Graph Sparsifiers

Hrishik Koley

LIMIT Offline Camp 2026

May 9, 2026

Outline

Motivation from Games

Think of a large game world such as *Minecraft*, *Elden Ring*, or *GTA VI*.

At any moment, the world contains many entities:

- players and NPCs,
- bullets, vehicles, and obstacles,
- trees, walls, buildings, and other objects.

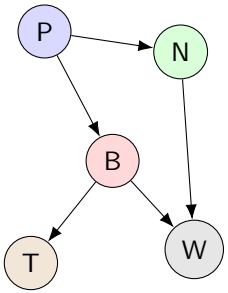
The physics engine must constantly reason about which objects may interact.

The World as a Graph

Our model starts with a graph:

- every object is a node,
- an edge (u, v) exists if the two objects can interact,
- for us, the main interaction is when collision is possible.

So a game world gives rise to an interaction graph.



P: player, N: NPC, B: bullet, T: tree, W: wall

A Probabilistic Weighted Interaction Graph

This is not just a graph. This is a probabilistic weighted interaction graph.

For each edge (u, v) we attach two quantities:

- Weight $w(u, v)$: importance of the interaction
  - damage potential,
  - gameplay relevance,
  - strategic significance.
- Probability $p(u, v)$: likelihood of collision
  - distance,
  - velocity,
  - behavior patterns.

Why Not Keep Every Edge?

In a dense world, the number of possible interactions can be enormous.

- If every object may interact with many others, the graph becomes very large.
- Collision detection, path planning, and physics updates become expensive.
- Most edges are rarely important at the same time.

We therefore ask: can we keep only a small set of representative edges?

The Sparsification Question

Given a large interaction graph G, we want a much sparser graph H such that:

- important interactions are still represented,
- distances or cuts are approximately preserved,
- algorithmic tasks remain fast and meaningful.

In this talk, we will look at several ways to formalize what it means to preserve structure.

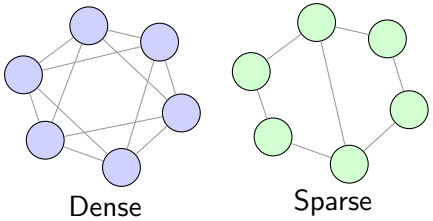
Graphs, Dense Graphs, Sparse Graphs

A graph $G = (V, E)$ has:

- vertices in V,
- edges in E joining pairs of vertices.

A graph is:

- dense if it has many edges,
- sparse if it has relatively few edges.



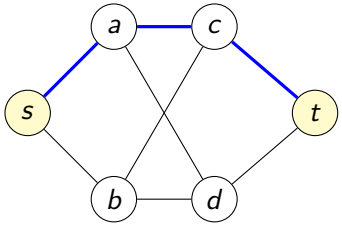
Paths and Shortest Paths

A path is a sequence of vertices connected by edges.

The length of a path is:

- the number of edges in an unweighted graph, or
- the total weight in a weighted graph.

A shortest path is a path of minimum length between two vertices.



blue edges indicate one shortest path

Adjacency Matrix and Laplacian Matrix

Suppose $G = (V, E)$ has vertices $V = \{1, 2, \ldots, n\}$.
The adjacency matrix $A = (a_{ij})$ is defined by

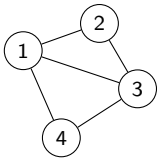
$$a_{ij} = \begin{cases} 1, & \text{if } \{i, j\} \in E, \\ 0, & \text{otherwise.} \end{cases}$$

For a weighted graph, we replace 1 by the edge weight w_{ij}.
The degree matrix $D = (d_{ij})$ is the diagonal matrix defined by

$$d_{ij} = \begin{cases} \deg(i), & \text{if } i = j, \\ 0, & \text{if } i \neq j. \end{cases}$$

