DISSECTING LONG-RANGE DEPENDENCY IN GRAPH IMPLICIT MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Capturing long-range dependencies in an effective manner has been one of the key bottlenecks for advances in graph neural networks. Recently, graph implicit models (GIMs) have gained popularity as they provide a means to capture long-range dependency by solving for embeddings that are fixed-point solutions to 'infinitely deeply stacked' graph neural networks, giving an effectively unbounded receptive field. Prior work shows examples where standard graph neural networks demonstrably fail to capture long-range dependencies, while graph implicit models succeed. Yet, graph implicit models do not necessarily outperform standard graph neural networks in real-world long-range benchmarks. In this paper, we theoretically disentangle this conundrum. We first analyze why linear GIMs can solve the simple $(1, L)$ chain task through an aligned learning-dynamics surrogate. We then establish a quantitative vanishing-influence bound: under contraction and sensitivity bounds uniform in graph size, GIMs cannot fit exact scores or maintain a fixed positive classification margin on the generalized (L, L) chain-parity task for sufficiently large L. Motivated by this diagnosis, we propose `AdaGIM`, a GIM with per-node adaptive selection that retains intermediate representations while propagation continues. Benchmarking `AdaGIM` against implicit and standard graph models on synthetic chain-parity and four graph-property datasets shows that it achieves the best mean on three of the four graph datasets in our comparison. GLoRa and selection ablations provide further evidence for the proposed design.

1 INTRODUCTION

Graph-structured data arise throughout machine learning problems, spanning scientific domains (e.g., molecular and protein interaction graphs (Gilmer et al., 2017)) and large-scale information systems (e.g., social and recommender networks (Wu et al., 2022)). Graph neural networks (GNNs) have become a standard tool for representation learning on graph data (Kipf & Welling, 2017). A dominant design paradigm is *message passing*, in which each layer updates a node representation by aggregating information from adjacent nodes followed by a learnable transformation (Gilmer et al., 2017). This paradigm is flexible and effective, but it also has a fundamental geometric constraint: after T layers of message passing, a node representation can only depend on information within its T-hop neighborhood. Capturing *long-range dependencies* with message passing thus seemingly requires building deep networks.

However, increasing the depth of GNNs can cause sharp performance degradation in practice (Chen et al., 2020a). This relates to a widely studied phenomenon called *over-smoothing*, where repeated aggregation drives node representations toward being indistinguishable, limiting discriminative power in deep GNNs. A variety of architectural and algorithmic techniques have been proposed to mitigate this issue (e.g., normalization and edge-dropping mechanisms), but robust long-range learning remains an open challenge in many settings.

Graph implicit models (GIMs) offer a conceptually different route to long-range graph learning (Gu et al., 2020). Instead of explicitly stacking many message-passing layers, implicit GNNs define node states as the solution to an *equilibrium* equation, which can be viewed as the limit of an infinite-depth computation with weight sharing. Representative examples include Implicit Graph Neural Networks (Gu et al., 2020) (IGNN) and Efficient Infinite-Depth Graph Neural Networks (Liu et al., 2021) (EIGNN). These models are explicitly motivated by the long-range limitation of finite-depth

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

aggregation, and empirical studies report improvements on synthetic and real-world benchmarks designed to stress long-range dependency, including chain-structured tasks where conventional finite-depth baselines deteriorate as the chain length increases.

Despite this promise, GIMs are not yet a mainstream default, even in long-range benchmarks. This leads to a basic question: *how well can GIMs capture long-range dependencies?* To probe this question in a controlled way, we introduce a simple long-range node-classification benchmark that generalizes the standard chain test often used by existing GIMs to demonstrate long-range dependency. We call this the (K, L) chain-parity benchmark.

The (K, L) chain-parity task. For $1 \leq K \leq L$, let $V = \{1, \ldots, L\}$ and $E = \{(i, i+1) : i < L\}$. Features satisfy $x_i \in \{0, 1\}$ for $i \leq K$ and $x_i = 0$ otherwise. Node i has label

$$y_i = \left(\sum_{j=1}^{i} x_j \right) \bmod 2 = \bigoplus_{j=1}^{i} x_j.$$

The complete dataset contains all 2^K informative bit patterns.

Given $(G, \{x_i\}_{i=1}^{L})$, the learning objective is node classification: predict y_i at every node i. In particular, for nodes far from the first K positions (i.e., $i \gg K$), accurate prediction requires propagating the parity information along the chain, which gets increasingly long-range as L grows. The synthetic *chain* benchmarks commonly used to evaluate implicit GNNs (Gu et al., 2020; Liu et al., 2021; 2022) similarly test propagation from a single informative source; our $(1, L)$ case captures this setting. This observation leads to a natural next question: *can graph implicit models fit the (K, L) chain-parity data when $K \geq 1$?*

In this work, we approach this question from two complementary angles. First, we analyze the learning dynamics of *linear* graph implicit models on the $(1, L)$ chain-parity family. Our theoretical results identify a locally stable equilibrium that generalizes on simple chains in an aligned free-matrix surrogate of a linear GIM, offering insight into how implicit propagation can support the existing synthetic chain experiments. Next, we show that this success does not automatically extend to $K > 1$. Using a different line of analysis (not restricted to linear models), we show that GIMs with contraction and sensitivity bounds uniform in graph size cannot maintain a fixed output-score separation for inputs differing at a distant node. At a high level, the difficulty is that GIMs propagate information through repeated local aggregation, and the contribution of early inputs can become increasingly weak as the distance grows. This limits the model's ability to preserve the alternating (parity) signal over long ranges. Our contributions are:

- We introduce the (K, L) chain-parity family, generalizing the chain benchmark used in prior work. We also provide a theoretical analysis identifying a locally stable equilibrium that generalizes across lengths on the $(1, L)$ subfamily in an aligned free-matrix surrogate.
- We establish a quantitative vanishing-influence bound (Lemma 3.5): with contraction and sensitivity bounds uniform in graph size, GIMs cannot fit exact parity scores or maintain a fixed positive classification margin on the full (L, L) task for sufficiently large L (Theorem 3.7).
- Inspired by our analysis, we propose `AdaGIM`, a GIM that decouples propagation from selection through a learned nodewise gate, retaining intermediate representations while propagation continues.
- We benchmark `AdaGIM` against several GIMs and GNNs on four real-world datasets and evaluate its behavior on chain-parity, GLoRa, and controlled ablations.

2 Generalization of graph implicit models on simple chains

The claim that GIMs are better at capturing long-range dependencies has been supported by several existing works (Gu et al., 2020; Liu et al., 2021; 2022; Baker et al., 2023), often through experiments on synthetic chain datasets. In this setup, the goal is to test whether a model can propagate a weak signal across many hops. Each graph is a chain of length L, and the task is a binary node-classification problem induced by two chain types. The only cue distinguishing the two types is a single bit, either 0 or 1, injected into the feature vector of the first node; all other nodes carry no type-specific information.

All nodes in the chain are assigned label 1 if the injected bit is 1, and label 0 otherwise. In the standard setting, labeled nodes are sampled uniformly at random along the chain to form the training set, while the rest are used for validation and testing.

We consider a harder non-uniform split. Instead of sampling labeled nodes uniformly, the training set contains only nodes up to position $l < L$, while nodes beyond l are reserved for validation and testing. This setting is more challenging because the model receives supervision only near the signal source and must *generalize* to farther nodes that are never labeled during training. For a forward directed chain with the same propagation rule, its supervised prefix coincides with a shorter $(1, l)$ instance. Evaluation on the held-out positions tests generalization beyond the supervised region; evaluation on separately generated longer graphs additionally tests graph-length generalization.

Setting: We begin by analyzing *EIGNN* (Liu et al., 2021) on the chain-parity task introduced above. We focus on EIGNN because, once the parameters are fixed, its propagation rule is linear. This allows us to explicitly characterize how information from the first nodes is propagated along the chain.

Consider the general (K, L) chain-parity data. There are $N = 2^K$ length-L chains, corresponding to all possible binary assignments on the first K nodes. Let $x^{(i)} \in \{0, 1\}^L$ denote the scalar node-feature vector of the i-th chain, where the first K entries encode one of the 2^K possible binary patterns and the remaining entries carry no type-specific information. Let $y^{(i)} \in \{0, 1\}^L$ denote its node-wise parity labels.

EIGNN's node embeddings $H^{(i)} \in \mathbb{R}^{L\times d}$ solve the fixed-point equation

$$H^{(i)} = AH^{(i)}M + x^{(i)}v^\top, \qquad M = \gamma g(W), \qquad g(W) = \frac{W^\top W}{\|W^\top W\|_F + \epsilon}, \tag{1}$$

where $A \in \mathbb{R}^{L\times L}$ is the propagation adjacency, $v \in \mathbb{R}^d$ is the input projection vector, $W \in \mathbb{R}^{d\times d}$ is learnable, $\gamma \in [0, 1)$ controls contractivity, and $\epsilon > 0$ is a numerical stabilization constant. Predictions use a linear readout, $\hat{y}^{(i)} = H^{(i)}u$, with $u \in \mathbb{R}^d$. Let P_l select the first l coordinates. Supervision on these nodes gives the squared loss $\mathcal{L} = \frac{1}{2}\sum_i \|P_l(y^{(i)} - \hat{y}^{(i)})\|_2^2$.

Unrolling (1) gives $H^{(i)} = \sum_{j\geq 0} A^j x^{(i)} v^\top M^j$. This expansion converges when $\|A\|_2\|M\|_2 < 1$; on a directed chain without self-loops it is finite because $A^L = 0$. Therefore, the training loss can be rewritten as

$$\mathcal{L} = \frac{1}{2}\sum_{i=1}^{N} \left\| P_l \left(y^{(i)} - \sum_{j\geq 0} A^j x^{(i)} v^\top M^j u \right) \right\|_2^2 . \tag{2}$$

The learnable parameters $\Theta = \{v, W, u\}$ are updated by gradient descent.

Learning dynamics of GIMs on chains. For the learning-dynamics analysis, we treat the effective matrix $M := \gamma g(W)$ as a free optimization variable. This isolates the mechanism relevant to the chain-parity task: repeated propagation by A and repeated transformation by M. The resulting surrogate describes these effective dynamics, rather than the exact optimization trajectory of EIGNN's constrained parameterization.

Following mode-wise analyses of deep linear and recurrent networks (Saxe et al., 2014; Proca et al., 2025), we consider an aligned setting in which u and v lie in the same eigenmode of M. This assumption decouples the dynamics into a single spectral direction and reduces the high-dimensional gradient flow to a two-scalar system that can be analyzed explicitly.

Assumption 2.1 (Model alignment)**.** For a unit vector e, the model is initialized with $M(0)e = M(0)^\top e = \lambda(0)e$ and $u(0) = v(0) = a(0)e$, where $a(0) \neq 0$.

Gradient flow preserves this alignment. Under Assumption 2.1, let $c := v^\top u = a^2$ and define the supervised data correlations

$$\sigma_{yx}^k = \sum_i (P_l y^{(i)})^\top P_l A^k x^{(i)}, \qquad \sigma_{xx}^{k,j} = \sum_i (P_l A^k x^{(i)})^\top P_l A^j x^{(i)}.$$

With $\beta_k = \sigma_{yx}^k - c\sum_{j\geq 0} \sigma_{xx}^{k,j}\lambda^j$, the gradient flow reduces to

$$\dot{c} = 2c\sum_{k\geq 0} \beta_k \lambda^k, \qquad \dot{\lambda} = c\sum_{k\geq 1} k\beta_k \lambda^{k-1}. \tag{3}$$

For a directed chain $A_{r,r-1} = 1$ without self-loops, the $(1, L)$ correlations simplify and the loss becomes $\mathcal{L}(c, \lambda) = \frac{1}{2}\sum_{k=0}^{l-1}(1 - c\lambda^k)^2$. The reduced dynamics have a stable nontrivial fixed point $(c, \lambda) = (1, 1)$, corresponding to correct long-range propagation. The derivation and proof are provided in Appendix B.

