Beyond Edge Prediction: Residual Set Modeling for Combinatorial Gene Regulation

Siddhi Kanta Mishra[1] **Maryam Rahimimovassagh**[1] **Niloofar Yousefi**[1] **Ivan Garibay**[1]
[1]University of Central Florida
{siddhi.mishra, maryam.rahimimovassagh, niloofar.yousefi, ivan.garibay}@ucf.edu

Abstract

Despite recent advances in genomic machine learning, progress in gene regulatory network (GRN) inference remains constrained by evaluation objectives that fail to reflect the combinatorial nature of biological regulation. Most existing methods frame GRN inference as pairwise edge prediction, even though gene expression is often controlled by specific sets of interacting regulators. As a result, strong edge-level performance does not necessarily imply recovery of the underlying regulatory mechanism.

We reframe target-specific GRN inference as **Exact Regulator Set Recovery** and introduce a two-stage **Filter-and-Refine** pipeline. Stage-1 retrieves a high-recall candidate pool using a target-conditioned attention retriever, while Stage-2 selects the regulator set using a residual high-order set scorer built on top of a decomposable pairwise baseline. Through controlled experiments on synthetic single-cell data with known combinatorial ground truth, we show that improved retrieval ceilings are critical for exact recovery, and that residual modeling of non-additive regulator interactions substantially improves recovery of regulatory logic under sparse supervision.

1 Introduction

Gene regulatory networks (GRNs) govern cellular function through directed interactions between transcription factors and target genes. Single-cell transcriptomics provides unprecedented resolution for GRN inference, but also introduces sparsity, noise, and confounding effects that make causal discovery difficult (13; 6). Recent deep learning systems, including multiome-integrative approaches (23) and cell-specific GRN models (21), have improved *pairwise* edge prediction. Yet edge ranking is misaligned with a central biological reality: transcription factors rarely act alone. Gene regulation is mediated by interacting groups of transcription factors whose joint activity generates transcriptional outcomes (18; 12; 14; 11), rather than by independent regulator–target effects (3; 8; 16; 15; 5). These results suggest that evaluation design, rather than model capacity alone, is a primary barrier to progress in genomic machine learning.

Why edge ranking is not enough. In combinatorial regulation, a model can produce a plausible ranked list and still miss the functional team: recovering most—but not all—true regulators yields an incomplete and potentially misleading mechanism. This failure mode is amplified by hub confounders, where genes with strong marginal correlation dominate pairwise scores even when the true regulatory logic depends on a specific interacting set.

Proposed formulation: GRN inference as set recovery. We reframe target-specific GRN inference as **Exact Regulator Set Recovery**. For each target gene t, we seek the exact regulator set $\mathcal{R}^\star(t)$ rather than a ranked list of edges. We propose a two-stage **Filter-and-Refine** pipeline:

1. **Stage-1 (Retrieval):** produce a high-recall candidate pool $\mathcal{P}_t$ (Top-K).
2. **Stage-2 (Selection):** score subsets $S \subseteq \mathcal{P}_t$ of size R and output $\widehat{\mathcal{R}}(t)$.

Key finding: a sparse-regime optimization barrier. A natural approach is to apply a Set Transformer (9) to score candidate sets. We find that in this sparse genomic regime, "pure" set attention (without a decomposable anchor) can generalize poorly and remain far below residual set scoring, even after basic stabilization attempts (warmup / lower LR; Appendix E). These results motivate a residual formulation that (i) anchors optimization to robust additive evidence and (ii) lets attention focus exclusively on non-additive synergy.

Contributions.

- We formulate target-specific GRN inference as **Exact Regulator Set Recovery**, separating retrieval ceilings from set-level reasoning.
- We introduce a **context-aware attention retriever** that substantially improves candidate coverage.
- We propose **Residual HOS2**, a residual set scorer that stabilizes sparse training by learning non-additive interactions on top of a decomposable base.
- We provide controlled evidence on SERGIO DS3, including oracle injection, showing that residual high-order modeling is necessary to recover regulatory logic.

2 Related Work

Pairwise GRN inference. Most single-cell GRN benchmarks and algorithms evaluate pairwise edge recovery using edge-centric metrics, which assess marginal associations but do not capture the generative mechanisms underlying transcriptional programs (13; 10).

Modern deep-learning approaches improve edge inference using multiome integration and cell-specific modeling (23; 21), as well as representation learning and contrastive objectives (22). However, these systems remain bound to pairwise metrics, which do not measure recovery of cooperative regulator teams (18; 12).

High-order structure in genomics. Hypergraph and higher-order representations have been explored for GRN inference and related tasks, reflecting the broader recognition that biological systems exhibit non-pairwise generative structure (1; 2; 17; 20; 19).

These methods often model global structure, whereas our goal is *target-specific* team discovery under a strict exact-match metric.

Permutation-invariant set learning. Deep Sets (24) and Set Transformers (9) provide principled architectures for learning on unordered collections. Our work highlights an under-emphasized regime: *sparse supervision + noisy features*, where a residual inductive bias can materially improve training stability and sample efficiency.

3 Problem Setup

Let G denote the set of genes. For each target gene t, there is a ground-truth regulator set $\mathcal{R}^\star(t) \subset G$ of size $R = |\mathcal{R}^\star(t)|$. This formulation treats regulator sets as discrete latent variables, aligning GRN inference with structured generative modeling rather than edge ranking.

Exact set recovery metric. We evaluate **Unconditional Exact Match**:

$$\mathrm{Acc} \;=\; \mathbb{E}_t\Big[\mathbb{I}\big(\widehat{\mathcal{R}}(t) = \mathcal{R}^\star(t)\big)\Big].$$

Two-stage factorization and ceilings. Enumerating all possible regulator teams is intractable ($O(|G|^R)$). We therefore decouple the problem into (i) high-recall retrieval of $\mathcal{P}_t$ and (ii) high-precision selection within $\mathcal{P}_t$. Unconditional accuracy is upper-bounded by **Recall@**K:

$$\mathrm{Recall@}K \;=\; \mathbb{E}_t\Big[\mathbb{I}\big(\mathcal{R}^\star(t) \subseteq \text{Top-}K(t)\big)\Big].$$

