
Expander Graph Propagation

Anonymous Author(s)

Anonymous Affiliation

Anonymous Email

Abstract

1
2 Deploying graph neural networks (GNNs) on whole-graph classification or regres-
3 sion tasks is known to be challenging: it often requires computing node features
4 that are mindful of both local interactions in their neighbourhood and the global
5 context of the graph structure. GNN architectures that navigate this space need
6 to avoid pathological behaviours, such as bottlenecks and oversquashing, while
7 ideally having linear time and space complexity requirements. In this work, we
8 propose an elegant approach based on propagating information over *expander*
9 *graphs*. We leverage an efficient method for constructing expander graphs of a
10 given size, and use this insight to propose the EGP model. We show that EGP is
11 able to address all of the above concerns, while requiring minimal effort to set
12 up, and provide evidence of its empirical utility on relevant graph classification
13 datasets and baselines in the Open Graph Benchmark. Importantly, using expander
14 graphs as a template for message passing necessarily gives rise to negative cur-
15 vature. While this appears to be counterintuitive in light of recent related work
16 on oversquashing, we theoretically demonstrate that negatively curved edges are
17 likely to be **required** to obtain scalable message passing without bottlenecks. To
18 the best of our knowledge, this is a previously unstudied result in the context of
19 graph representation learning, and we believe our analysis paves the way to a novel
20 class of scalable methods to counter oversquashing in GNNs.

21 1 Introduction

22 Graph neural networks (GNNs) are a flexible class of models for learning representations over
23 graph-structured data [1]. Their versatility [2–4] and generality [5, 6] has made them a very attractive
24 approach, leading to considerable application in areas as diverse as virtual drug screening [7], traffic
25 prediction [8], combinatorial chip design [9] and pure mathematics [10, 11].

26 Most GNNs rely on repeatedly propagating information between neighbouring nodes in the graph.
27 This is commonly expressed in the *message passing* [4] paradigm: nodes send vector-based *messages*
28 to each other along the edges of the graph, and nodes update their representations by *aggregating*
29 all the messages sent to them, in a permutation-invariant manner. Under many industrially-relevant
30 tasks, this paradigm is very potent, often allowing for highly scalable model variants [12–14].

31 However, in many areas of scientific interest, purely local interactions are likely insufficient. Among
32 the principal graph tasks, *graph classification* is perhaps most ripe with such situations: to meaning-
33 fully attach a label to a graph, in many cases it is insufficient to treat graphs as “bags of nodes”. For
34 example, when classifying a molecule for its potency as a candidate drug [7], the label is driven by
35 complex substructure interactions in the molecule [15], rather than a naïve sum of atom-level effects.

36 Accordingly, GNNs deployed in this regime need to update node features in a manner that is mindful
37 of the *global* properties of the graph. It quickly became apparent that it is often inadequate to merely
38 stack more message passing layers over the input graph. In fact, for many graph classification tasks,
39 such approaches may be weaker than discarding the graph structure altogether [16, 17]. Now, it is
40 well-understood that stacking many local layers leaves GNNs vulnerable to pathological behaviours
41 such as oversquashing [18]. Intuitively, oversquashing occurs when nodes need to store quantities of
42 information that are *exponentially* increasing with model depth [18, Section 5]. Such nodes often arise
43 in the vicinity of *bottlenecks* in a graph—small collections of edges which are responsible for carrying

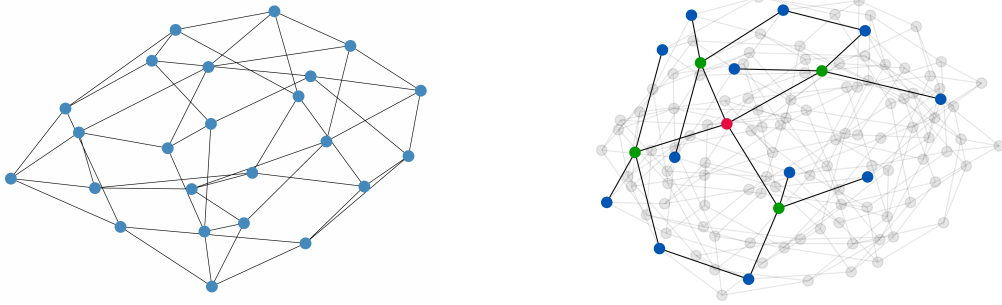
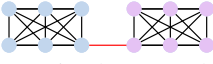


Figure 1: **Left:** The Cayley graph of $SL(2, \mathbb{Z}_3)$, constructed using our method. It has $|V| = 24$ nodes and it is 4-regular (implying $|E| = 2|V|$), hence it is sparse. Despite its sparsity, it is highly interconnected: any node is reachable from any other node by no more than 4 hops. Hence, it can serve as a strong “template” for globally propagating node features with a GNN. **Right:** The Cayley graph of $SL(2, \mathbb{Z}_5)$, constructed in an analogous way (with $|V| = 120$ nodes). A 2-hop neighbourhood of one node (in red) is highlighted, demonstrating its tree-like local structure.

44 representations between large groups of nodes. One typical example of such a bottleneck can be
 45 found in a *barbell graph* , where the red edge is under significant representational
 46 pressure to transport information between the two communities.

47 Within this space, we are interested in proposing a method that satisfies *four* desirable criteria: **(C1)** it
 48 is capable of propagating information *globally* in the graph; **(C2)** it is *resistant* to the oversquashing
 49 effect and does not introduce bottlenecks; **(C3)** its time and space complexity remain *subquadratic*
 50 (tighter than $O(|V|^2)$ for sparse graphs); and **(C4)** it requires *no dedicated preprocessing* of the input.
 51 Satisfying all four of these criteria simultaneously is challenging, and we will survey many of the
 52 popular approaches in the next section—demonstrating ways in which they fail to meet some of them.

53 In this paper, we identify *expander graphs* as very attractive objects in this regard. Specifically, they
 54 offer a family of graph structures that are fundamentally *sparse* ($|E| = O(|V|)$), while having *low*
 55 *diameter*: thus, any two nodes in an expander graph may reach each other in a short number of
 56 hops, eliminating bottlenecks and oversquashing (see Figure 1). Further, we will demonstrate an
 57 efficient way to construct a family of expander graphs (leveraging known theoretical results on the
 58 *special linear group*, $SL(2, \mathbb{Z}_n)$). Once an expander graph of appropriate size is constructed, we
 59 can perform a certain number of GNN *propagation* steps over its structure to globally distribute the
 60 nodes’ features. Accordingly, we name our method *expander graph propagation* (**EGP**).

61 A key contribution of our work extends the implications of prior art on oversquashing via curvature
 62 analysis [19]. According to [19], negatively curved edges are causing the oversquashing effect—yet,
 63 counterintuitively, the edges of the expander graphs we construct will *always be negatively curved!*
 64 We prove, however, that our expanders can never be sufficiently negatively curved to trigger the
 65 conditions necessary for the results in [19] to be applicable, and show that the existence of negatively
 66 curved edges might in fact be **required** in order to have sparse communication without bottlenecks.

67 **2 Related work**

68 We begin with a survey of the many prior approaches to handling global context in graph representation
 69 learning, evaluating them carefully against our four desirable criteria **(C1–C4)**; cf. Table 1). This list
 70 is by no means exhaustive, but should be indicative of the most important directions.

71 **Stacking more layers.** As already highlighted, one way to achieve global information propagation is
 72 to have a deeper GNN. In this case, we are capable of satisfying **(C1)** and **(C4)**—no dedicated preprocess-
 73 ing is needed. However, depending on the graph’s diameter, we may need up to $O(|V|)$ layers to
 74 cover the graph, leading to quadratic complexity (violating **(C3)**) and introducing a vulnerability to
 75 bottlenecks **(C2)**, as theoretically and empirically demonstrated in [18].

76 **Master nodes.** An attractive approach to introducing global context is to introduce a *master node*
 77 to the graph, and connect it to all of the graph’s nodes. This can be done either explicitly [4] or

Table 1: A summary of principal approaches to handling global context in graph representation learning (Section 2). “(✓)” indicates that a criterion *may* be satisfied, depending on the method’s tradeoffs. Our proposal, the expander graph propagation (EGP) method, satisfies all four criteria.

Approach	(C1)	(C2)	(C3)	(C4)
	(global prop.)	(no bottlenecks)	(subquadratic)	(no dedicated preproc.)
GNNs	✗	✗	✓	✓
Sufficiently deep GNNs	✓	✗	✗	✓
Master node [4, 20]	✓	✗	✓	✓
Fully connected [18, 21–25]	✓	✓	✗	✓
Feature aug. [26–31]	✓	(✓)	(✓)	✗
Graph rewiring [19, 32, 33]	✓	✓	✓	✗
Hierarchical MP [34–39]	✓	✓	(✓)	✗
EGP (ours)	✓	✓	✓	✓

78 implicitly, by storing a “global” vector [20]. It trivially reduces the graph’s diameter to 2, introduces
 79 $O(1)$ new nodes and $O(|V|)$ new edges, and requires no dedicated preprocessing, hence it satisfies
 80 (C1, C3, C4). However, these benefits come at the expense of introducing a bottleneck in the master
 81 node: it has a very challenging task (especially when graphs get larger) to continually incorporate
 82 information over a very large neighbourhood in a useful way. Hence it fails to satisfy (C2).

