Design-of-Experiments for Nonlinear, Multivariate Biology: Rethinking Experimental Design through Perturb-seq

Yuji Okano[1,†], Tetsuo Ishikawa[1], Yasunori Sato[2], Hideyuki Okano[3,4], and Kazuhiro Sakurada[1]

[1] Department of Extended Intelligence for Medicine, The Ishii-Ishibashi Laboratory, Keio University School of Medicine; 35 Shinanomachi, Shinjuku-Ku, Tokyo 160-8582, Japan
[2] Department of Biostatistics, Keio University School of Medicine; 35 Shinanomachi, Shinjuku-Ku, Tokyo 160-8582, Japan
[3] Keio University Regenerative Medicine Research Center; 3-25-10 Tonomachi, Kawasaki-Ku, Kawasaki, Kanagawa 210-0821, Japan
[4] Division of CNS Regeneration and Drug Discovery, International Center for Brain Science, Fujita Health University; 1-98 Dengakugakubo, Kutsukake-Cho, Toyoake, Aichi 470-1192, Japan
[†] Corresponding author: yujiokano@keio.jp

Abstract

Design of experiments (DOE) principles are increasingly applied to biological assays, yet it remains unclear whether their foundational assumption—orthogonal decomposition—holds in nonlinear biological systems.

We addressed this question using Perturb-seq as a case study. By benchmarking a design commonly used in Perturb-seq and related experiments against the orthogonal Plackett–Burman (PB) design via simulations, we uncovered a counter-intuitive phenomenon: while orthogonal designs generally excel, the multicollinearity inherent to the common design is functionally advantageous in systems with significant signal amplification. This challenges the blind application of DOE to biology.

Based on these findings, we developed the PB suitability index (PBSI), a simple, parameter-free metric that predicts the optimal design solely from network structure. Our work not only provides practical guidelines for Perturb-seq but also establishes a "biology-oriented DOE" framework, bridging the gap between statistical rigor and biological complexity.

1 Introduction

Biological experimentation involves extracting *events* (one-off occurrences) from *processes* (continuous sequences of changes driven by scale interactions) and interpreting these events through the causal inference schemes: correlation, intervention, and counterfactuals [1]. Recent progress in high-throughput quantitative measurements has dramatically expanded the scale of this causal inquiry, necessitating multivariate approaches.

In many multivariate biological experiments, a design that tests one factor at a time against a baseline—which we previously termed the "Control-and-Leave-One-Out (C+LOO) design" [2]—have long served as the standard approach for verifying causal effects. Despite its practical convenience, the C+LOO strategy and its variants are traditionally criticized for their limited capacity to detect interactions [3]. Therefore, the integration of multivariate biological assays and DOE—a statistical framework designed to balance feasibility and resolution under experimental constraints [4]—is gaining increasing attention [5].

However, while DOE provides a principled mathematical foundation—notably orthogonality and decomposability—it remains uncertain to what extent these properties hold in nonlinear, open, and nonequilibrium biological systems [6]. In other words, although DOE is increasingly invoked in modern biology, whether its classical principles can be seamlessly transplanted to nonlinear biological contexts without theoretical reassessment remains an open question. Given that biological experiments often extend to realms where discrete events and continuous processes conceptually fuse together, an epistemological classification of biological contexts—clarifying what can be effectively demonstrated by causal modeling—is essential before applying statistical tools.

To address this gap, we establish two complementary lines of inquiry. First, we examine whether experimental designs used in multivariate biological assays can be systematically extended and analyzed using DOE principles. Second, we evaluate how far the foundational assumption of orthogonal decomposability, central to the DOE dis-

cipline, remains meaningful when the generative process is nonlinear.

As an initial contribution toward a biology-oriented DOE framework, we analyze (i) statistical properties of the C+LOO design, (ii) their impacts on the result interpretation, and (iii) their benchmarking against the PB design [4], focusing on Perturb-seq—a CRISPR-based single-cell perturbation assay [7, 8]—as a prototypical case where the C+LOO design is widely employed [2]. We also provide an experimental design guideline for Perturb-seq to support understanding and comparison of experimental designs.

2 Results

2.1 Inherent multicollinearity of the C+LOO design is sensitive to the number of factors n

Having one control trial and n knockdown (KD) trials, the C+LOO design for n factors ($\forall n \in \mathbb{N}$)—which we denote $C(n)$—consists of $n+1$ trials (see Methods 4.1.1 for details). As an example, $C(9)$ is shown in Figure 1a.

By visualizing the absolute values of the Pearson correlation coefficients ρ among the main effects and interaction terms of $C(9)$ (Figure 1b), we confirmed a distinctive property of the C+LOO design: generally weak correlations among main effects and interaction terms [2].

To examine the behavior for general n, we derived closed-form expressions for the correlation coefficients of generic main effects $\mathbf{x}_i$ and interaction terms $\mathbf{x}_{i\times j}$ (Methods 4.1.4–4.1.5 and Supplementary Information (SI) A–B). As illustrated in Figure 1c for $2 \leq n \leq 12$, some absolute correlations decrease with n whereas others increase. While this "trade-off" structure achieves seamless high factor correlations for all n, the detailed composition differs according to the value of n. In other words, both the magnitude and the pattern of factor correlations in the C+LOO design vary with the number of factors n.

Notably, $\rho_{i\times j,i}$ and $\rho_{i\times j,i\times l}$ converge to positive values as n increases, suggesting that C+LOO becomes increasingly susceptible to these influences as the factor count grows. In sharp contrast, the well-known full factorial (FF) and PB designs ensure $\rho_{i\times j,i} = 0$ (Methods 4.1.2–4.1.3 and SI C), underscoring this as a characteristic feature of C+LOO.

We further found that the n-dependent evolution of these correlations exacerbates multicollinearity. For comparison, we instantiated the PB design—requiring a similar total number of runs as C+LOO for each n (Figure 1d and Methods 4.1.3)—and evaluated the condition number and variance inflation factor (VIF). Whereas PB remained constant across n, the C+LOO design showed clear increases with n (Figures 1e–1f).

In summary, the C+LOO design is characterized by pronounced multicollinearity and by n-dependent shifts in the trade-off structure of factor correlations. The latter poses a serious pitfall for comparative studies of experimental designs. Although benchmarking with real single-cell RNA-sequencing (scRNA-seq) datasets is standard practice in comparing scRNA-seq methodologies, the statistical properties of C+LOO change with the number of target genes, making broadly generalizable conclusions difficult. Thus, rather than relying solely on inductive comparisons from individual datasets, one should seek more abstract, design-level principles—grounded in theory and universality—to understand how experimental design influences inference.

2.2 Simulated models revealed strengths and weaknesses of the C+LOO design in Perturb-seq data analysis deriving from its design and multicollinearity

Given the unique multicollinearity patterns inherent in the C+LOO design, further investigation is essential to understand how these features influence the conclusions drawn from Perturb-seq experiments.

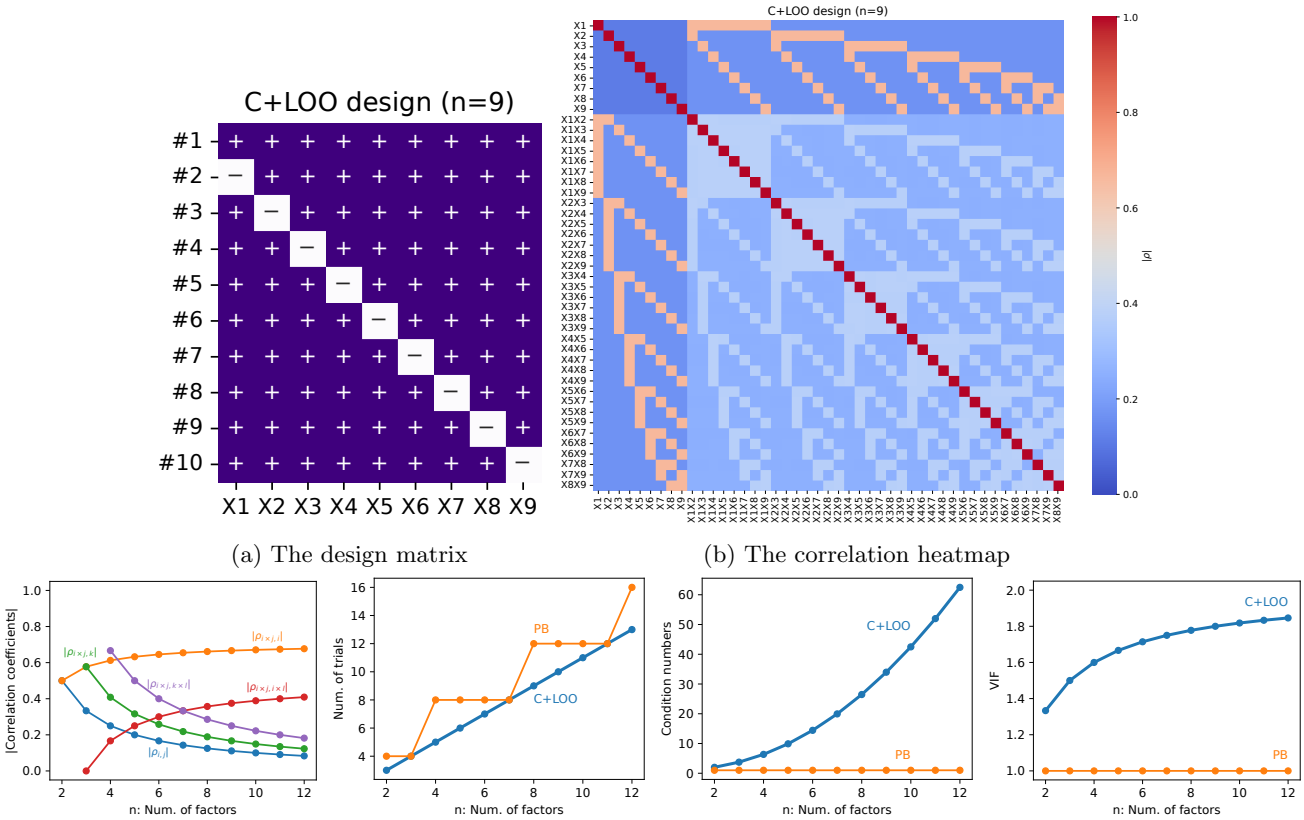
To evaluate the influence of experimental design on the interpretation of Perturb-seq results, it is essential to formally define its underlying logical framework and the biological phenomena it addresses. We established the following formalization to delineate the boundaries of the research domain where experimental design can intervene:

1. **Quantification Framework**: Conventional Perturb-seq (using the C+LOO design) quantifies phenotypes resulting from individual gene KD by summarizing them into scalar values (e.g., single-gene markers or phenotypic scores) and analyzing the difference relative to a baseline condition (Figure 2).

2. **Nature of the Causal Model**: This methodology relies on a discrete causal model between the perturbation and the phenotype, treating the detailed, intermediate mechanism as a black box (Methods 4.2 and SI D.1).

3. **Limitation of Applicability**: Due to the observational "blank time" inherent in Perturb-seq, the model's application is restricted to systems where the gene regulatory network (GRN) structure is static (time-invariant). Discussions regarding the efficiency of experimental design are therefore only valid under this constraint (Methods 4.2 and SI D.1).

Based on this formalization, we devised a methodology to validate the impact of experimental design. Since empirical analysis using real data presents limitations for



(a) The design matrix (b) The correlation heatmap

(c) Trajectories of $|\rho|$ (d) Numbers of trials (e) Condition numbers (f) VIF

Figure 1: **The statistical properties of the C+LOO design depend on the number of factors n**

a: Design matrix of the C+LOO design for nine factors (+, addition of factors; −, subtraction of factors). The rows (#1–#10) represent trials and the columns (X1–X9) represent the factors. **b**: Heatmap of the absolute correlation coefficients (denoted as $|\rho|$) among main effects (X1–X9) and interaction terms (e.g., X1X2$\cdots$) when they are investigated under the design matrix. **c**: $|\rho|$ for C+LOO designs of 2–12 factors ($\rho_{i,j}$ between two different main effects $\mathbf{x}_i$ and $\mathbf{x}_j$, blue; $\rho_{i\times j,i}$ between an interaction term $\mathbf{x}_{i\times j}$ and one of its corresponding main effects $\mathbf{x}_i$, orange; $\rho_{i\times j,k}$ between an interaction $\mathbf{x}_{i\times j}$ and a non-corresponding main effect $\mathbf{x}_k$, green; $\rho_{i\times j,i\times l}$ between two interaction sharing one main effect such as $\mathbf{x}_{i\times j}$ and $\mathbf{x}_{i\times l}$, red; and $\rho_{i\times j,k\times l}$ between two interaction terms with no shared main effect such as $\mathbf{x}_{i\times j}$ and $\mathbf{x}_{k\times l}$, purple) for $\forall n \in \mathbb{N}$, indices $i,j,k,l \in \mathbb{N}$ such that $1 \leq i,j,k,l \leq n$ and all indices are distinct. Notably, here we omitted $\rho_{i,i}$ and $\rho_{i\times j,i\times j}$ since they are constant. **d–f**: The numbers of trials (d), condition numbers (e), and VIF (f) for the C+LOO (blue) and PB (orange) designs of 2–25 factors. These results demonstrate that the statistical structure of the C+LOO design is not fixed but evolves with n, undermining naive design benchmarking across studies with different target sizes.

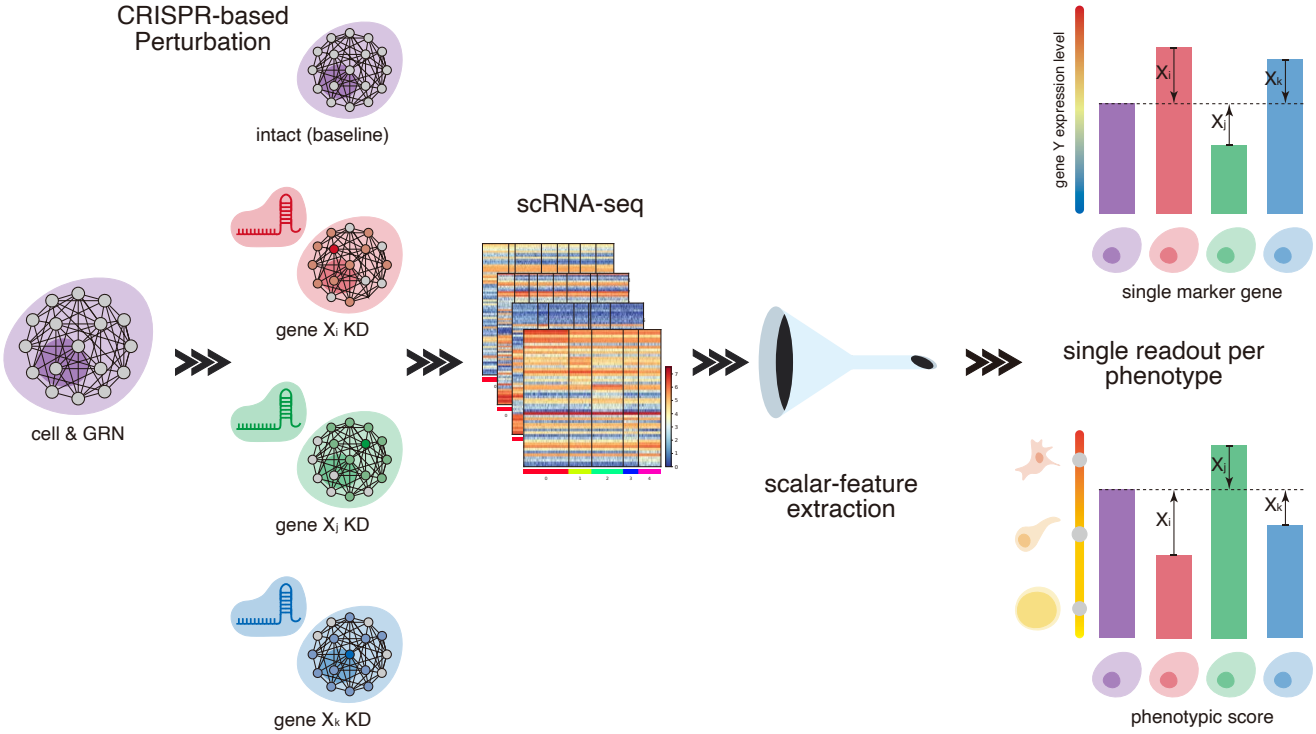


Figure 2: **Standard procedure for Perturb-seq experiments and data analysis**

Perturb-seq is a combinatorial approach that couples CRISPR-based screening with scRNA-seq [7, 8]. Using a C+LOO design, one-gene-at-a-time KD models are constructed, and changes in gene-expression patterns are profiled by scRNA-seq. The intact condition (purple cell) serves as the baseline, and each KD model (pink, green, blue cells) is evaluated in a pairwise comparison (e.g., Dunnett's test [9]) against this baseline.

To enhance interpretability for a phenotype, the readout is often (i) single-gene markers—commonly used proxies for specific phenotypes, including lineage markers (e.g., *HES1*), master regulators (e.g., *NANOG*, *FOXP3*), and oncogenes (e.g., *MYC*, *KRAS*)—or (ii) phenotypic scores that aggregate multiple marker genes (e.g., via SCENIC [10]). These readouts share a common property: they are scalar. As an example, the right panels illustrate pairwise comparisons to the baseline: the top-right uses the expression level of gene Y as the readout, and the bottom-right uses a phenotypic score. In both cases, comparisons are shown for genes X_i (pink), X_j (green), and X_k (blue).