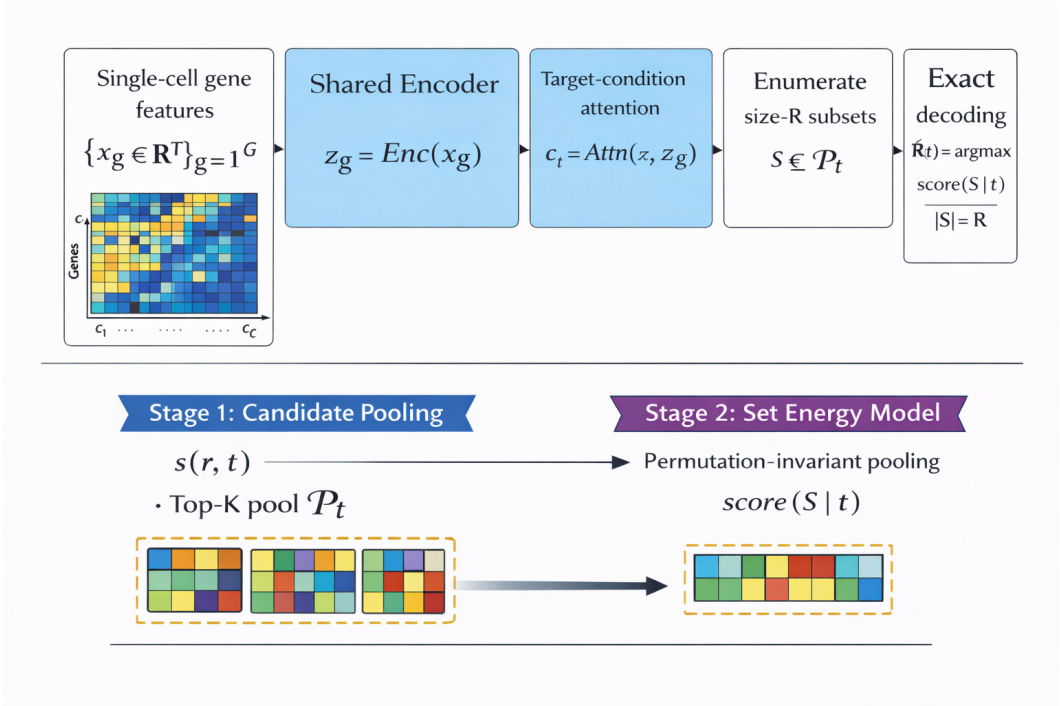

Figure 1: **Method overview: two-stage filter-and-refine pipeline for exact regulator set recovery.** Given gene expression features, Stage-1 retrieves a high-recall candidate pool of regulators for each target gene using a target-conditioned attention retriever. Stage-2 selects the regulator team by scoring candidate sets with a decomposable pairwise base and a residual high-order set module, followed by exact subset decoding. This stage-wise factorization separates retrieval ceilings from set-level reasoning and enables principled evaluation of combinatorial regulatory logic.

For exact subset decoding in Stage-2, we additionally cap the enumerated pool to M_R for tractability, yielding an *effective* ceiling:

$$\text{Coverage@}M_R \;=\; \mathbb{E}_t\Big[\mathbb{I}\big(\mathcal{R}^\star(t) \subseteq \mathcal{P}_t[:M_R]\big)\Big].$$

We report Recall@80 (retrieval quality), Coverage@M_R (ceiling under the decoding protocol), and conditional exact recovery on covered targets.

4 Method

4.1 Overview

Given a target t, Stage-1 produces a candidate pool $\mathcal{P}_t$. Stage-2 scores candidate sets $S \subseteq \mathcal{P}_t$ with $|S| = R$ and outputs the best set:

$$\widehat{\mathcal{R}}(t) \;=\; \arg\max_{S \subseteq \mathcal{P}_t,\, |S|=R} \text{Score}(S, t).$$

4.2 Stage-1: Retrieval

Context-aware attention pooler (Stage-1). We retrieve candidate regulators using a target-conditioned attention mechanism. Each gene is encoded into a normalized embedding via a shared encoder. For a given target gene, we compute a global context vector by attending from the target embedding to all gene embeddings, capturing dataset-level expression structure. Candidate regulators are then scored using an MLP over target, regulator, and context

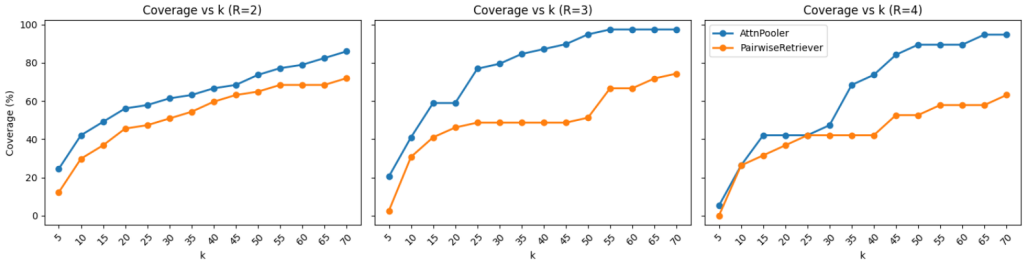

Figure 2: **Stage-1 retrieval coverage versus pool size.** Across regulator team sizes $R \in \{2, 3, 4\}$, the context-aware attention pooler (blue) consistently achieves higher coverage than a pairwise dot-product retriever (orange). The advantage increases with combinatorial complexity: for $R = 3$ and $R = 4$, the pairwise baseline exhibits an extended plateau at moderate coverage levels, while the attention pooler continues to improve and approaches near-saturation at moderate pool sizes. These results indicate that target-conditioned, contextual scoring is critical for reliably including all true regulators in the candidate pool, enabling accurate downstream exact set recovery.

features, including interaction terms. Training uses a weighted multi-label objective to address sparsity. Full architectural and training details are provided in Appendix C.

4.3 Stage-2: Selection

Given $\mathcal{P}_t$, we compare decomposable (pairwise) selection, high-capacity decomposable selection, and residual set selection.

Decomposable selection (PairS2). Let $\phi_{\text{pair}}(r, t)$ denote a learned edge score (MLP on features formed from target/regulator embeddings and their interaction). A decomposable set score is:

$$\text{Score}_{\text{pair}}(S, t) \;=\; \sum_{r \in S} \phi_{\text{pair}}(r, t).$$

For a decomposable scorer, exact decoding reduces to selecting the Top-R individual edges within the pool.

