Incompressible Graph Metrics

Greg Bodwin

Stanford University

Includes joint work with Amir Abboud and Seth Pettie
Distance Sketching

This talk is about **sketching distances in (und. unw.) graphs**:

A **Distance Oracle** is a **small-space** data structure that preprocessed a graph $G$, then **approximately** answers queries of the form $\text{dist}(u, v)$. 

This generalizes many well-studied objects: spanners, emulators, time-bounded Distance Oracles, ...
Distance Sketching

This talk is about **sketching distances in (und. unw.) graphs**:

A **Distance Oracle** is a **small-space** data structure that preprocesses a graph $G$, then **approximately** answers queries of the form $\text{dist}(u, v)$.

This generalizes many well-studied objects: spanners, emulators, time-bounded Distance Oracles, ...
This talk is about **sketching distances in (und. unw.) graphs**:

A **Distance Oracle** is a **small-space** data structure that preprocesses a graph $G$, then **approximately** answers queries of the form $\text{dist}(u, v)$.

This generalizes many well-studied objects: spanners, emulators, time-bounded Distance Oracles, ...
Distance Sketching

This talk is about **sketching distances in (und. unw.) graphs**:

A **Distance Oracle** is a **small-space** data structure that preprocesses a graph $G$, then **approximately** answers queries of the form $\text{dist}(u, v)$.

This generalizes many well-studied objects: spanners, emulators, time-bounded Distance Oracles, ...
Oracle has **error function** $f$ if, whenever $\text{dist}(u, v) = D$, we have

$$D \leq \overline{\text{dist}}(u, v) \leq D + f(D)$$
Distance Sketching

Oracle has **error function** $f$ if, whenever $\text{dist}(u, v) = D$, we have

$$D \leq \text{dist}(u, v) \leq D + f(D)$$

**Goal:** Determine the optimal tradeoff between the space usage of the Distance Oracle and the error function $f$. 
Oracle has **error function** $f$ if, whenever $\text{dist}(u, v) = D$, we have

$$D \leq \overline{\text{dist}}(u, v) \leq D + f(D)$$

**Goal:** Determine the optimal tradeoff between the **space usage** of the Distance Oracle and the **error function** $f$. 
Distance Sketching

Oracle has error function $f$ if, whenever $\text{dist}(u, v) = D$, we have

$$D \leq \overline{\text{dist}}(u, v) \leq D + f(D)$$

Goal: Determine the optimal tradeoff between the space usage of the Distance Oracle and the error function $f$. 
Distance Sketching

Oracle has **error function** $f$ if, whenever $\text{dist}(u, v) = D$, we have

$$D \leq \overline{\text{dist}}(u, v) \leq D + f(D)$$

**Goal:** Determine the optimal tradeoff between the **space usage** of the Distance Oracle and the **error function** $f$. 
Multiplicative Error

First work focused on multiplicative error functions $f(D) = c \cdot D$. 

Althöfer et al: Distance Oracles on $\tilde{O}(n^{1 + 1/k})$ bits with error function $f(D) = (2^k - 2) D$. This is optimal at $f(1)$ (assuming Erdő Girth Conjecture).
Multiplicative Error

First work focused on multiplicative error functions $f(D) = c \cdot D$.

Althöfer et al: Distance Oracles on $\tilde{O}(n^{1+1/k})$ bits with error function $f(D) = (2k - 2) \cdot D$.

This is optimal at $f(1)$ (assuming Erdös Girth Conjecture).
First work focused on **multiplicative** error functions $f(D) = c \cdot D$.

---

**Althöfer et al.** Distance Oracles on $\tilde{O}(n^{1+1/k})$ bits with error function $f(D) = (2k - 2) \cdot D$.

This is optimal at $f(1)$ (assuming Erdös Girth Conjecture).
First work focused on **multiplicative error functions** $f(D) = c \cdot D$.

\[ \text{Distance } D \]

\[ \text{Error } f(D) \]

\[ n^{4/3}, n^{3/2} \]

\[ 1, 2, \ldots \]

**Multiplicative Error**

\[ \Leftrightarrow \]

Lines through the origin

**Althöfer et al:** Distance Oracles on $\tilde{O}(n^{1+1/k})$ bits with error function $f(D) = (2k - 2) \cdot D$.

This is optimal at $f(1)$ (assuming Erdös Girth Conjecture).
First work focused on multiplicative error functions $f(D) = c \cdot D$.

**Althöfer et al:** Distance Oracles on $\tilde{O}(n^{1+1/k})$ bits with error function $f(D) = (2k - 2) \cdot D$.

This is optimal at $f(1)$ (assuming Erdös Girth Conjecture).
Althöfer et al [Disc. Comp. Geom.]: Distance Oracles on $\tilde{O}(n^{1+1/k})$ bits with error function $f(D) = (2k - 2) \cdot D$ (for any positive integer $k$).
Althöfer et al [Disc. Comp. Geom.]: Distance Oracles on $\tilde{O}(n^{1+1/k})$ bits with error function $f(D) = (2k - 2) \cdot D$ (for any positive integer $k$).

