Commute-Time-Optimised Graphs for GNNs

Igor Sterner [*1] **Shiye Su** [*1] **Petar Veličković** [1 2]

Editors: *S. Vadgama, E.J. Bekkers, A. Pouplin, S.O. Kaba, H. Lawrence, R. Walters, T. Emerson, H. Kvinge, J.M. Tomczak, S. Jegelka*

Abstract

We explore graph rewiring methods that optimise commute time. Recent graph rewiring approaches facilitate long-range interactions in sparse graphs, making such rewirings commute-time-optimal *on average*. However, when an expert prior exists on which node pairs should or should not interact, a superior rewiring would favour short commute times between these privileged node pairs. We construct two synthetic datasets with known priors reflecting realistic settings, and use these to motivate two bespoke rewiring methods that incorporate the known prior. We investigate the regimes where our rewiring improves test performance on the synthetic datasets. Finally, we perform a case study on a real-world citation graph to investigate the practical implications of our work.

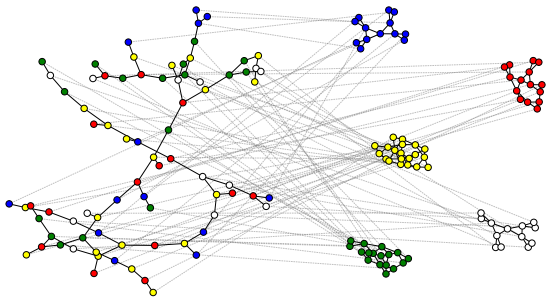


Figure 1. A base graph (left) and corresponding Cayley clusters rewiring (right). Each node takes one of four colours, or is uncoloured (white). Our Cayley clusters rewiring sparsely connects nodes of the same colour.

1. Introduction

Graphs are a natural data structure for a variety of relational networks such as those arising in chemistry, neuroscience, robotics and social sciences (Jing et al., 2020; DeZoort et al., 2023; Li et al., 2020). The dominant paradigm for learning on graph-based data is the graph neural network (GNN) (Gori et al., 2005), which models relational interactions by propagating messages along edges between neighbouring nodes. Improving GNNs' expressivity has become an area of attention in the machine-learning community (Bouritsas et al., 2022; Brody et al., 2022).

The message-passing mechanism suffers fundamentally from the oversquashing phenomenon, whereby bottlenecks in the graph topology result in the aggregation of large numbers of messages onto a small number of fixed-size vector messages. This makes global information propagation challenging, to the detriment of the GNN's expressivity. Topological studies on oversquashing show that negative curvature can give rise to these information bottlenecks (Topping et al., 2022). The Hessian of the learning target provides an alternate measure of oversquashing, quantifying the extent to which a GNN can 'mix' two nodes and model their pairwise interactions. This mixing ability is bounded by the GNN's weights and depth (Di Giovanni et al., 2024).

In practice, oversquashing would not challenge graph learning if the graph topology was highly aligned to the task, i.e. if nodes that need to interact were always connected. This motivates *rewiring*, a preprocessing step that constructs an alternate graph for message-passing (Alon & Yahav, 2020). A suitable rewiring may be found using gradient-based optimisation or reinforcement learning (Arnaiz-Rodríguez et al., 2022; You et al., 2018). These approaches however suffer from poor scaling due to the large combinatorial search space. Instead, there has been recent interest in non-parametric approaches, such as Stochastic Discrete Ricci Flow (Topping et al., 2022), which improves graph curvature, and expander graph propagation (Deac et al., 2022),

[*]Equal contribution [1]Department of Computer Science and Technology, University of Cambridge, UK [2]Google DeepMind. Correspondence to: Petar Veličković <petarv@google.com>.

Proceedings of the Geometry-grounded Representation Learning and Generative Modeling at the 41st International Conference on Machine Learning, Vienna, Austria. PMLR Vol Number, 2024.
Copyright 2024 by the author(s).

which improves average commute time. These are more stable, scalable, and theoretically motivated.

Existing non-parametric rewiring methods tend to alleviate oversquashing by optimising for *global* information propagation. In practice, however, domain experts have *priors* about which nodes need to interact, beyond what is represented in the input graph. Bottlenecks between two nodes that do not need to interact need not be relieved. Consider the problem of molecular property prediction, where atoms are modelled as nodes and chemical bonds are modelled as edges. Physical laws dictate that all node pairs interact, even distant ones in the original graph. This is particularly notable for protein molecules, where distant atoms in the primary or secondary structure become proximal in the folded tertiary form. Scientists' priors on which of these atoms should interact would offer a valuable headstart for a GNN.

To the best of our knowledge, our work is the first to attempt to incorporate such interaction priors into practical graph rewiring algorithms. We propose approaches to integrate input graphs with expert priors on which nodes should mix. The success of any such approach is contingent on having task-specific priors, and a reasonable means of using said prior in order to aid message passing. To disentangle these two effects, we begin by designing synthetic graphs where the priors are perfectly known and focus on developing non-parametric rewiring methods to incorporate them. As a case study, we then apply our methods to a real-world dataset with a naïve prior. Our contributions are as follows[1]:

- We introduce two new synthetic datasets for graph regression which mirror plausible real-world priors.
- For each of the two datasets, we introduce a non-parametric rewiring to incorporate the prior.
- We conduct extensive experiments to evaluate our proposed rewirings.
- We conduct a rewiring case study on `ogbn-arxiv`. Our results suggest that rewiring holds the potential for massive performance gains and faster convergence during training.

