# CRISPYDRUG: A CAUSAL FRAMEWORK FOR LEARNING JOINT REPRESENTATIONS OF GENETIC AND CHEMICAL PERTURBATIONS IN CELLS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Large-scale single-cell RNA-seq perturbation datasets provide detailed measurements of how interventions alter cellular state. However, existing perturbation models typically focus on either genetic or chemical perturbations, while causal methods for jointly modeling both perturbation types are missing. Here, we introduce CrispyDrug, a novel causal framework for learning joint representations of genetic and chemical perturbations from single-cell transcriptomic data. CrispyDrug treats both perturbation types as interventions on a shared cellular outcome and learns representations through their effects on post-treatment gene expression. Our framework proceeds in two stages. (1) First, it learns perturbation representations by predicting post-treatment expression and, at the same time, balances the representations wrt. to the perturbation type. (2) Second, it trains perturbation-type-specific decoders that map the joint representations back to genetic or chemical perturbations. Across large-scale experiments in the K562 leukemia cell line, we find that CrispyDrug captures heterogeneous transcriptional responses and achieves stronger predictive performance than existing perturbation models. By analyzing the learned latent space and cross-perturbation retrieval, we further find that the joint representation links genetic and chemical perturbations through similarities in their transcriptional effects. To the best of our knowledge, CrispyDrug is the first framework to jointly model both genetic and chemical perturbations in single-cell transcriptomics with an explicit causal interpretation.

## 1 INTRODUCTION

Single-cell perturbation transcriptomics measures how cellular state changes in response to controlled interventions (e.g., Peidli et al., 2024; Heumos et al., 2026; Dimitrov et al., 2026): (i) *genetic perturbations*, such as CRISPR-based gene perturbations, or (ii) *chemical perturbations*, such as small-molecule treatments (see Figure 1). Single-cell RNA-sequencing (scRNA-seq) is then performed post-treatment. Here, *we aim to jointly learn representations of genetic and chemical perturbations to predict their effects on post-treatment gene expression measured by scRNA-seq*.

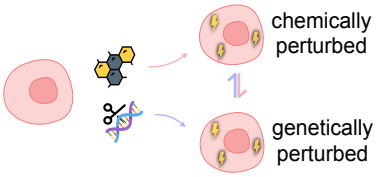



Figure 1: Cell lines perturbed by either a chemical or a genetic perturbagen, and their gene expression has been sequenced post-treatment.

Existing methods for this task typically model **either** genetic perturbations **or** chemical perturbations (see Section 2).[1] For example, state-of-the-art methods such as GEARS and CPA (Roohani et al., 2024; Hetzel et al., 2022) focus on genetic perturbations, whereas chemCPA, PRnet, and GraphVCI (Lotfollahi et al., 2023; Qi et al., 2024; Wu et al., 2023) primarily model chemical perturbations. Closest to our setting is BioLORD (Piran et al., 2024), which can be applied across both perturbation types, but does **not** jointly learn a shared representation from genetic and chemical perturbations. However, *jointly* modeling both perturbation types is desirable in practice for several reasons: (1) **knowledge sharing**, the response patterns learned from one perturbation type can inform the other through the common measured cellular outcome of both perturbation types (post-treatment gene expression); (2) **inverse therapeutic queries** such as identifying compounds

[1] For extensive literature overviews, see Peidli et al. (2024); Heumos et al. (2026); Dimitrov et al. (2026).



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.

that approximate the cellular effects of targeting gene X in cell type Y (i.e, target-to-drug), thereby offering an efficient way of designing therapies; (3) **target validation**, by guiding researchers about the potential context-dependent effect (Lamb et al., 2006; Krill-Burger et al., 2023; Moffat et al., 2024).

A key challenge is that perturbation-response prediction is inherently a *causal inference* problem. In causal terminology (Rubin, 1974; Rosenbaum & Rubin, 1983; Imbens & Rubin, 2015), genetic and chemical perturbations are interventions, the post-treatment gene expression is the outcome, and the causal target is the cellular state that would result under a given intervention. However, because scRNA-seq requires cell lysis, each cell is observed under only one treatment condition; responses to alternative interventions therefore are unobserved and remain counterfactual. Surprisingly, despite the causal structure of the task, existing perturbation models (e.g., Roohani et al., 2024; Hetzel et al., 2022; Lotfollahi et al., 2023; Qi et al., 2024) are formulated primarily as predictive or associational learning problems and do **not** allow for a causal interpretation to their predictions. To this end, a *causal machine learning* framework for multi-perturbation analysis is missing.

To address this gap, **we introduce *CrispyDrug*, a causal framework for learning joint representations of genetic and chemical perturbations from single-cell transcriptomic data.** *CrispyDrug* treats genetic and chemical perturbations as different interventions to learn a joint representation that predicts the causal effect on cellular outcomes. A non-trivial challenge is that the two perturbation types are defined in different treatment spaces, such that a shared representation should be aligned by similarity in the *causal effects* rather than by similarity between the perturbagens themselves. To do so, our *CrispyDrug* framework consists of two stages: (1) Perturbation-type-specific encoders map genetic and chemical perturbagens into a shared latent space; the representation is learned to be both (i) predictive of cellular outcomes and (ii) non-predictive of the perturbation type to promote outcome prediction *and* cross-perturbation generalization. (2) Perturbation-type-specific decoders map the learned representations back to the corresponding genetic or chemical perturbation space, thus allowing for rigorous causal interpretation.

Our **main contributions** are three-fold:[2]

(1) We introduce *CrispyDrug*, which is, to the best of our knowledge, the first representation learning framework for jointly estimating the causal effect of both genetic and chemical perturbations.

(2) We provide an explicit causal formulation based on potential outcomes, where we establish identification at both the perturbation and representation levels, and assign the causal interpretation to interventions in the representation space through natural soft interventions.

(3) Using large-scale perturbation datasets of the K562 leukemia cell line, we show that the predictive performance of our *CrispyDrug* outperforms existing baselines and that the learned joint representations capture biologically meaningful structure for cross-perturbation insights.

## 2 Related Work

**Single-cell perturbation modeling.** Single-cell perturbation modeling aims to predict how cellular gene expressions change in response to a genetic or chemical intervention. Recent approaches have focused on modeling genetic and chemical perturbations *separately* (see Table 1). • For *genetic* perturbations, **GEARS** (Roohani et al., 2024) uses prior knowledge to embed the perturbed and unperturbed states; and **CPA** combines disentanglement and contrastive learning (Hetzel et al., 2022). • For *chemical* perturbations, **chemCPA** relies on disentanglement and contrastive learning to train a perturbation and covariate discriminator (Lotfollahi et al., 2023); **GraphVCI** refines a gene relation graph (Kamimoto et al., 2023) and uses covariate and treatment to predict gene expression post perturbation; **PRnet** utilizes DMSO/control-treated cells as input, which provides a learnable perturbation embedding built from the perturbagen molecule and dose to then predict post-perturbation expression (Wu et al., 2023; Qi

Table 1: Comparison of key methods for single-cell perturbation modeling.

| Method | Chemical | Genetic | Joint | Causal |
|---|---|---|---|---|
| GEARS (Roohani et al., 2024) | ✗ | ✓ | ✗ | ✗ |
| CPA (Hetzel et al., 2022) | ✗ | ✓ | ✗ | ✗ |
| PRnet (Qi et al., 2024) | ✓ | ✗ | ✗ | ✗ |
| GraphVCI (Wu et al., 2023) | ✓ | ✗ | ✗ | ✓ |
| chemCPA (Lotfollahi et al., 2023) | ✓ | ✗ | ✗ | ✗ |
| Biolord (Piran et al., 2024) | ✓ | ✓ | ✗ | ✗ |
| ***CrispyDrug* (ours)** | ✓ | ✓ | ✓ | ✓ |

[2]Code is available via an anonymized Git repository (`https://anonymous.4open.science/r/crispydrug_iclr-B11F/`). Upon acceptance, we will move the code to a public GitHub.

et al., 2024); and **BioLord** (Piran et al., 2024) learns disentangled representations that can be used across perturbations and cellular contexts. ⇒ *However, these methods treat perturbation modeling as a prediction problem within a specific treatment space. Consequently, genetic and chemical perturbations are typically modeled* ***separately*** *and not represented within a common space. In contrast, we learn a* ***joint*** *representation to promote knowledge sharing.*

**Joint representation learning across perturbation types.** Given the richness of information that each perturbation type can deliver, this motivates an approach that focuses on this. Moreover, recently, works have incorporated both perturbation types through foundation models (Adduri et al., 2026; Taj & Stein, 2025), which seek to generalize across perturbation types and biological contexts. Nevertheless, foundation models tend to optimize at the cost of the richness of each perturbation type, and they do not address coupling chemical and genetic perturbations. Unipert (Li et al., 2026), which was published as we write this article, is the only method that addresses the coupling of genetic and chemical perturbations. They do so by using graph-based, contrastive objectives and database annotations to achieve a common space on which the perturbagens would be represented, namely, protein space. Unipert pairs perturbagens of different types independently of cellular outcome, thus ignoring the potential for perturbagen $\times$ cell type interactions. ⇒ *However, methods from this stream do* ***not*** *utilize perturbation transcriptomics. More importantly, existing works do* ***not*** *provide an explicit causal interpretation of the learned perturbation representations and, as we show later, do not address the* ***multiple-versions-of-treatment*** *problem (VanderWeele & Hernan, 2013) induced by mapping distinct perturbations to the same representation.*

**Causal modeling of cellular perturbations.** Perturbation transcriptomics is inherently a causal inference problem, where the perturbations act as controlled interventions, and the resulting molecular measurements quantify the effect on the cellular state (Dimitrov et al., 2026). Yet, most perturbation models discussed above are formulated primarily as predictive or associative models (Hetzel et al., 2022; Qi et al., 2024; Roohani et al., 2024; Lotfollahi et al., 2023), *without* an explicit causal estimand or causal identification strategy. Again, as we will show later, by ignoring causal identification, a many-to-one treatment representation can induce a *multiple-versions-of-treatment* problem.