Capacity ablation (Deep PairS2). We match capacity to the set model by increasing depth in both the encoder and edge scorer while keeping the score decomposable:

$$\text{Score}_{\text{deep}}(S, t) \;=\; \sum_{r \in S} \phi_{\text{pair}}^{\text{deep}}(r, t).$$

This isolates whether depth/width alone resolves combinatorial logic.

Residual High-Order Set Scorer (Residual HOS2). We propose a residual decomposition:

$$\text{Score}(S, t) \;=\; \underbrace{\sum_{r \in S} \phi_{\text{pair}}(r, t)}_{\text{decomposable base}} \;+\; \underbrace{\psi_{\text{set}}(S, t)}_{\text{non-additive residual}} \quad .$$

Target-conditioned `[CLS]` and Set Transformer residual. Let $z_t \in \mathbb{R}^d$ be the target embedding from the Stage-2 encoder. We define a target-conditioned `[CLS]` token as a learned affine map of z_t:

$$u_{\text{cls}}(t) = W_{\text{cls}} z_t + b_{\text{cls}} \in \mathbb{R}^d.$$

Given regulator embeddings $\{z_{r_i}\}_{i=1}^{R}$ for a candidate set $S = \{r_1, \ldots, r_R\}$, the residual module forms the sequence

$$U(S, t) = \big[u_{\text{cls}}(t),\; z_{r_1}, \ldots, z_{r_R}\big],$$

applies L layers of self-attention (Set Transformer encoder) (9), and maps the final [CLS] output to a scalar correction:

$$\psi_{\text{set}}(S,t) = w^\top \text{Enc}(U(S,t))_{\text{cls}}.$$

Zero-initialized correction head. To stabilize learning, we initialize the final linear layer (the vector w above) to all zeros, so $\psi_{\text{set}}(S,t) = 0$ at initialization. Training thus begins as a well-behaved decomposable scorer and learns high-order structure as a correction.

Set-ranking objective with curriculum negatives. For Residual HOS2, we use a margin-ranking loss on sets: for each target t, we construct a true set $S^\star$ and a collection of negative sets $\{S_j^-\}$:

$$\mathcal{L}_{\text{rank}}(t) \;=\; \frac{1}{J}\sum_{j=1}^{J} \text{softplus}\Big(m - \big(\text{Score}(S^\star,t) - \text{Score}(S_j^-,t)\big)\Big),$$

with margin $m = 1.0$. Negatives follow a curriculum: early training mixes random negatives with near-miss negatives; later training emphasizes near-miss negatives formed by swapping one true regulator with a high-scoring confuser from the pool. This ranking objective encourages the model to discriminate the true generative regulator set from competing plausible alternatives under sparse supervision (7).

Exact decoding under tractable pool sizes. To remove inference as a confounder, we evaluate Residual HOS2 using **exact subset decoding** within a capped pool size M_R: $M_2 = 80$, $M_3 = 80$, $M_4 = 55$. We enumerate all $\binom{M_R}{R}$ subsets, score them in GPU batches, and return the argmax.

Scalability note. Exact decoding scales combinatorially in R through $\binom{M_R}{R}$ and is therefore practical only for small R under moderate caps (as in our $R \leq 4$ setting). Importantly, the *scoring* function itself scales polynomially in R (a decomposable $O(R)$ base plus a Set Transformer residual over R+1 tokens), so larger regulator teams can be handled by replacing exhaustive enumeration with approximate decoding (e.g., beam search over incremental set construction, local swap-based refinement initialized from the decomposable base, or sample-and-rerank from a learned proposal). We leave a full evaluation of approximate decoding for larger R to future work.

5 Experiments

5.1 Setup

We evaluate on **SERGIO DS3 (de-noised)** (4), a synthetic benchmark in which gene expression is generated from known combinatorial regulator sets, enabling controlled evaluation of exact regulator set recovery.

The dataset contains $G = 1200$ genes and $C = 2700$ cells (9 cell types $\times$ 300 cells/type). We use log1p-transformed expression features and z-score per gene (Appendix A). We stratify targets by $R \in \{2, 3, 4\}$ and use an 80/20 train/test split per stratum. Unless stated otherwise, results report mean$\pm$std over 3 splits/seeds (42/43/44).

5.2 Stage-1 Retrieval Quality: Recall@80

Table 1 reports **Recall@80** on the test set. The context-aware attention pooler substantially improves retrieval quality over a dot-product pairwise retriever.

Table 1: **Stage-1 retrieval quality (Recall@80)** on SERGIO DS3 test targets. Recall@80 measures whether the *entire* true regulator set is contained in the Top-80 candidate regulators for each target.

R	Pairwise Dot-Prod Retriever	Context-Attn Pooler (Ours)	Δ
2	0.6491	0.8947	+0.2456
3	0.7179	0.8718	+0.1538
4	0.8421	1.0000	+0.1579

Table 2: **Exact Regulator Set Recovery on SERGIO DS3 (de-noised).** Stage-1 produces candidate pools of size K=80 using a **Context-Attn Retriever**; Stage-2 performs selection with an exact decoding cap $M_R = \{80, 80, 55\}$ for $R = \{2, 3, 4\}$. We report **Coverage@**M_R (fraction of test targets whose capped pool contains the full true set), **Unconditional Exact** (exact set match over all test targets), and **Conditional Exact** (exact set match restricted to covered targets). Values are mean±std over 3 seeds (42/43/44).

Model	Metric	R=2	R=3	R=4
Coverage@M_R (Stage-1 Context-Attn pools)		0.895 ± 0.018	0.949 ± 0.068	0.930 ± 0.061
Stage1TopR	Unconditional Exact	0.058 ± 0.027	0.034 ± 0.059	0.018 ± 0.030
	Conditional Exact	0.065 ± 0.029	0.034 ± 0.059	0.020 ± 0.034
PairS2 (decomposable)	Unconditional Exact	0.251 ± 0.056	0.222 ± 0.015	0.140 ± 0.061
	Conditional Exact	0.281 ± 0.064	0.236 ± 0.030	0.153 ± 0.072
Residual HOS2 (ours)	**Unconditional Exact**	$\mathbf{0.281 \pm 0.061}$	$\mathbf{0.342 \pm 0.059}$	$\mathbf{0.193 \pm 0.110}$
	Conditional Exact	$\mathbf{0.314 \pm 0.068}$	$\mathbf{0.360 \pm 0.048}$	$\mathbf{0.209 \pm 0.126}$

5.3 Baselines and Ablations

We compare against:

1. **Stage1TopR:** pick Top-R regulators from the Stage-1 pool (no set modeling).
2. **PairS2 (Shallow):** decomposable pairwise Stage-2 scorer.
3. **Deep PairS2:** high-capacity decomposable scorer (Seed 42).
4. **Residual HOS2 (Ours):** decomposable base + residual set attention trained with margin ranking.
5. **Appendix ablation (PureHOS2):** a non-residual Set Transformer scorer; see Appendix E.

