DAG Learning from Zero-Inflated Count Data Using Continuous Optimization

Noriaki Sato NORIAKIS@IMS.U-TOKYO.AC.JP
The Institute of Medical Science, The University of Tokyo

Marco Scutari SCUTARI@BNLEARN.COM
Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA)

Shuichi Kawano SKAWANO@MATH.KYUSHU-U.AC.JP
Faculty of Mathematics, Kyushu University

Rui Yamaguchi R.YAMAGUCHI@AICHI-CC.JP
Division of Cancer System Biology, Aichi Cancer Center Research Institute
Division of Cancer Informatics, Nagoya University Graduate School of Medicine

Seiya Imoto IMOTO@HGC.JP
The Institute of Medical Science, The University of Tokyo

Editors: Bijan Mazaheri and Niels Richard Hansen

Abstract

We address network structure learning from zero-inflated count data by casting each node as a zero-inflated generalized linear model and optimizing a smooth, score-based objective under a directed acyclic graph constraint. Our Zero-Inflated Continuous Optimization (ZICO) approach uses node-wise likelihoods with canonical links and enforces acyclicity through a differentiable surrogate constraint combined with sparsity regularization. ZICO achieves superior performance with faster runtimes on simulated data. It also performs comparably to or better than common algorithms for reverse engineering gene regulatory networks. ZICO is fully vectorized and mini-batched, enabling learning on larger variable sets with practical runtimes in a wide range of domains.

Keywords: Directed Acyclic Graph, Continuous Optimization, Gene Regulatory Networks

1. Introduction

Learning causal structural models from observational data is a central problem in statistics and machine learning, with applications ranging from genomics and medicine to economics and social sciences (Pearl, 2021). Such models represent causal relationships by a directed acyclic graph (DAG), where edges encode direct causal influences and the joint distribution factorizes into node-wise conditionals.

The literature has developed three major families of causal discovery algorithms. *Constraint-based* algorithms learn DAGs by testing the conditional independencies they imply and orienting their edges using Meek's rules, as in the PC algorithm (Colombo et al., 2012). Later, FCI, RFCI, and penalized FCI (Pal et al., 2025) also addressed latent confounding and selection bias, as summarized in Zhu et al. (2024). *Score-based* algorithms search over candidate graphs and select structures that optimize a goodness-of-fit score, often with greedy search. GES (Chickering, 2002) is a widely used example, with extensions to interventions (Hauser and Bühlmann, 2012) and latent confounding (Claassen and Bucur, 2022), as is DirectLiNGAM (Shimizu et al., 2011). *Hybrid* algorithms combine both approaches, pruning the DAG space prior to score-based selection to improve

© 2026 N. Sato, M. Scutari, S. Kawano, R. Yamaguchi & S. Imoto.

scalability and robustness. Examples in this family include MMHC (Tsamardinos et al., 2006) and frameworks such as GFCI (Ogarrio et al., 2016), which combines score-based search with FCI-style reasoning to handle potential latent confounding. All these algorithms are combinatorial in nature, iteratively performing independence tests or greedy score-based selection, which limits their scalability to high-dimensional settings.

More recently, algorithms such as NOTEARS (Zheng et al., 2018; Nazaret et al., 2024), GOLEM (Ng et al., 2020), and DAGMA (Bello et al., 2022) recast score-based causal discovery into a differentiable optimization task constrained by smooth acyclicity surrogates. Thus, they achieve scalability through vectorized computation and can be solved by standard numerical optimizers including regularization. However, they have been developed primarily for continuous or Gaussian settings, and extending them to discrete count data requires careful model specifications and numerical stabilization.

Zero-inflated count data arise frequently in real-world applications, including transcriptomics (e.g., single-cell RNA-seq) and microbiome studies (Jiang et al., 2022). Excess zeros may reflect multiple mechanisms: structural zeros, sampling zeros arising from limited depth under a count process, and technical zeros such as dropout or detection failures. Standard Poisson or negative binomial (NB) models are misspecified for such data, potentially biasing causal discovery by inducing spurious dependencies driven by the zero-generating process rather than the underlying biological interactions. This motivates explicitly modeling the zero-generating process (within the data-generating process) with causal discovery methods specific to zero-inflated count data. Park and Raskutti (2015) proposed a greedy search algorithm based on overdispersion, while Choi et al. (2020) used zero-inflated Poisson distributions. Choi and Ni (2023) then developed a state-of-the-art model, ZiG-DAG, using the generalized hypergeometric distribution and greedy search. ZiG-DAG provides robust performance on simulated data and enables recovery of biologically relevant gene regulatory networks (GRN) from single-cell transcriptomics (SCT) data. Yu et al. (2023) proposed ZiDAG building on hurdle graphical models (McDavid et al., 2019).

While these approaches are well-suited to learning DAGs from zero-inflated count data, the computational cost of underlying greedy searches can be substantial. In such cases, continuous optimization offers much better scalability. In this work, we develop ZICO (Zero-Inflated Continuous Optimization), a score-based differentiable DAG learning framework tailored to zero-inflated count data. ZICO optimizes a zero-inflated Poisson (ZIP) or zero-inflated negative binomial (ZINB) node likelihood under sparsity regularization and a differentiable acyclicity constraint (based on DAGMA's) using mini-batched, vectorized training for scalability. It also estimates separate, process-specific weighted adjacency matrices for the zero-inflation and for the count-mean processes, enabling downstream interpretation of whether a putative parent affects the probability of extra zeros, the expected count level, or both. We empirically evaluate ZICO on synthetic zero-inflated data sets and simulated GRN data, demonstrating improved accuracy and favorable runtime compared to baseline methods. In addition, a real-world SCT data analysis shows that the algorithm can recover literature-validated networks. We emphasize that ZICO's contribution is primarily methodological and practical, addressing the numerical challenges of bringing zero-inflated count likelihoods into scalable differentiable DAG learning. We do not claim new identifiability results beyond those in Choi and Ni (2023).

2. Methods

Preliminaries Bayesian networks are a class of graphical models defined over a DAG $G = (V, E)$ with vertex set $V = \{1, \ldots, d\}$ and edge set $E \subseteq V \times V$ (Koller and Friedman, 2009). Each vertex $j \in V$ corresponds to a random variable X_j and $Pa(j)$ denotes its parents. Bayesian networks encode the conditional independencies among nodes, leading to the probability factorization

$$p(X_1, \ldots, X_d) = \prod_{j=1}^{d} p\left(X_j \mid X_{\mathrm{Pa}(j)}\right).$$

Problem Setup and Assumptions We address the DAG learning problem for zero-inflated count data by formulating the zero-inflated structural equation model underlying ZICO. Each variable $X_j, j = 1, 2, \ldots, d$ in the observational count data $X \in \mathbb{N}_0^{n \times d}$ follows a ZINB distribution conditional on its parents,

$$X_j \mid X_{\mathrm{Pa0}(j)}, X_{\mathrm{Pa1}(j)} \sim \mathrm{ZINB}\left(\pi_j(X_{\mathrm{Pa0}(j)}), \mu_j(X_{\mathrm{Pa1}(j)}), r_j\right),$$

with link functions:

$$\mathrm{logit}\left(\pi_j(x)\right) = \gamma_j + x^\top w_j^{(0)} \qquad \text{and} \qquad \log\left(\mu_j(x)\right) = \delta_j + x^\top w_j^{(1)},$$

where $w_j^{(0)}, w_j^{(1)} \in \mathbb{R}^d$ are the weights for node j in the zero and count components; γ_j, δ_j are intercepts; $r_j > 0$ is a dispersion parameter in the negative binomial (NB) distribution; $\pi_j(X_{\mathrm{Pa0}(j)})$ is the zero-inflation probability for variable j; and $\mu_j(X_{\mathrm{Pa1}(j)})$ is the conditional mean of the NB count component for variable j. The resulting node-wise log-likelihood naturally separates zeros due to a structural process from the sampling zeros arising under an NB count model. Its expression for sample i and node j is

$$\ell_{ij} = \begin{cases} \log\Big(\pi_{ij} + (1 - \pi_{ij})\, p_{ij}^{r_j}\Big) & (x_{ij} = 0) \\ \log(1 - \pi_{ij}) + \log\Gamma(x_{ij} + r_j) - \log\Gamma(r_j) - \log\Gamma(x_{ij} + 1) & \\ \quad + r_j \log p_{ij} + x_{ij} \log(1 - p_{ij}), & (x_{ij} > 0) \end{cases},$$

where $\pi_{ij} = \mathrm{sigmoid}\left(\gamma_j + x_i^\top w_j^{(0)}\right)$, $p_{ij} = \frac{r_j}{r_j + \mu_{ij}}$, $\Gamma(\cdot)$ is the Gamma function, and $\mu_{ij} = \exp\left(\delta_j + x_i^\top w_j^{(1)}\right)$.