83 **Fully connected graphs.** The converse approach is to make *every* node a master node: in this case,
 84 we make all pairs of nodes connected by an edge—this was initially proposed as a powerful method
 85 to alleviate oversquashing by [18]. This strategy proved highly popular in the recent surge of Graph
 86 Transformers [22, 23, 25], and is common for GNNs used in physical simulation [21] or reasoning
 87 [24] tasks. The graph’s diameter is reduced to 1, no bottlenecks remain, and the approach does not
 88 require any dedicated preprocessing. Hence (C1, C2, C4) are trivially satisfied. The main downside
 89 of this approach is the introduction of $O(|V|^2)$ edges, which means (C3) can never be satisfied—and
 90 this approach will hence be prohibitive even for modestly-sized graphs.

91 **Feature augmentation.** An alternative approach is to provide additional features to the GNN which
 92 directly identify the structural role each node plays in the graph [26]. If done properly (i.e., if the
 93 computed features are relevant to the target), this can drastically improve expressive power. Hence, in
 94 theory, it is possible to satisfy (C1) while not violating (C2, C3). However, computing appropriate
 95 features requires either specific domain knowledge, or appropriate pre-training [27–31], in order to
 96 obtain such embeddings. Hence all of these gains come at the expense of failing to satisfy (C4).

97 **Graph rewiring.** Another promising line of research involves modifying the edges of the original
 98 graph to alleviate bottlenecks. Popular examples of this approach involve using diffusion [32]—which
 99 diffuse additional edges through the application of kernels such as the personalised PageRank, and
 100 stochastic discrete Ricci flows [19]—which surgically modify a small quantity of edges to alleviate
 101 the oversquashing effect on the nodes with negative Ricci curvature. Recent concurrent work [33]
 102 also uses constructions inspired by expander graphs to randomly locally rewire a given input graph.
 103 If realised carefully, such approaches will not deviate too far from the original graph, while provably
 104 alleviating oversquashing; hence it is possible to satisfy (C1, C2, C3). However, this comes at a cost
 105 of having to examine the input graph structure, with methods that do not necessarily scale easily with
 106 the number of nodes. As such, dedicated preprocessing is needed, failing to satisfy (C4).

107 **Hierarchical message passing.** Lastly, going beyond modifying the edges, it is also possible to
 108 introduce additional *nodes* in the graph—each of them responsible for a particular *substructure* in
 109 the graph¹. If done carefully, it has the potential to drastically reduce the graph’s diameter while not
 110 introducing bottlenecked nodes (hence, allowing us to satisfy (C1, C2)). However, in prior work,
 111 a cost has to be paid for this, usually in the need for dedicated preprocessing. Prior proposals for
 112 hierarchical GNNs that remain scalable require a dedicated pre-processing step [34–36], sometimes
 113 coupled with domain knowledge [36]—thus failing to satisfy (C4). In addition, such methods may
 114 require adding prohibitively large numbers of substructures [37, 38] or expensive pre-computation,
 115 e.g. computing the graph Laplacian eigenvectors [39]. This might make even (C3) hard to satisfy.

¹Master nodes are a special case: a single node is responsible for a “substructure” spanning the entire graph.

116 We remark that our work is not the first to study expander graph-related topics in the context of
 117 GNNs. Specifically, the ExpanderGNN [40] leverages expander graphs over neural network weights
 118 to sparsify the update step in GNNs. This is a direct application of Deep Expander Networks [41],
 119 which studied such constructs over CNNs. With respect to our contributions, neither of these cases
 120 discuss expanders in the context of the computational graph for a GNN, nor attempt to propagate
 121 messages over such a structure. Further, neither satisfy all four of our desired criteria (C1–C4).

122 3 Theoretical background

123 We now dedicate our attention to the key theoretical results over expander graphs, which will allow
 124 EGP to have favourable properties and be efficiently precomputable.

Definition 1. For a finite connected graph $G = (V(G), E(G))$, we consider functions $f: V(G) \rightarrow \mathbb{R}$.
 The Laplacian $Lf: V(G) \rightarrow \mathbb{R}$ of such a function is defined to be

$$Lf(v) = \deg(v)f(v) - \sum_{vw \in E(G)} f(w),$$

125 where $\deg(v)$ is the degree of the vertex v .

The mapping $L: \mathbb{R}^{V(G)} \rightarrow \mathbb{R}^{V(G)}$ sending a function f to its Laplacian Lf is a linear transformation.
 It is not hard to show [42] that L is symmetric with respect to the standard basis for $\mathbb{R}^{V(G)}$ and
 positive semi-definite and hence has non-negative real eigenvalues

$$0 = \lambda_0(G) < \lambda_1(G) \leq \lambda_2(G) \leq \dots$$

126 The smallest eigenvalue is 0 and its associated eigenspace consists of the constant functions (assuming
 127 G is connected). The smallest positive eigenvalue, $\lambda_1(G)$, is central to the definition of expander
 128 graphs, as the next definition shows.

129 **Definition 2.** An infinite collection $\{G_i\}$ of finite connected graphs is an *expander family* if there is
 130 a constant $c > 0$ such that for all G_i in the collection, $\lambda_1(G_i) \geq c$.

131 Expander families [43–45] have many remarkable and useful properties, particularly when there is a
 132 uniform upper bound on the degree of the vertices of G_i .

Definition 3. Let G be a finite graph. For $A \subset V(G)$, its *boundary* ∂A is the collection of edges
 with one endpoint in A and one endpoint not in A . The *Cheeger constant* $h(G)$ is defined to be

$$h(G) = \min \left\{ \frac{|\partial A|}{|A|} : A \subset V(G), 0 < |A| \leq |V(G)|/2 \right\}.$$

133 Thus, having a small Cheeger constant is equivalent to the graph having a ‘bottleneck’, in the sense
 134 that there is a collection of edges ∂A that, when removed, disconnects the vertices into two sets
 135 (A and its complement, $V(G) \setminus A$), with the property that the sizes of A and its complement are
 136 significantly larger than the size of ∂A .

137 Expander families can be reinterpreted using Cheeger constants, as follows (see, e.g., [46–49]):

138 **Theorem 4.** Let $\{G_i\}$ be an infinite collection of finite connected graphs with a uniform upper bound
 139 on their vertex degrees. Then the following are equivalent:

- 140 1. $\{G_i\}$ is an expander family;
- 141 2. there is a constant $\epsilon > 0$ such that for all graphs in the collection, $h(G_i) \geq \epsilon$.

142 Hence, expander graphs have higher Cheeger constants and will hence experience less severe problems
 143 arising due to bottleneck edges. The following result is one of the many useful properties of expander
 144 families, and it concerns their *diameter*. It was proved by Mohar [50, Theorem 2.3]. See also [47].

Theorem 5. The diameter $\text{diam}(G)$ of a graph G satisfies

$$\text{diam}(G) \leq 2 \left\lceil \frac{\Delta(G) + \lambda_1(G)}{4\lambda_1(G)} \log(|V(G)| - 1) \right\rceil,$$

where $\Delta(G)$ is the maximal degree of any vertex of G . Hence, if $\{G_i\}$ is an expander family of finite
 graphs with a uniform upper bound on their vertex degrees, then there is a constant $k > 0$ such that
 for all graphs in the family,

$$\text{diam}(G_i) \leq k \log V(G_i).$$

145 Therefore, if we want to globally propagate information over an expander graph which has $|V|$ nodes,
 146 we only need $O(\log |V|)$ propagation steps to do so—yielding subquadratic complexity.

147 We showed that expanders will experience less severe problems arising due to bottleneck edges, with
 148 favourable propagation qualities. What is missing is an efficient method of constructing an expander
 149 of (roughly) $|V|$ nodes. To demonstrate such a method, we leverage known results from group theory.

150 **Definition 6.** A group (Γ, \circ) is a set Γ equipped with a *composition* operation $\circ : \Gamma \times \Gamma \rightarrow \Gamma$ (written
 151 concisely by omitting \circ , i.e. $g \circ h = gh$, for $g, h \in \Gamma$), satisfying the following axioms:

- 152 • (*Associativity*) $(gh)l = g(hl)$, for $g, h, l \in \Gamma$.
- 153 • (*Identity*) There exists a unique $e \in \Gamma$ satisfying $eg = ge = g$ for all $g \in \Gamma$.
- 154 • (*Inverse*) For every $g \in \Gamma$ there exists a unique $g^{-1} \in \Gamma$ such that $gg^{-1} = g^{-1}g = e$.

