Inductive Biases for Disentangled Representation Learning with Correlated Treatment–Nuisance Factors

Luka Kovačević[1,2]* Sara-Jane Dunn[2] Hagen Triendl[2] Lindsay Edwards[2]
Sach Mukherjee[1] John Whittaker[1,2]†
Thomas Gaudelet[2]†

[1] MRC Biostatistics Unit, University of Cambridge, Cambridge, UK
[2] Relation Therapeutics, London, UK

Abstract

Accurately modelling experimental factors of variation is crucial to modern science. By understanding the distinct contributions of treatment and nuisance factors, researchers can better interpret, and generalise experimental findings. In many real-world experiments, treatment and nuisance factors are correlated, making standard assumptions of independence unrealistic. Classical *design of experiments* provides many approaches for mitigating confounding, yet their integration with modern deep generative models remains underexplored. We introduce a framework that adapts variational autoencoders (VAEs) with block design–inspired inductive biases to account for treatment–nuisance dependence. Specifically, we propose stop-gradient and independence-constraint mechanisms that respect experimental structure and enforce disentanglement even under correlated assignments. Our findings highlight both the promises and pitfalls of combining block design principles with disentangled generative modelling, paving the way for principled, causally informed use of deep learning in experimental sciences.

1 Introduction

Disentangling factors of variation is a fundamental problem in experimental science. Since the early development of statistical models in the 20th century (Fisher, 1949; Hill, 1965), researchers have sought to understand how experimental outcomes depend on changes in experimental conditions (Robins, 1997; Eberhardt and Scheines, 2007; Pearl, 2009).

Methods that learn *disentangled* representations have increasingly been applied to experimental data (Du et al., 2022; Lopez et al., 2023; Moinfar and Theis, 2024). The goal of disentangled representation learning is to map the underlying factors of variation in a dataset into latent variables that are both semantically and statistically independent (Bengio et al., 2013; Wang et al., 2024), and that, among other things (Locatello et al., 2019), *(i)* separate signal from nuisance variation (Kim et al., 2019; Tu et al., 2024), and *(ii)* provide a principled basis for constructing counterfactuals (Peters et al., 2017). These properties align with the aims of experiment design, where the goal is to isolate treatment effects while mitigating nuisance influences.

Modern machine learning methods handle nuisance variation *post-hoc*, by conditioning on the nuisance factor (Lopez et al., 2018) or imposing independence via the loss function (Tu et al., 2024; Makino et al., 2025; Adduri et al., 2025). These methods fall under *model-based* approaches to treatment-nuisance disentanglement. Conversely, block designs aim to separate treatment and nui-

*Correspondence: luka.kovacevic@mrc-bsu.cam.ac.uk
†Equal supervision.

Accepted to the NeurIPS 2025 Workshop on CauScien: Uncovering Causality in Science.

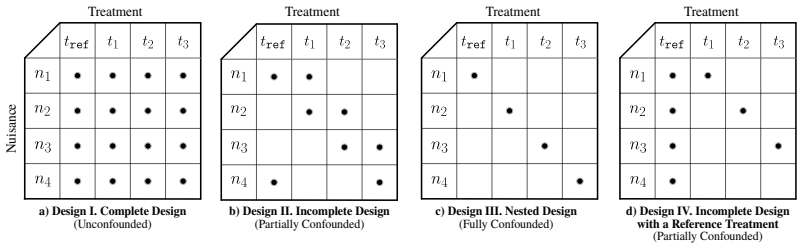


Figure 1: Classical block designs and treatment–nuisance confounding. Rows correspond to *nuisance* n_i and columns correspond to *treatment* t_i where $t_{\texttt{ref}}$ denotes the reference treatment. A filled dot (•) indicates that treatment t_j was observed with nuisance n_i.

sance effects with a *structure-based* approach[3]. Block designs should not replace optimal experiment design methods. Instead, we should use the well-establishing principles of block design to inform how and when inductive biases for disentangling treatment-nuisance effects are used.

Our contributions. In this work, we analyse independence in disentangled representations through the lens of block designs. Our specific contributions are: *(1)* We characterise when independence between treatments and nuisance factors is violated under classical block designs; *(2)* We propose inductive biases that adapt identifiable VAEs to data with correlated treatment-nuisance structure; *(3)* We empirically validate our approach using a real-world interventional dataset, showing improved disentanglement compared to baseline methods. These contributions highlight the importance of grounding representation learning methods in experimental design theory, thereby advancing the integration of disentanglement, causality, and scientific practice.

2 Background

Variational Autoencoders. VAEs (Kingma et al., 2013) are a class of deep generative models that perform amortized variational inference in latent variable models (see Appendix A for a full description). Given observed data $\mathbf{x} \in \mathcal{X}$, latent variable models aim to relate each $\mathbf{x}$ to latent $\mathbf{z} \in \mathcal{Z}$. The identifiable VAE (iVAE) framework (Khemakhem et al., 2020) provides identifiability guarantees on the standard VAE by conditioning on *auxiliary* variables. Formally, the conditional generative model is given by $p_\theta(\mathbf{x}, \mathbf{z}|\mathbf{u}) = p(\mathbf{x}|\mathbf{z}, \mathbf{u})p(\mathbf{z}|\mathbf{u})$, where $\mathbf{u}$ is an arbitrary auxiliary variable (e.g. treatment label). Under suitable conditions, this modification enables recovery of the true latent structure up to element-wise transformations.

