linear probe; 0.5 is chance) and utility on the y-axis: $\mathbf{z}$-MCC for the image datasets (left) and the single-cell datasets (center), counterfactual R^2 for the single-cell datasets (right). Bars are $\pm$ one std over split seeds.

metric: ≥ 0.99 on `DF-MNIST`, `Morpho-MNIST`, and `Haber`, and 0.905 on the 188-compound `Sci-Plex3` task. scDisInFact is the runner-up in `Haber`, but scoring 0.833. Without labels, the outcome depends on the cardinality of the subtype factor. For $K_w \leq 10$, splitDGD yields moderate-to-high scores on `DF-MNIST`, `Morpho-MNIST`, and `Haber` (0.539, 0.967, 0.599); DisCoVR matches it on `Morpho-MNIST` and `Haber` but collapses to chance on `DF-MNIST` (0.065), and contrastiveVI leads only on `Haber` (0.843). For $K_w = 188$ in `Sci-Plex3`, no label-unsupervised method recovers compound identity: every method lies around 0.01 chance, splitDGD included. CVAE has no $\mathbf{w}$-block and `Kang` no $\mathbf{w}$-subtype, so these entries do not apply.

Taken together, splitDGD stays within 0.04 of the best $\mathbf{z}$-MCC (**C2**) on single-cell data and 0.06 on images, and on $\mathbf{w}$-MCC (**C3**) it is the only unsupervised method that never collapses for $K_w \leq 10$ and the only method that scales to 188 classes with labels.

4.5 Reconstruction and counterfactual prediction (Q3)

Q3 asks whether reconstruction and counterfactual fidelity (**C4**) hold up for the models.

Reconstruction. On the single-cell datasets splitDGD has the lowest RMSE on every entry of Table 2: CVAE is always second best, worsening by at most 6% w.r.t. splitDGD, the split-latent baselines error increases from 4% to 38%, the largest gap being DisCoVR on `Sci-Plex3`. On the image datasets (Appendix A), CVAE has the lowest RMSE; splitDGD and DisCoVR sit 0.02–0.04 behind it, showing a similar cost in reconstruction error to separate condition information.

Counterfactual $\mathbf{R^2}$. splitDGD is best or second-best on every single-cell entry: best on `Kang` (0.960 vs. CVAE 0.944) and on `Haber` supervised (0.926 vs. scDisInFact 0.824), second to CVAE by at most 0.01 on `Haber` unsupervised and `Sci-Plex3`, and 0.036–0.106 ahead of contrastiveVI, DisCoVR, and scDisInFact throughout (splitDGD (sup-$\mathbf{w}$) is third on `Sci-Plex3`, 0.939).[1]

Overall, across **C4**, the split-latent baselines are the weakest group: contrastiveVI, DisCoVR, and scDisInFact score below splitDGD on every single-cell entry (both on RMSE and counterfactual R^2) whereas CVAE, with no $\mathbf{w}$-block, is the only method that comes close to it. On `Sci-Plex3`, counterfactual R^2 is nearly saturated, since predicting the target line's control profile already scores ≈ 0.95 (Appendix I); the rank metric of Appendix J, which a compound-agnostic prediction cannot pass, separates the methods instead, and it places splitDGD (sup-$\mathbf{w}$) first and CVAE at chance.

[1] On `Sci-Plex3` Counterfactual R^2 targets unseen (cell line, compound) pairs, with the compound's $\mathbf{w}$ transferred from the other two cell lines (see Appendix I); further metrics for this dataset are given in Appendix J.

4.6 LEAKAGE-VERSUS-UTILITY TRADE-OFF COMPARISON

Taken together, the four criteria sum to the *leakage-versus-utility trade-off* on which splitDGD occupies the best position. It has the lowest or second-lowest leakage on every dataset (**Q1**) while its utility stays with the leaders: within seed variance of the best $\mathbf{z}$-MCC on single-cell data, first or second on every counterfactual R^2 entry, best on every single-cell RMSE (**Q2**, **Q3**). Every baseline achieves one side at the expense of the other: CVAE keeps utility but is among the leakiest models, scDisInFact reduces leakage but falls behind on fidelity, contrastiveVI best recovers perturbation structure on `Haber` but at the expense of highest leakage, and DisCoVR falls behind on both aspects.

In Fig. 2 we report a visual summary along two main coordinates: the **C1**-*vs*-**C2** trade-off for image and single-cell datasets, and the **C1**-*vs*-**C4** trade-off for only single-cell datasets. The ideal position is at the top-left corner ($\mathbf{z}$-leakage $= 0.5$, utility $= 1$), and in no panel does a competitor improve on splitDGD along both axes. The analysis of the **C1**-*vs*-**C3** (ref. to Section E) portrays a similar picture with analogous results. We also design a catastrophic-failure analysis in Appendix F, where methods are accounted for as performing significantly worse than the best model: splitDGD has one event in 27 cells and splitDGD (sup-$\mathbf{w}$) none, whereas every baseline has at least four.

5 RELATED WORK

Perturbation decoupling with soft objectives. For methods in this family, the separation success mainly depends on the regulariser's strength and on the optimization; PerturBench documents this fragility across models and datasets (Wu et al., 2024). Other models with causal and identifiability-motivated priors (Gao et al., 2025; An et al., 2025; Jiang et al., 2025) give stronger guarantees, but require prior causal knowledge that is not available in many control-*vs*-perturbed datasets.

Contrastive analysis methods isolate the variation present in a target population from a background control (Abid et al., 2018; Abid & Zou, 2019; Severson et al., 2019; Weinberger et al., 2023). Almost all methods build on one structure: a shared block and a salient block, with the salient block fixed to zero for background samples. They employ amortised variational inference, the structure is only partly enforced: the zeroing does not enter the objective directly, leakage between the two blocks is acknowledged within the lineage and patched with distribution-matching penalties (Weinberger et al., 2022); moreover proovable decoupling is not guaranteed for this model class in general, but only under restrictive conditions on the decoder and the latent dimensions (Lopez et al., 2024).

Representational disentanglement. Outside perturbation biology, our setting of decoupling condition-specific latents to data-specific traits shares many similarities to the study of disentanglement of representations (Higgins et al., 2018; Suter et al., 2019; Schölkopf et al., 2021; Lippe et al., 2022; Marconato et al., 2022; Fumero et al., 2023). Provable conditions to obtain disentanglement often require extensive data variations or using auxiliary variables (Lachapelle et al., 2023; Hyvärinen et al., 2024); they do not consider isolating only specific factors of variation, e.g., a binary condition label, from other generic changes, which is the primary focus of our work.

6 CONCLUSION

We presented *splitDGD*, a split-latent extension of the encoder-free Deep Generative Decoder framework that decouples invariant content from the perturbation response architecturally rather than through soft regularisation at the loss level. Across two controlled image benchmarks and three single-cell perturbation datasets, splitDGD is the only method to consistently mitigate the leakage-versus-utility trade-off that other approaches exhibit; the cost on the complementary fidelity metrics is bounded and structural, and the lowest among methods that share its decoupling goal.

Two limitations remain. First, splitDGD targets a binary control-versus-perturbed setting; multi-level or continuous conditions lie beyond its current scope. Second, the architectural mask trades a small amount of latent factors recovery for the decoupling guarantee; the cost is bounded across all our experiments and never catastrophic. Future work points in two directions: generalising the framework to multi-level condition structures (drug-and-dose perturbation panels, time-course experiments, hierarchically nested perturbations), and transferring it to settings sharing the same baseline-versus-attribute decomposition (batch correction, fairness, domain adaptation, style transfer).

AI USE STATEMENT

We used generative AI tools for code review, for designing the plotting interface, for reviewing the manuscript, and for a general literature search. We did not use them to conceive the work, to design the method, or to write the main experimental setup, nor for any other task that requires disclosure. The authors reviewed all AI-assisted output, through extensive checks and manual editing of generated code, and take full responsibility for the content of this work, including any text, claims or artefacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

All experimental details can be found in Section 4 and in Appendix. We also attached an anonymized version of the code in Appendix C, which will be published upon acceptance.

REFERENCES

Abubakar Abid and James Zou. Contrastive variational autoencoder enhances salient features. *arXiv preprint arXiv:1902.04601*, 2019.

Abubakar Abid, Martin J Zhang, Vivek K Bagaria, and James Zou. Exploring patterns enriched in a dataset with contrastive principal component analysis. *Nature communications*, 9(1):2134, 2018.

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 2623–2631, 2019.

Shaokun An, Jae-Won Cho, Kai Cao, Jiankang Xiong, Martin Hemberg, and Lin Wan. sccausalvi disentangles single-cell perturbation responses with causality-aware generative model. *Cell Systems*, 16(11), 2025.

Michael Bereket and Theofanis Karaletsos. Modelling cellular perturbations with the sparse additive mechanism shift variational autoencoder. *Advances in Neural Information Processing Systems*, 36: 1–12, 2023.

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.

Daniel C Castro, Jeremy Tan, Bernhard Kainz, Ender Konukoglu, and Ben Glocker. Morpho-mnist: quantitative assessment and diagnostics for representation learning. *Journal of Machine Learning Research*, 20(178):1–29, 2019.

Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.

Marco Fumero, Florian Wenzel, Luca Zancato, Alessandro Achille, Emanuele Rodolà, Stefano Soatto, Bernhard Schölkopf, and Francesco Locatello. Leveraging sparse and shared feature activations for disentangled representation learning. *Advances in Neural Information Processing Systems*, 36: 27682–27698, 2023.

Aviv Gabbay and Yedid Hoshen. Demystifying inter-class disentanglement. *arXiv preprint arXiv:1906.11796*, 2019.

Yicheng Gao, Kejing Dong, Caihua Shan, Dongsheng Li, and Qi Liu. Causal disentanglement for single-cell representations and controllable counterfactual generation. *Nature communications*, 16 (1):6775, 2025.