Existing causal approaches in scRNA address two different causal tasks. (1) **Causal discovery** methods (Rohbeck et al., 2024; Kamimoto et al., 2023; Wen et al., 2023) aim to recover the regulatory graph from observational, interventional, or temporal data. However, their primary output is a *regulatory graph* but **not** the *causal effect* of a perturbation. (2) **Counterfactual attribution** methods (Dong et al., 2023; Bunne et al., 2023) address a *counterfactual* problem by coupling control and perturbed cellular populations to infer how a specific cell observed under one treatment would have behaved under another. In Pearl's causal hierarchy (Pearl, 2009; Bareinboim et al., 2022), this is a layer-three *counterfactual* query (i.e., *what would have happened under an alternative treatment?*) but **not** a layer-two *interventional* question of estimating treatment effects on cellular outcomes (i.e., $\mathrm{do}(T = t)$; i.e., *what happens to the cellular outcome if we intervene and set the treatment to t*?). ⇒ *However, despite the interventional nature of perturbation transcriptomics, explicit treatment effect modeling at the interventional level — particularly jointly across genetic and chemical perturbations — is underexplored.*

**Reserach gap.** Despite the inherently interventional nature of perturbation transcriptomics, there is no method for jointly modeling genetic and chemical perturbations wrt. their treatment effects.

## 3 Causal framework for learning joint representations

In the following, we introduce our causal framework for the prediction of the potential gene expressions after cell perturbations and motivate *CrispyDrug* from a theoretical perspective.

**Input.** We consider a fused experimental dataset $\mathcal{D}_N = \{(x_i, s_i, t_i, y_i)\}_{i=1}^N$ sampled i.i.d. from a (mixture) distribution $\mathbb{P}(S, X, T, Y)$ where $N$ is the dataset size. The dataset is 'fused' in the sense that it contains data from two types of perturbations (i.e., genetic and chemical). The corresponding mixture distribution $\mathbb{P}(X, S, T, Y)$ merges two perturbations types and has the following variables:

• $X \in \mathcal{X} \subseteq \mathbb{R}^{d_x}$ are *pre-treatment covariates*. In our case, those are average expressions of the unexposed cells (namely, cells under a baseline condition) from the same parent cell line. As we will see later, using such baseline covariates as $X$ is compatible with our causal assumptions. Specifically, the average expressions of the unexposed cells are computed from separately sequenced, unexposed cells and are thus neither affected by nor involved in the assignment of perturbations.

- $S \in \mathcal{S} = \{\mathrm{g}, \mathrm{c}\}$ is a random *perturbation type*: g is a genetic perturbation, and c is a chemical perturbation, respectively.
- $T \in \mathcal{T} = \{\mathrm{c}\} \times \mathcal{T}_\mathrm{c} \cup \{\mathrm{g}\} \times \mathcal{T}_\mathrm{g}$ is a *perturbation (treatment) variable* that depends on the perturbation type. Here, for $S = \mathrm{g}$ genetic perturbations, $\mathcal{T}_\mathrm{g} = \mathcal{A}_\mathrm{g} \times \mathcal{D}_\mathrm{g}$ is a *bundle genetic treatment space* consisting of a genetic embedding, $A \in \mathcal{A}_\mathrm{g}$, and a binary dosage indicator, $D \in \mathcal{D}_\mathrm{g} = \{0, 1\}$. At the same time, for $S = \mathrm{c}$ chemical perturbations, $\mathcal{T}_\mathrm{c} = \mathcal{A}_\mathrm{c} \times \mathcal{D}_\mathrm{c}$ is a *bundle chemical treatment space* consisting of a chemical embedding $A \in \mathcal{A}_\mathrm{c}$, and a continuous dosage on a log-scale, $D \in \mathcal{D}_\mathrm{c} \in \mathbb{R}^+$. For both perturbation type, the control is represented as zero (i.e., $(0, 0) \in \mathcal{T}_s$ for $s \in \{\mathrm{g}, \mathrm{c}\}$).
- $Y \in \mathbb{R}^{d_y}$ is an outcome, i.e., a $d_y$-dimensional gene expression after a perturbation $T$ of type $S$.

Prediction modeling often focuses on predicting the expression profiles of treated cells (Roohani et al., 2024; Hetzel et al., 2022; Wu et al., 2023; Qi et al., 2024). These approaches typically model only a single perturbation type (at a time). However, our method, apart from predicting gene expression, also incorporates multiple perturbation types and uses causal language to characterize the relationship between interventions and outcomes, thereby providing a causal interpretation of the learned representations.

**Causal prediction target.** In our work, we adopt the *potential outcomes framework* (Rubin, 1974) to formalize the causal task and thus ensure a causal interpretation. We denote by $Y[s, t]$ the potential gene-expression outcome that would be observed under perturbation $t$ of type $s$. For a cell exposed to the observed perturbation $(S, T) = (s, t)$, $Y[s, t]$ coincides with the *factual* outcome $Y$, whereas the outcomes $Y[s', t']$ under alternative perturbations $(s', t') \neq (s, t)$ are *counterfactual* and are therefore not jointly observed for the same cell (see Figure 2 (a) for an illustration).

We aim to learn a *low-dimensional representation of both perturbation types* $R_s(t) : \mathcal{T}_s \to \mathcal{R} \subseteq \mathbb{R}^{d_r}$, such that it satisfies the following two conditions. That is, it should be (1) predictive of the potential gene expression $Y[s, t]$ after a perturbation $t$ of type $s$ (which is called a *hard intervention*, where the perturbation type and treatment are externally fixed to $(s, t)$ rather than generated according to their observational assignment mechanism) and (2) represented in a shared latent space where it is agnostic to the perturbation type $S$ (we therefore refer to the representation space as *joint* across genetic and chemical perturbations). Both properties are as follows:

**(a)** For the fixed representation $r = R_s(t)$, the conditional distribution of the potential gene expression at the representation level (i.e., $\mathbb{P}(Y[r] \mid X = x)$) should be as close as possible to the analogous distribution in the original high-dimensional treatment space (i.e., $\mathbb{P}(Y[s, t] \mid X = x)$):

$$\mathbb{P}(Y[r] \mid X = x) \approx \mathbb{P}(Y[s, t] \mid X = x) \text{ for } r = R_s(t), \text{ where } t \in \mathcal{T}. \tag{1}$$

**(b)** Then, the random representation $R_S(T)$ should be independent of the perturbation type $S$:

$$S \perp\!\!\!\perp R_S(T). \tag{2}$$

Enforcing this independence property is also called a *balancing constraint* for representations $R_S(T)$ (Johansson et al., 2016; Ganin et al., 2016).

**Causal assumptions.** In order to identify both potential distributions of gene expressions (i.e., $\mathbb{P}(Y[r] \mid X = x)$ and $\mathbb{P}(Y[s, t] \mid X = x)$) from the observed data distribution $\mathbb{P}(X, S, T, Y)$, we rely on the following standard assumptions for causal inference (Rubin, 1974; Rosenbaum & Rubin, 1983; Imbens & Rubin, 2015). (i) ***Consistency***: If a cell is exposed to the treatment $T = t$ of type $S = s$, its potential gene expression coincides with the observed one: $Y = Y[s, t]$. (ii) ***Overlap***: the probability of receiving any treatment is bounded away from zero: $\mathbb{P}(T = t, S = s \mid X = x) > 0$. (iii) ***Full randomization***: Treatment assignment is fully randomized (i.e., the data is experimental): $(S, T) \perp\!\!\!\perp (Y[s, t], X)$ for every $s \in \mathcal{S}$ and $t \in \mathcal{T}$. Under these assumptions, both potential outcome distributions from Eq. (1) are identified. Also, the assumptions (a)–(c) are naturally satisfied in cell experiments and correspond to the causal diagram in Figure 2.

**High-dimensional treatment effects.** The distribution of the potential gene expression in the original treatment space is as follows:

$$\mathbb{P}(Y[s, t] \mid X = x) \overset{\text{Ass. (iii)}}{=} \mathbb{P}(Y[s, t] \mid X = x, T = t, S = s) \overset{\text{Ass. (i)}}{=} \mathbb{P}(Y \mid X = x, T = t, S = s), \tag{3}$$

and Ass. (ii) overlap ensures that the latter conditional distribution is well-defined. Yet, learning the right-hand side of Eq. (3) poses a big challenge, given that both $Y$ and $T$ can be high-dimensional.