Inductive Biases for Disentanglement. Many works aim to achieve unsupervised *dimension-wise disentanglement* of latent factors in VAEs where the components of $\mathbf{z} := [z_1, \ldots, z_p]^T \in \mathbb{R}^p$ are independent (Chen et al., 2016; Higgins et al., 2017; Kim and Mnih, 2018; Chen et al., 2018). Whilst this is an important problem for both structured and unstructured data, disentanglement of labelled factors of variation, or *supervised vector-wise disentanglement*, where two or more latent vectors $\mathbf{z}_1, \mathbf{z}_2 \in \mathbb{R}^p$ are mutually independent, is of particular relevance to the analysis of experimental data and – depending on the structure of the experiment and model – is a prerequisite to dimension-wise disentanglement[4]. The most prominent models for vector-wise disentanglement are the iVAE (Khemakhem et al., 2020), sparse VAE (Lachapelle et al., 2022), and their extensions (Lopez et al., 2023; Bereket and Karaletsos, 2023). Other approaches specific to the challenge of predicting transcriptional responses to cellular perturbations have also been proposed in the domain of computational biology. (Lotfollahi et al., 2021; Wu et al., 2024; Mao et al., 2024; Adduri et al., 2025), many of which assume an additive latent structure between latents[5].

Design of Experiments. In experimental design, the *experimental unit* is the subject to which a *treatment* or intervention is applied, while a *nuisance* factor is a source of confounding (e.g. experimental batch, time-of-day) that may obscure treatment effects (Dean et al., 2015). Block

[3]By ensuring that experimental units are allocated between factors in a structured way (Dean et al., 2015). We describe several block designs more formally in Section 2.

[4]Previous work has considered dimension-wise disentanglement with correlated factors, which can be viewed as a specific problem within disentanglement for nested block designs (Träuble et al., 2021).

[5]These models have weaker identifiability guarantees and thus require supervision in the form of statistical independence constraints or adversarial losses to enforce independence of treatment and nuisance factors.

designs aim to structurally prevent confounding of treatment-nuisance effects. In hierarchical experiments, nuisance factors can be organised into *groups* (e.g. clinical sites, laboratories).

In this work, we focus on the classical designs shown in Figure 1. These classical designs motivate our inductive biases: in nonlinear, high-dimensional settings, independence between treatments and nuisances should only be enforced where treatment and nuisance assignment are not confounded.

3 Methodology

We now describe how the iVAE can be adapted to block design experiments. Suppose that we have a dataset of triplets: $\mathcal{D} := \{(\mathbf{x}_i, t_i, u_i)\}_{i=1}^N$, with measurement, $\mathbf{x}_i \in \mathcal{X}$, treatment level, $t_i \in \mathcal{T}$, and nuisance label, $u_i \in \mathcal{U}$, corresponding to experimental unit *i*. The variational posteriors over treatment and nuisance latents are:

$$q_\phi(\mathbf{z}_t \mid t, \mathbf{x}) := \mathcal{N}(f_\mu^t(\mathbf{x}, t), f_\sigma^t(\mathbf{x}, t)), \qquad q_\phi(\mathbf{z}_u \mid u, \mathbf{x}) := \mathcal{N}(f_\mu^u(\mathbf{x}, u), f_\sigma^u(\mathbf{x}, u)),$$

with corresponding fully learnable priors

$$p_\theta(\mathbf{z}_t \mid t) := \mathcal{N}(f_\mu^t(t), f_\sigma^t(t)), \qquad p_\theta(\mathbf{z}_u \mid u) := \mathcal{N}(f_\mu^u(u), f_\sigma^u(u)).$$

Block Designs. We distinguish between five classical block designs (Figure 1; Appendix B). *Complete designs* (Design I) contain every treatment-nuisance pair, so treatment-nuisance effects are unconfounded. In *incomplete* and *nested designs* (Designs II, III), confounding arises depending on treatment-nuisance overlap. Design IV introduces a reference treatment across nuisance factors, which anchors nuisance effects.

While we focus on inductive bias for block designs with reference treatments variant (Design IV), we provide a more general formulation in Appendix C. The reference treatment variant is both more common in real experimental protocols and is more stable in experiments with sparse treatment-nuisance overlap, which includes our real-world dataset.

3.1 Stop-gradients with Reference Treatments

In incomplete designs (Designs II, IV), treatment and nuisance factors are correlated, leading to latent entanglement when conditioning on (t, u). A reference treatment, $t_{\texttt{ref}}$, is often present across nuisance factors. In this case, reference treatments, such as that seen in Figure 1d, provide a shared signal across nuisance factors. To exploit this, we allow $\mathbf{z}_u$ updates only for reference treatment samples:

$$\mathbb{E}_{\mathbf{z}}\left[\log p\left(\mathbf{x} | \mathbf{z}_t, \mathbf{z}_u^*\right)\right] \text{ where } \mathbf{z}_u^* = \begin{cases} \mathbf{z}_u & t_i = t_{\texttt{ref}}, \\ \mathrm{sg}(\mathbf{z}_u) & \text{otherwise}, \end{cases} \tag{1}$$

and where $\mathrm{sg}(\cdot)$ denotes the stop-gradient operator. **Intuition:** Only treatments observed across multiple nuisance factor levels carry information to separate treatment-nuisance effects. For non-reference treatments $\mathbf{z}_u$ is frozen to prevent it from absorbing treatment signal[6].