We additionally consider the ZIP model, in which each node follows a ZIP conditional distribution with the same link functions for the zero and count components:

$$X_j \mid X_{\mathrm{Pa0}(j)},\, X_{\mathrm{Pa1}(j)} \sim \mathrm{ZIP}\big(\pi_j(X_{\mathrm{Pa0}(j)}),\, \mu_j(X_{\mathrm{Pa1}(j)})\big),$$

where $\pi_j(X_{\mathrm{Pa0}(j)})$ is the zero-inflation probability for variable j and $\mu_j(X_{\mathrm{Pa1}(j)})$ is the conditional mean of the Poisson (or count) component for variable j. The node-wise ZIP log-likelihood for sample i and node j is

$$\ell_{ij} = \begin{cases} \log\Big(\pi_{ij} + (1 - \pi_{ij}) e^{-\mu_{ij}}\Big) & (x_{ij} = 0) \\ \log(1 - \pi_{ij}) + x_{ij} \log \mu_{ij} & \\ \quad - \mu_{ij} - \log\Gamma(x_{ij} + 1) & (x_{ij} > 0) \end{cases}.$$

We minimize the average negative log-likelihood (NLL) for both models. We compute it fully in the log domain in both ZIP and ZINB models using the log-gamma, log-sum-exp, and softplus link functions, ensuring finite scores even when $\pi \to 1$ or $\mu \to 0$ for numerical stability (Cui and Wang, 2023).

Let $W_0 = [\, w_1^{(0)}, w_2^{(0)}, \ldots, w_d^{(0)} \,]$ and $W_1 = [\, w_1^{(1)}, w_2^{(1)}, \ldots, w_d^{(1)} \,]$ denote the coefficient matrices for the zero and count components, respectively. We enforce acyclicity on W_0 and W_1 separately. The acyclicity is imposed via the log-determinant-based function of the M-matrix transformation of the weighted adjacency matrix, introduced in DAGMA, as follows:

$$h^s_{\text{ldet}}(W_k) = -\log\det(sI - W_k \circ W_k) + d\log s, \quad (k = 0, 1)$$

where s indicates the log-determinant parameter and $\circ$ denotes the Hadamard product. Here, the log-determinant parameter s controls the M-matrix transformation. Following DAGMA, we require $sI - (W_k \circ W_k)$ to lie in the domain of the log-determinant M-matrix characterization. In practice, this requires s to be larger than the spectral radius of $(W_k \circ W_k)$. Smaller s yields a stronger acyclicity penalty but can make optimization numerically stiffer, whereas larger s provides a smoother penalty. Unless otherwise stated, we fix $s = 1$ for all experiments and do not schedule it.

3. Objective Function and Continuous Optimization

Let $\Theta = \{W_0, W_1, \gamma, \delta, r\}$. To complete our ZICO proposal, we define the following objective with respect to Θ:

$$\min_{\Theta}\ \mu \left(-\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{d} \ell_{ij} \ + \ \lambda_{\text{group}} \sum_{j=1}^{d}\sum_{\substack{k=1\\k\neq j}}^{d} \left\| \big((W_0)_{kj},\, (W_1)_{kj}\big) \right\|_2 \right) \\ + \ h^s_{\text{ldet}}(W_0) \ + \ h^s_{\text{ldet}}(W_1) \ + \ \lambda_{\text{align}}\, \mathcal{R}_{\text{align}}(W_0, W_1)\,. \quad (1)$$

The alignment regularizer $\lambda_{\text{align}}\, \mathcal{R}_{\text{align}}(W_0, W_1)$ couples W_0 and W_1 while allowing flexibility in the assumed relationship between the two adjacency matrices. Depending on the application, one may favor similar supports with either matching or opposite signs. As an example, one may choose the squared Frobenius norm, though other choices are possible depending on the assumed relationship between the two components.

Following the central-path approach used in DAGMA, the multiplier $\mu > 0$ is gradually decreased during training at specified epoch intervals using a decay parameter α.

An optional group-lasso penalty over the off-diagonal $\big((W_0)_{kj},\, (W_1)_{kj}\big)$ pairs with nonnegative weight $\lambda_{\text{group}} \geq 0$ is applied with a cosine-annealed warm-up schedule. Specifically, the effective regularization at epoch t is

$$\lambda_{\text{eff}}(t) = \frac{\lambda_{\text{group}}}{2}\left(1 - \cos\big(\min\{1,\ t/\texttt{warm}\}\,\pi\big)\right),$$

where `warm` controls the length of the warm-up period. This schedule gradually increases the regularization strength from 0 to λ_{group}, dealing with early-training instability.

We use AdamW with gradient clipping (Loshchilov and Hutter, 2019) for solving (1) to learn DAGs using ZICO. To scale to large n, we compute the NLL on mini-batches $B \subseteq \{1, \ldots, n\}$ of size $|B|$:

$$\mathrm{NLL}_B = -\frac{1}{|B|} \sum_{i \in B} \sum_{j=1}^{d} \ell_{ij}.$$

By restricting training to the parameters in W_1 alone, the model naturally reduces to an NB or Poisson formulation (i.e., without the zero-inflation component). The optimization returns two weighted adjacency matrices (W_0, W_1) in the ZINB or ZIP case, and W_1 in the NB or Poisson case.

The default hyperparameter values used in each experiment, as well as which matrix was used for the performance evaluation, are summarized in Table S1.

Assumptions and scope of claims ZICO is an observational, likelihood-based score method requiring the following assumptions: (i) Acyclicity: the true causal structure is a DAG; (ii) Causal sufficiency: there are no hidden common causes among the measured variables (latent confounding is not modeled); (iii) IID sampling: samples are independent and identically distributed with no selection bias; (iv) Model specification: each node is well approximated by a ZIP, ZINB, Poisson, or NB regression on its parents. Under model misspecification, learned structures may be biased.

Identifiability The ZIP/ZINB models described above are particular cases of the generalized hypergeometric distribution used in ZiG-DAG. The form of the dependence between each node and its parents is also the same as in ZiG-DAG. Therefore, the identifiability proof from Choi and Ni (2023) applies to ZICO as well: individual DAGs are identifiable under conditions (i)–(iv), so any consistent causal discovery algorithm will identify the true DAG uniquely with probability converging to 1 as $n \to \infty$. Replacing the log-likelihood term with the Bayesian Information Criterion (BIC) in (1) is enough to ensure that, which is also noted in Choi and Ni (2023). Alternatively, we could ensure that the group-lasso penalty scales with n, similarly to BIC, to promote consistent model learning. Continuous optimization with DAGMA's acyclicity constraints also preserves causal identifiability (Bello et al., 2022).

In practice, zero inflation changes both the statistical and algorithmic properties of score-based DAG learning. Statistically, leaving it unmodeled makes the zero-generating process a latent confounder, violating (ii), and can mask conditional dependencies among positive counts or induce spurious edges as a result. Conversely, modeling the zero-generating process ensures that ZICO is correctly specified and satisfies (ii) as long as zeros arise from dropout events that are representable by the assumed ZIP or ZINB family. If this is not the case, likelihood misspecification degrades recovery, consistent with recent identifiability results and the conditions required to restore recovery by appropriately handling zero-valued observations (Dai et al., 2024). Algorithmically, we implemented a stable training objective that computes the likelihood fully in the log domain, uses warm-up scheduling for sparsity regularization, and optionally couples W_0 and W_1 to reduce degenerate solutions in early training.