Adam L Haber, Moshe Biton, Noga Rogel, Rebecca H Herbst, Karthik Shekhar, Christopher Smillie, Grace Burgin, Toni M Delorey, Michael R Howitt, Yarden Katz, et al. A single-cell survey of the small intestinal epithelium. *Nature*, 551(7680):333–339, 2017.

Leon Hetzel, Simon Boehm, Niki Kilbertus, Stephan Günnemann, Fabian Theis, et al. Predicting cellular responses to novel drug perturbations at a single-cell resolution. *Advances in Neural Information Processing Systems*, 35:26711–26722, 2022.

Irina Higgins, David Amos, David Pfau, Sebastien Racaniere, Loic Matthey, Danilo Rezende, and Alexander Lerchner. Towards a definition of disentangled representations. *arXiv preprint arXiv:1812.02230*, 2018.

Aapo Hyvärinen, Ilyes Khemakhem, and Ricardo Monti. Identifiability of latent-variable and structural-equation models: from linear to nonlinear. *Annals of the Institute of Statistical Mathematics*, 76(1):1–33, 2024.

Wenkang Jiang, Yuhang Liu, Yichao Cai, Erdun Gao, Jiayi Dong, Ehsan Abbasnejad, and Javen Qinfeng Shi. Identifying unperturbed cellular programs enables accurate single-cell perturbation prediction. 2025.

Andrew Jones, F William Townes, Didong Li, and Barbara E Engelhardt. Contrastive latent variable modeling with application to case-control sequencing experiments. *The annals of applied statistics*, 16(3):1268, 2022.

Hyun Min Kang, Meena Subramaniam, Sasha Targ, Michelle Nguyen, Lenka Maliskova, Elizabeth McCarthy, Eunice Wan, Simon Wong, Lauren Byrnes, Cristina M Lanata, et al. Multiplexed droplet single-cell rna-sequencing using natural genetic variation. *Nature biotechnology*, 36(1): 89–94, 2018.

Diederik P Kingma. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Sébastien Lachapelle, Tristan Deleu, Divyat Mahajan, Ioannis Mitliagkas, Yoshua Bengio, Simon Lacoste-Julien, and Quentin Bertrand. Synergies between disentanglement and sparsity: Generalization and identifiability in multi-task learning. In *International Conference on Machine Learning*, pp. 18171–18206. PMLR, 2023.

Phillip Lippe, Sara Magliacane, Sindy Löwe, Yuki M Asano, Taco Cohen, and Stratis Gavves. Citris: Causal identifiability from temporal intervened sequences. In *International Conference on Machine Learning*, pp. 13557–13603. PMLR, 2022.

Romain Lopez, Jan-Christian Huetter, Ehsan Hajiramezanali, Jonathan K Pritchard, and Aviv Regev. Toward the identifiability of comparative deep generative models. In *Causal Learning and Reasoning*, pp. 868–912. PMLR, 2024.

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

Mohammad Lotfollahi, Mohsen Naghipourfar, Fabian J Theis, and F Alexander Wolf. Conditional out-of-distribution generation for unpaired data using transfer vae. *Bioinformatics*, 36(Supplement_2): i610–i617, 2020.

Mohammad Lotfollahi, Mohsen Naghipourfar, Malte D Luecken, Matin Khajavi, Maren Büttner, Marco Wagenstetter, Žiga Avsec, Adam Gayoso, Nir Yosef, Marta Interlandi, et al. Mapping single-cell data to reference atlases by transfer learning. *Nature biotechnology*, 40(1):121–130, 2022.

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.

Emanuele Marconato, Andrea Passerini, and Stefano Teso. Glancenets: Interpretable, leak-proof concept-based models. *Advances in Neural Information Processing Systems*, 35:21212–21227, 2022.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Zoe Piran, Niv Cohen, Yedid Hoshen, and Mor Nitzan. Disentanglement of single-cell data with biolord. *Nature Biotechnology*, 42(11):1678–1683, 2024.

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

Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634, 2021.

Viktoria Schuster and Anders Krogh. The deep generative decoder: Map estimation of representations improves modelling of single-cell rna data. *Bioinformatics*, 39(9):btad497, 2023.

Viktoria Schuster, Emma Dann, Anders Krogh, and Sarah A Teichmann. multidgd: A versatile deep generative model for multi-omics data. *Nature Communications*, 15(1):10031, 2024.

Kristen A Severson, Soumya Ghosh, and Kenney Ng. Unsupervised learning with contrastive latent variable models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 4862–4869, 2019.

Yuli Slavutsky, Ozgur Beker, David Blei, and Bianca Dumitrascu. Variational learning of disentangled representations. *arXiv preprint arXiv:2506.17182*, 2025.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28, 2015.

Jordan W Squair, Matthieu Gautier, Claudia Kathe, Mark A Anderson, Nicholas D James, Thomas H Hutson, Rémi Hudelle, Taha Qaiser, Kaya JE Matson, Quentin Barraud, et al. Confronting false discoveries in single-cell differential expression. *Nature communications*, 12(1):5692, 2021.

Sanjay R Srivatsan, José L McFaline-Figueroa, Vijay Ramani, Lauren Saunders, Junyue Cao, Jonathan Packer, Hannah A Pliner, Dana L Jackson, Riza M Daza, Lena Christiansen, et al. Massively multiplex chemical transcriptomics at single-cell resolution. *Science*, 367(6473):45–51, 2020.

Raphael Suter, Djordje Miladinovic, Bernhard Schölkopf, and Stefan Bauer. Robustly disentangled causal mechanisms: Validating deep representations for interventional robustness. In *ICML*, 2019.

Ethan Weinberger, Nicasia Beebe-Wang, and Su-In Lee. Moment matching deep contrastive latent variable models. *arXiv preprint arXiv:2202.10560*, 2022.

Ethan Weinberger, Chris Lin, and Su-In Lee. Isolating salient variations of interest in single-cell data with contrastivevi. *Nature methods*, 20(9):1336–1345, 2023.

Yan Wu, Esther Wershof, Sebastian M Schmon, Marcel Nassar, Błażej Osiński, Ridvan Eksi, Zichao Yan, Rory Stark, Kun Zhang, and Thore Graepel. Perturbench: Benchmarking machine learning models for cellular perturbation analysis. *arXiv preprint arXiv:2408.10609*, 2024.

Yang Xu, Stephen Fleming, Matthew Tegtmeyer, Steven A McCarroll, and Mehrtash Babadi. Explainable modeling of single-cell perturbation data using attention and sparse dictionary learning. *Cell Systems*, 16(4), 2025.

Ziqi Zhang, Xinye Zhao, Mehak Bindra, Peng Qiu, and Xiuwei Zhang. scdisinfact: disentangled learning for integration and prediction of multi-batch multi-condition single-cell rna-sequencing data. *Nature Communications*, 15(1):912, 2024.

A IMAGE DATASETS RESULTS

The main results relative to the experiment on `DF-MNIST` and `Morpho-MNIST` datasets are reported in Table 3. CVAE and DisCoVR baselines are the only that could be applied to images.

Table 3: Main comparison on image datasets (`DF-MNIST`, `Morpho-MNIST`). **Bold** and underline mark the best and second-best value per column within each dataset; splitDGD (sup-$\mathbf{w}$) is the only method that uses the $\mathbf{w}$-subtype label at training. Dashes (—) mark cases where the metric does not apply.

		Cond. $\mathbf{z}$-leakage ↓		$\mathbf{z}$-recovery ↑	$\mathbf{w}$-recovery ↑	Recon. ↓
	Method	AUC (lin)	AUC (NL)	$\mathbf{z}$-MCC	$\mathbf{w}$-MCC	RMSE
DF-MNIST	CVAE	0.909 ±.025	0.903 ±.012	**0.664** ±.016	—	**0.067** ±.004
	DisCoVR	0.650 ±.024	0.701 ±.019	0.632 ±.009	0.065 ±.015	0.089 ±.001
	splitDGD	0.519 ±.018	0.560 ±.029	0.620 ±.027	0.539 ±.112	0.090 ±.003
	splitDGD (sup-$\mathbf{w}$)	**0.502** ±.019	**0.538** ±.051	0.626 ±.031	**0.996** ±.008	0.089 ±.004
Morpho-MNIST	CVAE	0.651 ±.115	0.813 ±.035	**0.766** ±.020	—	**0.035** ±.003
	DisCoVR	0.590 ±.054	0.626 ±.036	0.747 ±.025	0.989 ±.005	0.076 ±.004
	splitDGD	0.516 ±.026	0.616 ±.046	0.722 ±.030	0.967 ±.038	0.078 ±.014
	splitDGD (sup-$\mathbf{w}$)	**0.514** ±.038	**0.542** ±.051	0.706 ±.030	**0.994** ±.019	0.077 ±.009

B DATASETS

The five datasets summarised in Table 1 of the main text are constructed as follows.

DigitFashion-MNIST. A 60 000-image synthetic construction built from the FashionMNIST and MNIST training sets (60 000 images each, via `torchvision.datasets`). We sample 60 000 FashionMNIST images without replacement and partition them into two disjoint pools of 30 000, assigned to $c = 0$ (baseline) and $c = 1$ (stim) respectively; for the stim pool we additionally sample 30 000 distinct MNIST overlays without replacement. The $c = 0$/$c = 1$ split is therefore $50/50$ by construction.

With $f = 0.4$, the baseline image is computed as $(1 - f)\,\mathbf{F}_0$ and the stim image as $\mathrm{clip}[(1 - f)\,\mathbf{F}_1 + f\,\mathbf{M}_1,\ 0,\ 1]$, where $\mathbf{F}_c$ is a FashionMNIST image from the c-pool and $\mathbf{M}_1$ is its assigned MNIST overlay. Scaling the baseline by the same factor $(1 - f)$ ensures the FashionMNIST contribution has identical intensity in both conditions, so the only systematic difference between $c = 0$ and $c = 1$ is the additive MNIST overlay rather than a brightness shift. The disjoint-pool construction prevents pixel-level memorisation of baseline/stim pairs, forcing generalisation across unseen FashionMNIST shapes.