Proposition 2.2 (Generalization on simple chains)**.** *For the $(1, L)$ directed chain-parity task with $l \geq 2$, under Assumption 2.1, the free-matrix gradient flow* (3) *has a locally asymptotically stable, L-invariant equilibrium $(c, \lambda) = (1, 1)$. It predicts every node correctly on every longer directed chain with the same propagation rule.*

This calculation explains long-range generalization on the simple $(1, L)$ task: the same learned mode transfers a single bit to arbitrarily distant nodes. The ideal solution $\lambda = 1$ belongs to the free-matrix surrogate; contractive EIGNN has $\|M\|_2 < 1$ and produces scores $c\lambda^{r-1}$ in an aligned mode. The proposition characterizes local learning and length generalization of this ideal equilibrium, while the next section considers the additional demands of multi-bit parity.

3 Long-range dependency bottlenecks in graph implicit models

Proposition 2.2 explains generalization on the simple $(1, L)$ task. Figure 1 shows that IGNN, EIGNN, and MGNNI retain high accuracy far beyond the supervised prefix for $K = 1$, while their performance drops substantially for $K = 2, 3$. This prompts us to extend the analysis beyond the single-source setting. To this end, we study the (L, L) chain-parity task, where the input dimension and graph length grow together. Multi-bit XOR already exceeds the expressivity of a linear encoder and readout; the following analysis applies also to nonlinear GIMs and isolates the effect of propagation distance.

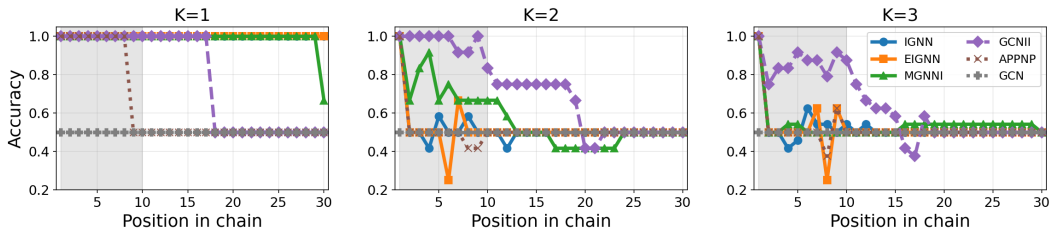


Figure 1: Per-position accuracy on chain-parity for $K \in \{1, 2, 3\}$. The shaded prefix contains the supervised positions; the remaining positions are held out. The legend applies to all three panels.

A vanishing-influence result on a chain. We formalize a simple statement: under uniform contraction and Lipschitz assumptions, the fixed-point encoder cannot transmit a bit-flip at the first node to a distant node without exponential attenuation.

Assumption 3.1 (Encoder–decoder form)**.** The encoder is a fixed point $H = f(x, A, H)$ of a local message-passing operator, with nodewise updates $h_i = f_i(x_i, \{h_j : j \in \mathcal{N}(i)\})$. Prediction scores use a nodewise decoder, $s_i = g(h_i)$.

Assumption 3.2 (Lipschitz dependence on features)**.** With hidden states fixed, each f_i is L_x-Lipschitz in x_i.

Assumption 3.3 (Uniform nodewise contraction)**.** There are nonnegative coefficients B_{ij}, zero outside $\mathcal{N}(i)$, such that

$$\|f_i(x_i, H) - f_i(x_i, H')\| \leq \sum_j B_{ij}\|h_j - h'_j\|, \qquad \max_i \sum_j B_{ij} \leq \rho < 1.$$

For a family of graph lengths, ρ and L_x are independent of length.

This formalizes the local stability used in the analysis. For a 1-Lipschitz activation applied to $\sum_j A_{ij}h_jW + Ux_i$, one sufficient choice is $B_{ij} = |A_{ij}|\|W\|$ and $L_x = \|U\|$, with $\|W\| \max_i \sum_j |A_{ij}| < 1$. Appendix C.5 gives sufficient conditions for the architectures considered here.

Remark 3.4. In the norm $\|H\|_{\infty,\text{node}} = \max_i \|h_i\|$, Assumption 3.3 gives strict contraction with constant ρ. On a complete invariant state space, the iteration therefore converges to a unique equilibrium.

On a chain, write $p_i = B_{i,i+1}/(1-B_{ii})$ and $q_i = B_{i,i-1}/(1-B_{ii})$, setting missing-edge coefficients to zero. Define $Q_1 = 1$ and $Q_k = 1 - p_{k-1}q_k/Q_{k-1}$. These denominators are positive under Assumption 3.3.

Lemma 3.5 (Sensitivity bound)**.** *Under Assumptions 3.1–3.3, let x, x' differ only at the first node of a chain of length L. Their fixed-point embeddings satisfy*

$$\|h_L - h'_L\| \le \frac{L_x}{1-B_{11}} \prod_{k=2}^{L} \frac{q_k}{Q_k} \|x_1 - x'_1\|. \tag{4}$$

Remark 3.6 (Vanishing influence). The homogeneous-coefficient case yields exponential attenuation along the chain. More generally, the length-uniform contraction bound gives $\|h_L - h'_L\| \le L_x \rho^{L-1} \|x_1 - x'_1\|/(1-\rho)$, including heterogeneous node coefficients (Appendix C).

Theorem 3.7 (Chain-parity difficulty)**.** *Consider a family of chain encoders satisfying Assumptions 3.1–3.3, with $0 < \rho < 1$, $L_x < \infty$, and a decoder Lipschitz bound $L_g < \infty$ independent of L. For inputs differing only in the first bit,*

$$|s_L(x) - s_L(x')| \le D\rho^{L-1}, \qquad D = \frac{L_g L_x}{1-\rho}.$$

Consequently, exact fitting of the (L, L) parity scores $0, 1$ is impossible once $D\rho^{L-1} < 1$. Classification with a fixed margin $\eta > 0$ on each side of a common threshold is impossible once $D\rho^{L-1} < 2\eta$.

The proof relies on a contradiction between the output separation required by the task and the attenuation enforced by the model. A first-bit flip changes the last parity label, whereas its influence decays exponentially with distance. The stated margin is essential: arbitrarily small score differences can still cross a threshold. This sharpens the sense in which uniform contraction limits reliable long-range prediction. Figure 2 illustrates this distance-dependent decay through embedding changes induced by a source-node perturbation. The corresponding chain diagnostic and the paired plots appear in Appendix G.5.

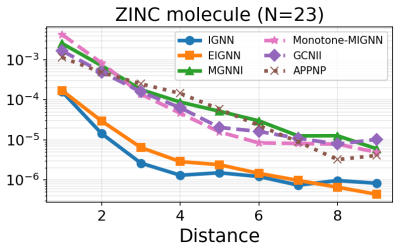


Figure 2: Influence versus distance on ZINC ($N = 23$).

4 Proposed Method: GIM with Adaptive Selection

The vanishing-influence analysis motivates controlling how much propagation is used in each node's prediction. We first consider a construction for chain data: unroll an IGNN-style recurrence $H^t = \phi(AH^{t-1}W + xv^\top)$ and let node i use its representation after i updates, $Z_i = H_i^i$. Different nodes thus use different numbers of updates while receiving information from their predecessors. We call this approach *IGNN-finite*.

Evaluation on (K, L) chain-parity. Figure 3 shows that IGNN-finite achieves perfect accuracy through $L = 150$ and $K = 20$ after training on $(10, 10)$. This motivates extending node-dependent selection to arbitrary graphs. For predecessor-only directed chains without self-loops, $H_i^i = H_i^\star$, so diagonal and equilibrium readouts coincide; the benefit of this construction therefore depends on the propagation rule.

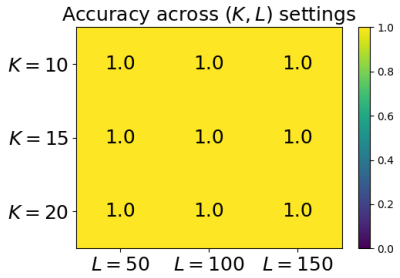


Figure 3: IGNN-finite accuracy after training on $(10, 10)$, using the diagonal readout $Z_i = H_i^i$.

4.1 GIM WITH ADAPTIVE SELECTION

The chain construction uses known node positions. We now introduce `AdaGIM`, short for Adaptive Graph Implicit Model, which extends the selection idea to arbitrary graphs. Our core idea is to decouple propagation from selection. The propagation state H^t evolves according to a base graph implicit model, while the selected state Z^t stores the representation that the model chooses to retain. At each iteration, the model first computes a propagated embedding and then decides whether to update the selected embedding:

$$H^{t+1} = \bar{f}_\theta(H^t), \qquad Z^{t+1} = a^{t+1} \odot H^{t+1} + (1 - a^{t+1}) \odot Z^t. \tag{5}$$

Here $\bar{f}_\theta = (1 - \alpha_P)I + \alpha_P f_\theta$, $0 < \alpha_P \leq 1$, and each scalar node gate a_v^{t+1} is broadcast over the feature dimension. This defines an augmented update map $F_{\theta,\xi}$ on (H, Z).

Practical selection gate. A hard accept/reject rule is difficult to optimize directly. We implement a soft per-node gate $a^{t+1} = q^{t+1} \odot \widetilde{\sigma}^{t+1}$ with $\tau_q, \tau_g > 0$, where

$$q_v^{t+1} = 1 - \exp\left(-\frac{\|H_v^{t+1} - H_v^t\|_2^2}{\tau_q^2}\right), \qquad \widetilde{\sigma}_v^{t+1} = \mathrm{sigmoid}\left(\frac{\mathrm{MLP}_\xi([H_v^{t+1}, H_v^t])}{\tau_g}\right).$$

The learned factor controls acceptance; the residual factor vanishes at the propagation equilibrium. A gate close to one accepts the new embedding, while a gate close to zero preserves the selected representation. This halts updates to the retained state while propagation continues.

Well-posedness of the initialized model. Fix both initial states and the update rule; the first core uses $H^0 = Z^0 = 0$. Convergent propagation and convex selection give the following guarantee.

Proposition 4.1 (Equilibrium selected by initialization)**.** *For fixed input, parameters, damping, and initialization ι, suppose $H^t \to H^\star$ and $\bar{f}_\theta$ is continuous at $H^\star$. Then* (5) *with the continuous residual gate converges to a unique selected equilibrium $(H^\star, Z_\iota^\infty)$ along that initialized trajectory.*

If the total write mass is finite, selection can retain earlier states; if it diverges, $Z^t \to H^\star$. Under contractive propagation, the residual gate has finite total write mass and a geometric finite-unrolling error bound (Appendix D). Thus adaptive selection is compatible with convergence. The guarantee selects one limiting representation from the prescribed initialization; it does not assert uniqueness of every algebraic root of F. The appendix also gives the original sufficient condition for global uniqueness when selection is strictly contractive in Z.

Exact architecture of `AdaGIM`. We instantiate $f_\theta(H) = \mathrm{LN}(\phi(\hat{A}^\top HW + U_X))$, with optional LayerNorm, $\phi \in \{\tanh, \mathrm{ReLU}\}$, and Cayley weights $W = s_W(I - S)(I + S)^{-1}$, $S = C - C^\top$ (Baker et al., 2023). Contraction of the complete operator, including normalization, is the sufficient condition for the geometric bound. All cores share U_X and warm-start propagation from the preceding selected output. Unrolling the gate assigns each H_v^t the last-write weight $\omega_v^t = a_v^t \prod_{s=t+1}^T (1 - a_v^s)$, with no-write mass $\omega_v^0 = \prod_{s=1}^T (1 - a_v^s)$. We penalize entropy and no-write mass and train through finite unrolling; full architecture and training details are in Appendix E. Related gated and adaptive-depth models are discussed in Appendix A.