5.4 Main Results: Exact Set Recovery

Table 2 reports **Exact Regulator Set Recovery**. **Coverage@**M_R depends only on the Stage-1 pool and the decoding cap M_R; it is identical for all Stage-2 models under the same Stage-1 retriever and caps. We use exact decoding for Residual HOS2 with $M_R = \{80, 80, 55\}$. For decomposable models, the optimal decoding is Top-R edges within the same capped pool. Appendix G isolates the effect of Stage-1 retrieval by training the same PairS2 selector on pools produced by the pairwise dot-product retriever.

Capacity is not the bottleneck. Deep PairS2 does not resolve combinatorial logic recovery and remains low on $R = 4$, indicating that depth alone does not overcome additive ambiguity.

Residual high-order modeling improves exact team recovery under ambiguity. At $K = 80$, Residual HOS2 improves unconditional exact recovery over PairS2 for all complexities, with the strongest gains at higher order: $R = 3$: 0.342 ± 0.059 vs. 0.222 ± 0.015 and $R = 4$: 0.193 ± 0.110 vs. 0.140 ± 0.061. Importantly, conditional exact gains persist

Table 3: **Oracle Injection (Perfect Retrieval).** Exact set recovery when $\mathcal{R}^\star(t) \subseteq \mathcal{P}_t$ is guaranteed by injection and true-safe trimming. Decomposable models (PairS2) decode by Top-R edges; Residual HOS2 performs exact search over subsets of size R in $\mathcal{P}_t$.

Model	**R = 2**	**R = 3**	**R = 4**
Shallow PairS2	0.2456	0.1538	0.0526
Deep PairS2	0.1579	0.1282	0.1053
Residual HOS2	**0.3509**	**0.4359**	**0.3684**

when restricting to covered targets, indicating improvements are not explained solely by retrieval ceiling effects.

Non-residual set scoring remains far below residual set scoring. PureHOS2 can pass a memorization sanity check (near-zero training loss on a small target subset), confirming correct wiring, but generalizes poorly on unconditional exact recovery. A small stabilization sweep (warmup / lower LR) does not close the gap (Appendix E).

Warm-start for PureHOS2. PureHOS2 (Warm-start) copies *only* the **PairS2 encoder** weights into the PureHOS2 encoder (same MLP shape), then freezes/unfreezes; it does *not* inherit the Residual HOS2 residual module or scoring decomposition.

5.5 Oracle Injection: isolating set logic from retrieval and exposing hub confusers

A key difficulty in GRN inference is that *high pairwise association is not equivalent to correct regulatory logic*. In realistic regimes, many targets exhibit strong correlations with a small number of broadly co-expressed genes (*"hubs"* or *transitive confusers*). Pairwise edge predictors therefore tend to rank the same few genes highly across many targets, yielding plausible edge lists that nonetheless fail to recover the *specific non-additive combination* of regulators that functionally controls a given target.

Oracle protocol. To disentangle failures of *retrieval* from failures of *set selection*, we run an *oracle injection* evaluation. For each target t with true regulator set $\mathcal{R}^\star(t)$ of size R, we construct a candidate pool $\mathcal{P}_t$ by taking the Stage-1 Top-M candidates and *injecting* any missing true regulators, followed by *true-safe trimming* that never removes members of $\mathcal{R}^\star(t)$ (Appendix H). This guarantees $\mathcal{R}^\star(t) \subseteq \mathcal{P}_t$ for every evaluated target, so any remaining error must come from the Stage-2 model's inability to identify the correct set *even when the answer is present*.

Why pairwise decoding fails under oracle retrieval. For a strictly decomposable model (PairS2), the optimal set under additive scoring is always obtained by selecting the Top-R edges individually:

$$\hat{S}_{\text{decomp}}(t) \;=\; \arg\max_{|S|=R} \sum_{r\in S} s_{\text{pair}}(r,t) \;=\; \text{Top-}R\big(\{s_{\text{pair}}(r,t)\}_{r\in\mathcal{P}_t}\big).$$

This decoding is fundamentally vulnerable to hub confusers: a broadly co-expressed confuser can crowd out weaker marginal regulators. Increasing MLP capacity does not remove this limitation, because the *additive* structure cannot represent cooperative effects in which a regulator becomes informative only in the presence of others.

Oracle results: residual high-order modeling is necessary. Table 3 reports exact set recovery under oracle injection on de-noised SERGIO DS3. The Residual HOS2 substantially outperforms both shallow and deep decomposable baselines, with gains growing sharply with regulatory complexity. In particular, for $R = 4$ the residual set model attains 0.3684 exact recovery vs. 0.1053 for the strongest decomposable baseline (Deep PairS2), a $\approx 3.5\times$ gap under perfect retrieval that directly isolates the value of modeling non-additive interactions.

Real-data retrieval evaluation. We additionally evaluate the Stage-1 retriever on a real mESC ChIP-seq gene regulatory network paired with single-cell expression data to assess scalability beyond synthetic benchmarks. Because real-world GRNs lack ground-truth regulator teams, this evaluation focuses on retrieval quality rather than exact set recovery. The context-aware attention retriever substantially improves strict regulator-set coverage over a pairwise baseline. Full experimental details and results are provided in Appendix I .

6 Discussion and Limitations

A structural optimization barrier. Pure set attention must learn both marginal regulatory effects and higher-order interactions from sparse supervision. In this regime, optimization often collapses before meaningful set structure is learned. Residual learning reframes the problem by anchoring training to robust decomposable evidence and allocating model capacity exclusively to non-additive structure, revealing a previously underappreciated optimization barrier in GRN inference.

Why capacity alone is insufficient. Increasing depth or width in decomposable models does not resolve this barrier. The oracle injection experiment shows that even under perfect retrieval, additive scoring fails when hub confusers dominate marginal associations. This indicates that the limitation is not representational capacity but inductive bias: additive objectives are fundamentally misaligned with combinatorial regulatory logic.

