An Introduction to Vector Symbolic Architectures and Hyperdimensional Computing

VSA Tutorial

Peer Neubert Kenny Schlegel Stefan Schubert

firstname.lastname@etit.tu-chemnitz.de

https://www.tu-chemnitz.de/etit/proaut/vsa_ecai20

ECAI DIGITAL - 2020

TECHNISCHE UNIVERSITÄT CHEMNITZ

Self-Test

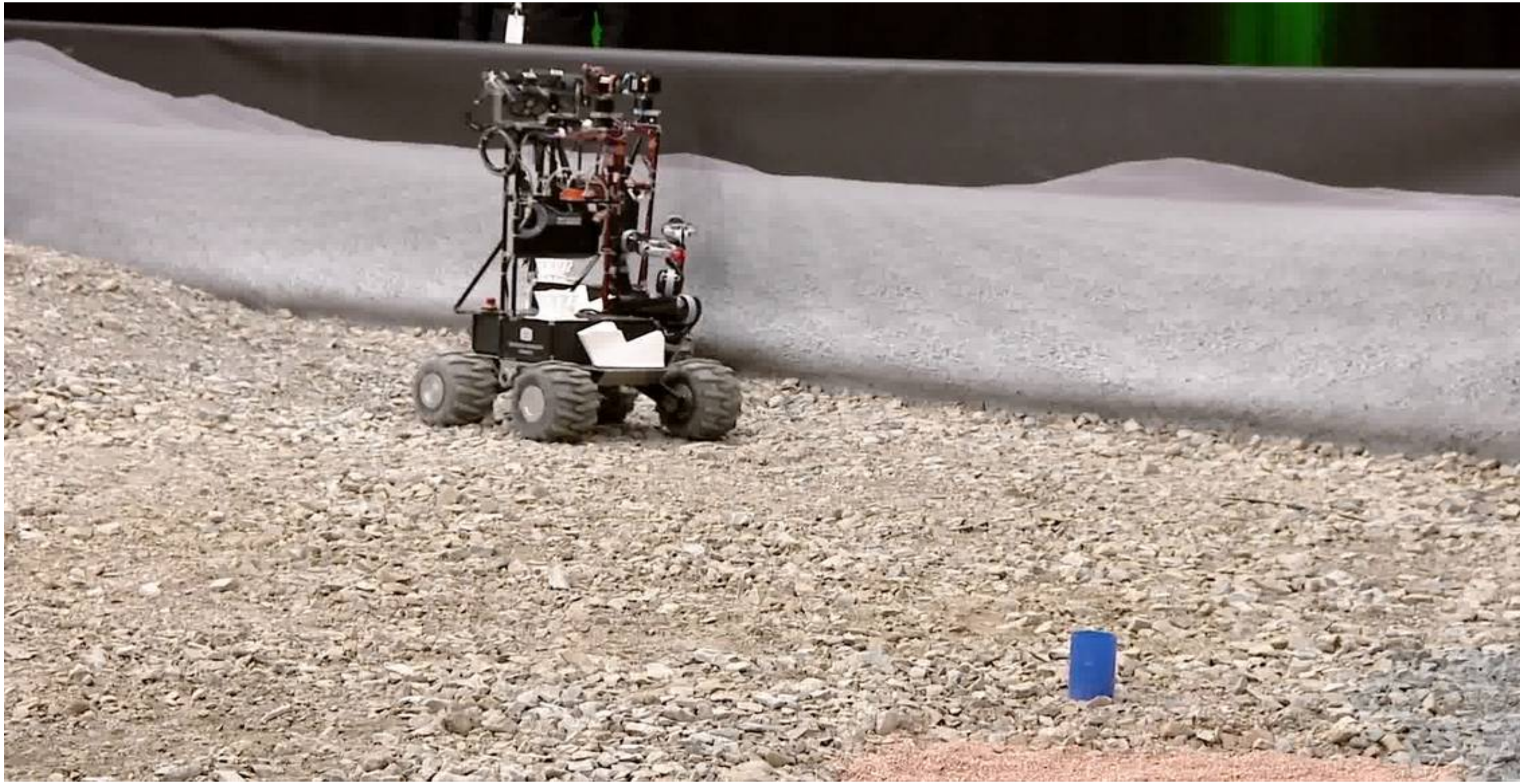
1) What is a typical size of VSA vector space?
2) In your own words: What are the four presented properties high-dimensional vector spaces?

The Trivial
The Bad
The Surprising
The Good

https://www.tu-chemnitz.de/etit/proaut/vsa_ecai20

3) Is the result of **bundling** A and B similar to A?
4) Is the result of **binding** A and B similar to A?
5) How many lines of code are required to implement a VSA (rough estimate)?
6) With the taxonomy of binding operators in mind, what is the requirement for the presented VSA approach to the “The Dollar of Mexico” example?
7) The place recognition demo uses sequences of CNN descriptors of camera images. Why is the binding operator important for the presented solution?

What we are doing



Topic: (Symbolic) Computation with large vectors

2D

$$\begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

3D

$$\begin{pmatrix} -3 \\ 2.3 \\ -0.1 \end{pmatrix} \longrightarrow$$

thousands of dimensions

$$\begin{pmatrix} 1.0 \\ 3.9 \\ -0.5 \\ \cdot \\ \cdot \\ \cdot \\ 2.9 \\ -6.0 \\ 9.8 \end{pmatrix}$$

Topic: (Symbolic) Computation with large vectors

Roughly synonyms:

- Vector Symbolic Architectures
- High dimensional Computing
- Hyperdimensional Computing
- Hypervectors
- Computing with large random vectors
- ...

2D

$$\begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

3D

$$\begin{pmatrix} -3 \\ 2.3 \\ -0.1 \end{pmatrix} \longrightarrow$$

thousands of dimensions

$$\begin{pmatrix} 1.0 \\ 3.9 \\ -0.5 \\ . \\ . \\ . \\ 2.9 \\ -6.0 \\ 9.8 \end{pmatrix}$$

Topic: (Symbolic) Computation with large vectors

Vector
Symbolic
Architecture

=

Vector space with thousands of dimensions

$$\begin{pmatrix} 1.0 \\ 3.9 \\ -0.5 \\ . \\ . \\ . \\ 2.9 \\ -6.0 \\ 9.8 \end{pmatrix}$$

+

Set of carefully designed operators

- Bundling()
- Binding()
- ...

Topic: (Symbolic) Computation with large vectors

Vector Symbolic Architecture = Vector space with thousands of dimensions + Set of carefully designed operators

$$\begin{pmatrix} 1.0 \\ 3.9 \\ -0.5 \\ . \\ . \\ . \\ 2.9 \\ -6.0 \\ 9.8 \end{pmatrix}$$

- Bundling()
- Binding()
- ...

Important principles

1) Every entity is an element of the same high-dimensional vector space
2) In the vectors, information is distributed across dimensions
3) Computations (algorithms) are implemented by vector operations
4) Relations of vectors (e.g. entities and computation results) are evaluated based on vector similarity

References

General introduction

Pentti Kanerva. 2009. *Hyperdimensional Computing: An Introduction to Computing in Distributed Representation with High-Dimensional Random Vectors*. Cognitive Computation 1, 2 (2009), 139–159. https://doi.org/10.1007/s12559-009-9009-8

Basic introduction, mathematical properties, and application to robotics

Neubert, P., Schubert, S., Protzel, P. 2019. *An Introduction to Hyperdimensional Computing for Robotics*. KI - Künstliche Intelligenz. https://doi.org/10.1007/s13218-019-00623-z

Experimental comparison of available VSAs

Schlegel, K., Neubert, P., Protzel, P. (2020) *A comparison of Vector Symbolic Architectures*. CoRR, abs/2001.11797

Community website

https://www.hd-computing.com

Outline

1) Introduction to VSA
   - High dimensional vector spaces and where they are used
   - Mathematical properties of high dimensional vector spaces
   - Vector Symbolic Architectures or “How to do symbolic computations using vectors spaces”

2) Available VSA implementations

3) Where do the vectors come from?

4) Demo application

5) Discussion

Teaser application 1: “What is the Dollar of Mexico?”

Credits: Pentti Kanerva

Teaser application 1: “What is the Dollar of Mexico?”

Credits: Pentti Kanerva

Given are 2 records:

United States of America

Name:	USA
Capital City:	Washington DC
Currency:	Dollar

Mexico

Name:	Mexico
Capital City:	Mexico City
Currency:	Peso

Teaser application 1: “What is the Dollar of Mexico?”

Credits: Pentti Kanerva

Given are 2 records:

United States of America

Name:	USA
Capital City:	Washington DC
Currency:	Dollar

Mexico

Name:	Mexico
Capital City:	Mexico City
Currency:	Peso

Question: What is the Dollar of Mexico?

Teaser application 1: “What is the Dollar of Mexico?”

Credits: Pentti Kanerva

Given are 2 records:

United States of America

Name:	USA
Capital City:	Washington DC
Currency:	Dollar

Mexico

Name:	Mexico
Capital City:	Mexico City
Currency:	Peso

Question: What is the Dollar of Mexico?

Hyperdimensional computing approach:

1. Assign a **random** high-dimensional vector to each entity
   ”Name” is a random vector NAM
   ”USA” is a random vector USA
   ”Capital city” is a random vector CAP
   …

Teaser application 1: “What is the Dollar of Mexico?”

Credits: Pentti Kanerva

Given are 2 records:

United States of America

Name:	USA
Capital City:	Washington DC
Currency:	Dollar

Mexico

Name:	Mexico
Capital City:	Mexico City
Currency:	Peso

Question: What is the Dollar of Mexico?

Hyperdimensional computing approach:

1. Assign a **random** high-dimensional vector to each entity
   ”Name” is a random vector NAM
   ”USA” is a random vector USA
   ”Capital city” is a random vector CAP
   …

2. Calculate a **single** high-dimensional vector that contains all information
   F = (NAM*USA+CAP*WDC+CUR*DOL)*(NAM*MEX+CAP*MCX+CUR*PES)

Teaser application 1: “What is the Dollar of Mexico?”

Credits: Pentti Kanerva

Given are 2 records:

United States of America

Name:	USA
Capital City:	Washington DC
Currency:	Dollar

Mexico

Name:	Mexico
Capital City:	Mexico City
Currency:	Peso

Question: What is the Dollar of Mexico?

Hyperdimensional computing approach:

1. Assign a **random** high-dimensional vector to each entity
   ”Name” is a random vector NAM
   ”USA” is a random vector USA
   ”Capital city” is a random vector CAP
   …

2. Calculate a **single** high-dimensional vector that contains all information
   F = (NAM*USA+CAP*WDC+CUR*DOL)*(NAM*MEX+CAP*MCX+CUR*PES)

3. **Calculate** the query answer: F*DOL ~ PES

Teaser application 2: Place recognition in changing environments

Problem: Visual place recognition

Image credits: M. Milford and G. F. Wyeth. Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2012.

Teaser application 2: Place recognition in changing environments

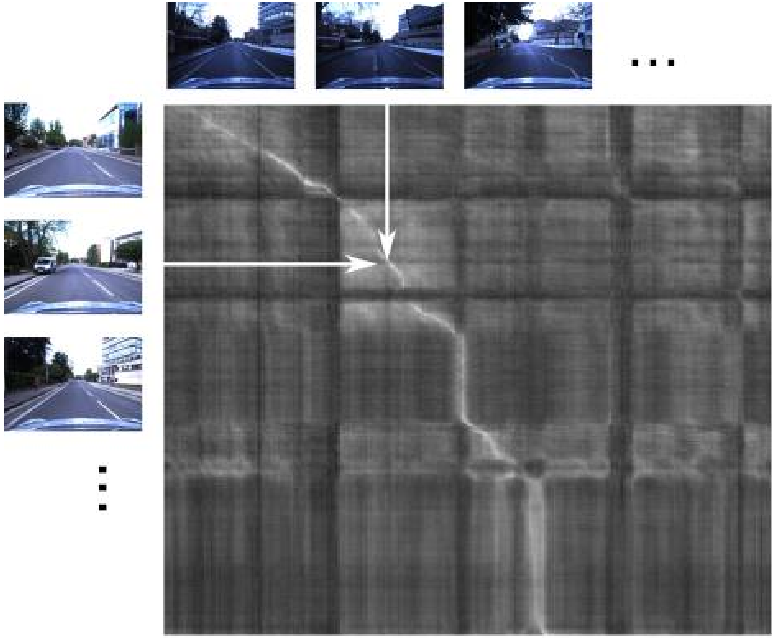
Deep Neural Network

$$\begin{pmatrix} 1.0 \\ 3.9 \\ -0.5 \\ . \\ . \\ . \\ 2.9 \\ -6.0 \\ 9.8 \end{pmatrix}$$

Teaser application 2: Place recognition in changing environments

Deep Neural Network

$$\begin{pmatrix} 1.0 \\ 3.9 \\ -0.5 \\ . \\ . \\ . \\ 2.9 \\ -6.0 \\ 9.8 \end{pmatrix}$$



Teaser application 2: Place recognition in changing environments

Problem: Visual place recognition in changing environments

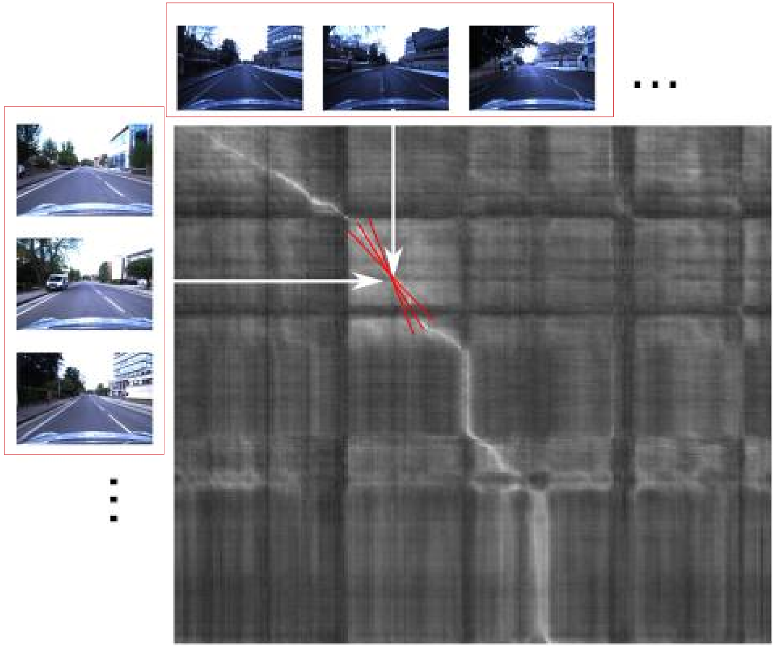
Image credits: M. Milford and G. F. Wyeth. Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2012.

Teaser application 2: Place recognition in changing environments

Deep Neural Network

$$\begin{pmatrix} 1.0 \\ 3.9 \\ -0.5 \\ \cdot \\ \cdot \\ \cdot \\ 2.9 \\ -6.0 \\ 9.8 \end{pmatrix}$$

Using Context:
e.g., **SeqSLAM** postprocessing



Teaser application 2: Place recognition in changing environments

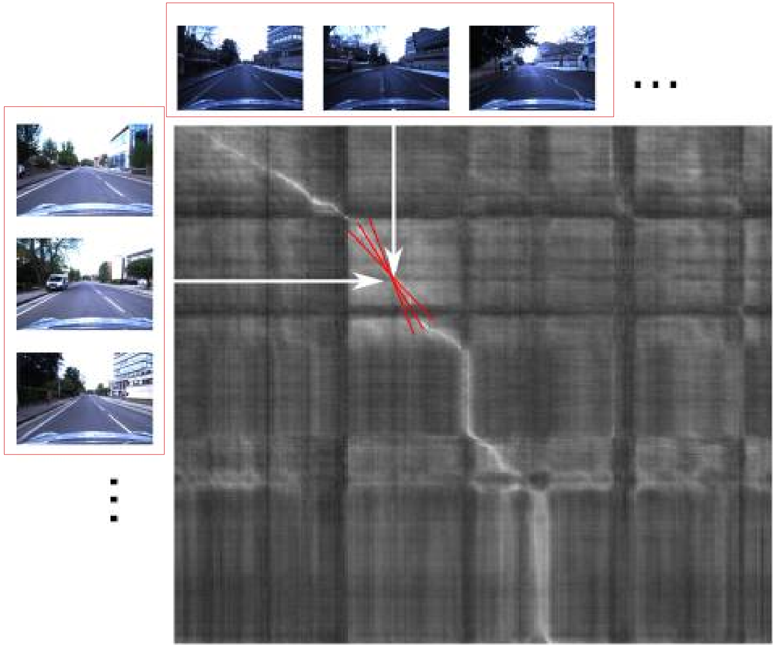
Deep Neural Network

$$\begin{pmatrix} 1.0 \\ 3.9 \\ -0.5 \\ \cdot \\ \cdot \\ \cdot \\ 2.9 \\ -6.0 \\ 9.8 \end{pmatrix}$$

Some VSA magic

Using Context:

~~e.g., **SeqSLAM** postprocessing~~



Where are high-dimensional vectors used?