**Takeaway:** More multiplicative error tolerance $\rightarrow$ Smaller space (down to nearly-$O(n)$ size for large $k$!)
Multiplicative Error is Unsatisfying

Lower bound for $f(1)$ is not a complete picture!
Multiplicative Error isn’t the right thing

Lower bound for $f(1)$ is not a complete picture!

- {Thorup and Zwick} and {Elkin and Peleg} showed that sublinear error is achievable (more on this shortly)
Multiplicative Error isn’t the right thing

Lower bound for $f(1)$ is not a complete picture!

$\frac{n^3}{\sqrt{2}}$ also $n^{3/2}$

- {Thorup and Zwick} and {Elkin and Peleg} showed that sublinear error is achievable (more on this shortly)
- So multiplicative error pays too much when $D > 1$. 
Additive Error

In particular: lower bound for $f(1)$ still allows for purely additive error!

$$f(D) = c$$

Additive Error ⇔ Horizontal Lines

If we can get nearly $O(n)$-size Oracles with purely additive error, that would be amazing!
Additive Error

In particular: lower bound for $f(1)$ still allows for **purely additive error**!

$$f(D) = c$$

<table>
<thead>
<tr>
<th>Distance $D$</th>
<th>Error $f(D)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>+2</td>
</tr>
<tr>
<td>2</td>
<td>$n^{3/2}$?</td>
</tr>
</tbody>
</table>

Additive Error $\Leftrightarrow$ Horizontal Lines

If we can get **nearly** $O(n)$-size Oracles with purely additive error, that would be amazing!
Additive Error

In particular: lower bound for \( f(1) \) still allows for purely additive error!

\[
f(D) = c
\]

Additive Error \( \iff \) Horizontal Lines

If we can get nearly \( O(n) \)-size Oracles with purely additive error, that would be amazing!
Additive Error

In particular: lower bound for $f(1)$ still allows for purely additive error!

$$f(D) = c$$

Additive Error $\iff$ Horizontal Lines

If we can get nearly $O(n)$-size Oracles with purely additive error, that would be amazing!
Additive Error

In particular: lower bound for $f(1)$ still allows for purely additive error!

$$f(D) = c$$

Additive Error $\iff$ Horizontal Lines

If we can get nearly $O(n)$-size Oracles with purely additive error, that would be amazing!
Additive Error

In particular: lower bound for $f(1)$ still allows for **purely additive error**!

$$f(D) = c$$

Additive Error $\iff$ Horizontal Lines

If we can get **nearly $O(n)$-size Oracles** with purely additive error, that would be amazing!

If $n^{4/3}$? **YES!** [DHZ00]

If $n^{3/2}$? **YES!** [ACIM99]
Additive Error

**Upper Bound:** Distance Oracles with $f(D) = 2$ on $\tilde{O}(n^{3/2})$ bits [Aingworth, Chekuri, Indyk, Motwani SODA ’99]

**Upper Bound:** Distance Oracles with $f(D) = 4$ on $\tilde{O}(n^{4/3})$ bits [Dor, Halperin, Zwick FOCS ’96]

Question: does the tradeoff continue? Can we get (e.g.) $\tilde{O}(n^{1+1/k})$ bits with $f(D) = c_k$ error? (i.e. does more additive error give sparser spanners?)
**Additive Error**

**Upper Bound:** Distance Oracles with $f(D) = 2$ on $\tilde{O}(n^{3/2})$ bits [Aingworth, Chekuri, Indyk, Motwani SODA ’99]

**Upper Bound:** Distance Oracles with $f(D) = 4$ on $\tilde{O}(n^{4/3})$ bits [Dor, Halperin, Zwick FOCS ’96]

**Question:** does the tradeoff continue? Can we get (e.g.) $\tilde{O}(n^{1+1/k})$ bits with $f(D) = c_k$ error?

(i.e. does more additive error give sparser spanners?)
Additive Error

In particular: lower bound for $f(1)$ still allows for purely additive error!

$$f(D) = D + c$$

Distance $D$

Error $f(D)$

Additive Error $\iff$ Horizontal Lines

$\frac{n^{4/3}}{\text{?}} \; \text{YES!} \; [\text{DHZ00}]$

$\frac{n^{3/2}}{\text{?}} \; \text{YES!} \; [\text{ACIM99}]$
Additive Error

In particular: lower bound for $f(1)$ still allows for **purely additive error**!

$$f(D) = D + c$$

Distance $D$

Error $f(D)$

Additive Error $\iff$ Horizontal Lines

$\cdot$ ? \times \text{NO!}$

$\cdot$ +4

$\cdot$ +2

$\cdot$ $n^{4/3}$? \textbf{YES!} [DHZ00]

$\cdot$ $n^{3/2}$? \textbf{YES!} [ACIM99]