2. Expanders for Graph Propagation

A natural way to alleviate the information bottleneck between two nodes is to reduce their shortest path distance or commute time. Simultaneously, we desire a sparse solution to ensure tractable time and space complexity. Expander graphs, which are sparse yet low-diameter, satisfy these desiderata. The latter property ensures high connectivity across all nodes, since the diameter of an expander graph G scales logarithmically with the number of nodes,

$$\text{diameter}(G) \leq k \log |V(G)|, \quad k \in \mathbb{R}^+$$

a GNN with $\mathcal{O}(\log(|V(G)|)$ layers suffices to reach all pairwise node interactions.

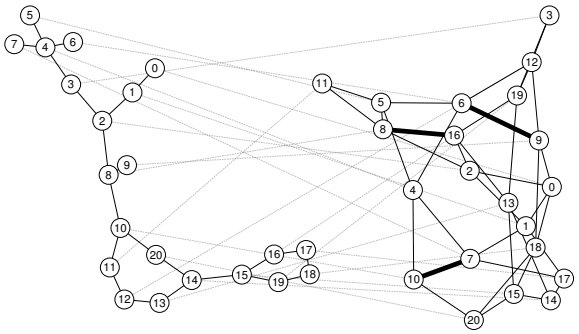


Figure 2. A base graph (left) and corresponding trimmed Cayley expander (right). Nodes are randomly allocated to the base graph nodes; the light grey lines show this random allocation. The three thick edges in the Cayley graph are edges which link nodes at distance 5 in the base graph, which is a prior in one of our synthetic datasets.

Deac et al. (2022) construct sparse expanders using Cayley graphs (see right of Figure 2 for a visualisation of a Cayley graph) generated by the special linear group $\text{SL}(a, \mathbb{Z}_n)$, where $a, n \in \mathbb{Z}^+$.

In our work, we follow their choice of $a = 2$ and the generating set

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

and similarly intervene on standard GNN message passing by propagating every other layer on the Cayley expander, instead of the base graph. Expander graphs are less susceptible to oversquashing because the Cheeger constants of a family of expander graphs provably satisfy a constant lower bound $\epsilon > 0$ (Alon, 1986). Therefore interleaving in this way retains node–neighbour interactions in the base graph and enables global message propagation in the interleaved graph. Hence, Deac et al. (2022)'s approach can help solve tasks with long-range dependencies. These ideas have been extended to address higher-order data correlations using hypergraphs (Christie & He, 2023).

Cayley graphs only exist for certain sizes. For our choice of the $\text{SL}(2, \mathbb{Z}_n)$ group, graph size grows $\mathcal{O}(n^3)$ per:

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

[1]Our code is available at https://github.com/igorsterner/commute-opt-gnn.

Following Deac et al. (2022), we trim Cayley graphs to arbitrary size $|V|$ by retaining only the first $|V|$ visited nodes in a breadth-first search. An example resulting graph is visualised in Figure 2. However, we note that trimming compromises the graph's expansion properties; keeping the excess nodes as 'virtual' zero feature nodes may be preferable (Wilson & Veličković, 2023).

A related body of work on graph transformers aims to enable global information propagation. Graph transformers enable attention-weighted interactions between all node pairs. But this approach incurs $\mathcal{O}(|V|^2)$ time and space complexity, and can suffer from a large number of redundant messages causing poor out-of-distribution (OOD) generalisation Di Giovanni et al. (2024). In particular, they show that to allow all nodes to mix in a complete graph, the magnitude of the model's weights need to grow linearly with the size of the graph. This is due to the fact that complete graphs—such as the ones used in graph Transformers—likely have redundant messages. In practice, over-specialising on identifying redundant messages on specific training distributions then leads to poor OOD performance on larger graphs. Exphormers (Shirzad et al., 2023) combine these two lines of work by using expanders as attention masks to sparsify the attention layers.

Existing works to date have been concerned with achieving scalable setups that are both sparse and optimise commute time between all nodes. But sharing information between all nodes is rarely a required characteristic of a graph. Instead, there will likely exist a subset of the graph nodes which are far apart on the base graph but need to interact to solve the task. In other words, these approaches are commute time optimal *on average*, but not if there is additional information available on which node pairs should interact most or least.

3. Synthetic Data Construction

In this section, we motivate and describe our synthetic datasets. For simplicity, we choose whole-graph regression tasks, but our ideas generalise to categorical targets and to node- or edge-wise tasks with the same underlying interactions. In both datasets, we begin with a graph $G = (V(G), E(G))$, where V and E are the graph nodes and edges. Node values are sampled from a uniform random variable, i.e. each node takes value sampled from $\mathcal{U}(0, 1)$.

We consider pairwise node interactions via exponential mixing. Our choice of exponential mixing is motivated by the Hessian-based maximal mixing view of oversquashing (Di Giovanni et al., 2024). Exponential interactions have high mixing and monotonic second derivatives. The latter property means it is easy to discern, from features only, which pairwise interactions are more salient for the task.