- Feature vectors, e.g., in computer vision or information retrieval
- (Intermediate) representations in deep ANN
- Vector models for natural language processing, e.g., Latent Semantic Analysis
- Memory and storage models, e.g., Pentti Kanerva’s Sparse Distributed Memory or Deepmind’s long-short term memory
- Computational brain models, e.g. Jeff Hawkins’ HTM or Chris Eliasmith’s SPAUN
- Quantum cognition approaches
- ...

Outline

1) Introduction to VSA
   - High dimensional vector spaces and where they are used
   - **Mathematical properties of high dimensional vector spaces**
   - Vector Symbolic Architectures or “How to do symbolic computations using vectors spaces”
2) Available VSA implementations
3) Where do the vectors come from?
4) Demo application
5) Discussion

Four properties of high-dimensional vector spaces

“The good, the bad, and the ugly”

The Trivial
The Bad
The Surprising
The Good

Properties 1/4: High-dimensional vector spaces have huge capacity

The Trivial
The Bad
The Surprising
The Good

Properties 1/4: High-dimensional vector spaces have huge capacity

The Trivial
The Bad
The Surprising
The Good

- Capacity grows exponentially
- Here: “high-dimensional” means thousands of dimensions

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \dots \\ x_{4000} \end{pmatrix}$$

Properties 1/4: High-dimensional vector spaces have huge capacity

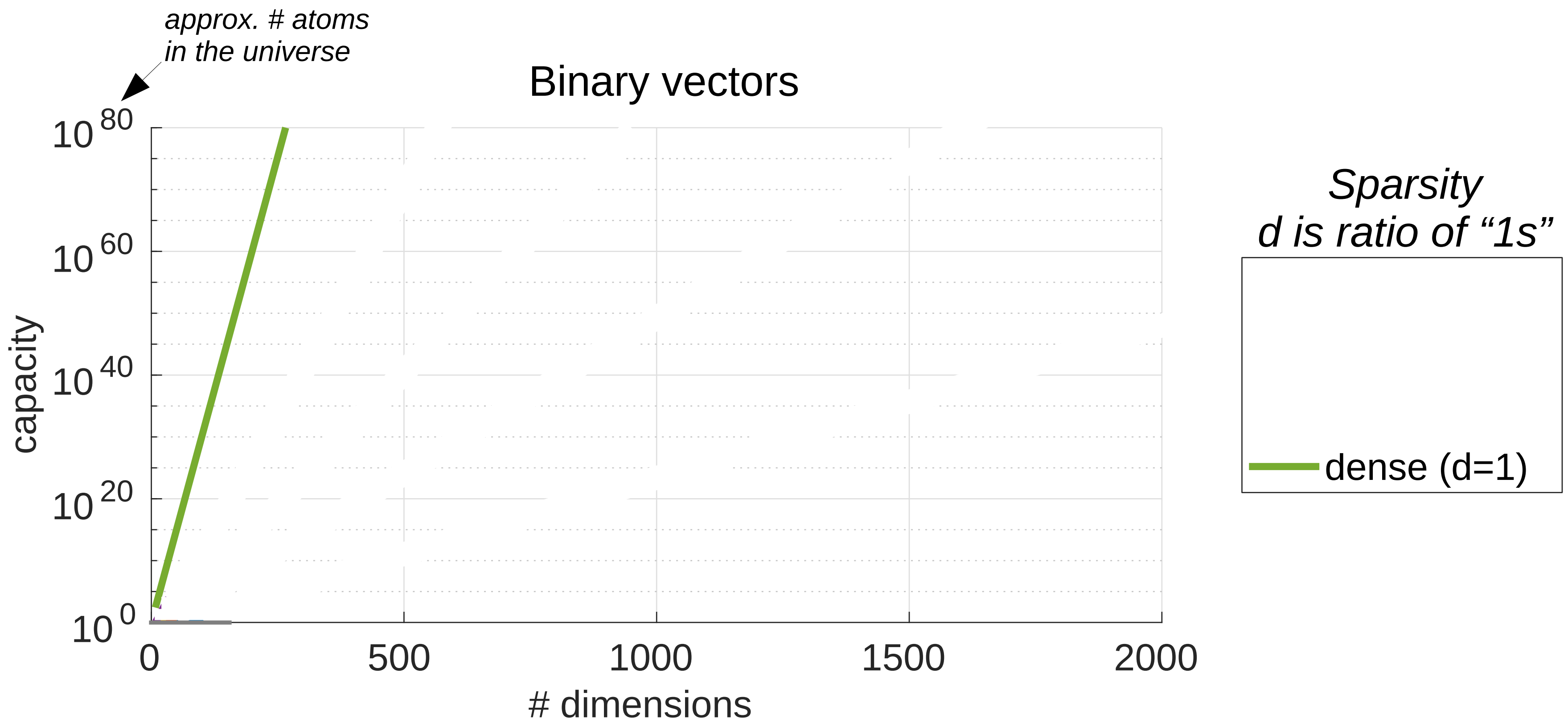
The Trivial
The Bad
The Surprising
The Good

- Capacity grows exponentially
- Here: “high-dimensional” means thousands of dimensions

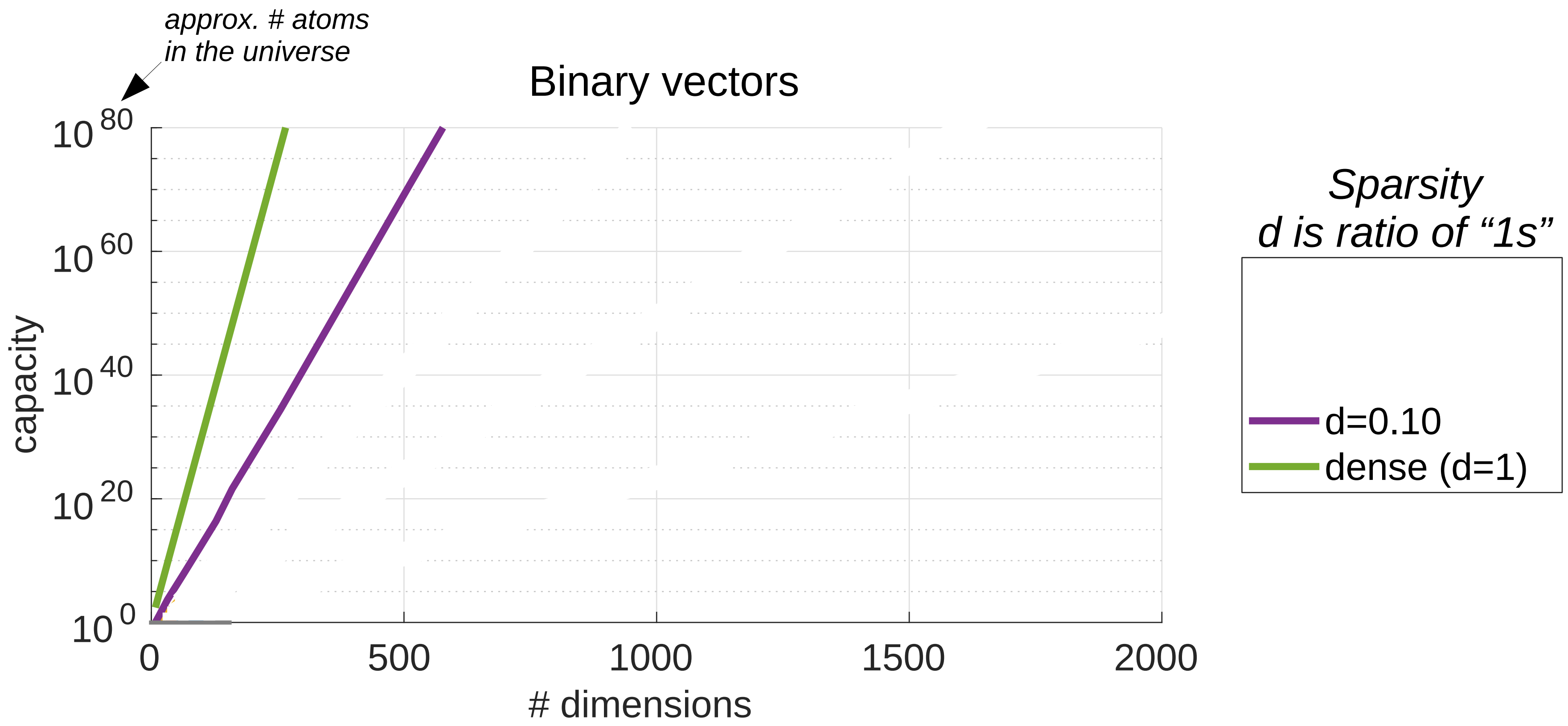
$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \dots \\ x_{4000} \end{pmatrix}$$

- This property also holds for other vector spaces than $\mathbb{R}^n$
  - Binary, e.g. $\{0, 1\}^n$, $\{-1, 1\}^n$
  - Ternary, e.g. $\{-1, 0, 1\}^n$
  - Real, e.g. $[-1, 1]^n$
  - Sparse or Dense

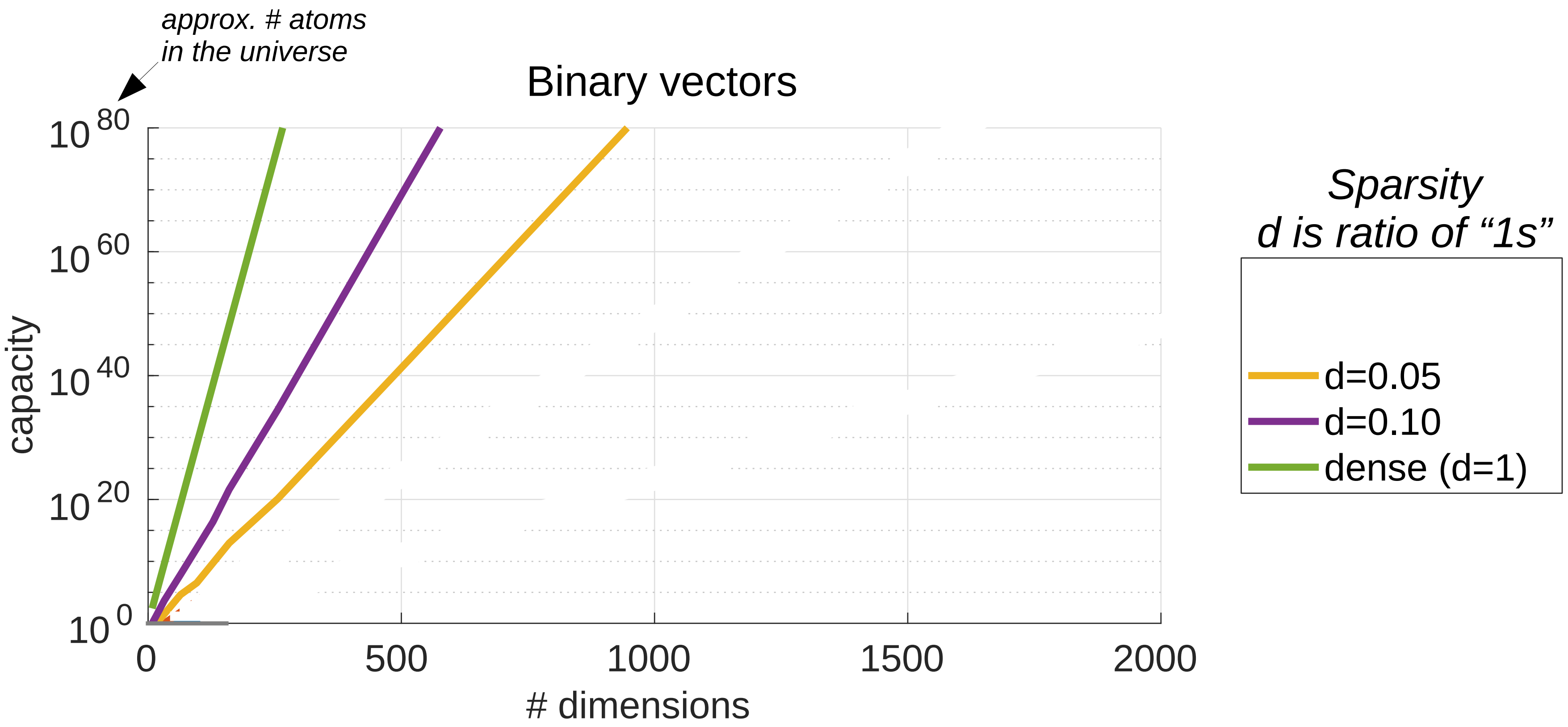
Properties 1/4: High capacity



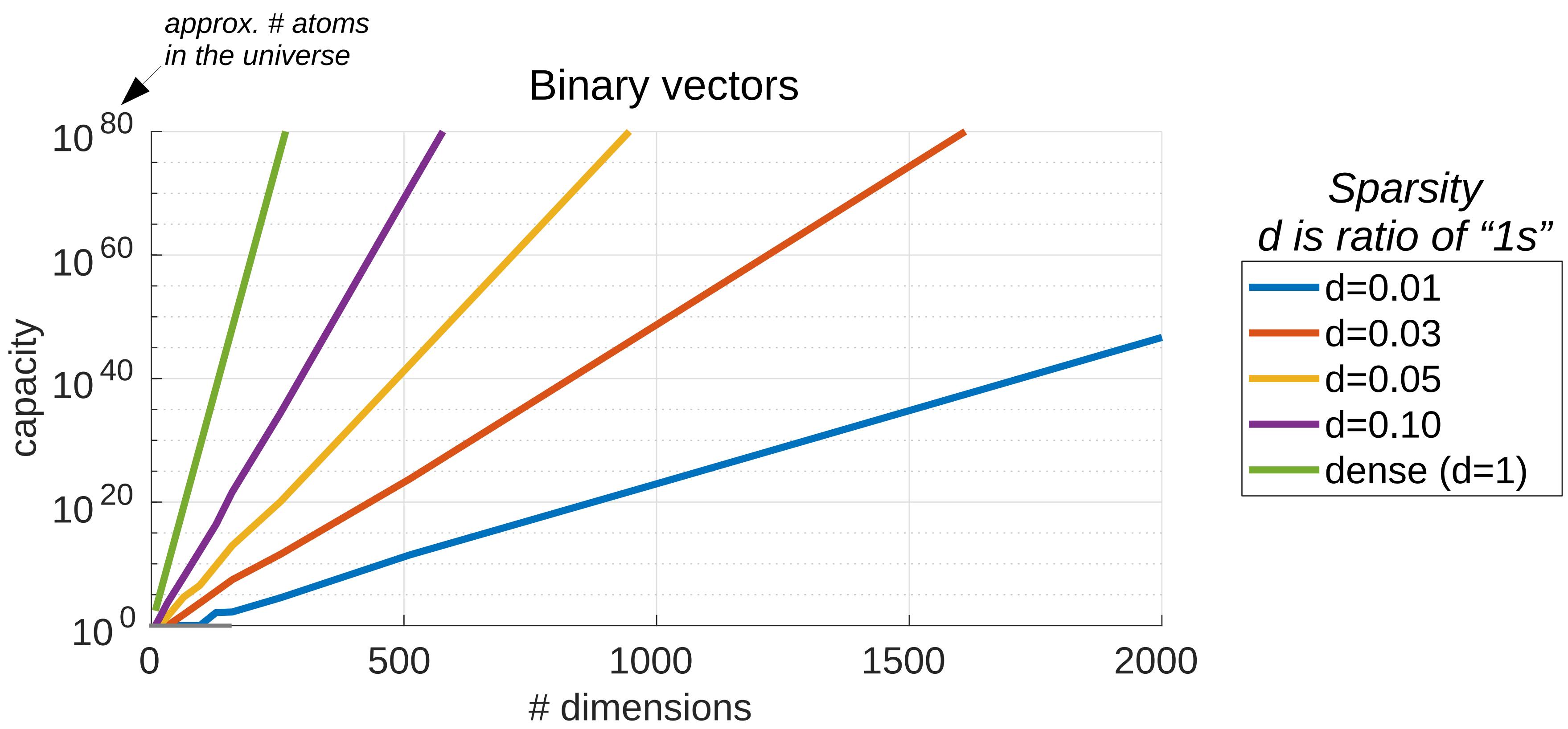
Properties 1/4: High capacity



Properties 1/4: High capacity



Properties 1/4: High capacity



Properties 2/4: Nearest neighbor becomes unstable or meaningless

The Trivial
The Bad
The Surprising
The Good

Properties 2/4: Nearest neighbor becomes unstable or meaningless

The Trivial
The Bad
The Surprising
The Good

Downside of so much space:

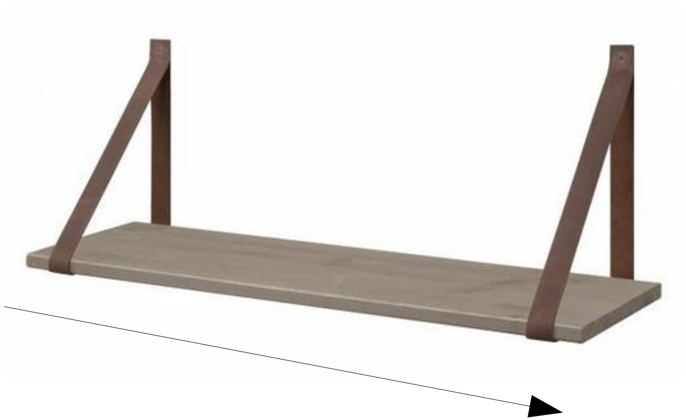
Bellman, 1961: “**Curse of dimensionality**”

- “Algorithms that work in low dimensional space fail in higher dimensional spaces”
- We require exponential amounts of samples to represent space with statistical significance (e.g., Hastie et al. 2009)

Bellman, R. E. (1961) Adaptive Control Processes: A Guided Tour. MIT Press, Cambridge

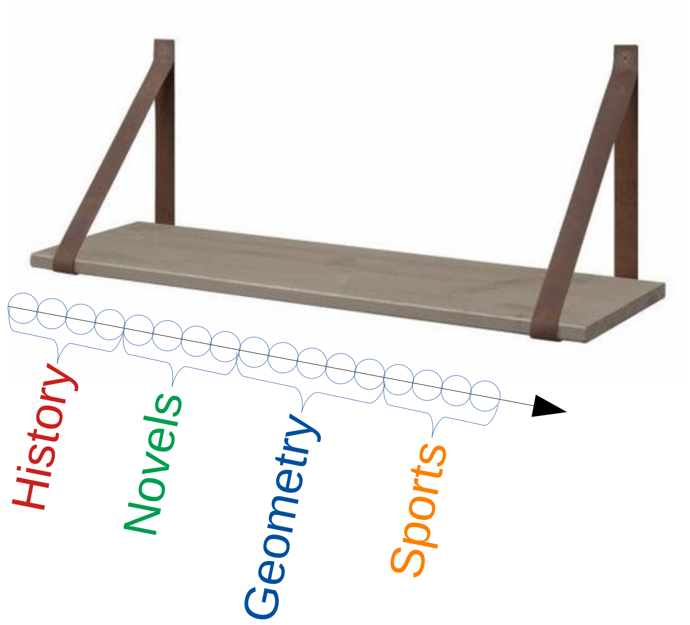
Hastie, Tibshirani and Friedman (2009). The Elements of Statistical Learning (2nd edition)Springer-Verlag

Example: Sorted library



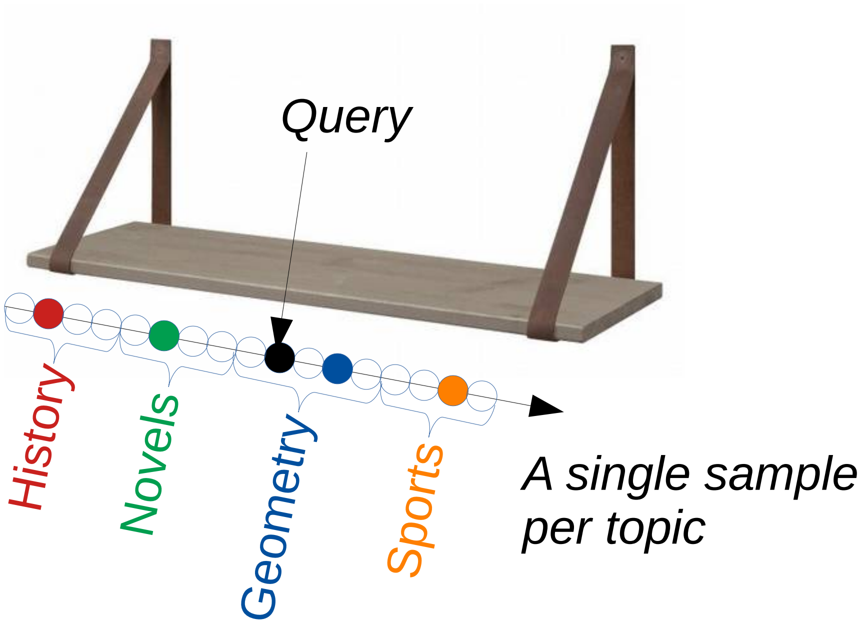
The Trivial
The Bad
The Surprising
The Good

Example: Sorted library



- Library contains books about 4 topics

Example: Sorted library

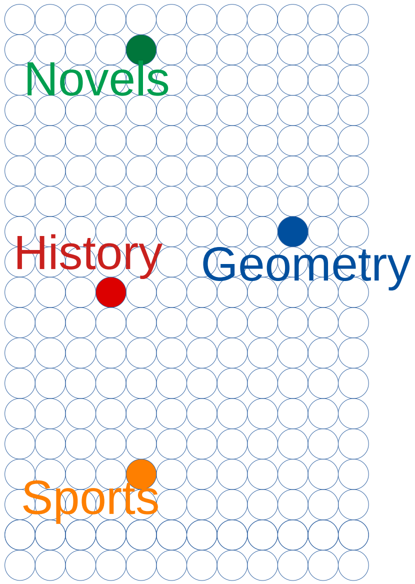


- Library contains books about 4 topics
- We can’t infer the topic from the pose directly, only by nearby samples.

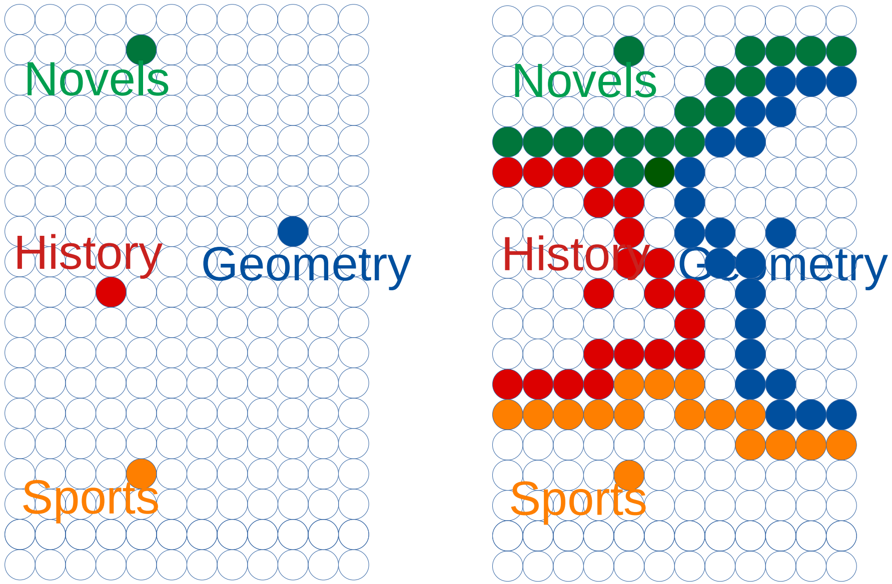
Example: Sorted library

The Trivial
The Bad
The Surprising
The Good

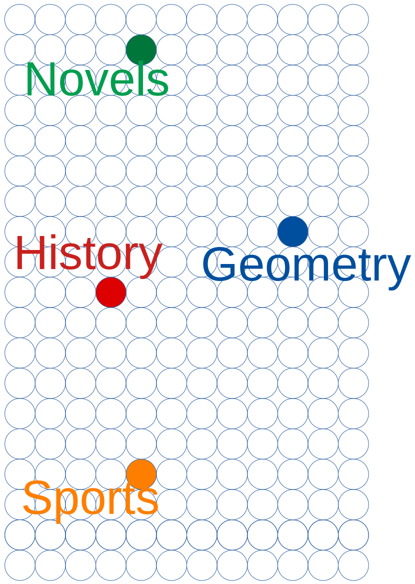
Example: Sorted library

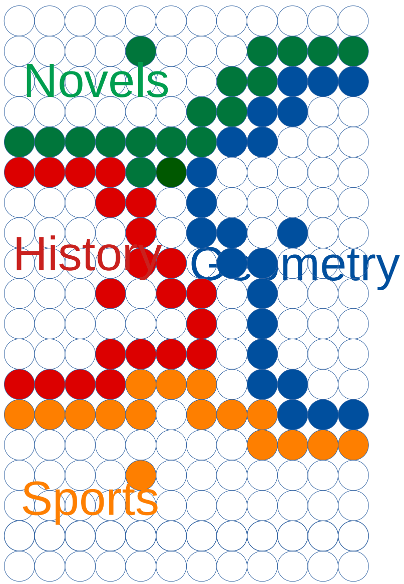


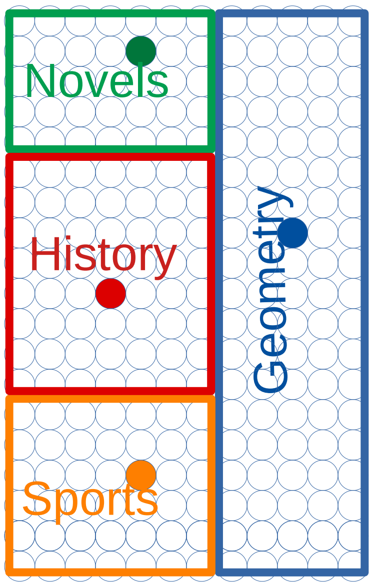
Example: Sorted library



Example: Sorted library







The more dimensions, the more samples are required to represent the shape of the clusters.

Exponential growth!

Properties 2/4: Nearest neighbor becomes unstable or meaningless

- Beyer K, Goldstein J, Ramakrishnan R, Shaft U (1999) **When Is nearest neighbor meaningful?** In: Database theory—ICDT’99. Springer, Berlin, Heidelberg, pp 217–235

The Trivial
The Bad
The Surprising
The Good

Properties 2/4: Nearest neighbor becomes unstable or meaningless

- Beyer K, Goldstein J, Ramakrishnan R, Shaft U (1999) **When Is nearest neighbor meaningful?** In: Database theory—ICDT'99. Springer, Berlin, Heidelberg, pp 217–235

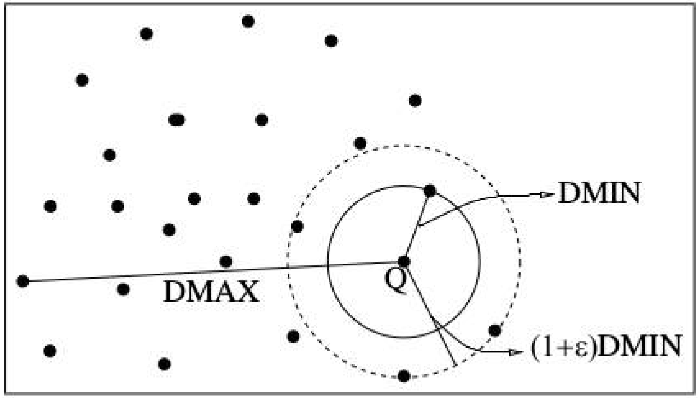


Fig. 4. Illustration of query region and enlarged region. (DMIN is the distance to the nearest neighbor, and DMAX to the farthest data point.)

Properties 2/4: Nearest neighbor becomes unstable or meaningless

- Beyer K, Goldstein J, Ramakrishnan R, Shaft U (1999) **When Is nearest neighbor meaningful?** In: Database theory—ICDT'99. Springer, Berlin, Heidelberg, pp 217–235

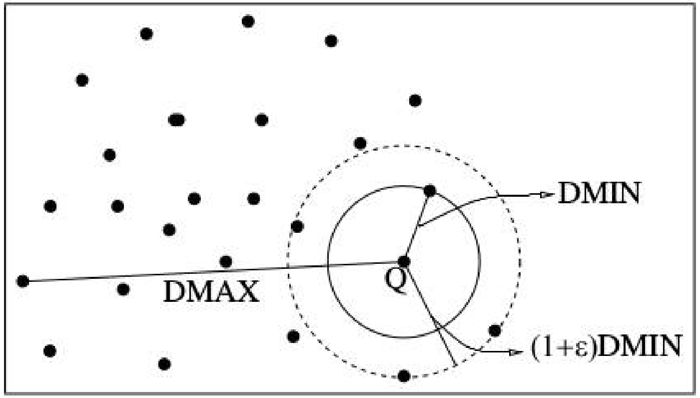


Fig. 4. Illustration of query region and enlarged region. (DMIN is the distance to the nearest neighbor, and DMAX to the farthest data point.)

"under a broad set of conditions (much broader than independent and identically distributed dimensions)"

⇒

Then for every $\varepsilon > 0$

$$\lim_{m \to \infty} P\left[DMAX_m \leq (1+\varepsilon) DMIN_m\right] = 1$$

Increasing #dimensions

Properties 2/4: Nearest neighbor becomes unstable or meaningless

- Beyer K, Goldstein J, Ramakrishnan R, Shaft U (1999) **When Is nearest neighbor meaningful?** In: Database theory—ICDT'99. Springer, Berlin, Heidelberg, pp 217–235

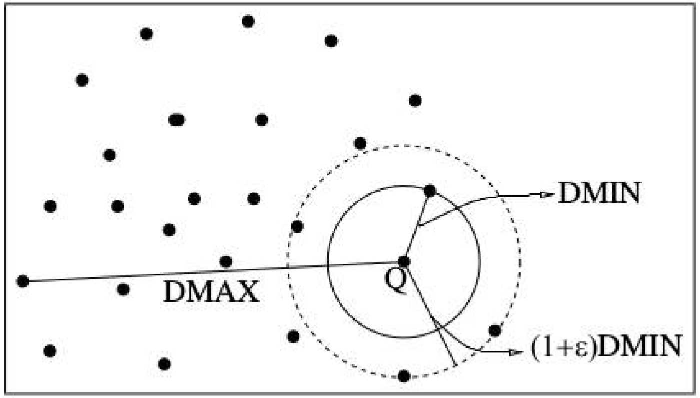


Fig. 4. Illustration of query region and enlarged region. (DMIN is the distance to the nearest neighbor, and DMAX to the farthest data point.)

"under a broad set of conditions (much broader than independent and identically distributed dimensions)"

➡

Then for every $\varepsilon > 0$

$$\lim_{m \to \infty} P\left[DMAX_m \leq (1+\varepsilon) DMIN_m\right] = 1$$

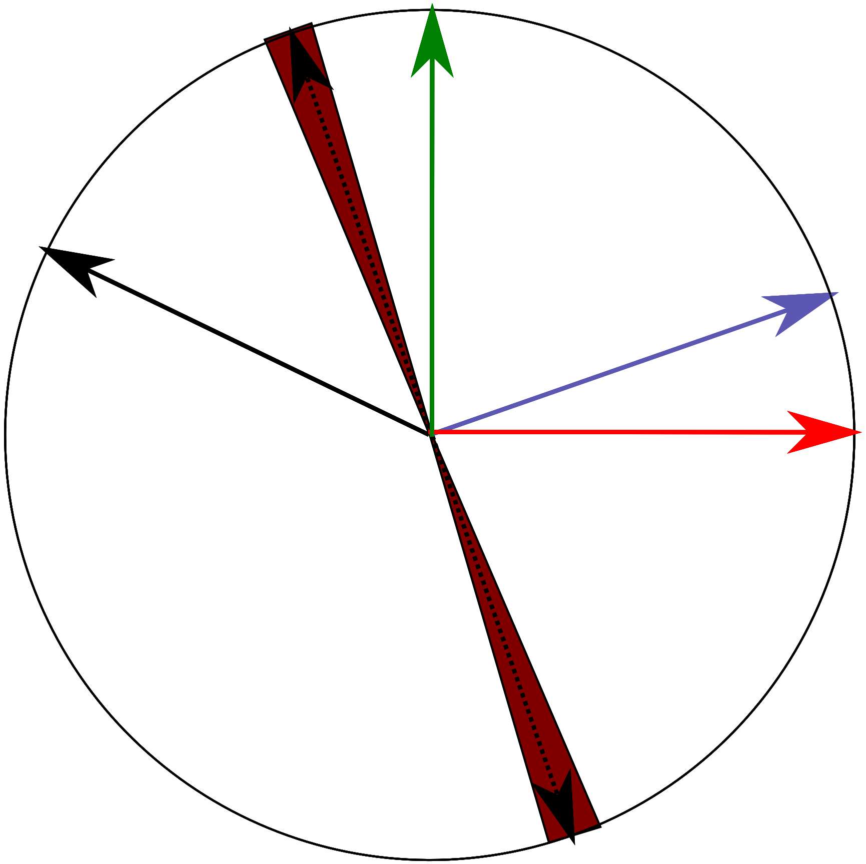
Increasing #dimensions

- Aggarwal CC, Hinneburg A, Keim DA (2001) On the surprising behavior of **distance metrics** in high dimensional space. In: Database theory—ICDT 2001. Springer, Berlin Heidelberg, pp 420–434

Properties 3/4: Time to gamble!