The Laplacian matrix is then defined by

$$L = D - A.$$



Example graph

$$A = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

$$L = \begin{pmatrix} 3 & -1 & -1 & -1 \\ -1 & 2 & -1 & 0 \\ -1 & -1 & 3 & -1 \\ -1 & 0 & -1 & 2 \end{pmatrix}$$

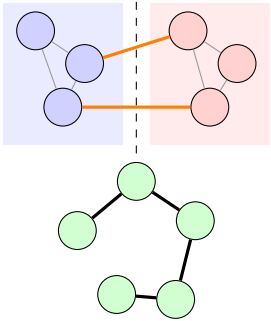
Cuts and Spanning Trees

A cut splits the vertex set into two parts.
The edges with one endpoint on each side form the cut-set.

A spanning tree is a subgraph that:

- contains all vertices,
- is connected,
- has no cycles.

It always has exactly $|V| - 1$ edges.



Common Sparsifiers

Cut Sparsifier: Let $G = (V, E, w)$ be a weighted undirected graph. A weighted graph $G' = (V, E', w')$ is an ϵ-cut sparsifier of G if it approximately preserves every cut. That is, for all $S \subseteq V$,

$$w_{G'}(S, \bar{S}) \in (1 \pm \epsilon)\, w_G(S, \bar{S})$$

In words, every cut value is approximately preserved.

Spectral Sparsifier: H is a spectral sparsifier for G with parameter ϵ if for every $x \in \mathbb{R}^n$,

$$(1-\epsilon)x^T L_G x \leq x^T L_H x \leq (1+\epsilon)x^T L_G x$$

In words, a spectral sparsifier preserves the Laplacian quadratic form, so it approximately preserves the graph's global connectivity structure.

Why Study Distance Preservation?

In many graph problems, the key information is metric rather than cut-based.

We care about:

- how far two states or objects are from one another,
- how quickly information or influence propagates,
- which paths are available for routing, navigation, or escape.

This leads to two natural goals:

- preserve some distances *exactly*, or
- preserve all distances *approximately*.

Exact vs Approximate Distance Preservation

Two closely related notions are:

- Distance preservers: preserve selected distances exactly.
- Spanners: preserve all distances approximately.

They answer two different questions:

- When do we need exact shortest-path information for a chosen set of pairs?
- When is bounded stretch enough for the whole graph?

The first notion is stricter, while the second usually allows much sparser subgraphs.

Distance Preservers: Definition

Let $P \subseteq V \times V$ be a set of ordered pairs.
A subgraph $H = (V, E_H)$ is a distance preserver for P if

$$d_H(u, v) = d_G(u, v) \qquad \text{for every } (u, v) \in P.$$

Important special cases:

- $P = \{s\} \times V$: all distances from one source,
- $P = S \times V$: all distances from a set of sources,

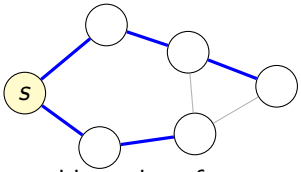
Single-Source Distance Preservers

If we only want to preserve distances from one source s, a shortest-path tree already works.

So for $P = \{s\} \times V$ we get a preserver with exactly $n - 1$ edges.

Thus in this example of sparsification:

- exact distances are preserved,
- the subgraph is very sparse,



blue edges form a shortest-path tree from s

Formally, a $\{s\} \times V$ distance preserver of a graph $G = (V, E)$ and a node $s \in V$ is a subgraph $H = (V, E_H)$ with $E_H \subseteq E$ such that $d_H(s, v) = d_G(s, v)$ for every $v \in V$.

Bounds and Forbidden Structures for Distance Preservers

Theorem (Forbidden Subgraphs)

The resulting graph H contains no $C_{\leq n}$ subgraphs, that is, no cycles.

Theorem (Extremal Bound)

The maximum number of edges in an n-node graph with no such forbidden subgraphs is $ex(n, C_{\leq n}) \leq n - 1$.