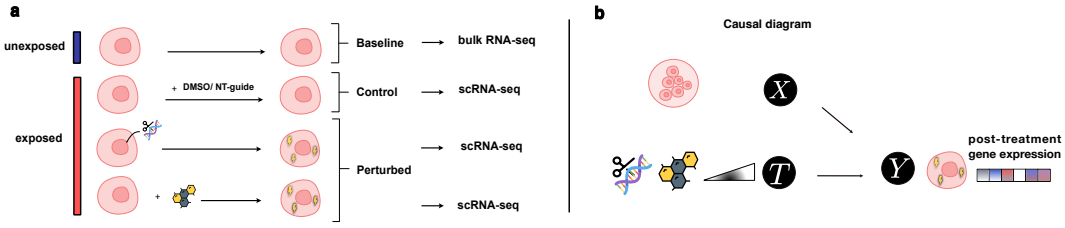



Figure 2: **A causal view on scRNA perturbations**. **(a)** The destructive nature of scRNA sequencing (Chi et al., 2026; Dong et al., 2023) precludes observing the same cell's response under more than one intervention assignment. Moreover, it also prevents measuring exposed cells' individual pre-treatment expressions. Consequently, cells derived from a clone of a common parent cell line are assigned to separate samples that are either left unexposed (= baseline condition); exposed to dimethyl sulfoxide (DMSO) in the case of chemical perturbations or non-targeting guides in the case of genetic perturbations (= the control); or exposed to a genetic/chemical perturbagen. Importantly, we use the averaged expression (also known as bulk RNA-seq) of the unexposed cells as a proxy for the unobservable pre-treatment cell expression $X$. **(b)** A causal diagram that characterizes the fused experimental data and motivates our *CrispyDrug*.

Hence, our main goal is to learn low-dimensional representations of treatments $R_s(t)$ with a proper causal interpretation.

**Representation-level effects & multiple-versions-of-treatment problem.** The representation-level potential gene expressions, on the other hand, require slightly more attention. Since $R_s(\cdot)$ is a deterministic and, in general, many-to-one transformation from the original high-dimensional perturbation space into a low-dimensional representation space, ***setting $R$ to $r$ does not uniquely determine an original perturbation*** $t$. We therefore interpret $Y[r]$ as the potential outcome under some *soft intervention*[3] $\pi_r(s,t)$ (Muñoz & van der Laan, 2012; VanderWeele & Hernan, 2013), namely, a probability distribution supported on the set $\mathcal{G}(r) := \{(s,t) \in \mathcal{S} \times \mathcal{T} : R_s(t) = r\}$ (= pre-image of $R_s$). Then, under the assumptions (a)–(c), the conditional distribution of $Y[r]$ is identified as

$$\begin{aligned} \mathbb{P}(Y[r] \mid X = x) &\overset{\text{def.}}{=} \sum_{(s,t)\in\mathcal{G}(r)} \mathbb{P}(Y[s,t] \mid S = s, T = t, X = x)\, \pi_r(s,t) \\ &\overset{\text{Ass. (iii),(i)}}{=} \sum_{(s,t)\in\mathcal{G}(r)} \mathbb{P}(Y \mid S = s, T = t, X = x)\, \pi_r(s,t). \end{aligned} \tag{4}$$

We immediately see that different choices of $\pi_r$ generally lead to different potential distributions of gene expressions,[4] $\mathbb{P}(Y[r] \mid X = x)$, even when the representation $R_s(t)$ is kept fixed. It turns out, one particular choice of the soft intervention that is based on the observed data distribution $P(S, T, R_S(T))$, namely, $\pi_r(s,t) = \mathbb{P}(S = s, T = t \mid R = r)$, yields a much simpler identification formula:

$$\begin{aligned} \mathbb{P}(Y[r] \mid X = x) &\overset{\text{Eq. (4)}}{=} \sum_{(s,t)\in\mathcal{G}(r)} \mathbb{P}(Y \mid S = s, T = t, X = x)\, \mathbb{P}(S = s, T = t \mid R = r) \\ &\overset{\text{Ass. (iii)}}{=} \sum_{(s,t)\in\mathcal{G}(r)} \mathbb{P}(Y \mid S = s, T = t, X = x, R = r)\, \mathbb{P}(S = s, T = t \mid R = r, X = x) \\ &= \mathbb{P}(Y \mid R = r, X = x), \end{aligned} \tag{5}$$

where the last equality holds by the law of total probability. Therefore, we call this soft intervention ***natural*** and, from now on, we use it as our *main causal interpretation* of the representation-level potential gene expression $Y[r]$. Informally, the natural soft intervention $r$ corresponds to a mixture over all genetic and chemical perturbations that map to the same representation $r$, weighted according to their prevalence in the observed perturbation distribution.

**How to learn a representation with desired properties?** So far, we have shown how an *arbitrary* representation $r = R_s(t)$ can be assigned a causal interpretation from the observed data. However, our initial goal was to learn a representation that is **(a)** predictive of the potential outcomes $Y[s,t]$

[3] Unlike hard interventions, that apply the same genetic or chemical perturbation to every cell; soft interventions, in our setting, assign cells a distribution over perturbations that map to the same representation $r$.

[4] Unless the representation $R_s(t)$ is invertible, or the right-hand side of Eq. (4) is invariant wrt. $\pi_r$.

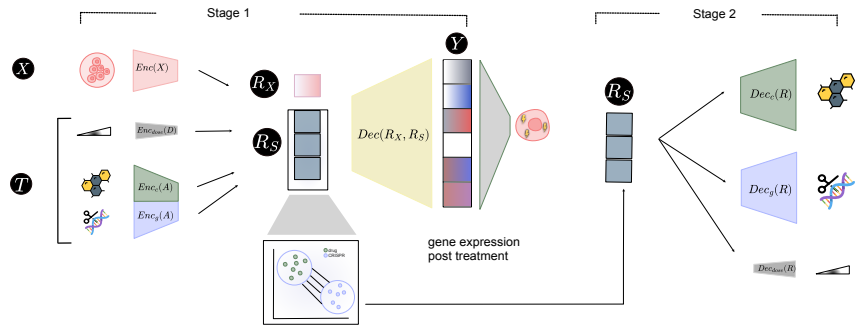



Figure 3: ***CrispyDrug* architecture.** Stage (1): We encode cell-line bulk transcriptomics and the perturbagen to predict post-perturbation gene expression, while aligning genetic and chemical perturbation representations. Stage (2): We freeze the representations and use perturbation-type-specific decoders to reconstruct the respective gene or molecule embedding from the learned representation.

and **(b)** agnostic to a perturbation type $S$. Luckily, both **(a)** and **(b)** can be achieved within our causal framework. For example, Eq. (5) suggests a direct strategy for **(a)**: Since $\mathbb{P}(Y[r] \mid X = x) = \mathbb{P}(Y \mid R = r, X = x) = \mathbb{P}(Y \mid R = R_s(t), X = x)$, we can learn an ideal representation $R_s^*(t)$ together with an outcome model $P^*(y \mid r, x)$ that estimates the ground truth $\mathbb{P}(Y[r] \mid X = x)$ by *jointly maximizing the log-likelihood* of the observed gene expression $Y$:

$$P^*, R_s^* = \arg\max_{P, R_s} \mathbb{E}\big[\log P(Y \mid R_S(T), X)\big]. \tag{6}$$

At the same time, to achieve **(b)** the independence wrt. the perturbation type, we can minimize some distributional distance $\mathrm{dist}\{\cdot;\cdot\}$ between the representations of the genetic perturbations, $\mathbb{P}(R_S \mid S = \mathrm{g})$, and the representations of the chemical distributions, $\mathbb{P}(R_S \mid S = \mathrm{g})$. Thus, a perfectly balanced representation $R_s^*(t)$ would minimize the following risk:

$$R_s^* = \arg\min_{R_s} \mathbb{E}\big[\,\mathrm{dist}\,\big\{\mathbb{P}(R_S(T) \mid S = \mathrm{g}); \mathbb{P}(R_S(T) \mid S = \mathrm{c})\big\}\big]. \tag{7}$$

Therefore, enforcing both properties **(a)** and **(b)** requires combining both risks in Eq. (6)-(7).

**Motivation of our *CrispyDrug*.** Our causal framework naturally motivates a two-stage architecture. In *Stage* (1), we learn the representation $R_s(t)$ to be simultaneously **(a)** predictive of post-perturbation gene expression and **(b)** balanced across perturbation types, *directly operationalizing* Eq. (6)-(7). In *Stage* (2), we fix the learned joint representations; and, by using perturbation-type-specific probabilistic decoders, we recover the underlying *natural soft interventions* mechanism, $\mathbb{P}(S, T \mid R = r)$. Both stages thus enable cross-perturbation-type learning of perturbations with similar transcriptional effects.

## 4 CrispyDrug: Neural instantiation of the causal framework

**Overview.** *CrispyDrug* proceeds in two stages (see Figure 3). In **Stage** (1), perturbation-type-specific encoders map genetic and chemical perturbagens into a common latent space. These representations are learned such that they are simultaneously predictive of post-perturbation gene expression and balanced across perturbation types, thereby encouraging perturbations with similar cellular effects to be represented similarly. In **Stage** (2), the Stage (1) encoders are frozen, and the learned perturbation representations are used as fixed inputs to perturbation-type-specific decoders, which reconstruct the corresponding genetic or chemical perturbagen embeddings. We provide pseudocode in Algorithm 1 in the appendix.

### 4.1 Stage (1): Perturbed expression prediction

The goal of Stage (1) is to learn a perturbation representation $R_S$ that satisfies the two properties introduced in Sec. 3: it should be **(a)** *predictive* of the post-perturbation cellular outcome while being **(b)** *independent* of the perturbation type. To achieve this, we jointly encode the perturbation and the cellular covariates, predict the post-perturbation gene-expression distribution, and regularize the perturbation representations to align across genetic and chemical interventions.