Thus, a standard Perturb-seq workflow can be abstracted as follows: from scRNA-seq data acquired under a C+LOO design, extract a phenotype-aligned scalar feature and assess the KD-induced difference by pairwise comparison to the baseline.

In this study, to examine how the experimental design and scalar readout shape downstream biological inference, we explicitly consider the underlying GRN as a mediating variable. For further details, see Methods 4.2 and SI D.1.

constructing systematic knowledge (SI D.2), we developed a minimalist simulator that extracts only the essential elements of Perturb-seq (Methods 4.3): (i) a static GRN, (ii) nonlinear interactions based on a specified true GRN structure, and (iii) scalar output. Through this generalized model, we aim to derive universal principles governing the relationship between experimental design and result interpretation, ultimately providing guiding principles for decision-making in future Perturb-seq experiments.

Based on this framework, we investigated the impact of the inherent multicollinearity within the C+LOO design. We first established three interpretable scenarios (Methods 4.3.1–4.3.3) and constructed corresponding simulators—Models Φ, Ψ, and Λ—for each (Figures 3a–3c). We confirmed that the simulators reproduce Taylor's power law (TPL), a ubiquitous property of scRNA-seq data [11–15], ensuring biological plausibility for the subsequent benchmarks (Methods 4.3.5 and SI D.3).

Using these simulators, we evaluated how the C+LOO design's multicollinearity affects the interpretation of experimental results through the following evaluation pipeline:

1. **Definition of ground truth main effects**: To quantify the accuracy of analytical results from C+LOO and PB designs, ground truth main effects were defined for each gene, serving as an anchor for evaluating performance (Methods 4.4.6).

2. **Simulation of experiments**: *In silico* experiments were conducted based on the C+LOO and comparative PB designs to generate scalar output values for each condition.

3. **Analysis of experimental results**: Given that conventional pairwise comparisons suffer from context-dependency due to reliance on specific baselines (detailed in SI E), we adopted standard DOE multivariate analyses—specifically multivariate linear regression (MLR) and analysis of variance (ANOVA)—to evaluate the directionality and statistical significance of the estimated effects.

4. **Benchmarking experimental design performance**: The concordance between the regulatory states inferred by the C+LOO and PB designs and the ground truth labels was quantified using weighted Cohen's κ coefficients (Methods 4.5.1), with uncertainty estimated via 95% bootstrap confidence intervals (CIs); comparative benchmarks against pairwise testing are detailed in SI G.4.

The relative performance of the designs varied across simulators (Figures 3d–3f): the PB design generally outperformed in Model Φ, whereas the C+LOO design performed better in Models Ψ and Λ. We also observed consistent trends across different Gaussian noise levels (SI G), with one exception in Model Λ: The C+LOO performed better under low noise conditions, whereas the PB design showed a clear advantage under high noise.

Importantly, multicollinearity plays a central role in determining design performance. The characteristic differences between the two designs—and their performance trends across simulators—persisted even after applying D-optimization, a common strategy for improving design matrices by reducing multicollinearity [16] (SI H):

- D-optimized C+LOO (DO-C+LOO) designs exhibited worse multicollinearity conditions than PB and D-optimized PB (DO-PB) designs

- D-optimization improved the performance of the C+LOO in scenarios favoring PB designs, while degrading it in cases where C+LOO outperformed

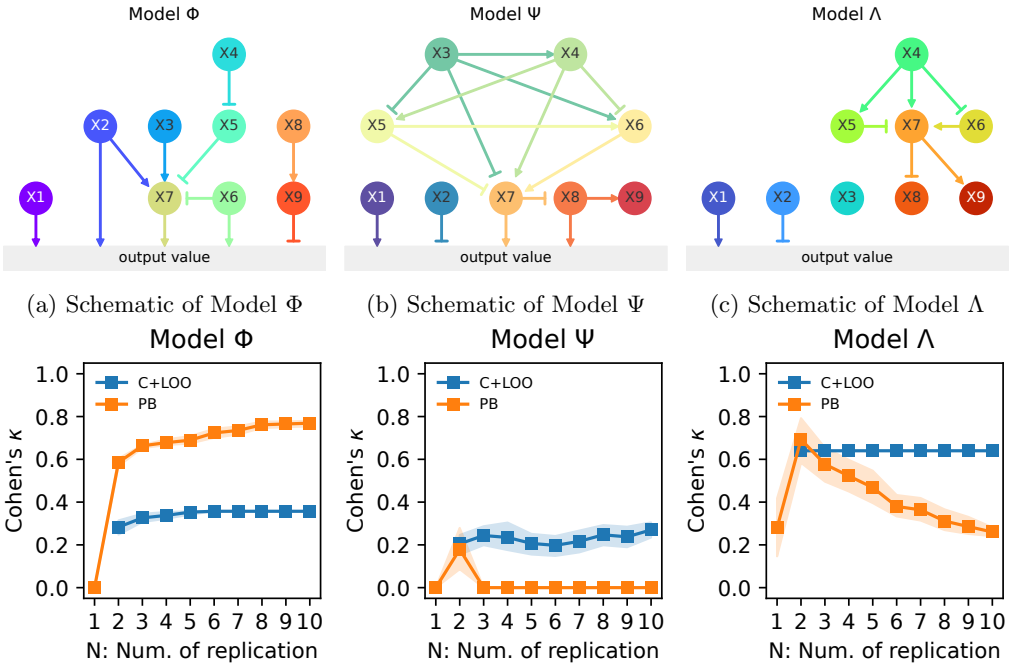
These simulated results collectively show that the performance of experimental designs is closely linked to the network structures of the simulated models. In systems regulated by parallel cascades—as modeled in Model Φ—the C+LOO design may underperform compared to the PB design, suffering from undesirable "contribution leakage" (shown in SI F) due to multicollinearity. However, that characteristic may prove beneficial in situations when the target phenomenon is predominantly governed by a cascade involving complex interactions—such as Model Ψ—since C+LOO designs can benefit from accounting for interaction terms via $\rho_{i\times j,i}$, rather than cutting off.

Results from Model Λ can also be explained with the network structure and factor correlations. Model Λ reflects a scenario with poor candidate gene selection where most factors do not influence the target outcome. Under such circumstances, model performance prediction can be difficult due to the numerical instability of the model (SI G.3).

Since both multivariate analyses and pairwise testing operate within the general linear model framework, they fundamentally rely on inverting the model matrix (Methods 4.4.3–4.4.4). Consequently, design-inherent multicollinearity is unavoidable in the C+LOO design, even when factors are independent. Depending on the network structure among factors, such multicollinearity can enhance, impair, or show no consistent effect on design performance.

2.3 Deriving a topological indicator for experimental design selection

Thus far, we have analyzed the C+LOO design from a DOE perspective, elucidating its inherent multicollinearity. Our simulations of three biologically plausible scenarios revealed that while this multicollinearity can lead to erroneous conclusions, it can paradoxically improve estimation accuracy in specific contexts—a phenomenon non-trivial to traditional DOE values. This underscores



(a) Schematic of Model Φ (b) Schematic of Model Ψ (c) Schematic of Model Λ

(d) Performance with Model Φ (e) Performance with Model Ψ (f) Performance with Model Λ

Figure 3: **Simulated experiments highlighted the pros and cons of the C+LOO design**

a–c: Schematics of the minimalist simulators. Constructed upon a GRN framework, these models feature nine interacting genes (X1–X9) regulating a non-negative scalar output via positive (pointed arrows) or negative (T-shaped arrows) edges, consistent with conventions in Boolean networks—a popular class of biological network modeling [17]. Model Φ (a) incorporates nine factors modulating the output via multiple direct and indirect pathways (Methods 4.3.1). Model Ψ (b) emulates a complex cascade featuring both synergistic and antagonistic interactions (Methods 4.3.2). Model Λ (c) represents a sparse factor scenario (poor candidate filtering) where the majority of genes exert no influence (Methods 4.3.3). All simulators share a unified architecture: (i) directed acyclic graph (DAG)-based GRN structures; (ii) parameters sampled from log-normal distributions; (iii) scalar outputs mimicking Perturb-seq readouts (e.g., single-gene markers or phenotypic scores); and (iv) rectified linear unit (ReLU) activation to introduce biological nonlinearity (Methods 4.3). Biological relevance was ensured by validating adherence to TPL—a ubiquitous power-law scaling between means and variances observed in scRNA-seq data [11–15] (SI D.3). **d–f**: Performance comparison of the C+LOO (blue) and PB (orange) designs based on the κ metric across replication levels $N = 1$–10. Markers represent the mean over 30 random seeds, with shaded areas indicating 95% bootstrap CIs. Data for the C+LOO design at $N = 1$ are omitted as ANOVA requires replication (SI E). For comparisons with Dunnett's test (a representative pairwise method showing lower noise robustness), refer to Figures S8–S10 and SI G.4.

a critical reality: there is no single "omnipotent" experimental design in biology; rather, the optimal strategy is strictly context-dependent.

How, then, should researchers select an appropriate design? Bridging this gap between theory and practice requires a framework that translates qualitative domain knowledge into quantitative decision-making. To achieve this, we adopted a data-driven approach: we first employed machine learning (ML) to identify key structural features governing design preference—C+LOO or PB—using exhaustive simulations, and subsequently distilled these findings into a simple, scalable indicator named the PBSI.

We began by defining a set of terms to quantitatively describe GRN structures, including *regulations*, *pathways*, *cascades*, *regulation weight*, *factor weight*, and *cascade length* (see Figure 4 and Methods 4.7.1 for definitions). Based on these concepts, we engineered 31 features designed to be scalable regardless of the number of factors (Methods 4.7.2).

To uncover the relationship between these features and experimental design performance, we developed Exhaustive Search Model of four factors (ESM4), a framework generating simulators with 3^{10} exhaustive structural configurations (Figure 5a). For each simulator, we evaluated the performance of C+LOO and PB designs using the *aptitude score*—an extension of Cohen's κ (Methods 4.5.2)—and binned them into "C+LOO-suited" or "PB-suited" classes (Figure 5b). Using the 31 network features, we trained a binary classification gradient boosting decision tree (GBDT) model to predict the "PB-suited" class (Methods 4.8.2–4.8.3). The receiver operating characteristic (ROC) curve shows that the model achieved high predictive performance with an area under the ROC (AUROC) of 0.892 (Figure 5c), and Shapley additive explanations (SHAP) analysis revealed the top 10 features contributing to the prediction (Figure 5d).

While the GBDT model was effective, training separate ML models for arbitrary numbers of factors is practically infeasible. We therefore sought a simple, model-free surrogate metric usable for any n. Although individual top features identified by SHAP showed only modest discrimination (AUROC $\approx$ 0.5–0.6; Figure 5e), combining them yielded a robust index. We defined PBSI as the mean of the three features: $P_{(+)}\%$, $1 - \mathrm{MaxRW}_{(+)}/R^*$, and $1 - \mathrm{RW}_{(+)}\%$ (SI I.1). On the ESM4 dataset, PBSI achieved an AUROC of 0.743, outperforming any single feature (Figure 5f).

To validate the scalability of PBSI beyond four-factor networks, we developed the Exhaustive Search Model of nine factors (ESM9) framework, a validation dataset for nine-factor network structures, finding that PBSI consistently outperformed all other features and composite indices in predicting design suitability (SI I.2).

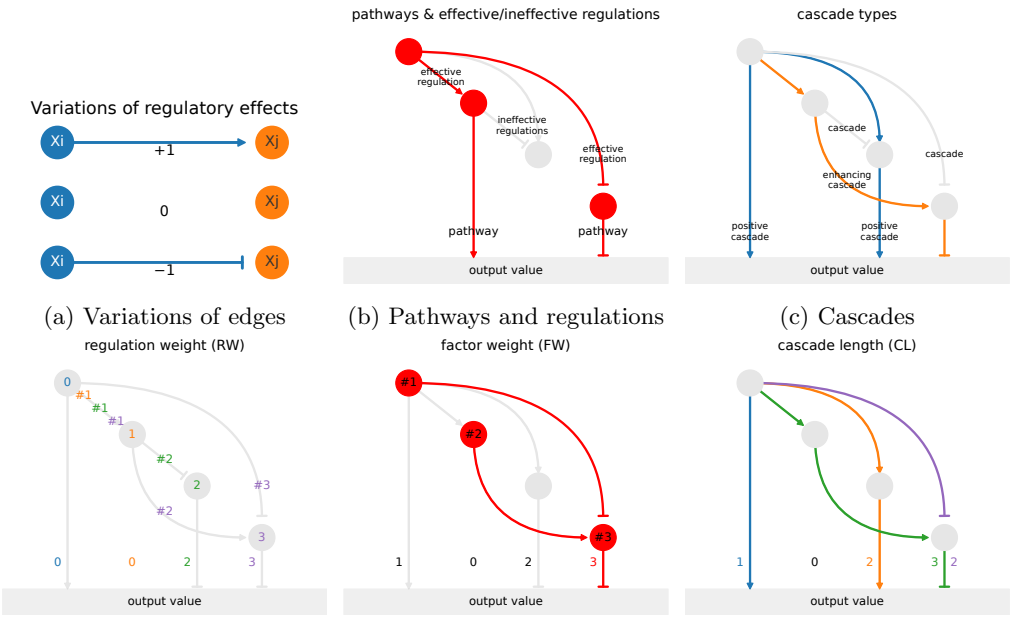
Conceptually, the three components of PBSI characterize a network topology that avoids the convergence of complex positive interactions—a structural feature that typically favors the C+LOO design due to nonlinearity. To validate this, we employed newly designed nine-factor simulators explicitly engineered to test these topological characteristics (Methods 4.8.8). These included Model Π (pure parallel topology yielding high PBSI), Model Δ (entangled topology driving signal amplification, yielding low PBSI), and crucially, Model Σ (entangled topology where amplification is dampened by a negative terminal edge, yielding high PBSI).

Our benchmarking confirmed that mere topological entanglement is insufficient to necessitate the C+LOO design. Instead, it is the signal amplification—triggered by unbroken chains of positive interactions—that causes the breakdown of orthogonal decomposability, thereby rendering the interaction-sensitive C+LOO design superior. PBSI successfully captures the presence or absence of this amplification structure, accurately predicting the distinct design preferences across these diverse topologies (SI I.3).

In situations where neither the *de facto* standard C+LOO nor the theoretically high-performing PB is universally superior, PBSI offers a breakthrough: it determines design suitability based solely on GRN topology. This requirement is minimalistic compared to the complexity of a full-scale Perturb-seq experiment.

To validate this applicability, we examined the consistency among biological context, inferred GRN topology, and PBSI scores using unperturbed plain scRNA-seq data. Focusing on TNF signaling (PB-candidate) and Myc response (C+LOO-candidate) as representative drivers of the S-phase (Figures 6a–6c), and filtering genes based on correlation profiles (Figures 6d–6e), we inferred GRNs positioning the S-phase score as the downstream output. Consistent with our simulations, the TNF network exhibited a fragmented parallel architecture with a negative terminal edge suppressing amplification, correctly yielding PBSI > 0.6 (Figures 6f–6g)—consistent with the threshold derived from our benchmarking with ESM9 and Models Φ and Ψ (SI I.3). Conversely, the Myc network displayed a convergence of positive interactions culminating in a positive terminal edge—a hallmark of signal amplification—correctly yielding PBSI < 0.6 (Figures 6h–6i). Further details and additional cases ensuring reproducibility are provided in SI I.4.