155 A group is hence a natural construct for reasoning about transformations that leave an object invariant
 156 (unchanged). Further, we define a relevant notion of a group’s generating set:

157 **Definition 7.** Let Γ be a group. A subset $S \subseteq \Gamma$ is a *generating set* for Γ if it can be used to “generate”
 158 all of Γ via composition. Concretely, any element $g \in \Gamma$ can be expressed by composing elements in
 159 the generating set, or their inverses; that is, we can express $g = s_1^{\pm 1} s_2^{\pm 1} s_3^{\pm 1} \cdots s_{n-1}^{\pm 1} s_n^{\pm 1}$ for $s_i \in S$.

160 Now we are ready to define a Cayley graph of a group w.r.t. its generating set.

161 **Definition 8.** Let Γ be a group with a finite generating set S . Then the associated *Cayley graph*
 162 $\text{Cay}(\Gamma; S)$ has vertex set Γ and it has an edge $g \rightarrow gs$ for each $g \in \Gamma$ and each $s \in S$. We say that
 163 s is the *label* on this edge. This is a potentially non-simple graph, as it may have edges with both
 164 endpoints on the same vertex and it may have multiple edges between a pair of vertices. In particular,
 165 when s has order 2, then we view the edge $g \rightarrow gs$ and the edge $gs \rightarrow gs^2 = g$ as distinct edges.

166 Note that the degree of each vertex of a Cayley graph $\text{Cay}(\Gamma; S)$ is $2|S|$. This is because each vertex
 167 g is joined by edges to gs and gs^{-1} for each $s \in S$. Thus, we shall be particularly interested in the
 168 case where there is a uniform upper bound on $|S|$. The specific group we use for EGP is as follows.

For each positive integer n , the *special linear group* $\text{SL}(2, \mathbb{Z}_n)$ denotes the group of 2×2 matrices
 with entries that are integers modulo n and with determinant 1. One of its generating sets is:

$$S_n = \left\{ \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \right\}.$$

169 Central to our constructions is the following important result.

170 **Theorem 9.** *The family of Cayley graph $\text{Cay}(\text{SL}(2, \mathbb{Z}_n); S_n)$ forms an expander family.*

171 The proof uses a result of Selberg [51] who showed that the smallest positive eigenvalue of the
 172 Laplacian of certain hyperbolic surfaces is at least $3/16$. One can use this to produce a lower bound
 173 on the first eigenvalue of the Laplacian on $\text{Cay}(\text{SL}(2, \mathbb{Z}_n); S_n)$. Full proofs are given in [44, 45].

174 Lastly, it is useful to state a known result: the number of nodes of $\text{Cay}(\text{SL}(2, \mathbb{Z}_n); S_n)$ is:

$$|V(\text{Cay}(\text{SL}(2, \mathbb{Z}_n); S_n))| = n^3 \prod_{\text{prime } p|n} \left(1 - \frac{1}{p^2}\right), \quad (10)$$

175 hence, it is of the order of $O(n^3)$. We now study the local properties of Cayley graphs in detail.

176 4 Local structure of the Cayley graphs, and the utility of negative curvature

177 Recent work [19] has suggested that the local structure of the graph G underlying a GNN may play
 178 an important role in the way that information propagates around G . In particular, various notions of
 179 ‘Ricci curvature’ such as Forman curvature [52], Ollivier curvature [53, 54] and balanced Forman
 180 curvature [19] have been examined. These are all local quantities, in the sense that they depend on the
 181 structure of the graph within a small neighbourhood of each edge. In this section, we will therefore
 182 examine the local structure of the Cayley graphs $G_n = \text{Cay}(\text{SL}(2, \mathbb{Z}_n); S_n)$.

183 The various notions of curvature given above are defined for each e of the graph G . Since, as defined
 184 by [19], the balanced Forman curvature of an edge depends only on local structures (i.e. triangles

185 and squares) around that edge, they can be determined by only observing the immediate 2-hop
 186 surrounding of that edge. Formally, for an edge e of a graph G , let $N_2(e)$ be the induced subgraph
 187 with vertices that are at most two hops away from at least one endpoint of e . Then the curvature of e
 188 only depends on the isomorphism type of $N_2(e)$. More specifically, if e and e' are edges in possibly
 189 distinct graphs, and there is a graph isomorphism between $N_2(e)$ and $N_2(e')$ that sends e to e' , then
 190 this guarantees that the curvatures of e and e' are equal.

191 This situation arises prominently in the Cayley graphs that we are considering, as follows.

Proposition 11. *Let s be one of*

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \quad \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}.$$

192 *Let $n, n' > 18$ and let e and e' be s -labelled edges in G_n and $G_{n'}$. Then there is a graph isomorphism
 193 between $N_2(e)$ and $N_2(e')$ taking e to e' .*

194 We prove Proposition 11 in Appendix A. This immediately allows us to characterise the balanced
 195 Forman curvature and Ollivier curvature for all of the Cayley graphs we generate:

Proposition 12. *The balanced Forman curvatures $\text{Ric}(n)$, and the Ollivier curvatures $\kappa(n)$ of all
 edges of Cayley graphs G_n are given by:*

$$\text{Ric}(n) = \begin{cases} 0 & \text{if } n = 2 \\ -1/4 & \text{if } n = 3 \\ -1/2 & \text{if } n = 4 \\ -1 & \text{if } n \geq 5, \end{cases} \quad \kappa(n) = \begin{cases} 0 & \text{if } n = 2 \\ -1/8 & \text{if } n = 3 \\ -1/4 & \text{if } n = 4 \\ -3/8 & \text{if } n = 5 \\ -1/2 & \text{if } n \geq 6. \end{cases}$$

196 *Proof.* Proposition 11 implies that the balanced Forman and Ollivier curvatures are all equal for
 197 $n > 18$. Their values for $2 \leq n \leq 19$ can all be empirically computed, and are given as above. \square

198 Prior work [19] suggests it is preferable for GNNs to operate on graphs with positive Ricci curvature,
 199 whereas our graphs G_n ($n > 2$) all have negative Ricci curvature. However, we contend that negative
 200 Ricci curvature is not in itself an impediment to efficient propagation around a GNN. Indeed, it was
 201 shown in [19, Theorem 4] that poor propagation arises when the balanced Forman curvature is close
 202 to -2 , specifically if it is at most $-2 + \delta$ for some $\delta > 0$. Here, δ is required to satisfy certain
 203 inequalities. But, with certainty, $\delta = 1$ can *never* be satisfied in the hypotheses of [19, Theorem 4].

204 Furthermore, positive Ricci curvature may have *downsides* when used for GNNs. One significant
 205 downside can be derived using the main result of [55], which says that the three properties of
 206 expansion, sparsity and non-negative Ollivier curvature are incompatible, in the following sense.

207 **Theorem 13.** *For any $\delta > 0$ and $\Delta > 0$, there are only finitely many graphs with maximum vertex
 208 degree Δ , Cheeger constant at least δ and non-negative Ollivier curvature.*

209 We prove Theorem 13 in Appendix B. Furthermore, quoting directly from [55]:

210 *“The high-level message is that on large sparse graphs, non-negative curvature (in an even weak sense)
 211 induces extremely poor spectral expansion. This stands in stark contrast with the traditional idea
 212 – quantified by a broad variety of functional inequalities over the past decade – that non-negative
 213 curvature is associated with good mixing behavior.”*

214 In our view, it is highly desirable that the graphs used for GNNs have high Cheeger constants, in
 215 the sense of globally lacking bottlenecks. Having bounded vertex degree is certainly useful too,
 216 since it implies that the graphs will be sparse, and the nodes will not have to handle ever-increasing
 217 neighbourhoods for message passing as graphs grow larger in size.

218 However, by proving Theorem 13, we showed non-negative Ollivier curvature is *incompatible* with
 219 these properties for sufficiently large graphs. Specifically, given the *finite* supply of non-negatively
 220 curved sparse graphs, we can define N' as the largest number of nodes of such graphs. Then, for
 221 all graphs G where $|V(G)| > N'$, we will be *unable* to produce a computational graph for a GNN
 222 which is non-negatively curved everywhere. It remains an interesting challenge to provide an bound
 223 on N' (as a function of δ and Δ). It is possible that a careful analysis of [55] may provide this.

224 Further, while the expander graphs we generate are negatively-curved at -1 everywhere, and we
 225 will empirically show this helps alleviate oversquashing, we also believe that it is worthy of further

226 investigation to theoretically examine whether performance of GNNs decreases significantly when
 227 the curvature is less than -1 .

228 The negative curvature of each edge in G_n implies that they are locally ‘tree-like’. In Appendix C,
 229 we make this statement precise by showing that G_n is ‘tree-like’ up to scale $c \log(n)$ about each node,
 230 for $c \simeq (1/2)(\log((1 + \sqrt{5})/2))^{-1}$ (see Figure 1 (Right) for a schematic view).

