CAUSAL INTEGRATED GRADIENTS: SCALABLE FEATURE ATTRIBUTION ALONG CAUSAL PATHS

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing feature attribution methods that account for feature dependencies do not scale to high dimensions. We propose causal integrated gradients (CIG) and causal expected gradients (CEG), which extend integrated gradients to incorporate prior causal structure among input features while retaining its computational efficiency and, under conditions on the causal graph, its axiomatic guarantees. In many settings, some causal structure between features is already known—for example, that a transcription factor causally regulates a target gene's expression. Yet existing gradient-based methods cannot exploit this structure, instead crediting a feature's downstream effects over its true upstream cause, such as ranking the target gene above the transcription factor that regulates it. Our key idea is to embed this causal structure by routing the integration path along the causal graph. The method thereby decomposes each feature's attribution into a direct effect and effects mediated through its causal descendants. We evaluate CIG in three settings of increasing realism. First, on a synthetic benchmark with a known causal model, CIG's attributions best match the ground-truth causal effect from Pearl's do-calculus, matching the accuracy of causal Shapley values at a fraction of the computational cost. Second, on simulated single-cell gene expression data with a known gene regulatory network (Dyngen), CIG most accurately recovers the transcription factors driving the simulation. Third, on real interferon-β stimulation data, where only an approximate causal graph is available, CIG better prioritizes the transcription factors driving the measured response than DeepSHAP and integrated gradients. Together, these results show that incorporating causal structure into gradient-based attributions recovers upstream drivers that existing methods miss, without sacrificing scalability.

1 INTRODUCTION

As machine learning models grow increasingly complex, particularly deep neural networks trained on large-scale datasets, the need for interpretable explanations of their predictions becomes critical. Local post-hoc explanation methods have emerged as a popular approach for understanding the contribution of individual input features to model predictions. However, most existing methods assume that input features are independent, which is often unrealistic in practice.

Feature dependencies are common in real-world applications—in gene expression, for instance, genes exhibit complex inter-dependencies reflecting underlying biological processes. Ignoring them can produce misleading attributions that fail to uncover the true data-generating mechanism (Aas et al., 2021; Chen et al., 2020), with significant consequences for biological discovery.

Some approaches have been proposed to incorporate feature dependencies into attribution methods. Methods based on Shapley values, such as conditional SHAP (Aas et al., 2021) and causal Shapley values (Heskes et al., 2020), can account for dependencies but rely on sampling-based computations that become prohibitively expensive in high-dimensional settings. Conversely, gradient-based methods like integrated gradients (IG) (Sundararajan et al., 2017) and expected gradients (Erion et al., 2021) offer computational efficiency but fundamentally assume feature independence.

To address this gap, we propose causal integrated gradients (CIG) and causal expected gradients (CEG), principled extensions of integrated gradients that incorporate causal structure among input features. Like causal Shapley values (Heskes et al., 2020), our methods require the structure of a

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

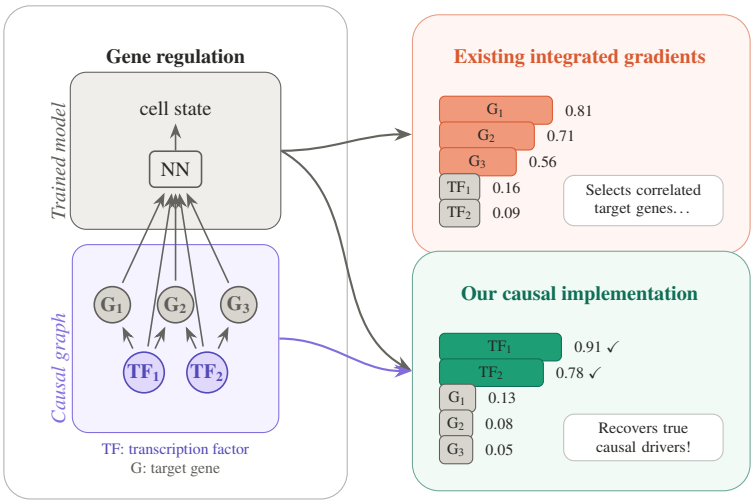


Figure 1: Simulated example inspired by attribution of gene regulation. Existing graph-agnostic methods assign the highest attribution scores to the directly correlated target genes, missing the true upstream regulators; our graph-aware method correctly pushes the transcription factors to the top.

causal graph over the inputs; unlike them, we need only a deterministic, off-the-shelf surrogate rather than a full conditional distribution to sample from. Conditional SHAP needs no causal graph but does need that conditional distribution, while standard integrated and expected gradients need neither. In return for this graph, our methods keep the efficiency of integrated gradients and, under conditions on the causal graph and surrogate, its axiomatic guarantees. They scale to high-dimensional settings, recovering upstream causal drivers that graph-agnostic methods miss, as in gene regulation (Figure 1).

The main contributions of this paper are as follows:

- We propose causal integrated and expected gradients (CIG and CEG), extending integrated gradients to incorporate causal structure among input features. Both methods are applicable to any differentiable model for which causal dependencies among inputs are available.
- We evaluate the proposed methods on a synthetic benchmark with known causal effects and compare against the state-of-the-art (causal Shapley values (Heskes et al., 2020) and conditional SHAP (Aas et al., 2021)), demonstrating improved causal attribution performance at substantially lower computational cost — reducing runtimes from hours to minutes.
- We demonstrate the real-world applicability of our method on two high-dimensional single-cell datasets: identifying transcription factors driving cell differentiation in simulated data, and prioritizing transcription factors that are most sensitive to interferon-β stimulation. Across both tasks, our method outperforms existing approaches in recovering causal transcription factors, while scaling efficiently to high-dimensional RNA expression data.

2 Background and Related Work

A natural framework for gradient-based feature attribution is that of *path methods*: let $F : \mathbb{R}^n \to \mathbb{R}$ be a differentiable model and $\gamma : [0, 1] \to \mathbb{R}^n$ a differentiable path with $\gamma(0) = b$ (a baseline) and $\gamma(1) = \boldsymbol{x}$ (the input to explain). By the gradient theorem, $\int_0^1 \nabla F(\gamma(\alpha)) \cdot \frac{d\gamma(\alpha)}{d\alpha}\, d\alpha = F(\boldsymbol{x}) - F(b)$; expanding the dot product gives the feature attribution R_i, interpreted as feature $\boldsymbol{x}_i$'s effect on the model output:

$$\sum_{i=1}^{n} \int_0^1 \frac{\partial F(\gamma(\alpha))}{\partial \gamma_i(\alpha)} \frac{d\gamma_i(\alpha)}{d\alpha}\, d\alpha = \sum_{i=1}^{n} R_i = F(\boldsymbol{x}) - F(b). \tag{1}$$

Equation 1 guarantees the *completeness* property: attributions sum exactly to the difference in model output between the input and the baseline. Integrated gradients (Sundararajan et al., 2017) and expected gradients (Erion et al., 2021) are both instances of this framework: integrated gradients uses a straight-line path $\gamma(\alpha) = b + \alpha(\boldsymbol{x} - b)$ from a fixed baseline, approximated in practice via a

K-step Riemann sum; expected gradients instead averages over a distribution of baselines to remove the sensitivity to this single fixed choice, sampling the baseline and path parameter $\alpha \sim \mathcal{U}(0, 1)$ jointly to form a Monte Carlo estimator. Both methods, however, implicitly assume that features are mutually independent along the path.

The incorporation of feature dependencies in local attribution methods has instead been most thoroughly explored in the context of Shapley values. Conditional SHAP (Aas et al., 2021) and causal Shapley values (Heskes et al., 2020) are prominent examples. Conditional SHAP accounts for statistical dependencies by replacing the marginal feature distribution with the conditional distribution when computing Shapley values, while causal Shapley values incorporate an explicit causal graph to distinguish direct effects from indirect effects mediated through other features. Another extension of Shapley values is the Owen value, which takes into account that features belong to groups (Giménez & Puente, 2019). The partition explainer in the `shap` Python package (Lundberg & Lee, 2017) computes Owen values by reducing the number of coalitions on which the marginal contributions are computed, based on an initial partitioning of the features. While incorporating dependencies is contested (Janzing et al., 2020; Aas et al., 2021), it is preferable when the goal is to understand the true data-generating process (Chen et al., 2020; Michiels et al., 2024) — a consideration especially consequential in biological discovery. To our knowledge, no prior work has extended this line of thinking to gradient-based attribution methods, which is the primary contribution of this paper.

Numerous modifications to the integrated gradients algorithm have been proposed, though none address feature dependencies. Most target either attribution noise (Kapishnikov et al., 2021), the reliance on an arbitrary baseline (Erion et al., 2021), or both (Yang et al., 2023; Zhuo & Ge, 2024). Two works are particularly relevant to ours, as they also modify the integration path: Kapishnikov et al. (2021) alters it to avoid regions of high partial derivatives, hypothesized to introduce noise, while Zhuo & Ge (2024) defines a path that follows the model's gradient. Our work differs fundamentally from both — rather than modifying the path for numerical reasons, we do so to encode causal structure among input features.

Feature attribution methods have been widely used to interpret deep learning models trained on RNA expression data and extract biologically meaningful insights (Yap et al., 2021; Janizek et al., 2023; Tang et al., 2023). Yap et al. (2021), for instance, applied expected gradients to models trained on the GTEx RNA-seq dataset (47 tissue types) to recover genes associated with tissue identity and function, while Janizek et al. (2023) used explainability methods to assess whether unsupervised models capture meaningful gene relationships and pathways. However, most existing attribution methods ignore strong gene dependencies, potentially producing misleading attributions that fail to reflect the underlying biological structure, and their application to high-dimensional RNA expression data remains largely unexplored due to computational complexity.

3 Causal integrated and expected gradients

As shown in the background section, all path methods satisfy the same completeness property and differ only in their choice of path. This freedom in the path has been exploited before (Kapishnikov et al., 2021; Zhuo & Ge, 2024), but always for numerical reasons. We instead exploit it to encode causal structure by reparametrizing the path to reflect dependencies between input features. Concretely, this requires three steps:

1. Reparametrize the path function to incorporate dependencies between features, so that direct and indirect effects can be separated via the chain rule.
2. Determine the gradients of the path function that appear in the resulting attribution formula.
3. Model the feature dependencies needed to evaluate those gradients.

3.1 Incorporating dependencies in the path function

In standard path methods, each path variable γ_i depends only on α, meaning attributions reflect only the direct effect of each feature on the model output. To incorporate indirect effects, we allow γ_i to depend on other path variables γ_j. This modification is best understood through the chain rule diagram, which for standard integrated gradients with n features takes the form of the **black** edges in the left panel of Figure 2, with $x_i = \gamma_i(\alpha)$.

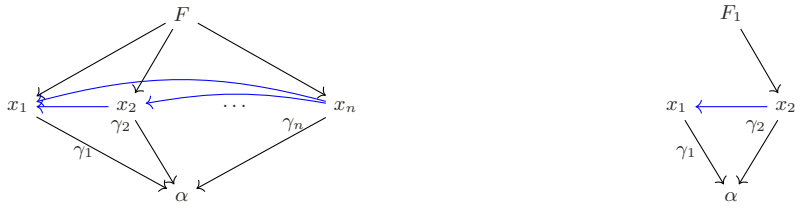


Figure 2: Chain rule diagrams illustrating the reparametrization of path functions to incorporate causal dependencies. *Left:* general n-feature case. *Right:* two-feature example. Black edges are standard path method dependencies; blue edges are the causal dependencies introduced by our reparametrization.

Note that since the chain rule diagram is a directed acyclic graph (DAG), any dependency we add between input features must be directed — determining which feature causes the other — and must not introduce a cycle. We continue with the most general case of a fully connected DAG, with features ordered topologically: if x_u depends on x_v, then $u > v$. Such dependencies are represented as blue edges in the left panel of Figure 2, and under this ordering the individual path functions become $x_i = \gamma_i(x_1, \ldots, x_{i-1}, \alpha)$: each feature's path depends on all features that causally precede it, plus α.

With the modified path variables, computing the attributions in Equation 1 requires an additional application of the chain rule. We illustrate this with a simple example: a function F_1 that depends only on x_2, where x_2 itself depends on x_1, as shown in the right panel of Figure 2. Expanding the left-hand side of Equation 1 for this example gives:

$$\int_0^1 \frac{\partial F_1}{\partial \gamma_2} \frac{d\gamma_2}{d\alpha}\, d\alpha = \int_0^1 \frac{\partial F_1}{\partial \gamma_2} \frac{\partial \gamma_2}{\partial \alpha}\, d\alpha + \int_0^1 \frac{\partial F_1}{\partial \gamma_2} \frac{\partial \gamma_2}{\partial \gamma_1} \frac{d\gamma_1}{d\alpha}\, d\alpha, \tag{2}$$

where the first term is the direct attribution of x_2 and the second is the indirect attribution of x_1 through x_2. Generalizing to a fully connected DAG with n features, the attribution of feature i is:

$$R_i = \int_0^1 \frac{\partial F}{\partial \gamma_i} \frac{\partial \gamma_i}{\partial \alpha}\, d\alpha + \sum_{j>i} \int_0^1 \frac{\partial F}{\partial \gamma_j} \frac{\partial \gamma_j}{\partial \gamma_i} \frac{d\gamma_i}{d\alpha}\, d\alpha = R_{i,\text{direct}} + R_{i,\text{indirect}}, \tag{3}$$

where the first term captures the direct effect of x_i and the second captures its indirect effect through all features that depend on it. This formulation applies to any DAG that represents model and input dependencies: missing edges simply correspond to zero terms ($\frac{\partial \gamma_j}{\partial \gamma_i} = 0$ if x_j does not depend on x_i). Additionally, any DAG can be reordered to match the topological ordering (see Appendix A for why this choice does not affect attributions). One subtlety is that the indirect terms use the full derivative $\frac{d\gamma_i}{d\alpha}$, meaning part of the indirect effect of x_i may itself pass through features that x_i depends on. In other words, this formulation attributes only indirect effects that are one hop long (also referred to as first-order). We provide an extension to multi-hop indirect effects in Appendix A. Finally, to be able to compute the above integrals, we need to find a practical way to compute the gradients of the path function, which we will discuss in the next section.

3.2 Determining the gradients of the path function

To evaluate Equation 3 via a Riemann sum over K steps, we require the following extra gradients at each point $\alpha^{(p)} = p/K$, $p = 1, \ldots, K$, along the path:

$$\frac{\partial \gamma_i}{\partial \alpha}(\alpha^{(p)}), \quad \frac{\partial \gamma_j}{\partial \gamma_i}(\gamma_i^{(p)}), \tag{4}$$

with $\gamma_i^{(p)} = b_i + \alpha^{(p)}(x_i - b_i)$, i.e. a straight-line path as in regular integrated gradients. To compute these, we need an explicit formulation of the path functions γ_i.

To this end, we interpret the model and the feature dependencies as a structural causal model (SCM) (as previously in Janzing et al., 2020; Heskes et al., 2020). If the arrow directions in the chain rule diagram of the left panel of Figure 2 are switched, the graph effectively represents a causal graph,

and the path functions can represent the structural mechanism between parent and child nodes of the graph. We will interpret α as residual information, which explains the part of the variation in the children that is not explained by their parents. As is common in causal inference, we assume this residual enters additively. Concretely, for a fully connected DAG with features ordered topologically, we model each feature as:

$$x_i = \gamma_i(x_1, \ldots, x_{i-1}, \alpha) = c_i(x_1, \ldots, x_{i-1}) + h_i(\alpha), \quad i = 1, \ldots, n, \tag{5}$$

where c_i is the causal mechanism mapping the parents of x_i to its expected value, and $h_i(\alpha)$ is an additive residual term. We approximate the causal mechanism c_i with a surrogate model s_i trained to predict x_i from its causal parents, so that the residual $h_i^{(p)} \approx x_i^{(p)} - s_i(x_1^{(p)}, \ldots, x_{i-1}^{(p)})$ captures the variation in x_i not directly explained by its parents. The required gradients are then:

$$\frac{\partial \gamma_j}{\partial \gamma_i}(\gamma_i^{(p)}) = \frac{\partial s_j}{\partial x_i}(x_i^{(p)}), \qquad \frac{\partial \gamma_i}{\partial \alpha}(\alpha^{(p)}) \approx \frac{h_i^{(p)} - h_i^{(p-1)}}{\Delta \alpha}, \tag{6}$$

where the last one is approximated via a backward finite difference, consistent with related work (Zhuo & Ge, 2024). The quality of these gradients depends on the choice of surrogate model s_i, which we discuss in the next section.

3.3 Modeling feature dependencies through a surrogate model

Any differentiable surrogate model can be employed to model the feature dependencies under a known causal structure. Here, we introduce a surrogate model that is flexible enough to model complex dependencies while being efficient and scalable to high-dimensional datasets. We propose a sparse masked surrogate s that parameterizes parent-to-child relationships as a sum of edge-wise contributions, operating on a directed graph defined by an adjacency matrix $M \in \mathbb{R}^{n \times n}$, where $M_{ij} = 1$ indicates that feature i is a parent of feature j.

Each parent feature x_i is first transformed using a per-feature basis expansion $\phi(x_i) \in \mathbb{R}^Q$ of size Q, to give the surrogate model more flexibility in learning nonlinear effects. The basis consists of three components applied to standardized inputs: 1) polynomial terms, 2) truncated power functions with learnable knots, and 3) Fourier features (see Appendix B for a detailed explanation of the basis expansion). For every active edge $(i \to j)$ such that $M_{ij} = 1$, the prediction for child node j is obtained by summing contributions from all parent nodes:

$$s_j = \hat{x}_j = \sum_{i: M_{ij}=1} \langle w_{ij}, \phi(x_i) \rangle + \mu_j, \tag{7}$$

where μ_j and $w_{ij} \in \mathbb{R}^Q$ are learnable per-node intercept and edge-specific coefficients, respectively.

Rather than materializing all possible n^2 parent-child interactions, the model stores only active graph edges. This sparse formulation reduces both memory usage and computation, making the model scalable to large sparse graphs. The surrogate is trained to reconstruct the set of predictable child nodes from their parent nodes by minimizing the mean squared reconstruction error, with early stopping on validation loss (full hyperparameters in Tables G.3 and H.1). The additive edge structure also makes the surrogate Jacobian needed for attribution analytical and sparse: for a child node j and parent feature x_i, the derivative is

$$\frac{\partial s_j}{\partial x_i} = \sum_{q=1}^{Q} \frac{\partial \phi_q(x_i)}{\partial x_i} w_{ij,q} M_{ij}, \tag{8}$$

which is nonzero only for active edges.