Tuning protocol We adopt a tuning protocol to ensure stable optimization under the log-determinant acyclicity constraint. We first tune the learning rate so that the acyclicity value remains finite and non-negative throughout training. Next, we tune the balance between data fit and the constraint via the μ schedule, avoiding overly rapid decay that can cause the constraint to dominate and destabilize optimization as dimensionality increases. For higher-dimensional settings (larger number of

variables), we further employ a two-stage optimization strategy: we first train with the constraint disabled (`ignore_logdet` parameter in the PyTorch code) to obtain a sparse, well-scaled initialization driven by the likelihood and sparsity penalties, and then enable the constraint to enforce acyclicity during refinement. Finally, we tune sparsity-inducing regularization (group penalty) using a warm-up schedule to keep weights small early in training, and apply gradient clipping to mitigate occasional large updates.

4. Experiments

4.1. Simulated Data

We generated synthetic data sets with D nodes and a defined sample size. First, the true DAGs were sampled under the Erdős-Rényi (ER) and Barabási-Albert (BA) models, with an edge probability of 0.25 for ER and expected number of three edges per node in BA (Csárdi and Nepusz, 2006, Python igraph library).

Once the network structure was determined, we sampled the model parameters. For edges present in the DAG, coefficients for the zero-inflation component were randomly drawn from a $U(0.5, 2)$. Coefficients for the mean component were drawn from a $U(-2, -0.5)$. In addition, intercept terms for the zero-inflation component were sampled from a $N(1.5, 0.2)$ and those for the mean component were sampled from a $N(1.5, 0.2)$. The dispersion parameter of the NB distribution was fixed at 5.0 for all nodes.

Data were simulated using logic sampling to preserve the structure implied by the DAG (Koller and Friedman, 2009). For each node, the probability of generating a structural zero was determined from the values of its parents and the corresponding zero-inflation parameters. When a nonzero value was generated, it was sampled from a negative binomial distribution with parameters determined by the mean component of the model. This procedure produced data sets that exhibited both zero inflation and overdispersion, consistent with the topology of the specified DAG structure. For each of $D = 20, 30, 50, 100$ ($N = 500$ in $D = 20, 30, 50$ and $N = 1000$ in $D = 100$), we generated 10 replicate data sets and learned the network structure using the methods described below. The estimated graphs were then compared with the true underlying graph using Structural Hamming Distance (SHD; Tsamardinos et al., 2006), Structural Intervention Distance (SID; Peters and Bühlmann, 2015), True Positive Rate (TPR), and False Discovery Rate (FDR). SHD and SID were computed with the bnlearn R package (Scutari, 2010). For evaluation, we thresholded the absolute values of the learned W_1 entries to identify the DAG edges. We use W_1 because it encodes the conditional dependence structure of the count-mean mechanism that is the primary target of causal interpretation in our setting. Where applicable, the area under the precision-recall curve (AUPRC) was also calculated.

We compared ZICO with the GES (score-based), MMHC (hybrid) and DirectLiNGAM (score-based). As for algorithms tailored to zero-inflated data, we considered ZiDAG and ZiG-DAG (score-based, hyper-Poisson and NB distributions) as well as NOTEARS with a Poisson loss (continuous optimization). For GES, MMHC, DirectLiNGAM, and ZiDAG, $\log(x + 1)$-transformed count data was used as input for the structure learning. For the other algorithms, the raw simulated count data were used as the input. We used the implementation of GES in the pcalg R package (Kalisch et al., 2012), and those of DirectLiNGAM and MMHC in bnlearn. The equivalence class learned by GES was replaced by its consistent DAG extension before computing SHD and SID. In this experiment,

Algorithm	TPR	FDR	AUPRC	Time (s)	SHD	SID	Graph model
DirectLiNGAM	0.022 (0.024)	0.930 (0.062)	NA	78.886 (0.560)	150.8 (2.394)	2435.0 (55.102)	BA
GES	0.365 (0.057)	0.580 (0.056)	NA	**0.622 (0.096)**	125.0 (10.853)	2257.6 (135.318)	BA
MMHC	0.306 (0.034)	0.543 (0.044)	NA	11.112 (8.264)	129.4 (8.959)	2328.4 (82.141)	BA
NOTEARS	0.274 (0.076)	**0.229 (0.089)**	0.271 (0.078)	28.933 (6.401)	123.7 (7.514)	2393.7 (102.723)	BA
ZICO (NB)	**0.795 (0.044)**	0.503 (0.040)	0.680 (0.069)	112.892 (97.732)	137.3 (18.252)	**1195.1 (222.697)**	BA
ZICO (Poisson)	0.658 (0.062)	0.495 (0.049)	0.485 (0.098)	108.625 (100.263)	113.8 (10.602)	1644.6 (269.026)	BA
ZICO (ZINB)	0.790 (0.036)	0.253 (0.028)	**0.686 (0.053)**	117.467 (95.069)	**60.5 (7.075)**	1258.2 (174.839)	BA
ZICO (ZIP)	0.744 (0.039)	0.259 (0.028)	0.667 (0.054)	113.021 (97.264)	66.6 (8.449)	1400.9 (232.865)	BA
ZiDAG	0.050 (0.033)	0.535 (0.249)	NA	840.235 (152.673)	139.1 (3.573)	2453.2 (58.818)	BA
ZiG-DAG (HP)	0.402 (0.059)	0.511 (0.048)	NA	14530.830 (999.781)	102.3 (7.558)	2264.1 (115.928)	BA
ZiG-DAG (NB)	0.413 (0.041)	0.494 (0.043)	NA	2289.778 (179.211)	100.6 (6.381)	2237.9 (119.419)	BA

Table 1: Performance comparison ($D = 50$; BA). Values are averages over 10 replicates. The values inside parentheses are standard deviations. The best performing values are shown in bold. HP, hyper-Poisson; NB, negative binomial; ZICO: our proposal.

α and the first μ were set to 0.1 and 1, respectively. Weighted adjacency matrices were binarized at the absolute value threshold of 0.3.

The performance of various algorithms for recovering DAGs using the simulated zero-inflated data is summarized in Table 1. The results for $D = 100$ are summarized in Table S2. Our ZICO proposal demonstrated comparable or superior accuracy to other approaches, with faster execution times than ZiG-DAG and ZiDAG. The speed-up was most pronounced when the number of variables was large. As for structural accuracy, ZICO also yielded the lowest errors, achieved in the ZINB model with $D = 50$ and in the BA model, indicating close reconstruction of the true networks. As an additional check, we performed a threshold sensitivity analysis, and the results are shown in Figure S1. The runtime and peak memory usage are reported in Table S3. Also, the accuracy obtained using $log(x + 1)$-transformed input data is additionally reported in Table S4.

The superior performance of the methods based on ZINB and ZIP is expected, as these methods explicitly model the zero-inflated nature of the data. By aligning the statistical assumptions with the underlying data-generating process, they achieve higher accuracy compared with conventional approaches.

4.2. Sign and Norm-Comparison Experiment

To assess the sensitivity of ZICO to different data-generating processes, we evaluated the sign patterns between W_0 and W_1. For each graph type $g \in \{\mathrm{BA}, \mathrm{ER}\}$ and dimension $D = 20$, we generated a random DAG B as described previously, fixed $N = 500$ observations, and drew edge weights from uniform ranges under different sign configurations.

We experimented with four sign configurations:

$$(\mathrm{sign}(W_0), \mathrm{sign}(W_1)) \in \{(+,+),\ (-,-),\ (+,-),\ (-,+)\}\,.$$

Unless otherwise stated, $(+,-)$ denotes $W_0 > 0$ and $W_1 < 0$, and $(-,+)$ denotes the opposite. For example, in $(+,+)$, parents make zeros more likely, but conditional on being nonzero, counts are larger. This yields many zeros with occasionally large positive counts.

