Density-Accuracy Tradeoffs in Real-World Graph Spanners*

Nadia Kōshima*† Tim Rieder*‡ Sabyasachi Basu§ Talya Eden¶ Omri Ben-Eliezer‖

Abstract

Graph spanners are a classical and well-studied concept in graph theory with a plethora of algorithmic applications. A spanner is a sparse spanning subgraph which preserves distances in the original graph up to some additive or multiplicative stretch. The sparsity and approximation guarantees raise a natural question: can spanners serve as (the backbone of) space-efficient data structures for approximate distance computation in real-world networks?

Toward this goal, we make the first systematic study of the size and accuracy (additive error) of existing spanner construction on real-world networks. Our work exhibits a number of surprising findings, which shed light on the structure of these networks, and may inspire downstream algorithm design. As it turns out, state of the art techniques based on low diameter decompositions tend to produce very sparse, almost tree-like, spanners in several large real-world networks. However, the real-world additive error of these spanners is suboptimal, and we show that slightly denser constructions offer a more accurate alternative.

1 Introduction

Graph spanners—sparse subgraphs that approximately preserve distances—are widely studied due to their applications in routing, data compression, and distributed computing, among others. They allow for efficient, on-demand distance estimation without storing or repeatedly processing the entire graph, making them valuable for large-scale graph analysis tasks.

Despite their theoretical appeal, the empirical accuracy of spanners for approximate distance estimation remains largely unexplored. To address this gap, we conduct a systematic study, evaluating spanners on medium-sized social and information networks. We also introduce a new spanner leveraging the core-periphery structure of social networks and demonstrate its competitive performance against existing methods, which often overlook these structural insights.

Problem definition. Real-world social and information networks have extremely small distances, with the 90th percentile diameter often around 4 or 5. This makes additive error the most meaningful accuracy measure. A subgraph G' of G is an additive k-spanner if

$$d_{G'}(u, v) \leq d_G(u, v) + k.$$

for all vertex pairs u, v. Ideally, we seek very small k, such as 1 or 2. That said, given the complex, beyond-worst-case nature of real-world networks, assessing spanner quality via the *average error* may be more informative than enforcing a universal bound k.

Existing spanner constructions. Numerous spanner constructions have been studied in the theoretical literature, but these are less relevant for sparse real-world networks with small diameters and near-linear size constraints. Indeed, for worst-case graphs, Abboud and Bodwin [1] show that restricting the spanner to $O(n^{4/3-\epsilon})$ edges requires polynomial additive stretch.

Thus, we focus on practical techniques: two methods by Miller et al. [4] and Forster et al. [3] based on low-diameter decompositions (LDDs), and a heuristic by Pathak et al. [5] that improves sparsity using high-coverage independent sets. We refer to the original methods as MPVXbase and FGVbase, and their enhanced versions as MPVXcompact and FGVcompact.

A Core-Periphery spanner. Basu et al. [2] recently showed that the core-periphery structure of real-world networks enables efficient shortest path approximation with sublinear time and space. Their approach partitions the graph into a small inner core L_0 , an outer core L_1 consisting of L_0's neighbors, and a periphery L_2 of all remaining vertices, and all paths are forcibly routed through L_0. Building on this, we introduce the *CP-spanner*, which for every vertex in L_0 retains only the edge to the highest-degree neighbor in L_1. This simple modification effectively sparsifies the graph.

Dataset	$\|V\|$	$\|E\|$	Dataset	$\|V\|$	$\|E\|$
Euemail	1K	16.1K	Livejournal	4.0M	34.7M
Skitter	1.7M	11.1M	Hollywood	1.1M	56.3M

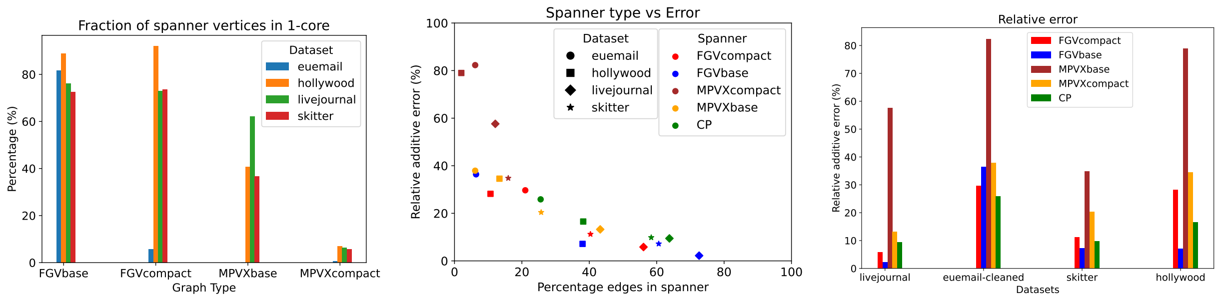
Table 1: Datasets used in experiments

2 Experimental Results

We conduct several experiments on four representative datasets, see Table 1. A striking observation is that

*Equal contribution

†Independent Researcher, nadianw36@gmail.com
‡ETH Zurich, timrieder@ethz.ch
§UC Santa Cruz, sbasu3@ucsc.edu
¶Bar-Ilan University, talyaa01@gmail.com
‖Technion, omribene@cs.technion.ac.il



(a) 1-core of the spanners

(b) Sparsification vs Error

(c) Percentage additive error by graph

Figure 1: (a) Percentage of vertices in 1-cores of spanners. (b) The x-axis is the size of the spanner as fraction of the original graph, and the y-axis is the average additive error as a percentage of the true distance. (c) Percentage error by spanner type for different datasets.

many spanners are extremely sparse, often with only a few more edges than vertices, suggesting a tree-like structure. To investigate this, we compute the 1-core of these graphs. While not a perfect metric for tree-likeness (as the NP-complete Feedback Vertex Set problem provides a definitive answer), a tree's 1-core is empty, making it a useful indicator. Indeed, as shown in Figure 1(a), the sparsest spanners retain very few vertices in their 1-cores, with the MPVX spanners yielding an empty graph for the Euemail dataset.