Limits of current real-world evaluation. Synthetic benchmarks such as SERGIO allow controlled diagnosis by providing exact regulator sets (4). In contrast, real-world GRNs lack context-specific team annotations, making strict evaluation infeasible (13). Bridging this gap will require new benchmarks or proxy objectives that align learned structure with downstream biological effects.

Beyond small regulator sets. While exact decoding is tractable only for small R, the residual scoring formulation itself is agnostic to decoding strategy. Approximate inference methods can extend this framework to larger regulatory teams without altering the core model.

7 Conclusion

This work argues that progress in GRN inference is bottlenecked by a mismatch between combinatorial biological regulation and edge-centric objectives that fail to model the underlying generative structure of gene expression.

We show that additive and pure set-based models struggle in sparse genomic regimes where regulatory function depends on non-additive interactions, even under favorable retrieval conditions. Residual HOS2 overcomes this barrier by decomposing the problem into robust additive evidence and a learnable high-order correction, enabling stable optimization of combinatorial logic. Oracle injection experiments isolate this effect and demonstrate that high-order residuals are necessary when confounding is strong. These results suggest that reframing learning objectives, rather than scaling existing architectures, may be key to advancing generative and structured genomic modeling.

References

[1] Federico Battiston, Giulia Cencetti, Iacopo Iacopini, Vito Latora, Maxime Lucas, Alice Patania, Jean-Gabriel Young, and Giovanni Petri. Networks beyond pairwise interactions: Structure and dynamics. *Physics Reports*, 874:1–92, 2020.

[2] Austin R. Benson, David F. Gleich, and Jure Leskovec. Higher-order organization of complex networks. *Science*, 353(6295):163–166, 2016.

[3] Eric H. Davidson and Michael Levin. Gene regulatory networks. *Proceedings of the National Academy of Sciences*, 102(14):4935–4935, 2005.

[4] Payam Dibaeinia and Saurabh Sinha. Sergio: A single-cell expression simulator guided by gene regulatory networks. *Cell Systems*, 11(3):252–271.e11, 2020.

[5] Ilias Georgakopoulos-Soares, Chengyu Deng, Vikram Agarwal, Candace S. Y. Chan, Jingjing Zhao, Fumitaka Inoue, and Nadav Ahituv. Transcription factor binding site orientation and order are major drivers of gene regulatory activity. *Nature Communications*, 14(1):2333, 2023.

[6] Akshata Hegde, Tom Nguyen, and Jianlin Cheng. Machine learning methods for gene regulatory network inference. *Briefings in Bioinformatics*, 26(5):bbaf470, 2025.

[7] Volodymyr Kuleshov, Nathan Fenner, and Stefano Ermon. Accurate uncertainties for deep learning using calibrated regression. In *International Conference on Machine Learning*, pages 2796–2804. PMLR, 2018.

[8] Samuel A. Lambert, Arttu Jolma, Loris F. Campitelli, Pratyush K. Das, Yimeng Yin, Mihai Albu, Xiaoting Chen, Jussi Taipale, Timothy R. Hughes, and Matthew T. Weirauch. The human transcription factors. *Cell*, 172(4):650–665, 2018.

[9] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3744–3753. PMLR, 2019.

[10] Daniel Marbach, James C. Costello, Robert Küffner, Nicole M. Vega, Robert J. Prill, Diogo M. Camacho, Kyle R. Allison, Manolis Kellis, James J. Collins, and Gustavo Stolovitzky. Wisdom of crowds for robust gene network inference. *Nature Methods*, 9(8):796–804, 2012.

[11] Leonid A. Mirny. Nucleosome-mediated cooperativity between transcription factors. *Proceedings of the National Academy of Sciences*, 107(52):22534–22539, 2010.

[12] Daniel Panne, Tom Maniatis, and Stephen C. Harrison. An atomic model of the interferon-β enhanceosome. *Cell*, 129(6):1111–1123, 2007.

[13] Aditya Pratapa, Amogh P. Jalihal, Jeffrey N. Law, Aditya Bharadwaj, and T. M. Murali. Benchmarking algorithms for gene regulatory network inference from single-cell transcriptomic data. *Nature Methods*, 17(2):147–154, 2020.

[14] Srilakshmi S. Rao, Kamil Ahmad, and Sindhu Ramachandran. Cooperative binding between distant transcription factors is a hallmark of active enhancers. *Molecular Cell*, 81(8):1651–1665.e8, 2021.

[15] Franziska Reiter, Sebastian Wienerroither, and Alexander Stark. Combinatorial function of transcription factors and cofactors. *Current Opinion in Genetics & Development*, 43:73–81, 2017.

[16] François Spitz and Eileen E. M. Furlong. Transcription factors: from enhancer binding to developmental control. *Nature Reviews Genetics*, 13(9):613–626, 2012.

[17] Guangxin Su, Hanchen Wang, Ying Zhang, Marc R. Wilkins, Pablo F. Canete, Di Yu, Yang Yang, and Wenjie Zhang. Inferring gene regulatory networks by hypergraph generative model. *Cell Reports Methods*, 5(4):101026, 2025.

[18] Dimitris Thanos and Tom Maniatis. Virus induction of human ifnβ gene expression requires the assembly of an enhanceosome. *Cell*, 83(7):1091–1100, 1995.

[19] Ramon Viñas, Chaitanya K. Joshi, Dobrik Georgiev, Phillip Lin, Bianca Dumitrascu, Eric R. Gamazon, and Pietro Liò. Hypergraph factorization for multi-tissue gene expression imputation. *Nature Machine Intelligence*, 5(7):739–753, 2023.

[20] Songyang Wu, Kui Jin, Mingjing Tang, Yuelong Xia, and Wei Gao. Inference of gene regulatory networks based on multi-view hierarchical hypergraphs. *Interdisciplinary Sciences: Computational Life Sciences*, 16(2):318–332, 2024.

[21] Junlin Xu, Changcheng Lu, Shuting Jin, Yajie Meng, Xiangzheng Fu, Xiangxiang Zeng, Ruth Nussinov, and Feixiong Cheng. Deep learning-based cell-specific gene regulatory networks inferred from single-cell multiome data. *Nucleic Acids Research*, 53(5):gkaf138, 2025.