- **Structural Upper Bound:** There always exists an $\{s\} \times V$ distance preserver with no forbidden subgraphs.
- **Matching Lower Bound:** If the original graph already contains no forbidden subgraphs, then no edges can be removed.
- **Best possible bound:** $|E(H)| = n - 1$.

Spanners: Definition

A subgraph $H = (V, E_H \subset E)$ is a k-spanner of $G = (V, E)$ if for every $u, v \in V$,

$$d_H(u, v) \leq k \, d_G(u, v).$$

Here we relax exactness:

- every pairwise distance is allowed to stretch by at most a factor k,
- in exchange, the subgraph can be much sparser than G.

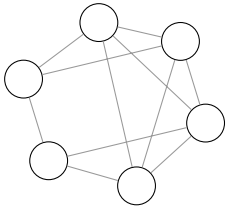
A 3-Spanner Example

Let $k = 2$. Then a $(2k - 1)$-spanner is a 3-spanner.

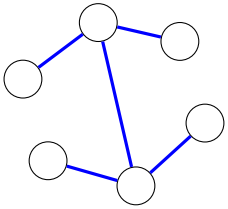
The graph on the left is the original graph G.

The graph on the right is a sparse subgraph H in which every distance is stretched by at most a factor of 3.

Some edges are deleted, but short detours still remain.



G



H

H is a 3-spanner

Greedy $(2k-1)$-Spanners

A classical greedy rule is:

- inspect edges in nondecreasing order of weight,
- add an edge (u, v) only if the current graph does not already contain a path from u to v of length at most $(2k-1)w(u, v)$.

This rule is natural because:

- if a short enough detour already exists, the edge is unnecessary,
- if not, the edge encompasses genuinely missing metric information,

Theorem

For every n-node graph G and integer $k \geq 1$, there exists a $(2k-1)$-spanner H with $|E(H)| \leq O(n^{1+1/k})$.

Bounds and Forbidden Structures for $(2k-1)$-Spanners

Theorem (Forbidden Subgraph)

The resulting graph H contains no $C_{\leq 2k}$ subgraphs.

Theorem

The maximum possible number of edges in an n-node graph without $C_{\leq 2k}$ subgraphs is $ex(n, C_{\leq 2k}) \leq O(n^{1+1/k})$.

- **Structural Upper Bound:** There exists a $(2k-1)$-spanner with no forbidden subgraphs.
- **Matching Lower Bound:** If the original graph already avoids the forbidden subgraphs, no edges can be removed.
- So, the best possible bound for spanners is $ex(n, C_{\leq 2k})$, but the exact value is still unknown in general.

What the Bounds Mean Conceptually

The spanner theorem expresses a real tradeoff:

- if we want smaller stretch, we must usually keep more edges,
- if we allow larger stretch, we can compress the graph more aggressively.

For example:

- a 3-spanner typically needs about $O(n^{3/2})$ edges,
- a 5-spanner needs about $O(n^{4/3})$ edges,
- in general a $(2k-1)$-spanner needs about $O(n^{1+1/k})$ edges.

The bounds on $ex(n, C_{\leq k})$ are called the Moore bounds, and while the general Moore bound on $ex(n, C_{\leq 2k})$, $ex(n, C_{\leq 2k-1}) \leq O(n^{1+1/k})$ is known, we do not know whether the bound is tight or not.

Demand-Aware Sparsification

Sometimes we do not care equally about all pairs of vertices.

The input now consists of:

- a graph $G = (V, E)$,
- a demand set $D \subseteq V \times V$ of pairs that matter most.

The goal is to build a sparse graph H that still serves the important pairs in D. Depending on the application, this may mean preserving:

- distances,
- reachability, or
- routing quality for those demands.

Demand Pairs and Distance Preservers

Let $P \subseteq V \times V$ be a set of demand pairs.