The ground-truth $\mathbf{z}$-factor is the FashionMNIST class of the underlying garment (10 categories: T-shirt, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag, Ankle boot), defined for all samples; the ground-truth $\mathbf{w}$-factor is the overlaid MNIST digit class (10 categories, 0–9), defined only on $c = 1$ samples. Images are single-channel 28×28 tensors with pixel values in $[0, 1]$.

Morpho-MNIST. `Morpho-MNIST` (Castro et al., 2019) contains three stroke-stylised versions of every MNIST digit: *normal* (digits with unmodified stroke), *thick* (strokes thickened via parametric morphological perturbations on the digit skeleton), and *thin* (analogously thinned), for 60 000 images per variant and 180 000 in total.

We map *normal* to $c = 0$ and pool *thick* and *thin* together into $c = 1$, so baselines see normal-stroke digits and the perturbed condition consists of stylistic stroke modifications applied on top. Because two stim variants pool against one normal variant, $c = 1$ carries twice as many samples as $c = 0$ (120 000 vs 60 000); the stratified split preserves this 2:1 class imbalance across train, val, and test. This collapsed binary mapping differs from the original use of `Morpho-MNIST` in (Lotfollahi et al., 2020), which encoded the three styles as a 3-class condition with selected digits held out from the

perturbed strokes for out-of-distribution style transfer; our setup keeps all 10 digits available under both conditions so the disentanglement task is symmetric in the digit factor.

The ground-truth $\mathbf{z}$-factor is digit identity (10 categories, 0–9), defined for all samples; the ground-truth $\mathbf{w}$-factor is the stroke style (2 categories: thick, thin), defined only on $c = 1$ samples. Images are single-channel 28×28 tensors with pixel values rescaled to $[0, 1]$.

Kang PBMC IFN-β. The Kang et al. PBMC dataset (Kang et al., 2018) profiles peripheral blood mononuclear cells from eight systemic-lupus-erythematosus (SLE) patients, sequenced with multiplexed droplet scRNA-seq under two conditions: unstimulated control (CTRL) and 6-hour IFN-β stimulation (STIM). We use the same preprocessed AnnData as DisCoVR (Slavutsky et al., 2025), containing 13 999 cells (7 451 CTRL, 6 548 STIM) on 2 000 highly variable genes.

The data is already preprocessed: cell-type labels are consolidated to 10 categories, expression values are library-size normalised and log-transformed, and the feature space is restricted to the 2 000 most variable genes. At training time we additionally standardise each gene to zero mean and unit variance using statistics computed on the training partition only.

The ground-truth $\mathbf{z}$-factor is the cell-type label (10 categories: CD4 T, CD8 T, NK, B, CD14 Mono, CD16 Mono, DC, Eryth, Mk, T), defined for all cells. The dataset has no ground-truth $\mathbf{w}$-subtype within STIM—all stim cells are exposed to the same IFN-β stimulus—so `Kang` is excluded from supervised-$\mathbf{w}$ analyses and from the $\mathbf{w}$-MCC column.

Haber H.poly. The Haber et al. dataset (Haber et al., 2017) profiles 9 842 mouse small-intestinal epithelial cells measured under four experimental groups: healthy control, two time points of *Heligmosomoides polygyrus* infection (Day 3 and Day 10), and *Salmonella* infection. We start from the same raw-count AnnData released in (Lotfollahi et al., 2020), originally containing 15 215 genes. We apply library-size normalisation followed by a $\log(1 + x)$ transform, and retain the 2000 most variable genes. We collapse the three infected groups into c=1 so that the binary condition measures infected-vs.-healthy, and retain the four-way label as the $\mathbf{w}$-subtype factor (three subtypes within c=1: Hpoly.Day3, Hpoly.Day10, Salmonella). The $\mathbf{z}$-factor is the published cell-type label (8 epithelial cell types: Stem, TA, TA.Early, Endocrine, Enterocyte, Enterocyte.Progenitor, Goblet, Tuft), applicable to all cells. Together with `Sci-Plex3`, `Haber` is one of the two datasets in our suite with a multi-modal stim condition; with three subtypes, all observed in every cell type at training, it is the natural vehicle for the per-perturbation evaluation in Appendix I.

`Sci-Plex3`. sci-Plex 3 (Srivatsan et al., 2020) is a chemical-transcriptomics screen in which three human cancer cell lines (A549, lung adenocarcinoma; K562, chronic myelogenous leukaemia; MCF7, breast adenocarcinoma) were exposed for 24 h to 188 small molecules at four doses and profiled by single-nucleus RNA-seq with nuclear hashing, with DMSO vehicle controls on every plate. We use the curated object released with PerturBench (Wu et al., 2024), which retains the highest dose only (10 μM) and provides 178 213 cells on 8 630 genes, already library-size normalised to 10^4 counts and $\log(1 + x)$-transformed. We further drop the 17 884 cells profiled at 72 h, which come from a separate A549-only follow-up experiment on 47 compounds and would alias exposure time with cell line, leaving 160 329 cells; all 188 compounds, all three lines and all 564 (cell line, compound) pairs survive this filter. At training time we standardise each gene on training-partition statistics, as for `Kang` and `Haber`.

The binary condition separates vehicle (c=0; 10 524 cells, 6.6%) from compound-treated cells (c=1). The $\mathbf{z}$-factor is the cell line (3 categories) and the $\mathbf{w}$-factor is compound identity (188 categories, defined on c=1). The partition is PerturBench's covariate-transfer split, shipped with the object: within each cell line 30% of the compounds (57) are held out, giving 85 (cell line, compound) pairs for validation and 86 for testing, with validation and test compounds disjoint and every held-out compound observed at training in the other two lines (109 775 / 24 237 / 26 317 cells). Seed 0 uses the shipped partition; seeds 1–4 redraw the same design under the seed, so the reported standard deviations cover split as well as initialisation variance.

C HYPERPARAMETER OPTIMISATION

Optimisation setup. splitDGD hyperparameters are selected per seed by multi-objective optimisation with the NSGA-II sampler (Deb et al., 2002) in Optuna (Akiba et al., 2019), separately for the $\mathbf{w}$-unsupervised and the $\mathbf{w}$-supervised regimes. Each (dataset, model) pair is run for 10 independent seeds (5 on `Sci-Plex3`, with 60 trials each); each seed is a self-contained study that jointly optimises three objectives, all evaluated on a held-out validation fold of the training partition:

- $J_1 = |\mathrm{AUC}(c \mid \mathbf{z}) - 0.5|$, the centred condition AUC of a random-forest probe predicting the condition from $\mathbf{z}$; *minimised*.
- $J_2 = \mathrm{MCC}(\hat{y}_z, y_z)$, the random-forest Matthews correlation coefficient for predicting the invariant factor from $\mathbf{z}$; *maximised*.
- $J_3 = \mathrm{MCC}(\hat{y}_w, y_w \mid c{=}1)$, the same for the $\mathbf{w}$-subtype from $\mathbf{w}$ on perturbed samples; *maximised*. On `Kang` (no $\mathbf{w}$-subtype), J_3 is dropped. On `Sci-Plex3`, J_3 scores the 188 compounds with a random forest of maximum depth 8, so that the probe's capacity does not cap the score.

After all trials complete, we select a Pareto-front trial and *re-train* a fresh model on train+val with the selected hyperparameters; the test-fold metrics reported in the main text come from this retrained model. Trials within a study run sequentially; no early-stopping pruner is used; trials that diverge during training return a sentinel score and are filtered out before selection.

Search space. Table 4 lists the hyperparameters that vary per trial together with their per-dataset ranges. Learning rates are log-uniform; the latent dimensions n_z, n_w are integer with step 4; β_w is categorical; n_{epochs} is integer with step 50. FiLM, Feature-wise Linear Modulation (Perez et al., 2018), modulates intermediate decoder activations by a learned affine transform conditioned on $\mathbf{w}$ only; it is disabled on the tabular datasets (`Kang`, `Haber`, `Sci-Plex3`). Hyperparameters not listed (mixture-component counts K_z, K_w, hidden widths, batch size, output likelihood, decoder family, feature standardisation) are fixed per dataset; their values are reported in Table 5.

Table 4: splitDGD search space per dataset. Learning rates are log-uniform; n_z, n_w are integer with step 4; β_w is categorical, except on `Sci-Plex3` where it is log-uniform over $[1, 10]$; n_{epochs} is integer with step 50. η_{rep} is sampled once and applied to both the train and test representation optimisers.