**How to model the multidimensional outcome $Y$.** To predict the distribution of multidimensional post-perturbation expression profiles, we first define the observed outcome $Y$ as the vector of gene-expression counts measured by scRNA-seq after applying perturbation $T$ of type $S$ to cells with

covariates $X$. Given that the observed outcomes are single-cell gene expression counts, we model the corresponding conditional distribution using a zero-inflated negative binomial (ZINB) distribution (Gayoso et al., 2022; Lopez et al., 2018):

$$Y \sim \mathbb{P}(Y \mid X = x, T = t, S = s) = \text{ZINB}(\mu, \theta, z \mid X = x, T = t, S = s), \tag{8}$$

where $\mu, \theta, z$ are the parameters of the conditional ZINB distribution. Here, $\mu$ is the expected negative binomial mean, $\theta$ is the log dispersion parameter and $z$ are the dropout logits.

**Encoding perturbations and cellular covariates.** To learn the parameters of the conditional ZINB distribution, we use (i) a treatment encoder, (ii) a covariate encoder, and (iii) a decoder as follows: (i) The treatment is represented by the perturbagen embedding $A$ and its dose $D$, while $S \in \mathrm{g}, \mathrm{c}$ indicates whether the perturbation is genetic or chemical. Hence, the treatment encoder outputs the representation

$$R_S \in \mathbb{R}^{d_r} := R_S(T) = \mathrm{Enc}_S(A) + \mathrm{Enc}_{\text{dose}}(D) \tag{9}$$

where $\mathrm{Enc}_S$ denotes a perturbation-type-specific encoder, with separate parameter sets for genetic and chemical perturbations. (ii) The covariate encoder maps the pre-treatment covariates $X$ into the same-dimensional latent space:

$$R_X \in \mathbb{R}^{d_r} := R_X(X) = \mathrm{Enc}(X), \tag{10}$$

(iii) Then, the decoder concatenates the representation of covariates $R_X$ and the representation of a perturbagen $R_S$ and predicts the parameters of the conditional ZINB distribution:

$$\hat{\mu}, \hat{\theta}, \hat{z} := \mathrm{Dec}(R_X, R_S). \tag{11}$$

**Learning the representations.** To train two encoders and the decoder, we jointly minimize a composite loss that consists of four components. The first three objectives encourage $R_S$ to retain information that is *predictive* of the cellular response, while the fourth encourages $R_S$ to be *independent* of the perturbation type by aligning the genetic and chemical representation distributions.

**(1)** First, to properly learn conditional distributions, we compute a negative log-likelihood loss wrt. the ground-truth samples $Y$, namely

$$\mathcal{L}_{\text{ZINB}} = -\frac{1}{N} \sum_{i=1}^{N} \log[P(y_i; \hat{\mu}_i, \hat{\theta}_i, \hat{z}_i)], \tag{12}$$

where is the probability mass function of the ZINB distribution.

**(2)** To increase the quality of the average predictions, we also compute a normalized mean squared error (nMSE) loss between the predicted gene expressions and the observed mean gene expression profiles:

$$\mathcal{L}_{\text{nMSE}} = \frac{\sum_{j=1}^{d_y} ([\hat{\bar{\mu}}]_j - [\bar{\mu}]_j)^2}{\sum_{j=1}^{d_y} ([\bar{\mu}]_j - \sum_{j=1}^{d_y} [\bar{\mu}]_j)^2}, \tag{13}$$

where $[\cdot]_j$ is a $j$-th coordinate of the vector, $\hat{\bar{\mu}} = 1/N \sum_{i=1}^{N} \hat{\mu}_i$ and $\bar{\mu} = 1/N \sum_{i=1}^{N} y_i$ are the average predicted and observed gene expressions, respectively.

**(3)** In addition to the biological response, the observed count distribution depends on library size, i.e., the total number of observed counts, reads, or unique molecular identifiers (UMIs) assigned to an individual cell. Library size primarily reflects sequencing depth and capture efficiency rather than the biological effect of the perturbation, but it determines the scale at which the expression signal is observed (Gayoso et al., 2022; Lopez et al., 2018). We therefore use both $R_X$ and $R_S$ to predict the library size. For this, we introduce an auxiliary loss based on the Huber loss:

$$\mathcal{L}_{\text{library}} = \mathrm{HuberLoss}(\text{library}_{\text{pred}}, \text{library}_{\text{target}}). \tag{14}$$

**(4)** To make the representations of the treatment $R_S$ independent of the perturbation types, we adopt an idea from balancing representations (Johansson et al., 2016) and enforce a Wasserstein distance:

$$\mathcal{L}_{\text{bal}} = \hat{W}^2 \big\{ \mathbb{P}(R_S \mid S = \mathrm{g}); \mathbb{P}(R_S \mid S = \mathrm{c}) \big\}, \tag{15}$$

where $\hat{W}^2$ is the empirical Wasserstein distance. Minimizing this loss discourages the representation from encoding perturbation type and promotes alignment between the genetic and chemical latent distributions.

**Combined Stage ① loss:** Together, the components **(1)–(3)** are responsible for **(a)** making the representations $R_S$ predictive of post-perturbation gene expression; while **(4)** makes the same representation **(b)** independent of the perturbation type (see Sec. 3). We then optimize the combined loss

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{ZINB}} + \gamma \cdot \mathcal{L}_{\text{nMSE}} + \omega \cdot \mathcal{L}_{\text{bal}} + \Omega \cdot \mathcal{L}_{\text{library}}, \tag{16}$$

where $\gamma \geq 0$ , $\omega \geq 0$ and $\Omega \geq 0$ are the coefficients of corresponding losses from above. At the end of Stage ①, the perturbation encoders therefore provide a shared, outcome-informed representation $R_S$ for **both** genetic and chemical interventions.

By jointly optimizing the predictive and balancing objectives, Stage ① directly operationalizes the causal framework introduced in Sec. 3. Then, to recover the underlying natural soft interventions (and, thus, causal interpretations) from learned representations $R_S$, we proceed with the following.

### 4.2 Stage ②: Perturbagen reconstruction

In Stage ②, we keep the perturbation encoders fixed and use these learned representations to map perturbations back into their original modality-specific embedding spaces. This allows us to examine how perturbations represented through similar cellular effects relate across genetic and chemical treatment spaces.

**Perturbation-type-specific reconstruction.** We train two perturbation-type-specific decoders and one dosage decoder that reconstruct the corresponding perturbagen embedding:

$$\widehat{A}_g = \text{Dec}_g(R), \qquad \widehat{A}_c = \text{Dec}_c(R), \qquad \widehat{D} = \text{Dec}_{\text{dose}}(R), \tag{17}$$

where $\text{Dec}_g$ maps the shared representation back to the genetic perturbation space and $\text{Dec}_c$ maps it back to the chemical perturbation space. In this way, the decoders immediately allow for modelling a natural soft intervention, defined in Eq. (5), via a conditional normal distribution:

$$\hat{\pi}_r(S,T) = \hat{\mathbb{P}}(S,T \mid R) = N(\text{Dec}_S(R); \sigma^2 = 1). \tag{18}$$

**Cross-perturbation knowledge sharing.** Once trained, both decoders can be applied to the representation obtained from either a genetic or a chemical perturbation. For example, a representation obtained from a genetic perturbation can be decoded into the chemical embedding space, and vice versa. The reconstructed embedding can then be compared with perturbagens in the respective treatment space, allowing information learned from one perturbation type to be transferred to the other.

**Interpreting the shared representation.** Stage ② therefore provides an operational way to examine whether perturbations with similar representations (and, hence, similar predicted cellular effects) have meaningful structure across treatment spaces. If genetic and chemical perturbations with similar representations are mapped to corresponding perturbagens across the two treatment spaces, this suggests that the shared representation captures cross-perturbation structure associated with their predicted cellular responses.

**Reconstruction objective.** To train the perturbation-type-specific decoders and the dosage decoders, we minimize the sum of the following two reconstruction errors:

$$\mathcal{L}_{\text{rec}} = \mathcal{L}_{\text{pert}} + \lambda_d \, \mathcal{L}_{\text{dose}} = \frac{1}{N} \sum_{i=1}^{N} ||A_i - \widehat{A}_{S_i}||_2^2 + \lambda_d \, \frac{1}{N} \sum_{i=1}^{N} ||D_i - \widehat{D}_i||_2^2, \tag{19}$$

where $|| \cdot ||_2$ is an $L_2$ norm. Each decoder is optimized only using the corresponding perturbation types/dosages, while the Stage ① encoders remain fixed.

**Implementation.** We provide further implementation details in Appendix B.

## 5 Experiments

**Setup.** We evaluate both stages of *CrispyDrug*. For Stage ①, we evaluate how accurately the model predicts post-perturbation gene expression for held-out perturbations. For Stage ②, we (i) analyze the learned joint representation space and (ii) perform cross-perturbation retrieval between genetic and chemical perturbations.