5 EXPERIMENTS

We evaluate AdaGIM on four standard graph benchmarks spanning small-molecule property prediction, long-range biological graphs, and molecular-property classification, and on controlled synthetic tasks that probe long-range information routing along chains and branching dependency structures. Our code is at https://anonymous.4open.science/r/AdaGIM-E89F.

Datasets and tasks. We focus on four real-world graph benchmarks: ZINC (a 12,000-molecule subset; constrained-solubility regression; MAE) (Dwivedi et al., 2023); PEPTIDES-FUNC (multi-label biological graph classification; AP) and PEPTIDES-STRUCT (regression onto 11 structural descriptors; MAE) (Dwivedi et al., 2022); and OGBG-MOLHIV (binary HIV-inhibition classification; ROC-AUC) (Hu et al., 2020a). The two LRGB Peptides datasets are designed to expose long-range modelling capacity. ZINC and OGBG-MolHIV provide complementary molecular-property tasks. The main Peptides comparison uses the official datasets and splits with a reduced training protocol; the stronger-protocol comparison below is separate.

Baselines. We consider two types of baselines. (1) emphMessage-passing GNNs (MPGNN) like *GCN* (Kipf & Welling, 2017), *APPNP* (Klicpera et al., 2019), and *GCNII* (Chen et al., 2020b) — finite-depth, non-implicit reference points; and we (2) *Graph-implicit models (GIM). IGNN* (Gu et al., 2020) (infinity-norm contraction), *EIGNN* (Liu et al., 2021) (closed-form via eigendecomposition), *MGNNI* (Liu et al., 2022) (multi-scale, attention over depths), *GIND* (Chen et al., 2022) (implicit nonlinear graph diffusion), *Monotone-MIGNN* (Baker et al., 2023) (monotone-operator equilibrium). Appendices F and G provide training, tuning, and aggregation details.

5.1 MAIN RESULTS

Table 1 reports the four real-world graph benchmarks. AdaGIM attains the best mean on **three of the four datasets** (Peptides-func, Peptides-struct, ZINC) and is competitive on the fourth (OGBG-MolHIV), where Monotone-MIGNN has the best mean.

Adaptive selection helps across graph regimes. On the long-range LRGB Peptides datasets, AdaGIM improves over the fixed-point GIM baselines on both tasks. The same pattern holds on ZINC. The gains span regression and classification, suggesting that adaptive selection is useful across prediction objectives. The better mean performance than IGNN-finite also motivate examining learned retention beyond a predefined stopping rule; the gate and readout ablations below test this distinction. Aggregate selection weights are reported in Appendix G.

Table 1: Results under the reduced graph-task protocol. Seed and aggregation details are discussed in Appendix G. Bold marks the best mean.

	LRGB Peptides		Molecular	
	Peptides-func	Peptides-struct	ZINC	OGBG-MolHIV
Model	AP ↑	MAE ↓	MAE ↓	AUC ↑
GCN	0.4494 ± 0.0018	0.3768 ± 0.0016	0.5268 ± 0.0058	0.7113 ± 0.0287
APPNP	0.3815 ± 0.0032	0.3422 ± 0.0014	0.7042 ± 0.0018	0.6859 ± 0.0114
GCNII	0.4331 ± 0.0040	0.3437 ± 0.0046	0.5641 ± 0.0028	0.7142 ± 0.0160
IGNN	0.4097 ± 0.003	0.3266 ± 0.001	0.5734 ± 0.013	0.7505 ± 0.035
EIGNN	0.3989 ± 0.003	0.3285 ± 0.001	0.6290 ± 0.005	0.7320 ± 0.009
MGNNI	0.3669 ± 0.004	0.3325 ± 0.005	0.6338 ± 0.012	0.7584 ± 0.008
GIND	0.4545 ± 0.004	0.3410 ± 0.001	0.5721 ± 0.005	0.7112 ± 0.012
Monotone-MIGNN	0.4501 ± 0.005	0.3276 ± 0.002	0.5147 ± 0.007	$\mathbf{0.7607} \pm 0.010$
IGNN-finite	0.4270 ± 0.003	0.3265 ± 0.001	0.5072 ± 0.010	0.7497 ± 0.011
AdaGIM	$\mathbf{0.4934} \pm 0.007$	$\mathbf{0.3034} \pm 0.001$	$\mathbf{0.4701} \pm 0.029$	0.7527 ± 0.008

Table 2: $(K, 20)$ chain-parity results: accuracy on positions 11–20, with supervision on positions 1–10 of length-20 chains. Mean±standard deviation over three seeds. Bold marks the best mean.

Model	$K = 1$	$K = 2$	$K = 3$
GCNII	0.850 ± 0.000	0.708 ± 0.029	0.562 ± 0.012
IGNN	$\mathbf{1.000 \pm 0.000}$	0.492 ± 0.014	0.504 ± 0.007
EIGNN	$\mathbf{1.000 \pm 0.000}$	0.500 ± 0.000	0.500 ± 0.000
MGNNI	$\mathbf{1.000 \pm 0.000}$	0.492 ± 0.014	0.521 ± 0.036
GIND	0.750 ± 0.000	0.625 ± 0.000	0.562 ± 0.000
Monotone-MIGNN	0.683 ± 0.231	0.692 ± 0.080	0.533 ± 0.026
`AdaGIM`	$\mathbf{1.000 \pm 0.000}$	$\mathbf{0.800 \pm 0.043}$	$\mathbf{0.717 \pm 0.040}$

Table 3: Gate ablation in a separate configuration. Best entry per column is bold.

Gate	Peptides-func AP ↑	ZINC MAE ↓	Peptides-struct MAE ↓
Full, $\sigma = q\widetilde{\sigma}$	**0.5003**	0.4523	0.3078
MLP only, without q	0.4909	**0.4382**	**0.2956**
Residual q only	0.4396	0.4798	0.3252
No gate, $Z = H^\star$	0.4407	0.4785	0.3350

5.2 Long-range routing diagnostic: chain-parity

To complement the real-world results with a controlled probe of long-range information routing, we evaluate on the synthetic chain-parity benchmark: directed chains $v_1 \to \cdots \to v_L$ with binary input bits at the first K positions, zero inputs thereafter, and per-position labels equal to the prefix XOR, $Y_v = \bigoplus_{i \le v} X_i$. Models train on length-20 chains with supervision restricted to positions 1–10, and Table 2 reports *unseen*-position accuracy on positions 11–20 of length-20 chains. This protocol measures generalization to held-out positions on the training graph structure. For $K \le 3$, the evaluated suffix tests retention of the prefix parity.

At $K = 1$, three GIM baselines (IGNN, EIGNN, MGNNI) and `AdaGIM` all solve the task: a single informative bit at the chain head can be encoded into the fixed point and read back at every evaluated depth. At $K = 2, 3$, IGNN, EIGNN, and MGNNI approach chance, while `AdaGIM` attains the highest mean accuracy. GCNII, GIND, and Monotone-MIGNN retain above-chance performance at $K = 2$. Because chain length is fixed, increasing K raises the demand to combine several inputs and retain their parity. The gap between $K = 1$ and $K > 1$ therefore shows why success on single-source propagation is not sufficient to assess more complex long-range computation.

Branching dependencies: GLoRa. We additionally evaluate GLoRa (Zhou et al., 2025) using its authors' GraphGym framework: 2,000 directed acyclic graphs at each dependency depth $d \in \{3, \ldots, 15\}$, shared generated datasets and splits, 500 training epochs, validation-loss selection, and five seeds. Table 7 in Appendix G reports selected depths. At depth 15, `AdaGIM` retains accuracy 0.98, followed by SGC at 0.91, while the three gated baselines reach 0.50–0.52. Both `AdaGIM` and SGC remain above the 0.90 collapse threshold throughout the evaluated range. The separation at larger depths shows that similar shallow-task accuracy can conceal markedly different long-range behavior, even among gated architectures.

5.3 Selection components and readout

The ablations in Tables 3 and 4 use a common configuration within each comparison, separate from Table 1. Protocol details are in Appendix G. Learned gating improves all three datasets over residual-only gating, suggesting that prediction-relevant states depend on the task as well as convergence progress. The full gate leads on Peptides-func, while MLP-only gating gives lower MAE on ZINC and Peptides-struct. Thus the residual factor's finite-budget contribution is task-dependent, complementing its theoretical role in preserving earlier selections.

Table 4: Trajectory-readout comparison in the ablation configuration.

Readout	Peptides-func AP ↑	ZINC MAE ↓	Peptides-struct MAE ↓
Last-write	**0.5003**	**0.4523**	**0.3078**
Trajectory mean (JK-style)	0.4049	0.4954	0.3368
Fixed depth, $Z = H^T$	0.4389	0.4780	0.3337
Equilibrium, $Z = H^\star$	0.4407	0.4785	0.3350

Last-write selection outperforms trajectory averaging, fixed-depth readout, and equilibrium readout on all three datasets (Table 4). The advantage over trajectory averaging suggests that selecting which stages contribute is important, even when intermediate states are already accessible. Here, the JK-style baseline uses trajectory averaging.

5.4 Influence retention and computational cost

Influence on trained real-graph models. We perturb a source feature on Peptides-func and group $\|h_v(X + \Delta X_u) - h_v(X)\|/\|\Delta X_u\|$ by shortest-path distance. `AdaGIM` has the largest response at every displayed distance (Table 8, Appendix G), consistent with better retention of distant influence. Its larger fitted retention factor further indicates slower attenuation across the measured distances, complementing the comparison of absolute response magnitudes. The measurements support the connection between our diagnosis and real-graph behaviour, while also reflecting topology, normalization, embedding scale, and optimization. Appendix G gives the available measurement details.

Training cost. Table 9 measures the Peptides-func configurations from Table 1 on one NVIDIA A30 GPU, using five epochs after an untimed warm-up and peak allocated GPU memory. The comparison also includes GINE (Hu et al., 2020b). `AdaGIM` uses 14.78 seconds/epoch and 3,130 MB peak GPU memory versus 3.94 seconds and 150 MB for IGNN (Table 9, Appendix G). This trade-off makes storage of the unrolled trajectory a natural target for memory-efficient implementations of adaptive selection. Appendix G gives the configurations, measurements, and aggregate selection weights.

6 Limitations and scope

The analysis uses an aligned linear model for the simple-chain result and uniform contraction and sensitivity bounds for the long-range result. The real-graph evidence supports the proposed mechanism without isolating its causal contribution. `AdaGIM` adds a retained state and finite-unroll training cost (Table 9); developing a more memory-efficient implementation is a direction for future work.

7 Conclusion

In this paper, we revisited the role of graph implicit models (GIMs) for capturing long-range dependencies. We introduced the (K, L) chain-parity family as a generalization of the chain benchmarks used in prior GIM work, gave an aligned learning-dynamics analysis explaining generalization across lengths on the $(1, L)$ subfamily, and proved a quantitative obstruction in the form of a vanishing-influence bound (Lemma 3.5) that yields an exact-score and fixed-margin difficulty result under contraction and sensitivity bounds uniform in graph size (Theorem 3.7). Motivated by the diagnosis, we proposed `AdaGIM`, a GIM construction with per-node adaptive selection. Across synthetic chain-parity and four real-world datasets, `AdaGIM` shows competitive performance. Our work explains success on simple chains, identifies a long-range bottleneck under uniform contraction, and motivates adaptive selection in future implicit graph models.