3.4 Extension to causal expected gradients

The above method can be extended to multiple baselines in the same way as expected gradients extends integrated gradients: sampling α and the baseline b jointly still gives an unbiased single-sample estimate for either method. Standard expected gradients performs well with just one path point per baseline, since the straight-line path has a constant, closed-form derivative $d\gamma_i/d\alpha = x_i - b_i$, so a single sampled α already gives this derivative exactly. Our method's path derivative $\partial \gamma_i / \partial \alpha$ is

instead estimated via a backward finite difference (Equation 6), since $h_i(\alpha)$ is generally nonlinear; our method therefore benefits from allocating more of a fixed evaluation budget to additional path points per baseline rather than to additional baselines, since both the finite-difference and Riemann-sum errors (Appendix C) compound when paths are under-sampled.

3.5 Axiomatic properties

Since our method belongs to the family of path methods (Sundararajan et al., 2017), it inherits completeness and sensitivity(a) (for the direct term) unconditionally, and implementation invariance, symmetry, and linearity under precise preconditions on the causal graph and surrogate that we formalize in Appendix C. The one exception is the dummy axiom: a feature absent from F but causally upstream of a feature that does appear can receive non-zero indirect attribution, as shown in the example in the right of Figure 2.

4 Experiments

We evaluate our method in three settings of increasing realism: a synthetic benchmark with known ground-truth causal effects, and two biological datasets where understanding causal relationships is crucial for scientific discovery. Experimental setup and costs are reported in Appendix E.

4.1 Comparison with causal attributions

Conditional SHAP (Aas et al., 2021) and causal Shapley values (Heskes et al., 2020) are the most established approaches for dependency-aware feature attribution. To evaluate how well each method captures ground-truth causal effects, we compare their attributions against a reference causal effect derived from Pearl's do-calculus (Pearl, 2009). Among the causal effects this framework gives rise to, the Conditional Average Treatment Effect (CATE) is the most appropriate for local explanations, as it measures the effect of intervening on a single feature while conditioning on the root nodes $\mathcal{R}$:

$$\text{CATE}_i = \mathbb{E}[F \mid \text{do}(X_i = x_i), X_{\mathcal{R}\setminus\{i\}} = x_{\mathcal{R}\setminus\{i\}}] - \mathbb{E}[F \mid \text{do}(X_i = b_i), X_{\mathcal{R}\setminus\{i\}} = x_{\mathcal{R}\setminus\{i\}}]. \quad (9)$$

While several papers have analyzed the theoretical relationship between local attributions and Pearl's do-calculus (Heskes et al., 2020; Janzing et al., 2020; Ng et al., 2025), to the best of our knowledge no prior work has empirically compared attribution methods against the CATE directly. Doing so gives us an exact ground truth for each feature's local causal effect, available only in this fully controlled setting. Since the CATE is an effect rather than an attribution (e.g. it need not satisfy completeness), we compare feature rankings via Spearman correlation rather than raw values; the two experiments that follow instead use biological data, where ground truth is global (Dyngen) or approximate (IFN-β).

To enable computation of both the CATE and the local attributions, we construct a synthetic benchmark with a known SCM: the target is a random linear combination of a subset of features, and a Lasso regression model F is trained on the generated data, using the SCM's structural mechanisms directly for causal Shapley values and our own methods to isolate attribution quality from implementation noise (full details in Appendix F). Across an increasing number of features, we report the Spearman correlation between each method's attributions and the CATE, averaged over root nodes (left of Figure 3) and over all nodes, alongside runtime (right panel of Figure 3).

Results and discussion. On root nodes (left panel of Figure 3), CIG achieves near-perfect CATE correlation, with causal Shapley values trailing closely behind; conditional SHAP and integrated gradients fall considerably short, and conditional SHAP was excluded beyond moderate feature counts due to prohibitive runtime and consistently poor quality. Only causal methods shift attribution toward root nodes (see Appendix F for an intuitive example), which is needed to identify upstream drivers. Our methods achieve this at substantially lower computational cost than causal Shapley values (right panel of Figure 3), whose exact computation scales exponentially in the number of features versus our worst-case quadratic scaling (full derivation, including a sparse-graph case, in Appendix D); standard integrated gradients remains the fastest (a single batched pass) but captures none of the causal structure. CEG underperforms CIG despite using the same computational budget: given a fixed number of gradient evaluations, CEG trades path steps for additional baselines, and as

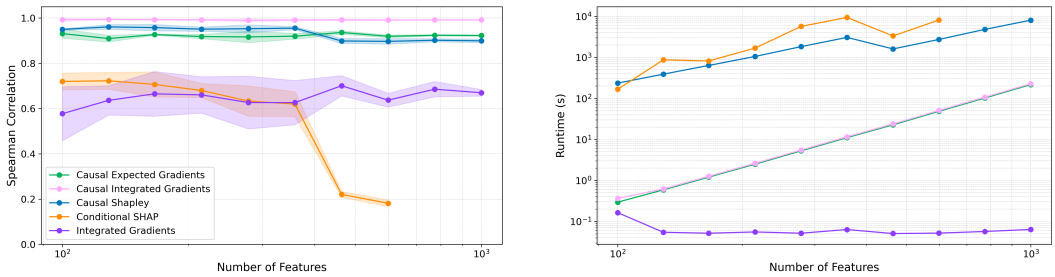


Figure 3: *Left:* Spearman correlation with the CATE on root nodes only. *Right:* wall-clock runtime (log–log scale) to explain all 100 samples; the dip for the Shapley-based methods reflects their reduced sampling budget at larger feature counts. Both as a function of the number of features. Shaded areas denote the standard deviation over 5 runs. Results on all nodes are shown in Figure F.1.

discussed in Section 3.4, path steps are critical for accurately approximating the causal path integral, so reducing them degrades attribution quality. Finally, on all nodes (Figure F.1), CIG still performs better than all other methods.

4.2 Causal gene recovery in a simulated biological dataset

We evaluate our method on a regression task using simulated single-cell gene expression data, predicting each cell's simulated differentiation time from its trajectory of activated and repressed transcription factors. The dataset is generated with Dyngen (Cannoodt et al., 2021) under a known gene regulatory network (GRN) with a backbone of transcription factors (TFs) that causally regulate downstream genes. Unlike the per-sample CATEs of the synthetic benchmark above, ground truth here is exact but global: the backbone TFs are the known drivers of the simulation, letting us test recovery at realistic single-cell scale and complexity. After filtering and normalization, a fully connected neural network is trained to predict simulated time.

The GRN defines both the feature ordering and surrogate structure for our method. We evaluate several CIG/CEG variants (linear or expanded-basis surrogate, first- or second-order terms), against attribution baselines from the `captum` and `shap` packages (Kokhlikyan et al., 2020; Lundberg & Lee, 2017), over five random seeds. Conditional SHAP and causal Shapley values were excluded due to their computational infeasibility at this scale. Performance is evaluated by backbone-TF recovery: area under the recall curve (AUC) (treating backbone TFs as positives), and discounted cumulative gain (DCG) over the top-ranked genes (relevance scores assigned according to each gene's role in the GRN). We also evaluate our method's ability to prioritize the causal regulators for each transition state. Dataset, model, surrogate, and evaluation details are given in Appendix G.

Results and discussion. Our methods achieve the strongest recovery of backbone TFs (Figure 4A). The expanded-basis with second-order terms is the best overall, and SHAP's partition explainer is the best baseline. Both second-order terms and the expanded–basis improve performance over the linear and first-order variants. Similar trends are observed for the DCG scores (Figure 4B): Our method variants are near-optimal at a threshold of 10 genes, and the expanded-basis CIG remains the best overall at 50 and 250. Similar performance is shown in recovering causal regulators per transition state (Figure J.1), with CIG beating DeepSHAP and integrated gradients in every state.

4.3 Perturbed gene discovery in the IFN-β stimulation dataset

We next evaluate our method on real single-cell RNA-sequencing data from Kang et al. (2018), in which peripheral blood mononuclear cells were stimulated with interferon-β (IFN-β) or left as controls. A fully connected neural network is trained to predict whether a cell was exposed to IFN-β stimulation. Unlike the previous two experiments, no true causal graph is available here, so ground truth is only approximate: we rely on curated prior knowledge (OmniPath) and state-of-the-art regulatory network inference methods (SCENIC) to test whether our method still identifies the transcription factors driving the interferon response. We compare our method against DeepSHAP (Lundberg & Lee, 2017) and integrated gradients (Sundararajan et al., 2017) (zero baseline). We

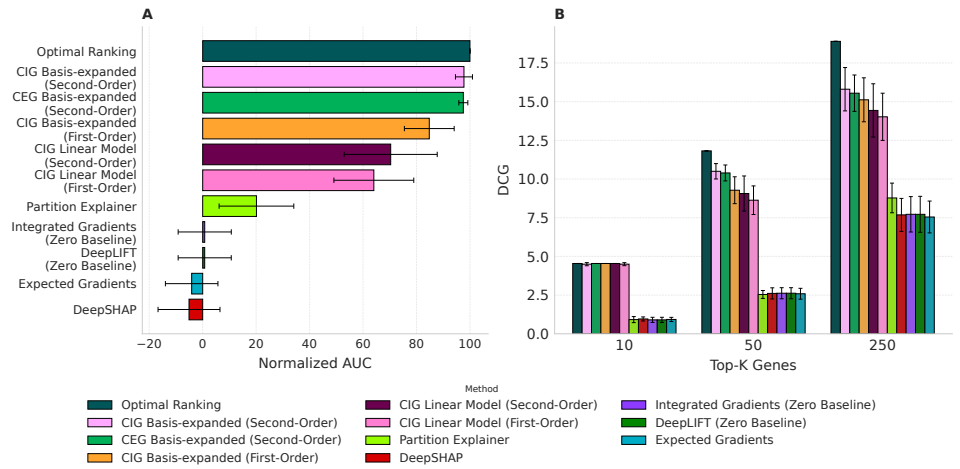


Figure 4: Evaluation of feature attribution methods on Dyngen. (A) Area under the recall curve for backbone transcription factors, normalized so that a random ranking scores 0 and the optimal ranking 100. (B) DCG over the top-K ranked genes, with the DCG of the optimal ranking shown for reference; relevance scores follow Table G.6. Error bars denote the standard deviation over 5 seeds.

derive a directed acyclic graph from OmniPath (Türei et al., 2021), used both for feature ordering and to train the surrogate. We compute attributions for all genes in held-out test cells across five seeds, summarized as the mean absolute attribution per gene, and evaluate rank by hop distance from the roots of both the OmniPath prior and a SCENIC-inferred regulatory network (Aibar et al., 2017). SCENIC leverages motif- and TF-curated references to derive gene regulatory networks from the data while OmniPath is not data-specific. Dataset, model, graph construction, surrogate, and evaluation details are given in Appendix H.

Next, we test whether the methods recover stimulation-responsive regulators: by ranking genes per cell type by their attribution difference between stimulated and control cells and testing whether SCENIC-responsive TFs rank above the rest, using rank- and top-k-enrichment tests (one-sided Mann–Whitney U test and hypergeometric test, respectively). As an orthogonal readout, we evaluate recovery of per-cell-type pseudobulk differentially expressed genes (DEGs). For the three cell types most responsive to IFN-β stimulation (Kang et al., 2018) (CD14$^+$/FCGR3A$^+$ monocytes, dendritic cells), we also run over-representation analysis (ORA) (Fang et al., 2023) on each method's unique top-100 genes against MSigDB Hallmark gene sets (Liberzon et al., 2015).

Results and discussion. The distribution of attribution ranks across hop-distance groups shows the CIG advantage is concentrated at the top of the regulatory hierarchy: on the OmniPath prior, CIG ranks root genes higher than DeepSHAP and integrated gradients, with a smaller advantage one hop downstream (Figure I.1); the effect also holds on the SCENIC graph (Figure 5A), where DeepSHAP and integrated gradients rank roots below genes one to three hops downstream. Rankings are reproducible across seeds, with CIG the most stable of the three at SCENIC roots (Tables I.1 and I.2).

Across cell types, CIG places stimulation-responsive TFs higher in its gene ranking than the remaining methods, with a rank-enrichment score two- to three-fold larger (Figure 5B). The top-k analysis separates the methods further (Appendix I.2): CIG alone is enriched at every cutoff across cell types, while DeepSHAP and integrated gradients show no enrichment among the top 5–10 genes and recover only a fraction of CIG's enrichment from k=25 onward. The recovery of DEGs (Appendix I.3, Table I.5) reveals that DeepSHAP and integrated gradients rank DEGs higher than CIG. CIG also shows the weakest correlation with the DE statistic across all cell types (Table I.3), likely because it prioritizes upstream regulators, which are rarely strong DEGs themselves. ORA results (Appendix I.4) confirm that CIG uniquely picks out pathways with transcription factors that regulate the interferon-β response, with significantly enriched pathways such as TNF-α/NF-κB, IL-6/JAK/STAT3, and interferon response. DeepSHAP, by contrast, enriches only for the interferon response, with the remaining pathways enriched by integrated gradients not reaching significance.

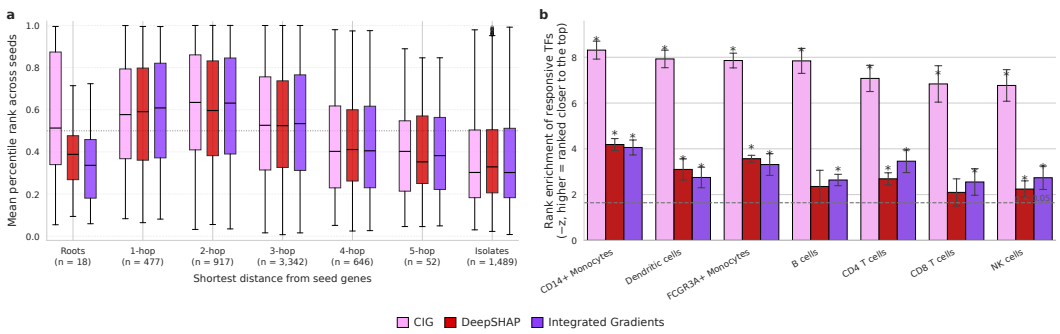


Figure 5: Recovery of stimulation-responsive regulators on the IFN-β dataset. (A) Distribution of attribution ranks for genes grouped by hop distance from the roots of the SCENIC validation graph, median indicated by a horizontal line within each box. (B) z-score of the Mann–Whitney U test statistic per method and cell type, averaged across 5 seeds; higher values indicate responsive TFs are ranked higher than other genes; error bars denote standard deviation across seeds; $*$: FDR < 0.05 for every seed.

5 Limitations and Future Work

Our method requires prior knowledge of the directed causal structure among features, which for biological data can often be sourced from curated knowledge bases such as OmniPath (Türei et al., 2021). Since such a graph may be imperfect in practice, we test how much this assumption can be violated before performance suffers by removing, adding, and reversing edges: CIG degrades under genuine information loss (removal or reversal) but tolerates spurious additions almost entirely, so the graph need not be exactly correct to retain most of the benefit (Appendix K). A separate concern is whether CIG's advantage merely reflects graph access, not correct use of it. We rule this out: CIG's attributions are instance-specific, tracking each cell's actual transition rather than the static graph alone (Appendix J). Like causal Shapley values (Heskes et al., 2020) and conditional SHAP (Aas et al., 2021), CIG is true to the data rather than to the model: it recovers causal drivers under the supplied structure and is not meant for debugging a model's idiosyncratic behavior. Extending the method to undirected or partially directed graphs is a natural direction for future work.

Our surrogate models each feature's mechanism as additive across parents (Section 3.3) and so cannot capture non-additive parent interactions. This is a limitation of our chosen surrogate, not of the method, since any differentiable surrogate can be substituted. The one case a different surrogate cannot fix is an unmodeled confounder, whose effect is folded into the additive residual and attributed to x_i's direct effect. In our experiments this biases attribution magnitudes for affected nodes (Appendix L).

Our methods scale quadratically in the number of features in the worst case, slower than integrated gradients but substantially faster than Shapley-based alternatives; our synthetic and real-data experiments (100–1000 features; 2,320–6,941 genes) already exceed the scale at which existing dependency-aware baselines have been validated (Heskes et al. (2020): $M{=}7$ features; Aas et al. (2021): 3–10 features, largest real-data example at 28). Finally, the current formulation concentrates attribution on root nodes, potentially underweighting relevant intermediate nodes; future work could explore distributing indirect effects more evenly across the causal path.

6 Conclusion

We introduced causal integrated gradients and causal expected gradients, which extend integrated gradients to incorporate known causal structure among input features. They apply to any differentiable model and, under conditions on the causal graph and surrogate, retain the axiomatic guarantees of integrated gradients. On a synthetic benchmark, CIG best matches ground-truth causal effects at a fraction of the cost of causal Shapley values; on simulated and real single-cell data, it best recovers the transcription factors driving the response while scaling to thousands of genes.

AI USE STATEMENT

Of the tasks requiring disclosure under the ICLR AI policy, we used generative AI tools only for assisting in the writing of proofs and for feedback on research methodology (Claude and ChatGPT); the underlying mathematical claims and all methodological decisions are the authors' own. We did not use generative AI for any other task in this category (e.g., generating synthetic datasets, developing theoretical models or conceptual frameworks, formulating mathematical claims, proposing or refining hypotheses, designing experiments, implementing methods, cleaning or reformatting datasets, or interpreting results). We also used generative AI tools for tasks under the recommended-disclosure category: drafting and editing portions of the paper text and improving its structure and readability (Claude, via Claude Code, and ChatGPT), formatting references, general-purpose coding assistance for data processing, plotting, and experiment scripts rather than the core CIG/CEG method implementation (GitHub Copilot), and brainstorming and literature search, such as surfacing related work and phrasing suggestions (Claude). We have reviewed all AI-assisted work: all AI-suggested text was reviewed and approved by the authors before inclusion, all AI-assisted proofs were verified by the authors, and all AI-assisted code was independently verified and tested by an author before being used to produce any reported results. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work uses only public, de-identified data: the real perturbation dataset (Kang et al., 2018) was collected and published by others, and no new human-subject data was collected for this study. The attributions produced by our method are intended to prioritize candidate causal genes for downstream experimental follow-up, not to be interpreted as definitive causal or clinical claims; any prioritized targets should be validated experimentally before drawing biological or clinical conclusions.

REPRODUCIBILITY STATEMENT

We provide an anonymized implementation of our method (CIG/CEG) and the code for the synthetic CATE benchmark (Section 4.1) as supplementary material. Full implementation and training details for the Dyngen and IFN-β experiments — including preprocessing, hyperparameters, and dataset statistics — are given in Appendix E and the corresponding experiment-specific appendices (Appendices G and H); both the Dyngen simulator and the IFN-β dataset (Kang et al., 2018) are themselves publicly available. Complete proofs of all theoretical claims, including the axiomatic properties and complexity analysis, are provided in Appendices C and D.

