# Approximate Nearest Neighbor (ANN): The Core Idea

## The Problem

```
Given: Database of n vectors
       D = {d_1, d_2, ..., d_n}

Query: Find k vectors closest to q

Exact solution requires:
  - Compute distance to ALL n vectors
  - Sort and return top-k
  - Time: O(n) per query
```

*n = 1 billion? That is 1 billion distance calculations per query!*

## The Mathematics

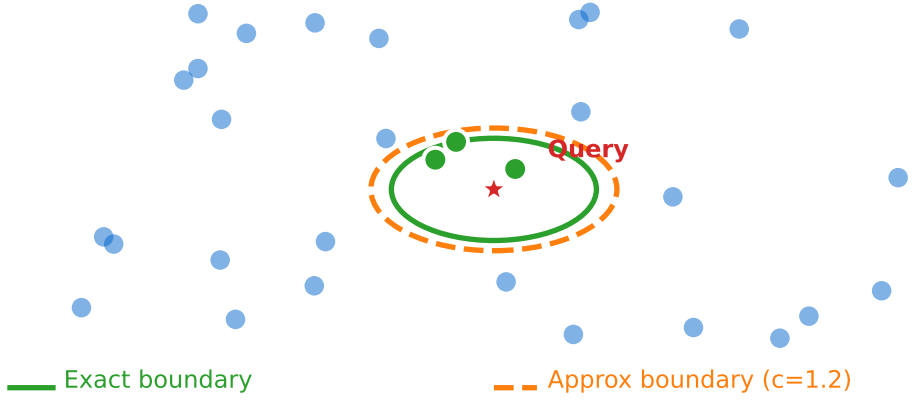
```
Exact k-NN:
  N_k(q) = argmin |S|=k  max  ||q - d||
                        d in S

c-Approximate k-NN:
  For all d in ANN_k(q):
    ||q - d|| <= c * ||q - d*||

  where d* is the true k-th neighbor
  and c >= 1 is the approximation factor
```

*c = 1.05 means we accept neighbors at most 5% farther than optimal*

## Visual Intuition




## The Trade-off

| Method | Time | Recall | Use Case |
|---|---|---|---|
| Exact (brute) | O(n) | 100% | Small datasets |
| IVF | O(sqrt(n)) | ~95% | Medium scale |
| HNSW | O(log n) | ~99% | Production |
| LSH | O(1)* | ~90% | Massive scale |

**Key: Accept 1-5% accuracy loss for 100-1000x speedup**

** LSH: O(1) query but O(n) space for hash tables*