Hyperparameter	`DF-MNIST`	`Morpho-MNIST`	`Kang`	`Haber`	`Sci-Plex3`
n_z	$\{8, 12, 16\}$	$\{8, 12, 16\}$	$\{8, 12, 16\}$	$\{8, 12, 16\}$	$\{8, 12, 16\}$
n_w	$\{8, 12, 16\}$	$\{8, 12, 16\}$	$\{8, 12, 16\}$	$\{8, 12\}$	$\{8, 12, 16\}$
η_{decoder}	$[10^{-4}, 5{\cdot}10^{-3}]$	$[10^{-4}, 2{\cdot}10^{-3}]$	$[10^{-4}, 2{\cdot}10^{-3}]$	$[3{\cdot}10^{-5}, 2{\cdot}10^{-3}]$	$[3{\cdot}10^{-5}, 2{\cdot}10^{-3}]$
η_{prior}	$[10^{-4}, 5{\cdot}10^{-3}]$	$[2{\cdot}10^{-4}, 2{\cdot}10^{-3}]$	$[10^{-4}, 2{\cdot}10^{-3}]$	$[10^{-4}, 2{\cdot}10^{-3}]$	$[10^{-4}, 2{\cdot}10^{-3}]$
η_{rep}	$[5{\cdot}10^{-3}, 2{\cdot}10^{-2}]$	$[5{\cdot}10^{-3}, 2{\cdot}10^{-2}]$	$[5{\cdot}10^{-3}, 2{\cdot}10^{-2}]$	$[5{\cdot}10^{-3}, 2{\cdot}10^{-2}]$	$[5{\cdot}10^{-3}, 3{\cdot}10^{-2}]$
β_w	$\{0.5, 1, 1.5\}$	$\{0.5, 1, 1.5\}$	$\{0.5, 1, 1.5\}$	$\{0.5, 1, 1.5\}$	$[4, 10]$
n_{epochs}	$\{200, \ldots, 600\}$	$\{100, \ldots, 400\}$	$\{150, \ldots, 400\}$	$\{250, \ldots, 600\}$	$\{250, \ldots, 600\}$
FiLM	{on, off}	{on, off}	off	off	off
Trials per seed	100	100	200	200	60

Reproducibility. The NSGA-II sampler, the dataset split, and the per-trial RNG are seeded deterministically from the seed index, so each seed's study is independently reproducible. The code to reproduce the experiments and the used datasets can be found at https://anonymous.4open.science/r/splitdgd-5570/.

D BASELINES

This appendix gives the full configuration of each baseline method compared against splitDGD in the main text. CVAE and DisCoVR run on all five datasets; contrastiveVI and scDisInFact are single-cell-only and run on `Kang`, `Haber` and `Sci-Plex3`. All four baselines reuse splitDGD's split seeds and partitions (10 stratified splits on `DF-MNIST`, `Morpho-MNIST`, `Kang` and `Haber`;

Table 5: splitDGD fixed (non-searched) hyperparameters per dataset. K_z and K_w are the GMM mixture-component counts on the invariant and condition-specific latent blocks, respectively. The Dirichlet α on the GMM-weight prior is 0.3 for both blocks on every dataset except `Sci-Plex3`, where it is 1.0: controls are only 6.7% of its cells, so the $\mathbf{z}$-mixture is fitted from few cells per batch and the $\alpha < 1$ hyper-prior term, unbounded below, destabilised training; dropout is disabled; the output Gaussian likelihood has a learned per-feature log-variance initialised at 0. The full parametrisation of the latent priors and their Bayesian hyperpriors is given in Appendix G.

	`DF-MNIST`	`Morpho-MNIST`	`Kang`	`Haber`	`Sci-Plex3`
Decoder	CNN	CNN	MLP	MLP	MLP
Output likelihood	Bernoulli	Bernoulli	Gaussian	Gaussian	Gaussian
Hidden widths	$(64, 512)$	$(64, 512)$	$(64, 512)$	$(128, 512)$	$(128, 512)$
K_z	10	10	10	10	3
K_w	10	2	1	3	188
Batch size	2048	4096	256	256	4096
Feature standardisation	no	no	yes	yes	yes

the 5 covariate-transfer splits on `Sci-Plex3`), so every reported score is on the same data. Every baseline is tuned per seed by the same Optuna NSGA-II multi-objective search used for splitDGD over the method-specific spaces given below, and then retrained on the full training partition: 200 trials per seed for contrastiveVI, DisCoVR and scDisInFact on `Kang` and `Haber`, and 60 for every model on `Sci-Plex3`. Model code follows the authors' reference implementations.

CVAE (Sohn et al., 2015). The conditional VAE is our no-disentanglement baseline: a standard VAE whose encoder $q(\mathbf{z} \mid \mathbf{x}, c)$ and decoder $p(\mathbf{x} \mid \mathbf{z}, c)$ both receive a one-hot encoding of the condition c concatenated to their inputs, trained against the ELBO with a fixed $\mathcal{N}(\mathbf{0}, \mathbf{I})$ latent prior. There is no structural mechanism enforcing that $\mathbf{z}$ be condition-invariant: the condition is available to the decoder directly, so the model can freely leave condition information in $\mathbf{z}$ whenever this helps reconstruction. We use a PyTorch implementation with hidden widths $(64, 512)$ on images datasets and on `Kang` and $(128, 512)$ on `Haber`, matching splitDGD's per-dataset architecture; the latent dimension, β, learning rate, and dropout are selected per seed by the NSGA-II search described above. Training uses Adam with early stopping on validation loss. Because CVAE has no $\mathbf{w}$-block, its full latent is reported as $\mathbf{z}$ in all tables and figures ($\mathbf{w}$-MCC is left blank).

contrastiveVI (Weinberger et al., 2023). contrastiveVI is a single-cell-only baseline built for *contrastive analysis*: the data are split into a *background* population (here the control cells) and a *target* population (the perturbed cells), and the model learns a shared latent $\mathbf{z}$, capturing variation common to both populations, alongside a *salient* latent $\mathbf{s}$ reserved for variation specific to the target population. A shared encoder $q(\mathbf{z} \mid \mathbf{x})$ is applied to every cell; a second encoder $q(\mathbf{s} \mid \mathbf{x})$ is applied to target cells only, and for background cells $\mathbf{s}$ is fixed to the zero vector before decoding. Both populations are reconstructed by a single decoder from $[\mathbf{z}; \mathbf{s}]$ and the observed library size under a zero-inflated negative-binomial likelihood on raw counts; the objective is the sum of the two populations' ELBOs (a KL term on $\mathbf{z}$ for all cells and on $\mathbf{s}$ for target cells), optionally augmented by a Wasserstein penalty $\lambda_{\mathrm{W}}\,(\|\boldsymbol{\mu}_{\mathbf{s}}\|_2^2 + \sum_k \sigma_{\mathbf{s},k}^2)$ on the salient posterior of background cells—the squared 2-Wasserstein distance to a point mass at $\mathbf{0}$—which further discourages shared variation from leaking into $\mathbf{s}$. Disentanglement is therefore architectural rather than adversarial or classifier-driven: nothing pushes $\mathbf{z}$ away from the condition, and the perturbation block is shaped only by being switched off on controls. We map $\mathbf{z}$ to our $\mathbf{z}$ and $\mathbf{s}$ to our $\mathbf{w}$. We use the `scvi-tools` implementation (`scvi.external.ContrastiveVI`), trained on raw counts with control cells as background and perturbed cells as target; only the binary condition is consumed, so contrastiveVI sits in the unsupervised-$\mathbf{w}$ band on every dataset, and test cells are encoded but never enter training. Per seed, NSGA-II selects the background and salient dimensions ($\{8, 10, 16, 32\}$ and $\{4, 8, 10, 16\}$), hidden width ($\{128, 256\}$), depth ($\{1, 2\}$), dropout ($\{0, 0.1\}$) and $\lambda_{\mathrm{W}} \in \{0, 1, 10, 50\}$—a space that contains the authors' defaults (10/10, 128, 1, 0.1, 0)—with a fixed 200-epoch schedule, batch size 128 and no early stopping. Predictions are converted to log-normalised space before RMSE / counterfactual-R^2 evaluation, matching the comparison space used for the other count-based baselines.

scDisInFact (Zhang et al., 2024). scDisInFact is a single-cell-only baseline with a comparable $\mathbf{z}/\mathbf{w}$-style split: a shared biological factor $\mathbf{z}_s$ (condition-invariant, mapped to $\mathbf{z}$), one unshared factor $\mathbf{z}_u$ per condition type (condition-specific, mapped to $\mathbf{w}$, with one encoder per perturbation axis), and an optional batch embedding that is absent in our setup. It is trained against a negative-binomial likelihood directly on raw counts, with no internal log-transform or standardisation; predictions are converted to log-normalised space before RMSE / counterfactual-R^2 evaluation, matching the comparison space used for the other baselines. Disentanglement is driven by three terms: MMD penalties applied to both $\mathbf{z}_s$ (across conditions, encouraging condition-invariance) and $\mathbf{z}_u$ (across batches within each condition); a classification loss that forces $\mathbf{z}_u$ to be condition-predictive; and a group-lasso penalty on the first layer of each unshared encoder, used for condition-key-gene selection. Because none of our datasets carry a batch label, the batch-axis MMD on $\mathbf{z}_u$ is inert at training time. We use the authors' package. Per seed, NSGA-II selects the unshared dimension $K_u \in \{2, 3, 4\}$ with $K_s = 8$ (the authors' recommended region), the learning rate ($[10^{-4}, 10^{-3}]$), the two MMD weights ($[10^{-5}, 10^{-3}]$ each), the group-lasso weight and the classification weight ($[0.1, 10]$ each), all log-uniform; the KL weights $\lambda^1_{\mathrm{KL}} = 10^{-5}$ and $\lambda^2_{\mathrm{KL}} = 10^{-2}$, the batch size (64; matched to splitDGD on `Sci-Plex3`) and the 100-epoch schedule are fixed. Because the classification term explicitly shapes $\mathbf{z}_u$ around condition identity during training, scDisInFact sits in the supervised-$\mathbf{w}$ band on `Haber`. On `Sci-Plex3` it is trained on the binary condition instead, because its data loader forms one group per (batch, condition) combination and the 189-way encoding is infeasible at that scale; its `Sci-Plex3` entries are therefore unsupervised-$\mathbf{w}$ results.