REFERENCES

Kjersti Aas, Martin Jullum, and Anders Løland. Explaining individual predictions when features are dependent: More accurate approximations to Shapley values. *Artificial Intelligence*, 298:103502, 2021.

Sara Aibar, Carmen Bravo González-Blas, Thomas Moerman, Vân Anh Huynh-Thu, Hana Imrichova, Gert Hulselmans, Florian Rambow, Jean-Christophe Marine, Pierre Geurts, Jan Aerts, Joost van den Oord, Zeynep Kalender Atak, Jasper Wouters, and Stein Aerts. SCENIC: single-cell regulatory network inference and clustering. *Nature Methods*, 14(11):1083–1086, November 2017. ISSN 1548-7105. doi: 10.1038/nmeth.4463. URL https://www.nature.com/articles/nmeth.4463.

Robrecht Cannoodt, Wouter Saelens, Louise Deconinck, and Yvan Saeys. Spearheading future omics analyses using dyngen, a multi-modal simulator of single cells. *Nature Communications*, 12(1):3942, June 2021. ISSN 2041-1723. doi: 10.1038/s41467-021-24152-2. URL https://www.nature.com/articles/s41467-021-24152-2.

Hugh Chen, Joseph D Janizek, Scott Lundberg, and Su-In Lee. True to the model or true to the data? *arXiv preprint arXiv:2006.16234*, 2020.

Peter Eades, Xuemin Lin, and W.F. Smyth. A fast and effective heuristic for the feedback arc set problem. *Information Processing Letters*, 47(6):319–323, 1993. ISSN 0020-0190. doi:

10.1016/0020-0190(93)90079-O. URL https://www.sciencedirect.com/science/article/pii/002001909390079O.

Gabriel Erion, Joseph D Janizek, Pascal Sturmfels, Scott M Lundberg, and Su-In Lee. Improving performance of deep learning models with axiomatic attribution priors and expected gradients. *Nature Machine Intelligence*, 3(7):620–631, 2021.

Zhuoqing Fang, Xinyuan Liu, and Gary Peltz. GSEApy: a comprehensive package for performing gene set enrichment analysis in Python. *Bioinformatics*, 39(1):btac757, January 2023. ISSN 1367-4811. doi: 10.1093/bioinformatics/btac757. URL https://doi.org/10.1093/bioinformatics/btac757.

José Miguel Giménez and María Albina Puente. The Owen and the Owen-Banzhaf values applied to the study of the Madrid Assembly and the Andalusian Parliament in Legislature 2015–2019. In *Proceedings of the 8th International Conference on Operations Research and Enterprise Systems (ICORES)*, pp. 45–52. SCITEPRESS, 2019. doi: 10.5220/0007297000002104.

Tom Heskes, Evi Sijben, Ioan Gabriel Bucur, and Tom Claassen. Causal Shapley values: Exploiting causal knowledge to explain individual predictions of complex models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 4778–4789, 2020.

Joseph D. Janizek, Anna Spiro, Safiye Celik, Ben W. Blue, John C. Russell, Ting-I Lee, Matt Kaeberlin, and Su-In Lee. PAUSE: principled feature attribution for unsupervised gene expression analysis. *Genome Biology*, 24(1):81, April 2023. ISSN 1474-760X. doi: 10.1186/s13059-023-02901-4. URL https://doi.org/10.1186/s13059-023-02901-4.

Dominik Janzing, Lenon Minorics, and Patrick Blöbaum. Feature relevance quantification in explainable AI: A causal problem. In *International Conference on Artificial Intelligence and Statistics*, pp. 2907–2916. PMLR, 2020.

Hyun Min Kang, Meena Subramaniam, Sasha Targ, Michelle Nguyen, Lenka Maliskova, Elizabeth McCarthy, Eunice Wan, Simon Wong, Lauren Byrnes, Cristina M. Lanata, Rachel E. Gate, Sara Mostafavi, Alexander Marson, Noah Zaitlen, Lindsey A. Criswell, and Chun Jimmie Ye. Multiplexed droplet single-cell RNA-sequencing using natural genetic variation. *Nature Biotechnology*, 36(1):89–94, January 2018. ISSN 1546-1696. doi: 10.1038/nbt.4042. URL https://www.nature.com/articles/nbt.4042.

Andrei Kapishnikov, Subhashini Venugopalan, Besim Avci, Ben Wedin, Michael Terry, and Tolga Bolukbasi. Guided integrated gradients: An adaptive path method for removing noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5050–5058, 2021.

Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, and Orion Reblitz-Richardson. Captum: A unified and generic model interpretability library for PyTorch, 2020. URL https://arxiv.org/abs/2009.07896. arXiv:2009.07896.

Stephan Kolassa and Wolfgang Schütz. Advantages of the MAD/Mean ratio over the MAPE. *Foresight: The International Journal of Applied Forecasting*, (6):40–43, 2007.

Arthur Liberzon, Chet Birger, Helga Thorvaldsdóttir, Mahmoud Ghandi, Jill P. Mesirov, and Pablo Tamayo. The Molecular Signatures Database Hallmark Gene Set Collection. *Cell Systems*, 1(6):417–425, December 2015. ISSN 2405-4712, 2405-4720. doi: 10.1016/j.cels.2015.12.004. URL https://www.cell.com/cell-systems/abstract/S2405-4712(15)00218-5.

Michael I. Love, Wolfgang Huber, and Simon Anders. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology*, 15(12):550, December 2014. ISSN 1474-760X. doi: 10.1186/s13059-014-0550-8. URL https://doi.org/10.1186/s13059-014-0550-8.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, volume 30, pp. 4765–4774, 2017.

Joran Michiels, Johan Suykens, and Maarten De Vos. Explaining the model and feature dependencies by decomposition of the Shapley value. *Decision Support Systems*, 182:114234, 2024.

Woon Yee Ng, Li Rong Wang, Siyuan Liu, and Xiuyi Fan. Causal SHAP: Feature attribution with dependency awareness through causal discovery. In *2025 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2025.

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

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3145–3153. PMLR, 2017.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *2nd International Conference on Learning Representations (ICLR), Workshop Track*, 2014.

Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. In *3rd International Conference on Learning Representations (ICLR), Workshop Track*, 2015.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pp. 3319–3328. PMLR, 2017.

Xin Tang, Jiawei Zhang, Yichun He, Xinhe Zhang, Zuwan Lin, Sebastian Partarrieu, Emma Bou Hanna, Zhaolin Ren, Hao Shen, Yuhong Yang, Xiao Wang, Na Li, Jie Ding, and Jia Liu. Explainable multi-task learning for multi-modality biological data analysis. *Nature Communications*, 14(1):2546, May 2023. ISSN 2041-1723. doi: 10.1038/s41467-023-37477-x. URL https://www.nature.com/articles/s41467-023-37477-x.

Dénes Türei, Alberto Valdeolivas, Lejla Gul, Nicolàs Palacio-Escat, Michal Klein, Olga Ivanova, Márton Ölbei, Attila Gábor, Fabian Theis, Dezső Módos, Tamás Korcsmáros, and Julio Saez-Rodriguez. Integrated intra- and intercellular signaling knowledge for multicellular omics analysis. *Molecular Systems Biology*, 17(3):MSB20209923, March 2021. ISSN 1744-4292. doi: 10.15252/msb.20209923. URL https://doi.org/10.15252/msb.20209923.

Peiyu Yang, Naveed Akhtar, Zeyi Wen, and Ajmal Mian. Local path integration for attribution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 3173–3180, 2023.

Melvyn Yap, Rebecca L. Johnston, Helena Foley, Samual MacDonald, Olga Kondrashova, Khoa A. Tran, Katia Nones, Lambros T. Koufariotis, Cameron Bean, John V. Pearson, Maciej Trzaskowski, and Nicola Waddell. Verifying explainability of a deep learning tissue classifier trained on RNA-seq data. *Scientific Reports*, 11(1):2641, January 2021. ISSN 2045-2322. doi: 10.1038/s41598-021-81773-9. URL https://www.nature.com/articles/s41598-021-81773-9.

Yue Zhuo and Zhiqiang Ge. IG^2: Integrated gradient on iterative gradient path for feature attribution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(11):7173–7190, 2024. doi: 10.1109/TPAMI.2024.3388092.

A Implementation for longer dependency paths

The topological ordering used to define the path is a bookkeeping device, not a modeling choice: any two valid topological orderings of the same DAG agree on every parent-child pair, so R_i is invariant to which one is used.

Indirect attributions through dependency paths of length two (referred to as second-order) can be incorporated by extending the formula with additional terms:

$$R_i = \int_0^1 \frac{\partial F}{\partial \gamma_i}\frac{\partial \gamma_i}{\partial \alpha}\,d\alpha + \sum_{j>i}\int_0^1 \frac{\partial F}{\partial \gamma_j}\frac{\partial \gamma_j}{\partial \gamma_i}\frac{\partial \gamma_i}{\partial \alpha}\,d\alpha + \sum_{\substack{j>i\\ i<k<j}}\int_0^1 \frac{\partial F}{\partial \gamma_j}\frac{\partial \gamma_j}{\partial \gamma_k}\frac{\partial \gamma_k}{\partial \gamma_i}\frac{d\gamma_i}{d\alpha}\,d\alpha. \tag{10}$$

The last (two-hop) term uses the full derivative $\frac{d\gamma_i}{d\alpha}$ rather than the local $\frac{\partial\gamma_i}{\partial\alpha}$ used in the direct and one-hop terms: it is the deepest dependency explicitly tracked, so it must fold in any effect of x_i's own ancestors that is not otherwise given its own term, exactly as the one-hop formula's indirect term already does for its (only) tracked layer. Paths of greater length can be handled analogously, tracking one additional layer with local partials and moving the full-derivative fold to the new deepest layer; our current implementation stops at two hops, which sufficed in all our experiments. This construction keeps $\sum_i R_i = F(\boldsymbol{x}) - F(b)$ exact at any truncation depth H (the number of explicitly tracked hops): the untracked remainder beyond the deepest tracked layer is always folded into that layer's full derivative rather than discarded.

The general H-hop formula. The one-hop (Equation 3) and two-hop (Equation 10) formulas above are the $H = 1$ and $H = 2$ instances of one pattern. Writing $(J^m)_{ji}$ for the (j, i) entry of the m-th power of the local-Jacobian matrix $J_{ji} = \partial\gamma_j/\partial\gamma_i$ — the sum, over every directed path of length exactly m from i to j, of the product of J along it, with $(J^0)_{ji} = \delta_{ji}$ — this pattern is, for general $H \geq 1$,

$$\begin{aligned} R_i^{(H)} = &\int_0^1 \frac{\partial F}{\partial\gamma_i}\frac{\partial\gamma_i}{\partial\alpha}\,d\alpha + \sum_{m=1}^{H-1}\sum_{j:\,(J^m)_{ji}\neq 0}\int_0^1 \frac{\partial F}{\partial\gamma_j}(J^m)_{ji}\frac{\partial\gamma_i}{\partial\alpha}\,d\alpha \\ &+ \sum_{j:\,(J^H)_{ji}\neq 0}\int_0^1 \frac{\partial F}{\partial\gamma_j}(J^H)_{ji}\frac{d\gamma_i}{d\alpha}\,d\alpha. \end{aligned} \tag{11}$$

Appendix C proves this satisfies the same axioms as the one-hop and two-hop instances above, at every H.

B Details of the basis expansion

To increase the expressive capacity of the surrogate while preserving interpretability and edge-wise decomposition, each parent feature is transformed through a structured basis expansion applied independently per feature. Let x_i denote the value of feature i. Inputs are first standardized:

$$z_i = \frac{x_i - \mu_i}{\sigma_i}, \tag{12}$$

where μ_i and σ_i are the feature-wise mean and standard deviation estimated from the training data. Standardization ensures that the basis functions operate on comparable scales across features and improves numerical stability during optimization.

The resulting basis representation $\phi(x_i)$ combines three complementary function families:

$$\phi(x_i) = \Bigg[\underbrace{z_i, z_i^2, \ldots, z_i^D}_{\text{Polynomial basis}}, \underbrace{(\max(0, z_i - c_k))^p,\ k = 1, \ldots, K_t}_{\text{Truncated power basis}}, \underbrace{\sin(\omega_f z_i), \cos(\omega_f z_i),\ f = 1, \ldots, K_f}_{\text{Fourier basis}}\Bigg]. \tag{13}$$

The polynomial component models smooth low-order nonlinearities. For a polynomial degree D, the expansion contributes:

$$\phi_{\text{poly}}(x_i) = [z_i, z_i^2, \ldots, z_i^D], \tag{14}$$

yielding D basis functions per feature.

The truncated power basis introduces localized nonlinear behavior through learnable knot locations $\{c_k\}_{k=1}^{K_t}$. Each basis element is defined as:

$$\phi_{\text{tp},k}(x_i) = (\max(0, z_i - c_k))^p, \tag{15}$$

where p is the knot power controlling smoothness and curvature. These functions behave similarly to spline basis functions and allow the surrogate to capture sharp transitions or regime-specific effects. This component contributes K_t basis functions per feature.

The Fourier component captures periodic structure:

$$\phi_{\text{fourier}}(x_i) = [\sin(\omega_1 z_i), \cos(\omega_1 z_i), \ldots, \sin(\omega_{K_f} z_i), \cos(\omega_{K_f} z_i)], \tag{16}$$

where ω_f denotes the frequency associated with Fourier feature f; these frequencies are set as hyperparameters. Since both sine and cosine terms are included for each frequency, this component contributes $2K_f$ basis functions per feature.

The total number of basis functions per feature is:

$$Q = D + K_t + 2K_f. \tag{17}$$

This construction provides a flexible yet structured functional approximation that can model smooth, localized, and periodic dependencies while preserving edge-level interpretability within the directed graph.

C AXIOMATIC PROPERTIES

Since our method belongs to the family of path methods (Sundararajan et al., 2017), it inherits many of the axiomatic properties of integrated gradients. We discuss each property in turn: a statement, a proof (or, for the one axiom we do not satisfy, an explanation of why), and, where our method requires more than the standard axiom does, the additional preconditions this needs. We would expect other dependency-aware attribution methods (e.g. causal Shapley values, Heskes et al., 2020; conditional SHAP, Aas et al., 2021) to require analogous preconditions on the supplied dependency structure, though to our knowledge this has not previously been examined for those methods. Proofs are given for the one-hop formula (Equation 3); implementation invariance, linearity, and sensitivity(a) carry over unchanged to the general H-hop formula (Equation 11), since its extra factors depend only on the graph and mechanisms, never on F, and its direct term is the same for every H. Completeness and symmetry need a short additional argument, given at the end of their proofs.

COMPLETENESS

Statement. The sum of attributions equals the difference in model output between the input and the baseline: $\sum_i R_i = F(\boldsymbol{x}) - F(b)$.

Proof. By the gradient theorem (fundamental theorem of calculus along the path γ),

$$\int_0^1 \nabla F(\gamma(\alpha)) \cdot \frac{d\gamma(\alpha)}{d\alpha}\, d\alpha = F(\gamma(1)) - F(\gamma(0)) = F(\boldsymbol{x}) - F(b). \tag{18}$$

Expanding the dot product into per-feature terms as in Equation 1, and grouping each feature i's direct and indirect contributions as in Equation 3, gives $\sum_i R_i = F(\boldsymbol{x}) - F(b)$. The fundamental theorem of calculus makes no reference to how γ is parametrized internally, so this holds exactly, regardless of the causal graph or which differentiable function is used for the mechanism c_i — whether the true structural mechanism or a learned surrogate s_i approximating it. □

Completeness at any truncation depth. The one-hop (Equation 3) and two-hop (Equation 10) formulas are the $H = 1$ and $H = 2$ instances of the general H-hop formula, Equation 11 (Appendix A), which tracks dependencies explicitly up to H hops with the local partial $\partial\gamma_i/\partial\alpha$ and folds every hop beyond H into that boundary layer's full derivative $d\gamma_i/d\alpha$. We show $\sum_i R_i^{(H)} = F(\boldsymbol{x}) - F(b)$ for every $H \geq 1$. Writing $\tilde{J} := (I - J)^{-1} = \sum_{m\geq 0} J^m$ (a finite sum, by nilpotency of J), $\tilde{J} = \sum_{m=0}^{H-1} J^m + J^H \tilde{J}$ for every H (both sides equal $\sum_{m\geq 0} J^m$). Multiplying by $\partial\gamma/\partial\alpha$ and using $d\gamma/d\alpha = \tilde{J}\,\partial\gamma/\partial\alpha$ gives, component-wise, $\frac{d\gamma_j}{d\alpha} = \sum_{m=0}^{H-1}\sum_i (J^m)_{ji}\frac{\partial\gamma_i}{\partial\alpha} + \sum_i (J^H)_{ji}\frac{d\gamma_i}{d\alpha}$ exactly, for every H. Substituting this into the completeness identity above and regrouping by attribution index i recovers exactly $\sum_i R_i^{(H)}$, so $\sum_i R_i^{(H)} = F(\boldsymbol{x}) - F(b)$ holds at any truncation depth H — provided the deepest tracked layer uses the full derivative, as in Equations 3 and 10.

Practical numerical error. The proof above is exact in the continuous limit for any differentiable choice of mechanism c_i, since the path itself does not depend on the accuracy of that choice. Two sources of numerical error remain when evaluating it in practice: (i) Riemann-sum discretization over K steps, identical in kind to vanilla integrated gradients, and (ii) the backward finite difference used for $\partial\gamma_i/\partial\alpha$ (Equation 6), which vanilla IG never needs since its path derivative is closed-form — genuinely new to our method, but vanishing as $\Delta\alpha \to 0$. Implementation invariance, symmetry,

and linearity (below) have no such gap, since they hold exactly for any K. None of our experiments apply post-hoc normalization; all reported attributions are raw.

We quantify this residual with the per-sample completeness delta $\delta^{(n)} = \sum_i R_i^{(n)} - (F(x^{(n)}) - F(b^{(n)}))$. On the synthetic benchmark (K=100), we report it relative to the output difference (Table C.1), computed batch-wise as a weighted absolute percentage error (WAPE), $\sum_n |\delta^{(n)}| / \sum_n |F(x^{(n)}) - F(b^{(n)})|$ over evaluation samples n within each seed, then averaged (mean $\pm$ std) across seeds. We use WAPE rather than the per-sample mean absolute percentage error (MAPE), since the latter is unstable whenever a single sample's $F(x^{(n)}) - F(b^{(n)})$ is near zero; WAPE is the standard fix for this in forecasting (Kolassa & Schütz, 2007). To also show the absolute magnitude of the residual, Figure C.1 gives the distribution of $|\delta^{(n)}|$ at one representative setting (774 synthetic features). For the two biological datasets, where near-zero output differences are more common, we report only this absolute delta (Figures C.2 and C.3), which sidesteps the ratio entirely.