[22] Weiming Yu, Zerun Lin, Miaofang Lan, and Le Ou-Yang. Gclink: a graph contrastive link prediction framework for gene regulatory network inference. *Bioinformatics*, 41(3):btaf074, 2025.

[23] Qiuyue Yuan and Zhana Duren. Inferring gene regulatory networks from single-cell multiome data using atlas-scale external data. *Nature Biotechnology*, 43(2):247–257, 2025.

[24] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabás Póczos, Ruslan Salakhutdinov, and Alexander Smola. Deep sets. In *Advances in Neural Information Processing Systems*, volume 30, pages 3391–3401. Curran Associates, Inc., 2017.

A Feature Construction and Preprocessing

We use SERGIO DS3 de-noised expression with $G = 1200$ genes and at least $C = 2700$ cells. We construct RAW2700 features by applying $\log(1 + x)$ to expression and then z-scoring per gene across the first 2700 cells, yielding a feature vector in $\mathbb{R}^{2700}$ per gene.

B Ground-Truth GRN Construction

A key requirement of our evaluation is access to *exact* ground-truth regulator sets for each target gene. This appendix clarifies how ground-truth gene regulatory networks (GRNs) are defined in SERGIO and why they enable principled evaluation of combinatorial regulator recovery.

Origin of Ground-Truth Regulatory Structure: SERGIO is a mechanistic gene expression simulator that generates single-cell RNA sequencing data from a predefined gene regulatory network (4). The GRN is specified *a priori* as a directed graph in which each target gene is regulated by a fixed set of transcription factors. These regulator sets are provided explicitly as part of the dataset specification and are not inferred from expression data.

For each target gene t, SERGIO defines a regulator set

$$\mathcal{R}^\star(t) = \{r_1, \ldots, r_R\},$$

where the regulator set size R is fixed and known. Gene expression trajectories are then generated by simulating transcriptional dynamics governed by these regulatory relationships, followed by the injection of realistic single-cell noise processes.

Combinatorial Regulation: Unlike pairwise GRN benchmarks that focus on independent regulator–target edges, SERGIO explicitly supports *combinatorial regulation*. Each target gene may depend on multiple regulators simultaneously, and its expression is governed by a nonlinear function of the joint regulator activity. As a result, the correct prediction target is not a ranked list of edges, but the *exact regulator set* associated with each gene.

In this work, we focus on targets with regulator set sizes $R \in \{2, 3, 4\}$, which correspond to increasing levels of combinatorial ambiguity. This setting is particularly challenging for additive or decomposable models, as many incorrect regulator combinations can explain marginal expression patterns equally well.

Ground-Truth Labels and Evaluation Targets: Ground-truth regulator sets are extracted directly from the SERGIO interaction files, which list for each target gene the identities of all regulating transcription factors. These sets are treated as immutable ground truth throughout training and evaluation.

Importantly, while gene expression data are simulated, the regulator labels are not noisy or approximate. This allows us to evaluate exact regulator set recovery without confounding effects from label uncertainty or incomplete annotations.

Implications for Evaluation: Because the ground-truth GRN is known exactly, SERGIO enables controlled evaluation of: (i) retrieval quality, measured by whether the true regulator set is included in a candidate pool, and (ii) reasoning quality, measured by exact recovery of the full regulator set.

This separation allows us to disentangle errors arising from candidate retrieval from those arising from higher-order set reasoning. In particular, conditional exact recovery isolates the ability of a model to resolve combinatorial ambiguity given sufficient candidate coverage.

While SERGIO is a simulated benchmark, it provides a necessary testbed for evaluating exact regulator team recovery—an objective that is currently infeasible to assess at scale using real scRNA-seq data due to incomplete ground-truth annotations.

C Stage-1 Architectures and Training Details

Dot-product retriever: A shared MLP encoder maps gene features to h_g, then two linear heads map to $z_g^{(s)}$ and $z_g^{(t)}$. Training uses weighted BCE on multi-label regulator prediction for each target, with self-edges masked.

Context-aware attention pooler: We encode genes to normalized embeddings z_g and compute a target-conditioned context vector using multi-head attention with query z_t over keys/values $\{z_g\}_{g \in G}$. Each candidate regulator r is scored using a learned MLP on features that include z_r, z_t, c_t, interaction terms (elementwise products), and dot products (e.g., $\langle z_r, z_t \rangle$, $\langle z_r, c_t \rangle$). We train with weighted BCE, add small Gaussian noise to inputs during training for stability, and evaluate Recall@80.

Stage-1 maps each gene g to an embedding and ranks candidate regulators for each target. We compare two retrievers.

(Baseline) Dot-product pairwise retriever: We train a lightweight dual-head embedding model: a shared encoder produces a hidden representation h_g, then two projection heads produce a *source* embedding $z_g^{(s)}$ (regulator role) and a *target* embedding $z_g^{(t)}$ (target role). Candidates are scored by a dot product:

$$s(r \to t) \;=\; \langle z_t^{(t)}, z_r^{(s)} \rangle, \qquad \mathcal{P}_t \;=\; \text{Top-}K\{s(r \to t)\}.$$

Training uses a multi-label (BCE) objective over regulators for each target, with self-edges masked.

Context-aware attention pooler (Stage-1): Let $x_g \in \mathbb{R}^D$ denote the feature vector of gene $g \in \{1, \ldots, G\}$. The model first produces a normalized embedding for every gene via a shared encoder

$$h_g = f_\theta(x_g) \in \mathbb{R}^d, \tag{1}$$

$$z_g = \frac{h_g}{\|h_g\|_2} \in \mathbb{R}^d. \tag{2}$$

For each target t, we compute a *target-conditioned global context* vector by attending from the target embedding z_t (as the query) to the set of all gene embeddings $\{z_g\}_{g=1}^G$ (as keys/values). Using multi-head attention with H heads, for head $h \in \{1, \ldots, H\}$ define

$$q_t^{(h)} = W_Q^{(h)} z_t, \qquad k_g^{(h)} = W_K^{(h)} z_g, \qquad v_g^{(h)} = W_V^{(h)} z_g, \tag{3}$$

$$\alpha_{t,g}^{(h)} = \frac{\exp\left(\langle q_t^{(h)}, k_g^{(h)} \rangle / \sqrt{d_h}\right)}{\sum_{j=1}^G \exp\left(\langle q_t^{(h)}, k_j^{(h)} \rangle / \sqrt{d_h}\right)}, \tag{4}$$

$$c_t^{(h)} = \sum_{g=1}^G \alpha_{t,g}^{(h)} v_g^{(h)}, \tag{5}$$

and concatenate heads to obtain the context

$$c_t = \text{Concat}\big(c_t^{(1)}, \ldots, c_t^{(H)}\big) \in \mathbb{R}^d. \tag{6}$$

(Equivalently, $c_t = \text{MHA}(z_t, Z, Z)$ where $Z = [z_1; \ldots; z_G]$.)