3.2 Independence Constraints with Reference Treatments

Independence penalties are often used to enforce disentanglement, but unconditional penalties are only valid for unconfounded designs (Design I). For incomplete designs (Designs II, IV), unconditional independence would contradict the experimental structure. Let $D(\mathbf{z}_t \perp \mathbf{z}_u)$ be a general unconditional independence criterion. Analogous to the stop-gradient variant, independence penalties can be restricted to reference treatment samples:

$$D_{\texttt{ref}}(\mathbf{z}_t \perp \mathbf{z}_u) = \begin{cases} D(\mathbf{z}_t \perp \mathbf{z}_u) & t = t_{\texttt{ref}}, \\ 0 & \text{otherwise}. \end{cases}$$

This ensures that nuisance disentanglement is anchored by the treatment shared across nuisance factors, while avoiding spurious independence constraints on confounded pairs. The training objective becomes $\mathcal{L}(\theta) = \mathcal{L}_{\mathrm{ELBO}}(\theta) + \lambda\, D_{\texttt{ref}}(\mathbf{z}_t \perp \mathbf{z}_u)$, with λ controlling the strength of the penalty. We implement independence constraints via the Hilbert-Schmidt independence criterion (HSIC) and apply an exponential moving average to stabilise updates across minibatches (see Appendix D).

[6] Stop-gradient operators have been shown to prevent cross-modal interference in multi-modal learning (Märtens and Yau, 2024).

Table 1: Performance of the baseline iVAE and its variants with different stop-gradient (sg) and independence constraint settings. *Ablated RMSE Rank* columns report performance when replacing $\mathbf{z}$ with its expectation $\mathbb{E}[\mathbf{z}]$, for different latent components. For each metric, (↑) indicates a higher value is better, and (↓) indicates a lower value is better. Model hyperparameters can be found in Appendix E.2.

sg(·)	Independence Constraint	RMSE (↓)	RMSE Rank (↓)	Ablated RMSE Rank		Treatment Factor Rank Importance (↑)
				Ablated latent (i.e. $\mathbf{z} := \mathbb{E}[\mathbf{z}]$)		
				$\mathbf{z}_t$ (↑)	$\mathbf{z}_u$ (↓)	
–	–	$\mathbf{0.5860 \pm 5.0 \times 10^{-3}}$	$\mathbf{0.1627 \pm 1.0 \times 10^{-2}}$	$0.1789 \pm 1.1 \times 10^{-2}$	$0.4763 \pm 3.7 \times 10^{-3}$	$0.0250 \pm 2.5 \times 10^{-3}$
✓	–	$0.6467 \pm 3.5 \times 10^{-3}$	$0.2670 \pm 7.6 \times 10^{-3}$	$0.3935 \pm 1.9 \times 10^{-2}$	$0.3599 \pm 1.3 \times 10^{-2}$	$0.5721 \pm 8.6 \times 10^{-2}$
–	$D(\mathbf{z}_t \perp \mathbf{z}_u)$	$0.5856 \pm 4.2 \times 10^{-3}$	$0.1625 \pm 7.1 \times 10^{-3}$	$0.1796 \pm 8.8 \times 10^{-3}$	$0.4762 \pm 3.8 \times 10^{-3}$	$0.0255 \pm 2.1 \times 10^{-3}$
–	$D_{\texttt{ref}}(\mathbf{z}_t \perp \mathbf{z}_u)$	$0.5875 \pm 4.6 \times 10^{-3}$	$0.1666 \pm 8.2 \times 10^{-3}$	$0.1830 \pm 8.9 \times 10^{-3}$	$0.4771 \pm 3.0 \times 10^{-3}$	$0.0240 \pm 2.5 \times 10^{-3}$
✓	$D(\mathbf{z}_t \perp \mathbf{z}_u)$	$0.6525 \pm 4.7 \times 10^{-3}$	$0.2833 \pm 9.3 \times 10^{-3}$	$0.4270 \pm 2.4 \times 10^{-2}$	$0.3483 \pm 1.5 \times 10^{-2}$	$0.7008 \pm 9.46 \times 10^{-2}$
✓	$D_{\texttt{ref}}(\mathbf{z}_t \perp \mathbf{z}_u)$	$0.6525 \pm 3.7 \times 10^{-3}$	$0.2778 \pm 8.1 \times 10^{-3}$	$\mathbf{0.4349 \pm 1.1 \times 10^{-2}}$	$\mathbf{0.3356 \pm 4.1 \times 10^{-3}}$	$\mathbf{0.7282 \pm 4.0 \times 10^{-2}}$

4 Evaluation

Dataset. The LINCS L1000 dataset (Lamb, 2006; Subramanian, 2017) is an interventional study measuring bulk gene expression across thousands of perturbations and multiple cell-lines. Data were collected with an incomplete block design (Appendix E). We condition on perturbation (treatment), batch (nuisance), and cell-type. Perturbations are confounded with batches, with $t_{\texttt{ref}}$ being present across all batches. Cell-types are consistently allocated across perturbations.

Metrics. To assess predictive performance we measure RMSE and the RMSE mean rank metric, which is a treatment-specific metric that measures how well the model discerns different treatments. A perfect score under the rank metric is 0, whilst predicting the outcome of a random treatment instead of the true treatment yields a score of 0.5. We also conduct sensitivity analysis using the *treatment factor rank importance* (T-FRI): $\text{FRI} := \frac{1}{|\mathcal{I}|}\sum_i \mathbb{I}\left\{\Delta_i^{(t)} > \Delta_i^{(u)}\right\}$, where $\Delta_i^{(k)}$ is the change in rank metric for treatment i when latent k is replaced with its expected value. The FRI measures the proportion of treatments for which the treatment latent is more important than the nuisance latent according to the rank metric. We provide mathematical descriptions of all metrics in Appendix E.3.