Table 2: **K562 datasets results**, mean ± standard error over 5 random seeds. Reported: mean absolute error (MAE), mean squared error (MSE), $R^2$-score, and Pearson correlation ($r$).

| **Chemical models** | MAE ↓ | MSE ↓ | $R^2$ ↑ | $r$ ↑ |
|---|---|---|---|---|
| chemCpa | 0.0968 ± 0.0002 | 0.0295 ± 0.0003 | 0.6667 ± 0.0040 | 0.8910 ± 0.0027 |
| PRnet | 0.0839 ± 0.0000 | 0.0391 ± 0.0000 | 0.4002 ± 0.0000 | 0.8725 ± 0.0000 |
| Biolord | 0.0362 ± 0.0003 | 0.0080 ± 0.0001 | 0.9223 ± 0.0007 | 0.9888 ± 0.0002 |
| *CrispyDrug* (c) | **0.0215 ± 0.0000** | **0.0031 ± 0.0000** | **0.9796 ± 0.0000** | **0.9922 ± 0.0000** |
| **Genetic models** | MAE ↓ | MSE ↓ | $R^2$ ↑ | $r$ ↑ |
| CPA | 1.5429 ± 0.0051 | 55.9765 ± 0.0153 | -0.0003 ± 0.0003 | 0.0080 ± 0.0059 |
| GEARS | 0.1208 ± 0.0005 | 0.1708 ± 0.0030 | 0.9971 ± 0.0000 | 0.9993 ± 0.0000 |
| *CrispyDrug* (g) | **0.0834 ± 0.0000** | **0.0701 ± 0.0000** | **0.9995 ± 0.0000** | **0.9998 ± 0.0000** |

**Baselines**. For Stage ①, we compare against the perturbation-response prediction methods summarized in Table 1, using the applicable baselines for each perturbation type. In Stage ②, since existing methods do not provide a direct counterpart to our joint representation and cross-perturbation retrieval, we evaluate this stage without a direct baseline.

**Data.** All K562 datasets were obtained through the `pertpy` package (Heumos et al., 2026). For genetic perturbations, we use the K562 Replogle dataset (Replogle et al., 2022), which contains 1,089 perturbations. For chemical perturbations, we use the K562 sci-Plex 3 dataset (Srivatsan et al., 2020), comprising 289 drugs. Genetic perturbations were preprocessed following the procedure used in GEARS to retain strong perturbations. Across datasets, we restricted the features to their common set and harmonized observations across perturbation types. In the fused dataset, the total number of cells is $N = 432448$. For further details on data preprocessing, we refer to Appendix C.

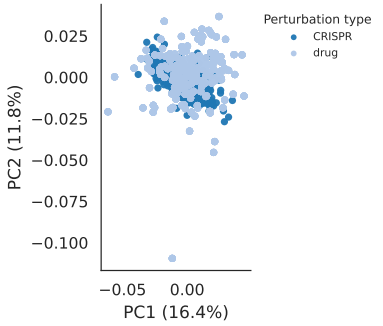



Figure 4: **Joint perturbation representations.** PCA of the learned *CrispyDrug* representations, showing substantial overlap between genetic and chemical perturbations.

**Stage ① results**. Table 2 reports post-perturbation gene-expression prediction performance for chemical and genetic perturbations, using identical perturbation-level held-out splits and datasets across methods. Furthermore, in Appendix C, we show that the joint predictions of our *CrispyDrug* are well-calibrated.

**Stage ② results: (i) Representation space.** Figure 4 shows substantial overlap between genetic and chemical perturbations in the learned representation space. This indicates that balancing reduces perturbation-type-specific structure while preserving outcome-predictive representations (see Table 2). Overall, these results confirm the effectiveness of *CrispyDrug* in learning a shared representation across perturbation types.

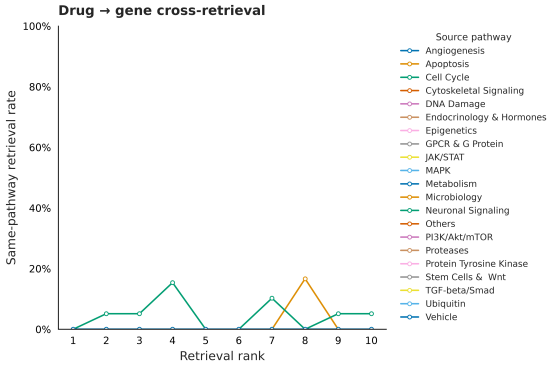



Figure 5: **Drug-to-gene cross-retrieval.** Same-pathway retrieval rate by rank; shaded regions show 95% Wilson confidence intervals.

**Stage ② results: (ii) Perturbagen retrieval.** Stage ② enables cross-retrieval between perturbation types. Figure 5 shows drug-to-gene retrieval, where retrieved genes are evaluated by whether they share the source drug's annotated pathway. The observed same-pathway retrieval confirms that the learned representations capture biologically sound cross-perturbation structure.

**Ablation study**. In Appendix C, we compare two learned representation spaces with different balancing loss weights $\omega$. Therein, when $\omega = 0$, genetic and chemical perturbations are strongly separated, whereas $\omega = 0.25$ substantially aligns the two representation distributions.

**Conclusion.** We introduced *CrispyDrug*, a causal framework for learning joint representations of genetic and chemical perturbations through their effects on cellular state.

### AI USE STATEMENT

In this work, we used generative AI tools to draft and debug portions of the experimental code. The code was reviewed, executed, and tested by the authors, and all reported results were independently reproduced. We also used generative AI tools in this work to improve grammar, spelling, and readability. We have not used generative AI tools for generating synthetic data, developing theoretical models or conceptual frameworks, formulating mathematical claims, assisting with proofs, proposing hypotheses, designing the methodology or experiments, implementing the proposed method, translating content, cleaning or reformatting datasets, conducting qualitative or thematic analysis, or interpreting results; these uses are not applicable to this work. We did not use generative AI tools for any additional tasks for which disclosure is recommended. We reviewed all AI-assisted text and verified that it accurately reflects the authors' intended meaning. We take responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

### ETHICS STATEMENT

This work uses previously published, publicly available single-cell perturbation datasets and does not involve the collection of new human or animal data. *CrispyDrug* is intended as a research framework for modeling genetic and chemical perturbations in vitro. Predictions and retrieved perturbations should be regarded as computational hypotheses rather than evidence of biological efficacy or safety and require experimental validation before downstream use. Moreover, our experiments are restricted to the K562 cell line, and the learned relationships should not be assumed to generalize to other cell types, organisms, or clinical settings.

### REPRODUCIBILITY STATEMENT

We provide the information required to reproduce our experiments throughout the main paper and appendix. We describe the datasets and preprocessing procedure, model architecture and training objectives, perturbagen representations, hyperparameters, and evaluation protocol. All datasets used in this work are publicly available through `pertpy` and their original sources, and experimental results are reported across random seeds. Code for preprocessing, training, and evaluation is provided through an anonymized repository and will be made publicly available upon acceptance.

## REFERENCES

Abhinav K. Adduri, Dhruv Gautam, Beatrice Bevilacqua, Mohsen Naghipourfar, Alishba Imran, Rohan Shah, Noam Teyssier, Rishi Verma, Christopher Carpenter, Basak Eraslan, Francis Chalissery, Rajesh Ilango, Vishvak Subramanyam, Chiara Ricci-Tam, Sanjay Nagaraj, Aidan Winters, Mingze Dong, Stefanie Fellinger, Adam Krejci, Tilmann Burckstummer, Sravya Tirukkovular, Jeremy Sullivan, Brian S. Plosky, Nicholas D. Youngblut, Jure Leskovec, Luke A. Gilbert, Silvana Konermann, Patrick D. Hsu, Alexander Dobin, Dave P. Burke, Hani Goodarzi, and Yusuf H. Roohani. Predicting cellular responses to perturbation across diverse contexts with State. *Cell*, 189(19):5914–5931.e20, September 2026. ISSN 00928674. doi: 10.1016/j.cell.2026.07.052.

Elias Bareinboim, Juan D. Correa, Duligur Ibeling, and Thomas Icard. On Pearl's hierarchy and the foundations of causal inference. In *Probabilistic and Causal Inference: The Works of Judea Pearl*, pp. 507–556. Association for Computing Machinery, 2022.

Charlotte Bunne, Stefan G. Stark, Gabriele Gut, Jacobo Sarabia Del Castillo, Mitch Levesque, Kjong-Van Lehmann, Lucas Pelkmans, Andreas Krause, and Gunnar Rätsch. Learning single-cell perturbation responses using neural optimal transport. *Nature Methods*, 20(11):1759–1768, November 2023. ISSN 1548-7091, 1548-7105. doi: 10.1038/s41592-023-01969-x.

Changxi Chi, Jun Xia, Yufei Huang, Zhuoli Ouyang, Cheng Tan, Yunfan Liu, Jingbo Zhou, Chang Yu, Liangyu Yuan, Siyuan Li, et al. Doloris: Dual conditional diffusion implicit bridges with sparsity masking strategy for unpaired single-cell perturbation estimation. In *International Conference on Learning Representations*, 2026.

Oriol Corcoll, Athanasios Vlontzos, Michael O'Riordan, and Ciarán M Gilligan-Lee. Contrastive representations of structured treatments. *npj Artificial Intelligence*, 2(1):49, 2026.

Daniel Dimitrov, Stefan Schrod, Martin Rohbeck, and Oliver Stegle. Interpretation, extrapolation and perturbation of single cells. *Nature Reviews Genetics*, 27(5):349–370, May 2026. ISSN 1471-0056, 1471-0064. doi: 10.1038/s41576-025-00920-4.

Mingze Dong, Bao Wang, Jessica Wei, Antonio H. De O. Fonseca, Curtis J. Perry, Alexander Frey, Feriel Ouerghi, Ellen F. Foxman, Jeffrey J. Ishizuka, Rahul M. Dhodapkar, and David Van Dijk. Causal identification of single-cell experimental perturbation effects with CINEMA-OT. *Nature Methods*, 20(11):1769–1779, November 2023. ISSN 1548-7091, 1548-7105. doi: 10.1038/s41592-023-02040-5.

Jean Feydy, Thibault Séjourné, François-Xavier Vialard, Shun-ichi Amari, Alain Trouvé, and Gabriel Peyré. Interpolating between optimal transport and mmd using sinkhorn divergences. In *International Conference on Artificial Intelligence and Statistics*, 2019.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(59):1–35, 2016.