We also compared three ways of coupling W_0 and W_1: 1) alignment with the Frobenius norm, at $\lambda_{\mathrm{align}} \in \{0, 0.1, 1\}$; 2) alignment with an elementwise ℓ_1 norm, at the same λ_{align}; 3) a combined

Configuration	W_0low	W_0high	W_1low	W_1high	g	λ_{group}	λ_{align}	Norm	AUPRC
$(-,-)$	-2.0	-0.5	-2.0	-0.5	BA	0.010	0.0	none	0.8189 (0.0759)
$(-,-)$	-2.0	-0.5	-2.0	-0.5	ER	0.010	0.0	none	0.8177 (0.0618)
$(-,+)$	-2.0	-0.5	0.5	2.0	BA	0.001	0.0	NA	0.3452 (0.0725)
$(-,+)$	-2.0	-0.5	0.5	2.0	ER	0.000	0.0	NA	0.4672 (0.1266)
$(+,-)$	0.5	2.0	-2.0	-0.5	BA	0.010	0.1	ℓ_1	0.7129 (0.0840)
$(+,-)$	0.5	2.0	-2.0	-0.5	ER	0.010	0.1	ℓ_1	0.7724 (0.0452)
$(+,+)$	0.5	2.0	0.5	2.0	BA	0.010	0.1	ℓ_1	0.8498 (0.0764)
$(+,+)$	0.5	2.0	0.5	2.0	ER	0.010	0.1	ℓ_1	0.8716 (0.0323)

Table 2: Best ZICO parameter combinations for each (W_0, W_1, g) group maximizing AUPRC (W_0W_1). W_0low, W_0high, W_1low, W_1high indicate the lower and upper bounds of uniform distribution in W_0 and W_1. The values inside parentheses are standard deviations.

approach that aggregates the two matrices via elementwise ℓ_2 pooling $\left(\widetilde{W} = \sqrt{W_0^{\circ 2} + W_1^{\circ 2} + \varepsilon}\right)$ and enforces acyclicity on $\widetilde{W}$. Furthermore, we vary $\lambda_{\text{group}} \in \{0, 0.001, 0.01\}$. We evaluate couplings using the AUPRC for the combined W_0 and W_1, based on the edge ranking obtained from the weighted adjacency matrix. We generated five data replicates for each experimental setting.

The results of the experiments are summarized in Table 2. The optimal tuning parameter values for ZICO varied across the four sign configurations. Still, we found that the separate acyclicity plus light ℓ_1 alignment ($\lambda_{\text{align}} = 0.1$) when the W_0 is positive, yields the best AUPRC in this experimental setting. Heavy alignment and large group penalties consistently underperformed. Notably, the $(-,+)$ combination performed markedly worse than other choices across all experimental settings. We considered the reason to be a mismatch between data generation and likelihood, and the improvements there likely require changes beyond the model architecture or hyperparameters rather than stronger regularization. Also, different norms encourage different types of coupling, and can therefore lead to different inferred structures and empirical performance.

4.3. Simulations for different support for W_0 and W_1

To decouple patterns between the zero-inflation and count components and characterize the effect of the alignment penalty term, we partitioned the edge set of the DAG into two support masks with a user-specified overlap fraction ρ, yielding $\mathbf{M}_0$ for the zero-inflation mechanism and $\mathbf{M}_1$ for the mean model. Edge weights W_0 and W_1 were sampled on their respective supports, $\mathbf{M}_0$ and $\mathbf{M}_1$. As zero-inflation and count processes may not hold in all contexts, we varied $\rho \in \{0, 0.25, 0.5, 0.75, 1\}$ to simulate how the proposed algorithm performs under different alignment penalties and λ_{align}. We evaluated $g \in \{\text{BA}, \text{ER}\}$ and $(+,+)$ for generating data under the ZINB model in this experiment. We generated five replicates of the data for each condition.

Under $(+,+)$, the alignment strength directly applied to the chosen norm yields a ρ-dependent trade-off. When the overlap is low, a light ℓ_1 alignment is the best choice. In contrast, no alignment with the coupled acyclicity rose steadily with ρ and performed comparably to the alignment setting at high overlap in our experiments, indicating that when the common backbone between W_0 and W_1 is strong, forcing acyclicity to the coupled adjacency matrix can be beneficial, yielding comparable performance in the $(+,+)$ case under the BA model (Figure 1).

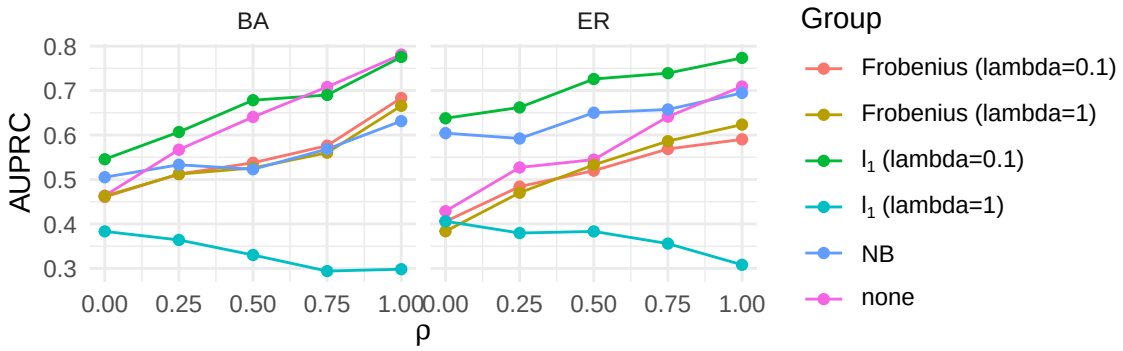

Figure 1: The AUPRC across multiple overlap levels ρ. At higher overlap, the no alignment setting achieves performance comparable to the alignment settings, whereas at lower overlap, introducing a light alignment term can improve performance. Note that AUPRC for W_1 is reported for NB model.

4.4. Simulated SCT Data

Bayesian networks have been widely applied to GRN inference from transcriptomics data obtained from microarray and RNA-seq technology (Imoto et al., 2003). We conducted an evaluation using scMultiSim, a state-of-the-art SCT data simulator that supports GRN simulations (Li et al., 2025).

We first generated random DAGs under the BA model with the expected number of three neighbours per node, and assigned effect sizes to each edge by sampling from a $U(1, 5)$. We then generated cell-count data with $N = 500$ cells, trained networks on them with ZICO, and evaluated the learned DAGs' accuracy. For comparison, we also inferred GRNs using GENIE3 (Huynh-Thu et al., 2010), GRNBoost2 (Moerman et al., 2019), LEAP (Specht and Li, 2017), SINCERITIES (Papili Gao et al., 2018), NOTEARS (Zheng et al., 2018). For GENIE3, GRNBoost2, and ZICO, we used the raw count data. For the other methods, a $\log(x + 1)$ transformation was applied before the inference. We used L2 loss in NOTEARS inference. Pseudo-time ordering, which is necessary for some algorithms, was calculated using the slingshot R package (Street et al., 2018). As most GRN inference methods do not explicitly produce a DAG, accuracy was evaluated using the AUPRC ratio, defined as the ratio of the AUPRC value to that obtained when using random edges.

Whether the unique molecular identifier count data are zero-inflated is a matter of debate (Cao et al., 2021). This can be modeled naturally by omitting the W_0 term in (1), thus reducing it to an NB or Poisson regression NLL. Therefore, in this experiment, we used NB, Poisson, ZINB, and ZIP models, and computed AUPRC using the absolute values of the W_1 coefficient matrix in the NB and Poisson models and the sum of the absolute values of W_0 and W_1 in the ZINB and ZIP models (union-type aggregation). This is to avoid discarding potentially meaningful signal from the zero component and to provide a single comparable ranking across models. We chose the alignment penalty to be the ℓ_1 norm with $\lambda_{\text{align}} = 0.1$, which performed best in earlier $(+, -)$ experiments. In SCT data, such a configuration is consistent with the empirical behavior of dropout stochastic detection failures, which are more likely at low true expression, so that higher dropout co-occurs with lower observed positive counts (Kharchenko et al., 2014).

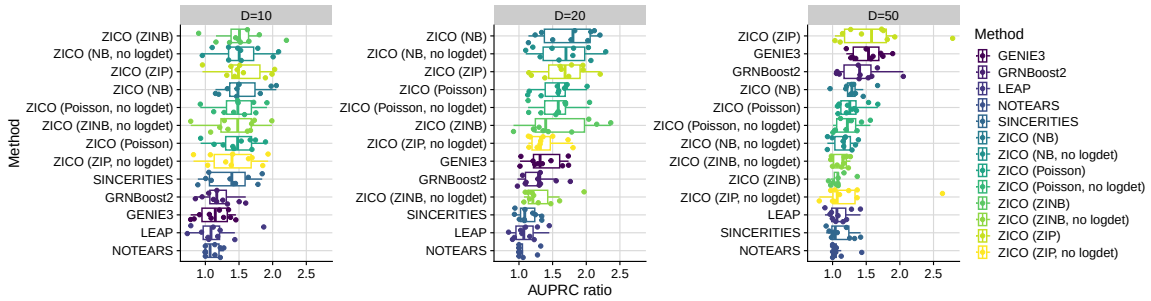

Figure 2: Performance in recovering transcriptomic regulatory relationships, assessed by AUPRC/AUPRC$_{\text{random}}$. Algorithms are ordered by the median values.