AI USE STATEMENT

During preparation of this paper, generative AI tools assisted with literature searches, LaTeX formatting, and grammar checking. AI tools. AI tools are not involved in idea formulation, analysis, and experiments. The authors are responsible for the accuracy and integrity of the final manuscript and its supporting materials.

REPRODUCIBILITY STATEMENT

The theoretical assumptions accompany each result, with proofs in Appendices B–D. Appendix E specifies the architecture, update rule, training procedure, and finite-forward algorithm. Appendices F and G document experimental settings, evaluation protocols, and the available seed and aggregation information for each comparison. Section 5 provides the anonymous code link.

REFERENCES

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 2623–2631, 2019.

Cem Anil, Yuhuai Wu, Anders Andreassen, Aitor Lewkowycz, Vedant Misra, Vinay Ramasesh, Ambrose Slone, Guy Gur-Ari, Ethan Dyer, and Behnam Neyshabur. Exploring length generalization in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 38546–38556, 2022.

Shaojie Bai, J Zico Kolter, and Vladlen Koltun. Deep equilibrium models. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Justin Baker, Qingsong Wang, Cory D Hauck, and Bao Wang. Implicit graph neural networks: A monotone operator viewpoint. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 1521–1548. PMLR, 2023.

Víctor Campos, Brendan Jou, Xavier Giró-i-Nieto, Jordi Torres, and Shih-Fu Chang. Skip RNN: Learning to skip state updates in recurrent neural networks. In *International Conference on Learning Representations*, 2018.

Deli Chen, Yankai Lin, Wei Li, Peng Li, Jie Zhou, and Xu Sun. Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 3438–3445, 2020a.

Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. Simple and deep graph convolutional networks. In *International Conference on Machine Learning*, pp. 1725–1735. PMLR, 2020b.

Qi Chen, Yifei Wang, Yisen Wang, Jiansheng Yang, and Zhouchen Lin. Optimization-induced graph implicit nonlinear diffusion. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pp. 3648–3661. PMLR, 2022.

Akash Choudhuri, Yongjian Zhong, and Bijaya Adhikari. Implicit hypergraph neural network. In *2025 IEEE International Conference on Big Data (BigData)*, 2025.

Francesco Di Giovanni, Lorenzo Giusti, Federico Barbero, Giulia Luise, Pietro Liò, and Michael M. Bronstein. On over-squashing in message passing neural networks: The impact of width, depth, and topology. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 7865–7885, 2023.

Shaoxiong Duan, Yining Shi, and Wei Xu. From interpolation to extrapolation: Complete length generalization for arithmetic transformers. *arXiv preprint arXiv:2310.11984*, 2023.

Vijay Prakash Dwivedi, Ladislav Rampášek, Michael Galkin, Ali Parviz, Guy Wolf, Anh Tuan Luu, and Dominique Beaini. Long range graph benchmark. In *Advances in Neural Information Processing Systems*, volume 35, pp. 22326–22340, 2022. Datasets and Benchmarks Track.

Vijay Prakash Dwivedi, Chaitanya K. Joshi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Benchmarking graph neural networks. *Journal of Machine Learning Research*, 24 (43):1–48, 2023.

Laurent El Ghaoui, Fangda Gu, Bertrand Travacca, Armin Askari, and Alicia Tsai. Implicit deep learning. *SIAM Journal on Mathematics of Data Science*, 3(3):930–958, 2021.

Ying Fan, Yilun Du, Kannan Ramchandran, and Kangwook Lee. Looped transformers for length generalization. In *The Thirteenth International Conference on Learning Representations*, 2025.

Guoji Fu, Mohammed Haroon Dupty, Yanfei Dong, and Wee Sun Lee. Implicit graph neural diffusion networks: Convergence, generalization, and over-smoothing. *arXiv preprint arXiv:2308.03306*, 2023.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pp. 1263–1272. PMLR, 2017.

Alex Graves. Adaptive computation time for recurrent neural networks. *arXiv preprint arXiv:1603.08983*, 2016.

Riccardo Grazzi, Julien Siems, Arber Zela, Jörg K. H. Franke, Frank Hutter, and Massimiliano Pontil. Unlocking state-tracking in linear RNNs through negative eigenvalues. In *The Thirteenth International Conference on Learning Representations*, 2025.

Fangda Gu, Heng Chang, Wenwu Zhu, Somayeh Sojoudi, and Laurent El Ghaoui. Implicit graph neural networks. In *Advances in Neural Information Processing Systems*, volume 33, pp. 11984–11995, 2020.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In *Advances in Neural Information Processing Systems*, volume 33, pp. 22118–22133, 2020a.

Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. In *International Conference on Learning Representations*, 2020b.

Amirhossein Kazemnejad, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Payel Das, and Siva Reddy. The impact of positional encoding on length generalization in transformers. In *Advances in Neural Information Processing Systems*, volume 36, pp. 24892–24928, 2023.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. In *International Conference on Learning Representations*, 2019.

Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated graph sequence neural networks. In *International Conference on Learning Representations*, 2016.

Juncheng Liu, Kenji Kawaguchi, Bryan Hooi, Yiwei Wang, and Xiaokui Xiao. EIGNN: Efficient infinite-depth graph neural networks. In *Advances in Neural Information Processing Systems*, volume 34, pp. 18762–18773, 2021.

Juncheng Liu, Bryan Hooi, Kenji Kawaguchi, and Xiaokui Xiao. MGNNI: Multiscale graph neural networks with implicit layers. In *Advances in Neural Information Processing Systems*, volume 35, pp. 21358–21370, 2022.

Joël Mathys, Henrik Christiansen, Federico Errica, Takashi Maruyama, and Francesco Alesiani. LRIM: A physics-based benchmark for provably evaluating long-range capabilities in graph learning. In *The Fourteenth International Conference on Learning Representations*, 2026.

Junyoung Park, Jinhyun Choo, and Jinkyoo Park. Convergent graph solvers. In *International Conference on Learning Representations*, 2022.

Alexandra Maria Proca, Clémentine Carla Juliette Dominé, Murray Shanahan, and Pedro A. M. Mediano. Learning dynamics in linear recurrent neural networks. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 49860–49902, 2025.

Luana Ruiz, Fernando Gama, and Alejandro Ribeiro. Gated graph recurrent neural networks. *IEEE Transactions on Signal Processing*, 68:6303–6318, 2020. doi: 10.1109/TSP.2020.3033962.

Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *International Conference on Learning Representations*, 2014.

Indro Spinelli, Simone Scardapane, and Aurelio Uncini. Adaptive propagation graph convolutional network. *IEEE Transactions on Neural Networks and Learning Systems*, 32(10):4755–4760, 2021.

Surat Teerapittayanon, Bradley McDanel, and Hsiang-Tsung Kung. Branchynet: Fast inference via early exiting from deep neural networks. In *2016 23rd international conference on pattern recognition (ICPR)*, pp. 2464–2469. IEEE, 2016.

Loek Van Rossem and Andrew M. Saxe. Algorithm development in neural networks: Insights from the streaming parity task. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 60764–60791. PMLR, 2025.

Xin Wang, Fisher Yu, Zi-Yi Dou, Trevor Darrell, and Joseph E Gonzalez. Skipnet: Learning dynamic routing in convolutional networks. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 409–424, 2018.

Shiwen Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. Graph neural networks in recommender systems: a survey. *ACM Computing Surveys*, 55(5):1–37, 2022.

Zuxuan Wu, Tushar Nagarajan, Abhishek Kumar, Steven Rennie, Larry S Davis, Kristen Grauman, and Rogerio Feris. Blockdrop: Dynamic inference paths in residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8817–8826, 2018.

Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. Representation learning on graphs with jumping knowledge networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pp. 5453–5462, 2018.

Yongjian Zhong, Hieu Vu, Tianbao Yang, and Bijaya Adhikari. Efficient and effective implicit dynamic graph neural network. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 4595–4606, 2024.

Yongjian Zhong, Liao Zhu, Hieu Vu, and Bijaya Adhikari. Implicit subgraph neural network. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 78602–78615. PMLR, 2025.

Dongzhuoran Zhou, Evgeny Kharlamov, and Egor Kostylev. GLoRa: A benchmark to evaluate the ability to learn long-range dependencies in graphs. In *The Thirteenth International Conference on Learning Representations*, 2025.

Hattie Zhou, Arwen Bradley, Etai Littwin, Noam Razin, Omid Saremi, Joshua Susskind, Samy Bengio, and Preetum Nakkiran. What algorithms can transformers learn? a study in length generalization. In *The Twelfth International Conference on Learning Representations*, 2024.

A EXTENDED RELATED WORK

Graph Implicit Models. Implicit models define their outputs through fixed-point equations: instead of stacking a finite sequence of layers, they specify a transformation whose equilibrium solution is taken as the representation, and the forward pass amounts to finding (or characterizing) this fixed point. A prominent example is Deep Equilibrium Models (DEQs), where Bai et al. (2019) formulate sequence modeling via the fixed point of an equilibrium equation, enabling effectively infinite-depth networks with constant memory through implicit differentiation. In the graph domain, Implicit Graph Neural Networks (IGNN) (Gu et al., 2020) instantiate this principle by defining node representations as the equilibrium of a message-passing operator, using repeated graph convolution updates to propagate information and capture long-range dependencies. Convergence and well-posedness are central in such models: Park et al. (2022) consider equilibrium GNNs with linear transition maps and enforce contractiveness to guarantee existence and uniqueness of the fixed point, while El Ghaoui et al. (2021) provide a general implicit deep learning framework and discuss conditions under which implicit formulations are well posed. Relatedly, Liu et al. (2021) propose EIGNN, an infinite-depth GNN that aims to capture long-range dependencies while avoiding costly iterative solvers by deriving a closed-form solution under specific model structure, and Chen et al. (2022) connect equilibrium computation to diffusion dynamics, obtaining the fixed point by solving a convex optimization problem.

Beyond these settings, several recent works have extended GIMs to other graph-learning regimes. For example, Zhong et al. (2024) adapt GIMs to dynamic graphs by looping over time stamps, Choudhuri et al. (2025) study hypergraph learning via fixed-point hyperedge embeddings, and Zhong et al. (2025) propose an implicit formulation for capturing subgraph structure. Fu et al. (2023) analyze implicit graph diffusion and give conditions under which implicit GNNs over-smooth during training or inference, complementing our chain-parity vanishing-influence diagnosis from a different theoretical angle.

Adaptive Neural Networks. How deep a network should be is an essential problem in learning a good representation. Many papers have proposed mechanisms that allow models to adapt their effective depth to the input. Graves (2016) propose an algorithm that allows recurrent neural networks to learn how many computational steps to take for each input. Wang et al. (2018) propose SkipNet that learns lightweight gating functions that, for each input, decide which residual blocks to execute or skip, enabling adaptive depth. Related directions include learning to skip recurrent updates (Campos et al., 2018), executing only a subset of residual blocks via learned policies (Wu et al., 2018), and early-exit architectures that attach intermediate classifiers so easy examples can terminate early (Teerapittayanon et al., 2016).

On graphs, Gated Graph Sequence Neural Networks use GRU-style recurrent updates (Li et al., 2016), while Ruiz et al. (2020) introduce time, node, and edge gates for graph processes. AP-GCN uses a node-level halting unit to adapt the number of communication steps (Spinelli et al., 2021); Jumping Knowledge networks combine representations from different neighborhood ranges (Xu et al., 2018). `AdaGIM` shares the use of learned gates, with a specific motivation from the contraction analysis of GIMs: the propagation state follows the base dynamics, while a separate state selects what each node retains along the convergence trajectory. This yields an augmented implicit formulation whose initialized limit is characterized in Proposition 4.1. Tables 3 and 4 examine the gate and trajectory readout, and Table 7 compares with gated graph architectures.