Adam Gayoso, Romain Lopez, Galen Xing, Pierre Boyeau, Valeh Valiollah Pour Amiri, Justin Hong, Katherine Wu, Michael Jayasuriya, Edouard Mehlman, Maxime Langevin, Yining Liu, Jules Samaran, Gabriel Misrachi, Achille Nazaret, Oscar Clivio, Chenling Xu, Tal Ashuach, Mariano Gabitto, Mohammad Lotfollahi, Valentine Svensson, Eduardo Da Veiga Beltrame, Vitalii Kleshchevnikov, Carlos Talavera-López, Lior Pachter, Fabian J. Theis, Aaron Streets, Michael I. Jordan, Jeffrey Regier, and Nir Yosef. A Python library for probabilistic analysis of single-cell omics data. *Nature Biotechnology*, 40(2):163–166, February 2022. ISSN 1087-0156, 1546-1696. doi: 10.1038/s41587-021-01206-w.

Max Goplerud, Kosuke Imai, and Nicole E. Pashley. Estimating heterogeneous causal effects of high-dimensional treatments: Application to conjoint analysis. *The Annals of Applied Statistics*, 19(2):866–888, 2025.

Shonosuke Harada and Hisashi Kashima. GraphITE: Estimating individual effects of graph-structured treatments. In *International Conference on Information & Knowledge Management*, 2021.

Miguel A Hernán, Babette Brumback, and James M Robins. Marginal structural models to estimate the joint causal effect of nonrandomized treatments. *Journal of the American Statistical Association*, 96(454):440–448, 2001.

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

Lukas Heumos, Yuge Ji, Lilly May, Tessa D. Green, Stefan Peidli, Xinyue Zhang, Xichen Wu, Johannes Ostner, Antonia Schumacher, Karin Hrovatin, Michaela Müller, Faye Chong, Gregor Sturm, Alejandro Tejada, Emma Dann, Mingze Dong, Gonçalo Pinto, Mojtaba Bahrami, Ilan Gold, Sergei Rybakov, Altana Namsaraeva, Amir Ali Moinfar, Zihe Zheng, Eljas Roellin, Isra Mekki, Chris Sander, Mohammad Lotfollahi, Herbert B. Schiller, and Fabian J. Theis. Pertpy: An end-to-end framework for perturbation analysis. *Nature Methods*, 23(2):350–359, February 2026. ISSN 1548-7091, 1548-7105. doi: 10.1038/s41592-025-02909-7.

Kosuke Imai and Kentaro Nakamura. Leveraging generative AI for causal inference with unstructured data. *Proceedings of the National Academy of Sciences*, 123(36):e2530532123, 2026.

Guido W. Imbens and Donald B. Rubin. *Causal Inference for Statistics, Social, and Biomedical Sciences: An Introduction*. Cambridge University Press, New York, 2015. ISBN 9780521885881.

Fredrik Johansson, Uri Shalit, and David Sontag. Learning representations for counterfactual inference. In *International Conference on Machine Learning*, 2016.

Jean Kaddour, Yuchen Zhu, Qi Liu, Matt J. Kusner, and Ricardo Silva. Causal effect inference for structured treatments. *Advances in Neural Information Processing Systems*, 2021.

Kenji Kamimoto, Blerta Stringa, Christy M. Hoffmann, Kunal Jindal, Lilianna Solnica-Krezel, and Samantha A. Morris. Dissecting cell identity via network inference and in silico gene perturbation. *Nature*, 614(7949):742–751, February 2023. ISSN 0028-0836, 1476-4687. doi: 10.1038/s41586-022-05688-9.

Patrick Kramer, Edward H. Kennedy, and Isaac M. Opper. Causal inference with high-dimensional treatments. *arXiv preprint arXiv:2602.21423*, 2026.

J. Michael Krill-Burger, Joshua M. Dempster, Ashir A. Borah, Brenton R. Paolella, David E. Root, Todd R. Golub, Jesse S. Boehm, William C. Hahn, James M. McFarland, Francisca Vazquez, and Aviad Tsherniak. Partial gene suppression improves identification of cancer vulnerabilities when CRISPR-Cas9 knockout is pan-lethal. *Genome Biology*, 24(1):192, August 2023. ISSN 1474-760X. doi: 10.1186/s13059-023-03020-w.

Justin Lamb, Emily D. Crawford, David Peck, Joshua W. Modell, Irene C. Blat, Matthew J. Wrobel, Jim Lerner, Jean-Philippe Brunet, Aravind Subramanian, Kenneth N. Ross, Michael Reich, Haley Hieronymus, Guo Wei, Scott A. Armstrong, Stephen J. Haggarty, Paul A. Clemons, Ru Wei, Steven A. Carr, Eric S. Lander, and Todd R. Golub. The Connectivity Map: Using Gene-Expression Signatures to Connect Small Molecules, Genes, and Disease. *Science*, 313(5795): 1929–1935, September 2006. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.1132939.

Yiming Li, Min Zeng, Jun Zhu, Linjing Liu, Fang Wang, Longkai Huang, Fan Yang, Min Li, and Jianhua Yao. UniPert-G2CP bridges genetic and chemical screens from molecular representation to phenotype modeling. *Cell*, 189(16):4946–4963.e8, August 2026. ISSN 00928674. doi: 10.1016/j.cell.2026.06.005.

Shiangyi Lin, Hui Lan, and Vasilis Syrgkanis. Learning treatment representations for downstream instrumental variable regression. In *International Conference on Machine Learning*, 2026.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, Allan Dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Salvatore Candido, and Alexander Rives. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, March 2023. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.ade2574.

Romain Lopez, Jeffrey Regier, Michael B. Cole, Michael I. Jordan, and Nir Yosef. Deep generative modeling for single-cell transcriptomics. *Nature Methods*, 15(12):1053–1058, December 2018. ISSN 1548-7105. doi: 10.1038/s41592-018-0229-2.

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

Mohammad Lotfollahi, Anna Klimovskaia Susmelj, Carlo De Donno, Leon Hetzel, Yuge Ji, Ignacio L Ibarra, Sanjay R Srivatsan, Mohsen Naghipourfar, Riza M Daza, Beth Martin, Jay Shendure, Jose L McFaline-Figueroa, Pierre Boyeau, F Alexander Wolf, Nafissa Yakubova, Stephan Günnemann, Cole Trapnell, David Lopez-Paz, and Fabian J Theis. Predicting cellular responses to complex perturbations in high-throughput screens. *Molecular Systems Biology*, 19(6):MSB202211517, June 2023. ISSN 1744-4292. doi: 10.15252/msb.202211517.

Jason Moffat, Alexis C. Komor, and Lawrence Lum. Impact of CRISPR in cancer drug discovery. *Science*, 386(6720):378–379, October 2024. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.adi6884.

Iván Díaz Muñoz and Mark van der Laan. Population intervention causal effects based on stochastic interventions. *Biometrics*, 68(2):541–549, 2012.

Kentaro Nakamura, Adam Breuer, Michael H Crespin, Bryce J. Dietrich, and Kosuke Imai. Causal inference with video features as treatments. *arXiv preprint arXiv:2607.06126*, 2026.

Sonali Parbhoo, Stefan Bauer, and Patrick Schwab. NCoRE: Neural counterfactual representation learning for combinations of treatments. *arXiv preprint arXiv:2103.11175*, 2021.

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

Stefan Peidli, Tessa D. Green, Ciyue Shen, Torsten Gross, Joseph Min, Samuele Garda, Bo Yuan, Linus J. Schumacher, Jake P. Taylor-King, Debora S. Marks, Augustin Luna, Nils Blüthgen, and Chris Sander. scPerturb: Harmonized single-cell perturbation data. *Nature Methods*, 21(3):531–540, March 2024. ISSN 1548-7091, 1548-7105. doi: 10.1038/s41592-023-02144-y.

Zoe Piran, Niv Cohen, Yedid Hoshen, and Mor Nitzan. Disentanglement of single-cell data with biolord. *Nature Biotechnology*, 42(11):1678–1683, November 2024. ISSN 1087-0156, 1546-1696. doi: 10.1038/s41587-023-02079-x.

Xiaoning Qi, Lianhe Zhao, Chenyu Tian, Yueyue Li, Zhen-Lin Chen, Peipei Huo, Runsheng Chen, Xiaodong Liu, Baoping Wan, Shengyong Yang, and Yi Zhao. Predicting transcriptional responses to novel chemical perturbations using deep generative model for drug discovery. *Nature Communications*, 15(1):9256, October 2024. ISSN 2041-1723. doi: 10.1038/s41467-024-53457-1.

Joseph M. Replogle, Reuben A. Saunders, Angela N. Pogson, Jeffrey A. Hussmann, Alexander Lenail, Alina Guna, Lauren Mascibroda, Eric J. Wagner, Karen Adelman, Gila Lithwick-Yanai, Nika Iremadze, Florian Oberstrass, Doron Lipson, Jessica L. Bonnar, Marco Jost, Thomas M. Norman, and Jonathan S. Weissman. Mapping information-rich genotype-phenotype landscapes with genome-scale Perturb-seq. *Cell*, 185(14):2559–2575.e28, July 2022. ISSN 00928674. doi: 10.1016/j.cell.2022.05.013.

James M Robins, Miguel Angel Hernan, and Babette Brumback. Marginal structural models and causal inference in epidemiology, 2000.