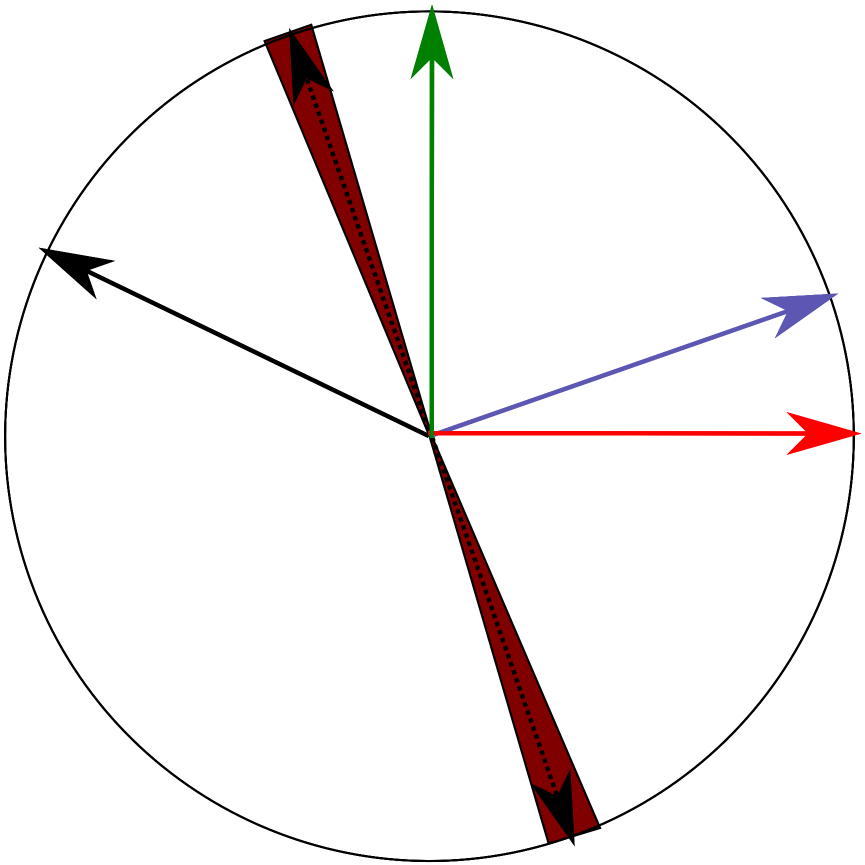
The Trivial
The Bad
The Surprising
The Good

Properties 3/4: Experiment



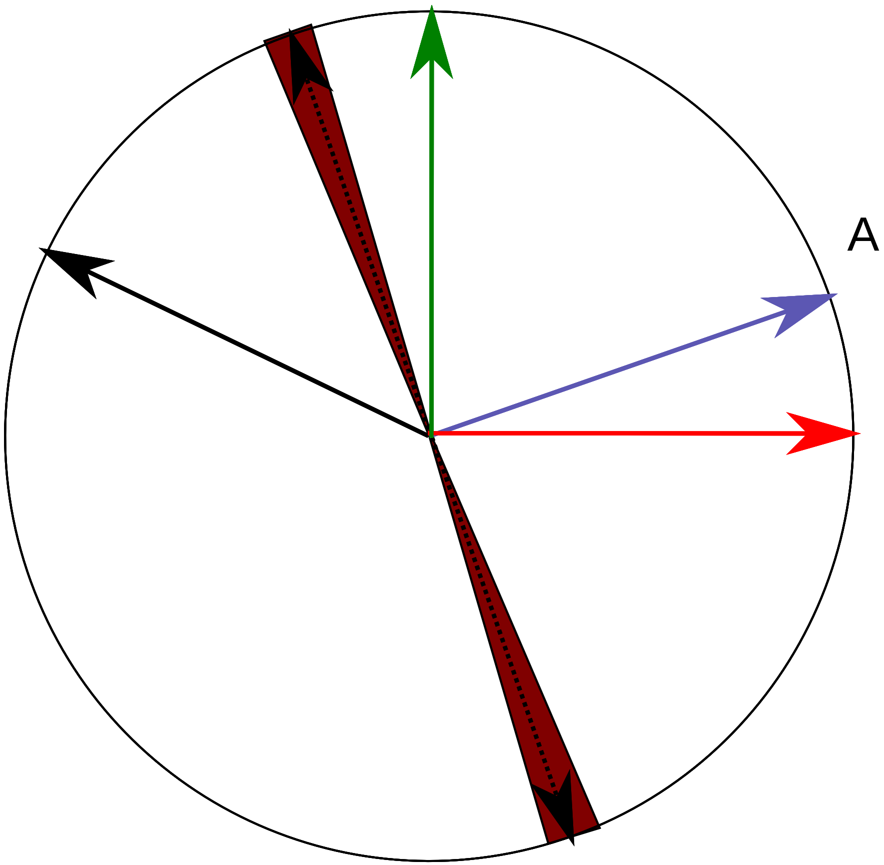
- Random vectors:
  - uniformly distributed angles
  - obtained by sampling each dimension iid. ~N(0,1)

Properties 3/4: Experiment



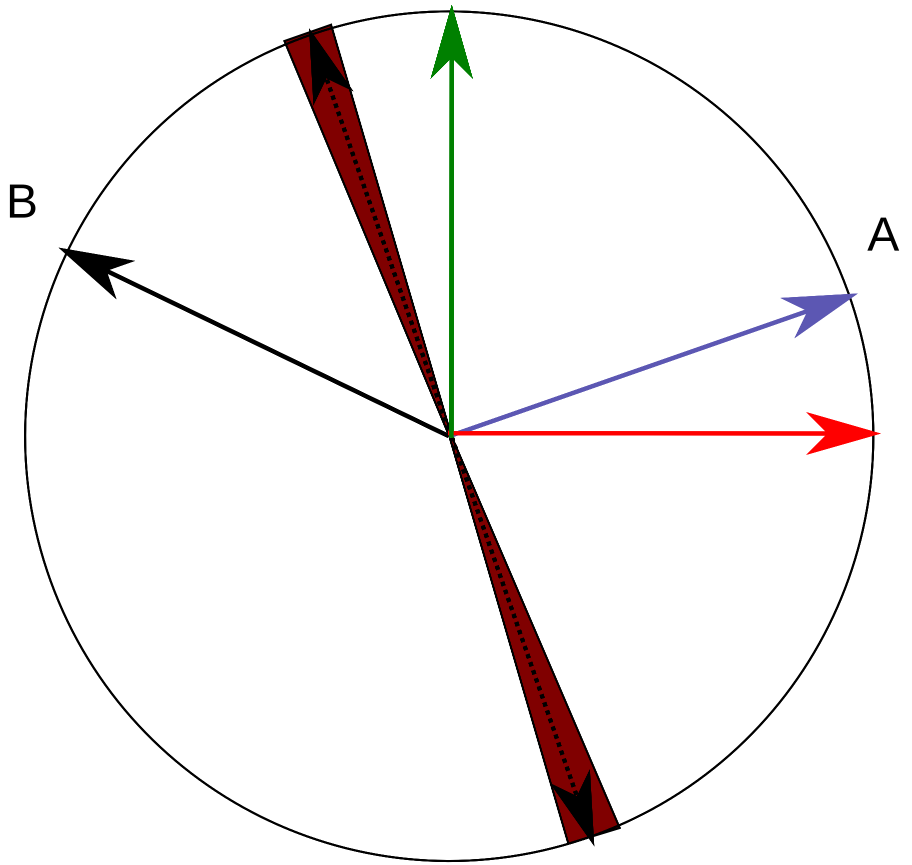
- Random vectors:
  - uniformly distributed angles
  - obtained by sampling each dimension iid. ~N(0,1)

Properties 3/4: Experiment



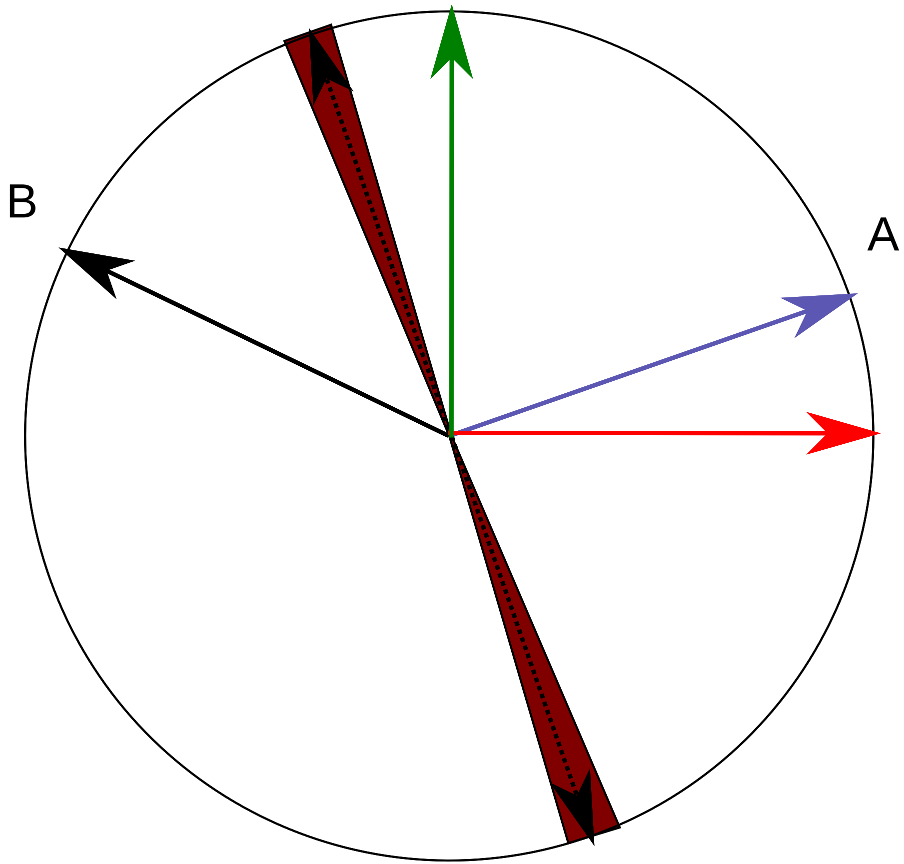
- Random vectors:
  - uniformly distributed angles
  - obtained by sampling each dimension iid. ~N(0,1)
- I bet:
  - Given a random vector A, we can independently sample a second random vector B and it will be almost orthogonal (+/- 5°) ...

Properties 3/4: Experiment



- Random vectors:
  - uniformly distributed angles
  - obtained by sampling each dimension iid. ~N(0,1)
- I bet:
  - Given a random vector A, we can independently sample a second random vector B and it will be almost orthogonal (+/- 5°) ...

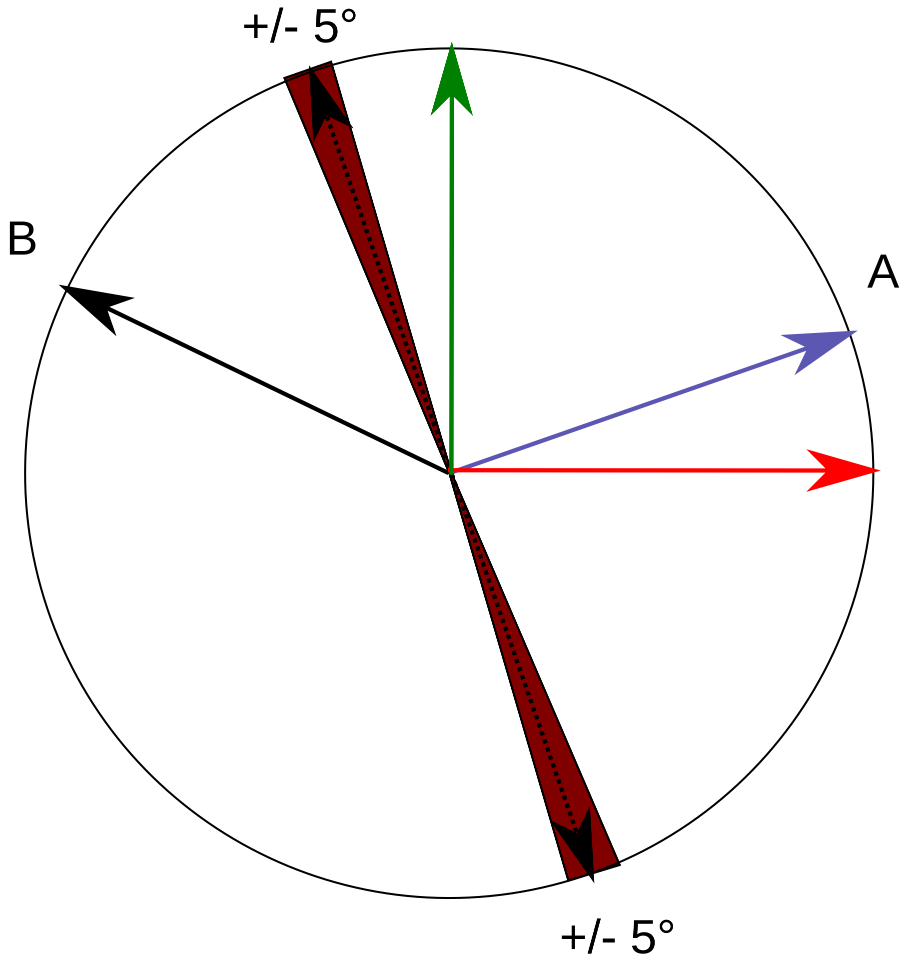
Properties 3/4: Experiment



- Random vectors:
  - uniformly distributed angles
  - obtained by sampling each dimension iid. ~N(0,1)
- I bet:
  - Given a random vector A, we can independently sample a second random vector B and it will be almost orthogonal (+/- 5°) ...

Properties 3/4: Experiment

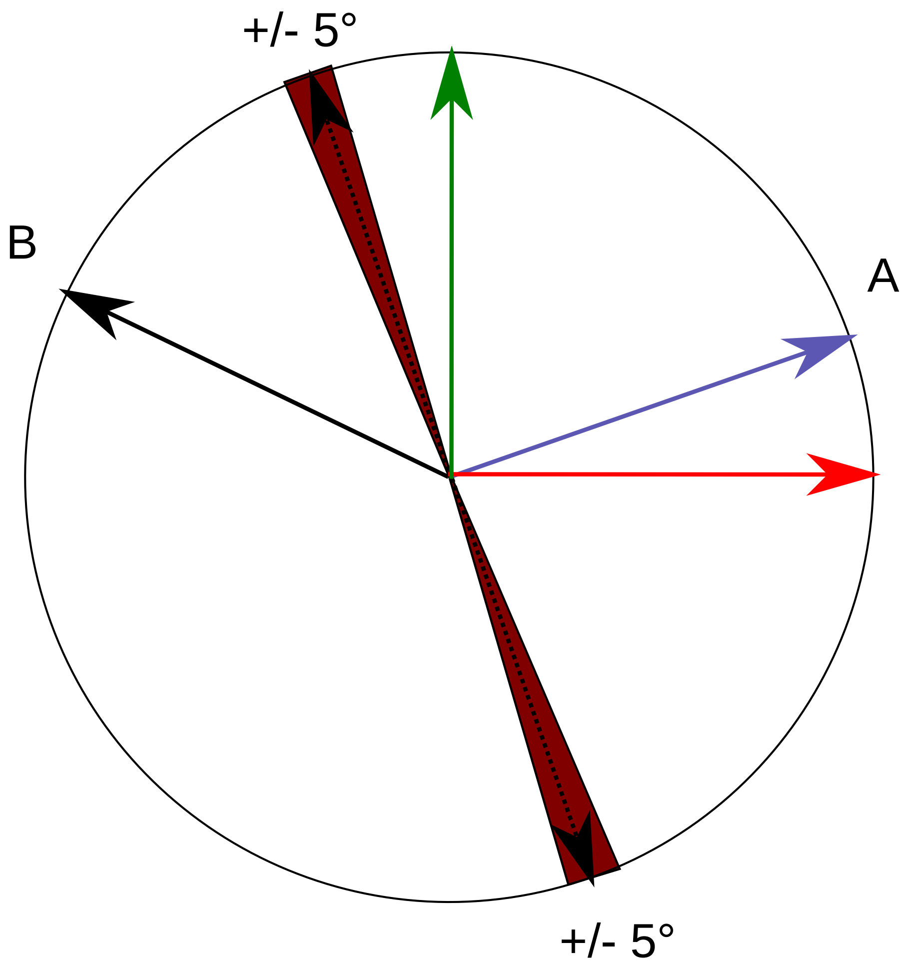
The Trivial
The Bad
The Surprising
The Good



- Random vectors:
  - uniformly distributed angles
  - obtained by sampling each dimension iid. ~N(0,1)
- I bet:
  - Given a random vector A, we can independently sample a second random vector B and it will be almost orthogonal (+/- 5°) ...