These empirical results confirm that the theoretical dichotomy observed in our simulations—the superiority of orthogonal decomposition versus its negation due to signal amplification-induced nonlinearity—is not a hypothetical artifact but a tangible reality in actual biological systems. This validation elevates PBSI from a theoretical construct to a powerful, parameter-free "nudge" for experimental design. Notably, PBSI is a concept born neither from biology alone nor from DOE alone, but from their organic integration—a framework we term the

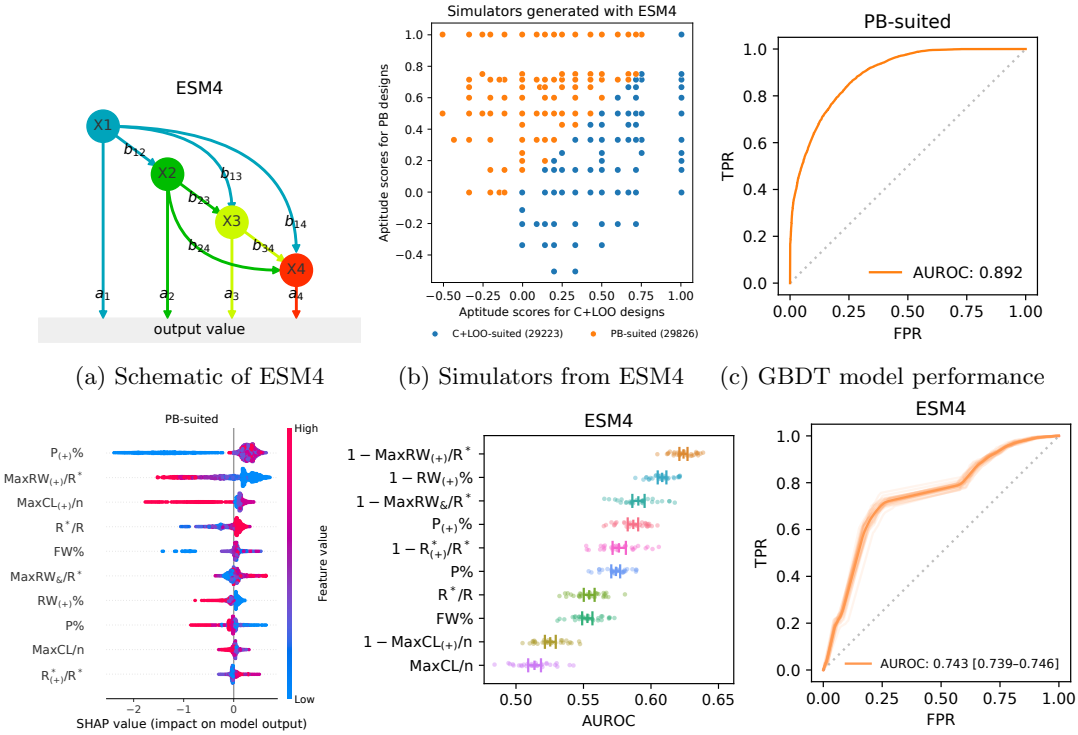


(a) Variations of edges (b) Pathways and regulations (c) Cascades

(d) Regulation weight (e) Factor weight (f) Cascade length

Figure 4: **Terminology for simulated model networks**

a: The three states of edges in the GRNs are considered—having positive (+1), null (0), or negative effects (−1). **b**: Regulations represent inter-factor edges, whereas pathways are edges between a factor and the output value. A regulation is defined as effective if it can ultimately influence output values via pathways, and ineffective otherwise. **c**: A cascade is defined as a linear series of regulations including the target pathway. An enhancing cascade (orange and blue) consists solely of positive regulations and becomes a positive cascade (blue) if the terminal pathway is also positive. Note that positive cascade is a subdivision of enhancing cascade. **d**: Regulation weight is defined as the number of regulations corresponding to respective pathways. **e**: Factor weight is defined as the number of factors corresponding to respective pathways. **f**: Cascade length is defined as the number of edges (i.e., regulations and pathways) in a cascade corresponding to respective pathways. If a pathway belongs to multiple cascades, its cascade length is determined by the longest cascade containing it (e.g., the rightmost pathway is part of the green cascade, resulting in a cascade length of 3).

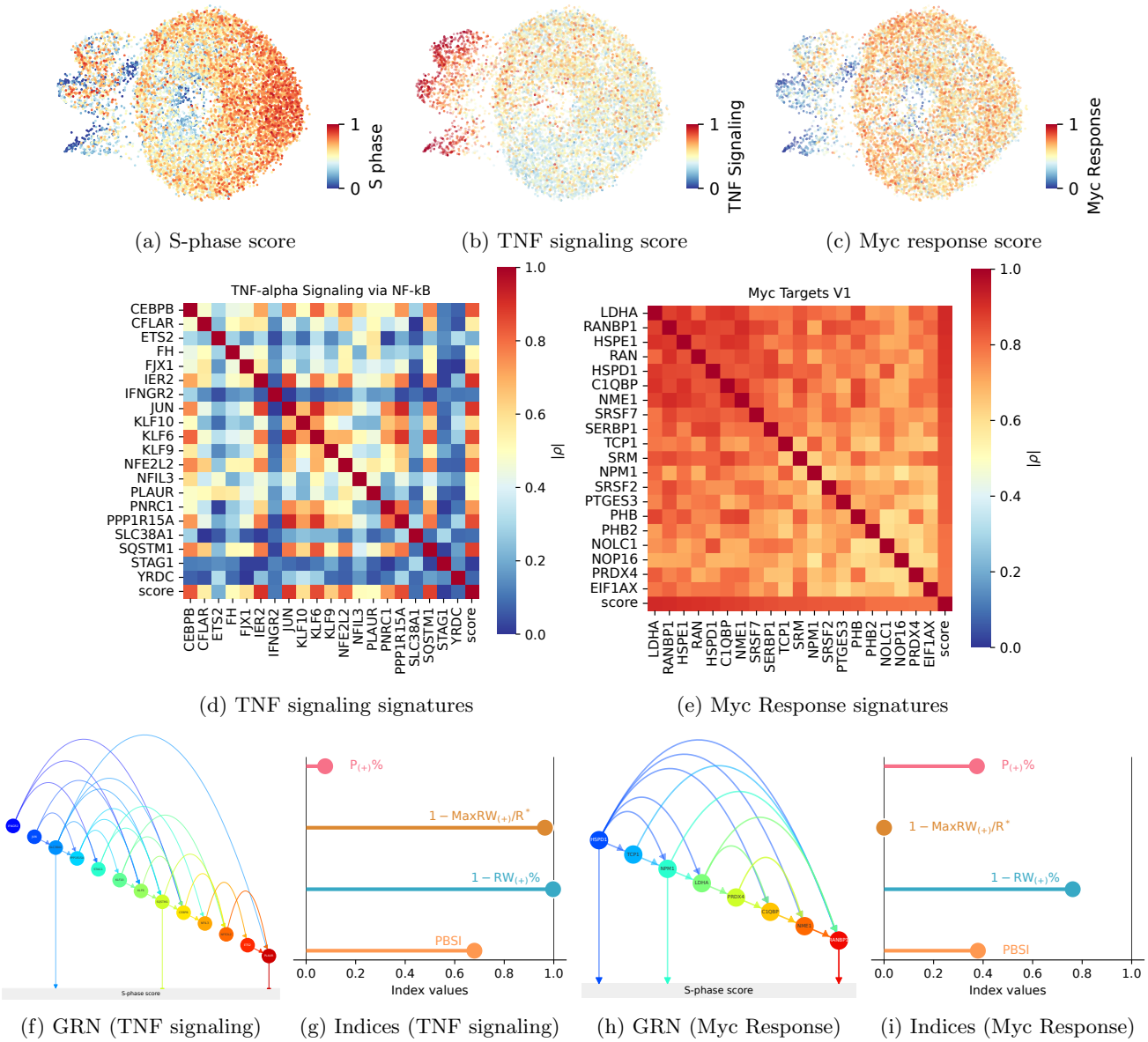


(a) Schematic of ESM4 (b) Simulators from ESM4 (c) GBDT model performance

(d) SHAP values (e) Performance of the top 10 features (f) Performance of PBSI

Figure 5: A data-driven approach identified network features associated with high performance of PB relative to C+LOO

a: Example network structure generated from ESM4. **b**: Classification of the 3^{10} ESM4 simulators for the ML task. Using aptitude scores for C+LOO (κ^*_{C}) and PB (κ^*_{P}), simulators were binned into "C+LOO-suited" (blue; $\kappa^*_{\mathrm{C}} \geq \kappa^*_{\mathrm{P}}$) and "PB-suited" (orange; $\kappa^*_{\mathrm{C}} < \kappa^*_{\mathrm{P}}$) classes. Since C+LOO is the *de facto* standard, the ML model specifically targets the "PB-suited" class to identify scenarios where PB outperforms C+LOO. Numbers in parentheses indicate the count of simulators in each category. **c**: ROC curve of the GBDT model predicting the "PB-suited" class. The horizontal axis represents the false positive rate, and the vertical axis represents the true positive rate. **d**: Beeswarm plot of the top 10 features with the highest mean absolute SHAP values for the "PB-suited" class. Markers are colored according to their corresponding feature values (pink, high; blue, low), and their positions on the horizontal axis indicate the SHAP values. **e**: AUROC scores of the top 10 features with the highest SHAP scores. The error bars indicate the mean and 95% bootstrap CIs. For features associated with the "C+LOO-suited" class, values were inverted so that all AUROC scores were ≥ 0.5. **f**: ROC curve of the PBSI showing its predictive performance (mean AUROC and the 95% bootstrap CI) for the "PB-suited" class.



(a) S-phase score (b) TNF signaling score (c) Myc response score

(d) TNF signaling signatures (e) Myc Response signatures

(f) GRN (TNF signaling) (g) Indices (TNF signaling) (h) GRN (Myc Response) (i) Indices (Myc Response)

Figure 6: **Identification of GRN topologies favoring the PB or C+LOO designs in real-world scRNA-seq data**

a–c: Uniform Manifold Approximation and Projection (UMAP) plots of K562 cells (non-targeting control condition from Replogle et al. [18]). Cells are colored by the S-phase score (a), TNF signaling score (b), and Myc response score (c). These scores summarize the expression levels of E2F targets, TNF-α signaling via NF-κB, and Myc targets V1 genes registered in the MSigDB Hallmark 2020 collection, respectively (see Methods 4.9.2). **d–e**: Heatmaps of the 20 selected genes and phenotypic scores. The absolute Pearson correlation coefficients are shown for the 20 low-correlated genes associated with TNF signaling (d) and the 20 high-correlated genes associated with Myc response (e). The "score" in each panel represents the phenotypic score for the respective biological context. See Methods 4.9.3 for detailed selection criteria. **f–g**: The inferred GRN representing the effect of TNF signaling on the S-phase score (f) and its four topology indices (g). The high PBSI score (> 0.6) indicates that the GRN, consisting of low-correlated TNF-α-related genes, exhibits fragmented cascades, as reflected by the high values of $1 - \mathrm{MaxRW}_{(+)}/\mathrm{R}^*$ and $1 - \mathrm{RW}_{(+)}\%$. Although the topology includes some entangled substructures, the presence of negative terminal edges effectively prevents signal amplification. **h–i**: The inferred GRN representing the effect of Myc response on the S-phase score (h) and its four topology indices (i). The low PBSI score (< 0.6) indicates that the GRN, consisting of high-correlated Myc-related genes, exhibits chains of positive gene–gene interactions, as reflected by the low values of $1 - \mathrm{MaxRW}_{(+)}/\mathrm{R}^*$ and $1 - \mathrm{RW}_{(+)}\%$. See Methods 4.9.4 for detailed GRN inference methods.

biology-oriented DOE.

3 Discussion

In Perturb-seq, C+LOO has been adopted almost by default across diverse biological systems. By introducing a DOE perspective, we formalized design performance as a quantity to be predicted and compared, in addition to showcasing the PB design as a potential alternative. Our analysis, spanning theoretical simulations and real-world validation, empirically demonstrated that the established dogma of DOE—the supremacy of orthogonality—does not always hold in nonlinear, overdispersed biological contexts. This discovery necessitates a paradigm shift towards a "biology-oriented DOE" framework.

Uncovering the universal principles of design suitability through exhaustive search represents a fundamental theoretical advancement in the interdisciplinary integration of biology and DOE. This comprehensive understanding, unattainable solely through empirical comparisons of specific biological contexts, establishes a robust theoretical foundation that transcends the limitations of individual case studies. Operationalizing these principles into the simple PBSI metric and confirming its applicability to real scRNA-seq data served as the critical validation step to prove that this theoretical abstraction is indeed anchored in biological reality.

However, true interdisciplinary integration is achieved not when a theory is proposed, but when it survives the complexity of real-world implementation. To ensure that our "biology-oriented DOE" avoids becoming a mere abstraction and effectively navigates the constraints of wet-lab reality, we distill our findings into the following actionable policies. Note that these guidelines serve as a high-level navigational compass; for comprehensive technical protocols, mathematical rationales, and epistemological discussions, we direct readers to the extensive documentation in SI J.

- **Candidate gene selection**: Ensure plausibility. For C+LOO, a large candidate list mitigates undesirable factor correlations; we specifically recommend a threshold of $n \geq 11$ factors (SI J.1).
- **Experimental design selection**: Prioritize C+LOO for systems with significant signal amplification. Otherwise, adopt PB as the robust default. Refer to PBSI for the optimal design. Quantitatively, we recommend adopting the PB design when PBSI > 0.6 (SI J.2).
- **Preliminary experiments**: Leverage existing resources—such as public databases and GRN inference tools—to estimate network topology before wet-lab experimentation (SI J.1–J.2). In the near future, large-scale Perturb-seq models will further aid in virtual sample size planning (SI J.4).
- **Strategies without preliminary data**: In the absence of data, hypothesize GRNs using prior domain knowledge (SI J.3).
- **Analytical framework**: Transition from pairwise testing to MLR/ANOVA to enhance statistical power without additional cost (SI J.4).
- **Result interpretation**: Exercise caution in generalization. Mechanistic insights inferred from experiments are not always guaranteed to hold in different experimental settings; interpretation must not be decoupled from experimental premises (SI J.5).
- **Combinatorial effect verification**: While *post hoc* pair addition to C+LOO is common, we caution against its theoretical complexity. Future work should rigorously compare this approach with established DOE methods for interaction analysis (SI J.6).

The application of DOE principles extends beyond individual experiments to the meta-scale of biological research. With the advent of foundation models, the field is increasingly attempting to adapt to the nonlinear and overdispersed complexity of biological systems through sheer data volume and model complexity [19]. While large-scale Perturb-seq models share this approach, they often face challenges in predicting combinatorial effects within extrapolation regimes [20]. Here, strategic experimental design can play a pivotal role by altering how the data space is populated, potentially leading to more efficient model construction than standard datasets. Analogous to physics-informed machine learning, which enhances generalization by constraining model complexity through physical laws [21], large-scale biological models could benefit from structural constraints derived from experimental designs and the essential logic of phenotype generation. In this regard, the simulator architecture proposed herein—representing the minimal constitution of biological plausibility—could serve as a foundational testbed for developing such sophisticated, structure-aware models.

In addition to offering a principled basis for design selection, this study holds significant epistemological importance by reframing Perturb-seq as a discrete causal model connecting perturbations to phenotypes, given the inherent "blank time" between intervention and observation. This perspective necessitates a distinction between biological phenomena that can be adequately handled by discrete causal models and those that cannot. In this context, we propose a shift in perspective from strictly "scientific understanding" to "engineering control." That is, by advancing Perturb-seq as a discrete causal model, we can adopt an engineering approach: optimizing pre-defined phenotypic metrics to control the system, even if the internal regulatory mechanisms remain partially black-boxed.

While biology has traditionally prioritized mechanistic elucidation, the rise of black-box optimization [22] suggests that complete interpretability is not always a necessary condition for controlling complex systems, provided that the system's behavior is stable and empirically reproducible. Importantly, this engineering stance does not deviate drastically from the traditions of biomedical science; randomized controlled trials, for instance, establish causality without necessarily waiting for a complete mechanistic explanation. Thus, an engineering approach that temporarily bypasses mechanism is compatible with biomedical pragmatism. Fortunately, this does not imply a loss of scientific insight; scRNA-seq data measured during Perturb-seq still allows for the *post-hoc* inference of mechanistic insights, preserving the ability to bridge control and understanding.

Ultimately, perhaps our most significant contribution lies in providing an early case study suggesting that the functional integration of biology and mathematics—a realm where simplicity is often revered as an aesthetic—requires a fundamental epistemological re-framing. We hope that this study serves as a cornerstone for this interdisciplinary integration, accelerating the evolution of multivariate biology in the era of data science.

4 Methods

4.1 Design matrices and their statistics

4.1.1 An overview of the C+LOO design

The C+LOO design consists of (i) a control trial in which all factors included (i.e., all genes are intact); and (ii) additional trials in which the factors are subtracted one at a time (i.e., single genes are knocked down). This design has traditionally been applied to biological experiments [2]—such as the discovery of the four pluripotency-inducing factors [23].

From a DOE perspective, the C+LOO design can be conceptualized as a variant of the One-Factor-At-a-Time (OFAT) family [3]. However, the status of OFAT is nuanced; while traditionally criticized for its statistical inefficiency and inability to detect interactions, it retains prevalence in engineering practice due to its intuitive simplicity [3].

Unlike generalized OFAT strategies—which often involve multi-level factors without a fixed reference [3]—the C+LOO design represents a specialized framework. It is distinctly characterized by its reliance on a fixed baseline (the "all-intact" control) and a subtractive approach to verify the causal necessity of each factor relative to that baseline.

When we denote the C+LOO design matrix for n factors ($\forall n \in \mathbb{N}$) as $C(n)$ symbolizing the addition and subtraction of factors with ± 1, $C(5)$ with five factors for another example is given as follows:

$$C(5) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 & 1 \\ 1 & 1 & -1 & 1 & 1 \\ 1 & 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 \end{bmatrix} \tag{1}$$

$$= \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_5 \end{bmatrix} = \begin{bmatrix} \text{trial}_1 \\ \text{trial}_2 \\ \vdots \\ \text{trial}_6 \end{bmatrix}. \tag{2}$$

Note that the column vectors (i.e., $\mathbf{x}_i$ where $\forall i \in \mathbb{N}$ s.t. $1 \leq i \leq n$) indicate the assignments of experimental conditions for the corresponding factors, and the row vectors (i.e., trial_j where $\forall j \in \mathbb{N}$ s.t. $1 \leq j \leq n+1$) indicate the overall experimental conditions per run. In general, $C(n)$ can be represented as follows:

$$C(n) = \begin{bmatrix} \tilde{c}_{1,1} & \tilde{c}_{2,1} & \cdots & \tilde{c}_{n,1} \\ \tilde{c}_{1,2} & \tilde{c}_{2,2} & \cdots & \tilde{c}_{n,2} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{c}_{1,n+1} & \tilde{c}_{2,n+1} & \cdots & \tilde{c}_{n,n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_n \end{bmatrix}, \tag{3}$$

where $\tilde{c}_{i,j} \in \{-1, 1\}$ represents experimental conditions (1, addition of the factor; -1, subtraction of the factor). Note that $\tilde{c}_{i,j} := -1$ if and only if $j = i + 1$; otherwise, $\tilde{c}_{i,j} := 1$.