Specifically, we decompose our graph regression target into three components:

1. A component which depends on nearest-neighbour interactions. This justifies the input graph connectivity for the context and task at hand.
2. A component which depends on pairwise interactions between salient node pairs. These interactions are not necessarily captured in the base graph—an expert could inform these through a prior.
3. A component which depends on pairwise interactions between all remaining nodes. This reflects any residual interactions between all pairs of entities in a network not covered by prior components.

By studying different variants of these components, we construct two synthetic datasets.

Data A: Salient pair interactions. To concretely motivate this dataset, consider protein property prediction. The nearest-neighbour interactions could reflect bonds between consecutive residues on an amino acid chain; the salient node pairs could reflect proximal residues in the folded structure—which may be very distant in the input chain; the all-nodes component could reflect interactions between all residues in the protein due to pairwise electromagnetic force.

We encapsulate this idea in a synthetic dataset where we define salience using a *target* pairwise distance—i.e.,

$$y = c_1 \sum_{\text{dist}(i,j)=1} \exp(x_i + x_j) + c_2 \sum_{\text{dist}(i,j)=d} \exp(x_i + x_j) + c_3 \sum_{\text{dist}(i,j)\notin\{1,d\}} \exp(x_i + x_j) \tag{2}$$

$d \in \mathbb{Z}^+$, and $c_1, c_2, c_3 \in \mathbb{R}_0^+$ are fixed constant parameters, and x_i is a scalar feature given by the node's value. The three terms correspond in turn to the three aforementioned components. In particular, the second term in Equation 2 represents the interaction of nodes at distance d, which is the "target" distance at which salient interactions emerge.

Data B: Community interactions. This data construction reflects a prior that one or more *subset(s)* of the nodes will need to interact highly in the task. For example, in the protein property prediction task, there could be a property that depends highly on the structure of one atom type in the input graph. It is then desirable to minimise commute time between atoms of the same type.

We represent the above atom types as node colours. Let the set of possible colours be C, where $|C| \ll |V|$. We assign a subset of the nodes $V(c) \subset V$ to each colour $c \in C$, such

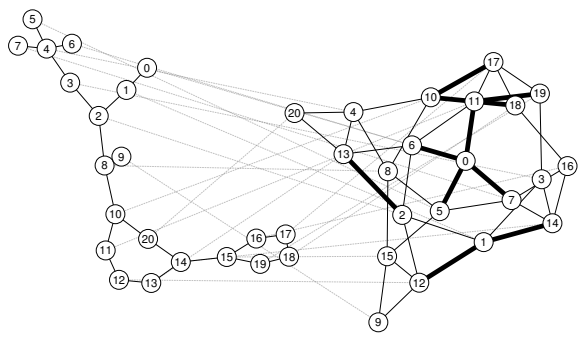


Figure 3. A base graph (left) and corresponding aligned Cayley expander informed by the prior (right). Compared to random Cayley graph placement (Figure 2), the number of important links captured by the aligned Cayley improves from 3 to 11 (dark edges). The Cayley graphs pictured here and in Figure 2 are isomorphic by construction, i.e. we have only changed how the nodes are aligned with the base graph.

that each node has at most one colour. Node features x_i are then a concatenation of node values and a one-hot encoding of c (or zero-vector for uncoloured nodes).

We define the regression target according to:

$$y = c_1 \sum_{\text{dist}(i,j)=1} \exp(x_i + x_j) + c_2 \sum_{c} \sum_{i,j \in V(c)} \exp(x_i + x_j) \tag{3}$$

where $c_1, c_2 \in \mathbb{R}_0^+$ are again fixed constant parameters. Alongside the same graph-dependent first term as in Equation 2, we introduce a new term here to model mixing between nodes of the same colour. In Data B, we implicitly set the term describing "all pairwise interactions" to zero. This is simply a design choice, since the colour term already entails a large number of connections.

Finally, we note that though we have described two specific priors, they convey two general, domain-agnostic classes covering pairwise and groupwise interactions respectively. We believe that successfully addressing these classes allows us to tackle many naturally occurring priors.

4. Method

We now propose bespoke rewiring constructions appropriate to the priors built into our two synthetic datasets. As discussed, these rewirings will be introduced by interleaved message passing alongside the base graph.

Data A The base graph naturally supports nearest neighbour interactions (the c_1 term). In addition, we desire a rewiring that can compromise between connecting all distance-d pairs (the c_2 term), and allowing all nodes to interact with each other (the c_3 term). Achieving the latter two objectives independently is straightforward: the former is attained by a graph composed of edges between all distance-d pairs; the latter by Cayley expanders if sparsity is also desired.

Algorithm 1 Greedy alignment of graphs g_1 and g_2.