231 This tree-like structure might seem, at first, to be counter-productive for good propagation across
 232 the graphs G_n . Indeed, GNNs based on trees have been shown to have provably poor performance
 233 [18]. The reason for this seems to be two-fold. On the one hand, trees have small Cheeger constant.
 234 Indeed, any tree G on n vertices has a Cheeger constant $1/\lfloor n/2 \rfloor$, since we may find an edge that,
 235 when removed, decomposes the graph into subgraphs with $\lfloor n/2 \rfloor$ and $\lceil n/2 \rceil$ vertices. As discussed
 236 in Section 3 and in [19], when a graph has small Cheeger constant, its performance when used as
 237 a template for a GNN is likely to become poor. Secondly, GNNs based on trees are susceptible
 238 to oversquashing. For a k -regular infinite tree, there are $k(k-1)^{r-1}$ vertices at distance r from a
 239 given vertex. Hence, if information is to be propagated at least distance r from a given vertex, then
 240 seemingly an exponential amount of information is required to be stored.

241 However, neither of these issues are problematic for a GNN based on the Cayley graph G_n . By
 242 Theorem 9, their Cheeger constants are bounded away from 0. Secondly, although they are tree-like
 243 locally, this is only true up to scale $O(\log n)$. In fact, the r -neighbourhood of any vertex is the whole
 244 graph G_n as soon as $r > C \log n$, for some constant C , by Theorem 5. Being tree-like up to distance
 245 $O(\log n)$ does not lead to a requirement to store too much information as the message propagates.
 246 This is because $k(k-1)^{r-1}$ is polynomial in n when $r \leq O(\log n)$. Beyond this scale, there exist
 247 many additional connections, which lead to many possible paths joining any pair of vertices. The
 248 perspective of information transfer also gives rise to another perspective in which expanders fare very
 249 favourably: the *mixing time* of their corresponding Markov chain (see Appendix D for details).

250 5 Expander graph propagation

251 Let an input to a graph neural network be a node feature matrix $\mathbf{X} \in \mathbb{R}^{|V| \times d}$, and an adjacency matrix
 252 $\mathbf{A} \in \mathbb{R}^{|V| \times |V|}$. This setup is such that the feature vector of node u , $\mathbf{x}_u \in \mathbb{R}^d$, can be recovered by
 253 taking an appropriate row from \mathbf{X} . Note that the adjacency information can also be fed in an edge-list
 254 manner, which is desirable from a scalability perspective. Further, each edge in the graph may be
 255 endowed with additional features rather than a single real scalar. None of the above modifications
 256 would change the essence of our findings; we use a matrix formalism here purely for simplicity.

257 There exist many ways in which the computed Cayley graph $\text{Cay}(\text{SL}(2, \mathbb{Z}_n); S_n)$ can be leveraged
 258 for message propagation, and exploring these variations could be very useful for future work. Here,
 259 we opt for a simple construction: interleave running a standard GNN over the given input structure,
 260 followed by running another GNN layer over the relevant Cayley graph. If we let $\mathbf{A}^{\text{Cay}(n)}$ be an
 261 adjacency matrix derived from $\text{Cay}(\text{SL}(2, \mathbb{Z}_n); S_n)$, this implies:

$$\mathbf{H} = \text{GNN}(\text{GNN}(\mathbf{X}, \mathbf{A}), \mathbf{A}^{\text{Cay}(n)}) \quad (14)$$

262 Here, GNN refers to any preferred GNN layer, such as the graph isomorphism network [56, GIN]:

$$\mathbf{h}_u = \phi \left((1 + \epsilon) \mathbf{x}_u + \sum_{v \in \mathcal{N}_u} \mathbf{x}_v \right) \quad (15)$$

263 where \mathcal{N}_u is the neighbourhood of node u , i.e. in our setup, the set of all nodes v such that $a_{vu} \neq 0$.
 264 $\epsilon \in \mathbb{R}$ is a learnable scalar, and $\phi: \mathbb{R}^d \rightarrow \mathbb{R}^d$ is a two-layer MLP.

265 This procedure is iterated for a certain number of steps, after which the computed node embeddings
 266 in \mathbf{H} can be used for any downstream task of interest—such as node classification, link prediction
 267 or graph classification. Note that, unlike [18], who apply their custom layer only at the *tail* of the
 268 architecture, we apply the expander graph immediately after each layer over the input graph. We find
 269 that if the input graph given by \mathbf{A} contains bottlenecks, applying the GNN over $\mathbf{A}^{\text{Cay}(n)}$ only at the
 270 end may result in oversquashing occurring before any expander graph propagation can take place.

271 The setup so far assumed the number of nodes in our input graph to line up with the Cayley graph,
 272 that is, $\mathbf{A}^{\text{Cay}(n)} \in \mathbb{R}^{|V| \times |V|}$. However, there is no guarantee that we can find an appropriate n such

273 that $\text{Cay}(\text{SL}(2, \mathbb{Z}_n); S_n)$ would have $|V|$ nodes. What we can do in practice, as an approximation, is
 274 choose the smallest n such that the number of nodes of $\text{Cay}(\text{SL}(2, \mathbb{Z}_n); S_n)$ is $\geq |V|$, then consider
 275 $\mathbf{A}_{1:|V|, 1:|V|}^{\text{Cay}(n)}$ —i.e. only the subgraph containing the first $|V|$ nodes in the Cayley graph.

276 There is a slight misalignment to our theory in this slicing choice—if the $|V|$ vertices in this subgraph
 277 are chosen completely arbitrarily, we risk disconnecting the graph. However, in all our experiments
 278 we construct the Cayley graph in a breadth-first manner, starting from the identity element as “node
 279 zero”. Hence, the node at index i is always guaranteed to be reachable from the nodes at lower indices
 280 ($j < i$), and the graph cannot be disconnected under this construction. More interesting strategies for
 281 this step can also be considered in the future. Note that, much like the fully connected graph used by
 282 [18], we interpret the Cayley graph mainly as a *template* for global information propagation, in order
 283 to relieve bottlenecks in a scalable way. Our interpretation, hence, assumes that the efficient diffusion
 284 of information over the whole graph is of benefit to the learning task we perform. When this is not
 285 the case, it might be worthwhile to construct expanders that somehow align with the input graph, but
 286 no such expander constructions are currently known, to the best of our knowledge. There is also a
 287 possible effect of *stochasticity* due to arbitrarily having to align the Cayley graph’s nodes to the input
 288 graph—which would not appear when using master nodes or fully-connected graphs—though our
 289 preliminary experiments did not observe any such negative effects.

290 Algorithm 1 summarises the steps of our proposed EGP model. As direct corollaries of results we
 291 proved or demonstrated, we note that EGP satisfies all four of our desirable criteria: **(C1)** by Theorem
 292 5 (so long as logarithmically many layers are applied), **(C2)** by Theorem 4 (high Cheeger constant
 293 implies no bottlenecks), **(C3)** by the fact our Cayley graphs are 4-regular and hence sparse, and
 294 **(C4)** by the fact we can generate a Cayley graph of appropriate size without detailed analysis of the
 295 input—we may precompute a “bank” of Cayley graphs of various sizes to use in an ad-hoc manner.

Algorithm 1: Expander graph propagation (EGP) forward pass

Inputs : Node features $\mathbf{X} \in \mathbb{R}^{|V| \times d}$, Adjacency matrix $\mathbf{A} \in \mathbb{R}^{|V| \times |V|}$

Output : Node embeddings \mathbf{H}

// Choose the smallest Cayley graph from our family that has number of nodes equal to, or greater than, $|V|$
 $n \leftarrow \operatorname{argmin}_{m \in \mathbb{N}} |V(\text{Cay}(\text{SL}(2, \mathbb{Z}_m); S_m))| \geq |V|$; // We can use Equation 10 to determine n

$G^{\text{Cay}(n)} \leftarrow \text{Cay}(\text{SL}(2, \mathbb{Z}_n); S_n)$

$\mathbf{A}_{uv}^{\text{Cay}(n)} \leftarrow \begin{cases} 1 & (u, v) \in E(G^{\text{Cay}(n)}) \\ 0 & \text{otherwise} \end{cases}$; // Populate adjacency matrix of the Cayley graph

$\mathbf{H}^{(0)} \leftarrow \mathbf{X}$; // Initialise GNN inputs

for $t \in \{1, \dots, T\}$ **do**

if $t \bmod 2 = 0$ **then**

$\mathbf{H}^{(t)} \leftarrow \text{GNN}^{(t)}(\mathbf{H}^{(t-1)}, \mathbf{A})$; // GNN layer over input graph; e.g. Equation 15

end

else

$\mathbf{H}^{(t)} \leftarrow \text{GNN}^{\text{Cay}(t)}(\mathbf{H}^{(t-1)}, \mathbf{A}_{1:|V|, 1:|V|}^{\text{Cay}(n)})$; // GNN layer over Cayley graph; e.g. Eq. 15

end

end

return $\mathbf{H}^{(T)}$; // Return final embeddings for downstream use

296 **6 Empirical evaluation**

297 Our work provides mainly a theoretical contribution: demonstrating a simple, theoretically-grounded
 298 approach to relieving bottlenecks and oversquashing in GNNs without requiring quadratic complexity
 299 or dedicated preprocessing. Further, we prove several additional results which deepen our understand-
 300 ing of curvature-based analysis of GNNs, showing how our expanders can be favourable in spite of
 301 their negatively-curved edges. We now provide results that empirically supplement our claim.