Table C.1: Relative completeness delta (WAPE), mean $\pm$ std over seeds.

Setting	CIG	IG
Synthetic, 100 features	$1.90 \times 10^{-3} \pm 1.16 \times 10^{-3}$	$5.77 \times 10^{-8} \pm 6.45 \times 10^{-9}$
Synthetic, 129 features	$1.40 \times 10^{-3} \pm 3.78 \times 10^{-4}$	$5.53 \times 10^{-8} \pm 2.34 \times 10^{-9}$
Synthetic, 166 features	$1.38 \times 10^{-3} \pm 2.99 \times 10^{-4}$	$5.96 \times 10^{-8} \pm 4.62 \times 10^{-9}$
Synthetic, 215 features	$1.60 \times 10^{-3} \pm 4.11 \times 10^{-4}$	$6.56 \times 10^{-8} \pm 4.21 \times 10^{-9}$
Synthetic, 278 features	$1.75 \times 10^{-3} \pm 5.08 \times 10^{-4}$	$6.59 \times 10^{-8} \pm 3.51 \times 10^{-9}$
Synthetic, 359 features	$1.96 \times 10^{-3} \pm 3.64 \times 10^{-4}$	$7.09 \times 10^{-8} \pm 5.30 \times 10^{-9}$
Synthetic, 464 features	$1.91 \times 10^{-3} \pm 3.30 \times 10^{-4}$	$7.49 \times 10^{-8} \pm 1.15 \times 10^{-8}$
Synthetic, 599 features	$1.82 \times 10^{-3} \pm 2.17 \times 10^{-4}$	$8.22 \times 10^{-8} \pm 3.17 \times 10^{-9}$
Synthetic, 774 features	$1.99 \times 10^{-3} \pm 3.83 \times 10^{-4}$	$7.59 \times 10^{-8} \pm 5.66 \times 10^{-9}$
Synthetic, 1000 features	$1.95 \times 10^{-3} \pm 2.64 \times 10^{-4}$	$8.01 \times 10^{-8} \pm 5.38 \times 10^{-9}$

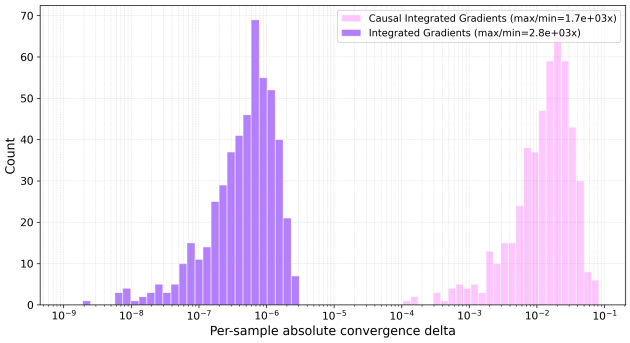


Figure C.1: Per-sample absolute completeness delta $|\delta^{(n)}|$ at 774 features ($n = 500$, pooled across 5 seeds). Median: CIG 1.48×10^{-2}, IG 5.74×10^{-7}. Proportional spread (max/min): CIG $1.71 \times 10^{3}\times$, IG $2.77 \times 10^{3}\times$ — the same order of magnitude.

CIG's delta is consistently one or more orders of magnitude larger than IG's, as expected rather than indicative of worse attribution quality: IG's path derivative is closed-form, while CIG additionally estimates the surrogate-residual derivative via finite differences, an extra discretization-error layer IG lacks. Real-data residuals exceed the synthetic ones: on synthetic data the mechanism is the exact generating function, so $h_i(\alpha)$ reduces to pure exogenous noise, whereas on real data $h_i(\alpha)$ additionally conflates the surrogate's fitting error, which need not vary smoothly with α along the path. Since all our evaluation metrics depend only on how attributions rank features, a residual of this size is unlikely to affect them.

This residual converges as expected for a first-order discretization scheme: Figure C.4 sweeps K at 100 synthetic features, showing the relative completeness delta falling off linearly on a log-log plot, roughly halving with every doubling of K — the textbook $O(1/K)$ signature shared by both

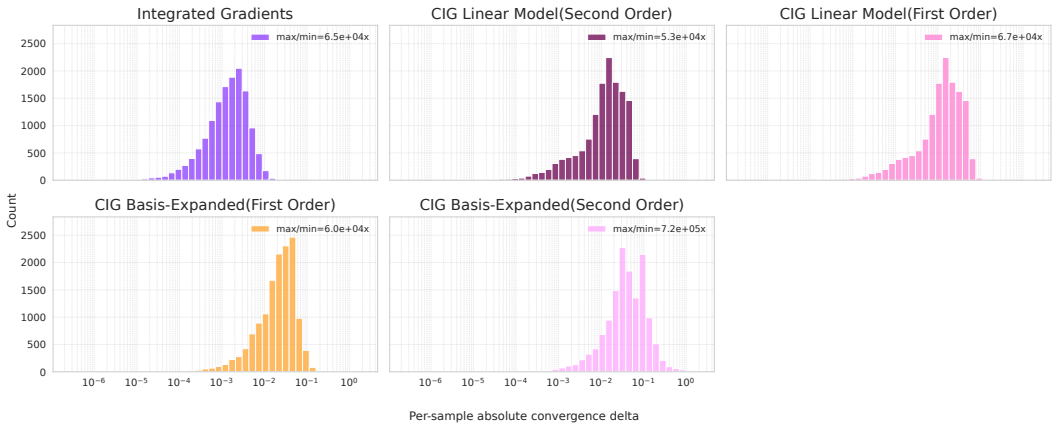


Figure C.2: Per-sample absolute completeness delta $|\delta^{(n)}|$ for the Dyngen dataset ($n = 14{,}076$, pooled across 5 seeds).

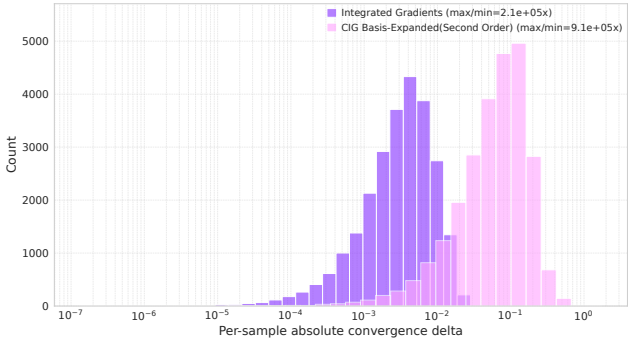


Figure C.3: Per-sample absolute completeness delta $|\delta^{(n)}|$ for the IFN-β dataset (6,941 features; $n = 25{,}450$, pooled across 5 seeds). Median: CIG 6.75×10^{-2}, IG 3.34×10^{-3}. Proportional spread (max/min): CIG $9.05 \times 10^{5}\times$, IG $2.11 \times 10^{5}\times$ — the same order of magnitude.

the Riemann sum and the finite-difference derivative. This sweep does not isolate which of the two dominates, only that the combined numerical error vanishes at the expected rate.

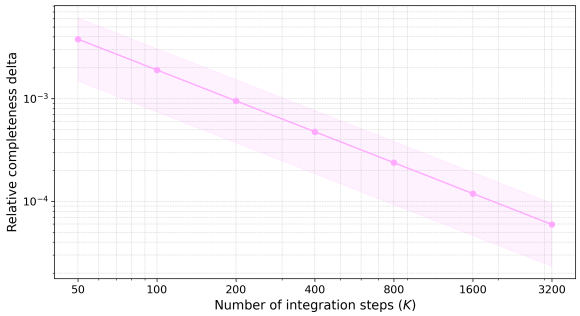


Figure C.4: Relative completeness delta of CIG as a function of the number of integration steps K, at 100 synthetic features (log-log axes). The shaded band shows the standard deviation over seeds. The linear trend on log-log axes is the $O(1/K)$ signature of first-order discretization error.

IMPLEMENTATION INVARIANCE

Statement. Attributions are invariant to the implementation of the model, depending only on the input-output behavior of F.

Precondition. The causal graph and the mechanism's input-output relation (c_i as a function, or s_i if using surrogates, not its specific implementation) are held fixed; only the model F varies.

Proof. Let F, F' be two models with identical input-output behavior, i.e. $F(z) = F'(z)$ for all z, evaluated under the same causal graph and the same mechanisms c_i (or surrogates s_i). The path points $\gamma^{(p)}$ visited by the Riemann sum, and the gradients $\partial\gamma_i/\partial\alpha$, $\partial\gamma_j/\partial\gamma_i$ evaluated at those points (Equation 6), depend only on the graph, $\alpha^{(p)}$, and c_i (or s_i) — never on F. Hence at every path point p,

$$\frac{\partial F(\gamma^{(p)})}{\partial \gamma_i} = \frac{\partial F'(\gamma^{(p)})}{\partial \gamma_i} \tag{19}$$

follows directly from $F \equiv F'$. Since every term of Equation 3 involves F only through such partial derivatives evaluated at these shared path points, $R_i^F = R_i^{F'}$ for every feature i and every K. □

This precondition is not an additional restriction beyond the standard axiom — it is what implementation invariance means: comparing two models under one shared explanation setup, not two different setups. Changing the graph or the mechanisms is meant to change attributions.

SENSITIVITY(A)

Statement. If the input and baseline differ in exactly one feature i and yield different predictions, then feature i should receive a non-zero attribution. We prove this for the direct attribution component.

Proof. Since the line integral uses a straight-line path between baseline and input, all features $j \neq i$ remain constant along the path, so their total derivatives vanish:

$$\frac{d\gamma_j}{d\alpha} = 0, \quad j \neq i. \tag{20}$$

By the chain rule, the total derivative of the varying feature γ_i along the path is:

$$\frac{d\gamma_i}{d\alpha} = \frac{\partial \gamma_i}{\partial \alpha} + \sum_{j \neq i} \frac{\partial \gamma_i}{\partial \gamma_j} \frac{d\gamma_j}{d\alpha} = \frac{\partial \gamma_i}{\partial \alpha} + \sum_{j \neq i} \frac{\partial \gamma_i}{\partial \gamma_j} \cdot 0 = \frac{\partial \gamma_i}{\partial \alpha}. \tag{21}$$

Consequently, the direct attribution of feature i reduces to the standard integrated gradients attribution:

$$R_{i,\text{direct}}^{\text{CIG}} = \int_0^1 \frac{\partial F}{\partial \gamma_i} \frac{\partial \gamma_i}{\partial \alpha}\, d\alpha = \int_0^1 \frac{\partial F}{\partial \gamma_i} \frac{d\gamma_i}{d\alpha}\, d\alpha = R_i^{\text{IG}}. \tag{22}$$

Since integrated gradients satisfies sensitivity(a), it follows that the direct attribution component of causal integrated gradients does as well. □

SYMMETRY

Statement. If two features have the same value and baseline and are symmetric with respect to F, they receive equal attributions.

Precondition. Beyond the standard requirement (that F be symmetric in features i, k with $x_i = x_k$, $b_i = b_k$), our method additionally requires: (S1) i and k have identical parents and identical children in the causal graph — which also implies they cannot be linked by an edge, since $i \to k$ would place i in k's parent set but not its own; and (S2) the mechanism is exchangeable ($c_i \equiv c_k$ as functions, or $s_i \equiv s_k$ if using surrogates, not necessarily identical implementation), with every shared child's mechanism symmetric in i, k.

Proof. By (S1), i and k share the same parent set. Together with $x_i = x_k$, $b_i = b_k$, and $c_i \equiv c_k$ (S2) applied to that shared parent set, an induction over the topological order gives $\gamma_i^{(p)} = \gamma_k^{(p)}$

at every path point p: the base case holds for root features by $x_i = x_k$, $b_i = b_k$ directly, and the inductive step holds because i and k receive identical inputs from their (identical, by (S1)) parents and apply the same mechanism (S2). Consequently $\partial\gamma_i/\partial\alpha = \partial\gamma_k/\partial\alpha$ at every path point, so the direct terms of R_i and R_k agree, since F is symmetric in i, k by hypothesis. For the indirect terms, (S1) gives i, k identical children, and (S2) makes every shared child's mechanism symmetric in i, k, so $\partial\gamma_j/\partial\gamma_i = \partial\gamma_j/\partial\gamma_k$ for every shared child j; combined with F's symmetry in i, k, the indirect sums in Equation 3 agree term-by-term. Hence $R_i = R_k$. □

If i, k occupy different positions in the graph, our method attributes them asymmetrically by design, since it is meant to reflect their differing causal roles.

Symmetry at any truncation depth. The same preconditions (S1)+(S2) suffice for $R_i^{(H)} = R_k^{(H)}$ (Equation 11) at every H, not only $H = 1$. Two extensions of the argument above are needed. First, since $\gamma_i^{(p)} \equiv \gamma_k^{(p)}$ holds identically as a function along the path, not merely pointwise, their derivatives agree at every order too, so $\frac{d\gamma_i}{d\alpha} = \frac{d\gamma_k}{d\alpha}$ in addition to $\frac{\partial\gamma_i}{\partial\alpha} = \frac{\partial\gamma_k}{\partial\alpha}$. Second, $(J^m)_{ji} = (J^m)_{jk}$ for every $m \geq 1$ and every j: decomposing a length-m path from i by its first step, $(J^m)_{ji} = \sum_{l \in \text{children}(i)} (J^{m-1})_{jl} J_{li}$, and since i, k have identical children (S1) and $J_{li} = J_{lk}$ for every shared child l (established above), this equals $\sum_{l \in \text{children}(k)} (J^{m-1})_{jl} J_{lk} = (J^m)_{jk}$ — one identity valid for every m simultaneously, not an induction on m. Combining both facts, every term of $R_i^{(H)}$ — direct, every middle layer, and the boundary layer — matches the corresponding term of $R_k^{(H)}$, so $R_i^{(H)} = R_k^{(H)}$ for every H.

Linearity

Statement. If $F = aF_1 + bF_2$, the attributions for F equal the same linear combination of the attributions for F_1 and F_2: $R_i^F = aR_i^{F_1} + bR_i^{F_2}$.

Precondition. F_1 and F_2 share the same causal graph and the same mechanism's input-output relation (c_i as a function, or s_i if using surrogates, not its specific implementation), differing only in how they are combined into F.

Proof. Under this precondition, the path points $\gamma^{(p)}$ and path derivatives $\partial\gamma_i/\partial\alpha$, $\partial\gamma_j/\partial\gamma_i$ are identical for F_1 and F_2 (they depend only on the shared graph and mechanisms). By linearity of the partial derivative,

$$\frac{\partial F}{\partial \gamma_i} = a\,\frac{\partial F_1}{\partial \gamma_i} + b\,\frac{\partial F_2}{\partial \gamma_i} \tag{23}$$

at every path point. Substituting into Equation 3 — both the direct term and every indirect term, which use only these shared path derivatives — gives $R_i^F = aR_i^{F_1} + bR_i^{F_2}$ for every feature i. □

This axiom is mainly of practical interest for ensembles: since the graph and mechanisms are properties of the data rather than of any individual ensemble member, an ensemble $F = aF_1 + bF_2$ trained on one dataset satisfies the precondition automatically, so linearity holds without needing to verify anything further. It would only fail if F_1, F_2 were deliberately given different graphs or different mechanisms — a scenario with little practical relevance.

Dummy

Statement. A feature that does not appear in F should receive zero attribution.

Our method does not satisfy this axiom. A feature absent from F but causally upstream of a feature that does appear can receive a non-zero indirect attribution: in the two-feature example in the right panel of Figure 2, if F_1 depends only on x_2 and x_2 depends on x_1, then x_1 receives the non-zero indirect term of Equation 2 even though F_1 never reads x_1 directly. This is a deliberate design choice — capturing such upstream influence is precisely the goal of incorporating causal dependencies — and the same behavior is exhibited by other dependency-aware methods such as conditional SHAP (Aas et al., 2021) and causal Shapley values (Heskes et al., 2020).

D COMPUTATIONAL COMPLEXITY ANALYSIS

We give a theoretical, not just empirical, complexity comparison against causal Shapley values (Heskes et al., 2020) and conditional SHAP (Aas et al., 2021). Let d be the number of input features, C the number of output classes attributed (typically 1), K the number of Riemann integration steps, M the number of SHAP permutations, and E the number of edges in the causal graph. Throughout, we assume a fixed-width, fixed-depth network, so a single forward(+backward) pass costs $O(d)$, dominated by the input/output layers rather than by hidden width; every entry below inherits this. Table D.1 gives the cost per explained instance.

Table D.1: Complexity per explained instance.

Method	Cost
IG (vanilla)	$O(K \cdot d \cdot C)$
CIG (ours)	$O(K \cdot d \cdot C)$ for a sparse causal graph ($E{=}O(d)$); $O(K \cdot d^2)$ for a dense graph (worst case)
Causal Shapley values (sampled, as implemented)	$O(M \cdot d^2)$
Conditional SHAP (sampled, as implemented)	$O(M \cdot d^3)$ (one-time $O(d^3)$ Cholesky)
Exact Shapley value (causal or conditional)	$O(2^d)$

CIG's cost beyond a single $O(d)$ forward/backward pass comes from the surrogate Jacobian needed for the indirect term. With a dense causal graph, computing this Jacobian via autograd needs one backward pass per output feature — d passes total, each itself $O(d)$ — giving $O(d^2)$: this is the generic cost of any dense $d \to d$ Jacobian, not specific to our method. With a sparse causal graph (low mean degree), our sparse masked surrogate's Jacobian is computed analytically via the chain rule, evaluating derivatives only for active edges — touching only the $O(d)$ entries that actually exist, and skipping this d-fold blow-up entirely. Our synthetic scaling experiments (right panel of Figure 3) in fact compute the full (dense) Jacobian rather than the closed-form sparse one, which is why the measured runtimes there follow the $O(d^2)$ line above rather than the $O(d)$ sparse case. Our real graphs support the sparse regime: the OmniPath-derived network (6,941 genes) has mean degree 3.38, and the Dyngen GRN (2,300–2,500 genes) has mean degree 3.07, both low and, notably, not growing with graph size, though both also contain a small number of high-degree hub genes (up to several hundred children each), consistent with the heavy-tailed degree distributions typical of real regulatory networks; what the $O(d)$ argument needs is the low mean, not a uniform cap on every node.

Second-order indirect effects. The optional second-order extension (Appendix A), which attributes effects mediated through two successive surrogate applications rather than one, applies the same Jacobian-and-contraction step a second time and so does not change CIG's complexity class in Table D.1 — only its constant, roughly $2\times$ the first-order-only cost.