4.1.2 An overview of the FF design

The experimental designs with complete control experiments—running all possible conditions across 2^n trials for n factors—are called the FF design. In theory, the FF design offers the highest resolution among experimental designs. The FF design is known to show no multicollinearity since any effects from the presence or absence of other factors are canceled out by calculating the mean values. Further information is provided in SI C.1.

4.1.3 An overview of the PB design

The PB design is a popular class of screening design introduced in 1946 [4, 24]. Based on a Hadamard matrix, the PB design assumes a first-degree polynomial model, where the results are explained by a linear combination of the products of factors and their coefficients [25].

The PB design is unique in having only $4 \times k$ ($k \in \mathbb{N}$) trials, making it suitable for up to $(4 \times k) - 1$ factors [24]. Since PB designs involve a comparable number of trials to C+LOO designs for a given n, we adopted the PB design as the comparison counterparts to the C+LOO design. Regarding the C+LOO design, the total number

of trials $n_{\max}$ for given numbers of replication $\forall N \in \mathbb{N}$ and factors $\forall n \in \mathbb{N}$ is given as follows:

$$n_{\max} = N \cdot (n+1). \tag{4}$$

On the other hand, the $n_{\max}$ that the PB design requires for given $\forall N \in \mathbb{N}$ and $\forall n \in \underline{99}$ is given as follows:

$$n_{\max} = \begin{cases} 96N & (88 \leq n \leq 91) \\ 4N \cdot (\lfloor \frac{n}{4} \rfloor + 1) & (\text{otherwise}) \end{cases}, \tag{5}$$

where $\lfloor\rfloor$ is the floor function. It is noteworthy that the PB design is defined for $4 \times k$ ($k \in \mathbb{N}$) trials up to 100 except for 92 [26].

Further information is provided in SI C.2.

4.1.4 Design matrices, main effects, and interaction terms

We denote a general experimental design $D(n, m)$ with n factors and m trials as follows:

$$\begin{aligned} D(n,m) &= \begin{bmatrix} \tilde{c}_{1,1} & \tilde{c}_{2,1} & \cdots & \tilde{c}_{n,1} \\ \tilde{c}_{1,2} & \tilde{c}_{2,2} & \cdots & \tilde{c}_{n,2} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{c}_{1,m} & \tilde{c}_{2,m} & \cdots & \tilde{c}_{n,m} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_n \end{bmatrix}, \end{aligned} \tag{6}$$

where $\tilde{c}_{i,j} \in \{-1, 1\}$ ($1 \leq i \leq n, 1 \leq j \leq m$) assigns specific conditions. Following the notation of the C+LOO design introduced in Equation (1), we denote the i-th column vector corresponding to the i-th factor of the design matrix as $\mathbf{x}_i$.

The main effects are defined as the independent contributions of individual factors [27]. Although they can be defined for specific levels of given factors (e.g., the main effect $M^i_{l_i}$ of the i-th factor at the l_i-th level), the two-level n-factor model simply deals the main effect M^i under existence of the i-th factor by introducing an assumption $\sum_{l_i=1}^{2} M^i_{l_i} := 0$ (thus, $M^i_1 = -M^i_2$).

Similarly, the second-order interactions are defined at respective levels of the given two factors (e.g., the second-order interaction $I^{i_1 \times i_2}_{l_{i_1}, l_{i_2}}$ of the i_1-th and i_2-th factors at the l_{i_1}-th and l_{i_2}-th levels). Since $\sum_{l_{i_q}}^{L_{i_q}} I^{i_1 \times i_2}_{l_{i_1}, l_{i_2}} := 0$ holds for all $q \in \{1, 2\}$, the following equation holds in the two-level n-factor models:

$$I^{i_1 \times i_2}_{1,1} = -I^{i_1 \times i_2}_{1,2} = I^{i_1 \times i_2}_{2,2} = -I^{i_1 \times i_2}_{2,1}. \tag{7}$$

Hence, we exclusively consider the interaction $I^{i_1 \times i_2}$ under co-existence or joint absence of the i_1-th and i_2-th factors.

Since the FF design covers complete experimental conditions, the definitions of the main effect M^i of the i-th factor and the second-order interaction term $I^{i_1 \times i_2}$ between the i_1-th and i_2-th factors can be derived using the column vectors $\mathbf{x}_1$–$\mathbf{x}_n$ of two-level n-factor FF designs ($1 \leq i, i_1, i_2 \leq n$) with N replication as follows:

$$M^i := \frac{\mathbf{x}_i \mathbf{y}}{2^n N} \tag{8}$$

$$I^{i_1 \times i_2} := \frac{(\mathbf{x}_{i_1} \odot \mathbf{x}_{i_2})\mathbf{y}}{2^n N} = \frac{\mathbf{x}_{i_1 \times i_2}\mathbf{y}}{2^n N}, \tag{9}$$

where $\mathbf{y}$ is the observation vector, $\odot$ refers the element-wise (i.e., Hadamard) product of vectors, and $\mathbf{x}_{i_1 \times i_2} := \mathbf{x}_{i_1} \odot \mathbf{x}_{i_2}$ is the column vector for the second-order interaction between the i_1-th and i_2-th factors—indicating the existence and absence of the interaction. Although M^i and $I^{i_1 \times i_2}$ are frequently scaled with $2^{n-1}N$ or $2^{n-2}N$ in DOE-contexts, here we align the definition to MLR coefficients for consistency.

Notably, the Pearson correlations among main effects and interaction terms can be calculated with the column vectors—instead of explicitly calculating M^i or $I^{i_1 \times i_2}$ from a concrete $\mathbf{y}$. In the field of DOE, this technique is extended to other experimental designs to visualize the characteristics of them without performing actual experiments. Adopting the same idea, we identify the column vectors $\mathbf{x}_i$ or $\mathbf{x}_{i \times j}$ as a main effect or an interaction in this study for simplicity.

4.1.5 Factor correlations in the C+LOO design

To understand the statistical properties of the C+LOO design, we derived the formula on the correlation coefficients among main effects and interaction terms at various conditions.

As we mentioned in Methods 4.1.4, $\mathbf{x}_i$ denotes the main effect of the i-th factor, and $\mathbf{x}_{i \times j}$ refers to the interaction term between the i-th and the j-th factors ($\forall n \in \mathbb{N}$, $\forall i, j \in \mathbb{N}$ such that $1 \leq i, j \leq n$, and $i \neq j$). We also introduce correlation coefficients $\rho_{i,j} := \mathrm{Corr}(\mathbf{x}_i, \mathbf{x}_j)$, $\rho_{i \times j,k} := \mathrm{Corr}(\mathbf{x}_{i \times j}, \mathbf{x}_k)$, and $\rho_{i \times j, k \times l} := \mathrm{Corr}(\mathbf{x}_{i \times j}, \mathbf{x}_{k \times l})$ while $\mathrm{Corr}(,)$ is the formula for the Pearson correlation coefficient. It is also noteworthy that $\mathrm{Corr}(,)$ is symmetrical, therefore, $\rho_{k, i \times j} := \mathrm{Corr}(\mathbf{x}_k, \mathbf{x}_{i \times j}) = \rho_{i \times j, k}$. Furthermore, $\rho_{i \times j, k} = \rho_{j \times i, k} = \rho_{k, j \times i} = \rho_{k, i \times j}$ because of the symmetry of the Hadamard multiplication in interaction terms.

To evaluate how much of correlations caused in $C(n)$ at different levels of n, we calculated the correlation coefficients $\rho_{i,j}$, $\rho_{i \times j, i}$, $\rho_{i \times j, k}$, $\rho_{i \times j, i \times l}$, and $\rho_{i \times j, k \times l}$ ($\forall n \in \mathbb{N}$, $\forall i, j, k, l \in \mathbb{N}$ such that $1 \leq i, j, k, l \leq n$ and $(i-j) \cdot (i-k) \cdot (i-l) \cdot (j-k) \cdot (j-l) \cdot (k-l) \neq 0$) at specific conditions as follows:

$$\rho_{i,j} = -\frac{1}{n} \quad (n \geq 2) \tag{10}$$

$$\rho_{i \times j, i} = \sqrt{\frac{n-1}{2n}} \quad (n \geq 2) \tag{11}$$

$$\rho_{i \times j, k} = -\sqrt{\frac{2}{n(n-1)}} \quad (n \geq 3) \tag{12}$$

$$\rho_{i\times j, i\times l} = \frac{n-3}{2(n-1)} \quad (n \geq 3) \tag{13}$$

$$\rho_{i\times j, k\times l} = -\frac{2}{n-1} \quad (n \geq 4). \tag{14}$$

Notably, here we omitted $\rho_{i,i}$ and $\rho_{i\times j, i\times j}$ since they are constant. More rigorous descriptions are provided in SI A–B.

We also calculated the limit values of them to have the following equations:

$$\lim_{n\to\infty} \rho_{i,j} = \lim_{n\to\infty} -\frac{1}{n} = 0 \tag{15}$$

$$\lim_{n\to\infty} \rho_{i\times j, i} = \lim_{n\to\infty} \sqrt{\frac{n-1}{2n}} = \sqrt{\frac{1}{2}} \tag{16}$$

$$\lim_{n\to\infty} \rho_{i\times j, k} = \lim_{n\to\infty} -\sqrt{\frac{2}{n(n-1)}} = 0 \tag{17}$$

$$\lim_{n\to\infty} \rho_{i\times j, i\times l} = \lim_{n\to\infty} \frac{n-3}{2(n-1)} = \frac{1}{2} \tag{18}$$

$$\lim_{n\to\infty} \rho_{i\times j, k\times l} = \lim_{n\to\infty} -\frac{2}{n-1} = 0. \tag{19}$$

Note that the limit operations are all considered as discrete since the correlation functions are defined for $n \in \mathbb{N}$.

4.1.6 Visualization of design matrices, correlation heatmaps, and the correlation curves

We created the C+LOO and FF design matrices using `NumPy` [28] and `pandas` [29]. We generated the PB design matrix using `pyDOE` [30] and other Python packages. The column vectors for interaction terms were calculated with `pandas` and other Python packages.

We created the design matrices and correlation heatmaps with `NumPy`, `pandas` and other Python packages. We also visualized the five correlation functions by evaluating Equations (10)–(14) for values ranging from 2 to 12 using `NumPy`.

4.1.7 Model matrices

The model matrix is a matrix representing independent variables (including the intercept) of the model. As a design matrix represents the states of the manipulated variables, it can serve as a part of the model matrix for MLR models.

For example, the model matrix X for the "additive model"—an MLR model comprising main effects and a constant term (Methods 4.4.2)—based on an experimental design $D(n, m)$ can be given as follows:

$$\begin{aligned} X &= \left[\begin{array}{c|c} 1 & \\ \vdots & D(n,m) \\ 1 & \end{array}\right] \\ &= \begin{bmatrix} \mathbf{1} & \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_n \end{bmatrix}, \end{aligned} \tag{20}$$

where $\mathbf{1}$ is the m-dimensional vector which elements are all 1.

4.1.8 Condition numbers and VIF

We calculated condition numbers and VIF values of a factor based on the model matrices of additive models (Methods 4.1.7 and 4.4.2). Then, we also plotted them using `NumPy`, `statsmodels` [31], and other Python packages. Notably, all factors in an additive model show the same VIF values when the model matrix is based on the C+LOO or PB designs. Thus, we reported VIF values of one factor per n.

4.2 Formalization of Perturb-seq

To rigorously evaluate the impact of experimental design, we first formalized the theoretical framework of Perturb-seq and delineated the scope of biological systems where DOE principles are applicable.

First, we define Perturb-seq as a framework that establishes discrete causal models mapping perturbation inputs (e.g., gene KD) to phenotypic outputs (scalar metrics derived from high-dimensional scRNA-seq data) (Figure S3). This approach inherently treats the intermediate molecular mechanisms between perturbation and observation as a black box, inferring causal relationships by contrasting perturbed states against an "all-gene intact" baseline.

Second, we distinguish between two classes of biological systems based on the temporal dynamics of their underlying GRNs during the interval between perturbation and observation:

- **Systems with time-varying GRN topology:** Systems where the GRN structure itself evolves in a condition-dependent manner (e.g., sign reversals, rewiring, or module coupling/decoupling as shown in Figure S4a). In these systems, the static snapshot nature of scRNA-seq leads to a loss of traceability, making mechanistic inference theoretically intractable without time-series data.

- **Systems with static GRN topology:** Systems where the GRN structure remains static (time-invariant), and perturbations manifest solely as quantitative changes in gene expression levels within a fixed topology (Figure S4b).

We posit that the discrete causal modeling used in Perturb-seq is primarily robust in systems with static GRN topology, where the structural invariance ensures that the input–response relationship can be reliably approximated. Consequently, this study restricts its scope to such static systems to isolate the effects of experimental design from the confounding factors of network evolution.

A detailed derivation of this formalization and illustrative examples of GRN dynamics are provided in SI D.1.

4.3 Minimalist simulators

Perturb-seq is primarily utilized to directly link perturbed genes with phenotypes, often bypassing the intricacies of underlying regulatory networks to interpret transcriptomic changes (SI D.1). However, to gain deeper insights into how experimental design shapes observable outcomes, it is crucial to dissect the impact of perturbations on signaling cascades in detail.

Given the difficulty of isolating the true effects of experimental design choices in real-world data (where the governing equations remain unknown), we constructed minimalist simulators of signaling cascades. These simulators provide tangible insights into how differences in experimental design alter the biological interpretation of Perturb-seq (see SI D.2 for a detailed rationale).

All minimalist simulators developed in this study share a unified architecture: gene–gene regulatory relationships are represented as a directed acyclic graph (DAG) mimicking transcriptional control, which ultimately regulates a scalar output value representing the phenotype (corresponding to single-gene markers or phenotypic scores). To balance computational efficiency with biological plausibility, we employed the rectified linear unit (ReLU) activation function. This choice introduces nonlinearity—preventing the output from reducing to a simple linear sum of gene expression levels—and effectively emulates the threshold-like behavior observed in biological systems while keeping computational costs low [32–35].

Consequently, our simulator exhibits a structure analogous to a real-valued extension of Boolean networks [36], similar to those employed in mathematical models of cell-signaling pathways [17]. Furthermore, model parameters were initialized via random sampling from log-normal distributions to recapitulate the long-tail distributions frequently observed in real-world biology.

4.3.1 Model Φ: a basic model simulating biological mechanisms

Model Φ is a toy simulation of Perturb-seq analysis applied to complex gene regulation, involving nine factors (X1–X9) and multiple regulatory routes that influence the output value either directly or indirectly. Among the models presented in this study, it serves as the most neutral baseline compared to Models Ψ and Λ, given its balanced configuration of pathways.

Let $n_{\max}$ denote the total number of trials—e.g., $n_{\max} = 10N$ for N replicates of $C(9)$. Define the following for all $i, j \in \mathbb{N}$ such that $1 \le i \le 9$ and $1 \le j \le n_{\max}$:

- $x_{i,j}$: observed value of factor i in trial j
- y_j: output value of trial j
- $v_i \sim \text{Lognormal}(1, (0.8)^2)$: latent constant of factor i
- $c_{i,j} \in \{0, 1\}$: perturbation indicator (0, factor included; 1, factor removed)
- $K := \{1, 2, 6, 7, 9\}$
- $L := \{27, 37, 45, 57, 67, 89\}$
- $a_k \sim \text{Lognormal}(2, (0.3)^2)$ for $k \in K$: direct coefficients on y_j
- $b_\ell \sim \text{Lognormal}(1, (0.5)^2)$ for $\ell \in L$: regulatory coefficients for inter-factor influence
- $\varepsilon_{i,j}, \varepsilon_{y,j} \sim \mathcal{N}(0, \sigma^2)$: Gaussian noise for $x_{i,j}$ and output y_j
- ReLU: ReLU function, defined as $f(x) = \max(0, x)$

Here, $\mathcal{N}(\mu, \sigma^2)$ denotes the normal distribution, and $\text{Lognormal}(\mu, \sigma^2)$ denotes a log-normal distribution such that $\ln X \sim \mathcal{N}(\mu, \sigma^2)$. The variables v_i, a_k, and b_ℓ are shared across trials and were generated using `NumPy` with a fixed random seed to ensure reproducibility and positivity. Note that the index ℓ in b_ℓ refers to a specific regulatory connection between factors (e.g., b_{27} affects the influence from X2 to X7), rather than to a particular factor itself.