The results of reverse engineering GRNs using ZICO are shown in Figure 2 along with those from other commonly used directed GRN inference methods. Overall, the assessment using the AUPRC ratio relative to random prediction shows that the performance in recovering GRN is generally low, consistent with previous studies. ZICO achieved results comparable to or better than other commonly used methods for GRN inference. We hypothesize that the reason is that ZICO provides a close surrogate for the Poisson–Beta models used in the scMultiSim. DAG constraints, which are absent from other GRN inference algorithms, could also enhance performance; the ablation of the log-determinant term degraded performance in some settings.

In addition, we performed dropout simulation experiments on the clean count data. Following previous studies (e.g., Dibaeinia and Sinha, 2020), we introduced expression-dependent dropout into the clean data. We then evaluated ZICO with both ZINB/ZIP and NB/Poisson models on the resulting data sets using the same AUPRC-ratio metric.

The results are shown in Figure 3. The ZINB and ZIP approaches, which incorporate W_0, consistently achieved higher AUPRC ratios across all D values than the NB and Poisson models. This suggests that estimators that account for zero inflation are advantageous when estimating DAG from SCT data with dropout events.

4.5. Real-world SCT data set

We used a publicly available kidney cell atlas data set from Gene Expression Omnibus (GSE183276) (Lake et al., 2023). From this data set, proximal tubule (PT) cells were extracted based on the provided cell-type annotations, and the PT subset was downsampled to a fixed number of cells. We used the count data as the input features for network inference. We assembled a PT-relevant candidate gene panel, resulting in a final set of 24 genes. Network robustness was then evaluated by repeatedly fitting ZICO with the ZINB loss to bootstrap-resampled count matrices and calculating the stability of each directed edge across replicates (Figure 4).

The inferred DAG revealed several biologically interpretable modules. The HNF4A–HNF1A connection is consistent with a known transcription factor module associated with epithelial maturation and proximal-tubule gene regulation (Eeckhoute et al., 2004; Martovetsky et al., 2013). Also, a negative GATM–LCN2 relationship supports an opposition between a mature proximal-tubule metabolic state and an injury-associated transcriptional state (Nickolas et al., 2012). Notably, these edges are consistent with the relationships described in the established literature. While the di-

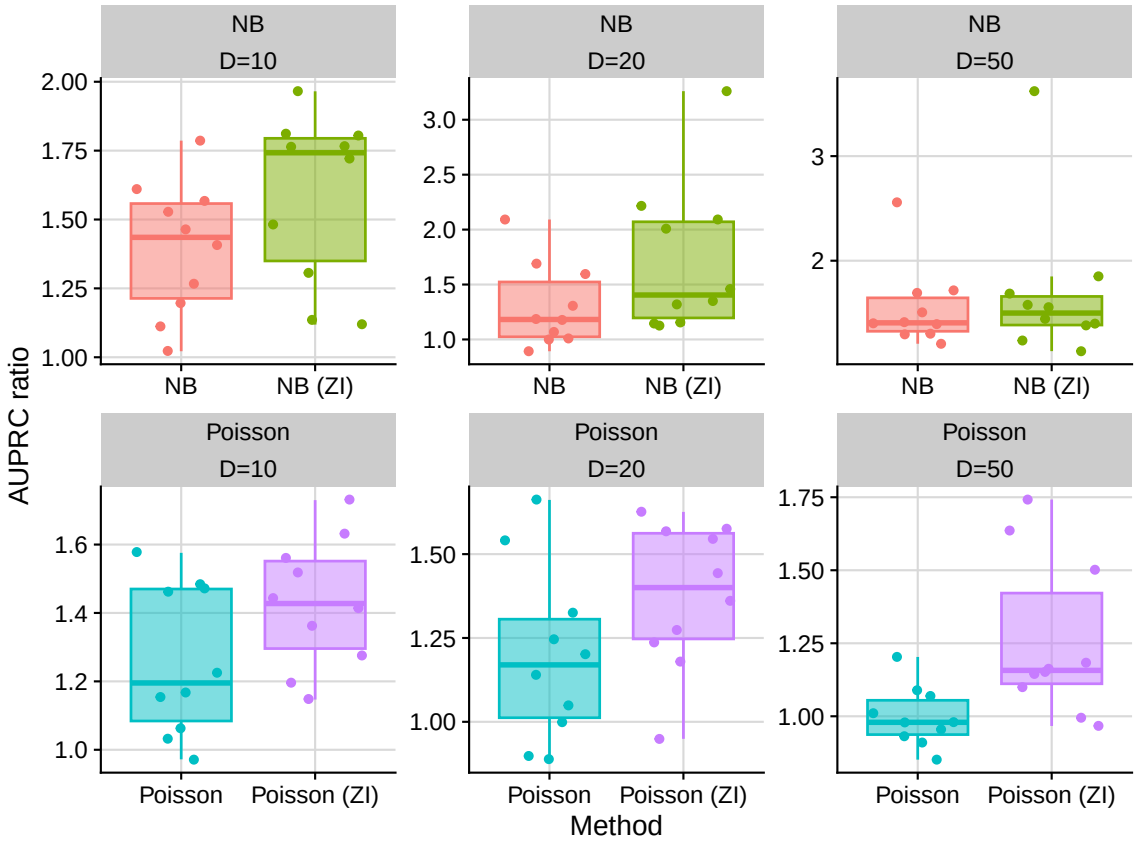

Figure 3: Performance of GRN recovery after applying dropout simulation to the count data. The boxplot comparing ZINB, ZIP, NB, and Poisson models assessed by the AUPRC ratio ($\mathrm{AUPRC}/\mathrm{AUPRC}_{\mathrm{random}}$) is shown.

rected edges provide a summary of conditional dependencies in the subset data, some directions may reflect co-expression rather than direct transcriptional regulation. Together, this PT-restricted, kidney-informed gene panel and the learned DAG using real-world data suggest a practical usage of the proposed model for evaluating GRNs.

4.6. Evaluation Environment

The proposed ZICO is implemented in PyTorch. All code is available from https://github.com/noriakis/ZICO. Training was carried out on the supercomputer SHIROKANE using PyTorch 2.6.0 (Paszke et al., 2019). The performance statistics were calculated using scikit-learn (Pedregosa et al., 2011). The supercomputing resource was provided by the Human Genome Centre, the Institute of Medical Science, the University of Tokyo.

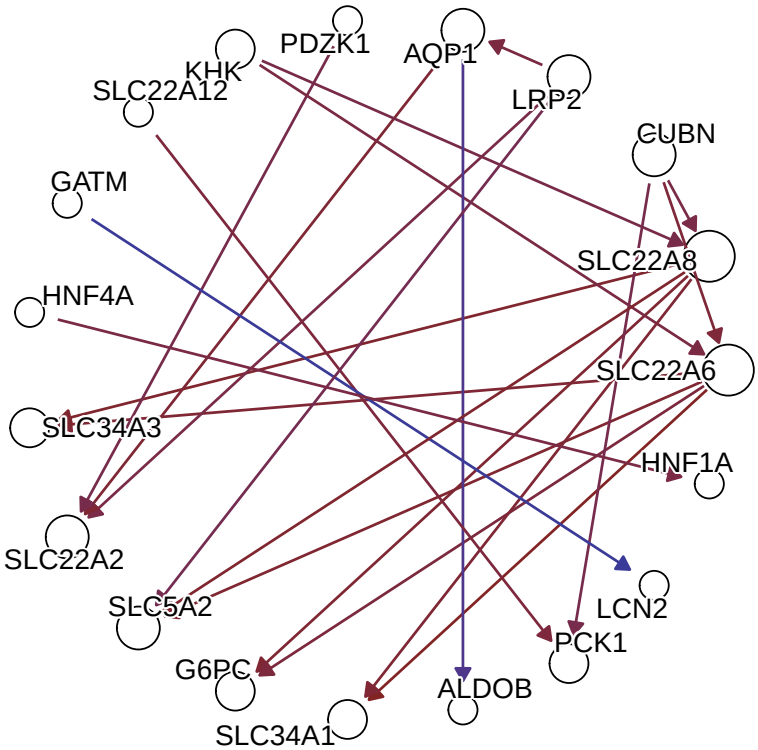

Figure 4: Inferred DAG from single-cell transcriptomics data from proximal tubule cells. The node size indicates node degree and the edge color indicates the mean weight (blue: negative, red: positive). Only genes with bootstrap support ≥ 0.8 were included in the figure.