Parity Task. Parity (i.e., XOR) has long served as a "minimal hard" benchmark for probing whether a model can represent and *learn* compositional, state-tracking computations. Recently, Van Rossem & Saxe (2025) study recurrent networks on the *streaming parity* task and show that, with sufficient training experience, RNNs can undergo a transition to perfect (infinite) length generalization, shedding light on how an algorithmic solution can emerge during training. Complementarily, Grazzi et al. (2025) analyze why modern linear RNNs struggle with parity and show that, under their finite-precision assumptions, nonnegative real state-transition spectra prevent parity computation over arbitrary lengths, while allowing negative eigenvalues restores this capability and improves performance on related tasks. Parity has also become a standard length-generalization stress test for Transformer-style models: Anil et al. (2022) report that naive fine-tuning often fails to extrapolate on parity to longer lengths, while appropriate scratchpad prompting can substantially improve performance. Subsequent work studies architectural and training choices that enable extrapolation,

including positional-encoding effects (Kazemnejad et al., 2023), attention-bias calibration (Duan et al., 2023), iterative computation with a repeated Transformer block (Fan et al., 2025), and the relation between algorithmic structure, scratchpad formats, and length generalization (Zhou et al., 2024).

Long-range Graph Evaluation. Graph-transfer tasks on lines, rings, and crossed rings connect source–target distance and graph topology to over-squashing (Di Giovanni et al., 2023). Our $K = 1$ setting is closely related to single-source transfer. Extending to $K > 1$ requires combining several informative inputs into a parity state, connecting graph propagation with the state-tracking tasks above. The directed chain isolates propagation along one path, supporting the analysis of contractive fixed-point GIMs. GLoRa (Zhou et al., 2025) and LRIM (Mathys et al., 2026) provide complementary controlled benchmarks for long-range graph learning. We include GLoRa results to evaluate branching dependency structures alongside the chain-parity and real-world tasks.

B Aligned learning dynamics

Let $R_l = P_l^\top P_l$ be the supervised-node mask and let $r^{(i)} = y^{(i)} - \sum_{j\geq 0} A^j x^{(i)} v^\top M^j u$. On the directed chain all sums below are finite. Differentiating the masked loss (2) gives

$$\nabla_M \mathcal{L} = -\sum_{k\geq 1} \beta_k \sum_{b=0}^{k-1} (M^b)^\top v u^\top (M^{k-1-b})^\top,$$

$$\nabla_u \mathcal{L} = -\sum_{k\geq 0} \beta_k (M^k)^\top v, \qquad \nabla_v \mathcal{L} = -\sum_{k\geq 0} \beta_k M^k u,$$

$$\beta_k = \sum_i (A^k x^{(i)})^\top R_l r^{(i)} = \sigma_{yx}^k - \sum_{j\geq 0} \sigma_{xx}^{k,j} v^\top M^j u.$$

The derivative of $v^\top M^k u$ with respect to M contains k terms, one for each occurrence of M. Under Assumption 2.1, $u = v = ae$ and $Me = M^\top e = \lambda e$. The gradient of M is proportional to $ee^\top$, and both vector gradients are proportional to e. This proves that the aligned manifold is invariant under gradient flow. Since $c = a^2$ and $\dot{M}$ changes only the corresponding eigenvalue,

$$\dot{c} = 2c \sum_{k\geq 0} \beta_k \lambda^k, \qquad \dot{\lambda} = c \sum_{k\geq 1} k \beta_k \lambda^{k-1},$$

which is (3). This flow optimizes M directly. The chain rule through $M = \gamma g(W)$ would give a different flow and enforce a different admissible spectrum.

B.1 Proof of Proposition 2.2

Proof. The two $(1, L)$ inputs are 0 and e_1, and their targets are 0 and $\mathbf{1}$. With $A_{r,r-1} = 1$ and no self-loops, $P_l A^k e_1$ is the $(k+1)$st coordinate vector for $0 \leq k < l$ and zero otherwise. Consequently,

$$\sigma_{yx}^k = \mathbf{1}\{k < l\}, \qquad \sigma_{xx}^{k,j} = \mathbf{1}\{k = j < l\}.$$

Set $P(\lambda) = \sum_{k=0}^{l-1} \lambda^k$ and $Q(\lambda) = \sum_{k=0}^{l-1} \lambda^{2k}$. Then

$$\dot{c} = 2c(P - cQ), \qquad \dot{\lambda} = c(P' - \tfrac{c}{2} Q').$$

At $(c, \lambda) = (1, 1)$ these derivatives vanish. The Jacobian there is

$$J = \begin{pmatrix} -2l & -l(l-1) \\ -l(l-1)/2 & -l(l-1)(2l-1)/6 \end{pmatrix}.$$

For $l \geq 2$, its trace is negative and $\det J = l^2(l-1)(l+1)/6 > 0$, so both eigenvalues have negative real parts. The equilibrium is locally asymptotically stable. At this point the nonzero input produces $\hat{y}_r = c\lambda^{r-1} = 1$ at every position of any longer chain, while the zero input produces zero. □

When $l = 1$, the eigenvalue is unidentifiable from supervision and there is no isolated asymptotically stable equilibrium in (c, λ). The collapsed set $c = 0$ has a positive transverse eigenvalue $2P(\lambda)$ for $\lambda \geq 0$, but this local calculation does not prove global convergence from all noncollapsed initializations. We do not make that claim. Nor does stationarity of an unconstrained surrogate establish the optimization behavior of EIGNN.

Linearity and two-bit parity. For fixed parameters, the encoder (1) and a linear readout give a linear function of the input bits. Even allowing an affine bias, its four scores at a node observing two bits satisfy $s(0,0) + s(1,1) = s(1,0) + s(0,1)$. Exact XOR targets violate this identity. A strictly separating affine threshold also cannot put both $(0,0), (1,1)$ below a threshold and both $(1,0), (0,1)$ above it. This obstruction is independent of chain length and should be distinguished from the nonlinear encoder's influence bound.

B.2 Undirected normalized chains

We also consider an undirected chain with normalized adjacency $A = \tilde{D}^{-1/2}(A_0 + I)\tilde{D}^{-1/2}$, where $\tilde{D}$ is the degree matrix of $A_0 + I$. This numerical illustration uses full-node supervision. For $|\lambda| < 1$, let $S(\lambda) = (I - \lambda A)^{-1}$ and define

$$P_0(\lambda) = \sum_i {y^{(i)}}^\top S(\lambda) x^{(i)}, \qquad Q_0(\lambda) = \sum_i \|S(\lambda) x^{(i)}\|_2^2.$$

Nonzero stationary points of the aligned surrogate satisfy

$$2P_0'(\lambda)Q_0(\lambda) - P_0(\lambda)Q_0'(\lambda) = 0, \qquad c = P_0(\lambda)/Q_0(\lambda).$$

Figure 4 shows the numerically computed stationary parameter $\lambda^*(L)$ for the $(1, L)$ task. Its drift is mild and it lies close to one for longer chains. This illustrates the dependence on normalization and chain length; the exact length-generalization result in Proposition 2.2 concerns the directed-chain setting.

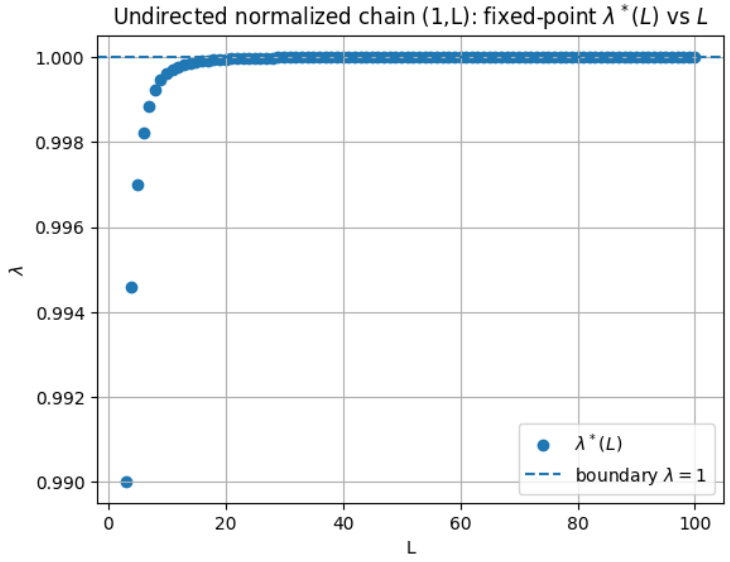


Figure 4: Stationary parameter $\lambda^*(L)$ for the aligned surrogate on undirected normalized $(1, L)$ chains with full-node supervision.

C Influence bounds and their scope

C.1 Proof of Lemma 3.5

Proof. Let $\Delta_i = \|h_i - h_i'\|$ and $\delta = \|x_1 - x_1'\|$. Add and subtract an update with unchanged features and changed hidden states. Since only the feature at node 1 changes, the assumptions give

$$(1 - B_{ii})\Delta_i \le B_{i,i+1}\Delta_{i+1} + B_{i,i-1}\Delta_{i-1} + L_x \delta \, \mathbf{1}\{i = 1\},$$

where terms for missing neighbors are zero. With $p_i = B_{i,i+1}/(1 - B_{ii})$, $q_i = B_{i,i-1}/(1 - B_{ii})$, and $b = L_x\delta/(1 - B_{11})$, this becomes

$$\Delta_1 \le p_1\Delta_2 + b, \qquad \Delta_i \le p_i\Delta_{i+1} + q_i\Delta_{i-1} \quad (2 \le i \le L).$$

Here $q_1 = p_L = 0$, and $p_i + q_i < 1$ by strict row-sum contraction.

Define $Q_1 = 1$ and $Q_k = 1 - p_{k-1}q_k/Q_{k-1}$ for $k \geq 2$. These elimination denominators are positive: $Q_1 > p_1$, and if $Q_{k-1} > p_{k-1} \geq 0$, then

$$Q_k \geq 1 - q_k > p_k \geq 0.$$

Starting with $S_1 = b$, forward elimination therefore yields

$$\Delta_k \leq \frac{p_k}{Q_k}\Delta_{k+1} + S_k, \qquad S_k = \frac{q_k}{Q_k}S_{k-1} = b\prod_{j=2}^{k}\frac{q_j}{Q_j}.$$

Indeed, substituting the bound for Δ_{k-1} into the inequality for Δ_k gives $Q_k\Delta_k \leq p_k\Delta_{k+1} + q_kS_{k-1}$. At $k = L$, $p_L = 0$, so

$$\Delta_L \leq S_L = \frac{L_x}{1 - B_{11}}\prod_{k=2}^{L}\frac{q_k}{Q_k}\delta,$$

which is (4), with $B_{11} = \alpha_{11}$. For $L = 1$, the same bound holds with the empty product equal to one. □

C.2 Proof of Remark 3.6

Proof. Suppose $p_i = p$ for $i < L$ and $q_i = q$ for $i > 1$, with $p, q \geq 0$ and $p + q < 1$. The recursion is $Q_k = 1 - pq/Q_{k-1}$. Define

$$Q_+ = \frac{1 + \sqrt{1 - 4pq}}{2}.$$

The inequality $p + q < 1$ implies $4pq < 1$ and $Q_+ > q$. To verify the latter, it is immediate for $q \leq 1/2$; for $q > 1/2$, squaring $2q - 1 < \sqrt{1 - 4pq}$ reduces it to $p + q < 1$. Since $Q_1 = 1 \geq Q_+$ and $z \mapsto 1 - pq/z$ is increasing on $(0, \infty)$ with fixed point Q_+, induction gives $Q_k \geq Q_+$ for every k. Consequently,

$$\prod_{k=2}^{L}\frac{q}{Q_k} \leq \left(\frac{q}{Q_+}\right)^{L-1}, \qquad 0 \leq \frac{q}{Q_+} < 1.$$

The product bound thus decays geometrically. Across chain lengths, fixed p, q and a uniform bound on $L_x/(1 - B_{11})$ give the stated exponential decay. The row-sum bound $B_{11} \leq \rho < 1$ supplies the latter when L_x is uniform. □

C.3 A uniform influence bound for general propagation graphs

The row-sum condition also gives a bound without homogeneous coefficients. If the inputs differ only at a source s, by $\delta = \|x_s - x'_s\|$, then

$$\|h_v - h'_v\| \leq L_x\delta\,[(I - B)^{-1}]_{vs} \leq \frac{L_x\delta}{1 - \rho}\rho^{d(s,v)}, \tag{6}$$

where $d(s, v)$ is the shortest propagation distance from s to v. An unreachable node's embedding does not change.