Require: Graphs g_1, g_2 with the same number of nodes, all nodes initially unassigned
  $M = \{\}$
  while there exist unassigned nodes **do**
    $n_1 \leftarrow$ unassigned node in g_1
    $n_2 \leftarrow$ unassigned node in g_2
    $M[n_1] = n_2$
    while there exist unassigned nodes in n_1.neighbours **do**
      $\tilde{n}_1 \leftarrow$ unassigned node in n_1.neighbours
      $\tilde{n}_2 \leftarrow$ unassigned node in n_2.neighbours
      if $\tilde{n}_2$ exists **then**
        $M[\tilde{n}_1] = \tilde{n}_2$
      end if
    end while
  end while
  return M

In prior work, nodes in the interleaved graph are randomly matched onto nodes in the base graph. We propose to align the Cayley expander non-randomly over the nodes of the base graph, such that the salient distance-d edges are more likely to correspond to edges in this aligned Cayley expander.

This problem distills to the problem of finding the maximum common edge subgraph. As a generalisation of the subgraph isomorphism problem, it is NP-hard. We implement a cheap, greedy strategy (Algorithm 1) to identify a correspondence M the two graphs g_1 and g_2, where g_1 and g_2 are `distance-d-pairs` and `cayley` in our use case. An example result of our greedy approximate alignment is visualised in Figure 3. We note, however, that any other alignment scheme (such as backtracking tree-search (McGregor, 1982) or reinforcement learning (Bai et al., 2021)) could be used as a drop-in replacement in this step, depending on the computational budget and conviction in the prior.

Data B In addition to the base graph, which retains information about 1-hop neighbours, we aim to find a rewiring that enables nodes of the same colour to interact as easily as possible. Meanwhile, we again prefer sparse graphs for computational efficiency and OOD generalisation.

We already have the tool to sparsely connect a graph (or here, subgraph) with optimised commute time: the Cay-

ley expanders. We therefore generate separate Cayley expanders for nodes of each colour. We could connect the clusters by introducing sparse random connections between the clusters. Since here we have strong conviction in our prior (trivially, as it is synthetic data), we do not connect the clusters.[2] Our Cayley clusters rewiring is visualized in Figure 1.

5. Experiment

5.1. Experimental Design

Base graphs We construct our synthetic data on base graphs from real-world graph datasets, keeping the graph topology and throwing away node and edge features. For Data A, base graphs are sourced from ZINC (Gómez-Bombarelli et al., 2016), which represent real molecules of up to 38 atoms. We sample train graphs of size 20-30 and test graphs of size 30-35. The OOD design choice allows for a more challenging assessment of the model's ability to generalise, and also probes robustness to distributional shifts that may occur in real-world deployment. For Data B, we use larger base graphs so that each colour cluster is sufficiently large and non-trivial to sparsely connect. Graphs are sourced from `Peptides-struct` of LRGB (Dwivedi et al., 2022). We sample train graphs of size 75-125 and test graphs of size 125-175. In each graph, 25-75 nodes are randomly selected and assigned a random colour in c. All remaining nodes are uncoloured. For both datasets, we sample graphs in bins of 5 nodes to ensure an even spread of graphs of different sizes within the specified boundaries.

Model architecture Our architecture is based on the Graph Isomorphism Network (GIN) (Xu et al., 2019). This is a message passing GNN with the node update function taking the form of Equation 4. ϕ is an MLP adapted to include a batch normalization layer before ReLU activation. ϵ is a learnt parameter.

$$\mathbf{h}_u = \phi\left((1+\epsilon)\mathbf{h}_u + \sum_{v\in\mathcal{N}_v} \mathbf{h}_v\right) \tag{4}$$

Our final model architecture consists an input linear layer to 8 hidden channels, a stack of five GIN layers, additive pooling and then an output MLP to the regression target.

Learning setup A summary of the setup used for experiments on each dataset is shown in Table 1.

Rewirers For both Data A and Data B, `base-graph-only` is our baseline; it describes message passing on only the original base graph. We also run experiments on `cayley`, a Cayley expander over all nodes, and `fully-connected`, a clique over all nodes. The bespoke rewirings are: `aligned-cayley` and `distance-d-pairs` for Data A; `cayley-clusters` and a dense variant `fully-connected-clusters` for Data B. All rewired graphs interleaved with the base graph in each layer of message propagation.

[2] In our experiments, we found negligible difference from using connected and unconnected Cayley clusters.

Table 1. Learning setup for the two synthetic datasets

	Data A	**Data B**
Train graphs	5000	1500
min size	20	75
max size	30	125
batch size	32	32
Test graphs	500	300
min size	30	125
max size	35	175
batch size	32	32
Peak learning rate	0.0001	0.001
Total epochs	200	200
Warmup epochs	50	50
Learning rate decay/epoch	0.95	0.95
Distance d	5	-
Num colours	-	4