5. Conclusions

We developed ZICO, an accurate and efficient algorithm to learn DAGs from zero-inflated count data through continuous optimization. Compared to existing methods, ZICO scales to larger numbers of variables, making it practical in real-world settings such as GRN inference from SCT data. Its novelty is not a theoretical identifiability breakthrough, but a practical extension that explicitly models excess zeros via ZIP and ZINB within continuous DAG optimization, together with stable mini-batched training and regularization. As with other score-based approaches, recovering a unique causal DAG is not guaranteed in full generality without additional assumptions, and model misspecification can degrade recovery. We therefore position ZICO as a principled and scalable tool for regimes where an explicit zero-inflation mechanism is appropriate.

Continuous optimization methods have been criticized for issues with varsortability (Reisach et al., 2021). Least-squares type objectives are generally not scale-invariant, so rescaling or stan-

dardization can change the learned structure, whereas a properly regularized log-likelihood-based score can be scale-invariant (shown explicitly for Gaussian models), and thus is not susceptible to varsortability effects arising from variable scaling (Deng et al., 2024). Our method uses the (penalized) log-likelihood objective rather than least squares, which aligns with this recommendation. Our ZIP/ZINB formulations explicitly include the relevant dispersion or zero-inflation components in the likelihood, rather than leaving them implicit or unmodeled. While it is less clear to claim the complete absence of an independent scale parameter in the presence of zero inflation, the key point is that these components are modeled directly in the likelihood. Taken together, these highlight our position that ZICO does not rely on marginal variance ordering for structure recovery; instead, it relies on likelihood-based fit under the stated modeling assumptions.

Acyclicity surrogates like DAGMA's log-determinant penalty have cubic-time complexity in the number of variables, which can become a bottleneck as D grows. While our mini-batched likelihood evaluation scales well in N, the acyclicity term may dominate runtime for large graphs. As a lightweight practical variant, one may impose the DAG constraint only on the mean component W_1 and treat the zero component W_0 as an undirected dependency structure, thereby reducing the number of costly acyclicity evaluations while yielding a single directed output graph. More broadly, we outlined lower-complexity acyclicity constraints such as spectral-radius–based formulations (Nazaret et al., 2024) and structured modeling assumptions for large-scale discovery, like modular connectivity as in DCD-FG (Lopez et al., 2022) for scaling further.

Since ZICO estimates two adjacency matrices, W_0 (zero-inflation) and W_1 (count-mean), the learned DAG depends on the downstream goal. When the primary aim is to infer quantitative regulatory relationships (e.g., GRN), we recommend reporting and interpreting the DAG induced by W_1. Users may also interpret a separate W_0 when structural zeros are of interest. In addition, for applications where both mechanisms may carry important signal, one can construct an integrated graph that combines evidence from the zero and count components (e.g., treating an edge as present if it is supported by either component).

Reverse engineering GRN from SCT data has been reported to be challenging in several works (Chen and Mar, 2018; Dibaeinia and Sinha, 2020). Zero-inflation is also still debated in SCT data; the ZIP or ZINB components in the model help with clearly zero-inflated data. They are otherwise over-parameterized (Cao et al., 2021), as we have shown using simulated SCT data: NB models may then outperform zero-inflated models. In practice, this suggests that model choice should account for both empirical characteristics of the data and prior knowledge of the underlying measurement technology. ZICO can naturally accommodate such choices, since the same optimization scheme can be coupled with different count distributions, including NB and Poisson variants. As a limitation, our empirical evaluation focused on explicitly acyclic structures; feedback loops and other non-DAG regulatory motifs, which are common in biology, and are not captured by the current formulation.

Acknowledgments

This study is partially supported by Takeda Science Foundation, JSPS KAKENHI 25K21331; and grant K25-2170 from the International Joint Usage/Research Center, the Institute of Medical Science, the University of Tokyo.

References

K. Bello, B. Aragam, and P. Ravikumar. DAGMA: Learning DAGs via M-Matrices and a Log-Determinant Acyclicity Characterization. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, pages 8226–8239, 2022.

Y. Cao, S. Kitanovski, R. Küppers, and D. Hoffmann. UMI or Not UMI, That Is the Question for scRNA-seq Zero-Inflation. *Nature Biotechnology*, 39(2):158–159, 2021.

S. Chen and J. C. Mar. Evaluating Methods of Inferring Gene Regulatory Networks Highlights Their Lack of Performance for Single Cell Gene Expression Data. *BMC Bioinformatics*, 19(1): 232, 2018.

D. M. Chickering. Optimal Structure Identification With Greedy Search. *Journal of Machine Learning Research*, 3:507–554, 2002.

J. Choi and Y. Ni. Model-Based Causal Discovery for Zero-Inflated Count Data. *Journal of Machine Learning Research*, 24(200):1–32, 2023.

J. Choi, R. Chapkin, and Y. Ni. Bayesian Causal Structural Learning with Zero-Inflated Poisson Bayesian Networks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, pages 5887–5897, 2020.

T. Claassen and I. G. Bucur. Greedy Equivalence Search in the Presence of Latent Confounders. *Proceedings of Machine Learning Research*, 180 (UAI):443–452, 2022.

D. Colombo, M. H. Maathuis, M. Kalisch, and T. S. Richardson. Learning High-Dimensional Directed Acyclic Graphs with Latent and Selection Variables. *Annals of Statistics*, 40(1):294–321, 2012.

G. Csárdi and T. Nepusz. The igraph Software Package for Complex Network Research. *Interjournal*, Complex Systems:1695, 2006.

T. Cui and T. Wang. A Comprehensive Assessment of Hurdle and Zero-Inflated Models for Single Cell RNA-Sequencing Analysis. *Briefings in Bioinformatics*, 24(5):bbad272, 2023.

H. Dai, I. Ng, G. Luo, P. Spirtes, P. Stojanov, and K. Zhang. Gene Regulatory Network Inference in the Presence of Dropouts: a Causal View. *International Conference on Learning Representations*, 2024:2429–2456, 2024.

Chang Deng, Kevin Bello, Pradeep Ravikumar, and Bryon Aragam. Markov equivalence and consistency in differentiable structure learning. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, volume 37 of *NIPS '24*, pages 91756–91797, Red Hook, NY, USA, December 2024. Curran Associates Inc. ISBN 979-8-3313-1438-5.

P. Dibaeinia and S. Sinha. SERGIO: A Single-Cell Expression Simulator Guided by Gene Regulatory Networks. *Cell Systems*, 11(3):252–271.e11, 2020.

J. Eeckhoute, P. Formstecher, and B. Laine. Hepatocyte nuclear factor 4alpha enhances the hepatocyte nuclear factor 1alpha-mediated activation of transcription. *Nucleic Acids Research*, 32(8): 2586–2593, 2004. ISSN 1362-4962. doi: 10.1093/nar/gkh581.

A. Hauser and P. Bühlmann. Characterization and Greedy Learning of Interventional Markov Equivalence Classes of Directed Acyclic Graphs. *Journal of Machine Learning Research*, 13:2409–2464, 2012.

V. A. Huynh-Thu, A. Irrthum, L. Wehenkel, and P. Geurts. Inferring Regulatory Networks From Expression Data Using Tree-Based Methods. *PLoS One*, 5(9):e12776, 2010.

S. Imoto, T. Higuchi, T. Goto, K. Tashiro, S. Kuhara, and S. Miyano. Combining Microarrays and Biological Knowledge for Estimating Gene Networks via Bayesian Networks. *Proceedings. IEEE Computer Society Bioinformatics Conference*, 2:104–113, 2003.