Proof. Let $\Delta_i = \|h_i - h'_i\|$. The assumptions give the componentwise inequality $\Delta \leq B\Delta + L_x\delta e_s$. Iterating it m times yields

$$\Delta \leq B^m\Delta + L_x\delta\sum_{k=0}^{m-1}B^k e_s.$$

Since $\|B\|_\infty \leq \rho < 1$, the first term tends to zero and the matrix series converges. Thus $\Delta_v \leq L_x\delta[(I - B)^{-1}]_{vs}$. Locality implies $(B^k)_{vs} = 0$ for $k < d(s, v)$, whereas $0 \leq (B^k)_{vs} \leq \|B^k\|_\infty \leq \rho^k$. Summing the geometric tail proves (6). If there is no path, every term is zero. For $\rho = 0$, only the source can change. □

If each finite graph has row sums below one but the maximum approaches one with graph length, its bound is graph-dependent and does not yield a single length threshold. The uniformity assumption is therefore essential to the stated asymptotic conclusion.

C.4 Proof of Theorem 3.7

Proof. A first-bit flip has $\delta = 1$ and changes the parity target at the final node. Lipschitz continuity of the decoder and (6), with $d(1, L) = L - 1$, give

$$|s_L(x) - s_L(x')| \le L_g \|h_L - h'_L\| \le \frac{L_g L_x}{1-\rho} \rho^{L-1}.$$

Exact target scores require gap one. If class-zero scores must be at most $b - \eta$ and class-one scores at least $b + \eta$, they require gap at least 2η. For any fixed required gap $m > 0$ the displayed upper bound tends to zero, so it eventually becomes smaller than m. For example, any integer $N \ge \max\{1, 1 + \log(m/D)/\log \rho\}$ suffices for all $L > N$ when $D > 0$. If $D = 0$, positive separation is impossible at every length. □

Why threshold accuracy needs a margin assumption. Small embedding differences can still lie on opposite sides of a threshold. A concrete contractive parity construction makes the distinction precise. On a directed chain let $h_1 = \cos(\pi x_1)$ and

$$h_i = \rho \cos(\pi x_i) \tanh(h_{i-1}), \qquad i \ge 2,$$

with $0 < \rho < 1$ and binary x_i. This is a local map on the complete invariant hidden-state domain $[-1, 1]^L$, with row-sum hidden-state Lipschitz constant at most ρ and feature Lipschitz constant at most $L_x = \pi$, independent of length. Its nonzero state has sign $(-1)^{x_1 + \cdots + x_i}$, so thresholding the smooth score $g(h) = -h$ at zero gives the correct parity at every finite length. Its margin vanishes with distance. This example satisfies the stability assumptions and demonstrates why the theorem cannot rule out arbitrarily small-margin classification.

C.5 Architecture-level sufficient conditions

Table 5 states sufficient conditions for the row-sum proof, rather than attributing one condition to every model called implicit.

Table 5: Scope of the nodewise influence bound. All bounds used across graph lengths must be uniform in length.

Encoder form	Sufficient coefficient bound	Scope restriction
IGNN-style $\phi(AHW + UX)$	$B_{ij} = \lvert A_{ij}\rvert \lVert W\rVert$, for a 1-Lipschitz ϕ; require $\lVert W\rVert \max_i \sum_j \lvert A_{ij}\rvert < 1$.	A spectral or Perron–Frobenius condition in another norm need not give this row-sum bound.
Linear EIGNN $AHM + Xv^\top$	$B_{ij} = \lvert A_{ij}\rvert \lVert M\rVert$; require $\lVert M\rVert \max_i \sum_j \lvert A_{ij}\rvert < 1$.	Symmetric degree normalization controls a spectral norm on undirected graphs, not necessarily each row sum.
Multi-scale propagation	Apply the test to the actual update and its propagation graph.	A step spanning multiple original edges changes the distance in the lemma; state-dependent attention needs its own Lipschitz bound.
Monotone or optimization-defined equilibria	No row-sum bound follows from the model name or uniqueness alone.	The actual local operator must satisfy the assumptions, or a separate sensitivity argument is required.
AdaGIM retained state	The backbone may satisfy a contraction test; the full selected state retains its history.	The residual gate gives identity updates in Z at equilibrium, so the augmented map is not strictly contractive.

An alternative under global contraction. A similar distance bound also follows from a length-uniform global contraction in Frobenius norm, without a row-sum certificate. Suppose F_x is local, contracts by $\rho < 1$, and changing x_s changes $F_x(H)$ only at s, with norm at most $L_x \delta$. Its equilibria

satisfy $\|H^\star_x - H^\star_{x'}\|_F \le L_x\delta/(1-\rho)$. Initialize $\widetilde{H}^0 = H^\star_{x'}$ and iterate $\widetilde{H}^{t+1} = F_x(\widetilde{H}^t)$. At a node of distance d from s, locality gives $\widetilde{H}^d_v = (H^\star_{x'})_v$: the first update changes only s, and each later update can propagate one edge further. Contraction therefore gives

$$\|(H^\star_x)_v - (H^\star_{x'})_v\| \le \|H^\star_x - \widetilde{H}^d\|_F \le \frac{L_x\delta}{1-\rho}\rho^d.$$

This covers, for example, a linear encoder with $\|A\|_2\|M\|_2 < 1$ uniformly in graph size. Locality and a strict uniform contraction are still required; existence and uniqueness alone are insufficient.

D THE EQUILIBRIUM SELECTED BY INITIALIZATION

D.1 CONVERGENCE AND FINITE-UNROLLING ACCURACY

For the quantitative guarantee, suppose $\bar{f}_\theta$ contracts with constant $0 < r < 1$ on a complete invariant state space, in a norm dominating each node's Euclidean norm. Fix the graph, parameters, damping, and both initial states. Let $C = \|H^1 - H^0\|$, and let $B < \infty$ bound the nodewise propagation and retained states. Then the initialized selection satisfies

$$\max_v \|Z^\infty_{\iota,v} - Z^T_v\| \le \frac{2BC^2}{\tau_q^2(1-r^2)} r^{2T}. \tag{7}$$

The convergence assertion in Proposition 4.1 requires only convergent propagation and continuity at its limit; contraction supplies this additional rate. We first isolate a property of convex retention updates. All limits below are for a fixed finite graph and fixed parameters.

Lemma D.1 (Convergence of a retained state)**.** *Let $h_t \to h_\star$ in a finite-dimensional normed space, $a_t \in [0,1]$, and $z_t = (1-a_t)z_{t-1} + a_t h_t$. Then z_t converges. If $\sum_t a_t = \infty$, its limit equals $h_\star$; if $\sum_t a_t < \infty$, it can depend on earlier states and on z_0.*

Proof. The sequence h_t is bounded, and each z_t is a convex combination of $z_0, h_1, \dots, h_t$. Choose $B < \infty$ bounding both sequences. If $\sum_t a_t < \infty$, then

$$\sum_{t\ge1} \|z_t - z_{t-1}\| \le 2B \sum_{t\ge1} a_t < \infty,$$

so z_t is Cauchy. Otherwise, for any $\varepsilon > 0$ choose n such that $\|h_t - h_\star\| \le \varepsilon$ for $t > n$. Iterating the update gives

$$\|z_t - h_\star\| \le \varepsilon + \prod_{k=n+1}^{t} (1-a_k)\|z_n - h_\star\|.$$

The product is at most $\exp(-\sum_{k=n+1}^t a_k)$ and tends to zero. Letting $\varepsilon \downarrow 0$ gives $z_t \to h_\star$. □

Proof of Proposition 4.1. Contraction on a complete invariant state space gives a unique propagation fixed point $H^\star$ and convergence of H^t to it. More generally, the same argument below needs only $H^t \to H^\star$ and continuity of $\bar{f}_\theta$ at that limit, which imply $\bar{f}_\theta(H^\star) = H^\star$. Lemma D.1 gives convergence of each node's retained state to $Z^\infty_{\iota,v}$. The actual augmented update is

$$F(H,Z) = \left(\bar{f}_\theta(H),\ a(\bar{f}_\theta(H),H) \odot \bar{f}_\theta(H) + (1 - a(\bar{f}_\theta(H),H)) \odot Z\right).$$

Its residual gate vanishes when its two propagation arguments coincide, so $(H^\star, Z^\infty_\iota)$ is a fixed point. The prescribed initialization and deterministic rule fix every iterate and hence its limit. A convergent sequence's closure consists of its iterates and limit. If any finite iterate is a fixed point, all subsequent iterates equal it, so it equals the limit. Thus there is exactly one fixed point in the closure of this initialized forward orbit.

For the quantitative result, set $C = \|H^1 - H^0\|$. Contraction and $1 - e^{-b} \le b$ for $b \ge 0$ give

$$\|H^{t+1} - H^t\| \le Cr^t, \qquad 0 \le a_v^{t+1} \le \frac{C^2 r^{2t}}{\tau_q^2}.$$

The gates are therefore summable. An explicit uniform bound on the nodewise propagation and retained states is

$$B = \max\left\{\max_v \|Z_v^0\|_2,\ \max_v \|H_v^0\|_2 + \frac{C}{1-r}\right\},$$

because the propagation increments sum to at most $C/(1-r)$ and retention is convex. Consequently,

$$\|Z_{\iota,v}^{\infty} - Z_v^T\| \le \sum_{t=T+1}^{\infty} \|Z_v^t - Z_v^{t-1}\| \le 2B \sum_{t=T+1}^{\infty} a_v^t \le \frac{2BC^2 r^{2T}}{\tau_q^2(1-r^2)}.$$

This proves (7). The propagation error separately satisfies $\|H^T - H^\star\| \le Cr^T/(1-r)$, so the full pair has an $O(r^T)$ bound and the retained component has the sharper $O(r^{2T})$ bound. □

A residual-based stopping certificate. Restarting the increment bound at T gives

$$\max_v \|Z_{\iota,v}^{\infty} - Z_v^T\| \le \frac{2B}{\tau_q^2(1-r^2)} \|H^{T+1} - H^T\|^2. \tag{8}$$

For $T \ge 1$, the numerator may instead use $2Br^2\|H^T - H^{T-1}\|^2$, requiring only the last computed increment. These certificates require valid bounds on B, r, τ_q. A relative-change tolerance by itself does not supply those constants. The result applies to each core with its prescribed initialization; an error bound for the entire finite stack would additionally propagate errors between cores.

D.2 Persistent trajectory memory and continuous dependence

Proposition D.2 (Infinite last-write representation)**.** *Under the contraction conditions in Appendix D.1,*

$$Z_{\iota,v}^{\infty} = \omega_v^{0,\infty} Z_v^0 + \sum_{t=1}^{\infty} \omega_v^{t,\infty} H_v^t,$$

$$\omega_v^{0,\infty} = \prod_{s=1}^{\infty}(1-a_v^s), \qquad \omega_v^{t,\infty} = a_v^t \prod_{s=t+1}^{\infty}(1-a_v^s).$$

These weights are nonnegative and sum to one. With finite states and gate logits, $\omega_v^{0,\infty} > 0$, and every step with $a_v^t > 0$ has $\omega_v^{t,\infty} > 0$.