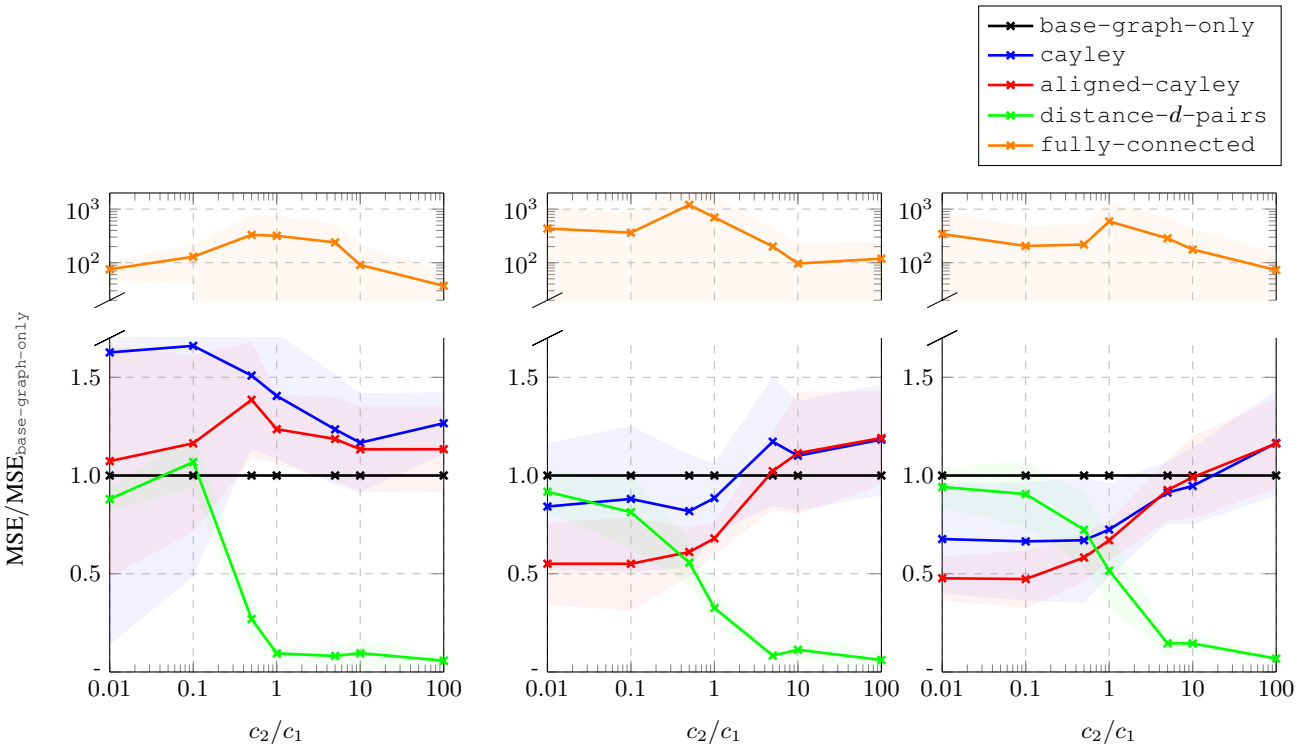
Metrics We measure model performance on the graph regression tasks by mean squared error (MSE) loss. All evaluation is performed OOD in graph size, as discussed. We report the means and standard deviations over three runs with randomly-seeded model initialisations. We compute, for each seed, the final validation set MSE of the rewired approach divided by the that of `base-graph-only` approach.

5.2. Results and Discussion

Data A Figure 4 shows sweeps over over c_2/c_1 for this dataset, at three values of c_3.

At $c_3 = 0$, standard message passing in `base-graph-only` outperforms both `cayley` and `aligned-cayley`. We surmise that the base graph is well-suited to a "distance counting" mechanism and hence capable of learning the distance d interactions in theory. Interleaving with rewirers corrupts this mechanism and impedes performance, while getting no wins from improving average commute times since $c_3 = 0$.

For $c_3 \neq 0$ and $c_2 < c_1$, we observe that `cayley` and `aligned-cayley` outperform the no-rewiring baseline. We argue that this regime is the most interesting regime



(a) $c_3 = 0.0$ (b) $c_3 = 0.1$ (c) $c_3 = 0.2$

Figure 4. A sweep of c_2/c_1 for three values of c_3 in Data A, fixing $c_1 = 1.0$. `base-graph-only` is the baseline with no rewiring. Each other colour denotes a rewirer introduced by interleaving with the base graph across GNN layers.

for realistic use cases. It has weak but nonzero interactions between all nodes ($c_3 \neq 0$), and moderate-strength special interactions that are weaker than 1-hop interactions on the base graph ($c_2 < c_1$). For example, at $c_3 = 0.2$ and $c_2 = 0.1$, the Cayley expander reduces MSEs to 70% and the aligned Cayley to 50% of the no-rewiring baseline. We believe `aligned-cayley` performs better than `cayley` on average because it captures a larger proportion of distance-d edges. As c_3 increases, the performance gap between `aligned-cayley` and `cayley` shrinks, because the relative importance of distance-d compared to global interactions weakens.

`distance-d-pairs` is a tempting rewiring solution. Results show that this rewirer allows the model to learn almost perfectly when c_2 is large – that is, when the special expert-provided priors dominate the type of interactions in the graph. However, this naive solution is suboptimal in other regimes. Namely, when $c_3 > 0$ and c_1 dominates c_2, `aligned-cayley` performs best by facilitating communication between all nodes while favouring distance-d interactions through our prior-informed alignment scheme.

Finally, `fully-connected` achieves the worst MSEs by several orders of magnitude across all tested parameter settings. We suspect the large number of redundant messages is responsible for this method's poor OOD generalisation.

Data B Figure 5 shows a sweep over c_2/c_1 for this dataset.