R. Jiang, T. Sun, D. Song, and J. J. Li. Statistics or Biology: The Zero-Inflation Controversy About scRNA-seq Data. *Genome Biology*, 23(1):31, 2022.

M. Kalisch, M. Mächler, D. Colombo, M. H. Maathuis, and P. Bühlmann. Causal Inference Using Graphical Models with the R Package pcalg. *Journal of Statistical Software*, 47(11):1–26, 2012.

P. V. Kharchenko, L. Silberstein, and D. T. Scadden. Bayesian Approach to Single-Cell Differential Expression Analysis. *Nature Methods*, 11(7):740–742, 2014.

D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

Blue B Lake, Rajasree Menon, Seth Winfree, Qiwen Hu, Ricardo Melo Ferreira, Kian Kalhor, Daria Barwinska, Edgar A Otto, Michael Ferkowicz, Dinh Diep, Nongluk Plongthongkum, Amanda Knoten, Sarah Urata, Laura H Mariani, Abhijit S Naik, Sean Eddy, Bo Zhang, Yan Wu, Diane Salamon, James C Williams, Xin Wang, Karol S Balderrama, Paul J Hoover, Evan Murray, Jamie L Marshall, Teia Noel, Anitha Vijayan, Austin Hartman, Fei Chen, Sushrut S Waikar, Sylvia E Rosas, Francis P Wilson, Paul M Palevsky, Krzysztof Kiryluk, John R Sedor, Robert D Toto, Chirag R Parikh, Eric H Kim, Rahul Satija, Anna Greka, Evan Z Macosko, Peter V Kharchenko, Joseph P Gaut, Jeffrey B Hodgin, KPMP Consortium, Michael T Eadon, Pierre C Dagher, Tarek M El-Achkar, Kun Zhang, Matthias Kretzler, and Sanjay Jain. An atlas of healthy and injured cell states and niches in the human kidney. *Nature*, 619 (7970):585–594, July 2023. ISSN 0028-0836. doi: 10.1038/s41586-023-05769-3. URL http://dx.doi.org/10.1038/s41586-023-05769-3.

H. Li, Z. Zhang, M. Squires, X. Chen, and X. Zhang. scMultiSim: Simulation of Single-Cell Multi-Omics and Spatial Data Guided by Gene Regulatory Networks and Cell-Cell Interactions. *Nature Methods*, 22(5):982–993, 2025.

Romain Lopez, Jan-Christian Hütter, Jonathan K. Pritchard, and Aviv Regev. Large-scale differentiable causal discovery of factor graphs. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, pages 19290–19303, Red Hook, NY, USA, November 2022. Curran Associates Inc. ISBN 978-1-7138-7108-8.

I. Loshchilov and F. Hutter. Decoupled Weight Decay Regularization. In *Proceedings of the International Conference on Learning Representations*, pages 1–11, 2019.

Gleb Martovetsky, James B. Tee, and Sanjay K. Nigam. Hepatocyte Nuclear Factors 4 and 1 Regulate Kidney Developmental Expression of Drug-Metabolizing Enzymes and Drug Transporters. *Molecular Pharmacology*, 84(6):808–823, December 2013. ISSN 0026-895X. doi: 10.1124/mol.113.088229. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC3834141/.

A. McDavid, R. Gottardo, N. Simon, and M. Drton. Graphical Models for Zero-Inflated Single Cell Gene Expression. *The Annals of Applied Statistics*, 13(2):848–873, 2019.

T. Moerman, S. Aibar Santos, C. Bravo González-Blas, J. Simm, Y. Moreau, J. Aerts, and S. Aerts. grnboost2 and Arboreto: Efficient and Scalable Inference of Gene Regulatory Networks. *Bioinformatics*, 35(12):2159–2161, 2019.

A. Nazaret, J. Hong, E. Azizi, and D. Blei. Stable Differentiable Causal Discovery. In *Proceedings of the 41st International Conference on Machine Learning*, pages 37413–37445, 2024.

I. Ng, A. Ghassami, and K. Zhang. On the Role of Sparsity and DAG Constraints for Learning Linear DAGs. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, pages 17943–17954, 2020.

Thomas L. Nickolas, Catherine S. Forster, Meghan E. Sise, Nicholas Barasch, David Solá-Del Valle, Melanie Viltard, Charles Buchen, Shlomo Kupferman, Maria Luisa Carnevali, Michael Bennett, Silvia Mattei, Achiropita Bovino, Lucia Argentiero, Andrea Magnano, Prasad Devarajan, Kiyoshi Mori, Hediye Erdjument-Bromage, Paul Tempst, Landino Allegri, and Jonathan Barasch. NGAL (Lcn2) monomer is associated with tubulointerstitial damage in chronic kidney disease. *Kidney International*, 82(6):718–722, September 2012. ISSN 0085-2538. doi: 10.1038/ki.2012.195.

J. M. Ogarrio, P. Spirtes, and J. Ramsey. A Hybrid Causal Search Algorithm for Latent Variable Models. *Proceedings of Machine Learning Research*, 52 (PGM):368–379, 2016.

Samhita Pal, Dhrubajyoti Ghosh, and Shu Yang. Penalized FCI for Causal Structure Learning in a Sparse DAG for Biomarker Discovery in Parkinson's Disease, June 2025. URL http://arxiv.org/abs/2507.00173. arXiv:2507.00173 [stat].

N. Papili Gao, S. M. M. Ud-Dean, O. Gandrillon, and R. Gunawan. SINCERITIES: Inferring Gene Regulatory Networks From Time-Stamped Single Cell Transcriptional Expression Profiles. *Bioinformatics*, 34(2):258–266, 2018.

G. Park and G. Raskutti. Learning Large-Scale Poisson DAG Models Based on Overdispersion Scoring. In *Proceedings of the 29th International Conference on Neural Information Processing Systems - Volume 1*, pages 631–639, 2015.

A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 8026–8037. 2019.

J. Pearl. *Causal Inference in Statistics*. Wiley, 2021.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-Learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

J. Peters and P. Bühlmann. Structural Intervention Distance (SID) for Evaluating Causal Graphs. *Neural Computation*, 27:771–799, 2015.

A. G. Reisach, C. Seiler, and S. Weichwald. Beware of the Simulated DAG! Causal Discovery Benchmarks May Be Easy to Game. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, pages 27772–27784, 2021.

M. Scutari. Learning Bayesian Networks with the bnlearn R Package. *Journal of Statistical Software*, 35(3):1–22, 2010.

S. Shimizu, T. Inazumi, Y. Sogawa, A. Hyvärinen, Y. Kawahara, T. Washio, P. O. Hoyer, and K. Bollen. DirectLiNGAM: A Direct Method for Learning a Linear Non-Gaussian Structural Equation Model. *Journal of Machine Learning Research*, 12(33):1225–1248, 2011.

A. T. Specht and J. Li. LEAP: Constructing Gene Co-Expression Networks for Single-Cell RNA-Sequencing Data Using Pseudotime Ordering. *Bioinformatics*, 33(5):764–766, 2017.

K. Street, D. Risso, R. B. Fletcher, D. Das, J. Ngai, N. Yosef, E. Purdom, and S. Dudoit. Slingshot: Cell Lineage and Pseudotime Inference for Single-Cell Transcriptomics. *BMC Genomics*, 19(1): 477, 2018.

I. Tsamardinos, L. E. Brown, and C. F. Aliferis. The Max-Min Hill-Climbing Bayesian Network Structure Learning Algorithm. *Machine Learning*, 65(1):31–78, 2006.

S. Yu, M. Drton, and A. Shojaie. Directed Graphical Models and Causal Discovery for Zero-Inflated Data. In *Proceedings of the Second Conference on Causal Learning and Reasoning*, pages 27–67, 2023.

X. Zheng, B. Aragam, P. Ravikumar, and E. P. Xing. DAGs with No Tears: Continuous Optimization for Structure Learning. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 9492–9503, 2018.

Yaochen Zhu, Yinhan He, Jing Ma, Mengxuan Hu, Sheng Li, and Jundong Li. Causal Inference with Latent Variables: Recent Advances and Future Prospectives. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, pages 6677–6687, New York, NY, USA, August 2024. Association for Computing Machinery. ISBN 979-8-4007-0490-1. doi: 10.1145/3637528.3671450. URL https://dl.acm.org/doi/10.1145/3637528.3671450.