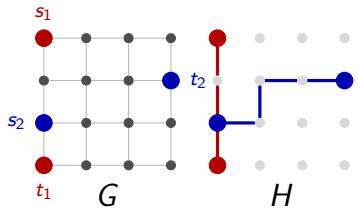
Definition

A P-distance preserver of a graph $G = (V, E)$ is a subgraph $H = (V, E_H)$ with $E_H \subseteq E$ such that

$$d_H(s,t) = d_G(s,t) \qquad \text{for every } (s,t) \in P.$$

This generalizes earlier examples:

- if $P = \{s\} \times V$, we recover single-source distance preservers,
- for larger P, we keep exact distances only for the pairs that matter.



Example: G and a sparse subgraph H preserving two demand pairs.

How Sparse Can a P-Distance Preserver Be?

Theorem (Coppersmith–Elkin, 2006)

Every n-node undirected graph, and more generally every DAG, has a P-distance preserver H with

$$|E(H)| \leq O\Big(\min\{n|P|^{1/2},\ n^{1/2}|P|\} + n\Big).$$

Theorem (Coppersmith–Elkin, 2006; Bodwin, 2018)

Every n-node directed graph has a P-distance preserver H with

$$|E(H)| \leq O\Big(\min\{n|P|^{1/2},\ n^{2/3}|P|\} + n\Big).$$

Corollary

- If $|P| = n^{1/2}$, then $O(n)$ edges suffice in the undirected case.
- If $|P| = n^{1/3}$, then $O(n)$ edges suffice in the directed case.

A Simple Construction Principle

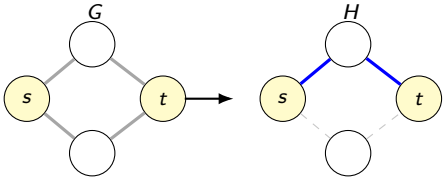
Observation

A P-distance preserver can be obtained by taking, for each demand pair $(s, t) \in P$, one shortest path $\pi(s, t)$ in G, and keeping the union of all these paths.

Key point: the only real algorithmic choice is how we break ties when several shortest paths exist.

Intuition: large preservers arise when there is little room for either

- overlap between demand paths, or
- helpful tie-breaking among equal-length paths.



Two equal shortest paths in G; tie-breaking decides which one enters H.

Worst-Case Intuition: Independence

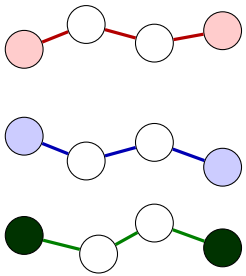
The hardest instances are the ones where shortest paths give us the least opportunity to simplify:

- each demand pair has a unique shortest path,
- different demand pairs use edge-disjoint shortest paths.

Independence Lemma (Bodwin–Hoppenworth–Trabelsi, 2023)

For weighted graphs, upper-bound analyses may reduce to the case where all demand pairs have unique edge-disjoint shortest paths.

So, when we seek worst-case upper bounds, it is enough to understand these highly independent path systems.



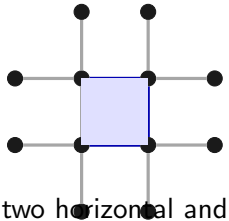
A stylized worst case: unique shortest paths with no shared edges.

Why Direct Forbidden-Subgraph Arguments Fail

Even after the independence reduction, the union of these unique edge-disjoint shortest paths can still be extremely rich.

- In fact, their union can contain arbitrary subgraphs.
- So forbidden-subgraph arguments on the original graph, or directly on the preserver H, do not by themselves bound $|E(H)|$.

Example: unique shortest paths can still induce a C_4.



two horizontal and two vertical demand paths create the middle 4-cycle

The Incidence-Graph Viewpoint

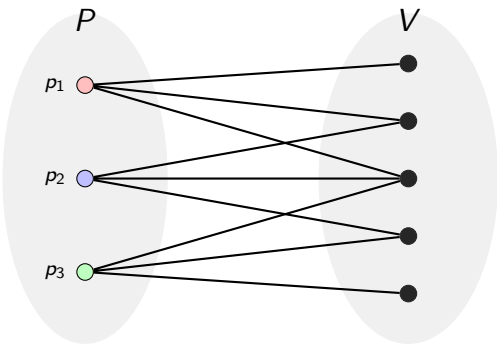
Idea: instead of studying the preserver directly, encode which vertices belong to which shortest paths.