Martin Rohbeck, Brian Clarke, Katharina Mikulik, Alexandra Pettet, Oliver Stegle, and Kai Ueltzhöffer. Bicycle: Intervention-based causal discovery with cycles. In *Conference on Causal Learning and Reasoning*, 2024.

Yusuf Roohani, Kexin Huang, and Jure Leskovec. Predicting transcriptional outcomes of novel multigene perturbations with GEARS. *Nature Biotechnology*, 42(6):927–935, June 2024. ISSN 1087-0156, 1546-1696. doi: 10.1038/s41587-023-01905-6.

Paul R. Rosenbaum and Donald B. Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 1983.

Jerret Ross, Brian Belgodere, Vijil Chenthamarakshan, Inkit Padhi, Youssef Mroueh, and Payel Das. Large-scale chemical language representations capture molecular structure and properties. *Nature Machine Intelligence*, 4(12):1256–1264, December 2022. ISSN 2522-5839. doi: 10.1038/s42256-022-00580-7.

Donald B. Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology*, 66(5):688, 1974.

Jonas Schweisthal, Dennis Frauen, Valentyn Melnychuk, and Stefan Feuerriegel. Reliable off-policy learning for dosage combinations. In *Advances in Neural Information Processing Systems*, 2023.

Jonas Schweisthal, Dennis Frauen, Maresa Schröder, Konstantin Hess, Niki Kilbertus, and Stefan Feuerriegel. Learning representations of instruments for partial identification of treatment effects. In *International Conference on Machine Learning*, 2025.

Ankit Sharma, Garima Gupta, Ranjitha Prasad, Arnab Chatterjee, Lovekesh Vig, and Gautam Shroff. Hi-ci: Deep causal inference in high dimensions. In *KDD Workshop on Causal Discovery*, 2020.

Lei Shi, David Arbour, Raghavendra Addanki, Ritwik Sinha, and Avi Feller. Leveraging semantic similarity for experimentation with ai-generated treatments. *Advances in Neural Information Processing Systems*, 2026.

Sanjay R. Srivatsan, José L. McFaline-Figueroa, Vijay Ramani, Lauren Saunders, Junyue Cao, Jonathan Packer, Hannah A. Pliner, Dana L. Jackson, Riza M. Daza, Lena Christiansen, Fan Zhang, Frank Steemers, Jay Shendure, and Cole Trapnell. Massively multiplex chemical transcriptomics at single-cell resolution. *Science*, 367(6473):45–51, January 2020. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.aax6234.

Farzan Taj and Lincoln D Stein. OmniPert: A Deep Learning Foundation Model for Predicting Responses to Genetic and Chemical Perturbations in Single Cancer Cells, July 2025.

Akira Tanimoto, Tomoya Sakai, Takashi Takenouchi, and Hisashi Kashima. Regret minimization for causal inference on large treatment space. In *International Conference on Artificial Intelligence and Statistics*, 2021.

Tyler J. VanderWeele and Miguel A. Hernan. Causal inference under multiple versions of treatment. *Journal of causal inference*, 1(1):1–20, 2013.

Yujian Wen, Jielong Huang, Shuhui Guo, Yehezqel Elyahu, Alon Monsonego, Hai Zhang, Yanqing Ding, and Hao Zhu. Applying causal discovery to single-cell analyses using CausalCell. *eLife*, 12:e81464, May 2023. ISSN 2050-084X. doi: 10.7554/eLife.81464.

Yulun Wu, Rob Barton, Zichen Wang, Vassilis N. Ioannidis, Carlo De Donno, Layne C Price, Luis F. Voloch, and George Karypis. Predicting cellular responses with variational causal inference and refined relational information. In *International Conference on Learning Representations*, 2023.

Qi Xu, Haoda Fu, and Annie Qu. Optimal individualized treatment rule for combination treatments under budget constraints. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 86(3):714–741, 2024.

Hao Zou, Peng Cui, Bo Li, Zheyan Shen, Jianxin Ma, Hongxia Yang, and Yue He. Counterfactual prediction for bundle treatment. *Advances in Neural Information Processing Systems*, 2020.

# A Extended Related Work

**High-dimensional treatments in the potential outcomes framework.** From the point of view of the potential outcomes framework (Rubin, 1974) (which we adopt in our work), comparatively little attention has been dedicated to the high-dimensional treatments setting. Some notable (and mostly recent) exceptions focus on time-varying treatment effect estimation (Robins et al., 2000; Hernán et al., 2001); off-policy learning (Schweisthal et al., 2023); treatment effects with structured treatments (e.g., graphs, images, text) (Harada & Kashima, 2021; Kaddour et al., 2021; Shi et al., 2026; Imai & Nakamura, 2026; Nakamura et al., 2026); and high-dimensional instrumental variables (Tanimoto et al., 2021; Schweisthal et al., 2025; Lin et al., 2026). We refer to Kramer et al. (2026) for a more detailed overview of the topic and the formalization of the statistical complexity associated with learning high-dimensional treatment effects.

**Static approaches to high-dimensionality.** Given this statistical complexity, existing approaches often impose simplifying structure on the treatment space (Kaddour et al., 2021; Shi et al., 2026) or restrict it to predefined low-dimensional projections (Robins et al., 2000; Goplerud et al., 2025). We, on the other hand, follow a more flexible strategy and learn low-dimensional treatment representations directly from data; and, to the best of our knowledge, we are the first to formalize the causal interpretation of such representations *specifically in the context of scRNA prediction*.

**Learnable representations of treatments.** Many works suggested a natural deep learning approach to the high-dimensionality of the treatments: learning low-dimensional representations (Sharma et al., 2020; Zou et al., 2020; Harada & Kashima, 2021; Tanimoto et al., 2021; Parbhoo et al., 2021; Xu et al., 2024; Corcoll et al., 2026). The application of deep learning also helps to enforce different constraints over the representations (independence from covariates (Harada & Kashima, 2021; Tanimoto et al., 2021), sparsity or structure of treatment interactions (Xu et al., 2024), and invariance to causally irrelevant treatment features (Corcoll et al., 2026)). In our work however, we apply a *completely novel constraint*: We aim to balance the representations wrt. two treatment types with different structures (i.e., genetic and chemical perturbations), hence enforcing a joint structure of the representations.

# B CrispyDrug: Neural instantiation of the causal framework

## B.1 Algorithm

In the following Algorithm 1, we describe our *CrispyDrug* in detail.

**Algorithm 1** CrispyDrug

**procedure** Stage1($\mathcal{D}$)
  Initialize $\mathrm{Enc}_g, \mathrm{Enc}_c, \mathrm{Enc}_X, \mathrm{Dec}_Y$
  **for** each minibatch $\mathcal{B} \subset \mathcal{D}$ **do**
    **for** each $(X, S, A, D, Y) \in \mathcal{B}$ **do**
      **if** $S = g$ **then**
        $R \leftarrow \mathrm{Enc}_g(A) + \mathrm{Enc}_{\mathrm{dose}}(D)$
      **else**
        $R \leftarrow \mathrm{Enc}_c(A) + \mathrm{Enc}_{\mathrm{dose}}(D)$
      **end if**
      $R_X \leftarrow \mathrm{Enc}_X(X)$
      $(\hat{\mu}, \hat{\theta}, \hat{\pi}, \hat{l}) \leftarrow \mathrm{Dec}_Y([R_X; R])$
      $\hat{Y} \leftarrow (1 - \hat{\pi}) \odot \hat{\mu}$
    **end for**
    $\mathcal{L}_{\text{total}} \leftarrow \mathcal{L}_{\text{ZINB}}(Y, \hat{\mu}, \hat{\theta}, \hat{\pi}) + \gamma\, \mathcal{L}_{\text{nMSE}}(Y, \hat{Y}) + \omega\, \mathcal{L}_{\text{bal}}(\mathbb{1}\{S = \mathrm{g}\} \cdot R, \mathbb{1}\{S = \mathrm{c}\} \cdot R)$
      $+\Omega\, \mathcal{L}_{\text{library}}(l, \hat{l})$
    Update $\mathrm{Enc}_g, \mathrm{Enc}_c, \mathrm{Enc}_X, \mathrm{Dec}_Y, \mathrm{Enc}_{\text{dose}}$ using $\nabla \mathcal{L}_{\text{total}}$
  **end for**
  Freeze encoders and cache all representations $R$
**end procedure**
**procedure** Stage2($\{R_i, S_i, A_i, D_i\}$)
  Initialize $\mathrm{Dec}_g, \mathrm{Dec}_c, \mathrm{Dec}_{\mathrm{dose}}$
  **for** each minibatch **do**
    **for** each $(R, S, A, D)$ **do**
      **if** $S = g$ **then**
        $\hat{A} \leftarrow \mathrm{Dec}_g(R)$
      **else**
        $\hat{A} \leftarrow \mathrm{Dec}_c(R)$
      **end if**
      $\hat{D} \leftarrow \mathrm{Dec}_{\mathrm{dose}}(R)$
    **end for**
    $\mathcal{L}_{\text{rec}} \leftarrow \mathcal{L}_{\mathrm{pert}}(A, \hat{A}) + \lambda_d \mathcal{L}_{\mathrm{dose}}(D, \hat{D})$
    Update $\mathrm{Dec}_g, \mathrm{Dec}_c, \mathrm{Dec}_{\mathrm{dose}}$ using $\nabla \mathcal{L}_{\text{rec}}$
  **end for**
**end procedure**
Stage1($\mathcal{D}$)
Stage2($\{R_i, S_i, A_i, D_i\}$)

## B.2 Implementation of losses

The **Huber loss** is as follows:

$$\mathrm{HuberLoss}(y, \hat{y}) = \begin{cases} \frac{1}{2}(y - \hat{y})^2, & \text{if } |y - \hat{y}| \le \delta, \\ \delta\left(|y - \hat{y}| - \frac{1}{2}\delta\right), & \text{otherwise.} \end{cases} \tag{20}$$

In our experiments, we kept $\delta = 1.0$.