The model equations are defined as:

$$x_{1,j} := \text{ReLU}(c_{1,j}(v_1 + \varepsilon_{1,j})) \tag{21}$$

$$x_{2,j} := \text{ReLU}(c_{2,j}(v_2 + \varepsilon_{2,j})) \tag{22}$$

$$x_{3,j} := \text{ReLU}(c_{3,j}(v_3 + \varepsilon_{3,j})) \tag{23}$$

$$x_{4,j} := \text{ReLU}(c_{4,j}(v_4 + \varepsilon_{4,j})) \tag{24}$$

$$x_{5,j} := \text{ReLU}(c_{5,j}(v_5 - b_{45}x_{4,j} + \varepsilon_{5,j})) \tag{25}$$

$$x_{6,j} := \text{ReLU}(c_{6,j}(v_6 + \varepsilon_{6,j})) \tag{26}$$

$$\begin{aligned} x_{7,j} := \text{ReLU}(c_{7,j}(v_7 + b_{27}x_{2,j} + b_{37}x_{3,j} \\ - b_{57}x_{5,j} - b_{67}x_{6,j} + \varepsilon_{7,j})) \end{aligned} \tag{27}$$

$$x_{8,j} := \text{ReLU}(c_{8,j}(v_8 + \varepsilon_{8,j})) \tag{28}$$

$$x_{9,j} := \text{ReLU}(c_{9,j}(v_9 + b_{89}x_{8,j} + \varepsilon_{9,j})) \tag{29}$$

$$\begin{aligned} y_j := \text{ReLU}(a_1x_{1,j} + a_2x_{2,j} + a_6x_{6,j} \\ + a_7x_{7,j} - a_9x_{9,j} + \varepsilon_{y,j}). \end{aligned} \tag{30}$$

4.3.2 Model Ψ: a highly complex regulatory circuit of interactions

Model Ψ represents a more intricate network of interactions than Model Φ. Using the similar notations introduced in Model Φ, we define the following for all $i, j \in \mathbb{N}$ such that $1 \le i \le 9$ and $1 \le j \le n_{\max}$:

- $x_{i,j}$, y_j, $c_{i,j}$, $\varepsilon_{i,j}$, $\varepsilon_{y,j}$, ReLU: identical to those in Model Φ
- $K := \{1, 2, 7, 8\}$

- $L := \{34, 35, 36, 37, 45, 46, 47, 56, 57, 67, 78, 89\}$
- $a_k \sim \text{Lognormal}(2, (0.3)^2)$ for $k \in K$
- $b_\ell \sim \text{Lognormal}(1, (0.5)^2)$ for $\ell \in L$

As in Model Φ, the parameters v_i, a_k, and b_ℓ are shared across trials and were sampled using `NumPy` with a fixed random seed.

The model equations are given by:

$$x_{1,j} := \text{ReLU}(c_{1,j}(v_1 + \varepsilon_{1,j})) \tag{31}$$

$$x_{2,j} := \text{ReLU}(c_{2,j}(v_2 + \varepsilon_{2,j})) \tag{32}$$

$$x_{3,j} := \text{ReLU}(c_{3,j}(v_3 + \varepsilon_{3,j})) \tag{33}$$

$$x_{4,j} := \text{ReLU}(c_{4,j}(v_4 + b_{34}x_{3,j} + \varepsilon_{4,j})) \tag{34}$$

$$x_{5,j} := \text{ReLU}(c_{5,j}(v_5 - b_{35}x_{3,j} + b_{45}x_{4,j} + \varepsilon_{5,j})) \tag{35}$$

$$\begin{aligned} x_{6,j} := \text{ReLU}(c_{6,j}(v_6 + b_{36}x_{3,j} \\ - b_{46}x_{4,j} + b_{56}x_{5,j} + \varepsilon_{6,j})) \end{aligned} \tag{36}$$

$$\begin{aligned} x_{7,j} := \text{ReLU}(c_{7,j}(v_7 - b_{37}x_{3,j} + b_{47}x_{4,j} \\ - b_{57}x_{5,j} + b_{67}x_{6,j} + \varepsilon_{7,j})) \end{aligned} \tag{37}$$

$$x_{8,j} := \text{ReLU}(c_{8,j}(v_8 - b_{78}x_{7,j} + \varepsilon_{8,j})) \tag{38}$$

$$x_{9,j} := \text{ReLU}(c_{9,j}(v_9 + b_{89}x_{8,j} + \varepsilon_{9,j})) \tag{39}$$

$$y_j := \text{ReLU}(a_1x_{1,j} - a_2x_{2,j} + a_7x_{7,j} + a_8x_{8,j} + \varepsilon_{y,j}). \tag{40}$$

4.3.3 Model Λ: a model with poorly curated candidates

Model Λ mimics a situation where the candidate genes are poorly filtered so that the majority of them shows no influence on the target phenomenon. With the similar notations introduced in Model Ψ, we define the following for all $i, j \in \mathbb{N}$ such that $1 \leq i \leq 9$ and $1 \leq j \leq n_{\max}$:

- $x_{i,j}$, y_j, $c_{i,j}$, $\varepsilon_{i,j}$, $\varepsilon_{y,j}$, ReLU: identical to those in Models Φ and Ψ
- $K := \{1, 2\}$
- $L := \{45, 46, 47, 57, 67, 78, 79\}$
- $a_k \sim \text{Lognormal}(2, (0.3)^2)$ for $k \in K$
- $b_\ell \sim \text{Lognormal}(1, (0.5)^2)$ for $\ell \in L$

As in Models Φ and Ψ, the parameters v_i, a_k, and b_ℓ are shared across trials and were sampled using `NumPy` with a fixed random seed.

The model equations are given by:

$$x_{1,j} := \text{ReLU}(c_{1,j}(v_1 + \varepsilon_{1,j})) \tag{41}$$

$$x_{2,j} := \text{ReLU}(c_{2,j}(v_2 + \varepsilon_{2,j})) \tag{42}$$

$$x_{3,j} := \text{ReLU}(c_{3,j}(v_3 + \varepsilon_{3,j})) \tag{43}$$

$$x_{4,j} := \text{ReLU}(c_{4,j}(v_4 + \varepsilon_{4,j})) \tag{44}$$

$$x_{5,j} := \text{ReLU}(c_{5,j}(v_5 + b_{45}x_{4,j} + \varepsilon_{5,j})) \tag{45}$$

$$x_{6,j} := \text{ReLU}(c_{6,j}(v_6 - b_{46}x_{4,j} + \varepsilon_{6,j})) \tag{46}$$

$$\begin{aligned} x_{7,j} := \text{ReLU}(c_{7,j}(v_7 + b_{47}x_{4,j} - b_{57}x_{5,j} \\ + b_{67}x_{6,j} + \varepsilon_{7,j})) \end{aligned} \tag{47}$$

$$x_{8,j} := \text{ReLU}(c_{8,j}(v_8 - b_{78}x_{7,j} + \varepsilon_{8,j})) \tag{48}$$

$$x_{9,j} := \text{ReLU}(c_{9,j}(v_9 + b_{79}x_{7,j} + \varepsilon_{9,j})) \tag{49}$$

$$y_j := \text{ReLU}(a_1x_{1,j} - a_2x_{2,j} + \varepsilon_{y,j}). \tag{50}$$

4.3.4 Implementation

We generated random variables using `NumPy`, setting different seeds for pseudorandom numbers in different trials. By default, Gaussian noise followed $\mathcal{N}(0, 1)$. To investigate the impact of noise intensity, we also evaluated $\sigma \in \{0.5, 2, 4\}$.

Regarding replication, we simply repeated the experiments sequentially without shuffling. In real-world experiments, it is essential to randomize the execution order—specifically, the row order of the design matrix—to minimize the impact of covariates associated with the trial sequence [37] (e.g., improvements in technical proficiency or workflow efficiency). However, for simplicity, we assumed no such covariates were associated with the order of experiments and treated the output values as independent and identically distributed variables.

4.3.5 Validation of the biological relevance of the simulators

Although Models Φ, Ψ, and Λ were constructed without any specific network motifs, we examined whether these simple networks could reproduce the nonlinear and overdispersed characteristics of real-world biological systems. Specifically, we tested whether the output values of these simulators followed the TPL, a log-log relationship between means and variances observed across a variety of biological systems, including scRNA-seq data [11–15].

As detailed in SI D.3, we performed FF-based simulations for each model ($N = 10$), varying the Gaussian noise levels ($\sigma \in \{0.5, 1, 2, 4\}$) to generate output values under a wide range of experimental conditions. From the generated data, the means and variances of output values were calculated within each experimental condition defined by the FF design. We then fitted linear regression models to the log-transformed means and variances to obtain the coefficient of determination (R^2) and the scaling exponent (β), which represent the degree of conformity to the TPL.

The implementation was carried out in Python using packages including `scikit-learn` [38].

4.4 Statistical analyses

4.4.1 Statistical testing for pairwise comparisons

In SI E and G, we performed Dunnett's test [9] using the Python package, `SciPy` [39], with the control condition defined as the trial including all factors. Statistical significance was set at $\alpha = 0.05$; p-values were reported as $*$ for $p < 0.05$, $**$ for $p < 0.01$, $***$ for $p < 0.001$, and N.S. for not significant. Plots were generated with Python packages, and we displayed error bars representing the mean$\pm$ SD.

We also conducted per-comparison power analysis for k treatment groups. The required quantities—the residual degrees of freedom ϕ, noncentrality parameters λ_i, correlation matrix R, test statistics T_i, and power $1-\beta_i$—were defined as follows:

$$\phi = N_0 + \sum_i N_i - (k+1) \tag{51}$$

$$\lambda_i = \frac{\mu_i - \mu_0}{S\sqrt{\frac{1}{N_0} + \frac{1}{N_i}}} \tag{52}$$

$$R_{i,j} := \begin{cases} 1 & \text{if } i = j \\ \frac{\frac{1}{N_0}}{\sqrt{(\frac{1}{N_i}+\frac{1}{N_0})\cdot(\frac{1}{N_j}+\frac{1}{N_0})}} & \text{otherwise} \end{cases} \tag{53}$$

$$T_i \sim t_\phi(\lambda_i) \tag{54}$$

$$1 - \beta_i = \Pr(|T_i| > c_\alpha), \tag{55}$$

where $1 \le i, j \le k$; N_0 and N_i are the sample sizes in the control and the i-th treatment group, respectively; μ_0 and μ_i are the corresponding group means; S denotes the common standard deviation (pooled across groups); $t_\phi(\lambda_i)$ is the noncentral t-distribution with ϕ degrees of freedom and noncentrality λ_i; Pr denotes the probability function; and c_α is the two-sided Dunnett critical value that controls the family-wise error rate at α.

For estimation, μ_0 and μ_i were taken as the sample group means. The common standard deviation was estimated as

$$S = \sqrt{\frac{SS_E}{\phi_E}} \tag{56}$$

(see Equations (71)–(72) for the definition of SS_E and ϕ_E). The Dunnett two-sided critical value c_α [9] corresponding to $\alpha = 0.05$ was obtained using the R package `mvtnorm` [40] by inverting the central multivariate t-distribution with correlation matrix R.

4.4.2 Additive models

Under the *additive model*—a multivariate model assuming no interaction terms—we modeled n factors at a replication level N ($\forall n, N \in \mathbb{N}$), the output value $y_{l_1,\cdots,l_n,r}$ when $\forall i$-th factors are assigned to their l_i-th level ($\forall i, \exists L_i, \forall l_i \in \mathbb{N}$ such that $1 \le i \le n$ and $1 \le l_i \le L_i$) at replication index r ($1 \le r \le N$) is modeled as follows:

$$\begin{aligned} y_{l_1,\cdots,l_n,r} &:= \mu + \sum_{i=1}^{n} M^i_{l_i} + \varepsilon_{l_1,\cdots,l_n,r} \\ &\sim \mathcal{N}(\mu + \sum_{i=1}^{n} M^i_{l_i}, \sigma^2), \end{aligned} \tag{57}$$

where μ is the population mean, $M^i_{l_i}$ is the main effect of the i-th factor at the l_i-th level, and $\varepsilon_{l_1,\cdots,l_n,r} \sim \mathcal{N}(0, \sigma^2)$ is the Gaussian noise term.

As described in Methods 4.1.4, main effects $M^i_{l_i}$ can be simplified to M^i to have the following:

$$y_j = \mu + \sum_{i=1}^{n} M^i \cdot \tilde{c}_{i,j} + \varepsilon_j, \tag{58}$$

where $\tilde{c}_{i,j} \in \{-1, 1\}$ is the element of a design matrix (refer to Equation (6)) assigning specific experimental conditions. Notably, $\tilde{c}_{i,j}$ can be introduced by rescaling $c_{i,j}$—the perturbation indicators introduced in Models Φ, Ψ, and Λ—with $\tilde{c}_{i,j} = 2c_{i,j} - 1$. Moreover, y_j and ε_j are equivalent to $y_{l_1,\cdots,l_n,r}$ or $\varepsilon_{l_1,\cdots,l_n,r}$ introduced in Equation (57), but indexed with the trial number j.

With a design matrix $D(n, m) = [\mathbf{x}_1 \cdots \mathbf{x}_n]$, Equation (58) can be represented as follows:

$$\begin{aligned} \mathbf{y} &= \begin{bmatrix} \mathbf{1} & \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_n \end{bmatrix} \begin{bmatrix} \mu \\ M^1 \\ M^2 \\ \vdots \\ M^n \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_j \end{bmatrix} \\ &=: X\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \end{aligned} \tag{59}$$

where $\mathbf{y}$ is the vector for output values y_j, X is the model matrix, and $\boldsymbol{\beta}$ is the coefficient vector containing μ and M^i. $\boldsymbol{\varepsilon}$ is the residual error vector following the m-dimensional Gaussian distribution $\mathcal{N}_m(\mathbf{0}, \sigma^2\mathbf{I}_m)$, where $\mathbf{I}_m$ denotes the $m \times m$ identity matrix.

4.4.3 MLR

To visualize the results from additive models with a simple method, we created ordinary least square MLR models referring Equation (59). By minimizing the mean squared error, the estimated coefficient vector $\hat{\boldsymbol{\beta}}$ is determined as follows:

$$\begin{aligned} \hat{\boldsymbol{\beta}} &:= \arg\min(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})^\top(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}) \\ &= (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}. \end{aligned} \tag{60}$$

We implemented MLR models with `statsmodels` and visualized the MLR coefficients with bar charts using Python packages.

4.4.4 ANOVA

We adopted Type II ANOVA, as recommended by Langsrud (2003) [41]. Since interaction terms are undetectable in OFAT designs (the superfamily of C+LOO; see Methods 4.1.1) [3] and PB designs rely on a first-degree polynomial model [25], Type II ANOVA is deemed appropriate for these contexts as it allows for analysis without postulating interaction terms [41].

To perform Type II ANOVA, we introduce three different model matrices X_Ω, X_i, and $X_{\Omega-i}$ as follows:

$$X_\Omega := \begin{bmatrix}\mathbf{1} & \mathbf{x}_1 & \cdots & \mathbf{x}_{i-1} & \mathbf{x}_i & \mathbf{x}_{i+1} & \cdots & \mathbf{x}_n\end{bmatrix} \quad (61)$$

$$X_i := \begin{bmatrix}\mathbf{x}_i\end{bmatrix} \quad (62)$$

$$X_{\Omega-i} := \begin{bmatrix}\mathbf{1} & \mathbf{x}_1 & \cdots & \mathbf{x}_{i-1} & \mathbf{x}_{i+1} & \cdots & \mathbf{x}_n\end{bmatrix}. \quad (63)$$

Notably, X_Ω is the model matrix of additive models, whereas X_i and $X_{\Omega-i}$ represent the partial models of X_Ω which parameters are exclusive and complementary to each other.

With these model matrices, the additive model can be derived as follows:

$$\begin{aligned}\mathbf{y} &= X_\Omega \boldsymbol{\beta}_\Omega + \boldsymbol{\varepsilon} \\ &= X_{\Omega-i}\boldsymbol{\beta}_{\Omega-i} + X_i\boldsymbol{\beta}_i + \boldsymbol{\varepsilon} \\ &\sim \mathcal{N}_{n_{\max}}(X_\Omega\boldsymbol{\beta}_\Omega, \sigma^2\mathbf{I}_{n_{\max}})\end{aligned} \quad (64)$$

where $\boldsymbol{\beta}_\Omega$, $\boldsymbol{\beta}_{\Omega-i}$, and $\boldsymbol{\beta}_i$ are the coefficient vectors for the respective model matrices.

Under Type II ANOVA, the sum of squares for the i-th factor $SS(X_i)$ are indirectly derived from the total sum of squares $SS(X_\Omega)$ subtracted with $SS(X_{\Omega-i})$ for given observations $\mathbf{y}$ as follows:

$$SS(X_\Omega) := \mathbf{y}^\top P(X_\Omega)\mathbf{y} \quad (65)$$

$$SS(X_{\Omega-i}) := \mathbf{y}^\top P(X_{\Omega-i})\mathbf{y} \quad (66)$$

$$\begin{aligned}SS(X_i) :=& SS(X_\Omega) - SS(X_{\Omega-i}) \\ =& \mathbf{y}^\top(P(X_\Omega) - P(X_{\Omega-i}))\mathbf{y} \\ =& \mathbf{y}^\top P((\mathbf{I}_{n_{\max}} - P(X_{\Omega-i}))X_i)\mathbf{y},\end{aligned} \quad (67)$$

where $P(A) := A(A^\top A)^{-1}A^\top$ denotes the orthogonal projection matrix of an arbitrary matrix A.