Construction

Let $\pi(p)$ denote the chosen shortest path for demand pair $p \in P$. Define the incidence graph

$$I = (V \cup P, E_I), \qquad (v, p) \in E_I \iff v \in \pi(p).$$

Interpretation: the bipartite graph I records membership of vertices in shortest paths.



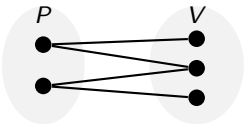
Left side: demand pairs. Right side: graph vertices. An edge means the vertex lies on the chosen shortest path for that demand.

What the Incidence Graph Tells Us

Undirected graphs

- Unique edge-disjoint shortest paths satisfy strong consistency constraints.
- Those constraints imply the incidence graph is C_4-free.
- Extremal bounds for bipartite C_4-free graphs then recover

$$O\left(\min\{n|P|^{1/2},\ n^{1/2}|P|\} + n\right).$$

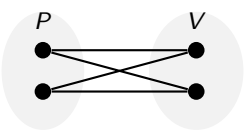


C_4 forbidden

Directed graphs

- The same undirected consistency argument breaks down.
- A C_4 may appear in the incidence graph even with unique edge-disjoint shortest paths.
- That is why the directed upper bound needs extra ideas, and ends up with the weaker

$$O\left(\min\{n|P|^{1/2},\ n^{2/3}|P|\} + n\right).$$



C_4 allowed

Undirected Case: Why C_4 Is Forbidden

Independence Lemma

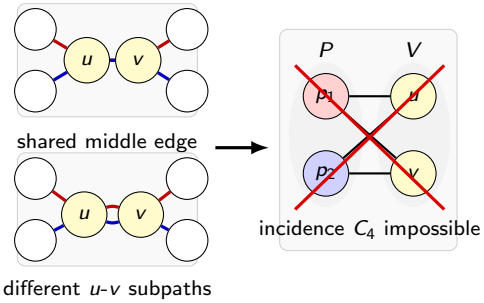
In undirected graphs, unique edge-disjoint shortest paths avoid two basic patterns:

- two demand paths cannot share an edge,
- if two paths meet twice, then the subpath between the meetings must be consistent.

Lemma 1 (Forbidden Subgraphs)

The incidence graph of unique edge-disjoint shortest paths contains no C_4.

Intuitively, a C_4 means two demand pairs both use the same two vertices, forcing one of the forbidden path patterns above.



Undirected Case: Extremal Bound and Consequence

Lemma 2 (Extremal Bound)

The maximum number of edges in an $(n, |P|)$ bipartite graph without C_4 is

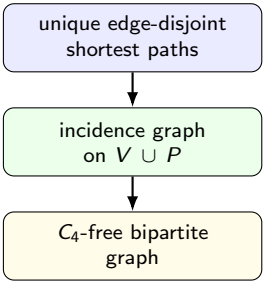
$$\mathrm{ex}(n, |P|, C_4) = O\Big(\min\{n|P|^{1/2},\ n^{1/2}|P|\}\Big).$$

Since the incidence graph lives on $V \cup P$, controlling its size also controls the shortest-path system and hence the preserver.

Undirected consequence

Hence every undirected graph has a P-distance preserver H with

$$|E(H)| \le O\Big(\min\{n|P|^{1/2},\ n^{1/2}|P|\} + n\Big).$$



Directed Case: Why the Same Argument Fails

Observation

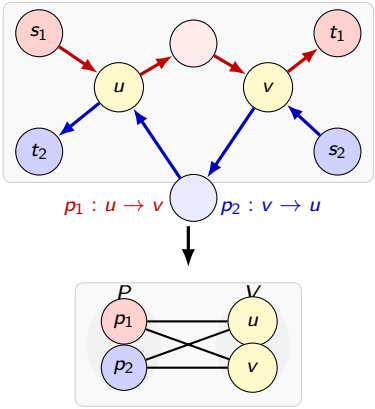
The same argument as in the undirected case does not apply directly.