Properties 3/4: Experiment

The Trivial
The Bad
The Surprising
The Good

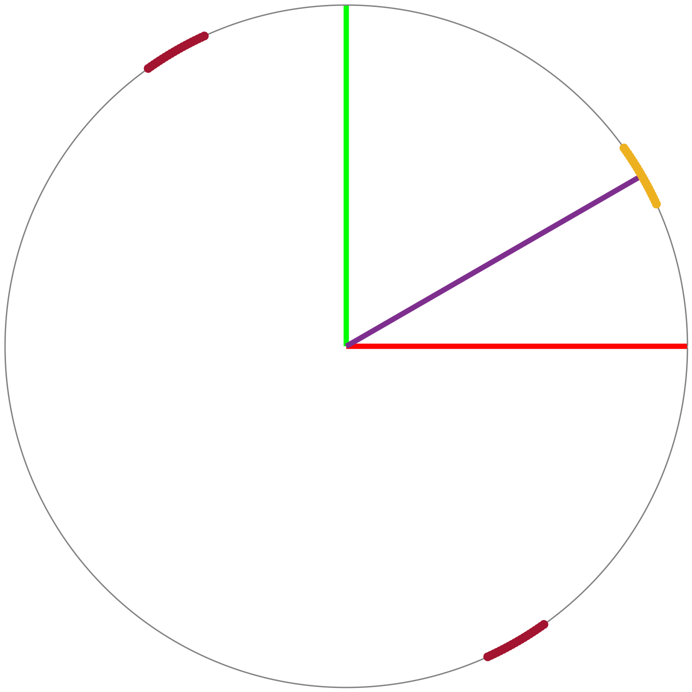


- Random vectors:
  - uniformly distributed angles
  - obtained by sampling each dimension iid. ~N(0,1)
- I bet:
  - Given a random vector A, we can independently sample a second random vector B and it will be almost orthogonal (+/- 5°) ...
  - … if we are in a 4,000 dimensional vector space.

Properties 3/4: Random vectors are very likely almost orthogonal

The Trivial
The Bad
The Surprising
The Good

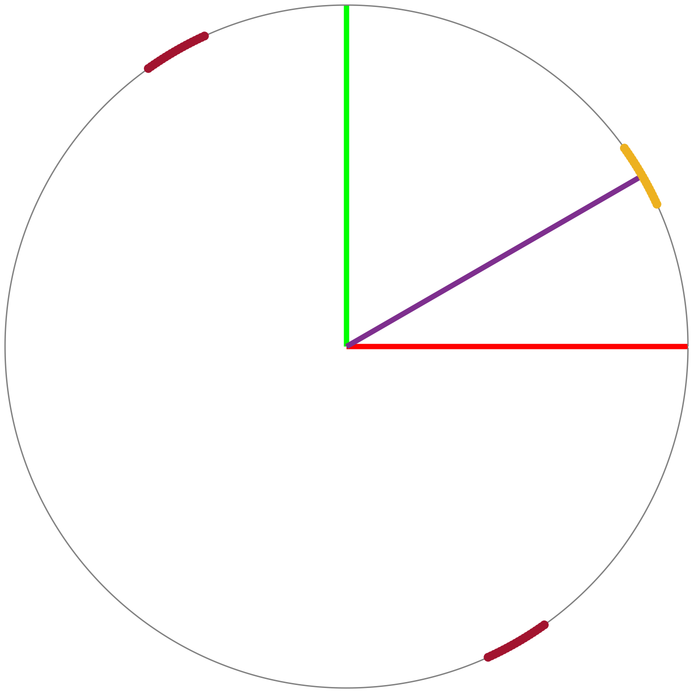
- Random vectors: iid, uniform

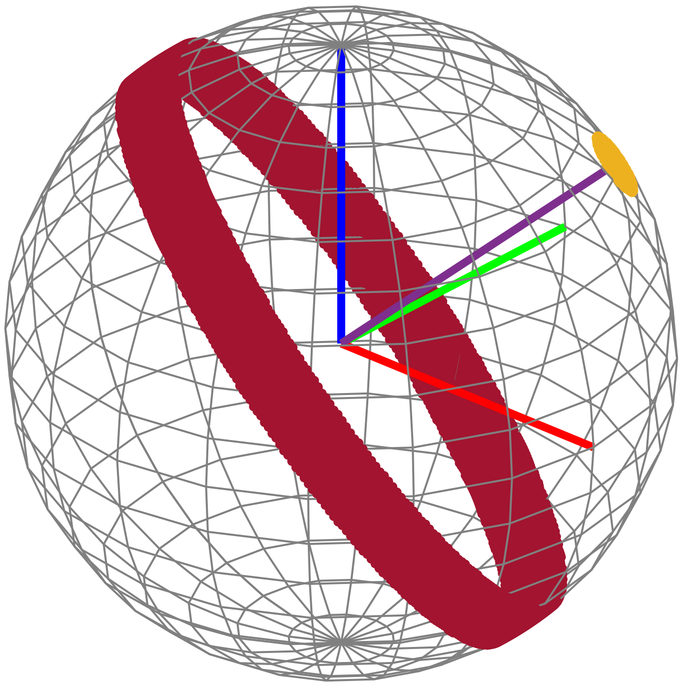


Properties 3/4: Random vectors are very likely almost orthogonal

The Trivial
The Bad
The Surprising
The Good

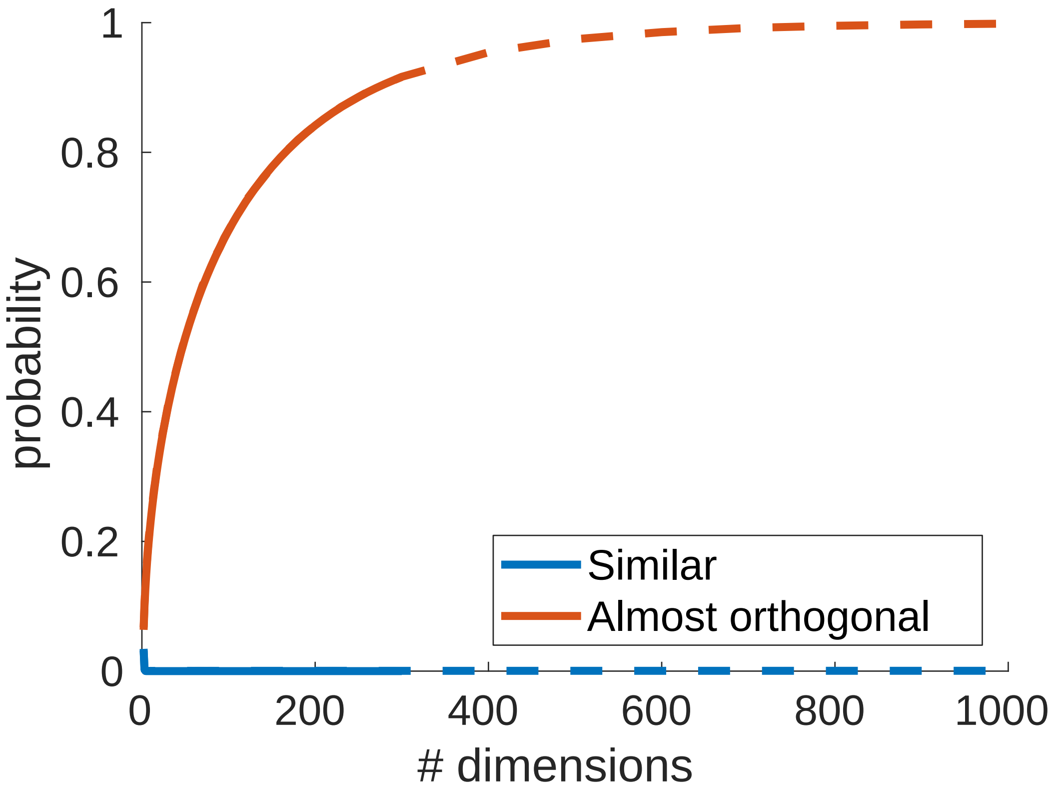
- Random vectors: iid, uniform





Properties 3/4: Random vectors are very likely almost orthogonal

- Random vectors: iid, uniform



Properties 4/4: Noise has low influence on nearest neighbor queries with random vectors

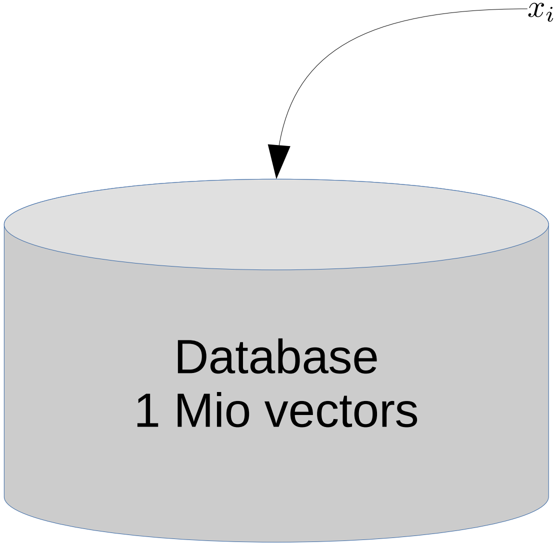
The Trivial
The Bad
The Surprising
The Good

Properties 4/4: Noise has low influence on nearest neighbor queries with random vectors

- Example 1:

$$x_i = \begin{pmatrix} 0.27 \\ 0.95 \\ 0.52 \\ . \\ . \\ . \\ 0.98 \\ 0.22 \\ 0.81 \end{pmatrix}$$

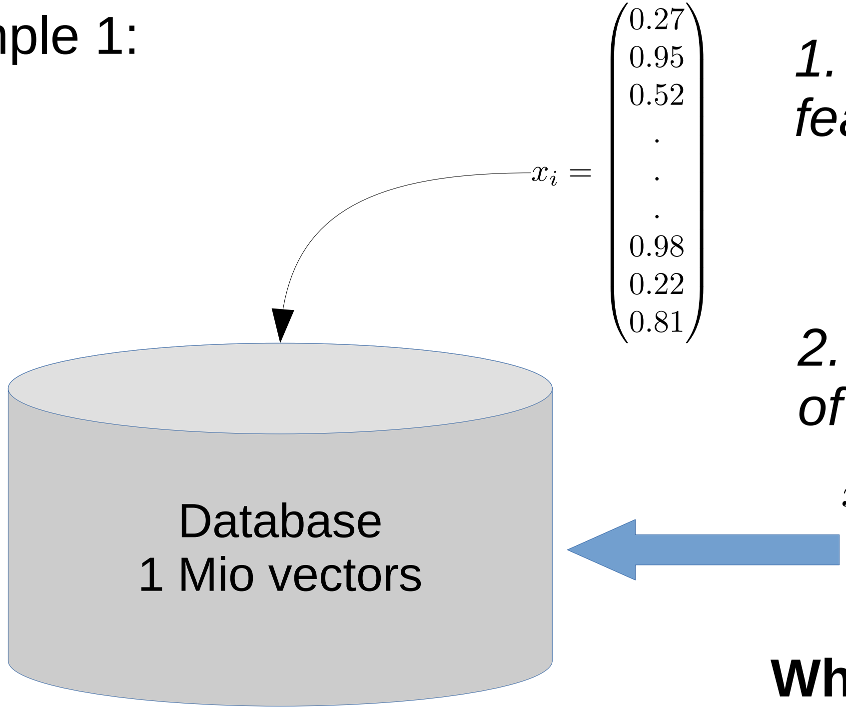
1. One million random feature vectors $[0,1]^d$



Properties 4/4: Noise has low influence on nearest neighbor queries with random vectors

The Trivial
The Bad
The Surprising
The Good

- Example 1:



1. One million random feature vectors [0,1]^d

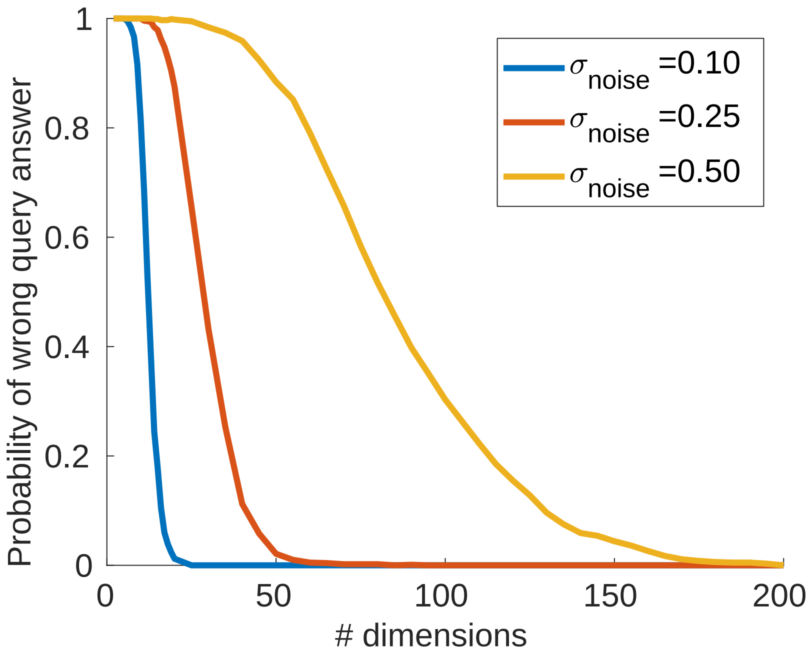
2. query: noisy measurements of feature vectors

$$x_i + \epsilon \text{ with } \epsilon \sim \mathcal{N}(0, \sigma)$$

What is the probability to get the wrong query answer?

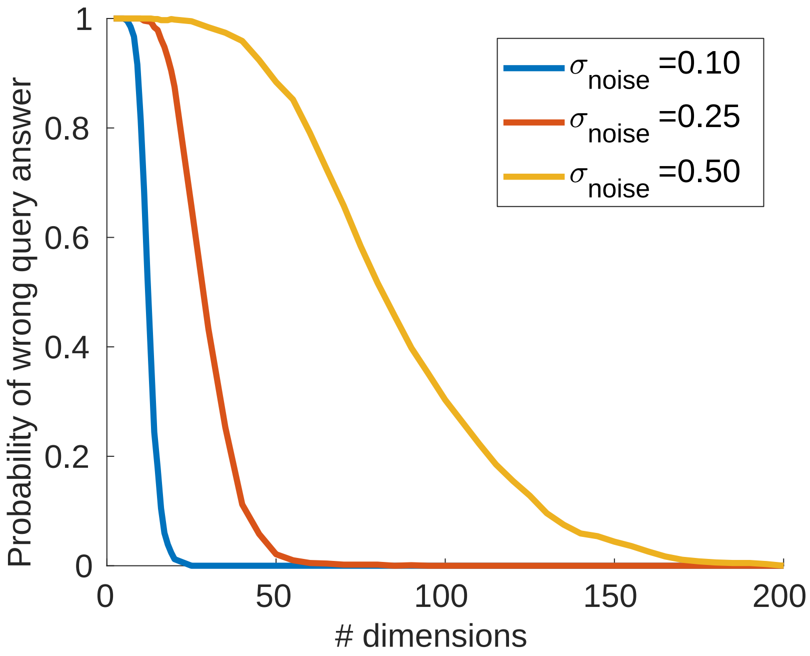
Properties 4/4: Noise has low influence on nearest neighbor queries with random vectors

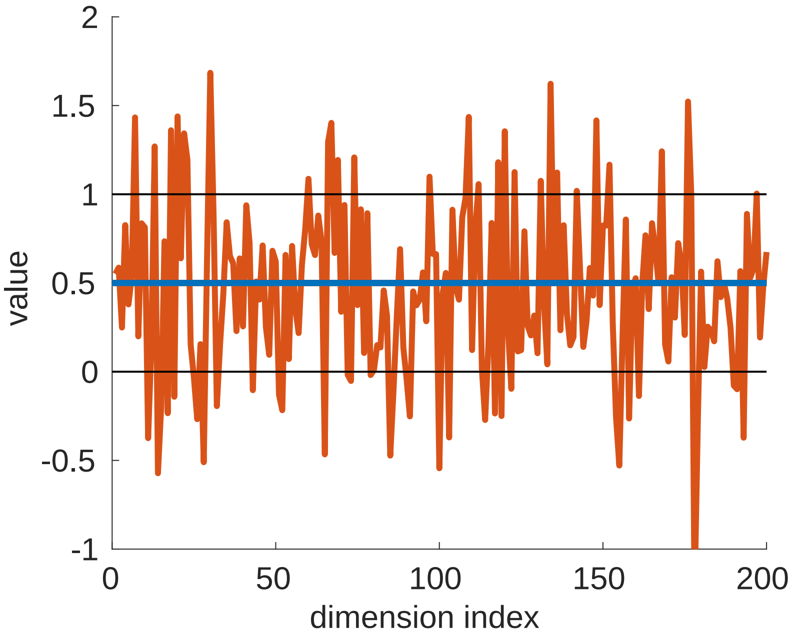
- Example 1:



Properties 4/4: Noise has low influence on nearest neighbor queries with random vectors

- Example 1:



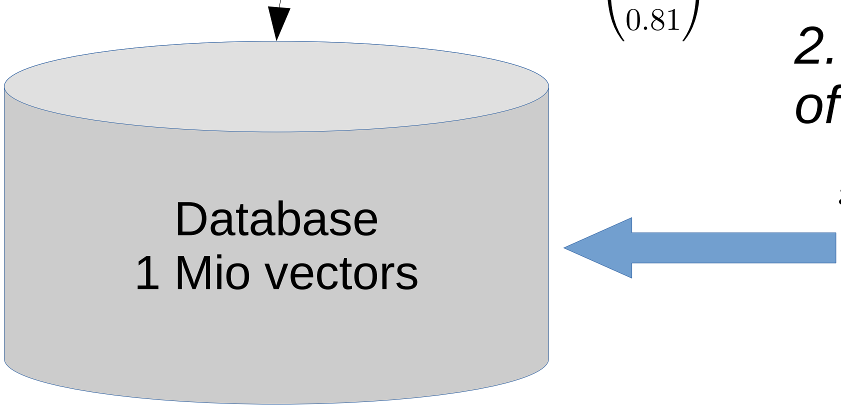


Properties 4/4: Noise has low influence on nearest neighbor queries with random vectors

- Example 2:

$$x_i = \begin{pmatrix} 0.27 \\ 0.95 \\ 0.52 \\ . \\ . \\ . \\ 0.98 \\ 0.22 \\ 0.81 \end{pmatrix}$$

1. One million random feature vectors [0,1]^d



2. query: noisy measurements of feature vectors

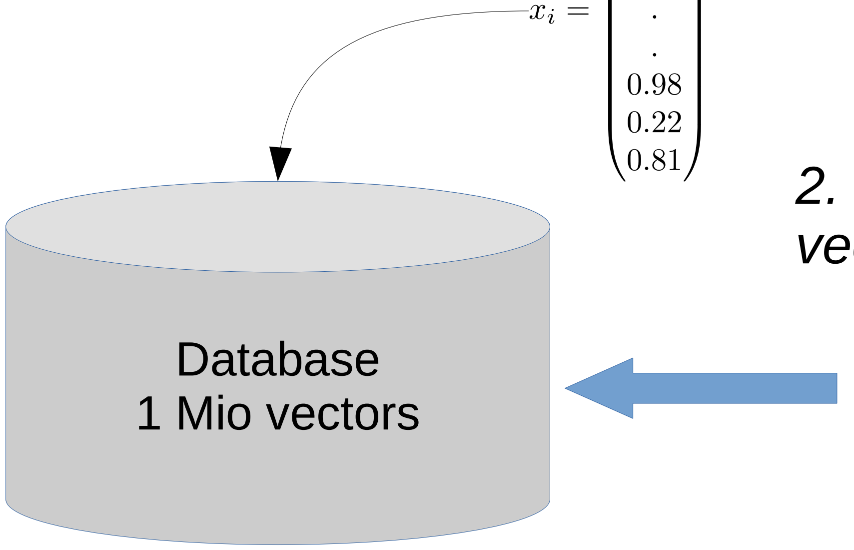
$$x_i + \epsilon \text{ with } \epsilon \sim \mathcal{N}(0, \sigma)$$

What if the noise-vector is again a database vector?

Properties 4/4: Noise has low influence on nearest neighbor queries with random vectors

- Example 2:

$$x_i = \begin{pmatrix} 0.27 \\ 0.95 \\ 0.52 \\ . \\ . \\ . \\ 0.98 \\ 0.22 \\ 0.81 \end{pmatrix}$$



1. One million random feature vectors [0,1]^d

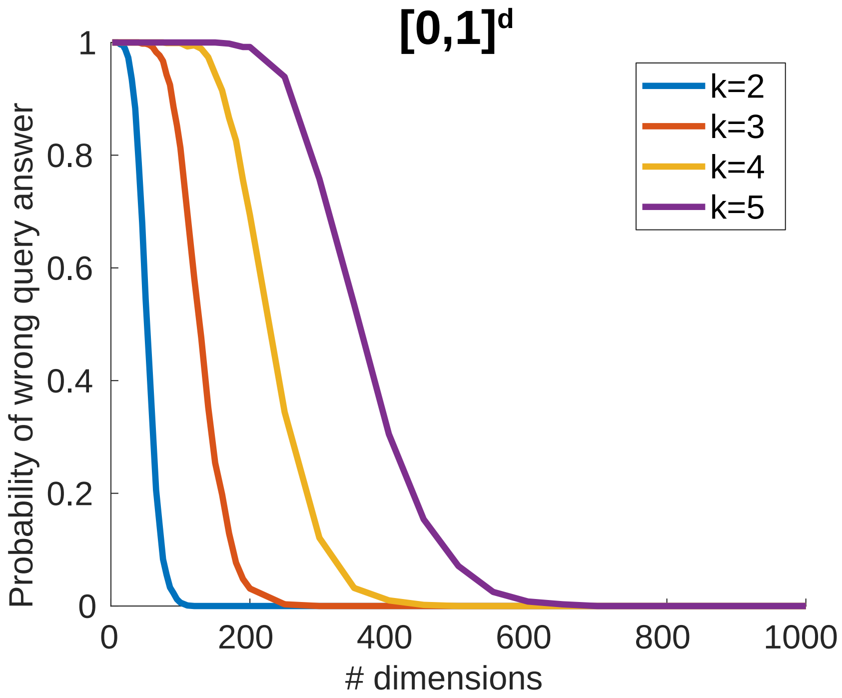
2. query: sum of feature vectors

$$\sum_{i=1}^{k} x_i$$

How many database vectors can we add (=bundle) and still get exactly *all* the added vectors as answer?

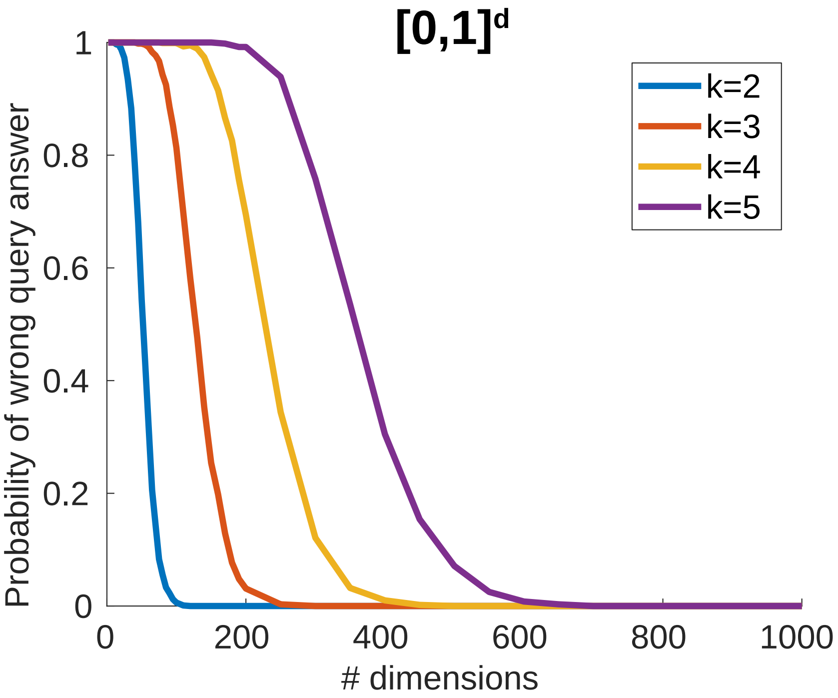
Properties 4/4: Noise has low influence on nearest neighbor queries with random vectors

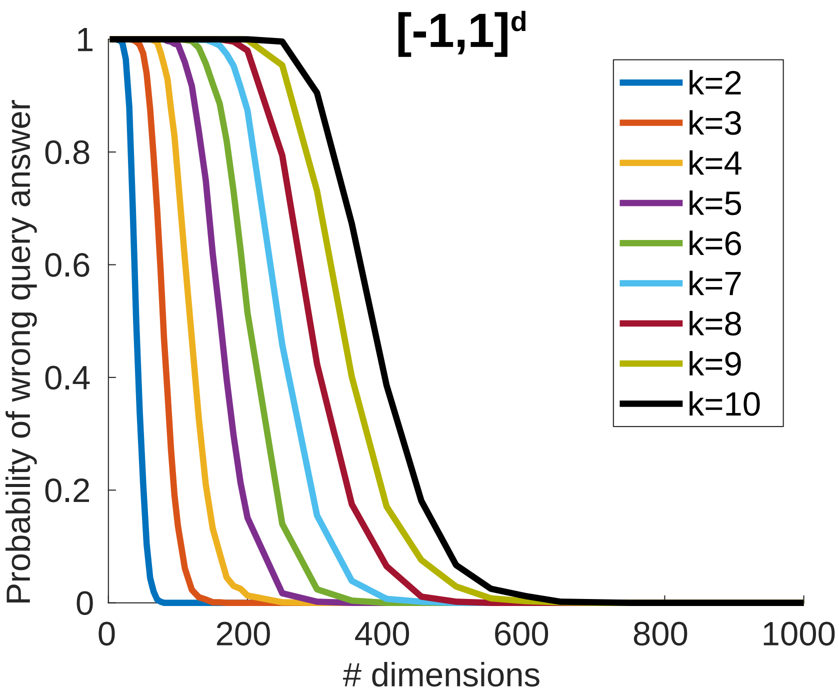
- Example 2:
  - How many database vectors can we add (=bundle) and still get exactly *all* the added vectors as answer?



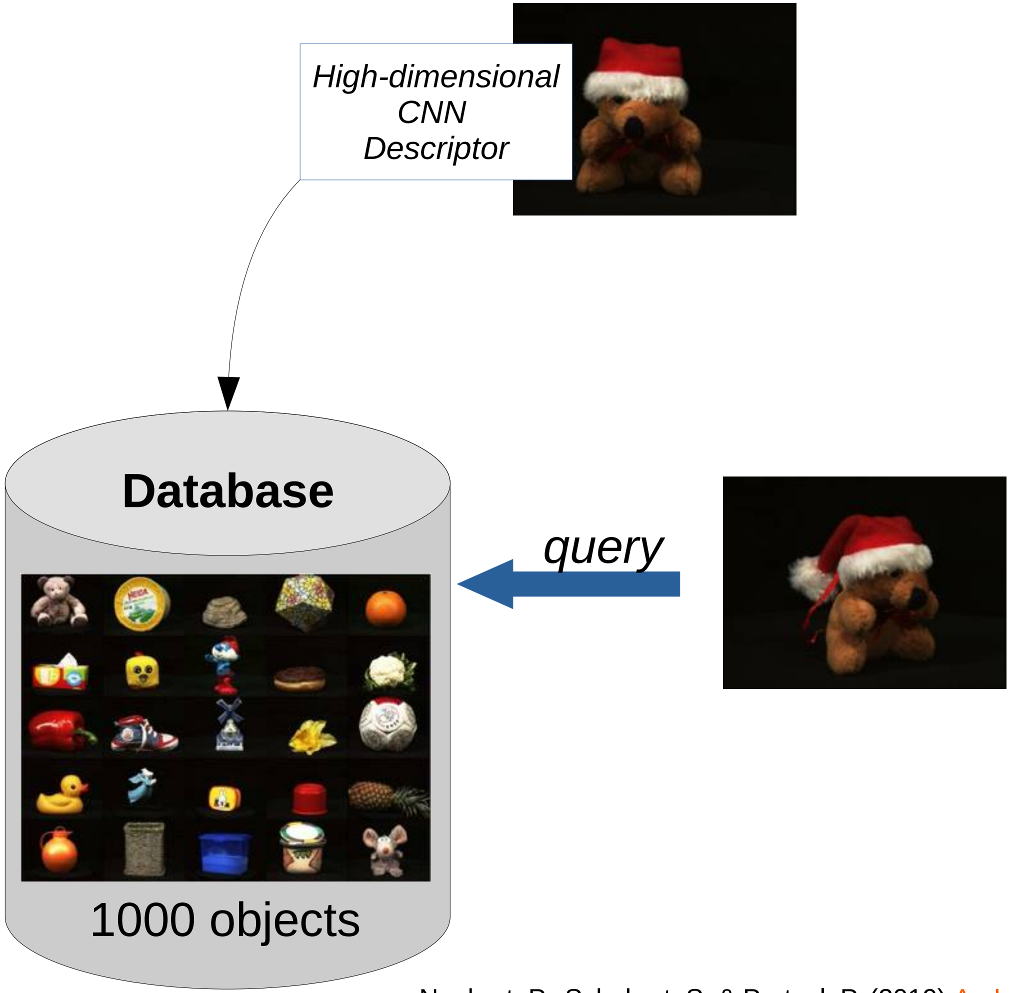
Properties 4/4: Noise has low influence on nearest neighbor queries with random vectors

- Example 2:
  - How many database vectors can we add (=bundle) and still get exactly *all* the added vectors as answer?



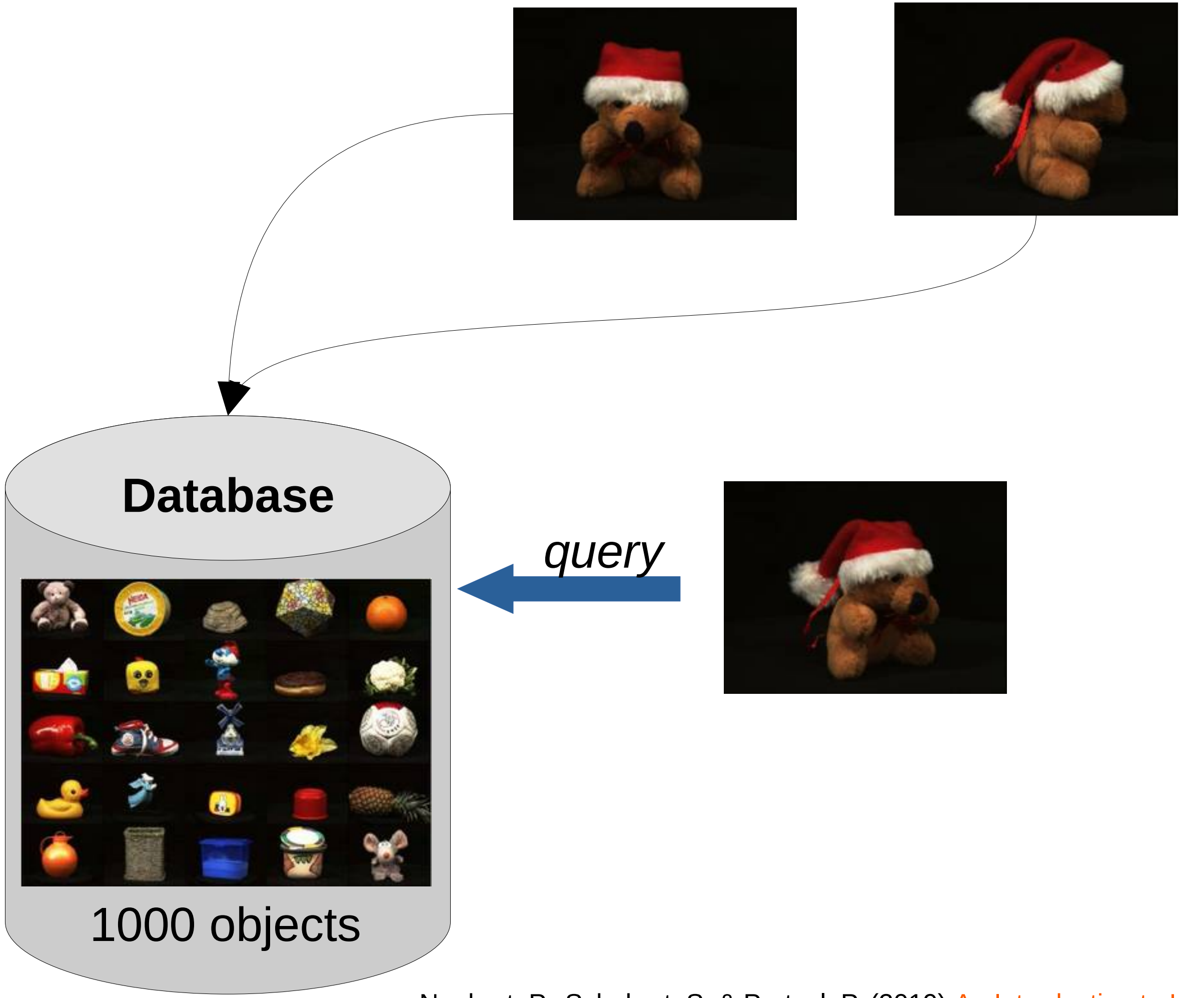


Example application: Object recognition



Neubert, P., Schubert, S. & Protzel, P. (2019) An Introduction to High Dimensional Computing for Robotics. German Journal of Artificial Intelligence