Consider the low c_2/c_1 regime, where nearest-neighbour interactions dominate. At $c_2/c_1 = 0.01$, the interleaved graphs and one-hot colour encodings are distractions. As expected, `base-graph-only` is here the best approach. Interleaving with `cayley`, `fully-connected-clusters`, or `fully-connected` results in worse MSEs by more than an order of magnitude. In contrast, `cayley-clusters` is relatively robust to the distracting colour encodings, though still suboptimal. We summarise this regime as follows: the prior correctly identifies existence of node colours, but they are irrelevant to the task. `cayley-clusters` learns this much better than the other rewiring solutions.

For moderate c_2/c_1, where both nearest-neighbour and colour interactions are important, we find that interleaving with either Cayley-style expander significantly outperforms `base-graph-only`. `cayley` and `cayley-clusters` are near-indistinguishable in this regime: the former achieves marginally lower MSEs and

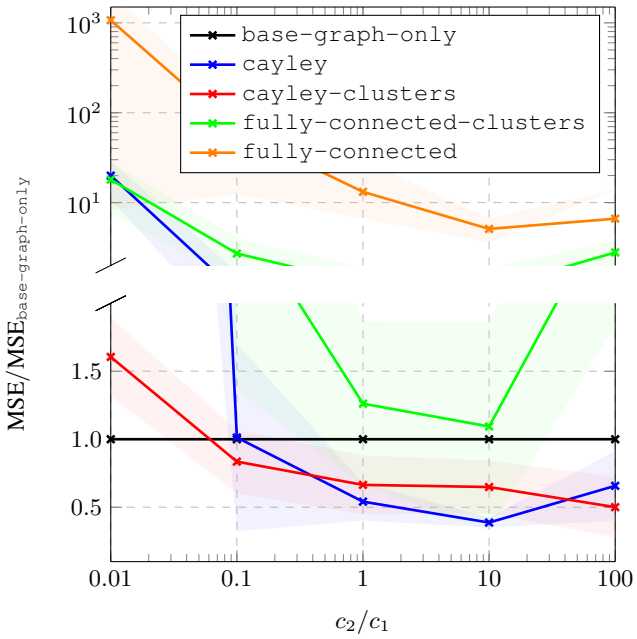


Figure 5. A sweep of c_2/c_1 in Data B, fixing $c_1+c_2=1$ (since the target grows very quickly for large c_2). `base-graph-only` is the baseline with no rewiring. Each other colour denotes a rewirer introduced by interleaving with the base graph across GNN layers.

the latter marginally smaller variance. This result is intuitive, as `base-graph-only` cannot capture all colour interactions, while the expanders make these colour connections accessible. Morever, we again find that sparse Cayley expanders are a better solution than dense cliques: `fully-connected` continues to perform very poorly, and `fully-connected-clusters` is hugely unstable while underperforming the baseline on average.

When c_2/c_1 is large, colour interactions dominate. At $c_2/c_1 = 100$, `cayley-clusters` outperforms all the other approaches on average, with `cayley` following closely behind. The Cayley rewirers outperform `base-graph-only`, which can only do well when same-colour nodes are close together on the base graph by chance. On the other hand, both fully-connected rewirers underperform `base-graph-only` by an order of magnitude. This is a particularly surprising result in the case of `fully-connected-clusters`, because this rewiring contains all node pairs that interact. The underperformance is reminiscent of the more general OOD issues in transformers, as discussed in Section 2. The largest cluster in `fully-conected-clusters` connects uncoloured nodes, representing our prior that such nodes should not interact. In our setup, the size of this cluster grows linearly (i.e. graph size minus number of coloured nodes) with the total number of nodes. This offers an explanation for the otherwise surprisingly poor performance of `fully-conected-clusters`, and further motivates our sparse rewiring approaches.

In summary, for this dataset, our `cayley-clusters` is consistently competitive with or better than other interleaving approaches. Meanwhile, it has consistently lower variance, lending the approach well to practical application where one has less certainty on the prior.

6. Case Study

The `cayley-clusters` rewiring aims to interact similar groups of nodes as much as possible. We now apply this rewiring approach to a real-world dataset, `ogbn-arxiv`. The task is to tranductively classify the subject area of papers in a citation network (Hu et al., 2020). We chose this dataset since citation graphs tend to exhibit homophily, making them well-suited for `cayley-clusters`. Additionally, `ogbn-arxiv` is sufficiently challenging that simpler flavours of GNNs models have not already oversaturated performance, in contrast to older citation graphs like Cora and PubMed.

We begin with a sanity check: can we boost performance by directly using ground truth class labels as a binary similarity measure? We compare `base-graph-only` and `cayley` with an interleaved `cayley-clusters` rewiring, where colours are assigned by the labels for all (train, val, and test) nodes. Though this constitutes data leakage, we use it to illustrate the potential performance improvement offered by our rewiring in the extreme case of a perfect prior. Additionally, we note that a fully-connected rewiring is computationally intractable; our rewiring is sparse.

Figure 6a shows the results of the sanity check. Compared to the baselines, where train accuracy converges to 67% and validation accuracy to 62%, `cayley-clusters` achieves high-90s% accuracy on train and validation nodes. This is consistent with our understanding that the Cayley clusters efficiently reinforce homophilic connections across the entire graph. Modifying the graph topology using ground truth labels, but *without* model parameter tuning on the validation labels, yields a massive performance boost.