1 2 ...
Additive Error Functions

**Upper Bound:** Distance Oracles with $f(D) = 2$ on $\tilde{O}(n^{3/2})$ bits [Aingworth, Chekuri, Indyk, Motwani SODA '99]

**Upper Bound:** Distance Oracles with $f(D) = 4$ on $\tilde{O}(n^{4/3})$ bits [Dor, Halperin, Zwick FOCS '96]

**Lower Bound:** No construction of Distance Oracles on $n^{4/3-\varepsilon}$ bits, $\varepsilon > 0$, with error $f(D) = n^{o(1)}$! [Abboud, B. STOC '16]
Additive Error Functions

**Upper Bound:** Distance Oracles with $f(D) = 2$ on $\tilde{O}(n^{3/2})$ bits [Aingworth, Chekuri, Indyk, Motwani SODA ’99]

**Upper Bound:** Distance Oracles with $f(D) = 4$ on $\tilde{O}(n^{4/3})$ bits [Dor, Halperin, Zwick FOCS ’96]

**Lower Bound:** No construction of Distance Oracles on $n^{4/3-\varepsilon}$ bits, $\varepsilon > 0$, with error $f(D) = n^{o(1)}$! [Abboud, B. STOC ’16]

**Mysteries:**
- What is the right error function, then?
- Why is $n^{4/3}$ special?
Thorup and Zwick [SODA ’06] introduced **sublinear error**:

\[ f(D) = D + O(D^{1 - 1/k}) \]  

(for integers \( k \geq 1 \), with space depending on \( k \)).

\[ TZ \text{ Sublinear Error} \leftrightarrow \text{polynomial functions} \]
Thorup and Zwick [SODA ’06] introduced sublinear error:
\[ f(D) = D + O(D^{1-1/k}) \] (for integers \( k \geq 1 \), with space depending on \( k \)).
Thorup and Zwick [SODA ’06] introduced sublinear error:
\[ f(D) = D + O(D^{1-1/k}) \] (for integers \( k \geq 1 \), with space depending on \( k \)).

\[ n^{8/7}(\sqrt{D}) \]
\[ n^{4/3}(4) \]

TZ Sublinear Error \iff polynomial functions
Thorup and Zwick [SODA ’06] introduced sublinear error: $f(D) = D + O(D^{1-1/k})$ (for integers $k \geq 1$, with space depending on $k$).
Thorup and Zwick [SODA ’06] introduced sublinear error:
\[ f(D) = D + O(D^{1-1/k}) \] (for integers \( k \geq 1 \), with space depending on \( k \)).
Thorup-Zwick Construction

- A series of constructions with error functions $O(1), D^{1/2}, D^{2/3}...$
Thorup-Zwick Construction

- A series of constructions with error functions $O(1), D^{1/2}, D^{2/3}...$
- **Weakness:** says nothing about what sparsity is possible with (e.g.) a $f(D) = D^{1/3}$ error budget.
Thorup-Zwick Construction

- A series of constructions with error functions $O(1), D^{1/2}, D^{2/3}...$
- **Weakness:** says nothing about what sparsity is possible with (e.g.) a $f(D) = D^{1/3}$ error budget.

**Theorem:** Distance Oracles on $\tilde{O}(n^{4/3})$ bits with error $f(D) = 4$.

(GAP)

**Theorem:** Distance Oracles on $\tilde{O}(n^{8/7})$ bits with error $f(D) = O(\sqrt{D})$.  

Greg Bodwin
Our Results

**Theorem:** Distance Oracles on $\tilde{O}(n^{4/3})$ bits with error $f(D) = 4$.

**Theorem:** [Abboud B. Pettie SODA ’17] Any Distance Oracle construction on $\tilde{O}(n^{4/3-\varepsilon})$ bits has worst-case error $f(D) = D + \Omega(\sqrt{D})$.

**Theorem:** Distance Oracles on $\tilde{O}(n^{8/7})$ bits with error $f(D) = O(\sqrt{D})$. 

There's a phase transition around $\sqrt{D}$ error!
Our Results

**Theorem:** Distance Oracles on $\tilde{O}(n^{4/3})$ bits with error $f(D) = 4$.

**Theorem:** [Abboud B. Pettie SODA ’17] Any Distance Oracle construction on $\tilde{O}(n^{4/3-\varepsilon})$ bits has worst-case error $f(D) = D + \Omega(\sqrt{D})$.