Causal Shapley values and conditional SHAP, as commonly implemented via permutation sampling, are quadratic and cubic in d, respectively (conditional SHAP pays an extra factor of d for Gaussian conditioning per sample). The exact Shapley value is exponential, $O(2^d)$, regardless of sparsity. So CIG's polynomial cost is exact (up to ordinary path discretization) and improves with sparsity, while the Shapley baselines' polynomial cost is already a sampling approximation of an otherwise exponential quantity. This is not only a theoretical gap: at the scale of our real Dyngen and IFN-β experiments, causal Shapley values and conditional SHAP are excluded outright as computationally infeasible, and on our larger synthetic benchmarks conditional SHAP is excluded and causal Shapley values must run on a reduced sampling budget, so the polynomial/exponential gap above corresponds to the difference between applications that are and are not reachable at all in practice.

One more cost worth disclosing: surrogate training. With n the training-set size and T epochs, our closed-form sparse surrogate trains in $O(T \cdot n \cdot d)$ (sparse graph, $E{=}O(d)$) or $O(T \cdot n \cdot d^2)$ (dense); a generic dense-MLP surrogate is $O(T \cdot n \cdot d)$ regardless of sparsity. This is a one-time, offline cost amortized over all explanations, not a per-instance one: over N_{explain} instances it adds (training cost)$/N_{\text{explain}}$ to each, vanishing as N_{explain} grows.

E EXPERIMENTAL SETUP AND END-TO-END COST ACCOUNTING

All experiments were run on a server equipped with two Intel Xeon Gold 6438Y+ CPUs (32 cores each), two NVIDIA A40 GPUs (46 GB VRAM each), and approximately 1 TB of system RAM. We report here a full end-to-end cost breakdown (surrogate training vs. CIG's own attribution computation) for both biological experiments at $K{=}100$ integration steps in Table E.1.

Table E.1: End-to-end pipeline profiling: wall-clock time and peak CUDA memory per stage, for the full test set at $K{=}100$ integration steps. Dyngen benchmark (2,514–3,000 test cells per seed); IFN-β (5,090 cells $\times$ 6,941 genes). Values are mean $\pm$ std over 5 seeds. CEG averages over 100 random baselines; all CIG rows use a zero baseline.

Dataset	Method	Stage	Time (s)	Peak CUDA alloc. (MB)	Peak CUDA reserved (MB)
Dyngen	CIG (linear basis, 1st order)	Surrogate training	129.7 ± 4.2 (∼130 s)	224 ± 7	980 ± 958
		Attribution computation	175.4 ± 19.9 (∼175 s)	17818 ± 927	18680 ± 1333
		Total	**305.1 ± 23.6 (∼305 s)**	**17818 ± 927**	**18680 ± 1333**
Dyngen	CIG (linear basis, 2nd order)	Surrogate training	130.4 ± 5.0 (∼130 s)	221 ± 12	966 ± 970
		Attribution computation	349.6 ± 39.6 (∼350 s)	26653 ± 1416	27500 ± 2004
		Total	**480.0 ± 44.6 (∼480 s)**	**26653 ± 1416**	**27500 ± 2004**
Dyngen	CIG (expanded basis, 1st order)	Surrogate training	1246.4 ± 35.4 (∼21 min)	8966 ± 305	12818 ± 374
		Attribution computation	248.1 ± 25.3 (∼248 s)	19393 ± 1020	30485 ± 1285
		Total	**1494.5 ± 59.1 (∼25 min)**	**19393 ± 1020**	**30485 ± 1285**
Dyngen	CIG (expanded basis, 2nd order)	Surrogate training	1246.6 ± 35.3 (∼21 min)	8966 ± 305	12815 ± 376
		Attribution computation	494.3 ± 50.1 (∼494 s)	28228 ± 1511	39314 ± 1774
		Total	**1740.9 ± 83.3 (∼29 min)**	**28228 ± 1511**	**39314 ± 1774**
Dyngen	CEG (expanded basis, 2nd order)	Surrogate training	1417.5 ± 251.9 (∼24 min)	8966 ± 305	13550 ± 1572
		Attribution computation	66089.2 ± 2021.1 (∼18.4 hr)	29174 ± 1532	35869 ± 2056
		Total	**67506.7 ± 1971.1 (∼18.8 hr)**	**29174 ± 1532**	**35869 ± 2056**
IFN-β	CIG (expanded basis, 2nd order)	Surrogate training	6465.1 ± 233.3 (∼1.8 hr)	10627 ± 7	14255 ± 12
		Attribution computation	7152.1 ± 14.0 (∼2.0 hr)	19112 ± 1	30143 ± 12
		Total	**13617.1 ± 242.2 (∼3.8 hr)**	**19112 ± 1**	**30143 ± 12**

The pipeline is split into surrogate fitting and attribution, and the stage that dominates depends on the surrogate. For CIG both are modest, and the cost of moving from first- to second-order indirect effects is roughly a doubling of attribution time with a smaller increase in memory. CEG reuses the same surrogate and the same integration algorithm but repeats the integration for each reference baseline, so its attribution stage is two orders of magnitude slower. However, this is purely a time cost and its peak memory is essentially that of CIG. Peak memory is set by the attribution stage rather than by surrogate training, and grows mainly with the order of indirect effects. The basis expansion shifts the burden entirely onto training, raising both fitting time and training memory by an order of magnitude while increasing attribution time only moderately. It is a one-off amortizable surcharge for a non-linear conditional model and not a recurring per-explanation cost.

F DETAILS OF THE SYNTHETIC CAUSAL ATTRIBUTION BENCHMARK

Experimental setup. For each feature dimensionality and each of the 5 independent runs, a random causal graph of maximum depth 2 is generated, with expected node degree $k = 5$. Root nodes (those without parents) receive no structural assignment function; their value is pure exogenous noise. For every non-root node j, a mechanism type is drawn independently and uniformly at random from three options — linear, tanh, and interaction, each with probability $1/3$:

- **Linear:** $c_j(x_{\mathrm{pa}(j)}) = \sum_{i\in\mathrm{pa}(j)} w_i x_i$, with weights $w_i \sim \mathrm{Uniform}(-1, 1)$ drawn i.i.d. per parent.
- **Tanh:** $c_j(x_{\mathrm{pa}(j)}) = \tanh\Big(\sum_{i\in\mathrm{pa}(j)} w_i x_i\Big)$, with the same weight distribution as the linear case.
- **Interaction:** $c_j(x_1, x_2) = w_1 x_1 + w_2 x_2 + w_1 w_2 x_1 x_2$, with $w_1, w_2 \sim \mathrm{Uniform}(-1, 1)$. Only the first two parents are used regardless of how many parents the node actually has in the DAG; any additional parents receive a true structural weight of exactly zero — this only matters for nodes with more than two parents. If a node has fewer than two parents, this case degenerates to the linear assignment above.

Input features are sampled according to these assignments with additive exogenous Gaussian noise. Targets are generated as a random linear combination of a randomly chosen 20% subset of the input

features, with added Gaussian noise. The reported metric is the sample-wise Spearman correlation (averaged over 100 samples) between the local attributions and the individual conditional average treatment effects (CATEs), computed either over all input features (Figure F.1) or over root nodes only (left panel of Figure 3).

Conditional SHAP is implemented assuming a multivariate Gaussian distribution over the inputs, which allows the conditional distribution of any feature subset to be computed in closed form (Aas et al., 2021). Background samples are drawn from these conditional distributions. Causal Shapley values directly use the available structural causal model to sample from the interventional distribution implied by the do-operator, avoiding the need for distributional approximations. Specifically, the expectation

$$E[F \mid do(X_S = x_S)], \tag{24}$$

which appears in the computation of causal Shapley values, is evaluated by running the structural causal model (including exogenous noise) with features S fixed to x_S and all remaining features drawn from the background dataset. The model output is averaged over these background samples.

Both methods are implemented using the `PermutationExplainer` from the `shap` package (Lundberg & Lee, 2017) with 10 permutations and 50 background samples. For experiments with 464 or more features, where runtime became prohibitive, these were reduced to 4 permutations and 10 background samples. As a result, the exponential scaling of exact Shapley-value computation is not fully visible in the runtime plot (right panel of Figure 3): maintaining comparable approximation quality at higher feature counts would require *increasing*, not decreasing, these parameters, so the measured curve underestimates the true asymptotic gap.

Both causal integrated gradients and integrated gradients use 100 steps for the Riemann sum and a zero baseline. Causal expected gradients uses 10 random baselines and 10 steps for each baseline.

Root-node intuition. The intuition behind the root-node results is best illustrated with a simple example. Consider a model F with two equally contributing features x_1 and x_2, where x_1 fully causes x_2 (i.e. $x_1 \to x_2 \to F$, $x_1 \to F$):

- **Conditional SHAP** gives $R_1 = R_2$, since it does not account for the direction of the dependency and therefore does not preferentially attribute importance to the root node x_1.
- **Causal Shapley values** give $R_1 > R_2$, since they take into account the direction of the dependency but still attribute some importance to x_2, because they consider a situation where x_1 is not set to a value, in which case x_2 does have an effect on F.
- **Causal integrated gradients** gives $R_2 = 0$, correctly identifying that x_2 contributes no independent effect beyond what is already captured by x_1.

All-node results. Figure F.1 shows attribution quality when evaluated over all nodes.

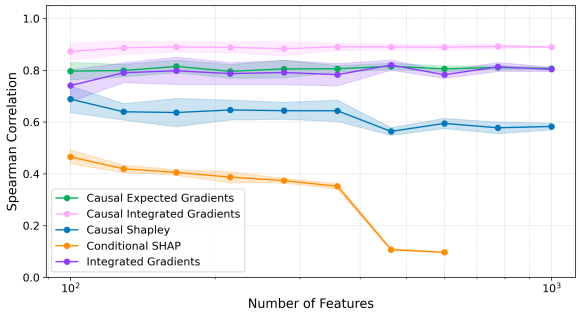
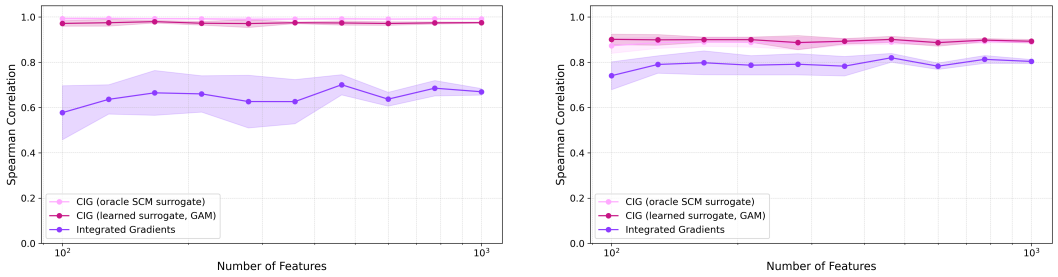


Figure F.1: Spearman correlation with the CATE averaged over all nodes, as a function of the number of features. Shaded areas denote the standard deviation over 5 runs.

Learned surrogate vs. true mechanisms. The main benchmark uses the SCM's true structural mechanisms directly when computing CIG's attributions, isolating attribution quality from surrogate error. To check the resulting gap between this idealized setting and practical use, we rerun the

benchmark using our learned surrogate — the expanded-basis variant of the sparse masked surrogate from Section 3.3 — in place of the true mechanisms, given only the causal graph's topology as prior knowledge; mechanisms themselves are learned from data, exactly as in our biological experiments. Figure F.2 reports Spearman correlation with ground-truth CATE across the full 100–1000 feature sweep, for both root-cause (eldest-ancestor) and all-node features, alongside graph-agnostic integrated gradients for reference.



(a) Root-cause (eldest-ancestor) features.

(b) All nodes.

Figure F.2: Spearman correlation with ground-truth CATE as a function of the number of features: CIG using the SCM's true mechanisms directly (oracle SCM surrogate) vs. CIG using our learned, expanded-basis surrogate (labeled "GAM" in the legend) vs. graph-agnostic integrated gradients. Shaded areas denote the standard deviation over 5 seeds.

The learned surrogate tracks the true-mechanism numbers closely across the entire sweep, and both remain far above integrated gradients, which ignores the graph entirely. The gap motivating this comparison is therefore negligible in practice and does not materially change attribution values or rankings relative to the idealized true-mechanism setting.

G Details of the Dyngen experiment

Table G.1: Summary statistics of the generated Dyngen linear dataset.

Statistic	Value
Trajectory topology	Linear
Cells	10,000
Transcription factors	48
Target genes	2,000
Housekeeping genes	500
Total genes	2,548
Total simulation time	1,000
Simulations	100

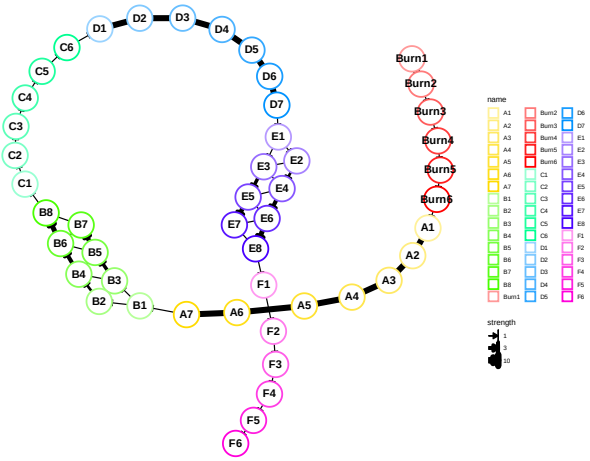


Figure G.1: Backbone module network for the generated Dyngen linear dataset.

Table G.2: Dyngen regression datasets across the five simulation seeds. Cells and genes are reported before and after quality-control filtering; the train/validation/test split is a nested 70/30 split of the filtered cells (49%/21%/30%).

	Cells		Genes		Filtered cells per split		
Seed	Raw	Filtered	Raw	Filtered	Train	Val.	Test
42	10,000	9,790	2,548	2,378	4,797	2,056	2,937
420	10,000	8,380	2,548	2,445	4,106	1,760	2,514
100	10,000	8,871	2,548	2,320	4,346	1,863	2,662
1000	10,000	10,000	2,548	2,388	4,900	2,100	3,000
2000	10,000	9,876	2,548	2,494	4,839	2,074	2,963

Table G.3: Hyperparameters of the surrogate model used for the Dyngen dataset.

Hyperparameter	**Value**
Number of knots	75
Knot power	3
Polynomial degree	3
Fourier features	10
Training epochs	2000
Weight decay	1×10^{-5}

Table G.4: Hyperparameters for data preprocessing and for the neural network model used to predict simulation time.

Hyperparameter	**Value**
Min genes expressed per cell	200
Min cells expressing a gene	100
Activation function	ReLU
Hidden layer sizes	(50, 25, 10)
Maximum iterations	500
Batch size	512
Solver	Adam
Alpha (L2 penalty)	1×10^{-4}
Learning rate	1×10^{-3}

Table G.5: Feature attribution methods evaluated in the simulated single-cell experiment.

Method	**Used**	**Baseline**	**Notes**
DeepSHAP (Lundberg & Lee, 2017)	✓	Random set	`captum` implementation
GradientSHAP (Lundberg & Lee, 2017)	✓	Random set	`captum` implementation
Integrated gradients (Sundararajan et al., 2017)	✓	Zero	`captum` implementation
DeepLIFT (Shrikumar et al., 2017)	✓	Zero	`captum` implementation
Saliency (Simonyan et al., 2014)	✓	×	`captum` implementation
Guided backpropagation (Springenberg et al., 2015)	✓	×	`captum` implementation
Partition explainer (Lundberg & Lee, 2017)	✓	Correlation	`shap` implementation
CIG (first-order)	✓	Zero	Linear/expanded basis (Table G.3)
CIG (second-order)	✓	Zero	Linear/expanded basis (Table G.3)
CEG extension (second-order)	✓	Random set	Expanded basis (Table G.3)
Conditional SHAP (Aas et al., 2021)	×	–	Not computationally feasible
Causal Shapley values (Heskes et al., 2020)	×	–	Not computationally feasible

Table G.6: Gene relevance scoring scheme based on the simulated gene regulatory network (GRN). The DCG at cutoff K is $\mathrm{DCG@}K = \sum_{i=1}^{K} \mathrm{rel}_i / \log_2(i+1)$, where rel_i is the score of the gene at rank i.

Gene Category	Score	Description
Backbone transcription factors (TFs)	+1.00	TFs defining main differentiation trajectory in GRN
Direct targets of backbone TFs	+0.25	Genes regulated by one backbone TF
Joint targets with opposing interaction signs	+0.25	Genes jointly regulated by two TFs with opposite regulatory signs
Burn-in phase TFs	-0.25	TFs active only during burn-in phase
Targets of burn-in TFs	-0.25	Genes regulated by burn-in phase TFs
Housekeeping genes	-0.25	Non-informative constitutively expressed genes
Other genes	0.00	Genes not belonging to the categories above

H Details of the IFN-β experiment

Table H.1: Hyperparameters of the surrogate model used for the IFN-β dataset.

Hyperparameter	Value
Number of knots	50
Knot power	3
Polynomial degree	5
Fourier features	5
Training epochs	1000
Weight decay	1×10^{-5}

Table H.2: Hyperparameters for data preprocessing and for the neural network model used to predict IFN-stimulated versus control cells.

Hyperparameter	Value
Min genes expressed per cell	200
Min counts per gene	250
Genes before filtering	15,706
Cells before filtering	24,673
Genes after filtering	6,941
Cells after filtering	24,562
Input dimension	6,941
Activation function	ReLU
Hidden layer sizes	(25,)
Maximum iterations	10
Batch size	512
Solver	Adam
Alpha (L2 penalty)	1×10^{-4}
Learning rate	1×10^{-3}
Train/validation/test sizes	17,448/2,024/5,090

OmniPath prior graph construction. Since no causal graph is available for this dataset, we leverage prior knowledge from OmniPath (Türei et al., 2021). After restricting the graph to genes present in the dataset and to the interactions annotated as transcriptional, the resulting network contains 6,941 genes and 22,887 edges. Each edge was then assigned a confidence score as a weighted mean of four evidence channels: curation effort, number of references, number of primary sources, and consensus direction. Duplicate source–target records were collapsed into a single edge by taking the maximum of each evidence channel, reducing the edges to 22,624 aggregated interactions. Edges without supporting evidence in any channel were removed, and the edges with the top 75% of weights were kept, leaving 12,136 edges. Because a topological ordering of the features is required, we converted the cyclic graph to a DAG by approximate minimum feedback arc set removal. Specifically, we first computed a linear vertex ordering using the weighted greedy heuristic of Eades et al. (1993), removed arcs inconsistent with this ordering, and then reconsidered removed arcs in descending confidence order, reinserting an arc whenever doing so preserves acyclicity. This removed 399 of 12,136 edges (3.3% of edges and 3.6% of total confidence mass), yielding an acyclic prior of 11,737 edges over 3,792 connected genes.