We use the **Sinkhorn approximation of the Wasserstein distance** as our balancing loss, implemented with `GeomLoss` (Feydy et al., 2019). Let $\hat{\alpha}$ and $\hat{\beta}$ denote the empirical distributions of the genetic and chemical perturbation representations, respectively. We compute

$$W^2\{\hat{\alpha}, \hat{\beta}\} = \mathcal{S}_\varepsilon(\hat{\alpha}, \hat{\beta}) = \mathrm{OT}_\varepsilon(\hat{\alpha}, \hat{\beta}) - \frac{1}{2}\,\mathrm{OT}_\varepsilon(\hat{\alpha}, \hat{\alpha}) - \frac{1}{2}\,\mathrm{OT}_\varepsilon(\hat{\beta}, \hat{\beta}), \tag{21}$$

where $\mathrm{OT}_\varepsilon$ denotes entropy-regularized optimal transport with regularization strength $\varepsilon$. In our case, GeomLoss uses the ground cost $C(r, r') = \frac{1}{2}\|r - r'\|_2^2$. In our experiments, we kept $\varepsilon = 0.5$.

We used the following weighting hyperparameters for the losses. In Stage (1), we set $\gamma = 0.5$ and $\Omega = 0.4$, and we vary $\omega \in (0.1, 0.3)$. In Stage (2), we set $\lambda_d = 1$.

### B.3 Other implementation details

All models are trained with a batch size of 64 using the AdamW optimizer (Loshchilov & Hutter, 2019), with a learning rate of $10^{-3}$ and a weight decay of $10^{-4}$. We apply early stopping with a patience of 5 epochs. All experiments are run on a single NVIDIA A40 GPU.

# C DATASETS & EXPERIMENTS

## C.1 DATA PREPROCESSING

**Cellular covariates.** Stage (1) takes as input the cellular covariates $X$ together with the perturbation $T$. Because scRNA-seq is destructive, the pre-treatment expression profile of an individual perturbed cell cannot be observed. We therefore use the average RNA expression of unexposed cells from the corresponding parent cell line as a proxy for the pre-treatment covariates $X$. The output of Stage (1) is the predicted post-perturbation gene-expression distribution $Y$ together with the learned perturbation representation $R_S(T)$.

**Genetic perturbation representation.** For genetic perturbations, we use protein-based embeddings derived from the genes being perturbed. Specifically, we obtain the corresponding protein sequences and encode them using the pretrained evolutionary-scale protein language model ESM2 (Lin et al., 2023). This provides the genetic perturbagen embedding $A_g$ used by the genetic perturbation encoder.

**Chemical perturbation representation.** For chemical perturbations, we represent the perturbagen directly based on the molecular structure using MolFormer embeddings (Ross et al., 2022). We deliberately avoid relying exclusively on annotated protein targets because compounds may act on multiple proteins or protein families, exhibit polypharmacology, and may lack complete target annotations. Molecular embeddings additionally allow the model to represent compounds for which such target information is unavailable.

**Out-of-distribution sets.** We have chosen identical sets as BioLord (Piran et al., 2024) for chemical perturbations and CPA (Hetzel et al., 2022) for genetic perturbations see Table 3.

Table 3: Out-of-distribution sets per perturbation type.

| Type | Perturbations |
|---|---|
| Genetic | MRPL24, MMS22L, GTF2H1, CDC73, IPO9, RPL6, AARS, NUMA1, KANSL3, TCERG1 |
| Chemical | Dacinostat, Givinostat, Belinostat, Hesperadin, Quisinostat, Alvespimycin,Tanespimycin, TAK-901, Flavopiridol, Trametinib |

### C.2 ADDITIONAL RESULTS

**Stage ① results**. To further assess the quality of the joint predictions learned in Stage ①, we compare predicted and observed gene-expression values on the held-out perturbations in Figure 6. Predictions closely track the observed expression values, yielding an $R^2$ of $0.841$ and a Pearson correlation of $0.918$. Although some large expression values tend to be underestimated, most predictions lie close to the identity line. This hints at the good predictive capabilities of our *CrispyDrug* wrt. *both perturbation types*.

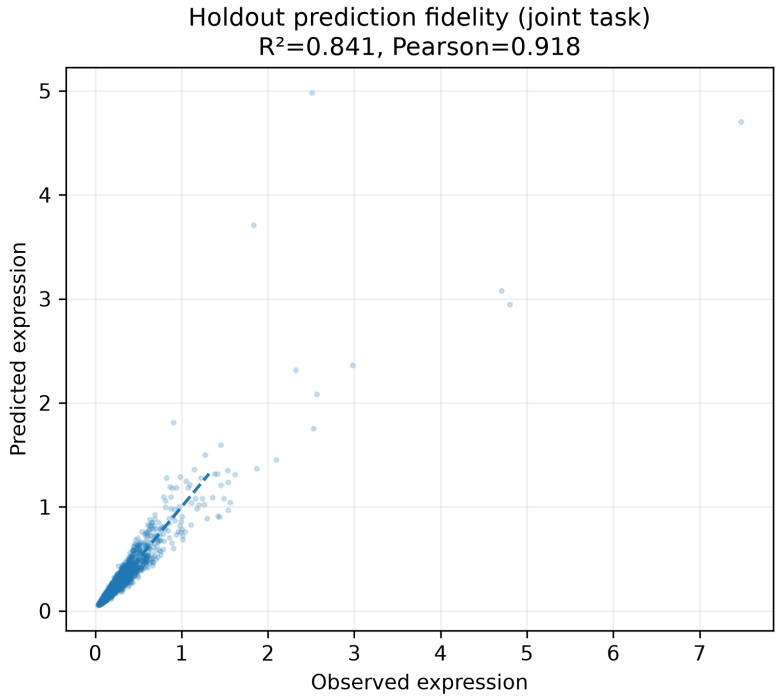



Figure 6: **Prediction fidelity on held-out perturbations.** Observed versus predicted gene-expression values for the joint *CrispyDrug* model. The dashed line denotes perfect agreement.

**Ablation study**. To illustrate the effect of the balancing objective, Figure 7 compares the learned representation spaces with $\omega = 0$ and $\omega = 0.25$. Without balancing ($\omega = 0$), genetic and chemical perturbations are almost perfectly separated along the first principal component. In contrast, with $\omega = 0.25$, the two perturbation types substantially overlap, showing that the balancing objective removes perturbation-type-specific separation and promotes a shared representation space.

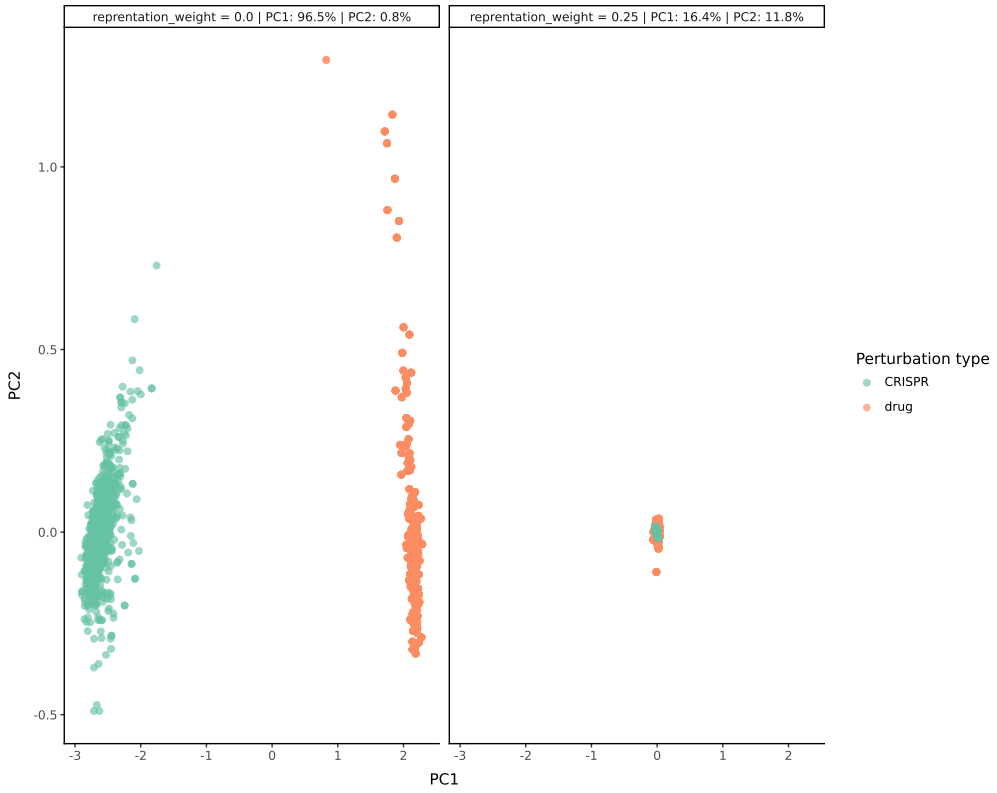



Figure 7: **Effect of the balancing weight on the learned representation space.** PCA projections of the learned perturbation representations without balancing ($\omega = 0$, left) and with balancing ($\omega = 0.25$, right), colored by perturbation type.