Table 2: Comparative evaluation performance on the four datasets studied. Our baseline model is a GIN [56], using exactly the same implementation as in [57]. See Appendix E for ablations.

Model	ogbg-molhiv	ogbg-molpcba	ogbg-ppa	ogbg-code2
GIN	0.7558 \pm 0.0140	0.2266 \pm 0.0028	0.6892 \pm 0.0100	0.1495 \pm 0.0023
GIN + EGP	0.7934 \pm 0.0035	0.2329 \pm 0.0019	0.7027 \pm 0.0159	0.1497 \pm 0.0015

302 **Tree-NeighborsMatch** We start by comparing our models on the Tree-NeighborsMatch task
 303 (for more details, see Alon and Yahav [18, Section 4.1]). Tree-NeighborsMatch is a synthetic
 304 benchmark explicitly designed to test a GNN’s ability to counter oversquashing, and therefore it
 305 allows us to empirically verify that EGP is capable of alleviating oversquashing. We augment the
 306 original GIN implementation from the authors [18] with EGP layers, and find that it is capable
 307 of solving the task at depth=5 **at 100% accuracy**, demonstrating alleviated oversquashing. In
 308 comparison, baseline GIN can only achieve 29% on this same task, and the best-performing GNN
 309 without EGP—i.e. propagating over the input tree only—cannot exceed 60% accuracy.

310 **OGB Datasets** For real-world evaluation, we leverage the established Open Graph Benchmark
 311 collection of tasks [57, OGB]. Specifically, we provide results on all of its graph classification
 312 datasets: ogbg-molhiv, ogbg-molpcba, ogbg-ppa and ogbg-code2. The first two are among the
 313 largest molecule property prediction datasets in the MoleculeNet benchmark [58]. The third dataset
 314 is concerned with classifying species into their taxa, from their protein-protein association networks
 315 [59, 60] given as input. The fourth dataset is a *code summarisation* task: it requires predicting the
 316 tokens in the name of a Python method, given the abstract syntax tree (AST) of its implementation.

317 We provide a summary of important dataset statistics in Appendix E; please see [57] for detailed
 318 information. These datasets are designed to span a wide variety of domains (virtual drug screening,
 319 molecular activity prediction, protein-protein interactions, code summarisation) and sizes (from small
 320 molecules to very large syntax trees—the largest graph in ogbg-code2 has 36, 123 nodes).

321 **Models** In all four datasets, we want to *directly* evaluate the empirical gain of introducing an EGP
 322 layer and completely rule out any effects from parameter count, or similar architectural decisions.

323 To enable this, we take inspiration from the experimental setup of [18]. Our baseline model is the
 324 GIN [56], with hyperparameters as given by [57]. We use the *official* publicly available model
 325 implementation from the OGB authors [57], and modify all *even* layers of the architecture to operate
 326 over the appropriately-sampled Cayley graph.

327 Note that our construction leaves both the parameter count and latent dimension of the model
 328 *unchanged*, hence any benefits coming from optimising those have been diminished.

329 **Results** The results of our evaluation are presented in Table 2. It can be observed that, in all four
 330 cases, propagating information over the Cayley graph yields improvements in mean performance—
 331 these improvements are most apparent on ogbg-molhiv, but also present in ogbg-molpcba and
 332 ogbg-ppa. We believe that these results provide encouraging empirical evidence that propagating
 333 information over Cayley graphs is an elegant idea for alleviating bottlenecks. We provide additional
 334 results on OGB, comparing EGP to various other oversquashing-countering methods, in Appendix E.

335 7 Conclusion

336 In this paper, we have presented expander graph propagation (EGP), a novel and elegant approach to
 337 alleviating bottlenecks in graph representation learning, which provably supports global communica-
 338 tion while not requiring quadratic complexity or dedicated preprocessing of the input.

339 To this end, we offered a detailed theoretical overview of Cayley graphs of special linear groups,
 340 $\text{Cay}(\text{SL}(2, \mathbb{Z}_n); S_n)$. We cite proofs that these graphs have highly favourable properties for infor-
 341 mation propagation in graph neural networks: they are sparse and 4-regular, they have logarithmic
 342 diameter, and they can be efficiently precomputed by a simple procedure that does not rely on the
 343 input structure. We show that, in spite of having negatively curved edges, our findings do not violate
 344 any prior results on understanding oversquashing via curvature. Even under a simple intervention—
 345 interleaving EGP layers inbetween standard GNN layers—we have been able to recover significant
 346 performance returns without changing the parameter count or latent space dimensionality.

References

- 347
- 348 [1] William L Hamilton. Graph representation learning. *Synthesis Lectures on Artificial Intelligence*
349 *and Machine Learning*, 14(3):1–159, 2020. 1
- 350 [2] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional
351 networks. *arXiv preprint arXiv:1609.02907*, 2016. 1
- 352 [3] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua
353 Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.
- 354 [4] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural
355 message passing for quantum chemistry. In *International conference on machine learning*,
356 pages 1263–1272. PMLR, 2017. 1, 2, 3
- 357 [5] Michael M Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning:
358 Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021. 1
- 359 [6] Petar Veličković. Message passing all the way up. *arXiv preprint arXiv:2202.11097*, 2022. 1
- 360 [7] Jonathan M Stokes, Kevin Yang, Kyle Swanson, Wengong Jin, Andres Cubillos-Ruiz, Nina M
361 Donghia, Craig R MacNair, Shawn French, Lindsey A Carfrae, Zohar Bloom-Ackermann, et al.
362 A deep learning approach to antibiotic discovery. *Cell*, 180(4):688–702, 2020. 1
- 363 [8] Austin Derrow-Pinion, Jennifer She, David Wong, Oliver Lange, Todd Hester, Luis Perez,
364 Marc Nunkesser, Seongjae Lee, Xueying Guo, Brett Wiltshire, et al. Eta prediction with graph
365 neural networks in google maps. In *Proceedings of the 30th ACM International Conference on*
366 *Information & Knowledge Management*, pages 3767–3776, 2021. 1
- 367 [9] Azalia Mirhoseini, Anna Goldie, Mustafa Yazgan, Joe Wenjie Jiang, Ebrahim Songhori, Shen
368 Wang, Young-Joon Lee, Eric Johnson, Omkar Pathak, Azade Nazi, et al. A graph placement
369 methodology for fast chip design. *Nature*, 594(7862):207–212, 2021. 1
- 370 [10] Charles Blundell, Lars Buesing, Alex Davies, Petar Veličković, and Geordie Williamson.
371 Towards combinatorial invariance for kazhdan-lusztig polynomials. *arXiv preprint*
372 *arXiv:2111.15161*, 2021. 1
- 373 [11] Alex Davies, Petar Veličković, Lars Buesing, Sam Blackwell, Daniel Zheng, Nenad Tomašev,
374 Richard Tanburn, Peter Battaglia, Charles Blundell, András Juhász, et al. Advancing mathemat-
375 ics by guiding human intuition with ai. *Nature*, 600(7887):70–74, 2021. 1
- 376 [12] Qian Huang, Horace He, Abhay Singh, Ser-Nam Lim, and Austin R Benson. Combining
377 label propagation and simple models out-performs graph neural networks. *arXiv preprint*
378 *arXiv:2010.13993*, 2020. 1
- 379 [13] Shyam A Tailor, Felix Opolka, Pietro Lio, and Nicholas Donald Lane. Do we need anisotropic
380 graph neural networks? In *International Conference on Learning Representations*, 2021.
- 381 [14] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger.
382 Simplifying graph convolutional networks. In *International conference on machine learning*,
383 pages 6861–6871. PMLR, 2019. 1
- 384 [15] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel,
385 Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning
386 molecular fingerprints. *Advances in neural information processing systems*, 28, 2015. 1
- 387 [16] Federico Errica, Marco Podda, Davide Bacciu, and Alessio Micheli. A fair comparison of graph
388 neural networks for graph classification. *arXiv preprint arXiv:1912.09893*, 2019. 1
- 389 [17] Enxhell Luzhnica, Ben Day, and Pietro Liò. On graph classification networks, datasets and
390 baselines. *arXiv preprint arXiv:1905.04682*, 2019. 1
- 391 [18] Uri Alon and Eran Yahav. On the bottleneck of graph neural networks and its practical
392 implications. *arXiv preprint arXiv:2006.05205*, 2020. 1, 2, 3, 7, 8, 9, 17
- 393 [19] Jake Topping, Francesco Di Giovanni, Benjamin Paul Chamberlain, Xiaowen Dong, and
394 Michael M Bronstein. Understanding over-squashing and bottlenecks on graphs via curvature.
395 *arXiv preprint arXiv:2111.14522*, 2021. 2, 3, 5, 6, 7
- 396 [20] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius
397 Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan
398 Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint*
399 *arXiv:1806.01261*, 2018. 3