Reason. A plain C_4 only tells us that two demand paths both contain the same two vertices.

In a directed graph, one path may see u before v, while another sees v before u. The incidence graph forgets this order information.

So a C_4 may still appear even with unique edge-disjoint shortest paths, and C_4 is no longer the right forbidden pattern.

We therefore need a more refined, order-sensitive obstruction.

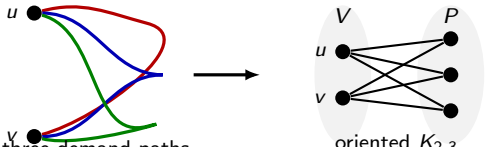


membership only; order lost

Directed Case: Oriented $K_{2,3}$ and the Final Bound

Independence Lemma

In directed graphs, three demand paths cannot all pass through the same two vertices in a unique edge-disjoint shortest-path system.



Lemma 1 (Forbidden Subgraphs)

The incidence graph avoids an oriented $K_{2,3}$, where the orientation remembers the order in which the two graph vertices appear along each path.

Lemma 2 (Extremal Bound) and Theorem

$$\mathrm{ex}(n, |P|, K_{2,3}) = O\Big(\min\{n|P|^{1/2},\ n^{2/3}|P|\}\Big).$$

Therefore every directed graph has a P-distance preserver H with

$$|E(H)| \le O\Big(\min\{n|P|^{1/2},\ n^{2/3}|P|\} + n\Big).$$

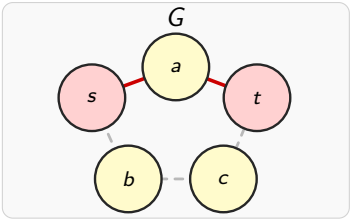
Reachability Preservers

Definition

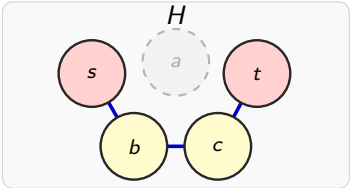
A P-reachability preserver of a directed graph $G = (V, E)$ is a subgraph $H = (V, E')$ such that

$$\text{dist}_G(s,t) < \infty \iff \text{dist}_H(s,t) < \infty \qquad \forall (s,t) \in P.$$

- We preserve only the existence of a directed path.
- So H may delete shortcuts, as long as each demanded pair stays reachable.
- This is weaker than distance preservation, but can be much sparser.



red path = shortest in G



blue path keeps reachability

Reachability: The Main Bound

Theorem

Every directed graph has a P-reachability preserver H with

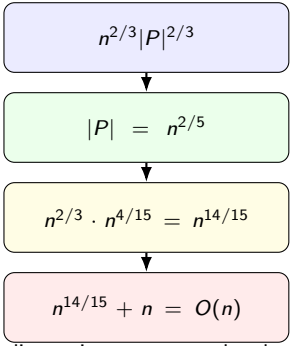
$$|E(H)| \leq O\left(\min\left\{n^{2/3}|P|^{2/3},\ \frac{n^2}{2^{\Theta(\log^* n)}},\ \frac{|P|^2}{2^{\Theta(\log^* |P|)}}\right\} + n\right).$$

Corollary

If $|P| = n^{2/5}$, then

$$n^{2/3}|P|^{2/3} = n^{2/3} \cdot n^{4/15} = n^{14/15},$$

so the theorem gives $|E(H)| = O(n)$. The other two terms in the minimum can only help.



linear-size preservers already appear when the demand set is only $n^{2/5}$ large

Reachability: Independence and Incidence

Independence Lemma

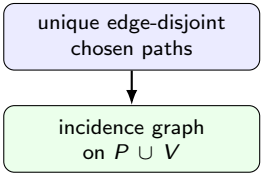
For upper-bound analysis, we may assume the chosen demand paths are unique and edge-disjoint.