**Theorem:** Distance Oracles on $\tilde{O}(n^{8/7})$ bits with error $f(D) = O(\sqrt{D})$.

There’s a **phase transition** around $\sqrt{D}$ error!
Phase Transitions

Turns out there’s a **hierarchy** of **phase transitions** for Distance Oracles:

<table>
<thead>
<tr>
<th>Stretch Allowance</th>
<th>Sparsity Limit</th>
</tr>
</thead>
</table>

...
Phase Transitions

Turns out there’s a **hierarchy** of **phase transitions** for Distance Oracles:

<table>
<thead>
<tr>
<th>Stretch Allowance</th>
<th>Sparsity Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f(D) = D + 1$</td>
<td>$n^2$</td>
</tr>
<tr>
<td>$f(D) = D + 4$</td>
<td>$n^{4/3}$</td>
</tr>
</tbody>
</table>
Phase Transitions

Turns out there’s a **hierarchy** of **phase transitions** for Distance Oracles:

<table>
<thead>
<tr>
<th>Stretch Allowance</th>
<th>Sparsity Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f(D) = D + 1$</td>
<td>$n^2$</td>
</tr>
<tr>
<td>$f(D) = D + 4$</td>
<td>$n^{4/3}$</td>
</tr>
<tr>
<td>$f(D) = D + \sqrt{D}$</td>
<td>$n^{4/3}$</td>
</tr>
<tr>
<td>$f(D) = D + 100\sqrt{D}$</td>
<td>$n^{8/7}$</td>
</tr>
</tbody>
</table>
Phase Transitions

Turns out there’s a hierarchy of phase transitions for Distance Oracles:

<table>
<thead>
<tr>
<th>Stretch Allowance</th>
<th>Sparsity Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f(D) = D + 1$</td>
<td>$n^2$</td>
</tr>
<tr>
<td>$f(D) = D + 4$</td>
<td>$n^{4/3}$</td>
</tr>
<tr>
<td>$f(D) = D + \sqrt{D}$</td>
<td>$n^{4/3}$</td>
</tr>
<tr>
<td>$f(D) = D + 100\sqrt{D}$</td>
<td>$n^{8/7}$</td>
</tr>
<tr>
<td>$f(D) = D + D^{2/3}$</td>
<td>$n^{8/7}$</td>
</tr>
<tr>
<td>$D \rightarrow D + 100D^{2/3}$</td>
<td>$n^{16/15}$</td>
</tr>
</tbody>
</table>
Generalized Lower Bounds

**Theorem: [Thorup Zwick ’06]** For any integer $k$, there are Distance Oracles on $O(n^{1+\frac{1}{2^{k+1}-1}})$ bits,

with error $f(D) = D + O(D^{1-1/k})$

**Theorem: [Abboud B. Pettie ’17]** For any integer $k$, any algorithm that compresses graphs into $O(n^{1+\frac{1}{2^{k-1}-\varepsilon}})$ bits (for any $\varepsilon > 0$)

has worst-case error $f(D) = D + \Omega(D^{1-1/k})$

PUNCHLINE: T-Z type error functions are exactly the right thing! Other error functions (like $f(D) = D^{1/3}$) aren’t worth considering!
Generalized Lower Bounds

**Theorem: [Thorup Zwick ’06]** For any integer $k$, there are Distance Oracles on $O(n^{1+\frac{1}{2^k+1}-1})$ bits, with error $f(D) = D + O(D^{1-1/k})$

**Theorem: [Abboud B. Pettie ’17]** For any integer $k$, any algorithm that compresses graphs into $O(n^{1+\frac{1}{2^k-1}-\epsilon})$ bits (for any $\epsilon > 0$) has worst-case error $f(D) = D + \Omega(D^{1-1/k})$

**PUNCHLINE:**

T-Z type error functions are exactly the right thing! Other error functions (like $f(D) = D^{1/3}$) aren’t worth considering!
New Lower Bounds

\[ f(D) = \frac{n^4}{3^2 (D + 4)} - \varepsilon \]

\[ f(D) = \frac{n^8}{7^2 (D + \sqrt{D})} - \varepsilon \]

\[ f(D) = \frac{n^{16}}{15^2 (D + D^2/3)} \]

\[ \text{Error } f(D) \]

\[ \text{Distance } D \]
New Lower Bounds

\[ f(D) = \frac{n^4}{\sqrt{3}} \left( D + 4 \right) - \varepsilon \]

\[ n^{4/3 - \varepsilon} \text{ space} \]

Distance \( D \)
New Lower Bounds

\[ f(D) \leq n^{4/3 - \varepsilon} \leq n^{8/7} (D + \sqrt{D}) \]

\[ n^{4/3} (D + 4) \]
New Lower Bounds

$\text{Error } f(D) = n^{16/15}(D + D^{2/3})$

Distance $D$

$n^{8/7}(D + \sqrt{D})$

$n^{4/3}(D + 4)$

$n^{4/3 - \epsilon}_{\text{space}}$

$n^{8/7 - \epsilon}_{\text{space}}$
New Lower Bounds

$$f(D) \leq n^{16/15} (D + D^{2/3})$$

$$f(D) \leq n^{8/7} (D + \sqrt{D})$$

$$f(D) \leq n^{4/3 - \epsilon} \text{space}$$

$$f(D) \leq n^{4/3} (D + 4)$$

Distance $D$
The Construction.