Candidate scoring with target context: For each candidate regulator r, we construct an interaction feature map

$$\phi(z_r, z_t, c_t) = \Big[z_r;\; z_t;\; c_t;\; z_r \odot z_t;\; z_r \odot c_t;\; z_t \odot c_t;\; \langle z_r, z_t \rangle;\; \langle z_r, c_t \rangle\Big] \in \mathbb{R}^{6d+2}, \tag{7}$$

where $\odot$ denotes elementwise product. The Stage-1 logit for the directed pair $(r \to t)$ is produced by an MLP scorer $s_\psi(\cdot)$:

$$\ell_{t,r} = s_\psi(\phi(z_r, z_t, c_t)), \quad \ell_{t,t} = -\infty \;\; \text{(mask self-loop)}. \tag{8}$$

Finally, the candidate pool for target t is the Top-K regulators under these logits:

$$\mathcal{P}_t = \text{Top-}K\Big(\{\ell_{t,r}\}_{r=1}^G\Big). \tag{9}$$

Table 4: **PureHOS2 cheap stabilization sweep** (Seed 42). Despite passing a memorization sanity check, PureHOS2 remains far below Residual HOS2 on unconditional exact recovery, especially for $R \geq 3$.

Setting (Seed 42)	R=2	R=3	R=4
A: baseline (lr 2×10^{-4}, no warmup)	0.0877	0.0000	0.0000
B: warmup500 (lr 2×10^{-4})	0.0175	0.0513	0.0000
C: lowerLR (lr 10^{-4}, no warmup)	0.0526	0.0256	0.0526

Training objective: Let $y_{t,r} = \mathbb{I}[r \in \mathcal{R}^\star(t)]$ be the multi-label indicator of true regulators for target t. We train Stage-1 using a weighted binary cross-entropy objective over all regulators:

$$\mathcal{L}_{\mathrm{S1}}(\theta, \psi) = -\sum_{t \in \mathcal{T}_{\mathrm{train}}} \sum_{r=1}^{G} \Big[w\, y_{t,r}\, \log \sigma(\ell_{t,r}) + (1 - y_{t,r})\, \log \big(1 - \sigma(\ell_{t,r})\big) \Big]. \tag{10}$$

where σ is the sigmoid and $w > 0$ is a positive-class weight to address sparsity.

D Optimization Details and Hyperparameters

Stage-1 attention pooler: 20 epochs, AdamW, lr 2×10^{-4}, wd 10^{-3}, batch of 16 targets, embedding dim 128, 4 attention heads, dropout 0.2, noise std 0.01. Stage-2: 15 epochs, AdamW, lr 2×10^{-4}, wd 10^{-3}, batch of 16 targets. Residual HOS2 uses margin $m = 1.0$, negative curriculum (random $\rightarrow$ near-miss), and exact decoding with caps $M_2 = 80$, $M_3 = 80$, $M_4 = 55$.

E PureHOS2 Stabilization Ablation

We evaluate a non-residual set scorer (PureHOS2) that directly scores sets with a Set Transformer (target-conditioned `[CLS]` token) and the same margin-ranking objective as Residual HOS2, but *without* a decomposable base term. A memorization sanity check reaches near-zero loss, confirming correct wiring; however, generalization remains poor. We perform a small stabilization sweep (Seed 42): baseline, LR warmup (500 steps), and lower LR.

F Capacity Ablation: Deep PairS2 (Single Seed)

A natural hypothesis is that the gap between decomposable scoring and Residual HOS2 could be closed by simply increasing model capacity (depth/width) while keeping the score decomposable. We therefore evaluate a **Deep PairS2** variant that increases encoder/scorer depth (implementation-matched to the set model's capacity budget) but retains an additive set score:

$$\mathrm{Score}_{\mathrm{deep}}(S, t) = \sum_{r \in S} \phi_{\mathrm{pair}}^{\mathrm{deep}}(r, t).$$

Because decoding for decomposable models is optimally Top-R edge selection within the capped pool, this ablation isolates *capacity* from *inductive bias*.

Table 5: **Deep PairS2 capacity ablation** on SERGIO DS3 (de-noised), **Seed 42**. We report **Unconditional Exact** set recovery under the same Stage-2 decoding protocol. Despite substantially increased depth, Deep PairS2 does not close the gap to Residual HOS2 on complex targets, indicating that the remaining errors stem from the limitations of decomposable scoring rather than insufficient capacity.

Model (Seed 42)	**R=2**	**R=3**	**R=4**
Deep PairS2 (deeper decomposable)	0.123	0.179	0.053

Table 6: **PairS2 under different Stage-1 candidate pools** on SERGIO DS3 (denoised). Coverage@M_R depends only on Stage-1 retrieval and the cap M_R, while unconditional/conditional exact quantify Stage-2 selection performance. PairS2 is identical in both blocks; only the Stage-1 retriever (pool generator) changes.

Stage-1 Pools	Metric	R=2	R=3	R=4
Pairwise Dot-Prod Pools	Coverage@M_R	0.731 ± 0.037	0.752 ± 0.015	0.649 ± 0.080
	Unconditional Exact	0.222 ± 0.010	0.274 ± 0.065	0.053 ± 0.053
	Conditional Exact	0.305 ± 0.028	0.363 ± 0.080	0.079 ± 0.084
Context-Attn Pools	Coverage@M_R	0.895 ± 0.018	0.949 ± 0.068	0.930 ± 0.061
	Unconditional Exact	0.251 ± 0.056	0.222 ± 0.015	0.140 ± 0.061
	Conditional Exact	0.281 ± 0.064	0.236 ± 0.030	0.153 ± 0.072

G Stage-1 Pooler Ablation: PairS2 on Pairwise vs Context-Attn Pools

To isolate the effect of **Stage-1 retrieval** from **Stage-2 selection**, we train the *same* decomposable selector (PairS2 SAFE) on candidate pools produced by two different retrievers: (i) the **pairwise dot-product retriever** and (ii) the **context-aware attention pooler**. All settings (features, caps $M_R = \{80, 80, 55\}$, negatives, epochs) are held fixed. We report **Coverage@M_R**, **Unconditional Exact**, and **Conditional Exact** on the test set (mean±std over 3 seeds: 42/43/44).