Proof. Unroll the finite recurrence. Summability of a_v^t implies convergence of the displayed products, and

$$\sum_{t=0}^{N} \omega_v^{t,\infty} = \prod_{s=N+1}^{\infty}(1-a_v^s) \longrightarrow 1.$$

The propagation states are bounded; the mass beyond N is at most $\sum_{s>N} a_v^s$, for both the finite and infinite mixtures. Their tails therefore vanish uniformly, justifying the limit and absolute convergence of the series. For finite states and logits, $a_v^t < 1$. Eventually $a_v^t \le 1/2$, so $-\log(1-a_v^t) \le 2a_v^t$; summability makes the full product strictly positive. The same argument applies to each tail product. □

Prescribing both initial states is substantive. For fixed H^0, two choices of Z^0 have the same propagation and gate sequences, and their limits differ by exactly

$$Z_{\iota,v}^{\infty} - Z_{\iota',v}^{\infty} = \omega_v^{0,\infty}(Z_v^0 - Z_v'^0).$$

The implementation fixes $Z^0 = 0$ and penalizes its no-write mass. In contrast, if the residual factor is removed, continuity and finite logits imply that the sigmoid-only gate converges to a value in $(0,1)$. Its total write mass is infinite; Lemma D.1 then gives $Z_v^{\infty} = H_v^\star$. Thus the residual factor supports a distinct limiting representation by preserving nonzero weights on earlier writes. This is an asymptotic mechanism, not a claim of uniformly better finite-budget prediction.

Corollary D.3 (Continuous dependence of the selected equilibrium)**.** *On a fixed finite graph, let the inputs and parameters vary over a compact set. Suppose the propagation maps, gates, and prescribed initializations depend continuously on these variables and states. If the contraction factors are uniformly at most $r < 1$, $\tau_q \geq \tau_{\min} > 0$, and B, C have common finite bounds, then Z_ι^∞ depends continuously on the inputs and parameters.*

Proof. Every fixed-budget iterate Z^T is continuous by induction. Equation (7) bounds its error uniformly over the stated set, so the selected limit is a uniform limit of continuous functions. □

This gives existence, uniqueness of the selected limit, and continuous dependence under explicit conditions. It does not assert continuity across changes in graph size or at boundaries where a discrete stopping rule changes the iteration count.

Scope of the initialized formulation. The bare algebraic map admits $F(H^\star, Z) = (H^\star, Z)$ for any Z; Proposition 4.1 selects one member through initialization and the update rule. At these equilibria the retained-state block satisfies $D_Z F_Z = I$, making $I - DF$ singular wherever the full derivative exists. Conventional implicit differentiation of an arbitrary augmented root therefore does not recover the trajectory-dependent derivative. The implemented finite-unroll differentiation preserves this dependence. Likewise, changing damping or substituting another solver's intermediate states can change the selected representation even when the propagation root is unchanged.

Limits of finite selection. For an input-independent selection time, unrolling the same uniformly contractive local map still attenuates remote perturbations. From a shared, input-independent initial state, its influence satisfies $\Delta^T \leq L_x \delta \sum_{k=0}^{T-1} B^k e_s$, bounded by the same infinite geometric tail (here B is the influence matrix of Assumption 3.3). Learned input-dependent gates introduce another sensitivity pathway. The existence and uniqueness results above do not by themselves prove a fixed positive parity margin on arbitrarily long chains.

D.3 A SUFFICIENT CONDITION FOR GLOBAL EQUILIBRIUM UNIQUENESS

The following sufficient condition applies when the selection map is strictly contractive in its retained state. It complements the prescribed-initialization guarantee for the practical residual gate.

Proposition D.4 (Well-posedness condition for the augmented map)**.** *Suppose f_θ is contractive with constant $\rho_f < 1$ and*

$$\|g_\xi(H, Z) - g_\xi(H', Z')\| \leq L_H \|H - H'\| + \rho_Z \|Z - Z'\|, \qquad L_H < \infty, \quad 0 \leq \rho_Z < 1.$$

On a complete invariant product state space, $F(H, Z) = (f_\theta(H), g_\xi(f_\theta(H), Z))$ is contractive in an equivalent product norm and admits a unique fixed point.

Proof. Choose $c \in (\max\{\rho_f, \rho_Z\}, 1)$ and $\alpha > 0$ large enough that $L_H \rho_f / \alpha + \rho_Z \leq c$. In the norm $\|(H, Z)\|_\alpha = \max\{\|H\|, \alpha^{-1}\|Z\|\}$, the two output differences are bounded by ρ_f and $L_H \rho_f / \alpha + \rho_Z$ times the input difference, respectively. Thus F is c-contractive, and the Banach fixed-point theorem gives the result. □

The practical residual gate permits zero acceptance, so it uses the initialized-trajectory guarantee rather than this strict-selection condition.

E ADAGIM ARCHITECTURE AND TRAINING DETAILS

BACKBONE. Let $S = C - C^\top$ and $W = s_W (I - S)(I + S)^{-1}$ with $0 < s_W < 1$. The Cayley factor is orthogonal, so $\|W\|_2 = s_W$. We initialize $C \sim \mathcal{N}(0, 0.01^2)$ and use

$$f_\theta(H) = \mathrm{LN}_\theta\Big(\phi(\hat{A}^\top H W + U_X)\Big), \qquad \hat{A} = D^{-1/2}(A + I)D^{-1/2}.$$

For an adjacency convention in which A_{ij} is an edge from i to j, the transpose aggregates incoming neighbors; it has no effect for an undirected graph. The Picard update is $H^{t+1} = (1 - \alpha_P)H^t + \alpha_P f_\theta(H^t)$. Iteration stops at a finite budget or a relative-change tolerance.

LayerNorm is optional (the `row_norm` flag). The Cayley constraint alone does not guarantee contraction of the complete operator. Without LayerNorm, an undirected symmetrically normalized adjacency and a 1-Lipschitz ϕ give $\mathrm{Lip}(f) \le s_W < 1$ in Frobenius norm. For directed graphs one must check $\|\hat{A}\|_2 s_W < 1$. With LayerNorm, a sufficient bound includes its Lipschitz constant: $L_{\mathrm{LN}}\|\hat{A}\|_2 s_W < 1$. Affine LayerNorm has no automatic bound of one. For fixed normalization offset $\epsilon_{\mathrm{LN}} > 0$ and affine scale vector γ_{LN}, a conservative global bound is $L_{\mathrm{LN}} \le \|\gamma_{\mathrm{LN}}\|_\infty / \sqrt{\epsilon_{\mathrm{LN}}}$. We do not claim that the reported configurations enforce this additional constraint. A finite iteration cap makes the computed forward map defined even when a convergence certificate is unavailable.

GATE AND RETAINED-STATE WEIGHTS. The nodewise gate MLP is $\mathrm{Lin}(2d \to d) \to \mathrm{GELU} \to \mathrm{Lin}(d \to 1)$. Weights are initialized from $\mathcal{N}(0, 0.01^2)$ and the last bias to -3. The temperature $\tau_g > 0$ is fixed; the residual bandwidth is $\tau_q = \exp(\rho_q) > 0$, with default initialization $\rho_q = \log 0.3$. The gate uses the pair $[H_v^{t+1}, H_v^t]$ and the residual mask in (5). With $Z^0 = 0$, the nonnegative last-write weights $\omega^0, \dots, \omega^T$ sum to one per node. A hard gate that writes at time v and never subsequently writes would yield $Z_v = H_v^v$. The implemented soft residual gate need not attain this hard selector exactly, and the empirical chain-diagonal rule is not assumed to emerge from training.

STACKING AND READOUT. For J cores, each core has independent $(C_\ell, \mathrm{MLP}_{\xi,\ell}, \rho_{q,\ell}, \mathrm{LN}_{\theta,\ell})$, while all share U_X. Initialize $H_1^0 = 0$ and warm-start $H_\ell^0 = Z_{\ell-1}^{T_{\ell-1}}$ for $\ell > 1$; each retained state starts at zero. The readout is $\mathrm{Lin}(\mathrm{Pool}_v Z_{J,v}^{T_J})$ for graph tasks, with mean or sum pooling, and $\mathrm{Lin}(Z_{J,v}^{T_J})$ for node tasks. Optional dropout precedes the linear head.

REGULARIZATION. The auxiliary terms, summed across cores, are

$$\mathcal{L}_{\mathrm{sharp}} = \frac{\lambda_{\mathrm{sharp}}}{n} \sum_v \sum_{t=0}^{T} -\omega_v^t \log(\omega_v^t + \varepsilon),$$

$$\mathcal{L}_{\mathrm{nowrite}} = \frac{\lambda_{\mathrm{nowrite}}}{n} \sum_v \omega_v^0, \qquad \mathcal{L}_{\mathrm{bin}} = \frac{\lambda_{\mathrm{bin}}}{nT} \sum_{v,t} \widetilde{\sigma}_v^t (1 - \widetilde{\sigma}_v^t),$$

with $\varepsilon = 10^{-12}$. The objective is the task loss plus these terms. The reported configurations use $\lambda_{\mathrm{bin}} = 0$. The first term encourages a concentrated last-write distribution, and the second penalizes retaining the initial zero state.

REPORTED SEARCH SPACE. The reported six-trial Optuna search uses hidden dimension at most 128, at most three cores, $T \le 30$, $s_W \in \{0.7, 0.9, 0.99\}$, $\alpha_P \in \{0.5, 1\}$, $\phi \in \{\tanh, \mathrm{ReLU}\}$, optional row normalization, initial $\rho_q \in \{\log 0.1, \log 0.3, \log 1\}$, $\tau_g \in \{0.5, 1, 2\}$, $\lambda_{\mathrm{sharp}} \in \{0, 0.005, 0.01\}$, $\lambda_{\mathrm{nowrite}} \in \{0, 0.01, 0.05\}$, mean or sum pooling, head dropout in $\{0, 0.1, 0.2\}$, learning rate in $\{10^{-3}, 5 \cdot 10^{-4}, 10^{-4}\}$, and weight decay in $\{10^{-5}, 10^{-4}\}$. Six trials sample only a small part of this space; they do not establish that each architecture has been comparably optimized.

COMPUTATION AND DIFFERENTIATION. Each propagation step costs $O(|E|d + Nd^2)$, and the gate adds $O(Nd^2)$. The retained state adds $O(Nd)$ forward storage. With J cores and budget T, the forward cost is $O(JT(|E|d + Nd^2))$ apart from parameter transforms and readout. Straightforward backpropagation through the finite trajectory uses memory proportional to the stored unrolling, rather than the constant-depth activation memory of a conventional implicit backward solve. We use automatic differentiation through the finite outer iterations. The memory cost depends on which cores remain attached to the computational graph and on checkpointing; the finite-unrolling formulation does not provide a constant-depth activation-memory guarantee.

F EXPERIMENTAL SETUP

This section describes the documented setup for the reduced graph-task protocol. Appendix G specifies the available seed and tuning information for Table 1 and distinguishes the supplementary protocols. Complete selected configurations and run identifiers are unavailable.