First this:

**Lower Bound:** No construction of Distance Oracles on $n^{4/3-\varepsilon}$ bits, $\varepsilon > 0$, with error $f(D) = n^{o(1)}$. [Abboud, B. STOC '16]
Lemma: For all $\varepsilon > 0$, there is a $\delta > 0$ and a graph $G$ on $\Omega(n^{2-\varepsilon})$ edges that is the union of unique edge-disjoint shortest paths of length exactly $\lceil n^\delta \rceil$.

[Alon '01, Coppersmith and Elkin '06]
Lemma: For all $\varepsilon > 0$, there is a $\delta > 0$ and a graph $G$ on $\Omega(n^{2-\varepsilon})$ edges that is the union of unique edge-disjoint shortest paths of length exactly $\lceil n^\delta \rceil$.

[Alon '01, Coppersmith and Elkin '06]

When $\varepsilon = \delta = 0$, think of a biclique ($P := \{a, b, c\} \times \{x, y, z\}$):
Lemma: For all $\varepsilon > 0$, there is a $\delta > 0$ and a graph $G$ on $\Omega(n^{2-\varepsilon})$ edges that is the union of unique edge-disjoint shortest paths of length exactly $\lceil n^\delta \rceil$.

[Alon '01, Coppersmith and Elkin '06]

When $\varepsilon = \delta = 0$, think of a biclique ($P := \{a, b, c\} \times \{x, y, z\}$):

For longer paths, this is a simple rephrasing of some basic facts from Additive Combinatorics ("there exist very dense sum-free sets of integers").
First move: subdivide the edges

\[ s \quad t \]

Length: \( n^\delta \)

Subdivide each edge \( n^\delta \) times.

Good News: Delete one edge \( \rightarrow \) introduce + \( n^\delta \) error in the graph.

Bad News: Added lots of new nodes to the graph.
First move: subdivide the edges

\[ n^\delta \text{ length} \]
First move: subdivide the edges

\[ A + 1 \text{ detour} \]

\[ n^\delta \text{ length} \]
First move: subdivide the edges

Subdivide each edge $n^{\delta}$ times
First move: subdivide the edges

Subdivide each edge $n^\delta$ times
First move: subdivide the edges

Subdivide each edge $n^\delta$ times

Good News: Delete one edge $\rightarrow$ introduce $+n^\delta$ error in the graph
First move: subdivide the edges

Subdivide each edge $n^\delta$ times

**Good News:** Delete one edge $\rightarrow$ introduce $+n^\delta$ error in the graph

**Bad News:** Added lots of new nodes to the graph
Next move: clique replacement

Replace each original node $v$ with a clique on $\text{deg}(v)$ nodes. Connect every edge entering $v$ to a different clique node.
Next move: clique replacement

Replace each original node $v$ with a clique on $\text{deg}(v)$ nodes. Connect every edge entering $v$ to a different clique node.

An example path in $G$ now looks like this:
Next move: clique replacement

Replace each original node \( v \) with a clique on \( \text{deg}(v) \) nodes. Connect every edge entering \( v \) to a different clique node.

An example path in \( G \) now looks like this:
After the clique replacement step

Error Analysis

These graphs have a nice property:

**Lemma:** If you delete all of the clique edges used by a path, then you stretch that path distance by at least $+n^\delta$. 
After the clique replacement step

Error Analysis

These graphs have a nice property:

**Lemma:** If you delete all of the clique edges used by a path, then you stretch that path distance by at least $+n^\delta$. 

![Diagram](image)
After the clique replacement step

Error Analysis

These graphs have a nice property:

Lemma: If you delete all of the clique edges used by a path, then you stretch that path distance by at least \( +n^\delta \).
Imagine we have a switch for each pair in $P$, which we can turn on or off in any combination. Each combination defines a graph.

Switch $j$ on = keep all clique edges used by $p_j$
Switch $j$ off = remove all clique edges used by $p_j$
General Incompressibility

Imagine we have a switch for each pair in $P$, which we can turn on or off in any combination. Each combination defines a graph.

Switch $j$ on = keep all clique edges used by $p_j$
Switch $j$ off = remove all clique edges used by $p_j$

Note: Paths are clique-edge-disjoint, so these switches don’t interfere with each other!
If switch \((s, t)\) is on, then \(\text{dist}(s, t)\) is [something].
Incompressibility

If switch \((s, t)\) is off, then \(\text{dist}(s, t)\) is at least [something] \(+n^\delta\).
Theorem: There is no Distance Oracle construction on \( < |P| \) bits with 
\( f(D) = n^\delta \) error.
Theorem: There is no Distance Oracle construction on $< |P|$ bits with $f(D) = n^\delta$ error.

- Can turn all $|P|$ switches on or off in any combination; each one defines a different graph.
Theorem: There is no Distance Oracle construction on $< |P|$ bits with $f(D) = n^\delta$ error.