Results. In Table 1, the baseline iVAE achieves the lowest RMSE (0.5860) and rank error (0.1627). Applying only independence constraints does not affect overall performance. In contrast, introducing $\text{sg}(\cdot)$ degrades predictive accuracy, with RMSE rising to around 0.65 and rank error to 0.27–0.28. This pattern holds regardless of whether independence penalties are also applied. As expected, enforcing disentanglement more aggressively trades off predictive performance for stronger separation between latent factors.

Ablating $\mathbf{z}_t$ in the baseline increases rank error only modestly to 0.179, whereas ablating $\mathbf{z}_u$ causes a much larger jump to near random (0.476). This indicates that most predictive power is absorbed into the nuisance latent when no additional inductive bias is applied. With $\text{sg}(\cdot)$, ablating $\mathbf{z}_t$ has a much stronger effect (0.39–0.43), while ablating $\mathbf{z}_u$ has a weaker effect (0.33–0.36). FRI also shows that the models that do not use $\text{sg}(\cdot)$ are not able to disentangle treatment and nuisance. This may be due to the strength of nuisance effects, which even for the models that are more disentangled according to FRI, have similar ablated RMSE rank for both $\mathbf{z}_t$ and $\mathbf{z}_u$. The confidence intervals for $\text{sg}(\cdot)$ with $D(\cdot)$, and $\text{sg}(\cdot)$ alone are comparably wide. Conversely, for $\text{sg}(\cdot)$ with $D_{\texttt{ref}}(\cdot)$ the confidence interval for FRI is narrower, suggesting that the reference-only independence constraint is more stable across runs, with this combination outperforming all others on average.

5 Discussion

We show that classical block design theory provides a principled route to improving disentanglement in modern experimental datasets. Viewing representation learning through the lens of experimental design clarifies when the common assumption of independence between treatment and nuisance factors is violated. Our results indicate that structure-aware inductive biases—based on the actual relationship between treatment and nuisance factors—yield better disentanglement than approaches that assume orthogonality by default. This work is especially relevant to joint analyses of datasets with overlapping, partially overlapping, or disjoint interventions. It may also be relevant to foundation models that pool diverse assays, where nuisance factors correlate with intervention sets across studies, so enforcing unconditional independence can degrade separation of treatment and nuisance signals.

Several extensions follow naturally. First, early results suggest that latent-dimension dropout can further aid disentanglement. We leave a systematic study of latent dropout, and ELBO-motivated vector-wise biases, to future work. Second, while our analysis assumed a fixed design, practitioners often have partial control over block structure. Planning with these constraints may improve identifiability of treatment effects and offers a promising bridge between experimental design and causal representation learning. Finally, since LINCS L1000 exhibits strong nuisance effects in our experiments, we plan to expand our experiments to include a synthetic dataset, a single-cell interventional dataset and model baselines that use adversarial losses and latent additive architectures.

References

Abhinav Adduri, Dhruv Gautam, Beatrice Bevilacqua, Alishba Imran, Rohan Shah, Mohsen Naghipourfar, Noam Teyssier, Rajesh Ilango, Sanjay Nagaraj, Chiara Ricci-Tam, et al. Predicting cellular responses to perturbation across diverse contexts with state. *bioRxiv*, pages 2025–06, 2025.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

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.

Ricky TQ Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. *Advances in neural information processing systems*, 31, 2018.

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in neural information processing systems*, 29, 2016.

Angela M Dean, Max Morris, John Stufken, and Derek Bingham. *Handbook of design and analysis of experiments*, volume 7. CRC Press Boca Raton, FL, USA:, 2015.

Yuanqi Du, Xiaojie Guo, Yinkai Wang, Amarda Shehu, and Liang Zhao. Small molecule generation via disentangled representation learning. *Bioinformatics*, 38(12):3200–3208, 2022.

Frederick Eberhardt and Richard Scheines. Interventions and causal inference. *Philosophy of science*, 74(5):981–995, 2007.

Ronald A Fisher. The design of experiments. 1949.

Thomas Gaudelet, Alice Del Vecchio, Eli M Carrami, Juliana Cudini, Chantriolnt-Andreas Kapourani, Caroline Uhler, and Lindsay Edwards. Season combinatorial intervention predictions with salt & peper. *arXiv preprint arXiv:2404.16907*, 2024.

Arthur Gretton, Kenji Fukumizu, Choon Teo, Le Song, Bernhard Schölkopf, and Alex Smola. A kernel statistical test of independence. *Advances in neural information processing systems*, 20, 2007.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International conference on learning representations*, 2017.

Austin Bradford Hill. The environment and disease: association or causation?, 1965.

Ilyes Khemakhem, Diederik Kingma, Ricardo Monti, and Aapo Hyvarinen. Variational autoencoders and nonlinear ica: A unifying framework. In *International conference on artificial intelligence and statistics*, pages 2207–2217. PMLR, 2020.

Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *International conference on machine learning*, pages 2649–2658. PMLR, 2018.

Minyoung Kim, Yuting Wang, Pritish Sahu, and Vladimir Pavlovic. Relevance factor vae: Learning and identifying disentangled factors. *arXiv preprint arXiv:1902.01568*, 2019.