Regarding $SS(X_i)$, the following equations hold:

$$\phi_i := \mathrm{rank}((\mathbf{I}_{n_{\max}} - P(X_{\Omega-i}))X_i) \quad (68)$$

$$\lambda_i := \frac{\|(\mathbf{I}_{n_{\max}} - P(X_{\Omega-i}))X_i\boldsymbol{\beta}_i\|^2}{\sigma^2} \quad (69)$$

$$\frac{SS(X_i)}{\sigma^2} \sim \chi^2_{\phi_i}(\lambda_i), \quad (70)$$

where $\chi^2_{\phi_i}(\lambda_i)$ denotes a noncentral chi-squared distribution with ϕ_i degrees of freedom and a noncentrality parameter λ_i.

Similarly, the error sum of squares SS_E is defined as follows:

$$\begin{aligned}SS_E :=& \mathbf{y}^\top\mathbf{y} - \mathbf{y}^\top P(X_\Omega)\mathbf{y} \\ =& \mathbf{y}^\top(\mathbf{I}_{n_{\max}} - P(X_\Omega))\mathbf{y}.\end{aligned} \quad (71)$$

In practice, the noncentrality parameter λ_E for SS_E/σ^2 is assumed to be zero to avoid the doubly noncentral F distribution regarding the F-statistics (refer to SI F). Therefore, the following equations hold:

$$\phi_E := n_{\max} - \mathrm{rank}(X_\Omega) \quad (72)$$

$$\frac{SS_E}{\sigma^2} \sim \chi^2_{\phi_E}, \quad (73)$$

$$F_i := \frac{\frac{SS(X_i)}{\phi_i}}{\frac{SS_E}{\phi_E}} \sim F_{\phi_i,\phi_E}(\lambda_i), \quad (74)$$

where $\chi^2_{\phi_E}$ denotes a chi-squared distribution with ϕ_E degrees of freedom, F_i is the F-statistic for the i-th main effect, and $F_{\phi_i,\phi_E}(\lambda_i)$ denotes the noncentral F-distribution with ϕ_i and ϕ_E degrees of freedom and a noncentrality parameter λ_i.

In the statistical testing on the i-th factor main effect, the null hypothesis H_0 assumes $\boldsymbol{\beta}_i := [\,_{M^i}\,] = \mathbf{0}$. Hence, the p-values are calculated under assumption that $\lambda_i = 0$.

In contrast, in the power analysis of the statistical testing, the statistical power $1-\beta$ is calculated under the noncentral F-distribution. More precisely, the upper probability of the noncentral F-distribution is calculated for an F such that the upper probability of the corresponding F-distribution equals the α used in the statistical testing. The noncentrality parameter λ_i was estimated as follows:

$$\hat{\lambda}_i := \frac{\|(\mathbf{I}_{n_{\max}} - P(X_{\Omega-i}))X_i\hat{\boldsymbol{\beta}}_i\|^2}{\sigma^2} \quad (75)$$

$$\hat{\boldsymbol{\beta}}_\Omega := (X_\Omega^\top X_\Omega)^{-1}X_\Omega^\top\mathbf{y}, \quad (76)$$

where $\hat{\boldsymbol{\beta}}_i$ corresponds to the i-th element (or sub-vector) of the estimated coefficient vector $\hat{\boldsymbol{\beta}}_\Omega$.

Notably, the implementation described above is mathematically equivalent to the formulas derived from the definitions of Type II ANOVA in Langsrud (2003) [41] and the general linear model framework in Muller and Peterson (1984) [42].

We implemented Type II ANOVA, power analysis, and visualization with Python packages including `statsmodels` and `SciPy`.

4.4.5 Visualization of MLR/ANOVA results

In Figures S6b–S6c, we visualized the results from MLR and ANOVA for Model Φ-based additive models using lollipop plots. To visualize both the direction and magnitude of the factor effects, we calculated the signed partial eta-squared values ($\mathrm{sign}(\hat{\beta})\eta_p^2$), defined as the product of

the sign of the MLR coefficient ($\text{sign}(\hat{\beta})$) and the partial-eta squared (η_p^2) [43]. The partial eta-squared for the i-th factor ($\eta_{p,i}^2$) is defined as follows:

$$\eta_{p,i}^2 := \frac{SS(X_i)}{SS(X_i) + SS_E}. \tag{77}$$

We computed the MLR coefficients and sums of squares using `statsmodels` and derived the signed metrics using `NumPy`.

4.4.6 Main effect estimation

After generating simulated results based on the FF design with 100 different random seeds, we calculated the main effects according to Equation (8) and estimated the bootstrap 95% CIs with `SciPy`, setting the number of bootstrapping $n_{\text{boot}} = 10000$. Then we interpreted the results based on the CIs as follows:

- "Up" if the lower bound of the CI is above 0
- "N.S." the CI includes 0
- "Down" if the upper bound of the CI is below 0

We leveraged these labels as the ground truth when comparing the experimental design performance.

4.4.7 Half-normal plot

The half-normal plot shown in Figure S6e is a graphical method to visualize the factor effects which is versatile even when ANOVA does not work—e.g., $\phi_E = 0$. The effects are suggested to be significant when the absolute values of the standardized effects are larger than the half-normal quantiles [44, 45].

We created a half-normal plot for the C+LOO-based Model Φ results at $N = 1$. The half-normal quantiles were calculated using `SciPy` implementation, and the standardized effects were computed with `NumPy`. Then, we visualized these results with `Matplotlib` and other Python packages.

4.5 Performance evaluation metrices

4.5.1 Weighted Cohen's κ

When evaluating the main effects in the simulators from generated results, we applied the following criteria to interpret the multivariate models: the effect of a factor is positive if the MLR coefficient is positive and ANOVA suggests $p < 0.05$; negative if the coefficient is negative and $p < 0.05$; and N.S. if $p \geq 0.05$.

We quantified the agreement between result interpretations by calculating weighted Cohen's κ coefficients relative to the ground truth labels (Methods 4.4.6).

To reduce strictness in identifying statistically significant effects as N.S. (or vice versa) and to relatively penalize predicting effects in opposite directions, we applied linear weights (0, 0.5, and 1) when calculating κ values.

We implemented this method using Python packages, including `NumPy`, `pandas`, and `scikit-learn`, and visualized the results with `Matplotlib` and `seaborn`.

4.5.2 Aptitude score

Cohen's κ is defined as follows [46]:

$$\kappa := \frac{\Pr(o) - \Pr(e)}{1 - \Pr(e)}, \tag{78}$$

where $\Pr(o)$ and $\Pr(e)$ denote the observed and expected agreement, respectively. As shown in Equation (78), κ is undefined when $\Pr(e) = 1$, a limitation that also applies to Weighted Cohen's κ.

To ensure robust evaluation of experimental design performance under diverse conditions, we extended the standard weighted Cohen's κ used in Models Φ, Ψ, and Λ by introducing the *aptitude score*, which returns raw accuracy values in cases where κ is undefined.

To compute aptitude scores, we measured the agreement between the results predicted by the multivariate model based on the experimental design of interest (C+LOO or PB) and the ground truth labels.

We implemented this metric using Python packages, including `NumPy`, `pandas`, and `scikit-learn`.

4.6 The D-criterion and D-optimization

With respect to the model matrix X (introduced in Methods 4.1.7), $X^\top X$ is called the information matrix, and $(X^\top X)^{-1}$ is the dispersion matrix [16]. The determinant of the dispersion matrix $\det(X^\top X)^{-1}$ is called the D-criterion.

D-optimization is a process of choosing the model matrix, as well as the design matrix which forms the model matrix, that minimizes the D-criterion. Such matrices are considered optimal for experimental designs because the lowest D-criterion indicates that the coefficients of the multivariate model are as independent of each other as possible [16]. Note that minimizing the determinant of the dispersion matrix are equivalent to maximizing the determinant of the information matrix [16].

We visualized the D-criterion values with Python packages including `Matplotlib`, `NumPy`, `seaborn`, and `statsmodels`. We performed D-optimization using an R package `AlgDesign` [47] and visualized the results with Python packages.

4.7 Terminology and features for simulator network structures

4.7.1 Terminology for feature engineering

To numerically characterize network structures, here we introduce a terminology to describe network structures of simulated models. *pathway*, *regulation*, and *cascade*. A *pathway* refers to an edge that directly leads to the

output value, while a *regulation* denotes an edge connecting two factors. Each edge can take one of three forms—positive, null, or negative (Figure 4a)—resulting in a power-of-three number of possible network structures for the simulated models. A *cascade* is a chain of a pathway and one or more regulations that are linearly connected. Notably, a single pathway without any upstream regulations is also a cascade.

Furthermore, if a regulation is directly or indirectly connected to a pathway—that is, if it forms part of a cascade—we refer to it as an *effective* regulation. In contrast, any regulation that lacks a downstream pathway is considered *ineffective* (Figure 4b).

To highlight impactful cascades that influence the output value, we introduce *enhancing cascades* and *positive cascades* (Figure 4c). A cascade is considered *enhancing* if it either has no regulations or if all existing regulations are positive. An enhancing cascade is *positive* when the pathway is also positive. Thus, all positive cascades are enhancing. Having at least one negative regulation, the effect of the pathway in non-enhancing cascades is directly or indirectly downregulated within the cascade itself.

Then, for all pathways, we define *regulation weight* and *factor weight*. A *regulation weight* is a lump-sum number of regulations belonging to cascades with the pathway of interest. For example, we present a graphical example in Figure 4d. The most upstream factor has zero regulation weight. The factor marked with orange has a regulation from the blue one but any pathway downstream. In such a case, regulation weight is not defined. For the sake of consistency, we assign zero instead. The factor marked with green has two upstream regulations—edges from the blue one to the orange one, and from the orange one to the green one—hence, the regulation weight becomes two. Having three regulations upstream, the purple factor has a pathway with regulation weight of three. Similarly, a *factor weight* is the number of factors involving to the pathway of interest. As Figure 4e shows, factor weight for the pathway in red is three since there are three factors directly or indirectly involving the pathway. Notably, we also assigned zero factor weight when a factor has no pathway.

As an alternative feature, we introduce *cascade length*, a total number of edges in the cascade containing the pathway of interest. Since a pathway can be a part of multiple cascades, the largest cascade represents the pathway in such cases. Thus, the cascade length for the pathway on the right in Figure 4f is three.

4.7.2 The 31 features of the simulator network structures

Based on the terminology introduced above, we defined 31 features—including pathway-related, regulation-related, regulation weight-related, factor weight-related, and cascade length-related features—for a network with n factors, p pathways, and r regulations ($n, r, p \in \mathbb{Z}$ such that $n \geq 1$ and $r, p \geq 0$). Notably, we defined the values to always fall within the range $[0, 1]$ for all n to gain generalizability for any number of factors.

As pathway-related features, we introduce *pathway coverage* P%, *positive pathway coverage* $\mathrm{P}_{(+)}\%$, *negative pathway coverage* $\mathrm{P}_{(-)}\%$, and *pathway positivity* $\mathrm{P}_{(+)}/\mathrm{P}$, and *pathway negativity* $\mathrm{P}_{(-)}/\mathrm{P}$ as follows:

$$\mathrm{P}\% := \frac{p}{n} \tag{79}$$

$$\mathrm{P}_{(+)}\% := \frac{p_+}{n} \tag{80}$$

$$\mathrm{P}_{(-)}\% := \frac{p_-}{n} \tag{81}$$

$$\mathrm{P}_{(+)}/\mathrm{P} := \begin{cases} \frac{p_+}{p} & \text{if } p \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{82}$$

$$\mathrm{P}_{(-)}/\mathrm{P} := \begin{cases} \frac{p_-}{p} & \text{if } p \neq 0 \\ 0 & \text{otherwise,} \end{cases} \tag{83}$$

where p_+ and p_- denote the number of positive and negative pathways, respectively. The metrics P%, $\mathrm{P}_{(+)}\%$, and $\mathrm{P}_{(-)}\%$ reflect the abundance of all, positive, and negative pathways, respectively, and are computed as ratios normalized by the number of factors n, which defines the maximum possible number of pathways. In contrast, $\mathrm{P}_{(+)}/\mathrm{P}$ and $\mathrm{P}_{(-)}/\mathrm{P}$ represent the proportions of positive and negative pathways relative to the total number of pathways.

Subsequently, we introduce regulation-related features—namely, *regulation coverage* R%, *positive regulation coverage* $\mathrm{R}_{(+)}\%$, *negative regulation coverage* $\mathrm{R}_{(-)}\%$, *regulation positivity* $\mathrm{R}_{(+)}/\mathrm{R}$, *regulation negativity* $\mathrm{R}_{(-)}/\mathrm{R}$, *regulation effectivity* R^*/R, *effective regulation positivity* $\mathrm{R}^*_{(+)}/\mathrm{R}^*$, and *effective regulation negativity* $\mathrm{R}^*_{(-)}/\mathrm{R}^*$—as follows:

$$\mathrm{R}\% := \frac{r}{\binom{n}{2}} \tag{84}$$

$$\mathrm{R}_{(+)}\% := \frac{r_+}{\binom{n}{2}} \tag{85}$$

$$\mathrm{R}_{(-)}\% := \frac{r_-}{\binom{n}{2}} \tag{86}$$

$$\mathrm{R}_{(+)}/\mathrm{R} := \begin{cases} \frac{r_+}{r} & \text{if } r \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{87}$$

$$\mathrm{R}_{(-)}/\mathrm{R} := \begin{cases} \frac{r_-}{r} & \text{if } r \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{88}$$

$$\mathrm{R}^*/\mathrm{R} = \begin{cases} \frac{r^*}{r} & \text{if } r \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{89}$$

$$\mathrm{R}^*_{(+)}/\mathrm{R}^* = \begin{cases} \frac{r^*_+}{r^*} & \text{if } r^* \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{90}$$

$$\mathrm{R}^*_{(-)}/\mathrm{R}^* = \begin{cases} \frac{r^*_-}{r^*} & \text{if } r^* \neq 0 \\ 0 & \text{otherwise,} \end{cases} \tag{91}$$

where r_+, r_-, r^*, r^*_+, and r^*_- denote the number of positive, negative, effective, positive effective, and negative effective regulations, respectively. Given that the theoretical maximum number of regulations for n factors is $\binom{n}{2}$, the values R%, $\mathrm{R}_{(+)}$%, and $\mathrm{R}_{(-)}$% are normalized by this quantity. The ratios $\mathrm{R}_{(+)}/\mathrm{R}$, $\mathrm{R}_{(-)}/\mathrm{R}$, and R^*/R are computed relative to the total number of regulations, while $\mathrm{R}^*_{(+)}/\mathrm{R}^*$ and $\mathrm{R}^*_{(-)}/\mathrm{R}^*$ reflect the proportions of positive and negative edges among effective regulations, respectively.

Then, we introduce regulation weight-related features: *maximum regulation density* $\mathrm{MaxRW}/\mathrm{R}^*$, *regulation weight coverage* RW%, *maximum positive regulation density* $\mathrm{MaxRW}_{(+)}/\mathrm{R}^*$, *positive regulation weight coverage* $\mathrm{RW}_{(+)}$%, *maximum enhancing regulation density* $\mathrm{MaxRW}_{\&}/\mathrm{R}^*$, and *enhancing regulation weight coverage* $\mathrm{RW}_{\&}$%.

$$\mathrm{MaxRW}/\mathrm{R}^* := \begin{cases} \frac{\max\{f_{\mathrm{RW}}(\pi)|\pi\in\Pi\}}{r^*} & \text{if } r^* \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{92}$$

$$\mathrm{RW}\% := \begin{cases} \frac{\sum_{\pi\in\Pi} f_{\mathrm{RW}}(\pi)}{\sum_{i=1}^{n}\sum_{j=0}^{i-1} j} & \text{if } n \geq 2 \\ 0 & \text{otherwise} \end{cases} \tag{93}$$

$$\mathrm{MaxRW}_{(+)}/\mathrm{R}^* := \begin{cases} \frac{\max\{f_{\mathrm{RW}_{\&}}(\pi)|\pi\in\Pi_+\}}{r^*} & \text{if } r^* \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{94}$$

$$\mathrm{RW}_{(+)}\% := \begin{cases} \frac{\sum_{\pi\in\Pi_+} f_{\mathrm{RW}_{\&}}(\pi)}{\sum_{i=1}^{n}\sum_{j=0}^{i-1} j} & \text{if } n \geq 2 \\ 0 & \text{otherwise} \end{cases} \tag{95}$$

$$\mathrm{MaxRW}_{\&}/\mathrm{R}^* := \begin{cases} \frac{\max\{f_{\mathrm{RW}_{\&}}(\pi)|\pi\in\Pi\}}{r^*} & \text{if } r^* \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{96}$$

$$\mathrm{RW}_{\&}\% := \begin{cases} \frac{\sum_{\pi\in\Pi} f_{\mathrm{RW}_{\&}}(\pi)}{\sum_{i=1}^{n}\sum_{j=0}^{i-1} j} & \text{if } n \geq 2 \\ 0 & \text{otherwise,} \end{cases} \tag{97}$$

where Π and Π_+ denote the sets of all pathways and positive pathways, respectively; f_{RW} and $f_{\mathrm{RW}_{\&}}$ are functions that return the total regulation weights of a given pathway or those restricted to enhancing cascades ($f_{\mathrm{RW}_{\&}}$ returns zero for pathways of non-enhancing cascades); and max denotes the function returning the maximum value in each set. The values $\mathrm{MaxRW}/\mathrm{R}^*$, $\mathrm{MaxRW}_{(+)}/\mathrm{R}^*$, and $\mathrm{MaxRW}_{\&}/\mathrm{R}^*$ respectively represent the impact of maximum regulation weight among cascades, positive cascades, and effective cascades, divided by the number of effective regulations. RW%, $\mathrm{RW}_{(+)}$%, and $\mathrm{RW}_{\&}$% represent the coverages of regulation weights of respective categories normalized by the theoretical maximum number $\sum_{i=1}^{n}\sum_{j=0}^{i-1} j = \frac{n(n+1)(n-1)}{6}$.