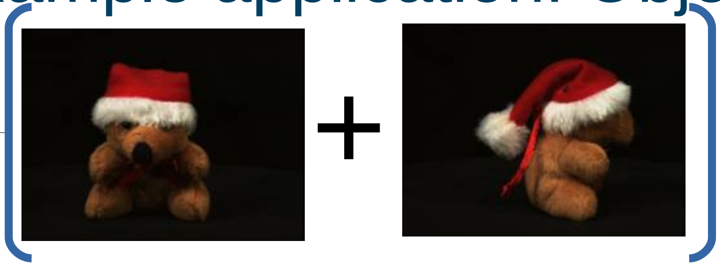
Example application: Object recognition



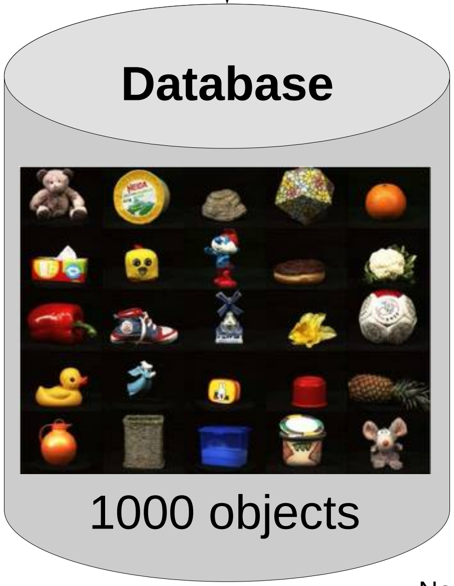
Adding a second view to the DB:
Twice as many comparisons

Neubert, P., Schubert, S. & Protzel, P. (2019) An Introduction to High Dimensional Computing for Robotics. German Journal of Artificial Intelligence

Example application: Object recognition



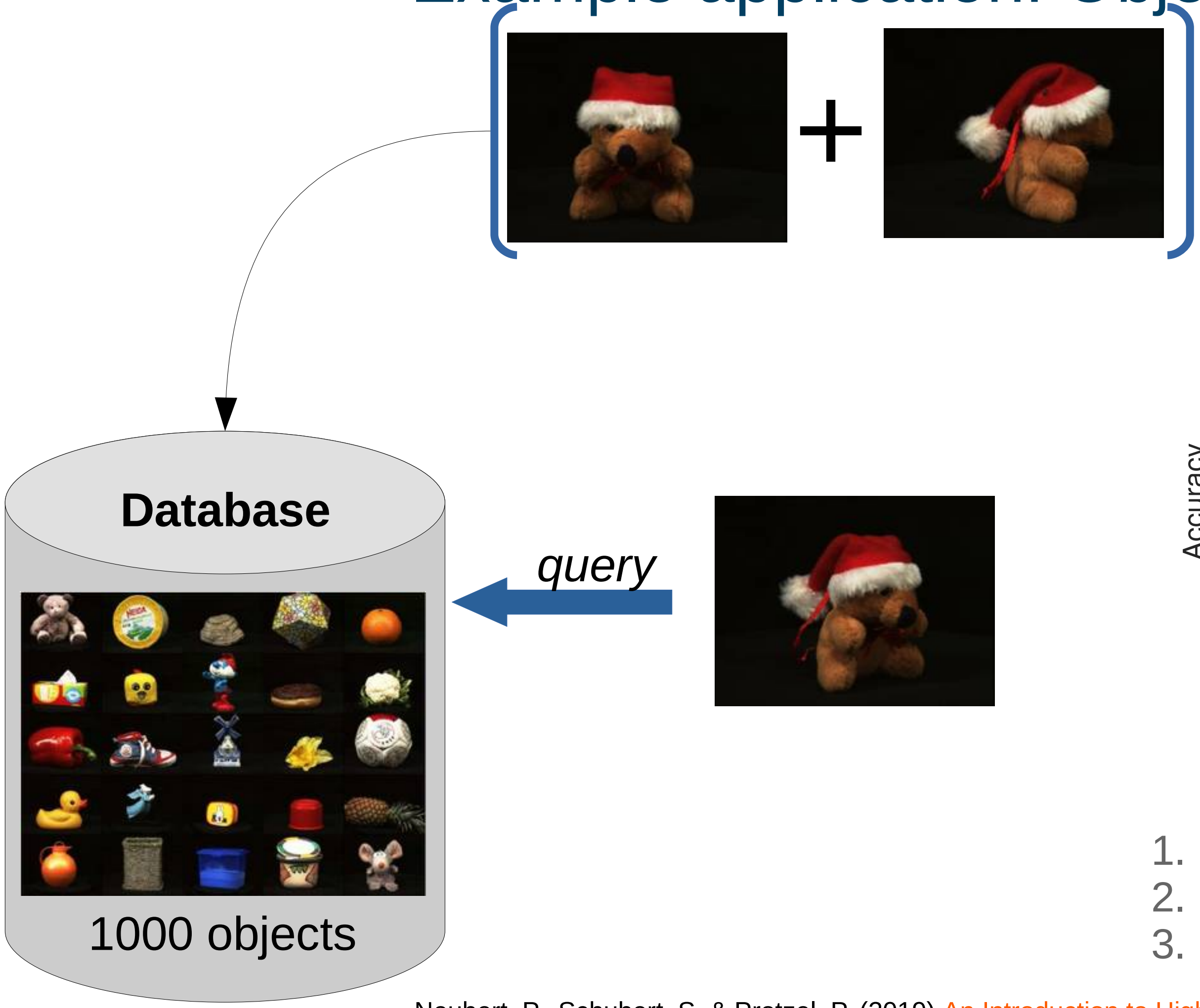
Adding a second view to the DB:
Twice as many comparisons
… unless we bundle



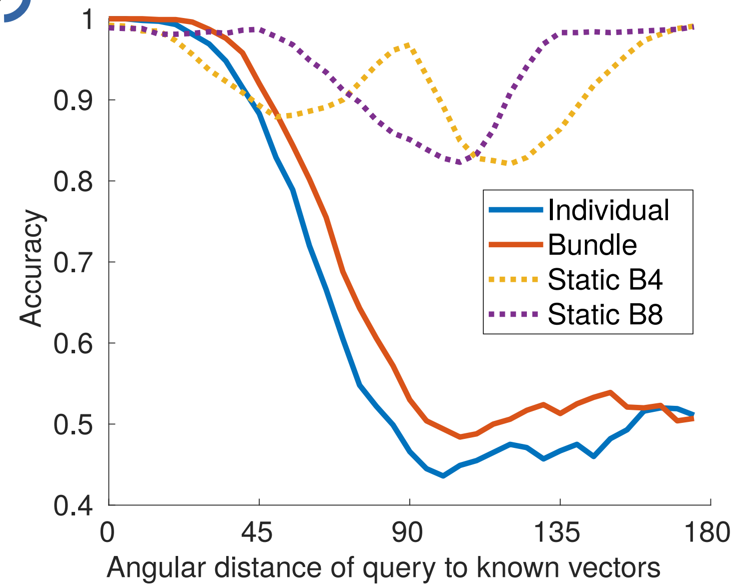
query

Neubert, P., Schubert, S. & Protzel, P. (2019) An Introduction to High Dimensional Computing for Robotics. German Journal of Artificial Intelligence

Example application: Object recognition



Adding a second view to the DB:
Twice as many comparisons
… unless we bundle



1. HDC approach is faster
2. HDC approach is even slightly more accurate
3. Bundling more than two views can help

Neubert, P., Schubert, S. & Protzel, P. (2019) An Introduction to High Dimensional Computing for Robotics. German Journal of Artificial Intelligence

How to store structured data?

Credits: Pentti Kanerva

Given are 2 records:

United States of America

Name:	USA
Capital City:	Washington DC
Currency:	Dollar

Mexico

Name:	Mexico
Capital City:	Mexico City
Currency:	Peso

roles

fillers

Question: What is the Dollar of Mexico?

How to store structured data?

Credits:
Pentti
Kanerva

Given are 2 records:

United States of America

Name:	USA
Capital City:	Washington DC
Currency:	Dollar

Mexico

Name:	Mexico
Capital City:	Mexico City
Currency:	Peso

roles

fillers

Question: What is the Dollar of Mexico?

Hyperdimensional computing approach:

1. Assign a **random** high-dimensional vector to each entity
   ”Name” is a random vector NAM
   ”USA” is a random vector USA
   ”Capital city” is a random vector CAP
   …

2. Calculate a **single** high-dimensional vector that contains all information
   F = (NAM*USA+CAP*WDC+CUR*DOL)*(NAM*MEX+CAP*MCX+CUR*PES)

3. **Calculate** the query answer: F*DOL ~ PES

Outline

1) Introduction to VSA
   - High dimensional vector spaces and where they are used
   - Mathematical properties of high dimensional vector spaces
   - **Vector Symbolic Architectures or “How to do symbolic computations using vectors spaces”**

2) Available VSA implementations

3) Where do the vectors come from?

4) Demo application

5) Discussion

Vector Symbolic Architectures (VSA)

- VSA = high dimensional vector space + operations

Vector Symbolic Architectures (VSA)

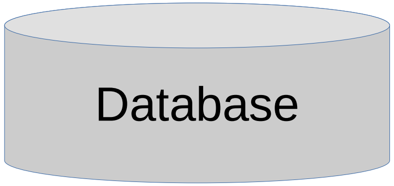
- VSA = high dimensional vector space + operations
- Operations in a VSA:
  - Binding() / Unbinding()
  - Bundling()
  - Permute()/Protect()

Pentti Kanerva. 2009. *Hyperdimensional Computing: An Introduction to Computing in Distributed Representation with High-Dimensional Random Vectors*. Cognitive Computation 1, 2 (2009), 139–159. https://doi.org/10.1007/s12559-009-9009-8

Term: Gayler RW (2003) Vector symbolic architectures answer Jackendoff’s challenges for cognitive neuroscience. In: Proc. of ICCS/ASCS Int. Conf. on cognitive science, pp 133–138. Sydney, Australia

Vector Symbolic Architectures (VSA)

- VSA = high dimensional vector space + operations
- Operations in a VSA:
  - Binding() / Unbinding()
  - Bundling()
  - Permute()/Protect()
- Additionally
  - Encoding/decoding
  - Similarity metric
  - Clean-up memory



Pentti Kanerva. 2009. *Hyperdimensional Computing: An Introduction to Computing in Distributed Representation with High-Dimensional Random Vectors*. Cognitive Computation 1, 2 (2009), 139–159. https://doi.org/10.1007/s12559-009-9009-8

Term: Gayler RW (2003) Vector symbolic architectures answer Jackendoff’s challenges for cognitive neuroscience. In: Proc. of ICCS/ASCS Int. Conf. on cognitive science, pp 133–138. Sydney, Australia

Vector Symbolic Architectures (VSA)

Name	elements X of vector space $\mathbb{V}$	Sim. metric	Bundling	Binding commu-tative	Binding asso-ciative	Unbinding commu-tative	Unbinding asso-ciative
MAP-C	$X \in \mathbb{R}^D$, $X \sim \mathcal{U}(-1,1)$	cosine sim.	elem. addition	elem. multipl. ✓	✓	elem. multipl. ✓	✓
HRR	$X \in \mathbb{R}^D$, $X \sim \mathcal{N}(0, \frac{1}{D})$	cosine sim.	elem. addition	circ. conv. ✓	✓	circ. corr. x	x
VTB	$X \in \mathbb{R}^D$, $X \sim \mathcal{N}(0, \frac{1}{D})$	cosine sim.	elem. addition	VTB x	x	transpose VTB x	x
BSC	$X \in \{0,1\}^D$, $p(X=1) \approx 0.5$	Hamming	elem. addition with threshold	XOR ✓	✓	XOR ✓	✓
MAP-B	$X \in \{-1,1\}^D$, $p(X=1) \approx 0.5$	cosine sim.	elem. addition with threshold	elem. multipl. ✓	✓	elem. multipl. ✓	✓
BSDC-CDT	$X \in \{0,1\}^D$, $p(X=1) = 1/\sqrt{D}$	overlap	disjunction	CDT ✓	✓	-	
BSDC-S	$X \in \{0,1\}^D$, $p(X=1) = 1/\sqrt{D}$	overlap	disjunction	shifting x	x	shifting x	x
FHRR	$X \in \mathbb{C}^D, X = e^{i \cdot \theta}$, $\theta \sim \mathcal{U}(-\pi, \pi)$	angle distance	angles of elem. addition	elem. angle addition ✓	✓	elem. angle subtraction x	x

Schlegel, K., Neubert, P. & Protzel, P. (2020) A comparison of Vector Symbolic Architectures. In arXiv:2001.11797

VSA operations

- Bundling +
  - Goal: combine two vectors in a single vector, such that
    - the result is **similar** to both vectors
  - Application: superpose information

+

VSA operations

- Bundling +
  - Goal: combine two vectors in a single vector, such that
    - the result is **similar** to both vectors
  - Application: superpose information


+


- Binding⊗
  - Goal: combine two vectors in a single vector, such that
    - the result is **non-similar** to both vectors
    - one can be recreated from the result using the other
  - Application: bind value “a” to variable “x” (or a “filler” to a “role” or ...)

Name:	USA
Capital City:	Washington DC
Currency:	Dollar

→

$$NAME_{HV} \otimes USA_{HV}$$
$$CAP_{HV} \otimes WDC_{HV}$$
$$CUR_{HV} \otimes DOL_{HV}$$

VSA operations

- Binding ⊗
  - Properties
    - Associative, commutative
    - Self-inverse: X⊗X=I (or additional *unbind* operator)
    - Result non-similar to input

VSA operations

- Binding ⊗
  - Properties
    - Associative, commutative
    - Self-inverse: $X \otimes X = I$ (or additional *unbind* operator)
    - Result non-similar to input
  - Example:
    - Hypervector space $\{-1,1\}^D$
    - binding can be elementwise multiplication

VSA operations

- Binding⊗
  - Properties
    - Associative, commutative
    - Self-inverse: X⊗X=I (or additional *unbind* operator)
    - Result non-similar to input
  - Example:
    - Hypervector space $\{-1,1\}^D$
    - binding can be elementwise multiplication
  - Application: bind value “a” to variable “x” (or a “filler” to a “role” or ...)

    Binding:

$$R_{HV} = NAME_{HV} \otimes USA_{HV}$$

VSA operations

- Binding⊗
  - Properties
    - Associative, commutative
    - Self-inverse: X⊗X=I (or additional *unbind* operator)
    - Result non-similar to input
  - Example:
    - Hypervector space $\{-1,1\}^D$
    - binding can be elementwise multiplication
  - Application: bind value “a” to variable “x” (or a “filler” to a “role” or ...)

Binding:

$$R_{HV} = NAME_{HV} \otimes USA_{HV}$$

Unbinding: Given R_{HV} what is the value of name?

$$\begin{aligned}
& NAME_{HV} \otimes R_{HV} \\
=& NAME_{HV} \otimes (NAME_{HV} \otimes USA_{HV}) \\
=& (NAME_{HV} \otimes NAME_{HV}) \otimes USA_{HV} \\
=& I \otimes USA_{HV} = USA_{HV}
\end{aligned}$$

VSA operations

- Bundling +
  - Properties
    - Associative, commutative, binding distributes over bundling
    - Result is similar to both inputs

VSA operations

- Bundling +
  - Properties
    - Associative, commutative, binding distributes over bundling
    - Result is similar to both inputs
  - Example
    - Hypervector space $[-1,1]^D$
    - bundling can be elementwise sum, elementwise normalized to [-1,1]

VSA operations

- Bundling +
  - Properties
    - Associative, commutative, binding distributes over bundling
    - Result is similar to both inputs
  - Example
    - Hypervector space $[-1,1]^D$
    - bundling can be elementwise sum, elementwise normalized to [-1,1]
  - Application:

$$R_{HV} = (NAME_{HV} \otimes USA_{HV}) \oplus (CAP_{HV} \otimes WDC_{HV})$$

Unbinding a bundle: Given R_{HV} what is the value of name?