Diederik P Kingma, Max Welling, et al. Auto-encoding variational bayes, 2013.

Sébastien Lachapelle, Pau Rodriguez, Yash Sharma, Katie E Everett, Rémi Le Priol, Alexandre Lacoste, and Simon Lacoste-Julien. Disentanglement via mechanism sparsity regularization: A new principle for nonlinear ica. In *Conference on Causal Learning and Reasoning*, pages 428–484. PMLR, 2022.

Justin et al. Lamb. The connectivity map: using gene-expression signatures to connect small molecules, genes, and disease. *Science*, 313(5795):1929–1935, 2006.

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *international conference on machine learning*, pages 4114–4124. PMLR, 2019.

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

Romain Lopez, Natasa Tagasovska, Stephen Ra, Kyunghyun Cho, Jonathan Pritchard, and Aviv Regev. Learning causal representations of single cells via sparse mechanism shift modeling. In *Conference on Causal Learning and Reasoning*, pages 662–691. PMLR, 2023.

Mohammad Lotfollahi, Anna Klimovskaia Susmelj, Carlo De Donno, Yuge Ji, Ignacio L Ibarra, F Alexander Wolf, Nafissa Yakubova, Fabian J Theis, and David Lopez-Paz. Compositional perturbation autoencoder for single-cell response modeling. *BioRxiv*, 2021.

Taro Makino, Ji Won Park, Natasa Tagasovska, Takamasa Kudo, Paula Coelho, Jan-Christian Huetter, Heming Yao, Burkhard Hoeckendorf, Ana Carolina Leote, Stephen Ra, et al. Supervised contrastive block disentanglement. *arXiv preprint arXiv:2502.07281*, 2025.

Haiyi Mao, Romain Lopez, Kai Liu, Jan-Christian Huetter, David Richmond, Panayiotis Benos, and Lin Qiu. Learning identifiable factorized causal representations of cellular responses. *Advances in Neural Information Processing Systems*, 37:121630–121669, 2024.

Kaspar Märtens and Christopher Yau. Disentangling shared and private latent factors in multimodal variational autoencoders. In *Machine Learning in Computational Biology*, pages 60–75. PMLR, 2024.

Amir Ali Moinfar and Fabian J Theis. Unsupervised deep disentangled representation of single-cell omics. *bioRxiv*, pages 2024–11, 2024.

Judea Pearl. *Causality*. Cambridge university press, 2009.

Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of causal inference: foundations and learning algorithms*. The MIT press, 2017.

James M Robins. Causal inference from complex longitudinal data. In *Latent variable modeling and applications to causality*, pages 69–117. Springer, 1997.

Aravind et al. Subramanian. A next generation connectivity map: L1000 platform and the first 1,000,000 profiles. *Cell*, 171(6):1437–1452, 2017.

Frederik Träuble, Elliot Creager, Niki Kilbertus, Francesco Locatello, Andrea Dittadi, Anirudh Goyal, Bernhard Schölkopf, and Stefan Bauer. On disentangled representations learned from correlated data. In *International conference on machine learning*, pages 10401–10412. PMLR, 2021.

Xinming Tu, Jan-Christian Hütter, Zitong Jerry Wang, Takamasa Kudo, Aviv Regev, and Romain Lopez. A supervised contrastive framework for learning disentangled representations of cellular perturbation data. In *Machine Learning in Computational Biology*, pages 90–100. PMLR, 2024.

Xin Wang, Hong Chen, Si'ao Tang, Zihao Wu, and Wenwu Zhu. Disentangled representation learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):9677–9696, 2024.

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.

A Variational Autoencoders

VAEs define a joint distribution over observed and latent variables, $p_\theta(\mathbf{x}, \mathbf{z}) = p_\theta(\mathbf{x} \mid \mathbf{z})p(\mathbf{z})$, where the likelihood $p_\theta(\mathbf{x} \mid \mathbf{z})$ is typically parametrised by a neural network decoder, and the prior over latents is chosen as a standard normal, $p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$. Inference is approximated via a variational posterior $q_\phi(\mathbf{z} \mid \mathbf{x})$, parametrised by an encoder network. Training proceeds by maximizing the evidence lower bound (ELBO), given by

$$\mathcal{L}(\mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x} \mid \mathbf{z})] - \mathbb{D}_{\mathrm{KL}}(q_\phi(\mathbf{z} \mid \mathbf{x}) \,||\, p(\mathbf{z})),$$

which balances reconstruction accuracy with regularization towards the prior via the Kullback–Leibler divergence.

B Classical Block Designs

In experimental design, a nuisance factor is a source of variability that is not of primary scientific interest but may influence the response. When such a factor can be explicitly controlled, it can be treated as a blocking factor. A block is then the group of experimental units sharing the same level(s) of the blocking factor, within which treatments are compared. Blocking thus reduces the impact of nuisance variation by ensuring that treatment contrasts are made within relatively homogeneous groups. Not all nuisance factors are suitable for blocking; some must instead be addressed by randomization or other design strategies. For example, in a cell culture experiment, measurements taken on different days may systematically differ because of incubator conditions; by treating day as a blocking factor and applying all drug treatments within each day, treatment comparisons are protected from day-to-day variation.

Design I: Complete Block Designs. Every treatment-nuisance pair appears, so treatment and nuisance effects are unconfounded. The unconditional constraint $\mathbf{z}_u \perp \mathbf{z}_t$ can therefore be applied (see Figure 1a).