- Can turn all $|P|$ switches on or off in any combination; each one defines a different graph.
- If two graphs disagree on a switch, then they disagree on a distance by $\pm n^\delta$. 
Incompressibility Argument

**Theorem:** There is no Distance Oracle construction on $< |P|$ bits with $f(D) = n^\delta$ error.

- Can turn all $|P|$ switches on or off in any combination; each one defines a different graph.
- If two graphs disagree on a switch, then they disagree on a distance by $\pm n^\delta$.
- Therefore, no two of these graphs can collide on a representation, because if they did, you can’t properly decode that representation.
- Theorem follows from Pigeonhole Principle.
A low-error Distance Oracle needs $|P| \approx n^2$ bits.
A low-error Distance Oracle needs $|P| \approx n^2$ bits.

We subdivided each edge, so the number of nodes in the graph is $\approx$ Edges in original graph $\approx n^2$ nodes.
A low-error Distance Oracle needs $|P| \approx n^2$ bits.

We subdivided each edge, so the number of nodes in the graph is $\approx$ Edges in original graph $\approx n^2$ nodes.

**Uninteresting Theorem:** Any Distance Oracle with error $f(D) = N^\delta$ needs $\Omega(N)$ bits. 😞
Our original graphs weren’t quite what we wanted.
Our original graphs weren’t quite what we wanted. Remember the properties of the starting graph:

- Each pair in $P$ has a unique shortest path
- These shortest paths are edge disjoint
- These shortest paths all have length exactly $n^\delta$
- $|P| = n^{2-\varepsilon-\delta}$
- (And therefore, $|E(G)| = n^{2-\varepsilon}$)
Each pair in $P$ has a unique shortest path
- These shortest paths are edge disjoint
- These shortest paths all have length exactly $n^\delta$
- $|P| = n^{2-\varepsilon-\delta}$
  - This determines the space requirement of the D.O. – keep this high!
- (And therefore, $|E(G)| = n^{2-\varepsilon}$)
  - This determines the number of nodes in the lower bound graph
  – I wish this were smaller
Each pair in $P$ has a unique shortest path

These shortest paths are edge disjoint

These shortest paths all have length exactly $n^\delta$

$|P| = n^{2-\varepsilon-\delta}$

This determines the space requirement of the D.O. – keep this high!

(And therefore, $|E(G)| = n^{2-\varepsilon}$)

This determines the number of nodes in the lower bound graph – I wish this were smaller

What other property can we change in order to reduce $|E(G)|$ while keeping $|P|$?
Back to the start

- Each pair in $P$ has a unique shortest path
- These shortest paths are edge disjoint This one!
- These shortest paths all have length exactly $n^\delta$
- $|P| = n^{2-\varepsilon-\delta}$
  This determines the space requirement of the D.O. – keep this high!
- (And therefore, $|E(G)| = n^{2-\varepsilon}$)
  This determines the number of nodes in the lower bound graph
  – I wish this were smaller

What other property can we change in order to reduce $|E(G)|$ while keeping $|P|$?
Do we really need our paths to be edge disjoint?

Edge disjoint paths imply clique edge disjoint paths (black clique edges unused).
Do we really need our paths to be edge disjoint?

Edge disjoint paths imply clique edge disjoint paths (black clique edges unused).

Consecutive-edge disjoint paths also imply clique edge disjoint paths.
Do we really need our paths to be edge disjoint?

In our edge-extended, clique-replaced graphs, we need our paths to be clique edge disjoint. One clique edge corresponds to a pair of edges entering/leaving a node in the original graph. So we only need our original paths to be consecutive-edge disjoint, not edge disjoint.

Definition: A pair of paths $p_1, p_2$ in a graph $G$ is consecutive-edge disjoint if there is no length-2 path in $G$ that is a subpath of both $p_1$ and $p_2$. 
Do we really need our paths to be edge disjoint?

- In our edge-extended, clique-replaced graphs, we need our paths to be *clique edge disjoint*.
Do we really need our paths to be edge disjoint?

- In our edge-extended, clique-replaced graphs, we need our paths to be **clique edge disjoint**.
- One **clique edge** corresponds to a **pair** of edges entering/leaving a node in the original graph.
Do we really need our paths to be edge disjoint?

- In our edge-extended, clique-replaced graphs, we need our paths to be **clique edge disjoint**.
- One **clique edge** corresponds to a **pair** of edges entering/leaving a node in the original graph.
- So we only need our original paths to be **consecutive-edge disjoint**, not **edge disjoint**.

**Definition:** A pair of paths $p_1, p_2$ in a graph $G$ is **consecutive-edge disjoint** if there is no length-2 path in $G$ that is a subpath of both $p_1$ and $p_2$. 
Lemma: There is a graph $G$ on $\approx n^{3/2}$ edges that is the union of $\approx n^2$ unique consecutive-edge disjoint shortest paths.
Lemma: There is a graph $G$ on $\approx n^{3/2}$ edges that is the union of $\approx n^2$ unique consecutive-edge disjoint shortest paths.