SCENIC evaluation graph construction. **Data.** We started from the PBMC scRNA-seq dataset of Kang et al. (2018), comprising 8 lupus patients, with each patient's cells split into IFN-β-stimulated and control samples. We used the raw UMI counts throughout the SCENIC steps.

Quality control. We kept cells with more than 200 detected genes, fewer than 4,000 detected genes, and less than 15% mitochondrial reads. We kept genes with more than 250 total counts across all cells, which matches the SCENIC recommendation of one count in 1% of cells. The filtered matrix contained 24,562 cells $\times$ 6,941 genes. We restricted the data to six training patients (1244, 1016, 1039, 1256, 1488, and 107), comprising 17,448 cells, leaving patients 1015 and 101 out.

Multi-seed SCENIC. GRNBoost2 is stochastic, so we ran pySCENIC ten times with seeds {0, 1, 3, 4, 5, 20, 42, 420, 1000, 2000}. Each run had three steps:

1. *GRN inference*: GRNBoost2 on the raw count matrix, with the hg38 list of human TFs as candidate regulators. Output: TF–target adjacencies weighted by importance.
2. *Module creation and cisTarget pruning*: co-expression modules were built with the default settings, so TF–target correlations use only cells where both genes are detected. Modules were then pruned by motif enrichment. We used two hg38 RefSeq-r80 mc9nr gene-based ranking databases (10 kb up/downstream of the TSS, and 500 bp up/100 bp down) and the motif-to-TF annotation motifs-v9-nr.hgnc-m0.001-o0.0.
3. *Regulon filtering*: we kept enriched motifs with NES greater than 3.0 and motif-similarity q-value less than 0.001. Surviving motifs were merged into one regulon per TF and context, which takes the union of their targets.

Consensus network across seeds. Each seed's regulons were flattened into directed TF-to-target edges. When a TF had several contexts, an edge kept its largest GRNBoost2 weight. Seeds agreed moderately: pairwise Jaccard similarity of the edge sets was 0.58 to 0.60. We chose the recurrence threshold from the edge-recurrence curve (Table H.3). An edge entered the consensus if it appeared in at least 8 of 10 seeds, and its weight is the median over the seeds that contained it. The training-patient data give 23,451 edges from 155 TFs to 5,442 targets.

Table H.3: Edges kept at each recurrence threshold (full data).

Min. seeds	**1**	**2**	**4**	**6**	**8**	**10**
Edges	74,426	58,082	42,448	33,172	26,379	19,055
TFs	252	229	193	174	156	131

Consensus regulon activity (AUCell). For each seed we scored regulon activity per cell with pyscenic aucell, using an AUC threshold of 0.10. That means the top 10% of each cell's ranked genes; the median cell detects about 500 genes. A regulon was kept if (i) its TF is in the consensus network and (ii) AUCell scored it in at least 8 of 10 seeds. Its activity is the mean AUC over the seeds that scored it, and a seed that did not score it counts as missing rather than zero. The result is 155 regulons.

Extraction of stimulation-responsive SCENIC regulons. For each cell type, SCENIC regulon activities, quantified using AUCell on the training replicates, were averaged within each replicate and contrasted between the stimulated and control groups. Regulons significant according to a one-sample t-test across replicates (BH-FDR < 0.05) were retained, and the 50 regulons with the largest $|\Delta\mathrm{AUC}|$ were selected.

I Additional IFN-β results

I.1 Attribution rankings by hop distance from the roots

The rankings are reproducible across training seeds: the per-gene spread of the percentile rank across seeds stays well below what independent seed-to-seed rankings would produce (Tables I.1 and I.2).

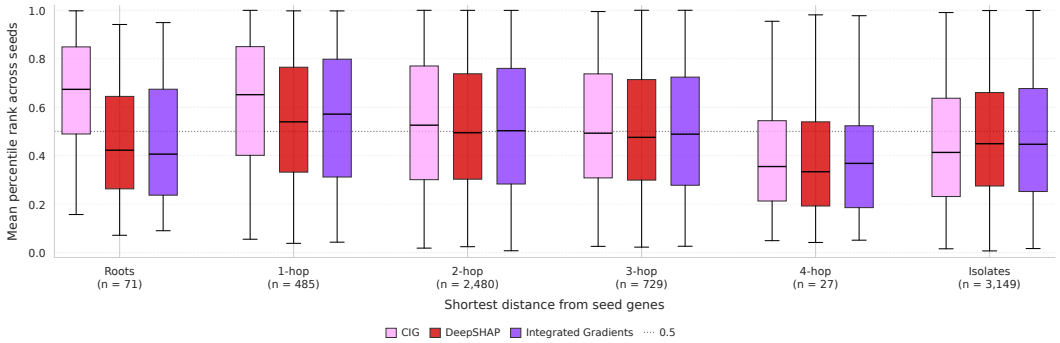


Figure I.1: Distribution of attribution ranks for genes grouped by their hop distance from the roots of the OmniPath prior graph. Each boxplot represents the distribution of ranks for genes within a specific hop-distance group, with the median rank indicated by a horizontal line within each box.

Table I.1: Median standard deviations of percentile rank across seeds (OmniPath); chance $= 0.278$, perfectly stable $= 0.0$.

	Roots	1-hop	2-hop	3-hop	4-hop	Isolates
CIG	0.1220	0.1085	0.1126	0.1197	0.1445	0.0959
DeepSHAP	0.1312	0.1194	0.1292	0.1335	0.1598	0.1428
Integrated gradients	0.0980	0.0891	0.0956	0.0987	0.1065	0.1017

Table I.2: Median standard deviations of percentile rank across seeds (SCENIC); chance $= 0.278$, perfectly stable $= 0.0$.

	Roots	1-hop	2-hop	3-hop	4-hop	5-hop	Isolates
CIG	0.1057	0.1026	0.1007	0.1041	0.1145	0.0872	0.1095
DeepSHAP	0.1940	0.1326	0.1242	0.1295	0.1502	0.1323	0.1535
Integrated gradients	0.1309	0.0974	0.0929	0.0969	0.1085	0.0535	0.1026

I.2 TOP-k ENRICHMENT OF STIMULATION-RESPONSIVE SCENIC REGULONS

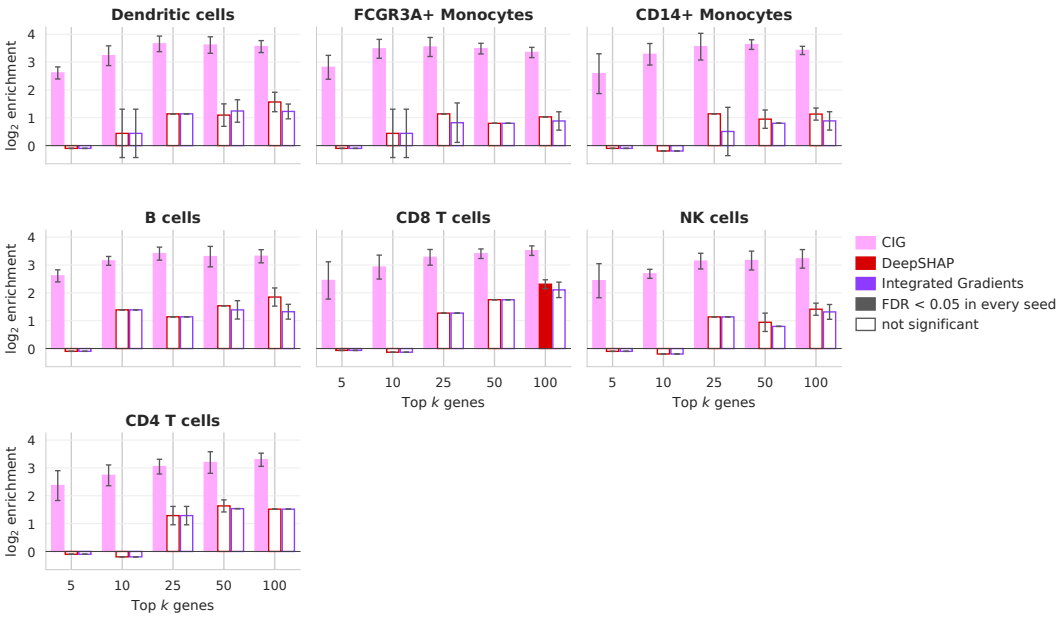


Figure I.2: Top-k enrichment of stimulation-responsive SCENIC regulons. The z-score of the hypergeometric test statistic is shown for each method, cell type, and top-k cutoff, averaged across five random seeds. Higher values indicate that responsive TFs are over-represented among the top-k genes. Error bars denote the standard deviation across seeds. P-values were BH-corrected across cell types and k. Only significant enrichment bars are shaded.

I.3 RECOVERY OF PSEUDOBULK DEGS

As an orthogonal readout, we asked how well each attribution ranking recovers the per-cell-type pseudobulk DEGs. Within each cell type, genes were ranked by their *absolute attribution difference* $\delta_g = |\bar{R}_g^{\text{stim}} - \bar{R}_g^{\text{ctrl}}|$, where $\bar{R}_g^{\text{stim}}$ and $\bar{R}_g^{\text{ctrl}}$ denote the attribution of gene g averaged over the stimulated and over the control cells of that cell type. Then, for each cell type, the rankings were scored against that cell type's DESeq2 (Love et al., 2014) DEG set using precision, recall, F1, normalized recovery AUC, and normalized discounted cumulative gain (NDCG; the DCG divided by that of the ideal ranking) graded by the absolute Wald statistic, at several fixed cutoffs and at a cutoff equal to the number of DEGs. The normalized recovery AUC is the area under the curve of recovered DEGs against rank depth, rescaled so that a random ranking scores 0 and the optimal ranking scores 1. At $k = n_{\text{DEGs}}$, precision, recall, and F1 coincide by construction, so only F1 is reported at that cutoff. We also measured the Spearman correlation between δ_g and the absolute DESeq2 Wald statistic over the entire gene panel. Every quantity was computed independently for all five seeds and is reported as mean $\pm$ s.d. over them. Megakaryocytes contributed no significant DEGs and are therefore absent from the recovery metrics.

The whole-panel correlation is stable seed-to-seed. No method tracks the DE statistic closely (Table I.3), so none of these methods can be considered equivalent to a differential expression test. CIG is the least DE-correlated of the three in all seven cell types. Over the full DEG set, scored at a cutoff equal to the number of DEGs (Table I.4), CIG has the lowest F1 in six of the seven cell types and the lowest recovery AUC in six of the seven; the exceptions are dendritic cells, where CIG and integrated gradients are tied on F1, and CD4 T cells, where CIG's recovery AUC exceeds that of DeepSHAP. CIG's NDCG is below that of both other methods in every cell type. Apart from dendritic cells and the CIG–DeepSHAP recovery-AUC comparison in CD4 T cells, these gaps are larger than the seed-to-seed spread.

The methods separate most sharply at the top of the ranking, and in the same direction in every cell type. Among their first ten genes, DeepSHAP and integrated gradients reach 94–100% precision and

an NDCG of 64–91, against 34–52% and 22–32 for CIG (Table I.5). The gap then narrows steadily with depth (Tables I.6–I.9). CIG's precision is also non-monotone in k. It is lower at $k = 10$ than at any cutoff up to $k = 100$, peaks at $k = 100$ in every cell type except NK cells (where it peaks at $k = 50$), and only then declines, whereas DeepSHAP and integrated gradients are at or near their maximum at $k = 10$ and decline thereafter. The genes CIG places in its first few dozen positions are therefore less likely to be DEGs than the ones it ranks immediately below them.

Taken together, these results are consistent with the following behavior. DeepSHAP and integrated gradients rank the downstream transcriptional response genes first, which lifts their precision and NDCG at shallow cutoffs and gives them a marginally closer alignment with the DE statistic overall. CIG instead prioritizes the upstream regulators driving that response, genes that are rarely strong DEGs themselves.

Table I.3: Whole-panel agreement between attribution and differential expression. For each cell type and method, the Spearman correlation between the absolute attribution difference δ_g (stimulated vs. control) and the absolute DESeq2 Wald statistic is computed over the full measured gene panel. Values are the mean and standard deviation across five random seeds.

Cell type	Method	Mean Spearman ρ	S.d. across seeds
B cells	CIG	0.3098	0.0072
B cells	DeepSHAP	0.3724	0.0015
B cells	Integrated gradients	0.3465	0.0039
$CD14^+$ monocytes	CIG	0.3530	0.0343
$CD14^+$ monocytes	DeepSHAP	0.3945	0.0215
$CD14^+$ monocytes	Integrated gradients	0.4119	0.0395
CD4 T cells	CIG	0.2678	0.0069
CD4 T cells	DeepSHAP	0.2926	0.0076
CD4 T cells	Integrated gradients	0.3273	0.0048
CD8 T cells	CIG	0.2253	0.0113
CD8 T cells	DeepSHAP	0.2800	0.0079
CD8 T cells	Integrated gradients	0.2429	0.0070
Dendritic cells	CIG	0.1524	0.0106
Dendritic cells	DeepSHAP	0.1571	0.0080
Dendritic cells	Integrated gradients	0.1537	0.0041
$FCGR3A^+$ monocytes	CIG	0.3918	0.0275
$FCGR3A^+$ monocytes	DeepSHAP	0.4358	0.0167
$FCGR3A^+$ monocytes	Integrated gradients	0.4378	0.0279
NK cells	CIG	0.1548	0.0065
NK cells	DeepSHAP	0.1566	0.0065
NK cells	Integrated gradients	0.1658	0.0033

Table I.4: Recovery of pseudobulk DEGs at a cutoff equal to the number of DEGs ($k = n_{\text{DEGs}}$). For each cell type, genes are ranked by the absolute attribution difference δ_g and scored against that cell type's DESeq2 DEG set. The recovery AUC is normalized so that a random ranking scores 0 and the optimal ranking scores 1; NDCG is graded by the absolute Wald statistic. At this cutoff, precision, recall, and F1 coincide by construction, so only F1 is reported. All values are percentages, given as mean $\pm$ s.d. across five random seeds. Megakaryocytes are omitted as they contributed no significant DEGs.

Cell type	CIG	DeepSHAP	Integrated gradients
F1-score at $k = n_{\text{DEGs}}$			
B cells	47.12 ± 0.48	57.56 ± 0.44	54.78 ± 0.40
CD14$^+$ monocytes	43.70 ± 1.56	46.41 ± 0.83	46.78 ± 1.72
CD4 T cells	35.56 ± 1.07	38.10 ± 1.06	39.39 ± 1.06
CD8 T cells	49.20 ± 1.90	67.15 ± 1.55	65.55 ± 0.80
Dendritic cells	35.55 ± 0.83	36.19 ± 0.93	35.52 ± 0.78
FCGR3A$^+$ monocytes	48.38 ± 2.90	55.04 ± 0.99	53.76 ± 2.63
NK cells	34.12 ± 1.33	39.88 ± 0.93	38.70 ± 1.16
Normalized recovery AUC			
B cells	74.45 ± 1.41	79.57 ± 1.35	79.78 ± 0.59
CD14$^+$ monocytes	30.18 ± 2.23	33.19 ± 1.72	35.05 ± 2.93
CD4 T cells	44.18 ± 3.56	41.47 ± 2.68	49.85 ± 3.12
CD8 T cells	89.53 ± 1.35	93.90 ± 1.66	93.40 ± 2.17
Dendritic cells	21.55 ± 1.76	21.79 ± 1.01	21.58 ± 0.89
FCGR3A$^+$ monocytes	63.78 ± 2.97	67.52 ± 1.70	68.07 ± 3.12
NK cells	44.05 ± 2.49	49.29 ± 1.93	46.90 ± 1.61
NDCG at $k = n_{\text{DEGs}}$			
B cells	69.64 ± 2.43	89.46 ± 0.65	87.22 ± 0.79
CD14$^+$ monocytes	73.01 ± 2.69	79.38 ± 1.21	79.43 ± 2.31
CD4 T cells	68.77 ± 1.59	84.35 ± 0.67	83.27 ± 0.47
CD8 T cells	59.80 ± 2.83	88.49 ± 0.95	87.78 ± 1.10
Dendritic cells	69.02 ± 0.45	76.70 ± 0.23	75.74 ± 0.78
FCGR3A$^+$ monocytes	70.86 ± 3.30	83.89 ± 1.69	81.87 ± 4.07
NK cells	67.25 ± 2.23	82.89 ± 1.08	80.73 ± 1.44

Table I.5: Recovery of pseudobulk DEGs per cell type at a top-k cutoff of $k = 10$. Genes are ranked by the absolute attribution difference δ_g and scored against that cell type's DESeq2 DEG set. NDCG is graded by the absolute Wald statistic. All values are percentages, given as mean $\pm$ s.d. across five random seeds. Megakaryocytes are omitted as they contributed no significant DEGs.

Cell type	CIG	DeepSHAP	Integrated gradients
Precision			
B cells	52.00 ± 13.04	100.00 ± 0.00	100.00 ± 0.00
CD14$^+$ monocytes	48.00 ± 8.37	100.00 ± 0.00	100.00 ± 0.00
CD4 T cells	34.00 ± 13.42	96.00 ± 5.48	96.00 ± 5.48
CD8 T cells	40.00 ± 12.25	100.00 ± 0.00	100.00 ± 0.00
Dendritic cells	48.00 ± 13.04	100.00 ± 0.00	96.00 ± 5.48
FCGR3A$^+$ monocytes	44.00 ± 16.73	96.00 ± 5.48	94.00 ± 8.94
NK cells	42.00 ± 13.04	100.00 ± 0.00	100.00 ± 0.00
Recall			
B cells	1.09 ± 0.27	2.11 ± 0.00	2.11 ± 0.00
CD14$^+$ monocytes	0.24 ± 0.04	0.51 ± 0.00	0.51 ± 0.00
CD4 T cells	0.47 ± 0.19	1.32 ± 0.08	1.32 ± 0.08
CD8 T cells	2.92 ± 0.89	7.30 ± 0.00	7.30 ± 0.00
Dendritic cells	0.30 ± 0.08	0.63 ± 0.00	0.61 ± 0.03
FCGR3A$^+$ monocytes	0.54 ± 0.21	1.18 ± 0.07	1.16 ± 0.11
NK cells	0.85 ± 0.26	2.03 ± 0.00	2.03 ± 0.00
F1-score			
B cells	2.14 ± 0.54	4.12 ± 0.00	4.12 ± 0.00
CD14$^+$ monocytes	0.48 ± 0.08	1.01 ± 0.00	1.01 ± 0.00
CD4 T cells	0.93 ± 0.37	2.61 ± 0.15	2.61 ± 0.15
CD8 T cells	5.44 ± 1.67	13.61 ± 0.00	13.61 ± 0.00
Dendritic cells	0.60 ± 0.16	1.25 ± 0.00	1.20 ± 0.07
FCGR3A$^+$ monocytes	1.07 ± 0.41	2.34 ± 0.13	2.29 ± 0.22
NK cells	1.67 ± 0.52	3.98 ± 0.00	3.98 ± 0.00
NDCG			
B cells	22.47 ± 6.81	80.36 ± 1.79	78.90 ± 2.41
CD14$^+$ monocytes	26.74 ± 7.25	65.73 ± 1.52	64.32 ± 2.23
CD4 T cells	31.53 ± 8.07	90.60 ± 1.90	90.09 ± 1.79
CD8 T cells	21.78 ± 5.40	85.18 ± 2.20	86.21 ± 3.02
Dendritic cells	23.25 ± 1.82	89.19 ± 3.35	86.88 ± 4.71
FCGR3A$^+$ monocytes	22.01 ± 4.67	80.28 ± 3.64	75.57 ± 10.85
NK cells	23.80 ± 7.67	82.10 ± 3.39	79.07 ± 5.60

Table I.6: Recovery of pseudobulk DEGs per cell type at a top-k cutoff of $k = 25$. Metrics and conventions are as in Table I.5.