We now turn to a more realistic scenario: without this perfect prior, can we extract a useful measure of similarity from node features? Inspired by Deac & Tang (2023), we train a weak MLP classifier on the train node features, ignoring the graph's connectivity. Colours are then assigned based on the predictions of this MLP, leading to the `mlp-cayley-clusters` rewiring. As controls, we also report the `cayley` baseline, and `random-cayley-clusters` where the colours are assigned randomly. Figure 6b shows that all methods reach comparable validation performance after approximately 200 epochs. However, our `mlp-cayley-clusters` enjoys

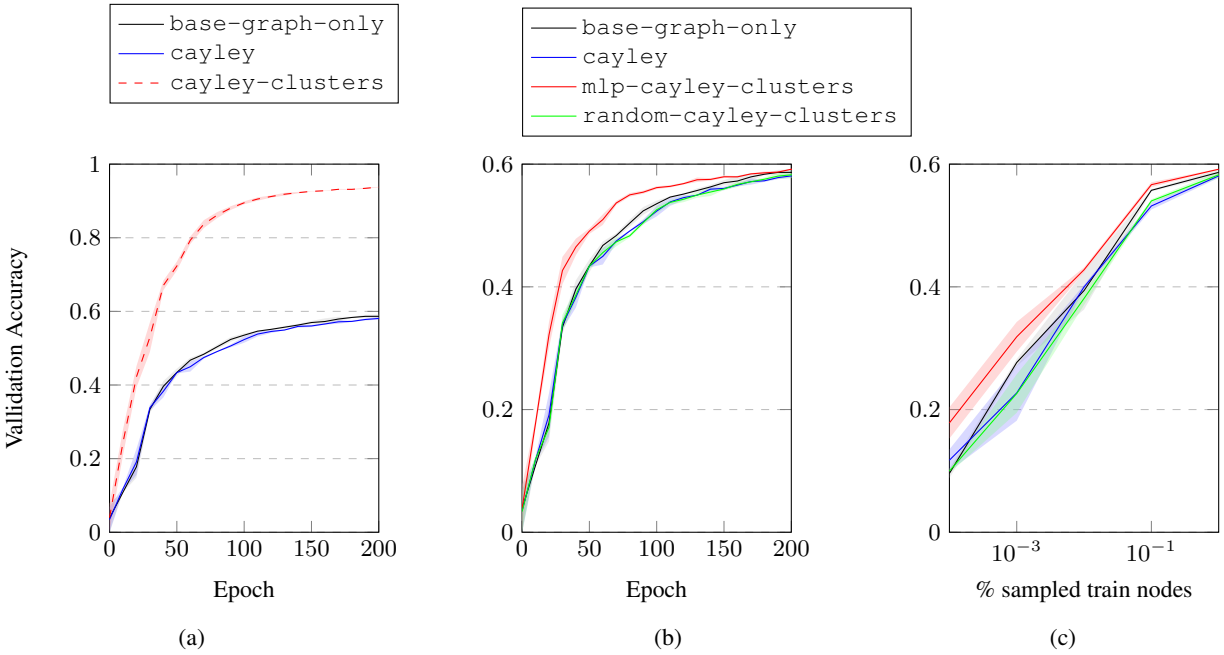


Figure 6. Results on `ogbn-arxiv`. (a) Validation accuracy vs. train epoch for the sanity check on `ogbn-arxiv`. Dashed lines represent cases where the ground truth labels are directly used to impute colours in the construction of the non-parametric rewirer. (b) Validation accuracy vs. train epoch shows that our `mlp-cayley-clusters` achieves faster convergence than baselines. (c) Final validation accuracy as a function of subsampling fraction shows that a valuable prior provided through the `mlp-cayley-clusters` rewiring improves final accuracies. Each line denotes the mean of three runs, with error bars representing one standard deviation.

substantially faster convergence. Notably, neither `cayley` nor `random-cayley-clusters` learns more quickly than the no-rewiring baseline. This suggests that the effective incorporation of our homophily prior, rather than a prior-agnostic reduction in global commute times, is responsible for the improved convergence.

Finally, we want to probe the intermediate situation where the prior offers information not available as labels. We simulate this effect by subsampling the train nodes when training the GNN, achieving a low-resourced setting where labels are sparse. Since our weak MLP classifier learns from the entire train set, its corresponding `mlp-cayley-clusters` rewiring offers information not otherwise available to the GNN. The more extreme the subsampling, the more "information-rich" the rewiring is. Results are reported in Figure 6c. As before, both `cayley` and `random-cayley-clusters` attain the same performance as the no-rewiring baseline. In contrast, `mlp-cayley-clusters` achieves statistically significant improvement to final validation accuracies when label data is sparse. This experiment shows that incorporating a prior is valuable in the low-data regime. As the label availability improves, the advantage of `mlp-cayley-clusters` narrows. Meanwhile, the `random-cayley-clusters` baseline demonstrates that if the prior turns out to be incorrect, our approach does not hurt performance.

For all experiments in this section, we used a GIN model with 3 layers, 128 hidden channels, 0.5 dropout probability, and batch norm. The model is trained for 200 epochs with 0.001 learning rate. We do not perform any hyperparameter tuning.