Similarly, factor weight-related factors, such as *maximum factor density* MaxFW/n, *factor weight coverage* FW%, *maximum positive factor density* $\mathrm{MaxFW}_{(+)}/\mathrm{n}$, *positive factor weight coverage* $\mathrm{FW}_{(+)}$%, *maximum enhancing factor density* $\mathrm{MaxFW}_{\&}/\mathrm{n}$, and *enhancing factor weight coverage* $\mathrm{FW}_{\&}$% are defined as follows:

$$\mathrm{MaxFW}/\mathrm{n} := \frac{\max\{f_{\mathrm{FW}}(\pi)|\pi\in\Pi\}}{n} \tag{98}$$

$$\mathrm{FW}\% := \frac{\sum_{\pi\in\Pi} f_{\mathrm{FW}}(\pi)}{\sum_{i=1}^{n} i} \tag{99}$$

$$\mathrm{MaxFW}_{(+)}/\mathrm{n} := \frac{\max\{f_{\mathrm{FW}_{(+)}}(\pi)|\pi\in\Pi_+\}}{n} \tag{100}$$

$$\mathrm{FW}_{(+)}\% := \frac{\sum_{\pi\in\Pi_+} f_{\mathrm{FW}_{(+)}}(\pi)}{\sum_{i=1}^{n} i} \tag{101}$$

$$\mathrm{MaxFW}_{\&}/\mathrm{n} := \frac{\max\{f_{\mathrm{FW}_{(+)}}(\pi)|\pi\in\Pi\}}{n} \tag{102}$$

$$\mathrm{FW}_{\&}\% := \frac{\sum_{\pi\in\Pi} f_{\mathrm{FW}_{(+)}}(\pi)}{\sum_{i=1}^{n} i}, \tag{103}$$

where f_{FW} and $f_{\mathrm{FW}_{(+)}}$ denotes functions returning factor weights or those of positive cascades for a given pathway, respectively.

Finally, we introduce cascade length-related features, *maximum cascade length scale* MaxCL/n, *cascade length coverage* CL%, *maximum positive cascade length scale* $\mathrm{MaxCL}_{(+)}/\mathrm{n}$, *positive cascade length coverage* $\mathrm{CL}_{(+)}$%, *maximum enhancing cascade length scale* $\mathrm{MaxCL}_{\&}/\mathrm{n}$, and *enhancing cascade length coverage* $\mathrm{CL}_{\&}$%, as follows:

$$\mathrm{MaxCL}/\mathrm{n} := \frac{\max\{f_{\mathrm{CL}}(\pi)|\pi\in\Pi\}}{n} \tag{104}$$

$$\mathrm{CL}\% := \frac{\sum_{\pi\in\Pi} f_{\mathrm{CL}}(\pi)}{\sum_{i=1}^{n} i} \tag{105}$$

$$\mathrm{MaxCL}_{(+)}/\mathrm{n} := \frac{\max\{f_{\mathrm{CL}_{(+)}}(\pi)|\pi\in\Pi_+\}}{n} \tag{106}$$

$$\mathrm{CL}_{(+)}\% := \frac{\sum_{\pi\in\Pi_+} f_{\mathrm{CL}_{(+)}}(\pi)}{\sum_{i=1}^{n} i} \tag{107}$$

$$\mathrm{MaxCL}_{\&}/\mathrm{n} := \frac{\max\{f_{\mathrm{CL}_{(+)}}(\pi)|\pi\in\Pi\}}{n} \tag{108}$$

$$\mathrm{CL}_{\&}\% := \frac{\sum_{\pi\in\Pi} f_{\mathrm{CL}_{(+)}}(\pi)}{\sum_{i=1}^{n} i}, \tag{109}$$

where f_{CL} and $f_{\mathrm{CL}_{(+)}}$ denotes functions returning cascade lengths or those of positive cascades for a given pathway, respectively.

4.8 Exhaustive search of network structure with four factors

4.8.1 ESM4: a template for diverse simulated models of different network structures

To explore all possible network structures for certain features that may serve as indicators of the effectiveness of the C+LOO design, we created ESM4, a template model of network structures that can generate various simulated models.

Define the following for all $i, j \in \mathbb{N}$ such that $1 \leq i \leq 4$ and $1 \leq j \leq n_{\max}$:

- $x_{i,j}$, y_j, $c_{i,j}$, $\varepsilon_{i,j}$, $\varepsilon_{y,j}$, ReLU: identical to those in Models Φ and Ψ
- $K := \{1, 2, 3, 4\}$
- $L := \{12, 13, 14, 23, 24, 34\}$
- $a_k \sim \text{Lognormal}(2, (0.3)^2)$ for $k \in K$
- $b_\ell \sim \text{Lognormal}(1, (0.5)^2)$ for $\ell \in L$
- $s_\lambda \in \{-1, 0, 1\}$ for $\lambda \in K \cup L$: edge status indicator (-1, negative; 0, null; and 1, positive)

Notably, s_λ modulates the connectivity patterns and the polarity of factor effects. By varying s_λ values, ESM4 can generate 59,049 different simulated models.

As in the other simulated models, the parameters v_i, a_k, and b_ℓ are shared across trials and network structures. They were sampled using `NumPy` with a fixed random seed.

The model equations are defined as follows:

$$x_{1,j} := \text{ReLU}(c_{1,j}(v_1 + \varepsilon_{1,j})) \tag{110}$$

$$x_{2,j} := \text{ReLU}(c_{2,j}(v_2 + s_{12}b_{12}x_{1,j} + \varepsilon_{2,j})) \tag{111}$$

$$\begin{aligned} x_{3,j} := \text{ReLU}(c_{3,j}(v_3 &+ s_{13}b_{13}x_{1,j} \\ &+ s_{23}b_{23}x_{2,j} + \varepsilon_{3,j})) \end{aligned} \tag{112}$$

$$\begin{aligned} x_{4,j} := \text{ReLU}(c_{4,j}(v_4 &+ s_{14}b_{14}x_{1,j} + s_{24}b_{24}x_{2,j} \\ &+ s_{34}b_{34}x_{3,j} + \varepsilon_{4,j})) \end{aligned} \tag{113}$$

$$y_j := \text{ReLU}(\sum_{i \in K} s_i a_i x_{i,j} + \varepsilon_{y,j}). \tag{114}$$

Downstream analyses—namely, simulated experiments and ANOVA—for the models generated with ESM4 were performed in the same way with the other simulated models.

4.8.2 Datasets for the ML model

Using the ESM4 template, we generated 59,049 simulators—all possible network structures for simulators consisted of four factors and their DAGs. After evaluating the C+LOO- and PB-aptitude scores for each simulator at $N = 3$, we classified them into the binary classifications—"C+LOO-suited" and "PB-suited"—referring their aptitude scores for C+LOO (κ^*_{C}) and PB (κ^*_{P}) designs. In cases where $\kappa^*_{\text{C}} = \kappa^*_{\text{P}}$, the C+LOO design was defined as the "C+LOO-suited" class, since it requires a smaller total number of trials (Figure 1d and Methods 4.1.3). We also calculated the 31 feature values detailed in Methods 4.7.2 for each simulator.

After the formation of the dataset consisting of the simulator categories and the 31 network features, we performed a random hold-out to split the dataset into the training, validation, and test sets in approximate proportions of 3:1:1.

4.8.3 ML binary classification model

To find plausible features that are associated with high performance of the PB design with a data-driven manner, we created an ML model to predict the simulator classes from the 31 network features. We adopted a GBDT model provided as LightGBM [48]. To optimize the hyper parameters in the LightGBM model, we adopted the `LightGBMTuner`, a LightGBM model employed in `Optuna` [49] that wraps LightGBM training and hyper-parameter optimization.

The performance of the GBDT model was evaluated with an ROC curve and its AUROC value. We implemented the ML model evaluation with `scikit-learn` and other Python packages.

4.8.4 SHAP scores

To understand which feature has the largest contribution to the classification, we computed SHAP scores to visualize the correspondence of feature values to the model output. Since the Python package SHAP [50] can analyze tree-based pretrained models including LightGBM, we passed the GBDT model introduced in Methods 4.8.3 and created a beeswarm plot for the "PB-suited" class showing the top 10 features with the highest impact on the ML model output.

4.8.5 PBSI

To determine which of the C+LOO or PB designs is more suitable for a given phenomenon, it is important to identify the conditions under which the PB design can outperform the C+LOO design, which is the standard choice for Perturb-seq experiments. Compared to creating ML models based on a comprehensive dataset of network features and experimental design performance for a desired

number of factors—as we did with ESM4, having an indicator that predicts high performance with the PB design is an efficient way to streamline experimental planning.

Since single features did not show sufficient prediction performance for the "PB-suited" class (Figure 5e), we developed an integrative metric identifying high performance of the PB design compared to the C+LOO design. We term this metric the PBSI, defined as follows:

$$\text{PBSI} := \frac{\sum_{\varphi \in \Phi} \varphi}{3}, \tag{115}$$

where $\Phi := \{\text{P}_{(+)}\%, 1 - \text{MaxRW}_{(+)}/\text{R}^*, 1 - \text{RW}_{(+)}\%\}$. Notably, we engineered PBSI by exhaustively searching all possible combinations of the top six individual features in terms of predictive performance—$1 - \text{MaxRW}_{(+)}/\text{R}^*$, $1 - \text{RW}_{(+)}\%$, $1 - \text{MaxRW}_{\&}/\text{R}^*$, $\text{P}_{(+)}\%$, $1 - \text{R}^*_{(+)}/\text{R}^*$, and P%—and selecting the combination that yielded the highest AUROC in the ESM4 dataset (SI I.1).

4.8.6 Prediction performance evaluation for the individual network features and composite indices

As described in Methods 4.7.2, the 31 features were designed to take values within the range $[0, 1]$. The composite indices constructed by calculating the arithmetic means of feature combinations also shared this property. Using these values directly, we evaluated the ROC curves, AUROC, and average precision (AP) scores to measure the predictive performance of each metric for the "PB-suited" class.

We applied `StratifiedKFold` from `scikit-learn` to divide the ESM4 or ESM9 datasets (Methods 4.8.7) into 30 subsets, and calculated ROC curves and AUROC scores for each fold. For the ROC curves, we linearly interpolated the curves across folds using `SciPy`, accompanied by the raw ROC curves from each fold. For the AUROC values, we calculated the mean and 95% bootstrap CIs across folds using `SciPy` and visualized them as error bars or annotated them directly in the figures.

Since PBSI was designed as a simple, model-free metric, it was calculated as the arithmetic mean of $\text{P}_{(+)}\%$, $1 - \text{MaxRW}_{(+)}/\text{R}^*$, and $1 - \text{RW}_{(+)}\%$. To examine the potential influence of unoptimized weighting, we built logistic regression models following the same procedure, fitting weights for each component and comparing their predictive performance with that of PBSI (SI I.1–I.2). The logistic regression models were implemented using `scikit-learn`.

4.8.7 ESM9: another exhaustive search model for nine factors

To evaluate the scalability of the features and metrics associated with relative high performance of PB designs, we created ESM9, another template model that can generate diverse simulators.

Define the following for all $i, j \in \mathbb{N}$ such that $1 \leq i \leq 9$ and $1 \leq j \leq n_{\max}$:

- $x_{i,j}$, y_j, $c_{i,j}$, $\varepsilon_{i,j}$, $\varepsilon_{y,j}$, ReLU: identical to those in Models Φ and Ψ
- $K := \{i \mid 1 \leq i \leq 9\}$
- $L := \{10i^* + i_* \mid 1 \leq i^* \leq 8,\ i^* + 1 \leq i_* \leq 9\}$
- $a_k \sim \text{Lognormal}(2, (0.3)^2)$ for $k \in K$
- $b_\ell \sim \text{Lognormal}(1, (0.5)^2)$ for $\ell \in L$
- $s_\lambda \in \{-1, 0, 1\}$ for $\lambda \in K \cup L$: edge status indicator (-1, negative; 0, null; and 1, positive)

Notably, s_λ modulates the connectivity patterns and the polarity of factor effects. By varying s_λ values, ESM9 can generate 2,954,312,706,550,833,698,643 ($= 3^{45}$) different simulators. To reduce computational cost, we randomly sampled 20,000 network configurations and evaluated the C+LOO- and PB-aptitude scores at $N = 3$.

As in the other simulated models, the parameters v_i, a_k, and b_ℓ are shared across trials and network structures. They were sampled using `NumPy` with a fixed random seed.

The model equations are defined as follows:

$$x_{1,j} := \text{ReLU}(c_{1,j}(v_1 + \varepsilon_{1,j})) \tag{116}$$

$$x_{2,j} := \text{ReLU}(c_{2,j}(v_2 + \sum_{\iota=1}^{2} s_{10\iota+2} b_{10\iota+2} x_{\iota,j} + \varepsilon_{2,j})) \tag{117}$$

$$x_{3,j} := \text{ReLU}(c_{3,j}(v_3 + \sum_{\iota=1}^{3} s_{10\iota+3} b_{10\iota+3} x_{\iota,j} + \varepsilon_{3,j})) \tag{118}$$

$$x_{4,j} := \text{ReLU}(c_{4,j}(v_4 + \sum_{\iota=1}^{4} s_{10\iota+4} b_{10\iota+4} x_{\iota,j} + \varepsilon_{4,j})) \tag{119}$$

$$x_{5,j} := \text{ReLU}(c_{5,j}(v_5 + \sum_{\iota=1}^{5} s_{10\iota+5} b_{10\iota+5} x_{\iota,j} + \varepsilon_{5,j})) \tag{120}$$

$$x_{6,j} := \text{ReLU}(c_{6,j}(v_6 + \sum_{\iota=1}^{6} s_{10\iota+6} b_{10\iota+6} x_{\iota,j} + \varepsilon_{6,j})) \tag{121}$$

$$x_{7,j} := \text{ReLU}(c_{7,j}(v_7 + \sum_{\iota=1}^{7} s_{10\iota+7} b_{10\iota+7} x_{\iota,j} + \varepsilon_{7,j})) \tag{122}$$

$$x_{8,j} := \text{ReLU}(c_{8,j}(v_8 + \sum_{\iota=1}^{8} s_{10\iota+8} b_{10\iota+8} x_{\iota,j} + \varepsilon_{8,j})) \tag{123}$$

$$x_{9,j} := \text{ReLU}(c_{9,j}(v_9 + \sum_{\iota=1}^{9} s_{10\iota+9} b_{10\iota+9} x_{\iota,j} + \varepsilon_{9,j})) \tag{124}$$

$$y_j := \text{ReLU}(\sum_{i \in K} s_i a_i x_{i,j} + \varepsilon_{y,j}). \tag{125}$$

Downstream analyses for the models generated with ESM9 were performed in the same way with the other simulators.

4.8.8 Benchmarking C+LOO and PB designs using PBSI-guided simulators

In SI I.3, we benchmarked the C+LOO and PB designs using newly introduced simulators, Models Π, Δ, and Σ. The network configurations for these models within the ESM9 framework were explicitly engineered based on PBSI logic to verify whether the relative performance of the two designs could be predicted and controlled by the PBSI score.

Model Π was constructed to represent a high-PBSI ("PB-suited") architecture characterized by pure parallelism. In this model, all direct pathways were enabled while all inter-factor regulations were disabled. Formally, the edge weights were assigned as $s_k = 1$ for pathway edges $k \in K$ and $s_\ell = 0$ for regulatory edges $\ell \in L$, ensuring distinct, independent signal transmission for each factor.

In contrast, Model Δ was constructed to represent a low-PBSI ("C+LOO-suited") architecture driven by signal amplification. The edge weights were assigned as follows:

$$s_\lambda = \begin{cases} 0 & \text{if } \lambda < 9 \\ 1 & \text{otherwise} \end{cases}, \tag{126}$$

where index $\lambda \in K \cup L$ represents the ordered edges in the ESM9 topology. This configuration results in a structure with a single positive pathway at the most downstream level, coupled with the full set of positive inter-factor regulations permitted in ESM9. This topology forces all upstream signals to merge and amplify through a dense regulatory network before converging on a single output node, thereby minimizing $\mathrm{P}_{(+)}\%$ and ensuring large values of $\mathrm{MaxRW}_{(+)}/\mathrm{R}^*$ and $\mathrm{RW}_{(+)}\%$.