Cell type	CIG	DeepSHAP	Integrated gradients
Precision			
B cells	56.80 ± 12.46	96.00 ± 0.00	96.00 ± 0.00
CD14$^+$ monocytes	60.80 ± 11.45	100.00 ± 0.00	97.60 ± 3.58
CD4 T cells	49.60 ± 8.29	96.00 ± 0.00	96.00 ± 0.00
CD8 T cells	48.80 ± 5.93	97.60 ± 2.19	94.40 ± 3.58
Dendritic cells	50.40 ± 8.29	98.40 ± 2.19	93.60 ± 6.07
FCGR3A$^+$ monocytes	52.00 ± 9.38	96.80 ± 1.79	89.60 ± 10.43
NK cells	55.20 ± 11.10	100.00 ± 0.00	94.40 ± 4.56
Recall			
B cells	2.99 ± 0.66	5.05 ± 0.00	5.05 ± 0.00
CD14$^+$ monocytes	0.77 ± 0.15	1.27 ± 0.00	1.24 ± 0.05
CD4 T cells	1.71 ± 0.29	3.31 ± 0.00	3.31 ± 0.00
CD8 T cells	8.91 ± 1.08	17.81 ± 0.40	17.23 ± 0.65
Dendritic cells	0.79 ± 0.13	1.55 ± 0.03	1.48 ± 0.10
FCGR3A$^+$ monocytes	1.60 ± 0.29	2.98 ± 0.06	2.76 ± 0.32
NK cells	2.80 ± 0.56	5.07 ± 0.00	4.79 ± 0.23
F1-score			
B cells	5.68 ± 1.25	9.60 ± 0.00	9.60 ± 0.00
CD14$^+$ monocytes	1.52 ± 0.29	2.51 ± 0.00	2.44 ± 0.09
CD4 T cells	3.31 ± 0.55	6.40 ± 0.00	6.40 ± 0.00
CD8 T cells	15.06 ± 1.83	30.12 ± 0.68	29.14 ± 1.10
Dendritic cells	1.57 ± 0.26	3.06 ± 0.07	2.91 ± 0.19
FCGR3A$^+$ monocytes	3.11 ± 0.56	5.79 ± 0.11	5.36 ± 0.62
NK cells	5.33 ± 1.07	9.65 ± 0.00	9.11 ± 0.44
NDCG			
B cells	36.41 ± 8.36	84.56 ± 1.06	84.29 ± 1.12
CD14$^+$ monocytes	37.72 ± 8.14	72.09 ± 2.33	68.99 ± 4.00
CD4 T cells	42.29 ± 7.64	86.70 ± 0.44	86.42 ± 1.35
CD8 T cells	38.39 ± 5.40	88.33 ± 2.70	87.86 ± 1.61
Dendritic cells	29.72 ± 0.93	88.77 ± 1.86	86.01 ± 4.65
FCGR3A$^+$ monocytes	33.65 ± 6.57	81.62 ± 3.73	75.42 ± 11.52
NK cells	40.16 ± 7.37	86.75 ± 1.38	82.96 ± 2.21

Table I.7: Recovery of pseudobulk DEGs per cell type at a top-k cutoff of $k = 50$. Metrics and conventions are as in Table I.5.

Cell type	CIG	DeepSHAP	Integrated gradients
Precision			
B cells	62.80 ± 4.82	98.00 ± 0.00	95.60 ± 2.19
CD14$^+$ monocytes	70.00 ± 8.60	98.80 ± 1.10	95.20 ± 3.35
CD4 T cells	59.60 ± 5.55	96.00 ± 0.00	96.00 ± 0.00
CD8 T cells	54.40 ± 9.84	94.80 ± 1.79	93.20 ± 2.28
Dendritic cells	61.20 ± 4.60	97.20 ± 2.28	94.40 ± 5.37
FCGR3A$^+$ monocytes	57.20 ± 7.01	95.60 ± 3.58	89.60 ± 11.70
NK cells	57.60 ± 3.29	97.60 ± 0.89	92.00 ± 2.45
Recall			
B cells	6.61 ± 0.51	10.32 ± 0.00	10.06 ± 0.23
CD14$^+$ monocytes	1.78 ± 0.22	2.51 ± 0.03	2.42 ± 0.08
CD4 T cells	4.11 ± 0.38	6.62 ± 0.00	6.62 ± 0.00
CD8 T cells	19.85 ± 3.59	34.60 ± 0.65	34.01 ± 0.83
Dendritic cells	1.93 ± 0.15	3.07 ± 0.07	2.98 ± 0.17
FCGR3A$^+$ monocytes	3.53 ± 0.43	5.89 ± 0.22	5.52 ± 0.72
NK cells	5.84 ± 0.33	9.90 ± 0.09	9.33 ± 0.25
F1-score			
B cells	11.96 ± 0.92	18.67 ± 0.00	18.21 ± 0.42
CD14$^+$ monocytes	3.46 ± 0.43	4.89 ± 0.05	4.71 ± 0.17
CD4 T cells	7.69 ± 0.72	12.39 ± 0.00	12.39 ± 0.00
CD8 T cells	29.09 ± 5.26	50.70 ± 0.96	49.84 ± 1.22
Dendritic cells	3.74 ± 0.28	5.94 ± 0.14	5.77 ± 0.33
FCGR3A$^+$ monocytes	6.64 ± 0.81	11.10 ± 0.42	10.41 ± 1.36
NK cells	10.61 ± 0.61	17.97 ± 0.16	16.94 ± 0.45
NDCG			
B cells	50.37 ± 5.57	88.95 ± 0.93	87.63 ± 0.86
CD14$^+$ monocytes	47.75 ± 8.26	76.77 ± 0.87	73.66 ± 3.80
CD4 T cells	49.28 ± 5.62	88.08 ± 0.44	87.25 ± 0.49
CD8 T cells	49.77 ± 6.69	88.84 ± 1.67	88.82 ± 1.46
Dendritic cells	43.99 ± 2.23	89.36 ± 1.69	87.23 ± 4.76
FCGR3A$^+$ monocytes	43.06 ± 5.69	82.92 ± 3.14	78.24 ± 10.25
NK cells	50.69 ± 4.57	91.35 ± 1.33	87.85 ± 2.83

Table I.8: Recovery of pseudobulk DEGs per cell type at a top-k cutoff of $k = 100$. Metrics and conventions are as in Table I.5.

Cell type	CIG	DeepSHAP	Integrated gradients
Precision			
B cells	64.00 ± 1.87	93.80 ± 2.68	90.40 ± 2.97
CD14$^+$ monocytes	74.60 ± 6.23	95.80 ± 1.30	92.60 ± 4.39
CD4 T cells	61.20 ± 4.60	91.00 ± 0.71	89.20 ± 1.79
CD8 T cells	57.00 ± 3.61	78.40 ± 2.30	75.40 ± 1.82
Dendritic cells	67.40 ± 2.19	92.40 ± 3.65	89.40 ± 4.16
FCGR3A$^+$ monocytes	64.40 ± 7.23	93.80 ± 4.97	87.60 ± 9.76
NK cells	56.60 ± 3.21	85.20 ± 1.79	80.40 ± 1.95
Recall			
B cells	13.47 ± 0.39	19.75 ± 0.56	19.03 ± 0.62
CD14$^+$ monocytes	3.78 ± 0.32	4.86 ± 0.07	4.70 ± 0.22
CD4 T cells	8.44 ± 0.64	12.55 ± 0.10	12.30 ± 0.25
CD8 T cells	41.61 ± 2.63	57.23 ± 1.68	55.04 ± 1.33
Dendritic cells	4.25 ± 0.14	5.83 ± 0.23	5.64 ± 0.26
FCGR3A$^+$ monocytes	7.94 ± 0.89	11.57 ± 0.61	10.80 ± 1.20
NK cells	11.48 ± 0.65	17.28 ± 0.36	16.31 ± 0.40
F1-score			
B cells	22.26 ± 0.65	32.63 ± 0.93	31.44 ± 1.03
CD14$^+$ monocytes	7.20 ± 0.60	9.25 ± 0.13	8.94 ± 0.42
CD4 T cells	14.84 ± 1.12	22.06 ± 0.17	21.62 ± 0.43
CD8 T cells	48.10 ± 3.04	66.16 ± 1.94	63.63 ± 1.53
Dendritic cells	8.00 ± 0.26	10.97 ± 0.43	10.61 ± 0.49
FCGR3A$^+$ monocytes	14.14 ± 1.59	20.59 ± 1.09	19.23 ± 2.14
NK cells	19.09 ± 1.08	28.74 ± 0.60	27.12 ± 0.66
NDCG			
B cells	59.58 ± 4.15	90.46 ± 0.74	89.41 ± 0.72
CD14$^+$ monocytes	56.36 ± 6.72	78.89 ± 1.23	77.29 ± 3.39
CD4 T cells	56.25 ± 3.86	89.83 ± 0.30	89.33 ± 0.92
CD8 T cells	58.47 ± 3.97	89.23 ± 1.19	88.31 ± 1.07
Dendritic cells	55.23 ± 1.11	88.55 ± 1.66	86.61 ± 3.17
FCGR3A$^+$ monocytes	54.48 ± 5.25	85.40 ± 2.68	80.86 ± 7.76
NK cells	60.68 ± 4.01	90.10 ± 1.11	87.18 ± 2.34

Table I.9: Recovery of pseudobulk DEGs per cell type at a top-k cutoff of $k = 250$. Metrics and conventions are as in Table I.5.

Cell type	CIG	DeepSHAP	Integrated gradients
Precision			
B cells	56.48 ± 1.43	76.64 ± 1.15	73.28 ± 1.37
CD14$^+$ monocytes	71.28 ± 4.21	84.88 ± 1.80	81.60 ± 3.97
CD4 T cells	52.64 ± 1.00	66.72 ± 1.58	66.00 ± 1.52
CD8 T cells	37.60 ± 1.96	45.84 ± 0.83	45.28 ± 0.87
Dendritic cells	64.08 ± 2.05	79.28 ± 1.18	75.76 ± 2.31
FCGR3A$^+$ monocytes	63.12 ± 6.52	82.24 ± 3.28	75.60 ± 6.80
NK cells	47.36 ± 1.00	59.84 ± 0.61	57.52 ± 1.11
Recall			
B cells	29.73 ± 0.75	40.34 ± 0.61	38.57 ± 0.72
CD14$^+$ monocytes	9.04 ± 0.53	10.77 ± 0.23	10.35 ± 0.50
CD4 T cells	18.15 ± 0.35	23.01 ± 0.55	22.76 ± 0.53
CD8 T cells	68.61 ± 3.58	83.65 ± 1.51	82.63 ± 1.58
Dendritic cells	10.11 ± 0.32	12.50 ± 0.19	11.95 ± 0.36
FCGR3A$^+$ monocytes	19.46 ± 2.01	25.35 ± 1.01	23.30 ± 2.10
NK cells	24.02 ± 0.51	30.34 ± 0.31	29.17 ± 0.56
F1-score			
B cells	38.95 ± 0.98	52.86 ± 0.79	50.54 ± 0.94
CD14$^+$ monocytes	16.05 ± 0.95	19.11 ± 0.40	18.37 ± 0.89
CD4 T cells	26.99 ± 0.51	34.22 ± 0.81	33.85 ± 0.78
CD8 T cells	48.58 ± 2.53	59.22 ± 1.07	58.50 ± 1.12
Dendritic cells	17.46 ± 0.56	21.60 ± 0.32	20.64 ± 0.63
FCGR3A$^+$ monocytes	29.75 ± 3.07	38.76 ± 1.55	35.63 ± 3.20
NK cells	31.87 ± 0.68	40.27 ± 0.41	38.71 ± 0.75
NDCG			
B cells	66.91 ± 3.33	91.38 ± 0.70	89.66 ± 0.85
CD14$^+$ monocytes	63.63 ± 4.69	80.91 ± 1.36	78.33 ± 3.34
CD4 T cells	65.69 ± 2.19	89.48 ± 0.47	88.29 ± 1.11
CD8 T cells	63.11 ± 3.55	87.34 ± 0.72	86.84 ± 1.32
Dendritic cells	64.07 ± 1.61	86.08 ± 0.57	83.70 ± 2.08
FCGR3A$^+$ monocytes	64.74 ± 4.81	85.70 ± 2.40	81.60 ± 6.17
NK cells	66.59 ± 3.00	87.85 ± 1.04	85.28 ± 1.88

I.4 Over-representation analysis of top-ranked genes

Table I.10: Over-representation analysis of the top-100 genes unique to CIG in dendritic cells. Terms are gene sets from the MSigDB Hallmark 2020 collection, and the measured gene panel was used as the background set. The ten most significant terms are listed. P-values come from a hypergeometric test and are Benjamini–Hochberg adjusted across the collection; the combined score is $\ln(p)$ multiplied by the z-score of the deviation from the expected rank. The last column gives the genes shared between the top-100 set and the term. Adjusted P-values in bold are significant at $P < 0.01$.

Term	Adjusted P-value	Odds Ratio	Combined Score	Genes
TNF-alpha Signaling via NF-kB	**0.0003**	14.1062	156.8329	JUN, KLF6, IRF1, MYC, FOS, NFE2L2
Hypoxia	**0.0003**	17.0927	178.5641	JUN, KLF6, FOS, NR3C1, ETS1
Interferon Gamma Response	**0.0003**	12.1222	124.9307	IRF4, IRF1, STAT2, IRF2, IRF8, HIF1A
G2-M Checkpoint	**0.0041**	11.7922	86.3737	MYC, CTCF, E2F4, HIF1A
IL-2/STAT5 Signaling	**0.0041**	10.6567	74.2016	KLF6, IRF4, MYC, IRF8
Inflammatory Response	**0.0041**	10.5641	73.2283	KLF6, IRF1, MYC, HIF1A
UV Response Dn	**0.0041**	17.5550	120.9222	MYC, NR3C1, RUNX1
Wnt-beta Catenin Signaling	**0.0047**	44.2436	291.5354	MYC, CTNNB1
IL-6/JAK/STAT3 Signaling	**0.0047**	15.1570	98.3114	JUN, IRF1, STAT2
Allograft Rejection	**0.0048**	8.9954	57.2430	IRF4, IRF8, ETS1, HIF1A

Table I.11: Over-representation analysis of the top-100 genes unique to CIG in CD14$^+$ monocytes, against the MSigDB Hallmark 2020 gene sets with the measured gene panel as background. Columns are as in Table I.10. Adjusted P-values in bold are significant at $P < 0.01$.

Term	Adjusted P-value	Odds Ratio	Combined Score	Genes
TNF-alpha Signaling via NF-kB	**0.0001**	17.6432	214.7754	CEBPB, JUN, IRF1, MYC, FOS, NFE2L2
Interferon Gamma Response	**0.0002**	15.1617	172.0823	STAT1, IRF1, STAT2, IRF2, IRF8, HIF1A
IL-6/JAK/STAT3 Signaling	**0.0003**	26.2874	267.9484	JUN, STAT1, IRF1, STAT2
Hypoxia	**0.0016**	15.7940	131.4392	JUN, FOS, NR3C1, ETS1
G2-M Checkpoint	**0.0018**	14.4212	115.2408	MYC, CTCF, E2F4, HIF1A
UV Response Dn	**0.0027**	21.2632	157.1334	MYC, NR3C1, RUNX1
Allograft Rejection	**0.0033**	11.0008	77.1754	STAT1, IRF8, ETS1, HIF1A
Interferon Alpha Response	**0.0088**	12.3993	73.2617	IRF1, STAT2, IRF2
Complement	0.0162	9.1004	46.1886	CEBPB, IRF1, IRF2
Inflammatory Response	0.0162	9.1004	46.1886	IRF1, MYC, HIF1A

Table I.12: Over-representation analysis of the top-100 genes unique to CIG in FCGR3A$^+$ monocytes, against the MSigDB Hallmark 2020 gene sets with the measured gene panel as background. Columns are as in Table I.10. Adjusted P-values in bold are significant at $P < 0.01$.

Term	Adjusted P-value	Odds Ratio	Combined Score	Genes
IL-6/JAK/STAT3 Signaling	**0.0021**	21.4953	204.3132	JUN, STAT1, STAT2, STAT3
TNF-alpha Signaling via NF-kB	**0.0028**	11.1166	94.6276	CEBPB, JUN, MYC, FOS, NFE2L2
Interferon Gamma Response	**0.0036**	9.5621	75.0479	STAT1, STAT2, IRF2, STAT3, IRF8
G2-M Checkpoint	**0.0046**	11.7922	86.3737	MYC, CTCF, E2F4, ATF5
UV Response Dn	**0.0057**	17.5550	120.9222	MYC, NR3C1, RUNX1
Wnt-beta Catenin Signaling	**0.0064**	44.2436	291.5354	MYC, CTNNB1
Hypoxia	0.0237	9.1681	47.0092	JUN, FOS, NR3C1
Cholesterol Homeostasis	0.0321	15.4910	72.6594	CTNNB1, ATF5
Allograft Rejection	0.0472	6.4055	26.8338	SPI1, STAT1, IRF8
Estrogen Response Early	0.0585	9.8520	38.1050	MYC, FOS

Table I.13: Over-representation analysis of the top-100 genes unique to integrated gradients in dendritic cells, against the MSigDB Hallmark 2020 gene sets with the measured gene panel as background; the most significant terms are listed. Columns are as in Table I.10. Adjusted P-values in bold are significant at $P < 0.01$.