Design II: Incomplete Block Designs. Not every treatment-nuisance pairs appears, which leads to partial confounding between treatment and nuisance effects. In this case, the unconditional constraint $\mathbf{z}_u \perp \mathbf{z}_t$ is invalid (see Figure 1b).

Design III: Nested Designs. Each treatment occurs under only one nuisance factor. Treatment and nuisance effects are fully confounded and cannot be separated without external information. Since nuisance factor assignment depends on treatment, any disentanglement constraint or inductive bias would violate the experimental design (see Figure 1c).

Design IV: Incomplete Designs with a Reference Treatment. All treatments appear under only one nuisance factor, except for a designated reference treatment $t_{\texttt{ref}}$. Nuisance effects can be identified using $t_{\texttt{ref}}$, under the assumption that it provides a consistent anchor across nuisance factors (see Figure 1d).

C Additional Inductive Biases

C.1 Stop-gradients on Overlapping Treatment–Nuisance Pairs

To disentangle $\mathbf{z}_u$ and $\mathbf{z}_t$ we can trigger nuisance latent updates only for treatments that provide cross-block overlap.

Let $\mathcal{U}(t)$ be the set of nuisance factors with which treatment t appears, and define the overlap indicator $r(t) = \mathbb{I}\{\, |\mathcal{U}(t)| \geq 2 \,\}$. We then modify the iVAE expected log-likelihood as

$$\mathbb{E}_{\mathbf{z}}[\log p(\mathbf{x} \mid \mathbf{z}_t, \mathbf{z}_u^*)] \quad \text{where} \quad \mathbf{z}_u^* = \begin{cases} \mathbf{z}_u & r(t_i) = 1 \;\text{ and }\; u_i \in \mathcal{U}(t_i), \\ \mathrm{sg}(\mathbf{z}_u) & \text{otherwise}, \end{cases} \tag{2}$$

and $\mathrm{sg}(\cdot)$ denotes the stop-gradient operator. Thus, for treatments confined to a single nuisance level, $\mathbf{z}_u$ is frozen to prevent it from absorbing treatment signal.

C.2 Independence Constraints on Overlapping Treatment–Nuisance Pairs

Let $D(\mathbf{z}_t \perp \mathbf{z}_u)$ be a dependence penalty (e.g. Hilbert-Schmidt independence criterion or adversarial models). We restrict it to samples with overlapping treatment-nuisance structures. Thus we have,

$$D^*(\mathbf{z}_t \perp \mathbf{z}_u) = \begin{cases} D(\mathbf{z}_t \perp \mathbf{z}_u) & \text{where } \; r(t_i) = 1 \; \text{ and } \; u_i \in \mathcal{U}(t_i), \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

D Unbiased HSIC Estimation with EMA Smoothing

We regularize independence between $\mathbf{z}_u$ and $\mathbf{z}_t$ using the unbiased small-sample Hilbert–Schmidt Independence Criterion (HSIC), which is valid for $m \geq 4$ (Gretton et al., 2007).

To stabilise the stochastic estimate over minibatches, we maintain an exponential moving average (EMA) buffer

$$\overline{h} \leftarrow \alpha\, \overline{h} + (1-\alpha)\bar{h}_{\text{prev}}$$

with decay $\alpha = 0.9$, and include an additional EMA-weighted penalty $L_{\text{HSIC}}^{\text{EMA}}$ in the loss (both terms share the same scalar weight). Class-wise HSIC is only computed when the subset has more than a minimum number of samples (threshold > 4 in our code), ensuring the unbiased estimator is well-defined.

Algorithm 1 Unbiased HSIC with EMA for Treatments and Blocks

Input: batch $(\mathbf{t}, \mathbf{u}) \in \mathbb{R}^{2\times\ell}$, previous EMA $\bar{h}_{\text{prev}}$, decay $\alpha \in (0,1)$, weight λ_{HSIC}
Output: $\bar{h}$ (updated EMA estimate), $\mathcal{L}_{\text{HSIC}}$ (λ-weighted EMA estimate)

$\mathbf{z}_t \leftarrow f_\mu^t(\mathbf{x}, \mathbf{t})$ ▷ treatment latent
$\mathbf{z}_u \leftarrow f_\mu^b(\mathbf{x}, \mathbf{u})$ ▷ nuisance latent
$\hat{h} \leftarrow \text{HSIC}(\mathbf{z}_t, \mathbf{z}_u)$ ▷ compute unbiased HSIC
$\bar{h}_{\text{new}} \leftarrow \alpha\, \bar{h}_{\text{prev}} + (1-\alpha)\, \hat{h}$ ▷ update EMA
$L_{\text{HSIC}} \leftarrow \lambda_{\text{HSIC}} \cdot \bar{h}$ ▷ Weight new EMA estimate
return $\left(\bar{h}_{\text{new}},\ L_{\text{HSIC}}\right)$