Appendix A. Supplementary Tables

Table S1: Hyperparameters used in the experiments.

Section	**Hyperparameters**	**Adjacency matrix for evaluation**	**Threshold**	**Rationale**
Simulation Data	D = 20, 30, 50 $\mu = 1$, $\mu_{\text{decay}} = 0.1$ $\lambda_{\text{align}} = 0.1$ $\lambda_{\text{group}} = 1 \times 10^{-3}$ learning rate $= 1 \times 10^{-3}$ Warmup $= 500$ Frobenius norm epoch $= 4000$	W_1	0.3	We evaluate the recovered DAG using W_1 (the count-mean component), since our target in these simulations is the directed dependency structure governing the quantitative/count mechanism.
Simulated SCT data	$\mu = 1$, $\mu_{\text{decay}} = 0.1$ $\lambda_{\text{align}} = 0.1$ $\lambda_{\text{group}} = 1 \times 10^{-3}$ learning rate $= 1 \times 10^{-3}$ Warmup $= 500$ ℓ_1 norm epoch $= 4000$	$\lvert W_0\rvert + \lvert W_1\rvert$	NA (AUPRC ratio is compared)	To examine the potential impact of the zero-inflation mechanism on structure recovery, we report metrics on an integrated adjacency. This allows us to test whether edges supported by either the zero component or the count component contribute to the recovered structure under zero-inflation-like settings.
Real-world SCT data	$\mu = 1$, $\mu_{\text{decay}} = 0.1$ $\lambda_{\text{align}} = 0.1$ $\lambda_{\text{group}} = 1 \times 10^{-3}$ learning rate $= 1 \times 10^{-3}$ Warmup $= 500$ Frobenius norm epoch $= 5000$	W_1	0.1	The downstream goal is to study relationships among gene expression counts/abundances, and W_1 encodes dependencies in the count-mean model that are most interpretable for these biological interactions.

Table S2: Performance comparison ($D = 100$; BA model).

Algorithm	TPR	FDR	AUPRC	Time (s)	SHD	SID
DirectLiNGAM	0.196 (0.026)	0.765 (0.039)	NA	1001.435 (30.78)	284.7 (14.72)	9644.8 (372.152)
GES	0.358 (0.023)	0.637 (0.023)	NA	4.849 (0.623)	294.6 (10.532)	9269.3 (267.281)
MMHC	0.379 (0.173)	0.562 (0.115)	NA	56.694 (50.477)	258.2 (63.011)	8886.5 (1397.983)
NOTEARS	0.227 (0.036)	0.182 (0.047)	0.215 (0.044)	145.165 (19.08)	260.5 (8.86)	9642.3 (206.28)
ZICO (NB)	0.813 (0.023)	0.335 (0.025)	0.707 (0.041)	203.748 (105.284)	142.6 (10.987)	4389 (461.164)
ZICO (Poisson)	0.687 (0.042)	0.372 (0.039)	0.494 (0.067)	207.789 (99.359)	144 (16.553)	6231.1 (511.681)
ZICO (ZINB)	0.793 (0.029)	0.201 (0.031)	0.719 (0.046)	187.714 (105.74)	87.3 (9.978)	4885.1 (563.624)
ZICO (ZIP)	0.754 (0.027)	0.201 (0.027)	0.711 (0.040)	252.6 (75.055)	98.5 (11.797)	5388.1 (446.703)
ZiDAG	0.481 (0.034)	0.303 (0.032)	NA	30692.257 (5129.947)	167 (9.393)	8307.3 (417.63)
ZiG-DAG (HP)	0.639 (0.035)	0.375 (0.038)	NA	291773 (9306.29)	135.1 (13.98)	6627.2 (365.303)
ZiG-DAG (NB)	0.647 (0.044)	0.362 (0.043)	NA	41599.827 (1310.747)	128.4 (14.607)	6538.4 (553.92)

Values are averages over 10 replicates. The values inside parentheses are standard deviations. The best performing values are shown in bold. NB, negative binomial; ZICO: our proposal.

Table S3: The runtime and peak memory usage of ZICO.

Algorithm	D	N	Epoch	Time (s)	Peak memory (MB)
ZICO (NB)	50	500	4000	112.892 (97.732)	581.84 (1.94)
ZICO (NB)	100	1000	5000	203.748 (105.284)	600.00 (3.40)
ZICO (NB)	200	2000	8000 (total)	229.801 (113.862)	666.07 (4.36)
ZICO (ZINB)	50	500	4000	117.467 (95.069)	584.07 (1.53)
ZICO (ZINB)	100	1000	5000	187.714 (105.740)	614.51 (2.54)
ZICO (ZINB)	200	2000	8000 (total)	291.629 (23.308)	712.50 (5.72)

The results of BA graph. The mean (standard deviation) is reported. On $D = 200$, two-stage training (first disabling the log-det constraint and subsequently enabling it) was used.

Table S4: Performance running ZICO on $\log(x + 1)$ transformed data ($D = 50$).

Algorithm	GT	TPR	FDR	AUPRC	Time	SHD	SID
NB	BA	0.660 (0.044)	0.736 (0.020)	0.519 (0.062)	98.129 (62.050)	283.7 (18.233)	1621.8 (168.198)
NB	ER	0.776 (0.022)	0.607 (0.021)	0.643 (0.031)	96.360 (62.723)	377.5 (16.085)	1731.1 (134.551)
Poisson	BA	0.668 (0.060)	0.739 (0.023)	0.539 (0.092)	85.538 (63.233)	292.0 (15.670)	1566.7 (146.590)
Poisson	ER	0.776 (0.033)	0.624 (0.023)	0.643 (0.039)	85.715 (63.179)	404.6 (15.876)	1718.4 (175.855)
ZINB	BA	0.669 (0.043)	0.587 (0.034)	0.529 (0.072)	125.869 (69.276)	158.1 (16.353)	1633.7 (145.183)
ZINB	ER	0.744 (0.031)	0.519 (0.030)	0.598 (0.047)	137.906 (81.395)	268.6 (15.328)	1869.3 (152.473)
ZIP	BA	0.702 (0.051)	0.547 (0.040)	0.547 (0.074)	126.481 (86.217)	143.8 (15.047)	1517.4 (193.953)
ZIP	ER	0.756 (0.034)	0.513 (0.034)	0.615 (0.045)	126.096 (60.530)	265.2 (21.627)	1831.8 (174.923)

Hyperparameter settings: the same as the $D = 50$ setting in "Simulated Data" section in Table S1.

Appendix B. Supplementary Figure

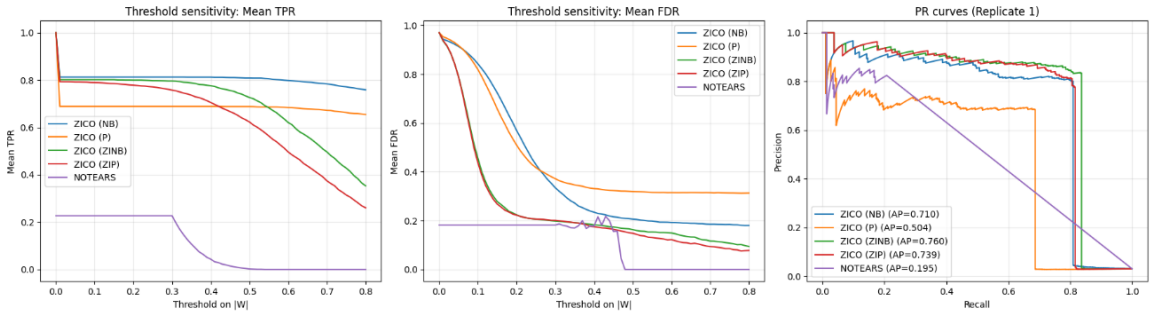

Figure S1: The PR-curve and threshold-sensitivity analysis of continuous optimization methods. Only the continuous optimization methods are listed. For the PR curve, the representative example of one replicate from $D = 100$ settings is used. For the threshold sensitivity of TPR and FDR, the mean value of each threshold is plotted per algorithm.