- 400 [21] Peter Battaglia, Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende, et al. Interaction
401 networks for learning about objects, relations and physics. *Advances in neural information*
402 *processing systems*, 29, 2016. 3
- 403 [22] Devin Kreuzer, Dominique Beaini, Will Hamilton, Vincent Létourneau, and Prudencio Tossou.
404 Rethinking graph transformers with spectral attention. *Advances in Neural Information Pro-*
405 *cessing Systems*, 34, 2021. 3
- 406 [23] Grégoire Mialon, Dexiong Chen, Margot Selosse, and Julien Mairal. Graphit: Encoding graph
407 structure in transformers. *arXiv preprint arXiv:2106.05667*, 2021. 3
- 408 [24] Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter
409 Battaglia, and Timothy Lillicrap. A simple neural network module for relational reasoning.
410 *Advances in neural information processing systems*, 30, 2017. 3
- 411 [25] Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen,
412 and Tie-Yan Liu. Do transformers really perform badly for graph representation? *Advances in*
413 *Neural Information Processing Systems*, 34, 2021. 3
- 414 [26] Giorgos Bouritsas, Fabrizio Frasca, Stefanos P Zafeiriou, and Michael Bronstein. Improving
415 graph neural network expressivity via subgraph isomorphism counting. *IEEE Transactions on*
416 *Pattern Analysis and Machine Intelligence*, 2022. 3
- 417 [27] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In
418 *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and*
419 *data mining*, pages 855–864, 2016. 3
- 420 [28] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social repre-
421 sentations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge*
422 *discovery and data mining*, pages 701–710, 2014.
- 423 [29] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-
424 scale information network embedding. In *Proceedings of the 24th international conference on*
425 *world wide web*, pages 1067–1077, 2015.
- 426 [30] Shantanu Thakoor, Corentin Tallec, Mohammad Gheshlaghi Azar, Mehdi Azabou, Eva L Dyer,
427 Remi Munos, Petar Veličković, and Michal Valko. Large-scale representation learning on graphs
428 via bootstrapping. In *International Conference on Learning Representations*, 2021.
- 429 [31] Petar Veličković, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon
430 Hjelm. Deep graph infomax. *arXiv preprint arXiv:1809.10341*, 2018. 3
- 431 [32] Johannes Gasteiger, Stefan Weissenberger, and Stephan Günnemann. Diffusion improves graph
432 learning. *arXiv preprint arXiv:1911.05485*, 2019. 3
- 433 [33] Pradeep Kr Banerjee, Kedar Karhadkar, Yu Guang Wang, Uri Alon, and Guido Montúfar.
434 Oversquashing in gnns through the lens of information contraction and graph expansion. *arXiv*
435 *preprint arXiv:2208.03471*, 2022. 3, 17
- 436 [34] Cristian Bodnar, Fabrizio Frasca, Nina Otter, Yuguang Wang, Pietro Lio, Guido F Montufar,
437 and Michael Bronstein. Weisfeiler and lehman go cellular: Cw networks. *Advances in Neural*
438 *Information Processing Systems*, 34:2625–2640, 2021. 3
- 439 [35] Cristian Bodnar, Fabrizio Frasca, Yuguang Wang, Nina Otter, Guido F Montufar, Pietro Lio,
440 and Michael Bronstein. Weisfeiler and lehman go topological: Message passing simplicial
441 networks. In *International Conference on Machine Learning*, pages 1026–1037. PMLR, 2021.
- 442 [36] Matthias Fey, Jan-Gin Yuen, and Frank Weichert. Hierarchical inter-message passing for
443 learning on molecular graphs. *arXiv preprint arXiv:2006.12179*, 2020. 3
- 444 [37] Christopher Morris, Gaurav Rattan, and Petra Mutzel. Weisfeiler and leman go sparse: Towards
445 scalable higher-order graph embeddings. *Advances in Neural Information Processing Systems*,
446 33:21824–21840, 2020. 3
- 447 [38] Christopher Morris, Martin Ritzert, Matthias Fey, William L Hamilton, Jan Eric Lenssen,
448 Gaurav Rattan, and Martin Grohe. Weisfeiler and leman go neural: Higher-order graph neural
449 networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages
450 4602–4609, 2019. 3
- 451 [39] Kimberly Stachenfeld, Jonathan Godwin, and Peter Battaglia. Graph networks with spectral
452 message passing. *arXiv preprint arXiv:2101.00079*, 2020. 3

- 453 [40] Johannes F Lutzeyer, Changmin Wu, and Michalis Vazirgiannis. Sparsifying the update step in
454 graph neural networks. *arXiv preprint arXiv:2109.00909*, 2021. 4
- 455 [41] Ameya Prabhu, Girish Varma, and Anoop Namboodiri. Deep expander networks: Efficient deep
456 networks from graph theory. In *Proceedings of the European Conference on Computer Vision*
457 (*ECCV*), pages 20–35, 2018. 4
- 458 [42] Fan R. K. Chung. *Spectral graph theory*, volume 92 of *CBMS Regional Conference Series in*
459 *Mathematics*. Published for the Conference Board of the Mathematical Sciences, Washington,
460 DC; by the American Mathematical Society, Providence, RI, 1997. ISBN 0-8218-0315-8. 4, 15
- 461 [43] Alexander Lubotzky. *Discrete groups, expanding graphs and invariant measures*, volume
462 125 of *Progress in Mathematics*. Birkhäuser Verlag, Basel, 1994. ISBN 3-7643-5075-X.
463 doi: 10.1007/978-3-0346-0332-4. URL <https://doi.org/10.1007/978-3-0346-0332-4>.
464 With an appendix by Jonathan D. Rogawski. 4
- 465 [44] Giuliana Davidoff, Peter Sarnak, and Alain Valette. *Elementary number theory, group theory,*
466 *and Ramanujan graphs*, volume 55 of *London Mathematical Society Student Texts*. Cambridge
467 University Press, Cambridge, 2003. ISBN 0-521-82426-5; 0-521-53143-8. doi: 10.1017/
468 CBO9780511615825. URL <https://doi.org/10.1017/CBO9780511615825>. 5
- 469 [45] Emmanuel Kowalski. *An introduction to expander graphs*, volume 26 of *Cours Spécialisés*
470 *[Specialized Courses]*. Société Mathématique de France, Paris, 2019. ISBN 978-2-85629-898-5.
471 4, 5
- 472 [46] N. Alon. Eigenvalues and expanders. volume 6, pages 83–96. 1986. doi: 10.1007/BF02579166.
473 URL <https://doi.org/10.1007/BF02579166>. Theory of computing (Singer Island, Fla.,
474 1984). 4
- 475 [47] N. Alon and V. D. Milman. λ_1 , isoperimetric inequalities for graphs, and superconcentrators.
476 *J. Combin. Theory Ser. B*, 38(1):73–88, 1985. ISSN 0095-8956. doi: 10.1016/0095-8956(85)
477 90092-9. URL [https://doi.org/10.1016/0095-8956\(85\)90092-9](https://doi.org/10.1016/0095-8956(85)90092-9). 4
- 478 [48] Jozef Dodziuk. Difference equations, isoperimetric inequality and transience of certain random
479 walks. *Trans. Amer. Math. Soc.*, 284(2):787–794, 1984. ISSN 0002-9947. doi: 10.2307/1999107.
480 URL <https://doi.org/10.2307/1999107>.
- 481 [49] R. Michael Tanner. Explicit concentrators from generalized N -gons. *SIAM J. Algebraic*
482 *Discrete Methods*, 5(3):287–293, 1984. ISSN 0196-5212. doi: 10.1137/0605030. URL
483 <https://doi.org/10.1137/0605030>. 4
- 484 [50] Bojan Mohar. Eigenvalues, diameter, and mean distance in graphs. *Graphs Combin.*, 7(1):
485 53–64, 1991. ISSN 0911-0119. doi: 10.1007/BF01789463. URL [https://doi.org/10.](https://doi.org/10.1007/BF01789463)
486 [1007/BF01789463](https://doi.org/10.1007/BF01789463). 4
- 487 [51] Atle Selberg. On the estimation of Fourier coefficients of modular forms. In *Proc. Sympos.*
488 *Pure Math., Vol. VIII*, pages 1–15. Amer. Math. Soc., Providence, R.I., 1965. 5
- 489 [52] R Forman. Discrete and computational geometry. 2003. 5
- 490 [53] Yann Ollivier. Ricci curvature of metric spaces. *Comptes Rendus Mathématique*, 345(11):
491 643–646, 2007. 5
- 492 [54] Yann Ollivier. Ricci curvature of markov chains on metric spaces. *Journal of Functional*
493 *Analysis*, 256(3):810–864, 2009. 5
- 494 [55] Justin Salez. Sparse expanders have negative curvature. *arXiv preprint arXiv:2101.08242*, 2021.
495 6, 14
- 496 [56] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural
497 networks? *arXiv preprint arXiv:1810.00826*, 2018. 7, 9, 17
- 498 [57] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele
499 Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs.
500 *Advances in neural information processing systems*, 33:22118–22133, 2020. 9, 16, 17
- 501 [58] Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S
502 Pappu, Karl Leswing, and Vijay Pande. Moleculenet: a benchmark for molecular machine
503 learning. *Chemical science*, 9(2):513–530, 2018. 9