DisCoVR (Slavutsky et al., 2025). DisCoVR uses an explicit $\mathbf{z}/\mathbf{w}$ split: two encoders—$q(\mathbf{z} \mid \mathbf{x})$ (condition-invariant) and $q(\mathbf{w} \mid \mathbf{x}, c)$ (condition-specific)—together with two decoders, a joint decoder from $[\mathbf{z}; \mathbf{w}]$ and an auxiliary $\mathbf{z}$-only decoder producing $\hat{\mathbf{x}}_z$. Disentanglement is enforced through a min–max classifier cross-entropy: an auxiliary classifier $g(c \mid \hat{\mathbf{x}}_z)$ is trained to recover c from the $\mathbf{z}$-only reconstruction, while the encoder is trained to maximise that cross-entropy, so the $\mathbf{z}$-only reconstruction retains as little condition information as the classifier can detect. Both reconstructions contribute to the ELBO, and training runs through Pyro stochastic variational inference. The $\mathbf{w}$-prior is a learned class-conditional Gaussian whose mean and variance are produced by a small MLP applied to the condition one-hot. We use the authors' original code with their paper defaults. For single-cell data we use the ZINBDLVAE variant—a zero-inflated negative-binomial likelihood applied directly to raw counts—with a linear encoder/decoder (`num_layers`$= 0$, `hidden_dim`$= 128$), $d_z = d_w = 10$, joint-reconstruction weight 0.9, $\mathbf{z}$-only weight 0.1, adversarial weight 100, KL weights 10^{-4}; predictions are converted to log-normalised space before RMSE / counterfactual-R^2 evaluation, ensuring a uniform comparison space across baselines. For images we use $d_z = 20$, $d_w = 2$, a `conv+pool` CNN architecture (32-64-128 channels), adversarial weight 0.1, 200 epochs. DisCoVR uses only the binary condition label, so it enters the unsupervised-$\mathbf{w}$ band regardless of dataset.

Counterfactual generation. For each control cell i and target perturbation p, the four baselines produce a counterfactual prediction $\hat{\mathbf{x}}_i^p$ as follows. **CVAE** encodes i with c=0 and decodes the resulting $\mathbf{z}_i$ with the binary stim flag c=1; CVAE never observes a multi-class condition, so the same prediction serves every perturbation and is sliced post-hoc by the ground-truth perturbation label. **contrastiveVI** holds $\mathbf{z}_i$ fixed and replaces $\mathbf{s}_i$ with the mean salient latent over stim cells of the same cell type as i, then decodes through the zero-inflated negative-binomial head and converts the resulting rate to log-normalised space; like DisCoVR it consumes only the binary condition, so the cell-type-mean $\mathbf{s}$ is the only mechanism by which its counterfactual differentiates perturbations. **DisCoVR** holds $\mathbf{z}_i$ from i fixed and replaces $\mathbf{w}_i$ with the mean $\mathbf{w}$ over stim cells of the same cell type as i; since DisCoVR consumes only the binary condition, the cell-type-mean $\mathbf{w}$ is the only mechanism by which the counterfactual differentiates perturbations. **scDisInFact** holds $\mathbf{z}_s^{(i)}$ fixed and replaces $\mathbf{z}_u^{(i)}$ with the mean $\mathbf{z}_u$ over stim cells of the same cell type, then decodes through the negative-binomial head and converts the result to log-normalised space.

On `Sci-Plex3`, the same transfer rule is applied to every baseline with a condition-specific latent: the cell-type mean is replaced by the compound's mean latent over its training cells in the other two cell lines, so no method sees the held-out pair.

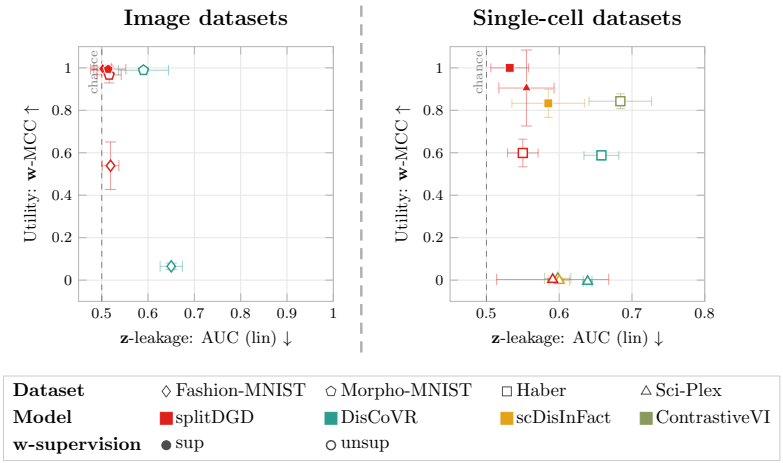


Figure 3: **Trade-off between C1 and C3**. Each point is one (method, dataset) pair: condition leakage from $\mathbf{z}$ (test AUC of a linear probe; 0.5 is chance, dashed line) against $\mathbf{w}$-MCC. Colour encodes the method, the marker the dataset, filled markers the supervised-$\mathbf{w}$ regime; bars are $\pm$ one std over split seeds. CVAE (no $\mathbf{w}$-block) and `Kang` (no $\mathbf{w}$-subtype) do not appear. Values are from Tables 2 and 3.

E TRADE-OFF ANALYSIS FOR C1-VS-C3

Figure 3 repeats the trade-off view of Figure 2 with $\mathbf{w}$-MCC (**C3**) as the utility axis; CVAE has no $\mathbf{w}$-block and `Kang` no $\mathbf{w}$-subtype, so both are absent. The picture matches the $\mathbf{z}$-MCC panels: on every dataset the left-most point is a splitDGD variant, splitDGD (sup-$\mathbf{w}$) occupies the top-left corner on all four datasets, and no baseline improves on splitDGD along both axes. The two cases in which a baseline recovers more perturbation structure, DisCoVR on `Morpho-MNIST` (0.989 vs. 0.967) and contrastiveVI on `Haber` (0.843 vs. 0.599), both come with higher leakage; DisCoVR collapses on `DF-MNIST` (0.065), and on `Sci-Plex3` every unsupervised method lies at chance, so only the supervised variant leaves the bottom of the panel.

F ROBUSTNESS ANALYSIS: CATASTROPHIC FAILURES

We analyze, per method and per metric, the number of (dataset, supervision-regime) blocks on which a method exhibits a *catastrophic failure*. We define a catastrophic failure on a block as a per-block mean score worse than the block's leader by more than a metric-specific tolerance τ_M: $\tau_{\text{AUC}} = \tau_{\text{MCC}} = 0.10$ on the bounded $[0, 1]$ metrics, and $\tau_{\text{RMSE}} = \tau_{\text{CF-}R^2} = 0.05$ on the smaller-scale fidelity metrics. Within a block, each method is compared with the leader among the methods that use no more label information than it does: unsupervised methods against the unsupervised regime of the dataset, and supervised-$\mathbf{w}$ methods against every method on that dataset. These thresholds are conservative relative to the seed-to-seed standard deviations reported in Tables 2 and 3 (typically below 0.05), so each catastrophic event reflects a gap not attributable to run-to-run variance.

Table 6 reports counts a/b, where b is the number of (dataset, regime) blocks on which the method is evaluated for that metric and a is the number of those blocks on which the method is catastrophic. splitDGD has a single event out of 27 cells: $\mathbf{w}$-MCC on `Haber` unsupervised, where contrastiveVI better recovers the infection subtypes (0.843 vs. 0.599). splitDGD (sup-$\mathbf{w}$) has none out of 22. Every baseline has at least four: CVAE $10/27$, DisCoVR $10/27$, scDisInFact $4/17$ and contrastiveVI $4/17$. `Sci-Plex3` contributes three of these events, all on fidelity: DisCoVR on RMSE (0.205 vs. 0.149) and on counterfactual R^2 (0.847 vs. 0.954), and scDisInFact on counterfactual R^2 (0.872 vs. 0.954); no method fails on leakage or on $\mathbf{z}$-MCC there. The conclusion is robust to the threshold: tightening τ_{RMSE} to 0.04 adds one borderline event on `Morpho-MNIST` to splitDGD (gap 0.043), to splitDGD (sup-$\mathbf{w}$) (0.042) and to DisCoVR (0.041); halving every tolerance leaves splitDGD at $3/27$ and splitDGD (sup-$\mathbf{w}$) at $2/22$, against $6/17$ for the closest baseline (scDisInFact) and $17/27$ for DisCoVR.

Table 6: Catastrophic-failure counts per method, per metric. Counts a/b: a catastrophic events out of b evaluated (dataset, regime) blocks. Leaders are taken per block among the methods using no more label information than the method scored (see text). CVAE, which has no $\mathbf{w}$-block, is scored at the worst possible $\mathbf{w}$-MCC: on a dataset with perturbation substructure it recovers none of it. $\mathbf{w}$-MCC on `Kang` and counterfactual R^2 on the image datasets are undefined and enter no denominator.

Method	AUC (lin)	AUC (NL)	$\mathbf{z}$-MCC	$\mathbf{w}$-MCC	RMSE	CF R^2	Total
CVAE	$4/5$	$3/5$	$0/5$	$3/4$	$0/5$	$0/3$	$10/27$
DisCoVR	$3/5$	$1/5$	$0/5$	$2/4$	$1/5$	$3/3$	$10/27$
scDisInFact	$0/3$	$0/3$	$0/3$	$1/2$	$0/3$	$3/3$	$4/17$
ContrastiveVI	$2/3$	$0/3$	$0/3$	$0/2$	$0/3$	$2/3$	$4/17$
splitDGD	$\mathbf{0/5}$	$\mathbf{0/5}$	$\mathbf{0/5}$	$\mathbf{1/4}$	$\mathbf{0/5}$	$\mathbf{0/3}$	$\mathbf{1/27}$
splitDGD (sup-$\mathbf{w}$)	$\mathbf{0/1}$	$\mathbf{0/1}$	$\mathbf{0/1}$	$\mathbf{0/1}$	$\mathbf{0/1}$	$\mathbf{0/1}$	$\mathbf{0/6}$

G Learnable Gaussian-mixture latent distributions and Bayesian hyperpriors

Existing variational disentanglement methods almost universally use standard normal priors $p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ (Lotfollahi et al., 2023; Zhang et al., 2024; Slavutsky et al., 2025). As noted in Section 2, the DGD admits any differentiable parametric distribution without modification; we exploit this by using learnable Gaussian mixtures for both blocks, with Bayesian priors on the component parameters.