E Evaluation Framework

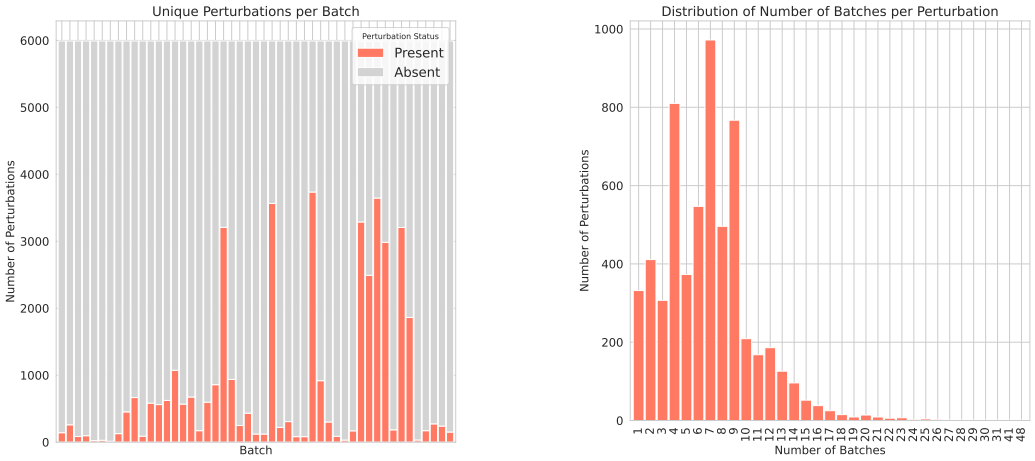
E.1 LINCS L1000 dataset.

The LINCS L1000 dataset (Lamb, 2006; Subramanian, 2017) is a large-scale transcriptional profiling resource generated within the NIH Library of Integrated Network-Based Cellular Signatures (LINCS) program. The L1000 assay measures the bulk expression of 978 carefully selected "landmark" genes, which are sufficient to capture the majority of variation in cellular transcriptional states. This approach enables cost-efficient, high-throughput profiling of cellular responses to a broad range of perturbations, including small molecules and genetic interventions, across multiple human cell lines. The resulting dataset comprises millions of gene expression signatures and serves as a widely used reference for studying perturbation biology, drug mechanisms of action, and gene regulatory networks.

The design of the LINCS L1000 dataset falls into the category of incomplete block design with a reference treatment introduced in Appendix B. Each experiment involves two biological sources of variation: perturbations and cell-lines. In our experiments we treat perturbations as the treatment and cell-line labels as covariates.

Each experiment involves multiple nuisance factors, commonly referred to as *batches* in biology, which must be taken into account when estimating perturbation effects. Because not every perturbation is profiled in every possible combination of cell line, and batch, the dataset constitutes an incomplete block design: each block (e.g., a plate or batch) contains only a subset of perturbations. However, each block contains at least the same control (e.g. DMSO for small molecules).

Due to its scale and design, the LINCS L1000 provides us with a real-world test-bed for evaluating our framework for disentangled representation learning.



(a) **(b)**

Figure 2: (a) The number of unique perturbations present in each batch. (b) The distribution of the number of batches that each unique perturbation appears in.

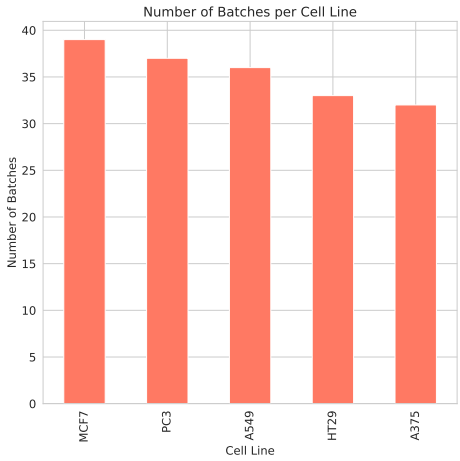


Figure 3: The number of batches in which each cell-line is present.

Preprocessing. The LINCS consortium distributes the L1000 dataset at 5 levels of processing, reflecting increasing degrees of normalization and inference. We downloaded the level 3 which provides normalised expression values for the 978 landmark genes across perturbations and control conditions, correcting for plate- and batch-specific effects. This preserves the treatment–block structure while harmonizing measurement scale. We apply gene-wise normalisation, which is standard practice for perturbation experiments. For gene i, we compute mean, μ_i^{ctl}, and standard deviation, σ_i^{ctl} of expression under control samples, and apply normalisation:

$$\tilde{x}_i := \frac{x_i - \mu_i^{\text{ctl}}}{\sigma_i^{\text{ctl}}}.$$

Filtering. For simplicity, we restrict ourselves to small-molecule perturbations, which constitute the majority of the LINCS dataset. Summary statistics illustrating the incomplete block design of LINCS L1000 are given for perturbations and batches in Figure 2 and for cell-lines and batches in Figure 3. Furthermore, we filter to a common set of cell lines across retained treatments so every perturbation in the dataset has at least one sample for every cell-line.

E.2 Model Implementation

Our dataset is comprised of tetrads:

$$\mathcal{D} := \{(\mathbf{x}_i, p_i, c_i, b_i)\}$$

with gene expression, $\mathbf{x}_i$, perturbation label p_i, cell-type label c_i, and batch label b_i. Our *treatments* of interest are the perturbations. Cell-types are an important factor of variation so we always condition on cell-type, however, they are not the main interest of the experiment and are not a nuisance factor since they are well-spread across perturbations. Batches are our *nuisance* factors as perturbation and batch assignment are partially confounded in the incomplete block design of LINCS L1000.