Algorithm 1 `AdaGIM` finite forward pass

Require: graph (A, X), cores J, budget T, damping α_P, tolerance $\epsilon_{\text{stop}} > 0$

$\hat{A} \leftarrow D^{-1/2}(A + I)D^{-1/2}$; $U_X \leftarrow \text{InputEmbed}(X)$
$H^{(0)} \leftarrow 0$; $\mathcal{L}_{\text{aux}} \leftarrow 0$
for $\ell = 1$ to J **do**
  $S_\ell \leftarrow C_\ell - C_\ell^\top$; $W_\ell \leftarrow s_W(I - S_\ell)(I + S_\ell)^{-1}$
  $H \leftarrow H^{(0)}$; $Z \leftarrow 0$; $\Sigma \leftarrow [\,]$
  for $t = 1$ to T **do**
    $\widetilde{H} \leftarrow \text{LN}_\ell(\phi(\hat{A}^\top H W_\ell + U_X))$ ▷ LayerNorm optional
    $H' \leftarrow (1 - \alpha_P)H + \alpha_P \widetilde{H}$
    $q_v \leftarrow 1 - \exp(-\|H'_v - H_v\|_2^2 / \tau_q^2)$ for each node v
    $\widetilde{\sigma}_v \leftarrow \text{sigmoid}(\text{MLP}_{\xi,\ell}([H'_v, H_v])/\tau_g)$
    $a \leftarrow q \odot \widetilde{\sigma}$; $Z \leftarrow a \odot H' + (1 - a) \odot Z$; append a to Σ
    $r \leftarrow \|H' - H\|_F / \max\{\|H\|_F, \epsilon_{\text{den}}\}$ ▷ $\epsilon_{\text{den}} > 0$ avoids division by zero
    $H \leftarrow H'$
    if $r < \epsilon_{\text{stop}}$ **then**
      break
    end if
  end for
  Compute $\omega^0, \ldots, \omega^t$ from Σ and add the auxiliary penalties
  $H^{(0)} \leftarrow Z$
end for
return $\text{Head}(\text{Pool}_v Z_v)$ and $\mathcal{L}_{\text{aux}}$ ▷ node tasks omit pooling

Architecture and iteration budgets. The recorded explicit GNN configurations use hidden dimension 128; their exact depths are not listed. The recorded implicit-model hidden dimensions are at most 128, with EIGNN restricted to 64 because of reported memory use on Peptides batches. The appendix lists iteration budgets of 50 for IGNN/EIGNN/MGNNI, 16 for GIND, 20 for Monotone-MIGNN, and 30 for `AdaGIM`. EIGNN is also described as using an eigendecomposition-based closed-form encoder; the role of its listed iteration budget therefore requires implementation verification. These settings do not establish matched parameter counts or matched computation.

Search and selection. The documented search uses six Optuna TPE trials (Akiba et al., 2019) per GIM and dataset. Shared ranges are hidden dimension $\{64, 96, 128\}$, number of cores $\{1, 2, 3\}$, optional row normalization, mean/sum pooling, head dropout $\{0, 0.1, 0.2\}$, learning rate $\{10^{-3}, 5 \cdot 10^{-4}, 10^{-4}\}$, and weight decay $\{10^{-5}, 10^{-4}\}$. For GIND, LayerNorm/InstanceNorm substitutes for the normalization option. Additional model-specific choices include contraction scales, activation, and gate parameters; the `AdaGIM` range appears in Appendix E. Selection minimizes validation MAE for ZINC/Peptides-struct and maximizes validation AP or ROC-AUC for Peptides-func/MolHIV. Test scores are interpreted at the validation-selected configuration and checkpoint.

Tuning allocations for explicit baselines are described in Appendix G. Six trials sample a subset of the search space.

Training and aggregation. The documented reduced protocol uses Adam (Kingma & Ba, 2015), batch size 128, validation early stopping with patience 30, at most 200 epochs for ZINC, and at most 100 epochs for Peptides and MolHIV. Seed and aggregation information for Table 1 is consolidated in Appendix G. The chain comparison separately reports means and standard deviations over three seeds under the held-out-position protocol in Section 5.2.

Hardware and reproducibility. The experiments use a server with eight NVIDIA A30 GPUs (24 GB each), with individual graph-task runs on a single GPU. Table 9 gives the timing and memory comparison and its measurement protocol. Total compute is not reported. The stronger-budget Peptides and GLoRa evaluations use the separate protocols in Appendix G.

G SUPPLEMENTARY EMPIRICAL EVIDENCE

This appendix gives supplementary evaluation protocols, measurements, and selection-weight summaries. Results from different configurations are presented separately.

Run metadata. The experiment documentation gives conflicting seed counts for Table 1: five in its original caption, versus three for explicit baselines and a single seed (0) for validation-selected GIM configurations in the experimental setup. The setup also describes explicit baselines without Optuna tuning, whereas supplementary records describe six trials per model. Thus, the seed count, aggregation of the displayed spreads, and tuning allocation for this table remain unverified; complete selected configurations and run identifiers are unavailable. Appendix F gives the available settings. Table 2 uses the length-20 held-out-position protocol specified in Section 5.2; separate short-to-long protocol descriptions are not associated with verified results in that table.

G.1 STRONGER TRAINING PROTOCOL AND GLORA

Table 6 uses approximately 500K parameters, 250 epochs, AdamW, a warmup–cosine schedule, GeLU, dropout 0.1, and a three-layer prediction head. The exact `AdaGIM` parameter count and remaining selected hyperparameters are not documented. Seed count and variability were not reported for this comparison.

Table 6: Stronger-protocol Peptides-func comparison.

Model	Parameters	AP ↑
GCN	498,986	0.6327
`AdaGIM`	approximately 498K	0.6641

Table 7 reports GLoRa (Zhou et al., 2025) results using the benchmark authors' GraphGym framework and data-generation procedure: 2,000 directed acyclic graphs at each dependency depth $d \in \{3, \ldots, 15\}$, shared generated datasets and splits, 500 training epochs, validation-loss selection, and five seeds. Variability was not reported. "Collapse depth" is the first evaluated depth with accuracy below 0.90; it need not be one of the displayed columns.

Table 7: GLoRa test accuracy over five seeds. Collapse depth is the first evaluated depth with accuracy below 0.90.

Model	$d = 3$	$d = 9$	$d = 12$	$d = 13$	$d = 15$	Collapse depth
GCN	1.00	1.00	0.99	0.96	0.69	15
GAT	1.00	0.87	0.86	0.62	0.64	9
SAGE	1.00	1.00	0.78	0.60	0.51	12
GIN	1.00	1.00	0.71	0.54	0.49	11
SGC	1.00	1.00	0.97	0.94	0.91	> 15
GGNN	1.00	0.77	0.51	0.56	0.52	6
GatedGCN	1.00	1.00	0.78	0.70	0.50	12
GatedGCN + LapPE	1.00	1.00	1.00	0.67	0.50	13
`AdaGIM`	1.00	0.99	0.99	0.99	0.98	> 15

G.2 COMPONENT AND READOUT CONFIGURATIONS

Tables 3 and 4 use a common configuration within each comparison. Their full-model values differ from Table 1; the exact configuration change is not documented, so these comparisons are presented separately. Seed counts, variability, and the fixed unrolling depth of the readout comparison are not documented.

G.3 INFLUENCE ON TRAINED REAL-GRAPH MODELS

Table 8 groups normalized Peptides-func perturbation responses by shortest-path distance. Perturbation magnitude and distribution, graph/node sampling, aggregation weights, embedding normalization, seed count, and the retention fit's range and uncertainty are not documented. Cross-model response magnitudes can depend on embedding scale; gate removal also changes the readout and optimization. The main-text interpretation accounts for these factors.

Table 8: Normalized perturbation responses on Peptides-func. Retention is a fitted per-hop factor, rather than an operator contraction constant.

Model	$d = 4$	$d = 6$	$d = 8$	Retention
IGNN	9.11×10^{-4}	1.49×10^{-5}	1.42×10^{-7}	0.109
EIGNN	1.19×10^{-2}	7.40×10^{-4}	2.48×10^{-5}	0.226
Monotone-MIGNN	2.07×10^{-2}	1.94×10^{-3}	2.52×10^{-4}	0.304
MGNNI	1.14×10^{-1}	2.03×10^{-2}	3.79×10^{-3}	0.451
`AdaGIM`	1.96×10^{-1}	4.00×10^{-2}	6.68×10^{-3}	0.496

G.4 TRAINING COST AND AGGREGATE SELECTION WEIGHTS

Table 9 measures the Peptides-func configurations from Table 1 on one NVIDIA A30 GPU: five epochs after an untimed warm-up, with peak allocated memory measured by `torch.cuda.max_memory_allocated()`. Timing variability, inference measurements, complete parameter counts, and synchronization/reset details were not documented.

Table 9: Peptides-func training cost. Memory is peak allocated GPU memory in MB.

Model	Seconds/epoch	Peak memory (MB)
GCN	2.69	198
GINE	2.61	328
APPNP	2.99	156
IGNN-finite	2.98	259
GCNII	3.25	305
IGNN	3.94	150
GIND	7.90	164
MGNNI	9.70	111
Monotone-MIGNN	11.92	163
EIGNN	12.71	2,022
`AdaGIM`	14.78	3,130

In this comparison `AdaGIM` requires about $3.75\times$ IGNN's epoch time and $20.9\times$ its allocated memory. The extra state and retained unrolled trajectory have a material cost; the implementation does not inherit the constant-memory training property of implicit differentiation.

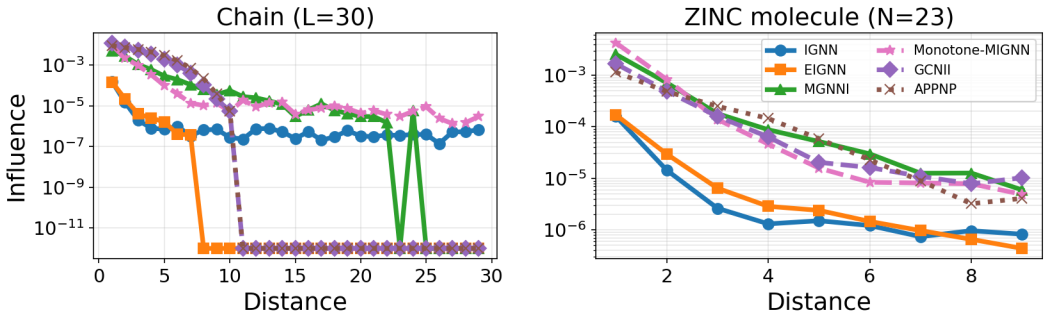
Table 10 summarizes aggregate readout weights. The averaging convention and treatment of the no-write mass ω^0 are unspecified; the tail percentages approximately equal one minus the first five entries. The available chain reference value "uniform $1/T$" is 0.032, which does not identify a horizon consistent with $K_{\max} \leq 30$, so no uniform-horizon comparison is made. The aggregate distributions show use of multiple trajectory states; they do not measure variation between nodes.

Table 10: Aggregate readout mass. The averaging and no-write normalization conventions are unspecified.

Dataset	$t=1$	$t=2$	$t=3$	$t=4$	$t=5$	Reported $t>5$
Chain, $K=1$	0.052	0.069	0.066	0.050	0.051	71%
Chain, $K=2$	0.044	0.024	0.032	0.033	0.048	82%
Chain, $K=3$	0.065	0.019	0.020	0.031	0.046	82%
Peptides-func	0.170	0.088	0.065	0.058	0.060	56%
Peptides-struct	0.201	0.140	0.106	0.085	0.070	40%
ZINC	0.438	0.190	0.082	0.054	0.044	19%
OGBG-MolHIV	0.449	0.211	0.145	0.076	0.040	8%

G.5 Influence on chains and molecular graphs

Figure 5 shows the influence diagnostics on a length-30 chain and a ZINC molecule. Influence is measured by the change in embeddings induced by a source-node perturbation and plotted against distance from that source. The curves illustrate attenuation over distance, with model-dependent plateaus and numerical floors. These diagnostics complement the trained Peptides-func measurements in Table 8.



(a) Length-30 chain

(b) ZINC molecule

Figure 5: Embedding influence versus distance on a chain (left) and a ZINC molecule (right). The ZINC panel is also shown in Figure 2.