Structural roles of the two distributions. p_{ϕ_z} has K_z components and p_{ϕ_w} has K_w components. Despite having the same mathematical form, the two distributions play distinct roles at training time. Through Mechanism 2 (Section 3.1), ϕ_z is updated only from baseline-sample embeddings, so the $\mathbf{z}$-mixture captures the cluster structure of the unperturbed population. The $\mathbf{w}$-mixture is fit from perturbed samples alone: the hard mask of Mechanism 1 freezes $\mathbf{w}_i = \mathbf{0}$ for baselines, so their contribution to the $\mathbf{w}$-likelihood sum is removed by construction. Both mixtures are used within an architecture that also accepts subtype labels structurally, via the single-component density of Eq. 8.

Log-density and parametrisation. Both priors use diagonal covariances, and their log-densities are computed using the numerically stable log-sum-exp form; for p_{ϕ_z},

$$\log p_{\phi_z}(\mathbf{z}) \;=\; \operatorname*{logsumexp}_{k=1,\dots,K_z}\left[\log \pi_k^z \;-\; \tfrac{d_z}{2}\log(2\pi) \;-\; \tfrac{1}{2}\sum_{d=1}^{d_z}\left(\log(\sigma_{kd}^z)^2 + \frac{(z_d-\mu_{kd}^z)^2}{(\sigma_{kd}^z)^2}\right)\right], \quad (10)$$

and the same expression holds for p_{ϕ_w} with (K_w, ϕ_w, d_w) in place of (K_z, ϕ_z, d_z). The component parameters $\phi_z = \{\boldsymbol{\mu}_k^z, \boldsymbol{\Sigma}_k^z, \pi_k^z\}_{k=1}^{K_z}$ defined in Eq. 6 are themselves given Bayesian hyperpriors. For unconstrained gradient-based optimisation we reparametrise each $\boldsymbol{\Sigma}_k^z$ via the log per-dimension variances $\log(\boldsymbol{\sigma}_k^z)^2 \in \mathbb{R}^{d_z}$ and $\boldsymbol{\pi}^z = (\pi_1^z, \dots, \pi_{K_z}^z)$ via softmax of an unconstrained logit vector $\boldsymbol{\eta}^z \in \mathbb{R}^{K_z}$, $\pi_k^z = \mathrm{softmax}(\boldsymbol{\eta}^z)_k$.

Bayesian hyperpriors on the component parameters. The hyperpriors follow standard DGD practice (Schuster & Krogh, 2023):

$$p(\boldsymbol{\mu}_k^z) \;=\; \mathrm{Softball}(r, s), \quad p(\log(\boldsymbol{\sigma}_k^z)^2) \;=\; \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad (\pi_1^z, \dots, \pi_{K_z}^z) \;\sim\; \mathrm{Dirichlet}(\alpha\mathbf{1}). \quad (11)$$

The Softball prior is an almost-uniform distribution inside a ball of radius r with a soft logistic boundary of sharpness s; it keeps mixture means in a bounded region without hard constraints. The Gaussian prior on log-variances and the symmetric Dirichlet on mixture weights provide mild regularisation that prevents the classic GMM pathologies of variance collapse and empty components. The same hyperprior structure is used for p_{ϕ_w} with its own hyperparameters. These hyperprior terms are included inside $\mathcal{L}_Z$ as $-\log p(\phi_z)$ (Eq. 6) and inside $\mathcal{L}_W$ as $-\log p(\phi_w)$ (Eq. 7).

Table 7: Per-perturbation counterfactual R^2 on `Haber`, mean $\pm$ std over 10 split seeds. Methods are grouped by $\mathbf{w}$-supervision; **bold** marks the best value within each supervision subgroup, underline the second-best. The *Mean* column is the per-seed average across the three perturbations (the `Haber` CF R^2 value reported in Table 2).

Method	Hpoly.Day3	Hpoly.Day10	Salmonella	*Mean*
Unsupervised $\mathbf{w}$				
CVAE	**0.917** $\pm$.016	0.894 $\pm$.006	**0.881** $\pm$.015	**0.897** $\pm$.006
DisCoVR	0.817 $\pm$.017	0.858 $\pm$.027	0.818 $\pm$.024	0.831 $\pm$.020
ContrastiveVI	0.822 $\pm$.017	0.835 $\pm$.014	0.799 $\pm$.026	0.819 $\pm$.016
splitDGD	0.884 $\pm$.049	**0.915** $\pm$.031	0.855 $\pm$.057	0.885 $\pm$.044
Supervised $\mathbf{w}$				
scDisInFact	0.810 $\pm$.020	0.837 $\pm$.010	0.824 $\pm$.019	0.824 $\pm$.013
splitDGD (sup-$\mathbf{w}$)	**0.935** $\pm$.018	**0.950** $\pm$.008	**0.893** $\pm$.026	**0.926** $\pm$.009

H Four-optimiser training scheme

The total loss in Eq. 9 has four disjoint groups of parameters that benefit from different learning rates: the prior parameters $\phi = (\phi_z, \phi_w)$, the decoder weights θ, the train-set representations $(\mathbf{Z}_{\text{train}}, \mathbf{W}_{\text{train}})$, and the test-set representations $(\mathbf{Z}_{\text{test}}, \mathbf{W}_{\text{test}})$. We maintain four separate Adam optimisers with learning rates $\eta_\phi, \eta_\theta, \eta_{\text{rep,tr}}, \eta_{\text{rep,te}}$ respectively. Each training epoch alternates between a train phase—in which θ, ϕ, and $(\mathbf{Z}_{\text{train}}, \mathbf{W}_{\text{train}})$ are updated against train batches (using the first three optimisers)—and a test phase—in which θ and ϕ are frozen and only $(\mathbf{Z}_{\text{test}}, \mathbf{W}_{\text{test}})$ are updated against test batches (using the fourth). The representation optimisers step once per epoch (deferred accumulation) rather than per batch: gradients from every batch in the epoch are accumulated into the representation table before the single Adam step is applied.

I Counterfactual R^2 breakdowns

This appendix defines the counterfactual R^2 procedure and reports stratifications along three axes: per perturbation on `Haber`, per cell type and per gene cut (top-N DEGs) on both `Haber` and `Kang`. For `Sci-Plex3`, whose covariate-transfer split holds out each drug in a single cell line, it reports the per-cell-line and top-N DEG breakdowns under the transfer protocol.

Procedure. Following the per-perturbation evaluation of (Lotfollahi et al., 2019; 2020), for each perturbation p and each control cell i the model produces a prediction $\hat{\mathbf{x}}_i^p$ of the cell's expression under p, the *counterfactual*. For splitDGD, $\hat{\mathbf{x}}_i^p$ is the decoder mean for $[\mathbf{z}_i; \bar{\mathbf{w}}_p^t]$, where $\bar{\mathbf{w}}_p^t$ is the mean $\mathbf{w}$-embedding of the perturbed cells of the same cell type t as i that received p; every method is decoded with its likelihood mean, and the baselines' analogues of this rule are given in Appendix D. On `Kang` and `Haber`, $\bar{\mathbf{w}}_p^t$ is taken over the perturbed cells of the test partition, which is adequate because every (cell type, perturbation) pair is observed at training; on `Sci-Plex3`, where the scored pair is unseen, it is taken over the compound's training cells in the other two cell lines (the *transfer* protocol below), and the control cells of the target line are a random subset of at most 400. For each pair (perturbation p, cell type t) we then compute two gene-length vectors: $\hat{\boldsymbol{\mu}}^{p,t}$, the mean prediction over the control cells of type t, and $\boldsymbol{\mu}^{p,t}$, the mean expression of the perturbed cells of type t that received p. The per-perturbation R_p^2 is the squared Pearson correlation between $\hat{\boldsymbol{\mu}}^{p,\cdot}$ and $\boldsymbol{\mu}^{p,\cdot} \in \mathbb{R}^{n_{\text{types}} \times n_{\text{genes}}}$ flattened to vectors, and the headline value in Table 2 is the unweighted average of $\{R_p^2\}_p$: three perturbations on `Haber`, one on `Kang`, and the held-out (cell line, compound) pairs on `Sci-Plex3`, where $n_{\text{types}} = 1$. The breakdowns below are stratifications of this same metric.

Per-perturbation, `Haber`. Within the unsupervised band, splitDGD and CVAE are within 0.04 of each other on every perturbation (Table 7): CVAE leads on Hpoly.Day3, Salmonella, and the mean, and splitDGD leads on Hpoly.Day10. DisCoVR and ContrastiveVI sit 0.06–0.10 below the leader on every perturbation. In the supervised band, splitDGD (sup-$\mathbf{w}$) leads scDisInFact by 0.07–0.13 on every perturbation and is the highest-scoring method overall on all three.

Table 8: Per-cell-type counterfactual R^2 on `Haber`, mean $\pm$ std over 10 split seeds. Methods are grouped by $\mathbf{w}$-supervision; **bold** marks the best value within each supervision subgroup, underline the second-best. "Entero.Prog" = Enterocyte.Progenitor.