Next, we examine the relation between the sparsity and the accuracy of the resultant spanenrs. Figure 1(b) shows that extremely sparse MPVX spanners tend to have high additive error, though their average error remains well below the worst-case multiplicative stretch (17 for MPVX, 7 for FGV). We also profile CP-spanners, which perform similarly to FGVbase in both accuracy and density. Figure 1(c) further breaks down errors by dataset. We note that better hyperparameter choices could enhance CP-spanner performance.

In our final set of experiments, we examine how the edges retained in LDD-based spanners align with the core-periphery structure (Figure 2). We find no consistent correlation across datasets—the fraction of preserved edges from each part varies significantly. This suggests that LDD methods select spanner edges independently of the core-periphery structure.

3 Future Directions

We highlight an interesting and likely inherent tradeoff between accuracy (additive error) and density (% of edges, compared to original graph) in spanners with low additive error for real-world networks. In our ongoing effort we perform a systematic study of the structure and guarantees of such spanners, considering a wider range of spanner constructions and conducting a large variety of experiments in the spirit of those proposed in this paper. We hope the study of empirically-efficient spanners will lead to their deployment in real-world data structures for distance computation and routing.

Acknowledgements As per SIAM policy, we acknowledge the use of ChatGPT for grammatical improvements in a few paragraphs.

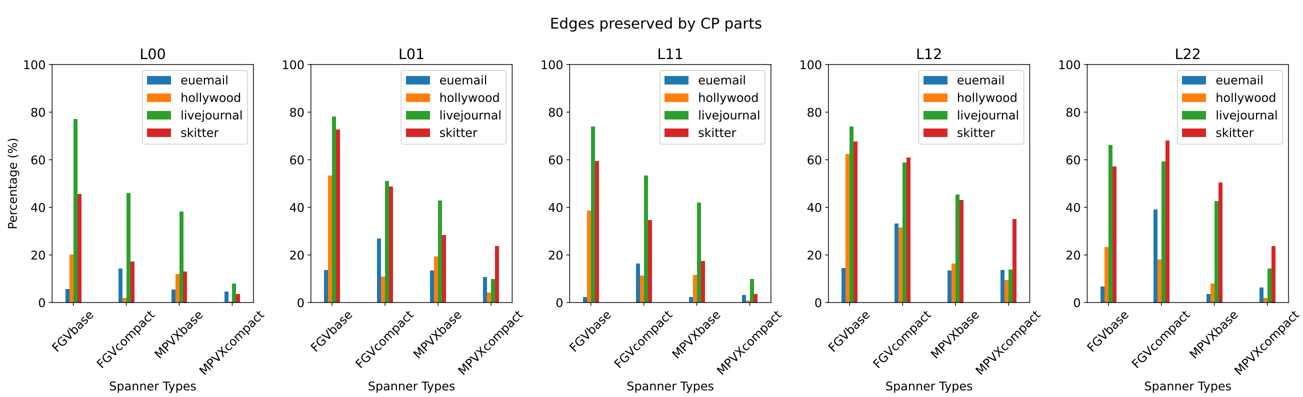


Figure 2: Percentage of edges from L_{xy} (one end point is in L_x and the other in L_y) preserved in LDD spanners.

References

[1] Amir Abboud and Greg Bodwin. The 4/3 additive spanner exponent is tight. *Journal of the ACM (JACM)*, 64(4):1–20, 2017.

[2] Sabyasachi Basu, Nadia Kōshima, Talya Eden, Omri Ben-Eliezer, and C. Seshadhri. A sublinear algorithm for approximate shortest paths in large networks. In *18th ACM International Conference on Web Search and Data Mining (WSDM 2025)*, 2025. To appear.

[3] Sebastian Forster, Martin Grösbacher, and Tijn de Vos. An Improved Random Shift Algorithm for Spanners and Low Diameter Decompositions. In *25th International Conference on Principles of Distributed Systems (OPODIS 2021)*, volume 217, pages 16:1–16:17, 2022.

[4] Gary L. Miller, Richard Peng, Adrian Vladu, and Shen Chen Xu. Improved parallel algorithms for spanners and hopsets. In *Proceedings of the 27th ACM Symposium on Parallelism in Algorithms and Architectures*, SPAA '15, pages 192–201, 2015.

[5] Maulein Pathak, Yogish Sabharwal, and Neelima Gupta. Scalable algorithms for compact spanners on real world graphs. In *Proceedings of the 37th ACM International Conference on Supercomputing (ICS 2023)*, pages 386–397, 2023.