Interpretation. Switching from pairwise dot-product pools to context-aware attention pools substantially increases Coverage@M_R (the retrieval ceiling), especially for larger R. This confirms that improved candidate retrieval is necessary to make exact set recovery feasible under capped decoding.

H Oracle injection protocol and decomposable Top-R optimality

Oracle pool construction: For each target t, we start from the Stage-1 Top-M list, inject any missing members of $\mathcal{R}^\star(t)$, and apply true-safe trimming that never removes regulators in $\mathcal{R}^\star(t)$, yielding $\mathcal{P}_t$ with $\mathcal{R}^\star(t) \subseteq \mathcal{P}_t$ by construction.

Why Top-R decoding is optimal for decomposable scores: If $\mathrm{Score}(S, t) = \sum_{r \in S} s(r, t)$, then the maximizing size-R subset is obtained by taking the R largest individual scores:

$$\arg\max_{|S|=R} \sum_{r \in S} s(r, t) = \text{Top-}R(\{s(r, t)\}_{r \in \mathcal{P}_t}).$$

I Real-World Retrieval Evaluation on mESC

Real-world Retrieval Evaluation on mESC (ChIP-seq network):

We additionally evaluate Stage-1 retrieval on a real gene regulatory network derived from an mESC ChIP-seq network paired with a single-cell expression matrix. The expression matrix contains G=18,386 genes and C=421 cells, and the candidate regulator set contains $|\mathcal{R}|$=229 regulators. After gene-name mapping and filtering, the directed network contains 806,646 kept edges and 12,284 target genes (mean/median/max in-degree: 65.7/71/182). We randomly split targets into 4,000 train targets and 1,500 test targets.

Metrics (regs-only ranking): Because the regulator universe is known ($|\mathcal{R}|$=229), we rank *only* regulators for each target and report two complementary retrieval metrics: (i) **Strict Coverage@K**, the fraction of targets whose *entire* ChIP regulator set is contained in the Top-K ranked regulators,

$$\text{StrictCov@}K = \mathbb{E}_t \left[\mathbb{I}\big(\mathcal{R}^\star(t) \subseteq \text{Top-}K(t)\big) \right],$$

Table 7: **mESC Stage-1 retrieval (regs-only) on 1,500 test targets.** StrictCov@K requires the *full* regulator set to be included in Top-K; EdgeRec@K is the average fraction of true regulators recovered within Top-K.

	Pairwise		Context-Attn (Ours)	
K	**StrictCov@K**	**EdgeRec@K**	**StrictCov@K**	**EdgeRec@K**
100	0.0627	0.8960	0.0727	0.9050
120	0.1227	0.9385	0.1627	0.9510
140	0.2633	0.9665	0.3980	0.9791
160	0.4247	0.9780	0.7040	0.9933
180	0.5567	0.9836	0.9047	0.9979
200	0.6873	0.9900	0.9893	0.9998

and (ii) **Edge-Recall@K**, the per-target fraction of true regulators recovered within the Top-K list, averaged over targets,

$$\text{EdgeRec@}K = \mathbb{E}_t \left[\frac{|\mathcal{R}^\star(t) \cap \text{Top-}K(t)|}{|\mathcal{R}^\star(t)|} \right].$$

We compare a dot-product **pairwise retriever** against our **context-aware attention pooler**.

Results: Figure 3 shows retrieval performance as K increases from 100 to 200. While both methods achieve high edge-level recall at moderate K, the attention pooler dramatically improves *strict* set coverage for these large regulator sets. At K=200, the attention pooler reaches near-saturation strict coverage (0.989) whereas the pairwise retriever remains substantially lower (0.687), indicating that contextual scoring is critical for reliably including the full long-tail regulator set in dense real networks.

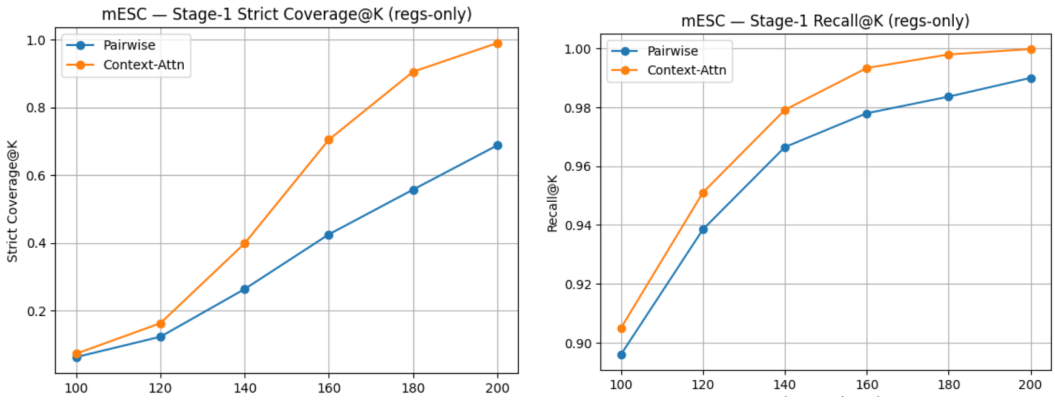

Figure 3: **mESC Stage-1 retrieval scaling (regs-only).** Left: Strict Coverage@K (full regulator set contained in Top-K). Right: Edge-Recall@K (fraction of true regulators in Top-K). The context-aware attention retriever substantially improves strict coverage for large regulator sets, approaching saturation by K=200.

Scope of this real-data evaluation: Unlike SERGIO (where $R \leq 4$ enables exact subset decoding and strict exact-match evaluation), mESC targets often have large regulator sets ($R \approx 70$ median), making exhaustive Stage-2 decoding infeasible. We therefore use mESC primarily to validate the *retrieval* component and highlight how contextual scoring improves the candidate ceiling in dense real GRNs. A full large-R Stage-2 evaluation requires approximate decoding and additional biological validation, which we leave to future work.