Method	Stem	TA	TA.Early	Endocrine	Enterocyte	Entero.Prog	Goblet	Tuft
Unsupervised $\mathbf{w}$								
CVAE	0.961 ±.013	**0.969** ±.009	**0.985** ±.004	**0.886** ±.047	**0.987** ±.004	**0.980** ±.006	0.972 ±.013	**0.961** ±.010
DisCoVR	0.948 ±.011	0.934 ±.012	0.966 ±.009	0.859 ±.087	0.971 ±.010	0.967 ±.010	0.963 ±.029	0.942 ±.021
ContrastiveVI	0.887 ±.029	0.890 ±.033	0.939 ±.021	0.779 ±.072	0.955 ±.008	0.959 ±.020	0.941 ±.011	0.935 ±.017
splitDGD	**0.967** ±.029	0.965 ±.032	0.971 ±.033	0.796 ±.098	0.978 ±.020	0.976 ±.024	**0.985** ±.006	0.918 ±.053
Supervised $\mathbf{w}$								
scDisInFact	0.976 ±.005	0.972 ±.005	0.975 ±.006	0.712 ±.068	0.836 ±.008	0.962 ±.009	0.855 ±.025	0.931 ±.010
splitDGD (sup-$\mathbf{w}$)	**0.979** ±.010	**0.976** ±.007	**0.979** ±.011	**0.891** ±.033	**0.975** ±.009	**0.988** ±.007	**0.988** ±.003	**0.965** ±.012

Table 9: Per-cell-type counterfactual R^2 on `Kang`, mean $\pm$ std over 10 split seeds. `Kang` has no $\mathbf{w}$-subtype label, so all methods are scored in a single supervision band; **bold** marks the best value per cell type, underline the second-best.

Method	B	CD14 Mono	CD16 Mono	CD4 T	CD8 T	DC	Eryth	Mk	NK	T
CVAE	0.973 ±.006	0.954 ±.027	0.971 ±.008	0.986 ±.003	0.966 ±.008	0.957 ±.016	**0.897** ±.040	0.913 ±.021	0.936 ±.019	0.939 ±.014
DisCoVR	0.947 ±.007	0.973 ±.025	0.973 ±.020	0.938 ±.006	0.929 ±.004	0.962 ±.027	0.757 ±.079	0.874 ±.019	0.937 ±.008	0.931 ±.011
scDisInFact	0.980 ±.002	0.927 ±.003	0.921 ±.005	0.988 ±.002	0.961 ±.009	0.943 ±.008	0.573 ±.082	0.884 ±.043	0.948 ±.005	0.945 ±.015
ContrastiveVI	0.940 ±.017	0.972 ±.005	0.965 ±.009	0.941 ±.018	0.920 ±.014	0.964 ±.012	0.740 ±.073	0.901 ±.026	0.935 ±.016	0.918 ±.013
splitDGD	**0.983** ±.007	**0.986** ±.012	**0.988** ±.005	**0.989** ±.008	**0.970** ±.009	**0.973** ±.011	0.885 ±.055	**0.925** ±.016	**0.961** ±.010	**0.960** ±.009

Per-cell-type, `Haber`. Counterfactual quality varies across the eight epithelial types (Table 8): Endocrine cells are the lowest-scoring type for every method, with R^2 in the 0.71–0.89 range. In the supervised band, splitDGD (sup-$\mathbf{w}$) leads on all eight cell types; scDisInFact is competitive on Stem and the two TA subtypes but drops on Endocrine, Enterocyte, and Goblet. In the unsupervised band, CVAE leads on six of eight cell types and splitDGD on the remaining two (Stem, Goblet); ContrastiveVI is lowest on seven of eight.

Per-cell-type, `Kang`. `Kang` has no $\mathbf{w}$-subtype, so all methods are scored in a single band in Table 9. splitDGD leads on nine of ten cell types, consistent with its top CF R^2 in Table 2; the exception is Erythroid (Eryth) cells, where CVAE leads (0.897 vs. 0.885). Eryth is the lowest-scoring cell type for every method, and the $\mathbf{z}/\mathbf{w}$-split baselines drop furthest there: DisCoVR to $R^2 \approx 0.76$, ContrastiveVI to ≈ 0.74, and scDisInFact to ≈ 0.57, against 0.89–0.90 for splitDGD and CVAE.

DEG-restricted top-N, `Haber`. DEGs (differentially expressed genes) are the top-N genes ranked by $|\bar{\mathbf{x}}_{\text{stim}} - \bar{\mathbf{x}}_{\text{ctrl}}|$, computed within each perturbation. The DEG-restricted R^2 uses the same per-cell-type-mean comparison as the all-genes R^2 in Table 2 but restricts the gene axis to the top-N DEGs, scoring how well predictions match the perturbation signal at the most-affected genes.

For `Haber`, reported values (Tables 10) average the per-perturbation top-N R^2s, matching the headline aggregation. In the supervised band, splitDGD (sup-$\mathbf{w}$) leads on every N (0.926–0.931, against 0.81–0.82 for scDisInFact). In the unsupervised band, splitDGD and CVAE trade leads as N grows: splitDGD leads at top-50, the two tie at top-100, and CVAE leads at top-200; ContrastiveVI and DisCoVR sit below both.

`Kang` has a single perturbation, so per-perturbation averaging does not apply: each method contributes one top-N R^2 value per seed. splitDGD leads on every N with CVAE second; ContrastiveVI, DisCoVR, and scDisInFact rank further below (Table 11).

`Sci-Plex3`: protocol. `Sci-Plex3` uses a covariate-transfer split in which every test drug is held out in exactly one cell line, so the per-perturbation flattening over cell types used for `Haber` reduces to a single (cell line, drug) pair. We therefore report the per-pair counterfactual R^2 under the *transfer* protocol: the drug's $\mathbf{w}$ is the mean over its training cells in the other two lines, and the target line's control cells are decoded with it. Each column averages the per-pair R^2 over the pairs it contains. Two reference rows calibrate the scale: *Control* predicts the line's control mean, i.e. no drug effect, and *Ceiling* predicts one half of each pair's observed cells from the other half.

Table 10: DEG-restricted counterfactual R^2 on `Haber`, mean $\pm$ std over 10 split seeds, averaged across the three per-perturbation R^2s. Methods are grouped by $\mathbf{w}$-supervision; **bold** = best within each subgroup, underline = second-best.

Method	top-50	top-100	top-200
Unsupervised $\mathbf{w}$			
CVAE	0.842 $\pm$.005	**0.872** $\pm$.005	**0.884** $\pm$.005
DisCoVR	0.798 $\pm$.019	0.811 $\pm$.021	0.817 $\pm$.021
ContrastiveVI	0.827 $\pm$.014	0.834 $\pm$.012	0.828 $\pm$.013
splitDGD	**0.855** $\pm$.058	**0.872** $\pm$.058	0.878 $\pm$.053
Supervised $\mathbf{w}$			
scDisInFact	0.813 $\pm$.022	0.817 $\pm$.021	0.819 $\pm$.018
splitDGD (sup-$\mathbf{w}$)	**0.926** $\pm$.010	**0.931** $\pm$.011	**0.931** $\pm$.009

Table 11: DEG-restricted counterfactual R^2 on `Kang` (single-perturbation dataset, IFN-β), mean $\pm$ std over 10 split seeds. **Bold** = best, underline = second-best.

Method	top-50	top-100	top-200
CVAE	0.932 $\pm$.020	0.938 $\pm$.018	0.941 $\pm$.016
DisCoVR	0.918 $\pm$.015	0.916 $\pm$.017	0.909 $\pm$.018
scDisInFact	0.891 $\pm$.015	0.889 $\pm$.013	0.879 $\pm$.011
ContrastiveVI	0.923 $\pm$.015	0.922 $\pm$.013	0.919 $\pm$.012
splitDGD	**0.967** $\pm$.016	**0.966** $\pm$.015	**0.965** $\pm$.014

Per-cell-line, `Sci-Plex3`. As reported in (Table 12) CVAE and splitDGD are within 0.003 of each other on every line and lead on A549 and K562; splitDGD (sup-$\mathbf{w}$) leads on MCF7 but trails both on K562. The do-nothing control scores 0.942–0.956, above ContrastiveVI, scDisInFact, and DisCoVR on every line, and CVAE and splitDGD exceed it by at most 0.007. On this dataset the all-genes counterfactual R^2 therefore mostly measures how closely a model reproduces the control profile of the target line.

J PerturBench metrics on Sci-Plex3

On `Sci-Plex3` the counterfactual R^2 cannot separate methods: a prediction of no drug effect already scores 0.948. We therefore also report four metrics from PerturBench (Wu et al., 2024) under the same transfer protocol (Table 13).

Common setup. All four metrics are computed per held-out (cell line, compound) pair from the same counterfactual as the R^2 of Table 2: up to 400 randomly drawn control cells of the target line are decoded with the compound's transferred $\mathbf{w}$, giving a predicted population $\hat{X}_p$ with mean profile $\hat{\boldsymbol{\mu}}_p$; the observed population X_p consists of the held-out pair's test cells (263 on average) with mean $\boldsymbol{\mu}_p$, and $\boldsymbol{\mu}_0$ is the mean of the same control cells. Log-fold-changes are differences of log-normalised means, $\hat{\boldsymbol{\delta}}_p = \hat{\boldsymbol{\mu}}_p - \boldsymbol{\mu}_0$ and $\boldsymbol{\delta}_p = \boldsymbol{\mu}_p - \boldsymbol{\mu}_0$. Every model is decoded with its likelihood mean, and every metric is averaged over pairs and reported as mean $\pm$ std over the five split seeds.

Rank. Within a cell line holding out P compounds, let $d(j,i) = 1 - \cos(\hat{\boldsymbol{\delta}}_j, \boldsymbol{\delta}_i)$ be the cosine distance between the prediction for compound j and the observed effect of compound i. The rank of compound i is the fraction of *other* compounds whose prediction is at least as close to its observed effect as its own prediction,

$$\mathrm{rank}_i = \frac{1}{P-1} \sum_{j \neq i} \Big(\mathbf{1}\big[d(j,i) < d(i,i)\big] + \tfrac{1}{2}\,\mathbf{1}\big[d(j,i) = d(i,i)\big] \Big),$$

averaged over the compounds of the line and then over the three lines (PerturBench, Eq. 2; ties count one half, so that a model issuing one prediction for every compound scores exactly 0.5 rather than 1). It is 0 when every compound's own prediction is the closest and 0.5 for a model that ignores the

Table 12: Per-cell-line counterfactual R^2 on `Sci-Plex3`, mean $\pm$ std over 5 split seeds. Transfer protocol: the drug's $\mathbf{w}$ is the mean over its training cells in the other two cell lines. Each value is the unweighted mean of the per-pair R^2 over the held-out (cell line, drug) pairs in the column. **Bold** = best, underline = second-best among the six methods; the reference rows are not ranked. Each held-out drug appears in exactly one cell line, so each line column averages a different set of drugs (25–33 per line and seed).