Now repeat the construction from before ...
Lemma: There is a graph $G$ on $\approx n^{3/2}$ edges that is the union of $\approx n^2$ unique \textbf{consecutive-edge disjoint} shortest paths.

Now repeat the construction from before ...

- Any low-error D.O. must keep $|P| \approx N^2$ edges \textbf{(just like before)}
Lemma: There is a graph $G$ on $\approx n^{3/2}$ edges that is the union of $\approx n^2$ unique consecutive-edge disjoint shortest paths.

Now repeat the construction from before ...

- Any low-error D.O. must keep $|P| \approx N^2$ edges (just like before)
- Nodes in graph $\approx$ Edges in the original graph $\cdot N^\delta \approx N^{3/2}$ nodes (much better than before!).
Lemma: There is a graph $G$ on $\approx n^{3/2}$ edges that is the union of $\approx n^2$ unique consecutive-edge disjoint shortest paths.

Now repeat the construction from before ...

- Any low-error D.O. must keep $|P| \approx N^2$ edges (just like before).
- Nodes in graph $\approx$ Edges in the original graph $\cdot N^\delta \approx N^{3/2}$ nodes (much better than before!).
- Lower bound now follows from the calculation $(n^{3/2})^{4/3} = n^2$. 
Generalizing the Construction.

**Theorem: [Abboud B. Pettie ’16]** For any integer $k$, any algorithm that compresses graphs into $O(n^{1+\frac{1}{2k-1}-\epsilon})$ bits (for any $\epsilon > 0$)

has worst-case error $f(D) = D + \Omega(D^{1-1/k})$
Our Old Construction

Our construction has this general structure:

\[ \approx N^{2/3} \text{ “Input Ports”} \]

\[ \approx N^{2/3} \text{ “Output Ports”} \]

\[ N \text{ nodes; hard to sparsify without introducing error} \]
Our Old Construction

Old strategy: a (bi)clique replacement product for building $G$

$\approx N^{2/3}$ “Input Ports”

$\approx N^{2/3}$ “Output Ports”
Our Old Construction

Old strategy: a \textit{(bi)clique replacement product} for building $G$

\[ \approx N^{2/3} \text{ “Input Ports”} \]

\[ \approx N^{2/3} \text{ “Output Ports”} \]
Our Old Construction

Old strategy: a \((\text{bi})\text{clique replacement product}\) for building \(G\)

\[
\approx N^{2/3} \quad \text{“Input Ports”}
\]

\[
\approx N^{2/3} \quad \text{“Output Ports”}
\]
Our Old Construction

New strategy: a recursive construction of $G$

$\approx N^{2/3}$ “Input Ports”

Recursive Copy of $G$

$\approx N^{2/3}$ “Output Ports”

Recursion to depth $k \rightarrow k^{th}$ lower bound in the hierarchy
Open Problems
Spanners

An extremely well-studied way to build a Distance Oracle for $G$ is by finding a \textbf{sparse subgraph} $H \subseteq G$ whose distances approximately match $G$.

\begin{figure}
\centering
\begin{tikzpicture}
\node (a) at (0,0) [circle,fill,inner sep=1.5pt] {};
\node (b) at (1,0) [circle,fill,inner sep=1.5pt] {};
\node (c) at (2,1) [circle,fill,inner sep=1.5pt] {};
\node (d) at (3,-1) [circle,fill,inner sep=1.5pt] {};
\node (e) at (4,0) [circle,fill,inner sep=1.5pt] {};
\node (f) at (5,0) [circle,fill,inner sep=1.5pt] {};
\node (g) at (6,1) [circle,fill,inner sep=1.5pt] {};
\node (h) at (7,-1) [circle,fill,inner sep=1.5pt] {};
\node (i) at (8,0) [circle,fill,inner sep=1.5pt] {};
\node (j) at (9,0) [circle,fill,inner sep=1.5pt] {};
\node (k) at (10,1) [circle,fill,inner sep=1.5pt] {};
\node (l) at (11,-1) [circle,fill,inner sep=1.5pt] {};
\node (m) at (12,0) [circle,fill,inner sep=1.5pt] {};
\node (n) at (13,0) [circle,fill,inner sep=1.5pt] {};
\node (o) at (14,1) [circle,fill,inner sep=1.5pt] {};
\node (p) at (15,-1) [circle,fill,inner sep=1.5pt] {};
\node (q) at (16,0) [circle,fill,inner sep=1.5pt] {};
\node (r) at (17,0) [circle,fill,inner sep=1.5pt] {};
\node (s) at (18,1) [circle,fill,inner sep=1.5pt] {};
\node (t) at (19,-1) [circle,fill,inner sep=1.5pt] {};
\node (u) at (20,0) [circle,fill,inner sep=1.5pt] {};
\node (v) at (21,0) [circle,fill,inner sep=1.5pt] {};
\node (w) at (22,1) [circle,fill,inner sep=1.5pt] {};
\node (x) at (23,-1) [circle,fill,inner sep=1.5pt] {};
\node (y) at (24,0) [circle,fill,inner sep=1.5pt] {};
\node (z) at (25,0) [circle,fill,inner sep=1.5pt] {};
\draw (a) -- (b);
\draw (b) -- (c);
\draw (c) -- (d);
\draw (d) -- (e);
\draw (e) -- (f);
\draw (f) -- (g);
\draw (g) -- (h);
\draw (h) -- (i);
\draw (i) -- (j);
\draw (j) -- (k);
\draw (k) -- (l);
\draw (l) -- (m);
\draw (m) -- (n);
\draw (n) -- (o);
\draw (o) -- (p);
\draw (p) -- (q);
\draw (q) -- (r);
\draw (r) -- (s);
\draw (s) -- (t);
\draw (t) -- (u);
\draw (u) -- (v);
\draw (v) -- (w);
\draw (w) -- (x);
\draw (x) -- (y);
\draw (y) -- (z);
\end{tikzpicture}
\end{figure}