- 504 [59] Damian Szklarczyk, Annika L Gable, David Lyon, Alexander Junge, Stefan Wyder, Jaime
 505 Huerta-Cepas, Milan Simonovic, Nadezhda T Doncheva, John H Morris, Peer Bork, et al.
 506 String v11: protein–protein association networks with increased coverage, supporting functional
 507 discovery in genome-wide experimental datasets. *Nucleic acids research*, 47(D1):D607–D613,
 508 2019. 9
- 509 [60] Marinka Zitnik, Rok Sosič, Marcus W Feldman, and Jure Leskovec. Evolution of resilience in
 510 protein interactomes across the tree of life. *Proceedings of the National Academy of Sciences*,
 511 116(10):4426–4433, 2019. 9
- 512 [61] Grigori A Margulis. Explicit constructions of graphs without short cycles and low density
 513 codes. *Combinatorica*, 2(1):71–78, 1982. 13
- 514 [62] Jean-Pierre Serre. *Trees*. Springer Science & Business Media, 2002. 16

515 A Proof of Proposition 11

Let s be one of

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \quad \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}.$$

516 Let $n, n' > 18$ and let e and e' be s -labelled edges in G_n and $G_{n'}$. Then there is a graph isomorphism
 517 between $N_2(e)$ and $N_2(e')$ taking e to e' .

518 *Proof.* Note first that, by the homogeneity of the Cayley graphs G_n and $G_{n'}$, we may assume that e
 519 and e' emanate from the identity vertex of each graph.

Let G_∞ be the Cayley graph of $\text{SL}(2, \mathbb{Z})$ with respect to the generators

$$S_\infty = \left\{ \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \right\}.$$

Let e_∞ be the s -labelled edge emanating from the identity vertex of G_∞ . The quotient homomorphism

$$\text{SL}(2, \mathbb{Z}) \rightarrow \text{SL}(2, \mathbb{Z}_n)$$

induces a graph homomorphism $G_\infty \rightarrow G_n$ sending e_∞ to e . We will show that it restricts to a graph
 isomorphism

$$N_2(e_\infty) \rightarrow N_2(e).$$

520 As there is a similar graph isomorphism $N_2(e_\infty) \rightarrow N_2(e')$, the proposition will follow.

Note that two elements of $\text{SL}(2, \mathbb{Z})$ map to the same element of $\text{SL}(2, \mathbb{Z}_n)$ if and only if they differ
 by multiplication by an element of the kernel K_n . This is

$$K_n = \left\{ \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \text{SL}(2, \mathbb{Z}) : a \equiv d \equiv 1 \pmod{n} \text{ and } b \equiv c \equiv 0 \pmod{n} \right\}.$$

521 The graph homomorphism sends edges to edges, and so it is distance non-increasing. Hence it
 522 certainly sends $N_2(e_\infty)$ to $N_2(e)$. It is also clearly surjective, because any element of $N_2(e)$ is
 523 reached from an endpoint of e by a path of length at most 2, and there is a corresponding path in
 524 $N_2(e_\infty)$.

525 We just need to show that this is an injection. If not, then two distinct vertices g_1 and g_2 in $N_2(e_\infty)$
 526 map to the same vertex in $N_2(e)$. Note then that as elements of $\text{SL}(2, \mathbb{Z})$, $g_2 = g_1 k$ for some $k \in K_n$.
 527 There are paths with length at most 3 joining the identity 1 to g_1 and g_2 respectively. Hence, the
 528 distance in G_∞ between g_1 and g_2 is at most 6. Therefore, the distance between 1 and $g_1^{-1}g_2$ is at
 529 most 6. This element $g_1^{-1}g_2$ lies in K_n . We will show that when $n > 18$, the only element of K_n
 530 that has distance at most 6 from the identity is the identity itself. This will imply that $g_1^{-1}g_2 = 1$ and
 531 hence $g_1 = g_2$. But this contradicts the assumption that g_1 and g_2 are distinct vertices. Our argument
 532 follows that of [61].

The operator norm $\|A\|$ of a matrix $A \in \text{SL}(2, \mathbb{Z})$ is

$$\|A\| = \sup\{|A(v)| : v \in \mathbb{R}^2, |v| = 1\}.$$

This is submultiplicative: $\|AB\| \leq \|A\| \|B\|$ for matrices A and B . It can be calculated as the square root of the largest eigenvalue of $A^t A$. In our case, the operator norms satisfy

$$\left\| \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \right\| = \left\| \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \right\| = \frac{1 + \sqrt{5}}{2}.$$

Consider an element

$$K = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

of K_n that is not the identity. Since $a \equiv d \equiv 1$ modulo n and $b \equiv c \equiv 0$ modulo n , we deduce that at least one $|a|$, $|b|$, $|c|$ and $|d|$ is at least $n - 1$. Therefore, this matrix acts on one of the vectors $(1, 0)^t$ or $(0, 1)^t$ by scaling its length by at least $n - 1$. Therefore, $\|K\| \geq n - 1$. Suppose now that K has distance at most 6 from the identity. Then K can be written as a word in the generators of $\text{SL}(2, \mathbb{Z})$ with length at most 6. Therefore, we obtain the inequality

$$\|K\| \leq \left(\frac{1 + \sqrt{5}}{2} \right)^6 < 17.95.$$

533 Hence, $n < 18.95$ and therefore, as n is integral, $n \leq 18$. □

534 B Proof of Theorem 13

535 For any $\delta > 0$ and $\Delta > 0$, there are only finitely many graphs with maximum vertex degree Δ ,
536 Cheeger constant at least δ and non-negative Ollivier curvature.

537 *Proof.* This is a consequence of the main result of Salez [55, Theorem 3]. This states if $G_n =$
538 (V_n, E_n) is a sequence of graphs with the following properties:

$$\sup_{n \geq 1} \left\{ \frac{1}{|V_n|} \sum_{v \in V_n} \deg(v) \log \deg(v) \right\} < \infty \quad (16)$$

539

$$\forall \epsilon > 0, \quad \frac{1}{|E_n|} |\{e \in E_n : \kappa(e) < -\epsilon\}| \rightarrow 0 \text{ as } n \rightarrow \infty, \quad (17)$$

then

$$\forall \rho < 1, \quad \liminf_{n \rightarrow \infty} \left\{ \frac{1}{|V_n|} |\{i : \mu_i(G_n) \geq \rho\}| \right\} > 0.$$

Here, $\kappa(e)$ is the Ollivier curvature of an edge e and

$$1 = \mu_0(G) \geq \mu_1(G) \geq \dots \geq 0$$

are the eigenvalues of the lazy random walk operator. To prove the theorem, we suppose that on the contrary, there are infinitely many distinct graphs $G_n = (V_n, E_n)$ with maximum vertex degree Δ , Cheeger constant at least δ and non-negative Ollivier curvature. Then

$$\sum_{v \in V_n} \deg(v) \log \deg(v) \leq |V_n| \Delta \log \Delta$$

and so condition 16 is satisfied. Condition 17 is trivially satisfied because the Ollivier curvature of each graph is non-negative. Thus, we deduce that the conclusion of Salez' theorem holds. Setting $\rho = 1 - (\delta^2/4\Delta^2)$, we deduce that a definite proportion of the eigenvalues of the lazy random walk operator are at least $1 - (\delta^2/4\Delta^2)$. In particular, $\mu_1(G_n) \geq 1 - (\delta^2/4\Delta^2)$. Denote the eigenvalues of the normalised Laplacian by

$$0 = \lambda'_0(G_n) \leq \lambda'_1(G_n) \leq \dots$$

These are related to the eigenvalues of the lazy random walk operator by $\lambda'_i(G_n) = 2 - 2\mu_i(G_n)$. Hence, $\lambda'_1(G_n) \leq \delta^2/(2\Delta^2)$. There is a variation of Cheeger's inequality that relates λ'_1 to the *conductance* of the graph. To define this, one considers subsets A of the vertex set, and defines their *volume* to be $\text{vol}(A) = \sum_{v \in A} \deg(v)$. The conductance $\phi(G)$ of a graph G is

$$\phi(G) = \min \left\{ \frac{|\partial A|}{\text{vol}(A)} : A \subset V(G), 0 < \text{vol}(A) \leq \text{vol}(V(G))/2 \right\}.$$

Then, by Chung [42, Theorem 2.2],

$$\phi(G) \leq \sqrt{2\lambda_1(G)}$$

Hence, in our case,

$$\phi(G_n) \leq \delta/\Delta.$$

Consider any subset A_n of the vertex set that realises $\phi(G_n)$. Thus $0 < \text{vol}(A_n) \leq \text{vol}(V_n)/2$ and $|\partial A_n|/\text{vol}(A_n) = \phi(G_n) \leq \delta/\Delta$. If A_n is at most half the vertices of G_n , then this implies that the Cheeger constant $h(G_n) \leq \delta$. On the other hand, if A_n is more than half the vertices of G_n , we consider its complement A_n^c . Its cardinality $|A_n^c|$ satisfies

$$|A_n^c| \geq \text{vol}(A_n^c)/\Delta.$$

Hence,

$$h(G_n) \leq \frac{|\partial A_n^c|}{|A_n^c|} \leq \frac{|\partial A_n|\Delta}{\text{vol}(A_n^c)} \leq \frac{|\partial A_n|\Delta}{\text{vol}(A_n)} = \phi(G_n)\Delta \leq \delta.$$

540 In either case, we deduce that the Cheeger constant of G_n is at most δ , contradicting one of our
 541 hypotheses. Hence, there must have been only finitely many graphs satisfying the conditions of the
 542 theorem. \square

543 C Cayley graph at infinity is quasi-isometric to a tree

544 As all vertices of G_n look the same, we focus attention on $N_r(1)$, the r -neighbourhood of the identity
 545 vertex. The proof of Proposition 11 immediately gives the following.