7. Conclusion

We began with the question: Given an expert prior on which node pairs need to interact in a graph learning problem, how can we optimise commute time between said pairs? We constructed two synthetic datasets that schematise two realistic graph priors and proposed non-parametric rewiring methods tailored to these priors. We investigated the regimes in which our graph rewiring methods outperform base-graph-only and Cayley expander baselines. In our case study, we found using a prior with our bespoke rewiring method leads to faster convergence on `ogbn-arxiv` and performance gains in low-data regimes.

Acknowledgements

We thank our anonymous reviewers for their comments, and Alex Vitvitskyi and Simon Osindero for reviewing the paper prior to submission.

References

Alon, N. Eigenvalues and expanders. *Combinatorica*, 6(2): 83–96, 1986.

Alon, U. and Yahav, E. On the bottleneck of graph neural networks and its practical implications. *CoRR*, abs/2006.05205, 2020. URL https://arxiv.org/abs/2006.05205.

Arnaiz-Rodríguez, A., Begga, A., Escolano, F., and Oliver, N. Diffwire: Inductive graph rewiring via the lov\'asz bound. *arXiv preprint arXiv:2206.07369*, 2022.

Bai, Y., Xu, D., Sun, Y., and Wang, W. Glsearch: Maximum common subgraph detection via learning to search. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 588–598. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/bai21e.html.

Bouritsas, G., Frasca, F., Zafeiriou, S., and Bronstein, M. M. Improving graph neural network expressivity via subgraph isomorphism counting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):657–668, 2022.

Brody, S., Alon, U., and Yahav, E. How attentive are graph attention networks? In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=F72ximsx7C1.

Christie, T. and He, Y. Higher-order expander graph propagation. In *NeurIPS 2023 Workshop: New Frontiers in Graph Learning*, 2023. URL https://openreview.net/forum?id=0lZjoxTZFb.

Deac, A. and Tang, J. Evolving computation graphs, 2023.

Deac, A., Lackenby, M., and Veličković, P. Expander graph propagation. In *NeurIPS 2022 Workshop on Symmetry and Geometry in Neural Representations*, 2022. URL https://openreview.net/forum?id=6cthqh2qhCT.

DeZoort, G., Battaglia, P. W., Biscarat, C., and Vlimant, J.-R. Graph neural networks at the large hadron collider. *Nature Reviews Physics*, 5(5):281–303, 2023.

Di Giovanni, F., Rusch, T. K., Bronstein, M., Deac, A., Lackenby, M., Mishra, S., and Veličković, P. How does over-squashing affect the power of GNNs? *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=KJRoQvRWNs.

Dwivedi, V. P., Rampášek, L., Galkin, M., Parviz, A., Wolf, G., Luu, A. T., and Beaini, D. Long range graph benchmark. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. URL https://openreview.net/forum?id=in7XC5RcjEn.

Gómez-Bombarelli, R., Duvenaud, D., Hernández-Lobato, J. M., Aguilera-Iparraguirre, J., Hirzel, T. D., Adams, R. P., and Aspuru-Guzik, A. Automatic chemical design using a data-driven continuous representation of molecules. *CoRR*, abs/1610.02415, 2016. URL http://arxiv.org/abs/1610.02415.

Gori, M., Monfardini, G., and Scarselli, F. A new model for learning in graph domains. In *Proceedings. 2005 IEEE international joint conference on neural networks, 2005.*, volume 2, pp. 729–734. IEEE, 2005.

Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., Catasta, M., and Leskovec, J. Open graph benchmark: Datasets for machine learning on graphs. *CoRR*, abs/2005.00687, 2020. URL https://arxiv.org/abs/2005.00687.

Jing, B., Eismann, S., Suriana, P., Townshend, R. J. L., and Dror, R. Learning from protein structure with geometric vector perceptrons. In *International Conference on Learning Representations*, 2020.

Li, Q., Gama, F., Ribeiro, A., and Prorok, A. Graph neural networks for decentralized multi-robot path planning. In *2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 11785–11792. IEEE, 2020.

McGregor, J. J. Backtrack search algorithms and the maximal common subgraph problem. *Software: Practice and Experience*, 12, 1982. URL https://api.semanticscholar.org/CorpusID:21921413.

Shirzad, H., Velingker, A., Venkatachalam, B., Sutherland, D. J., and Sinop, A. K. Exphormer: Scaling graph transformers with expander graphs, 2023. URL https://openreview.net/forum?id=8Tr3v4ueNd7.

Topping, J., Di Giovanni, F., Chamberlain, B. P., Dong, X., and Bronstein, M. M. Understanding over-squashing and bottlenecks on graphs via curvature. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=7UmjRGzp-A.

Wilson, J. and Veličković, P. Cayley graph propagation. In *NeurIPS 2023 Workshop on Symmetry and Geometry in Neural Representations*, 2023. URL https://openreview.net/forum?id=7jHGa1nS47.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019.

You, J., Liu, B., Ying, Z., Pande, V., and Leskovec, J. Graph convolutional policy network for goal-directed molecular graph generation. *Advances in neural information processing systems*, 31, 2018.