The deluge of data has made the use of compressed data structures indispensable. These structures build on two sources of compressibility: statistical properties and repetitiveness of the data. But there is a new promising kind of regularity to be studied: approximate linearity.

The predecessor search problem
Given \( n \) sorted keys implement
\[
\text{pred}(x) = \text{"largest key } \leq x\"
\]
Why do we care? Range queries and joins in DBs, conjunctive queries in search engines, IP routing, ...

The PGM-index
Fixed model
- error \( \varepsilon \)
  - Control the size of the search range
Optimal piecewise linear \( \varepsilon \)-approximation
- Fast to construct, captures non-linearities
Recursive design
- Adapt to the memory hierarchy

Features
- Optimal time and space complexity guarantees
- Never worse than traditional indexes such as B-trees
- Resistant to adversarial inputs and queries
- Supports multidimensional data and queries
- Auto-tunable to the desired memory usage or query time

Rank/Select dictionaries
Store an integer set \( S \) in compressed form, support
\[
\text{rank}(x) = \text{"# elements } \leq x \text{ in } S"
\]
\[
\text{select}(i) = \text{"ith smallest element in } S"
\]
Why do we care? Building block of compressed data structures for texts, genomes, graphs, ...

The LA-vector
Orchestrate piecewise linear \( \varepsilon \)-approximations, corrections and indexing. Faster \text{select} \ and competitive \text{rank} \ wrt well-engineered solutions