Proposition 18. *Let r be a positive integer satisfying*

$$r < \frac{1}{2} \left(\log \left(\frac{1 + \sqrt{5}}{2} \right) \right)^{-1} \log(n-1).$$

546 *Then there is a graph isomorphism between the r -neighbourhood of the identity vertex in G_n and*
 547 *the r -neighbourhood of the identity vertex in G_∞ . This isomorphism takes the identity vertex to the*
 548 *identity vertex.*

Proof. As shown in the proof of Proposition 11, there is a graph homomorphism from $N_r(1)$ in G_∞ to $N_r(1)$ in G_n that is a surjection. If it fails to be an injection, then there is a non-trivial element K in the kernel K_n of $\text{SL}(2, \mathbb{Z}) \rightarrow \text{SL}(2, \mathbb{Z}_n)$ satisfying

$$\|K\| \leq \left(\frac{1 + \sqrt{5}}{2} \right)^{2r}.$$

But any non-trivial element K in K_n satisfies

$$\|K\| \geq n-1.$$

549 Rearranging gives the required inequality. \square

550 This raises the question of the local structure of G_∞ . The answer is well-known: it is ‘tree-like’.
 551 Specifically, it is quasi-isometric to a tree. The formal definition of quasi-isometry is as follows.

552 **Definition 19.** A *quasi-isometry* between two metric spaces (X_1, d_1) and (X_2, d_2) is a function
 553 $f: X_1 \rightarrow X_2$ that satisfies the following two conditions:

1. there are constants $c, C > 0$ such that, for every $x, x' \in X_1$

$$c d_1(x, x') - c \leq d_2(f(x), f(x')) \leq C d_1(x, x') + C,$$

- 554 2. there is a constant $K \geq 0$ such that for every $y \in X_2$, there is an $x \in X_1$ with $d_2(f(x), y) \leq K$.

555 If there is such a quasi-isometry, we say that (X_1, d_1) and (X_2, d_2) are *quasi-isometric*.

Table 3: Statistics of the three graph classification datasets studied in our evaluation.

Name	Number of graphs	Avg. nodes/graph	Avg. edges/graph	Metric
ogbg-molhiv	41, 127	25.5	27.5	ROC-AUC
ogbg-molpcba	437, 929	26.0	28.1	Avg. precision
ogbg-ppa	158, 100	243.4	2, 266.1	Accuracy
ogbg-code2	452, 741	125.2	124.2	F ₁ score

556 This forms an equivalence relation on metric spaces. When two metric spaces are quasi-isometric,
557 they are viewed as being ‘essentially the same’ at large scales.

558 When S and S' are finite generating sets for a group Γ , the graphs $\text{Cay}(\Gamma; S)$ and $\text{Cay}(\Gamma; S')$ are
559 quasi-isometric. Hence, the quasi-isometry type of a finitely generated group is well-defined, and this
560 is the central object of study in geometric group theory.

561 The group $\text{SL}(2, \mathbb{Z})$ has a finite-index subgroup that is a free group F [62]. If S' denotes a free
562 generating set for F , then $\text{Cay}(F; S')$ is a tree. As passing to a finite-index subgroup preserves
563 its quasi-isometry class, we deduce that the Cayley graph $G_\infty = \text{Cay}(\text{SL}(2, \mathbb{Z}); S_\infty)$ is indeed
564 quasi-isometric to a tree, as claimed above.

565 D Mixing time properties of expander graphs

566 Expanders are well known to have small mixing time, in the following sense.

567 Let G be a graph. We will consider probability distributions π on $V(G)$. The lazy random walk
568 operator M acts on probability distributions as follows. We think of $\pi(v)$ as being the probability of
569 the random walk being at vertex v . If the current location of the walk is at v , then at the next step
570 of the walk, either we stay put with probability $1/2$ or we move to one of its neighbours with equal
571 probability. Then $M\pi$ is the new probability distribution.

572 In the case when G is k -regular, this takes a particular simple form. The operator M is represented by
573 the matrix $(1/2)I + (1/2k)A$, where A is the adjacency matrix. In that case, any initial distribution
574 π converges under powers of M to the uniform distribution.

This is true for any reasonable notion of convergence, but we will use the $\|\cdot\|_1$ norm, where for two probability distributions π and π' ,

$$\|\pi - \pi'\|_1 = \sum_{v \in V(G)} |\pi(v) - \pi'(v)|.$$

Definition 20. The *mixing time* for a regular graph G is the minimum value of ℓ such that for any starting probability distribution π on the vertex set of G ,

$$\|M^\ell \pi - u\|_1 \leq \frac{1}{4}.$$

575 Here, u is the uniform probability distribution on the vertex set, and M is the lazy random walk
576 operator.

577 Expanders have small mixing times in the following very strong sense.

578 **Theorem 21.** For any $k > 0$ and $\delta > 0$, there is a constant $c > 0$ with the following property. If G is
579 a connected k -regular graph on n vertices with Cheeger constant at least $\delta > 0$, then the mixing time
580 for G is at most $c \log(n)$.

581 E Additional experimental details and ablations

582 **OGB dataset statistics.** We provide additional details on the dataset statistics for the OGB tasks
583 we used in Table 3. More substantial details can be found in the OGB paper [57].

584 **Ablations on propagation graph.** Our work concerns sparse expander graphs, determined using
585 the Cayley graphs of the special linear group. We acknowledge that this approach, while theoretically

Table 4: Comparative ablation performance of various propagation templates on ogbg-molhiv, ogbg-molpcba and ogbg-ppa. Our baseline model is a GIN [56], using exactly the same implementation as in [57]. All models have *exactly* the same number of parameters—we only modify the connectivity in certain layers depending on the scheme. **N.B.** The fully-connected graph, used in the FA approach [18] can be seen as a dense expander graph, i.e. a special case of EGP. 'OOT' indicates timeout, and 'OOM' indicates out-of-memory (on a V100 GPU).

Model	ogbg-molhiv	ogbg-molpcba	ogbg-ppa
GIN	0.7558 \pm 0.0140	0.2266 \pm 0.0028	0.6892 \pm 0.0100
GIN + master node	0.7668 \pm 0.0096	0.2527 \pm 0.0064	0.6916 \pm 0.0154
GIN + FA [18]	0.7850 \pm 0.0090	0.2595 \pm 0.0049	OOM
GIN + G-RLEF [33]	0.7802 \pm 0.0024	OOT	OOM
GIN + EGP (ours)	0.7934 \pm 0.0035	0.2329 \pm 0.0019	0.7027 \pm 0.0159

586 beneficial, is not the only possible way to aid global information propagation in a GNN. Therefore, in
587 this subsection we compare against other classes of approaches.

588 Our additional baseline methods include: GINs with a *master node*, GINs with a *fully connected*
589 layer (FA), as done in Alon and Yahav [18], and GINs with applying a recently proposed rewiring
590 method, G-RLEF [33].

591 Note that both the FA method and G-RLEF have motivations related to expanders: the fully-connected
592 graph in the FA method is a trivial *dense* expander, whereas G-RLEF’s rewiring iterations can converge
593 to an expander for certain input graph distributions. Therefore, comparing against these methods
594 allows us to also evaluate the impacts of expander density, as well as proximity to the input graph
595 (since G-RLEF iteratively modifies the input graph). We run G-RLEF for $O(V)$ steps.

596 The results of our ablative analysis are summarised in Table 4. We find that, as expected, all of our
597 added methods outperform the baseline GIN, demonstrating that oversquashing had been alleviated.
598 When comparing them against each other, however, we find that EGP tends to be highly competitive
599 on two out of the three datasets considered (having the largest average overall). The fully-adjacent
600 dense expander method remains strong on both ogbg-molhiv and ogbg-molpcba, but runs out of
601 memory as graphs increase in size (as is the case with ogbg-ppa).

602 We find that this collection of ablation studies further supplements the analysis of EGP we have
603 conducted, and serves as a good starting point for further investigations of expander propagation
604 templates with various properties.