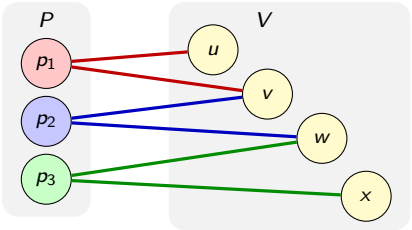
- This removes tie-breaking issues and repeated-edge overlap.
- What remains is a combinatorial question: how can different demand paths meet the same vertices?

Incidence Graph

Let $I = (P \cup V, E_I)$, where $(p, v) \in E_I$ exactly when the chosen path for p uses v.

Thus bounding $|E_I|$ also bounds $|E(H)|$.





edges record membership

Reachability: Forbidden Patterns

Forbidden Patterns

In the incidence graph, some alternating configurations cannot occur.

- C_4-type overlaps are already heavily restricted.
- The key reachability obstruction is an alternating C_6.

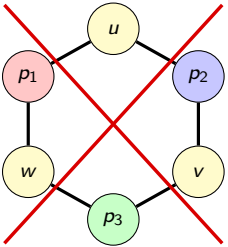
Lemma 1

The incidence graph avoids C_6.

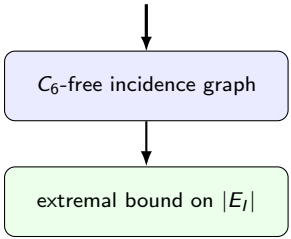
Lemma 2 (Extremal Bound)

$$\mathrm{ex}(n, |P|, C_6) \leq O\left(\min\left\{ n^{2/3}|P|^{2/3},\ \frac{n^2}{2^{\Theta(\log^* n)}},\ \frac{|P|^2}{2^{\Theta(\log^* |P|)}} \right\}\right).$$

Hence the theorem follows.



alternating C_6 forbidden



Why They Matter

Sparse graph structures are useful because they support:

- **collision filtering** in large interaction graphs,
- **real-time computation** by reducing per-frame work,
- **memory savings** through fewer maintained edges,
- **provable guarantees** on distances or cuts.

In both theory and practice, the objective is the same: preserve the right structure while discarding unnecessary edges.

Main Message

Takeaway

Graph sparsification is not one single object. Depending on the problem, we may want to preserve cuts, exact distances for selected pairs, or approximate distances for all pairs. The right sparse subgraph depends on what structure we need to keep.

References

Books

- D. B. West, *Introduction to Graph Theory*, 2nd ed., Prentice Hall, 2001.
- F. R. K. Chung, *Spectral Graph Theory*, CBMS Regional Conference Series in Mathematics 92, AMS, 1997.

Background Papers

- I. Althöfer, G. Das, D. Dobkin, D. Joseph, and J. Soares, "On sparse spanners of weighted graphs," *Discrete & Computational Geometry*, 9(1):81–100, 1993.
- A. A. Benczúr and D. R. Karger, "Approximating s-t minimum cuts in $\tilde{O}(n^2)$ time," in *Proc. STOC*, 1996, pp. 47–55.

Sparsification Papers

- D. A. Spielman and S.-H. Teng, "Spectral sparsification of graphs," *SIAM Journal on Computing*, 40(4):981–1025, 2011.
- J. Batson, D. A. Spielman, and N. Srivastava, "Twice-Ramanujan sparsifiers," *SIAM Journal on Computing*, 41(6):1704–1721, 2012.

Preserver Papers

- A. Abboud and G. Bodwin, "Reachability Preservers: New Extremal Bounds and Approximation Algorithms," *SIAM Journal on Computing*, 53(2):221–246, 2024.

Thank You

Thank you!

Questions and discussion welcome.