Method	A549	K562	MCF7	*All*
CVAE	**0.953** ±.008	**0.949** ±.006	0.961 ±.022	**0.954** ±.008
ContrastiveVI	0.922 ±.008	0.883 ±.011	0.949 ±.020	0.917 ±.007
scDisInFact	0.862 ±.011	0.890 ±.012	0.861 ±.014	0.872 ±.011
DisCoVR	0.841 ±.049	0.829 ±.037	0.874 ±.012	0.847 ±.019
splitDGD	0.952 ±.007	**0.949** ±.006	0.959 ±.020	0.953 ±.007
splitDGD (sup-$\mathbf{w}$)	0.942 ±.023	0.917 ±.025	**0.962** ±.021	0.939 ±.022
Reference (not ranked)				
Control (no drug effect)	0.947 ±.008	0.942 ±.008	0.956 ±.023	0.948 ±.008
Ceiling (observed half)	0.965 ±.003	0.946 ±.004	0.987 ±.003	0.965 ±.002

compound. It is computed within a line because each held-out compound appears in exactly one line, so ranking across lines would let line identity stand in for compound identity. Unlike R^2, rank does not reward reproducing the shared expression landscape; it asks only whether predictions are compound-specific.

Cosine LogFC. The cosine similarity $\cos(\hat{\boldsymbol{\delta}}_p, \boldsymbol{\delta}_p)$ between predicted and observed log-fold-changes. Since Pearson correlation is cosine similarity on centred vectors, this is the counterfactual R^2 with the control profile removed instead of the grand mean: it scores the predicted *change*, and a no-effect prediction scores 0 instead of 0.948. A single generic response still correlates with every specific one (one profile shared by all compounds, the mean of a model's own predictions, already reaches ≈ 0.39), so this metric rewards predicting *that* a compound acts more than *which*.

Energy distance. Following PerturBench, a PCA basis of 256 components is fitted on the observed test cells of each seed and both populations are projected onto it. The energy distance $2\,\mathrm{E}\|a-b\| - \mathrm{E}\|a-a'\| - \mathrm{E}\|b-b'\|$, with a, a' drawn from $\hat{X}_p$ and b, b' from X_p (self-pairs excluded; at most 300 cells per side), is 0 for identical distributions. It compares whole cell populations, so it is sensitive to the spread of the predicted cells and not only to their mean.

DEG recall. Per-gene Welch t-statistics of the predicted and of the observed cells against the control cells are ranked separately; the recall is the overlap between the two top-20 lists of up-regulated genes (largest signed t), divided by 20.

Reference values. Two model-free predictions calibrate every column. *Control* predicts no effect and returns the control cells themselves: rank 0.500, cosine LogFC 0, energy distance 0.489, DEG recall 0.003. *Ceiling* lets one random half of each pair's observed cells predict the other half, and so bounds each metric by the sampling noise of ≈ 260 cells per pair: rank 0.023, cosine LogFC 0.563, energy distance ≈ 0, DEG recall 0.331.

Results. Models' performance are reported in Table 13.On rank, splitDGD (sup-$\mathbf{w}$) has the lowest rank (0.280), covering 46% of the distance from chance to the ceiling, followed by ContrastiveVI (0.339), splitDGD (0.371), DisCoVR (0.394), and scDisInFact (0.434); CVAE, which has no $\mathbf{w}$, is at chance. Cosine LogFC orders the methods differently: CVAE scores highest (0.438) with a single prediction for every drug, so this metric rewards a generic drug response rather than a drug-specific one; among the split-latent methods, both splitDGD variants (0.370–0.400) score well above the baselines (0.07–0.11).

Energy distance and DEG recall compare predicted and observed cell populations rather than mean profiles. All models are decoded with their likelihood mean, so their predicted populations lack cell-to-cell noise, and both metrics are dominated by that for every model: the no-effect control,

Table 13: Counterfactual prediction on `Sci-Plex3` under the PerturBench metrics defined in the text (Wu et al., 2024), transfer protocol, mean ± std over 5 split seeds. Rank: 0 is perfect, 0.5 is chance. All models are decoded with their likelihood mean. **Bold** = best, underline = second-best.

Method	Rank ↓	Cos. LogFC ↑	Energy dist. ↓	DEG recall ↑
CVAE	0.501 ±.005	**0.438** ±.022	1.874 ±.352	0.002 ±.002
ContrastiveVI	0.339 ±.031	0.106 ±.022	3.057 ±.225	**0.007** ±.006
scDisInFact	0.434 ±.018	0.070 ±.024	11.039 ±.153	0.001 ±.001
DisCoVR	0.394 ±.025	0.076 ±.026	4.234 ±.527	0.000 ±.000
splitDGD	0.371 ±.083	0.400 ±.045	**1.865** ±.111	0.002 ±.002
splitDGD (sup-**w**)	**0.280** ±.034	0.370 ±.073	2.038 ±.341	0.006 ±.003

which consists of real cells, has the lowest energy distance (0.489), and no model recovers more top-20 DEGs than chance ($\leq$ 0.007, against 0.003 for the control and 0.331 for the ceiling). Among the models, the two splitDGD variants and CVAE have the lowest energy distance (1.87–2.04), against 3.06–11.04 for the other baselines.

K LATENT OVERVIEW: SUPERVISED-**w** VARIANT ON HABER

Figure 5 reports the latent overview for splitDGD (sup-**w**) on `Haber`, the supervised counterpart of Figure 4. The **z**-block panels are visually similar to the unsupervised regime; the **w**-block panel resolves Hpoly.Day3, Hpoly.Day10, and Salmonella into three disjoint clusters, as expected from the supervised-**w** loss (Eq. 8). The contrast with Figure 4(c) quantifies what subtype supervision adds at the representation level.

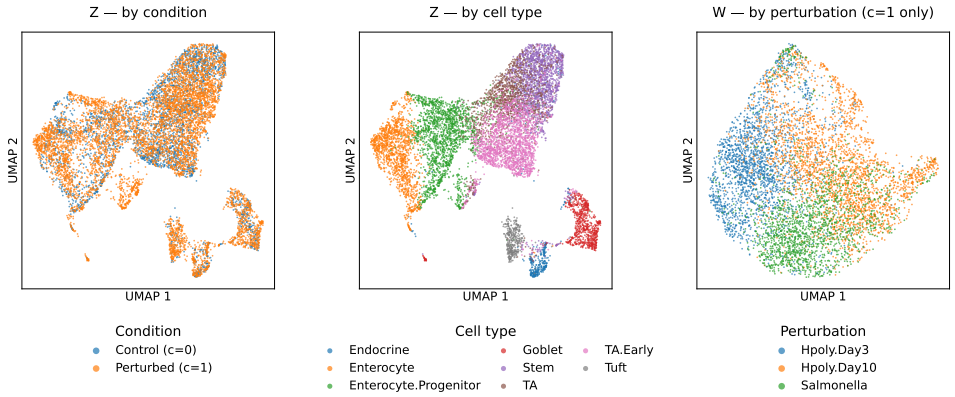


Figure 4: Latent overview on `Haber`, splitDGD (unsupervised regime). UMAP projections of (a) **z** coloured by condition, (b) **z** coloured by cell type, and (c) **w** (perturbed cells only) coloured by perturbation subtype. The **w**-block separates the three perturbations (Hpoly.Day3, Hpoly.Day10, Salmonella) without seeing subtype labels at training.

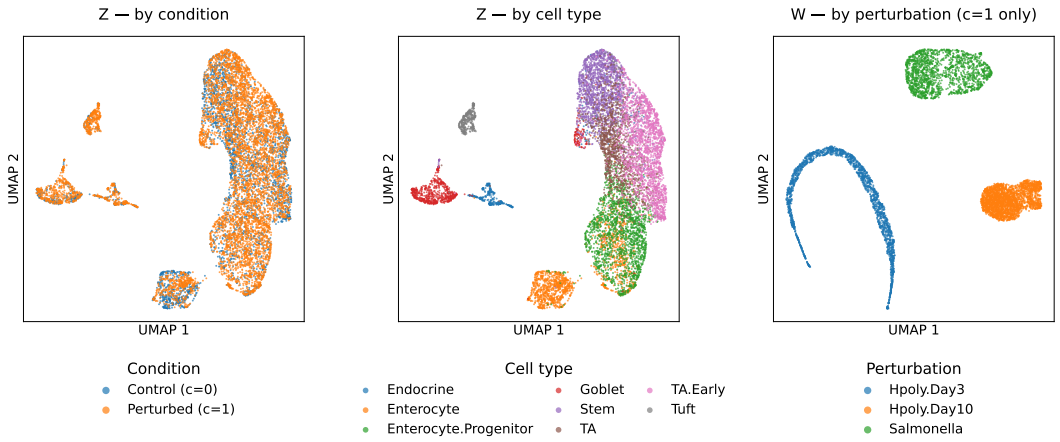


Figure 5: Latent overview on `Haber`, splitDGD (sup-**w**). UMAP projections of (a) **z** coloured by condition, (b) **z** coloured by cell type, and (c) **w** (perturbed cells only) coloured by perturbation subtype. Compare with Figure 4 for the unsupervised regime.