Term	Adjusted P-value	Odds Ratio	Combined Score	Genes
Allograft Rejection	**0.0049**	97.8273	656.8822	CD74, CCL22
Interferon Alpha Response	0.0685	38.4775	125.4228	CD74
Inflammatory Response	0.0685	28.4083	84.3200	CCL22
Interferon Gamma Response	0.0729	19.6686	51.4961	CD74

Table I.14: Over-representation analysis of the top-100 genes unique to integrated gradients in CD14$^+$ monocytes, against the MSigDB Hallmark 2020 gene sets with the measured gene panel as background; the most significant terms are listed. Columns are as in Table I.10. Adjusted P-values in bold are significant at $P < 0.01$.

Term	Adjusted P-value	Odds Ratio	Combined Score	Genes
Complement	0.0587	22.9076	117.2823	PLAUR, PFN1
Allograft Rejection	0.0587	19.5540	94.2961	SOCS1, B2M
Interferon Gamma Response	0.0587	15.8199	70.0418	SOCS1, B2M
Apical Surface	0.0587	96.1389	417.1983	PLAUR
Cholesterol Homeostasis	0.1113	30.2456	98.3633	PLAUR
Epithelial Mesenchymal Transition	0.1113	22.9467	68.5890	PLAUR
Myogenesis	0.1113	22.4935	66.8073	PDE4DIP
Androgen Response	0.1113	22.0577	65.1024	B2M
IL-6/JAK/STAT3 Signaling	0.1113	18.7787	52.5521	SOCS1
Apical Junction	0.1113	18.4731	51.4093	PFN1

Table I.15: Over-representation analysis of the top-100 genes unique to integrated gradients in FCGR3A$^+$ monocytes, against the MSigDB Hallmark 2020 gene sets with the measured gene panel as background; the most significant terms are listed. Columns are as in Table I.10. Adjusted P-values in bold are significant at $P < 0.01$.

Term	Adjusted P-value	Odds Ratio	Combined Score	Genes
Epithelial Mesenchymal Transition	0.1406	27.5400	86.4653	GLIPR1
Interferon Alpha Response	0.1406	15.3843	39.7786	PARP9
Apoptosis	0.1406	12.7626	30.7555	CD69
Inflammatory Response	0.1406	11.3583	26.1335	CD69
TNF-alpha Signaling via NF-kB	0.1406	9.1087	19.0955	CD69
Interferon Gamma Response	0.1406	7.8640	15.4290	CD69

Table I.16: Over-representation analysis of the top-100 genes unique to DeepSHAP in dendritic cells, against the MSigDB Hallmark 2020 gene sets with the measured gene panel as background; the most significant terms are listed. Columns are as in Table I.10. Adjusted P-values in bold are significant at $P < 0.01$.

Term	Adjusted P-value	Odds Ratio	Combined Score	Genes
Interferon Alpha Response	**0.0000**	236.2069	2761.8748	UBA7, PSME2, NCOA7
Interferon Gamma Response	**0.0090**	39.5673	222.7963	CXCL9, PSME2
IL-6/JAK/STAT3 Signaling	0.0588	37.5738	125.6833	CXCL9
Inflammatory Response	0.0788	18.9361	50.9210	CXCL9
Allograft Rejection	0.0788	16.1833	41.1132	CXCL9

Table I.17: Over-representation analysis of the top-100 genes unique to DeepSHAP in CD14$^+$ monocytes, against the MSigDB Hallmark 2020 gene sets with the measured gene panel as background; the most significant terms are listed. Columns are as in Table I.10. Adjusted P-values in bold are significant at $P < 0.01$.

Term	Adjusted P-value	Odds Ratio	Combined Score	Genes
Interferon Alpha Response	0.0335	31.1182	177.3636	DHX58, PARP12
Interferon Gamma Response	0.0597	15.8199	70.0418	DHX58, PARP12
Cholesterol Homeostasis	0.0992	30.2456	98.3633	CD9
Coagulation	0.0992	26.7093	83.6984	CD9
IL-6/JAK/STAT3 Signaling	0.0992	18.7787	52.5521	CD9
Protein Secretion	0.0992	16.8284	45.3449	NAPA
Estrogen Response Late	0.0992	16.3429	43.5836	CD9
Inflammatory Response	0.1448	9.4639	20.3969	OSM
mTORC1 Signaling	0.1557	7.6950	15.1238	CD9
Myc Targets V1	0.1698	6.2182	11.0261	RPL6

Table I.18: Over-representation analysis of the top-100 genes unique to DeepSHAP in FCGR3A$^+$ monocytes, against the MSigDB Hallmark 2020 gene sets with the measured gene panel as background; the most significant terms are listed. Columns are as in Table I.10. Adjusted P-values in bold are significant at $P < 0.01$.

Term	Adjusted P-value	Odds Ratio	Combined Score	Genes
Interferon Alpha Response	**0.0002**	78.7126	795.1581	BST2, IFITM2, IFI27
Interferon Gamma Response	**0.0007**	39.7941	324.4256	BST2, IFITM2, IFI27
Bile Acid Metabolism	0.0554	36.3000	123.5496	CH25H
Inflammatory Response	0.1252	11.3583	26.1335	BST2
Myc Targets V1	0.1474	7.4630	14.2879	RPL6

J ISOLATING THE VALUE OF CIG'S FORMULATION BEYOND GRAPH ACCESS

A possible concern is that CIG's advantage is simply a readout of having access to the causal graph, not of correctly using it. We address this on Dyngen with a test of instance-level specificity that a fixed, shared graph cannot explain on its own. Because the Dyngen causal graph is identical across cells, a graph-driven attribution method would be expected to produce similar TF importance across cells. In contrast, cells undergoing different state transitions should be explained by different sets of transition-specific TFs. For each transition, we therefore restrict the analysis to cells undergoing that transition and compare the attribution ranks of its known causal TFs against those of all other genes.

CIG consistently prioritizes the causal TFs for all six transitions, with significant enrichment in every case (Figure J.1). DeepSHAP shows significant prioritization in two of the six transitions, while integrated gradients does not show significant enrichment in any transition. The transition-specific differences in CIG attributions indicate that its explanations depend on the cells being explained, rather than simply reproducing the static causal graph.

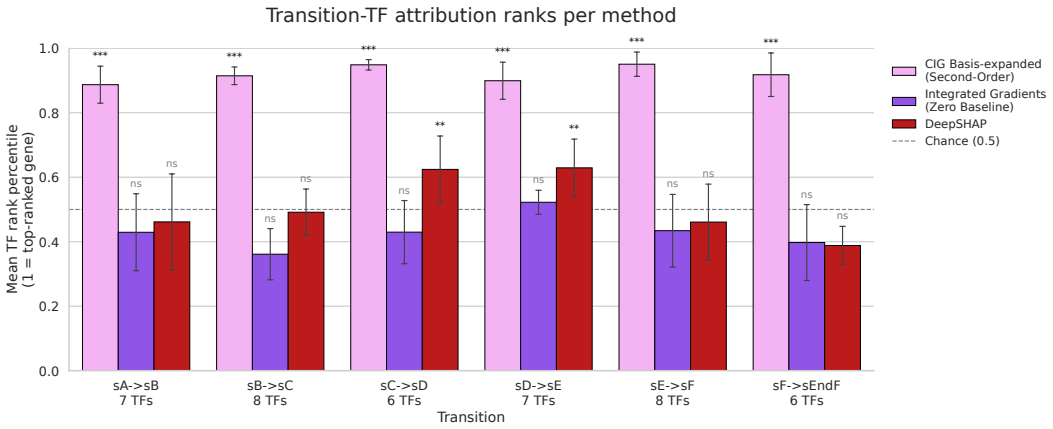


Figure J.1: Transition-TF attribution ranks by method. Bars show the mean TF rank across five seeds, with error bars indicating the standard deviation. Significance is assessed using a gene-label permutation test, Stouffer-combined across seeds and Benjamini–Hochberg adjusted. Significance levels: $***$: $q < 0.001$; $**$: $q < 0.01$; $*$: $q < 0.05$; ns, not significant.

K ROBUSTNESS TO CAUSAL GRAPH MISSPECIFICATION

We test how CIG's attribution quality degrades when the supplied causal graph is imperfect, using three corruption types applied at 25%, 50%, and 75% of edges: removing true edges, adding spurious edges (cycle-safe, never removing a true edge), and reversing the direction of existing true edges. We apply this to both the synthetic SCM/CATE benchmark (Section 4.1) and the Dyngen gene-regulatory network (Section 4.2).

On the synthetic benchmark, we measure the Spearman correlation between CIG's attribution and the true CATE ranking (root-cause features), averaged over 5 seeds, across feature counts 100–1000. Adding spurious edges or reversing edge direction has no effect when the corrupted graph is fed to the oracle SCM mechanisms directly, since the oracle's forward pass only ever reads its hardcoded true-parent columns; for these two corruption types we instead re-fit a learned surrogate for each perturbed graph, matching how the method is used in practice. Edge removal is tested directly on the oracle mechanisms, as in the main benchmark. Figures K.1–K.3 show results across the full feature sweep, one figure per corruption type.

On Dyngen, whose network separates *backbone genes* (the core transcription-factor cascade driving the simulated differentiation trajectory) from *housekeeping (HK) genes* (background genes uninvolved in any transition), we apply the same three corruption types to each community separately and report the normalized TF-recovery AUC metric used for the main Dyngen results (uncorrupted baseline: 97.76 ± 3.18). Table K.1 reports backbone/HK AUC pairs.

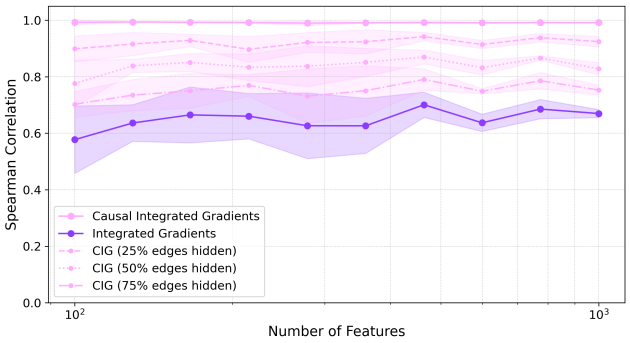


Figure K.1: Spearman correlation with ground-truth CATE under edge removal (root-cause features), as a function of the number of features, at 25/50/75% of edges removed. Shaded areas denote the standard deviation over 5 seeds.

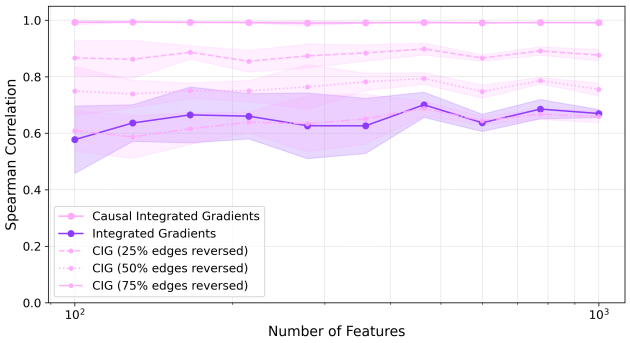


Figure K.2: Spearman correlation with ground-truth CATE under edge reversal (root-cause features), as a function of the number of features, at 25/50/75% of edges reversed. Shaded areas denote the standard deviation over 5 seeds.

Table K.1: Normalized TF-recovery AUC under graph corruption on Dyngen, reported separately for the backbone and housekeeping genes (HK). Values are mean $\pm$ standard deviation over five seeds; uncorrupted baseline 97.76 ± 3.18.

Corruption	Community	25%	50%	75%
Removed	Backbone	93.31 ± 3.14	89.50 ± 2.44	70.41 ± 4.78
	HK	98.00 ± 1.12	97.93 ± 1.01	97.93 ± 1.08
Reversed	Backbone	78.58 ± 0.74	61.80 ± 2.73	50.68 ± 4.22
	HK	96.68 ± 1.74	96.38 ± 1.91	96.89 ± 1.89
Added (spurious)	Backbone	96.38 ± 1.54	95.44 ± 1.29	93.85 ± 2.74
	HK	97.99 ± 1.10	97.91 ± 1.04	97.81 ± 1.09

Across both datasets, the same pattern holds: removing or reversing edges degrades the performance of CIG substantially, while adding spurious edges barely decreases it, and in Dyngen, corrupting the housekeeping genes does not degrade the recovery of the causal TFs regardless of corruption type, while removing or reversing backbone edges degrades recovery increasingly with severity. Our hypothesis is that a spurious added edge does not correspond to any real dependency in the data-generating process; the surrogate has no true signal to associate with that edge, so its weight converges toward zero during training, making the gradient through it negligible even though the edge is present. Reversing or removing an edge, by contrast, is a genuine loss of information that the surrogate cannot recover, which is why both corruptions degrade recovery substantially. Together, both experiments show CIG's advantage is not solely inherited from a perfectly correct prior graph:

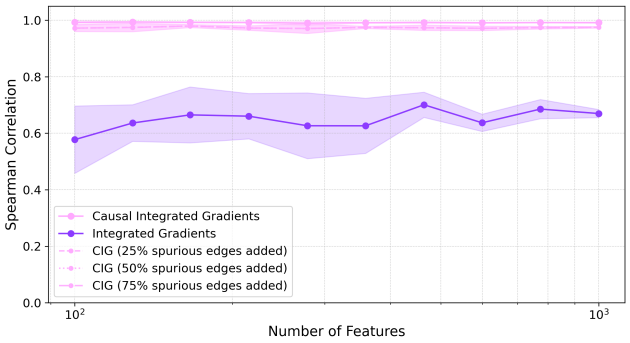


Figure K.3: Spearman correlation with ground-truth CATE under spurious edge addition (root-cause features), as a function of the number of features, at 25/50/75% of edges added. Shaded areas denote the standard deviation over 5 seeds.

it degrades under genuine information loss (removal, reversal) but tolerates redundant or causally irrelevant structural error (spurious edges; corruption confined to non-causal genes), performing about as well as with a perfect graph.

L Residual-dependence diagnostics under non-additive mechanisms

Our surrogate models each feature's mechanism as additive across parents (Section 3.3); when the true mechanism is non-additive, this can misallocate attribution between the direct and indirect terms (Section 5). We diagnose and quantify this bias directly, using the synthetic SCM/CATE benchmark (Section 4.1), restricted to nodes whose mechanism is either the explicit two-parent interaction term $w_1x_1 + w_2x_2 + w_1w_2x_1x_2$ or a matched linear control (eligibility criteria below). Since ground truth is known for every node, we compare directly rather than inferring statistically.

A diagnostic based on the residual $h_j = x_j - \hat{s}_j(\text{parents})$ conflates the systematic mechanism misfit with injected exogenous noise, making it an inherently noisy signal of the bias itself. Instead, we isolate the same systematic misallocation without this contamination by comparing CIG's indirect-effect attribution computed with the oracle mechanism against the same attribution computed with the learned surrogate: both quantities are deterministic functions of the parents, so no noise term enters at all. We test two scenarios: the interacting variable present as a graph node ("in-graph"), and the interacting variable removed from the graph entirely ("omitted-variable").

A node is eligible if its true mechanism matches the target type (interaction, or a matched linear control) and it is read directly by the downstream model, so that its attribution is nonzero; the linear control is further restricted to nodes with at least two parents (in-graph) or exactly two parents (omitted-variable), matching interaction's structural requirement of two active parents. The omitted-variable setting additionally restricts *both* types to exactly two total parents, so that removing one parent's edge deletes an equal share of each node's explanatory input regardless of mechanism type. Eligible nodes are found by searching randomly generated graphs per setting, not fixed in advance, and pooled across 5 seeds (in-graph) and 15 seeds (omitted-variable).

For each eligible node, we compute the discrepancy between the oracle and surrogate indirect effects across sampled instances, and derive two statistics from that same discrepancy. The indirect effect is a path integral from a zero baseline, so along a linear path the source feature's own value enters via $dx_i/d\alpha = x_i$; the discrepancy is therefore expected to scale with the product of the two parents, x_ix_k, rather than either parent alone (confirmed empirically: correlating against the co-parent alone gives $r = -0.16$, $p = 0.12$; against the product, $r = 0.83$, $p \approx 7 \times 10^{-27}$, on the same data). We report the Pearson correlation r of the discrepancy against x_ix_k, and its relative error, $\text{RMSE}(\text{discrepancy})/\text{RMSE}(\text{oracle indirect effect})$. Because both statistics are computed from the

same discrepancy per node, n is identical for both columns within a row.[1] Table L.1 reports the mean of $|r|$, the fraction of nodes with $p < 0.05$, the median relative error, and the fraction of nodes with relative error exceeding 50%.

Table L.1: CIG's oracle-vs-surrogate indirect-effect discrepancy, by setting. Correlation is against the product of the node's two parents; the correlation and relative-error statistics are computed from the same per-node discrepancy, so n is shared within each row.

Setting	Type	n	mean $\|r\|$	% sig.	Med. rel. err.	% >50% off
In-graph	Interaction	31	0.574	93.5%	0.286	6.5%
In-graph	Linear (control)	33	0.086	18.2%	0.288	0.0%
Omitted-variable	Interaction	42	0.264	73.8%	0.739	69.0%
Omitted-variable	Linear (control)	48	0.085	18.8%	0.789	81.3%

The discrepancy correlates far more strongly with the interaction term than with the linear control in both settings (3–7$\times$ the mean $|r|$, 4–5$\times$ the significance rate), confirming the bias is systematic rather than incidental: the indirect effect is predictably wrong wherever the mechanism is non-additive. The relative-error magnitude, by contrast, does not separate the two groups as cleanly: in the in-graph setting, interaction and linear are essentially identical (0.286 vs. 0.288), since overall surrogate fit quality there is dominated by factors common to both mechanism types (finite training, the downstream model's own regularization) rather than by the interaction term specifically; in the omitted-variable setting, both groups show substantial error because removing a parent's edge deletes its entire linear contribution regardless of mechanism type, which dominates the error budget for both. We therefore treat the correlation test as the primary evidence that the misallocation is systematic and tied to the interaction structure, and report relative error only to establish that indirect-effect errors of practically meaningful size do occur — not as an independent confirmation that interaction specifically causes larger errors than an equivalent linear mechanism. Despite these errors, the learned surrogate still achieves close to oracle-level Spearman correlation with the true CATE on the main benchmark (Section 4.1).

[1] p-values are uncorrected across nodes and computed on non-independent samples (two per node); they characterize consistency across nodes, not a formal hypothesis test.