Model Σ serves as a critical control: a system that is topologically entangled (like Model Δ) but yields a high PBSI score (like Model Π). Modifying the Model Δ architecture, we assigned edges as follows:

$$s_\lambda = \begin{cases} 0 & \text{if } \lambda \in \{5,6,7,8\} \\ -1 & \text{if } \lambda = 9 \\ 1 & \text{otherwise} \end{cases}. \tag{127}$$

This configuration introduces two key alterations. First, the terminal pathway edge ($\lambda = 9$) is set to -1, inverting the signal amplification generated by the positive regulations. Second, additional positive pathways (indices $\lambda \in \{1,2,3,4\}$) are enabled. These extra parallel paths elevate the numerator term $\mathrm{P}_{(+)}\%$, while the low covariance maintained by the dampened interactions ensures that the overall PBSI score remains high despite the presence of complex crosstalk.

Using the same benchmarking protocols applied to Models Φ, Ψ, and Λ, we evaluated the performance of the C+LOO and PB designs with these simulators.

4.9 scRNA-seq data analysis

To validate the biological interpretation of PBSI—specifically, that parallel gene modules exhibit high PBSI while signal amplification networks exhibit low PBSI—we analyzed the Perturb-seq dataset of K562 cells published by Replogle et al. (2022) [18]. The raw single-cell data (`K562_essential_raw_singlecell_01.h5ad`) were downloaded, and only the non-targeting (control) condition was utilized for this study. Our analysis aimed to construct GRNs corresponding to various biological phenomena and examine the relationship between their network topologies and PBSI scores. Implementation details are available in the GitHub repository described in Data and Code Availability 5.

4.9.1 Data preprocessing

Data preprocessing was performed using `Scanpy` [51]. Quality control metrics, including total counts, number of genes by counts, and the percentage of mitochondrial genes, were visualized to confirm that the deposited data had been appropriately preprocessed. Doublet removal was executed using the `Scanpy` interface for `Scrublet` [52]. Subsequently, total count normalization (scaling to 10,000 counts per cell), log-transformation, highly variable gene selection, Principal Component Analysis, construction of the nearest neighbor graph, and dimensionality reduction via Uniform Manifold Approximation and Projection (UMAP) [53] were performed using standard `Scanpy` workflows. To facilitate robust variable selection and causal discovery, we aggregated the cells into MetaCells using the `SEACells` algorithm [54], resulting in the assignment of 213 MetaCells. These MetaCells were used as pseudobulk samples for the subsequent GRN inference.

4.9.2 Phenotype scoring

For the S-phase score and other phenotype scores, gene sets were retrieved from the MSigDB Hallmark 2020 collection [55] via `GSEApy` [56]. Genes included in the dataset were extracted from the sub-categories, accounting for gene symbol aliases using `MyGene` [57]. Scoring was performed using the `score_genes` function implemented in `Scanpy`. The raw scores were standardized to Z-scores and subsequently transformed using a sigmoid function to rescale the values to the range $[0, 1]$. Specifically for the S-phase score, we calculated the score using genes from the "E2F Targets" category, explicitly excluding those that overlap with the "G2-M Checkpoint" category to ensure phase specificity.

4.9.3 Feature selection for GRNs

We determined the target configuration for each GRN—either parallel or entangled—based on the visualization of Pearson correlation heatmaps for gene sets within

the sub-categories of the MSigDB Hallmark 2020 collection. Theoretically, GRN inference aims to construct graphs representing causal regulatory structures. However, given that statistical dependencies and correlation coefficients are often employed as surrogate measures in the context of causal discovery [58], we selected gene sets with low mutual correlations for parallel GRNs and those with high correlations for entangled GRNs.

To identify gene sets for parallel GRNs, we selected the top 20 genes with low mutual correlation by performing Recursive Feature Elimination using LASSO regression as the estimator, with the phenotypic score of each sub-category as the target variable. Conversely, for entangled GRNs, we selected the top 20 genes exhibiting the highest individual correlation with the phenotypic score of the respective sub-category. All implementations were conducted using standard Python packages, including `scikit-learn` and `Pandas`.

4.9.4 Bayesian network inference and PBSI evaluation

To simulate a scenario where a phenotypic score serves as the primary readout—specifically, the impact of individual gene knockdowns on the S-phase score—we performed structural inference of GRNs using the 20 candidate genes selected for each sub-category and the S-phase score. We employed a Bayesian network-based causal discovery framework, which learns caulaisty from data structure, rather than specific biological domain knowledge (e.g., transcription factor binding), allowing for flexible feature selection and the inclusion of non-genetic variables such as phenotypic scores.

The single-cell data were standardized using the `StandardScaler` from `scikit-learn` and subsequently averaged per MetaCell. Using these aggregated data, we estimated the DAG structures by maximizing the Bayesian Information Criterion through a Hill-Climb search implemented in `Pgmpy` [59]. To ensure the S-phase score was positioned as the terminal node, we identified all genes upstream of the S-phase score and extracted the corresponding induced subgraphs using `NetworkX` [60]. The regulatory signs (upregulation or downregulation) for each edge were assigned based on the signs of the Pearson correlation coefficients between each gene and the S-phase score. This process resulted in a signed DAG where the output variable is located at the lowest downstream level, consistent with our simulation framework.

Finally, we evaluated $P_{(+)}\%$, $1 - \mathrm{MaxRW}_{(+)}/R^*$, $1 - \mathrm{RW}_{(+)}\%$, and PBSI for the inferred GRNs. Since the number of factors in these GRNs varied depending on the identified upstream components, we implemented a dynamic programming-based algorithm using `NumPy` to efficiently compute these topology metrics for varying graph sizes.

4.10 AI usage declaration

In this work, we used large language models, including ChatGPT and Gemini, for code review and language proof editing. All code and text were initially drafted by YO. Following proofreading by the AI models aimed at bug detection and readability improvement, all suggestions were scrutinized, verified, and modified if needed by YO. The manuscript drafted by YO was further reviewed by all other authors.

No AI models were used to create figures, including conceptual diagrams—namely, Figure 2 and Figures S3–S4—which were created by YO using Adobe Illustrator.

5 Data and Code Availability

The scRNA-seq data analyzed in this study were obtained from the Figshare repository associated with Replogle et al. (2022) [18] (https://plus.figshare.com/articles/dataset/_Mapping_information-rich_genotype-phenotype_landscapes_with_genome-scale_Perturb-seq_Replogle_et_al_2022_processed_Perturb-seq_datasets/20029387). All analysis code is available in our GitHub repository (https://github.com/yo-aka-gene/WhyDOE).

6 Author information

YO conceived the method and performed the experiments with input from TI and KS. YO assessed the statistical validity of the method under supervision from YS. YO developed biological interpretability of the method with input from YS, HO, and KS. YO wrote the manuscript with input from TI, YS, HO, and KS. All the authors reviewed the manuscript.

7 Acknowledgements

TI has received a grant from Japan Society for the Promotion of Science (JSPS) KAKENHI Grant Number JP25K21344.

We would like to express our sincere gratitude to Drs. Hikaru Sugimoto (The University of Tokyo), Herbert Yao (University of British Columbia), Profs. Shinya Kuroda (The University of Tokyo), and Nozomu Yachie (University of British Columbia) for their critical reading and their valuable suggestions.

Abbreviations

ANOVA analysis of variance

AP average precision

AUROC area under the ROC

C+LOO Control-and-Leave-One-Out

CI confidence interval

DAG directed acyclic graph

DO-C+LOO D-optimized C+LOO

DO-PB D-optimized PB

DOE design of experiments

ESM4 Exhaustive Search Model of four factors

ESM9 Exhaustive Search Model of nine factors

FF full factorial

GBDT gradient boosting decision tree

GRN gene regulatory network

KD knockdown

ML machine learning

MLR multivariate linear regression

OFAT One-Factor-At-a-Time

PB Plackett–Burman

PBSI PB suitability index

ReLU rectified linear unit

ROC receiver operating characteristic

scRNA-seq single-cell RNA-sequencing

SHAP Shapley additive explanations

SI Supplementary Information

TPL Taylor's power law

UMAP Uniform Manifold Approximation and Projection

VIF variance inflation factor

References

[1] Pearl, J. *Causality* (Cambridge university press, 2009).

[2] Okano, Y., Okano, H. & Kase, Y. Spinal cord injury regenerative therapy development: integration of design of experiments. *Neural Regeneration Research* **20**, 2571–2573 (2025).

[3] Montgomery, D. C. *Design and analysis of experiments* (John wiley & sons, 2017).

[4] Plackett, R. L. & Burman, J. P. The design of optimum multifactorial experiments. *Biometrika* **33**, 305–325 (1946).

[5] Moon, S., Saboe, A. & Smanski, M. J. Using design of experiments to guide genetic optimization of engineered metabolic pathways. *Journal of Industrial Microbiology and Biotechnology* **51**, kuae010 (2024).

[6] Sakurada, K. *et al.* Medical AI and AI for medical sciences. *JMA journal* **8**, 26–37 (2025).

[7] Dixit, A. *et al.* Perturb-Seq: dissecting molecular circuits with scalable single-cell RNA profiling of pooled genetic screens. *Cell* **167**, 1853–1866 (2016).

[8] Adamson, B. *et al.* A multiplexed single-cell CRISPR screening platform enables systematic dissection of the unfolded protein response. *Cell* **167**, 1867–1882 (2016).

[9] Dunnett, C. W. A multiple comparison procedure for comparing several treatments with a control. *Journal of the American Statistical Association* **50**, 1096–1121 (1955).

[10] Aibar, S. *et al.* SCENIC: single-cell regulatory network inference and clustering. *Nature methods* **14**, 1083–1086 (2017).

[11] Taylor, L. Aggregation, mean and variance. *Nature* **189**, 732–735 (1961).

[12] Giometto, A., Formentin, M., Rinaldo, A., Cohen, J. E. & Maritan, A. Sample and population exponents of generalized Taylor's law. *Proceedings of the National Academy of Sciences* **112**, 7755–7760 (2015).

[13] Lindquist, W. B. & Rachev, S. T. Taylor's law and heavy-tailed distributions. *Proceedings of the National Academy of Sciences* **118**, e2118893118 (2021).

[14] Grün, D., Kester, L. & Van Oudenaarden, A. Validation of noise models for single-cell transcriptomics. *Nature methods* **11**, 637–640 (2014).

[15] Awazu, A., Tanabe, T., Kamitani, M., Tezuka, A. & Nagano, A. J. Broad distribution spectrum from Gaussian to power law appears in stochastic variations in RNA-seq data. *Scientific Reports* **8**, 8339 (2018).

[16] de Aguiar, P. F., Bourguignon, B., Khots, M., Massart, D. & Phan-Than-Luu, R. D-optimal designs. *Chemometrics and intelligent laboratory systems* **30**, 199–210 (1995).

[17] Gardner, T. S. & Faith, J. J. Reverse-engineering transcription control networks. *Physics of life reviews* **2**, 65–88 (2005).

[18] Replogle, J. M. *et al.* Mapping information-rich genotype-phenotype landscapes with genome-scale Perturb-seq. *Cell* **185**, 2559–2575 (2022).

[19] Cui, H. *et al.* scGPT: toward building a foundation model for single-cell multi-omics using generative AI. *Nature methods* **21**, 1470–1480 (2024).

[20] Viñas Torné, R. *et al.* Systema: a framework for evaluating genetic perturbation response prediction beyond systematic variation. *Nature Biotechnology* 1–10 (2025).

[21] Karniadakis, G. E. *et al.* Physics-informed machine learning. *Nature Reviews Physics* **3**, 422–440 (2021).

[22] Terayama, K., Sumita, M., Tamura, R. & Tsuda, K. Black-box optimization for automated discovery. *Accounts of Chemical Research* **54**, 1334–1346 (2021).

[23] Takahashi, K. & Yamanaka, S. Induction of pluripotent stem cells from mouse embryonic and adult fibroblast cultures by defined factors. *Cell* **126**, 663–676 (2006).

[24] Vanaja, K. & Shobha Rani, R. Design of experiments: concept and applications of Plackett Burman design. *Clinical research and regulatory affairs* **24**, 1–23 (2007).

[25] Goupy, J. What kind of experimental design for finding and checking robustness of analytical methods? *Analytica chimica acta* **544**, 184–190 (2005).

[26] Wang, J. & Wan, W. Experimental design methods for fermentative hydrogen production: a review. *International journal of hydrogen energy* **34**, 235–244 (2009).

[27] Cleveland, J. L., Smith, J. A. & Collins, J. P. Factor effects in numerical simulations. *Journal of the Atmospheric Sciences* **77**, 2439–2451 (2020).

[28] Harris, C. R. *et al.* Array programming with NumPy. *Nature* **585**, 357–362 (2020).

[29] McKinney, W. *et al.* Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, vol. 445, 51–56 (Austin, TX, 2010).

[30] Lee, A. pyDOE: The experimental design package for Python. https://github.com/tisimst/pyDOE (2015). Version 0.3.8.

[31] Seabold, S. & Perktold, J. Statsmodels: econometric and statistical modeling with python. *SciPy* **7** (2010).

[32] Nair, V. & Hinton, G. E. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, 807–814 (2010).

[33] Ramachandran, P., Zoph, B. & Le, Q. V. Searching for activation functions. *arXiv preprint arXiv:1710.05941* (2017).

[34] Jarrett, K., Kavukcuoglu, K., Ranzato, M. & LeCun, Y. What is the best multi-stage architecture for object recognition? In *2009 IEEE 12th international conference on computer vision*, 2146–2153 (IEEE, 2009).

[35] Nwankpa, C., Ijomah, W., Gachagan, A. & Marshall, S. Activation functions: Comparison of trends in practice and research for deep learning. *arXiv preprint arXiv:1811.03378* (2018).

[36] Kim, M. & Kim, E. Mathematical model of the cell signaling pathway based on the extended Boolean network model with a stochastic process. *BMC bioinformatics* **23**, 515 (2022).

[37] Antony, J. *Design of experiments for engineers and scientists* (Elsevier, 2023).

[38] Pedregosa, F. *et al.* Scikit-learn: Machine learning in Python. *the Journal of machine Learning research* **12**, 2825–2830 (2011).

[39] Virtanen, P. *et al.* SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature methods* **17**, 261–272 (2020).

[40] Genz, A. & Bretz, F. *Computation of Multivariate Normal and t Probabilities.* Lecture Notes in Statistics (Springer-Verlag, Heidelberg, 2009).

[41] Langsrud, Ø. ANOVA for unbalanced data: Use Type II instead of Type III sums of squares. *Statistics and computing* **13**, 163–167 (2003).

[42] Muller, K. E. & Peterson, B. L. Practical methods for computing power in testing the multivariate general linear hypothesis. *Computational Statistics & Data Analysis* **2**, 143–158 (1984).

[43] Cohen, J. Eta-squared and partial eta-squared in fixed factor ANOVA designs. *Educational and psychological measurement* **33**, 107–112 (1973).

[44] Wilk, M. B. & Gnanadesikan, R. Probability plotting methods for the analysis of data. *Biometrika* **55**, 1–17 (1968).

[45] Daniel, C. Use of half-normal plots in interpreting factorial two-level experiments. *Technometrics* **1**, 311–341 (1959).

[46] McHugh, M. L. Interrater reliability: the kappa statistic. *Biochemia medica* **22**, 276–282 (2012).

[47] Wheeler, B. *AlgDesign: Algorithmic Experimental Design* (2024). URL https://CRAN.R-project.org/package=AlgDesign. R package version 1.2.1.1.

[48] Ke, G. *et al.* Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems* **30** (2017).

[49] Akiba, T., Sano, S., Yanase, T., Ohta, T. & Koyama, M. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2623–2631 (2019).

[50] Lundberg, S. M. *et al.* From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence* **2**, 2522–5839 (2020).

[51] Wolf, F. A., Angerer, P. & Theis, F. J. SCANPY: large-scale single-cell gene expression data analysis. *Genome biology* **19**, 15 (2018).

[52] Wolock, S. L., Lopez, R. & Klein, A. M. Scrublet: computational identification of cell doublets in single-cell transcriptomic data. *Cell systems* **8**, 281–291 (2019).

[53] McInnes, L., Healy, J., Saul, N. & Großberger, L. UMAP: Uniform Manifold Approximation and Projection. *Journal of Open Source Software* **3**, 861 (2018). URL https://doi.org/10.21105/joss.00861.

[54] Persad, S. *et al.* SEACells infers transcriptional and epigenomic cellular states from single-cell genomics data. *Nature biotechnology* **41**, 1746–1757 (2023).

[55] Liberzon, A. *et al.* The molecular signatures database hallmark gene set collection. *Cell systems* **1**, 417–425 (2015).

[56] Fang, Z., Liu, X. & Peltz, G. GSEApy: a comprehensive package for performing gene set enrichment analysis in Python. *Bioinformatics* **39**, btac757 (2023).

[57] Wu, C., Mark, A. & Su, A. I. MyGene. info: gene annotation query as a service. *bioRxiv* 009332 (2014).

[58] Okano, Y., Kase, Y. & Okano, H. A set-theoretic definition of cell types with an algebraic structure on gene regulatory networks and application in annotation of RNA-seq data. *Stem cell reports* **18**, 113–130 (2023).

[59] Ankan, A. & Panda, A. pgmpy: Probabilistic Graphical Models using Python. In *SciPy*, 6–11 (2015).

[60] Hagberg, A. A., Schult, D. A. & Swart, P. J. Exploring Network Structure, Dynamics, and Function using NetworkX. In *Proceedings of the 7th Python in Science Conference*, SciPy, 11–15 (SciPy, 2008). URL http://dx.doi.org/10.25080/TCWV9851.