(Distances approximately match)
Spanners

An extremely well-studied way to build a Distance Oracle for $G$ is by finding a sparse subgraph $H \subseteq G$ whose distances approximately match $G$.

(Distances approximately match)

Open Question: Can the optimal stretch function be obtained using spanners?
Spanners

An extremely well-studied way to build a Distance Oracle for $G$ is by finding a sparse subgraph $H \subseteq G$ whose distances approximately match $G$.

(Distances approximately match)

Open Question: Can the optimal stretch function be obtained using spanners?

- Currently: polynomial gaps in space usage between spanners and distance oracles with the same error function.
We close the problem … but only when $D = n^{o(1)}$. 

\[
\text{Error } f(D) = n^{4/3}(D + 4) - n^{8/7}(D + \sqrt{D}) - n^{4/3-\varepsilon} \text{space}
\]
We close the problem ... but only when $D = n^{o(1)}$.

Thorup-Zwick upper bounds AND our lower bounds fall off for large distances.
Error for Long Distances

We close the problem ... but only when $D = n^{o(1)}$.

Thorup-Zwick upper bounds AND our lower bounds fall off for large distances.
Error for Long Distances

We close the problem ... but only when \( D = n^{o(1)} \).

Thorup-Zwick upper bounds AND our lower bounds fall off for large distances.
Summary

Punchline #1: If you want to compress graph distances while tolerating only $+c$ error, you can't go below $n^{4/3}$ space.

Punchline #2: Below the $n^{4/3}$ threshold, distance compression has a series of similar discrete phase transitions (at $n^{8/7}$, $n^{16/15}$, ...). The tradeoff between error and space is not smooth!

Open #1: Are subgraphs (spanners) optimal compression schemes?

Open #2: Compression bounds still unknown in the regime of long distances.

Thanks!

Greg Bodwin
Punchline #1: If you want to compress graph distances while tolerating only $+c$ error, you can’t go below $n^{4/3}$ space.

Punchline #2: Below the $n^{4/3}$ threshold, distance compression has a series of similar discrete phase transitions (at $n^{8/7}$, $n^{16/15}$, ...). The tradeoff between error and space is not smooth!

Open #1: Are subgraphs (spanners) optimal compression schemes?

Open #2: Compression bounds still unknown in the regime of long distances.
Punchline #1: If you want to compress graph distances while tolerating only $+c$ error, you can’t go below $n^{4/3}$ space.

Punchline #2: Below the $n^{4/3}$ threshold, distance compression has a series of similar discrete phase transitions (at $n^{8/7}, n^{16/15}, \ldots$). The tradeoff between error and space is not smooth!
Punchline #1: If you want to compress graph distances while tolerating only $+c$ error, you can’t go below $n^{4/3}$ space.

Punchline #2: Below the $n^{4/3}$ threshold, distance compression has a series of similar discrete phase transitions (at $n^{8/7}, n^{16/15}, \ldots$).

- The tradeoff between error and space is not smooth!

Open #1: Are subgraphs (spanners) optimal compression schemes?
Punchline #1: If you want to compress graph distances while tolerating only $+c$ error, you can’t go below $n^{4/3}$ space.

Punchline #2: Below the $n^{4/3}$ threshold, distance compression has a series of similar discrete phase transitions (at $n^{8/7}$, $n^{16/15}$, ...).
- The tradeoff between error and space is not smooth!

Open #1: Are subgraphs (spanners) optimal compression schemes?

Open #2: Compression bounds still unknown in the regime of long distances.
Punchline #1: If you want to compress graph distances while tolerating only $+c$ error, you can’t go below $n^{4/3}$ space.

Punchline #2: Below the $n^{4/3}$ threshold, distance compression has a series of similar discrete phase transitions (at $n^{8/7}, n^{16/15}, \ldots$).

- The tradeoff between error and space is not smooth!

Open #1: Are subgraphs (spanners) optimal compression schemes?

Open #2: Compression bounds still unknown in the regime of long distances.

Thanks!