We define variational posteriors over perturbation (treatment), batch (nuisance), and cell-line latents:

$$q_\phi(\mathbf{z}_p|p, \mathbf{x}) := \mathcal{N}(f^p_\mu(\mathbf{x}, p), f^p_\sigma(\mathbf{x}, p)), \quad q_\phi(\mathbf{z}_b|b, \mathbf{x}) := \mathcal{N}(f^b_\mu(\mathbf{x}, b), f^b_\sigma(\mathbf{x}, b)),$$
$$\text{and,} \quad q_\phi(\mathbf{z}_c|c, \mathbf{x}) := \mathcal{N}(f^c_\mu(\mathbf{x}, c), f^c_\sigma(\mathbf{x}, c)),$$

with learnable priors,

$$p_\theta(\mathbf{z}_p|p) := \mathcal{N}(f^p_\mu(p), f^p_\sigma(p)), \quad p_\theta(\mathbf{z}_b|b) := \mathcal{N}(f^b_\mu(b), f^b_\sigma(b)),$$
$$\text{and,} \quad p_\theta(\mathbf{z}_c|c) := \mathcal{N}(f^c_\mu(c), f^c_\sigma(c)).$$

Each $f(\cdot)$ is an MLP consisting of two hidden layers with 64 units each and `tanh` activation function with a linear additive term. The latent space had a dimension of 256. Perturbations are represented as learned 160-dimensional embeddings. The model was trained with a batch size of 256 using the Adam optimiser (learning rate = 0.001, weight decay = 1×10^{-5}). Early stopping was employed based on the validation perturbation-wise RMSE with a patience of 75 epochs. Training was run a single Ampere 24GB GPU for up to 500 epochs. Where HSIC penalties are appled we set $\lambda = 100$.

E.3 Metrics

In our evaluation, we first average the predicted and observed expression profiles across replicates for each perturbation-cell-type pair $(\mathbf{p}, \mathbf{c})$. We then compute the Root Mean Squared Error (RMSE) for each perturbation within a given cell-type (Gaudelet et al., 2024; Wu et al., 2024),

$$\text{RMSE}(\mathbf{p}, \mathbf{c}) = ||\hat{\mathbf{x}}^{(\mathbf{p},\mathbf{c})} - \mathbf{x}^{(\mathbf{p},\mathbf{c})}||_2, \tag{4}$$

where $\hat{\mathbf{x}}^{(\mathbf{p},\mathbf{c})}$ and $\mathbf{x}^{(\mathbf{p},\mathbf{c})}$ are the predicted and observed average expression vectors, under perturbation $\mathbf{p}$ and cell-type $\mathbf{c}$. We report the average RMSE across all perturbations:

$$\text{RMSE}_{\text{avg}}(\mathbf{c}) = \frac{1}{|\mathcal{P}|} \sum_{\mathbf{p} \in \mathcal{P}} \text{RMSE}_{\text{avg}}(\mathbf{p}, \mathbf{c}). \tag{5}$$

Given a distance metric $\text{D}(\mathbf{x}_i, \mathbf{x}_j)$ between two vectors $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}$, the rank metric (Wu et al., 2024) measures the fraction of predictions that are closer to the average across true expression vectors $\mathbf{x}^{(\mathbf{p},\mathbf{c})}$ than the average across predictions $\hat{\mathbf{x}}^{(\mathbf{p},\mathbf{c})}$,

$$\text{rank}(\hat{\mathbf{x}}^{(\mathbf{p}=i,\mathbf{c})}; \mathbf{c}) = \frac{1}{|\mathcal{P}| - 1} \sum_{\substack{1 \le j \le |\mathcal{P}| \\ i \ne j}} \mathbb{I}\{\text{D}(\hat{\mathbf{x}}^{(j,\mathbf{c})}, \mathbf{x}^{(i,\mathbf{c})}) \le \text{D}(\hat{\mathbf{x}}^{(i,\mathbf{c})}, \mathbf{x}^{(i,\mathbf{c})})\},$$

where we use the RMSE as our distance metric. We then obtain the average rank across perturbations for a given cell label,

$$\text{rank}_{\text{avg}}(\mathbf{c}) = \frac{1}{|\mathcal{P}|} \sum_{1 \le i \le |\mathcal{P}|} \text{rank}(\hat{\mathbf{x}}^{(i,\mathbf{c})}; \mathbf{c}).$$

To quantify how well the model separates treatment effects from nuisance variation, we introduce the *treatment factor rank importance (FRI)*. We begin by computing the perturbation-wise rank metric for three model variants:

1. **Unablated model:** full latent representations;
2. **Perturbation-ablated model:** where the perturbation latent is replaced with its expected value across all perturbation labels;

3. **Batch-ablated model:** where the batch latent is replaced with its expected value across all perturbation labels.

For each perturbation i, we calculate the change in rank score when ablating a latent variable:

$$\Delta_i^{(k)} = \mathrm{rank}^{(k)}(\hat{\mathbf{x}}^{(i,\mathbf{c})}; \mathbf{c}) - \mathrm{rank}(\hat{\mathbf{x}}^{(i,\mathbf{c})}; \mathbf{c}),$$

where $k \in \{\mathrm{perturbation}, \mathrm{batch}\}$. Intuitively, $\Delta_i^{(k)}$ measures how much predictive accuracy depends on latent k.

The final FRI compares the relative importance of perturbation (treatment) versus batch (nuisance) latents:

$$\mathrm{FRI} = \frac{1}{|\mathcal{T}|} \sum_{i \in \mathcal{T}} \mathbb{I}\{\Delta_i^{(\mathbf{p})} > \Delta_i^{(\mathbf{b})}\}.$$

Thus, FRI reflects the fraction of treatments for which the perturbation latent contributes more to predictive accuracy than the batch latent. A higher score indicates better disentanglement of true treatment effects from nuisance variation.