VSA operations

- Bundling +
  - Properties
    - Associative, commutative, binding distributes over bundling
    - Result is similar to both inputs
  - Example
    - Hypervector space $[-1,1]^D$
    - bundling can be elementwise sum, elementwise normalized to [-1,1]
  - Application:

$$R_{HV} = (NAME_{HV} \otimes USA_{HV}) \oplus (CAP_{HV} \otimes WDC_{HV})$$

Unbinding a bundle: Given R_{HV} what is the value of name?

$$NAME_{HV} \otimes R_{HV} = NAME_{HV} \otimes ((NAME_{HV} \otimes USA_{HV}) + (CAP_{HV} \otimes WDC_{HV}))$$

VSA operations

- Bundling +
  - Properties
    - Associative, commutative, binding distributes over bundling
    - Result is similar to both inputs
  - Example
    - Hypervector space $[-1,1]^D$
    - bundling can be elementwise sum, elementwise normalized to [-1,1]
  - Application:

$$R_{HV} = (NAME_{HV} \otimes USA_{HV}) \oplus (CAP_{HV} \otimes WDC_{HV})$$

Unbinding a bundle: Given R_{HV} what is the value of name?

$$\begin{aligned} NAME_{HV} \otimes R_{HV} &= NAME_{HV} \otimes ((NAME_{HV} \otimes USA_{HV}) + (CAP_{HV} \otimes WDC_{HV})) \\ &= (NAME_{HV} \otimes NAME_{HV} \otimes USA_{HV}) + (NAME_{HV} \otimes CAP_{HV} \otimes WDC_{HV})) \end{aligned}$$

VSA operations

- Bundling +
  - Properties
    - Associative, commutative, binding distributes over bundling
    - Result is similar to both inputs
  - Example
    - Hypervector space $[-1,1]^D$
    - bundling can be elementwise sum, elementwise normalized to [-1,1]
  - Application:

$$R_{HV} = (NAME_{HV} \otimes USA_{HV}) \oplus (CAP_{HV} \otimes WDC_{HV})$$

Unbinding a bundle: Given R_{HV} what is the value of name?

$$\begin{aligned} NAME_{HV} \otimes R_{HV} &= NAME_{HV} \otimes ((NAME_{HV} \otimes USA_{HV}) + (CAP_{HV} \otimes WDC_{HV})) \\ &= (NAME_{HV} \otimes NAME_{HV} \otimes USA_{HV}) + (NAME_{HV} \otimes CAP_{HV} \otimes WDC_{HV})) \\ &= (I \otimes USA_{HV}) + noise \end{aligned}$$

VSA operations

- Bundling +
  - Properties
    - Associative, commutative, binding distributes over bundling
    - Result is similar to both inputs
  - Example
    - Hypervector space $[-1,1]^D$
    - bundling can be elementwise sum, elementwise normalized to [-1,1]
  - Application:

$$R_{HV} = (NAME_{HV} \otimes USA_{HV}) \oplus (CAP_{HV} \otimes WDC_{HV})$$

Unbinding a bundle: Given R_{HV} what is the value of name?

$$\begin{aligned}
NAME_{HV} \otimes R_{HV} &= NAME_{HV} \otimes ((NAME_{HV} \otimes USA_{HV}) + (CAP_{HV} \otimes WDC_{HV})) \\
&= (NAME_{HV} \otimes NAME_{HV} \otimes USA_{HV}) + (NAME_{HV} \otimes CAP_{HV} \otimes WDC_{HV})) \\
&= (I \otimes USA_{HV}) + noise \\
&= USA_{HV} + noise
\end{aligned}$$

Teaser application 1: “What is the Dollar of Mexico?”

Credits: Pentti Kanerva

Given are 2 records:

United States of America

Name:	USA
Capital City:	Washington DC
Currency:	Dollar

Mexico

Name:	Mexico
Capital City:	Mexico City
Currency:	Peso

Question: What is the Dollar of Mexico?

Hyperdimensional computing approach:

1. Assign a **random** high-dimensional vector to each entity
   ”Name” is a random vector NAM
   ”USA” is a random vector USA
   ”Capital city” is a random vector CAP
   …

2. Calculate a **single** high-dimensional vector that contains all information
   F = (NAME⊗USA+CAP⊗WDC+CUR⊗DOL)⊗(NAME⊗MEX+CAP⊗MCX+CUR⊗PES)

3. **Calculate** the query answer: F⊗DOL ~ PES

Teaser application 1: “What is the Dollar of Mexico?”

Credits: Pentti Kanerva

$$\mathrm{USTATES} = \left[(\mathrm{NAM} * \mathrm{USA}) + (\mathrm{CAP} * \mathrm{WDC}) + (\mathrm{MON} * \mathrm{DOL})\right]$$

$$\mathrm{MEXICO} = \left[(\mathrm{NAM} * \mathrm{MEX}) + (\mathrm{CAP} * \mathrm{MXC}) + (\mathrm{MON} * \mathrm{PES})\right]$$

Teaser application 1: “What is the Dollar of Mexico?”

Credits: Pentti Kanerva

$$\text{USTATES} = [(\text{NAM} * \text{USA}) + (\text{CAP} * \text{WDC}) + (\text{MON} * \text{DOL})]$$

$$\text{MEXICO} = [(\text{NAM} * \text{MEX}) + (\text{CAP} * \text{MXC}) + (\text{MON} * \text{PES})]$$

$$F_{UM} = \text{USTATES} * \text{MEXICO}$$

Teaser application 1: “What is the Dollar of Mexico?”

Credits: Pentti Kanerva

$$\text{USTATES} = [(\text{NAM} * \text{USA}) + (\text{CAP} * \text{WDC}) + (\text{MON} * \text{DOL})]$$
$$\text{MEXICO} = [(\text{NAM} * \text{MEX}) + (\text{CAP} * \text{MXC}) + (\text{MON} * \text{PES})]$$

$$F_{UM} = \text{USTATES} * \text{MEXICO}$$
$$= [(\text{USA} * \text{MEX}) +$$

$$NAM * USA * NAM * MEX = USA * MEX$$

Teaser application 1: “What is the Dollar of Mexico?”

Credits: Pentti Kanerva

$$\mathrm{USTATES} = [(\mathrm{NAM} * \mathrm{USA}) + (\mathrm{CAP} * \mathrm{WDC}) + (\mathrm{MON} * \mathrm{DOL})]$$

$$\mathrm{MEXICO} = [(\mathrm{NAM} * \mathrm{MEX}) + (\mathrm{CAP} * \mathrm{MXC}) + (\mathrm{MON} * \mathrm{PES})]$$

$$\begin{aligned} F_{UM} &= \mathrm{USTATES} * \mathrm{MEXICO} \\ &= [(\mathrm{USA} * \mathrm{MEX}) + (\mathrm{WDC} * \mathrm{MXC}) \\ &\quad +(\mathrm{DOL} * \mathrm{PES}) + \end{aligned}$$

$$NAM * USA * NAM * MEX = USA * MEX$$

...

Teaser application 1: “What is the Dollar of Mexico?”

Credits: Pentti Kanerva

$$\text{USTATES} = [(\text{NAM} * \text{USA}) + (\text{CAP} * \text{WDC}) + (\text{MON} * \text{DOL})]$$

$$\text{MEXICO} = [(\text{NAM} * \text{MEX}) + (\text{CAP} * \text{MXC}) + (\text{MON} * \text{PES})]$$

$$F_{UM} = \text{USTATES} * \text{MEXICO}$$

$$= [(\text{USA} * \text{MEX}) + (\text{WDC} * \text{MXC}) + (\text{DOL} * \text{PES}) + \text{noise}]$$

$$NAM * USA * NAM * MEX = USA * MEX$$

...

$$NAM * USA * CAP * MXC = noise$$

...

Teaser application 1: “What is the Dollar of Mexico?”

Credits: Pentti Kanerva

$$\text{USTATES} = [(\text{NAM} * \text{USA}) + (\text{CAP} * \text{WDC}) + (\text{MON} * \text{DOL})]$$

$$\text{MEXICO} = [(\text{NAM} * \text{MEX}) + (\text{CAP} * \text{MXC}) + (\text{MON} * \text{PES})]$$

Query: What corresponds to Dollar?

$$\begin{aligned} F_{UM} &= \text{USTATES} * \text{MEXICO} \\ &= [(\text{USA} * \text{MEX}) + (\text{WDC} * \text{MXC}) \\ &\quad + (\text{DOL} * \text{PES}) + \text{noise}] \end{aligned}$$

→ $\text{DOL} * F_{UM}$

Teaser application 1: “What is the Dollar of Mexico?”

Credits: Pentti Kanerva

$$\text{USTATES} = [(\text{NAM} * \text{USA}) + (\text{CAP} * \text{WDC}) + (\text{MON} * \text{DOL})]$$
$$\text{MEXICO} = [(\text{NAM} * \text{MEX}) + (\text{CAP} * \text{MXC}) + (\text{MON} * \text{PES})]$$

$$\begin{aligned} F_{UM} &= \text{USTATES} * \text{MEXICO} \\ &= [(\text{USA} * \text{MEX}) + (\text{WDC} * \text{MXC}) \\ &\quad + (\text{DOL} * \text{PES}) + \text{noise}] \end{aligned}$$

Query: What corresponds to Dollar?

$$\begin{aligned} \text{DOL} * F_{UM} &= \text{DOL} * [(\text{USA} * \text{MEX}) + (\text{WDC} * \text{MXC}) \\ &\quad + (\text{DOL} * \text{PES}) + \text{noise}] \end{aligned}$$

Teaser application 1: “What is the Dollar of Mexico?”

Credits: Pentti Kanerva

$$\text{USTATES} = [(\text{NAM} * \text{USA}) + (\text{CAP} * \text{WDC}) + (\text{MON} * \text{DOL})]$$

$$\text{MEXICO} = [(\text{NAM} * \text{MEX}) + (\text{CAP} * \text{MXC}) + (\text{MON} * \text{PES})]$$

$$\begin{aligned} F_{UM} &= \text{USTATES} * \text{MEXICO} \\ &= [(\text{USA} * \text{MEX}) + (\text{WDC} * \text{MXC}) \\ &\quad +(\text{DOL} * \text{PES}) + \text{noise}] \end{aligned}$$

Query: What corresponds to Dollar?

$$\begin{aligned} \text{DOL} * F_{UM} &= \text{DOL} * [(\text{USA} * \text{MEX}) + (\text{WDC} * \text{MXC}) \\ &\quad +(\text{DOL} * \text{PES}) + \text{noise}] \\ &= [(\text{DOL} * \text{USA} * \text{MEX}) \\ &\quad +(\text{DOL} * \text{WDC} * \text{MXC}) \\ &\quad +\boxed{(\text{DOL} * \text{DOL} * \text{PES})} + (\text{DOL} * \text{noise})] \end{aligned}$$

Teaser application 1: “What is the Dollar of Mexico?”

Credits: Pentti Kanerva

$$\text{USTATES} = [(\text{NAM} * \text{USA}) + (\text{CAP} * \text{WDC}) + (\text{MON} * \text{DOL})]$$
$$\text{MEXICO} = [(\text{NAM} * \text{MEX}) + (\text{CAP} * \text{MXC}) + (\text{MON} * \text{PES})]$$

$$\begin{aligned} F_{UM} &= \text{USTATES} * \text{MEXICO} \\ &= [(\text{USA} * \text{MEX}) + (\text{WDC} * \text{MXC}) \\ &\quad + (\text{DOL} * \text{PES}) + \text{noise}] \end{aligned}$$

→

Query: What corresponds to Dollar?

$$\begin{aligned} \text{DOL} * F_{UM} &= \text{DOL} * [(\text{USA} * \text{MEX}) + (\text{WDC} * \text{MXC}) \\ &\quad + (\text{DOL} * \text{PES}) + \text{noise}] \\ &= [(\text{DOL} * \text{USA} * \text{MEX}) \\ &\quad + (\text{DOL} * \text{WDC} * \text{MXC}) \\ &\quad + \boxed{(\text{DOL} * \text{DOL} * \text{PES})} + (\text{DOL} * \text{noise})] \\ &= [\text{noise}_1 + \text{noise}_2 + \text{PES} + \text{noise}_3] \\ &= [\text{PES} + \text{noise}_4] \\ &\approx \text{PES} \end{aligned}$$

Outline

1) Introduction to VSA
   - High dimensional vector spaces and where they are used
   - Mathematical properties of high dimensional vector spaces
   - Vector Symbolic Architectures or “How to do symbolic computations using vectors spaces”

2) **Available VSA implementations**

3) Where do the vectors come from?

4) Demo application

5) Discussion

VSA Implementation example

```
classdef VSA_MAPC < handle

    %VSA_MAPC class
    properties
        nDims % number of dimensions
    end

    methods
        function obj = VSA_MAPC(nDims)
            obj.nDims = nDims;
        end

        % random vector generation
        function vector = generate(obj,n)
            % generate n random vectors with 'nDim' number of dimensions
            vector = rand([n, obj.nDims])*2-1;
        end

        % bundling
        function bundled_vectors = bundle(obj,input_vectors)
            % element-wise addition
            bundled_vectors = sum(input_vectors);
            % cut the result at -1 and 1
            bundled_vectors(bundled_vectors>1)=1;
            bundled_vectors(bundled_vectors<-1)=-1;
        end

        % binding
        function bound_vector = bind(obj,v1,v2)
            % element-wise multiplication
            bound_vector = v1.*v2;
        end

        % unbinding
        function unbound_vector = unbind(obj,v1,v2)
            % MAPC binding is self-inverse
            unbound_vector = obj.bind(v1,v2);
        end

        % similarity measurement
        function similarity = sim(obj, v1, v2)
            % compute the cosine similarity of the input vectors
            similarity = dot(v1,v2);
            similarity = similarity/(norm(v1)*norm(v2));
        end
    end
end
```

Vector Spaces

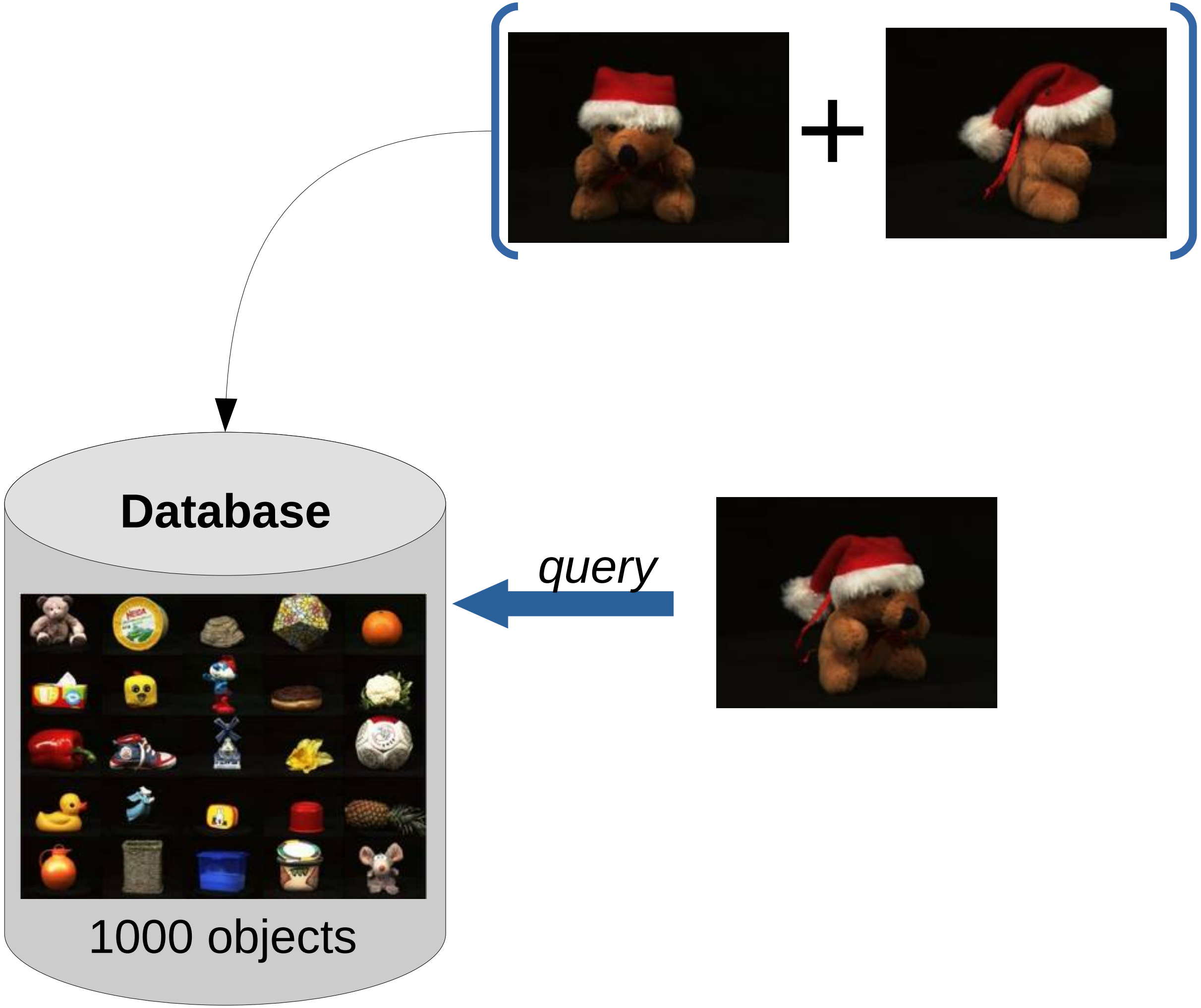
Bipolar		**Binary**		**Real**	**Complex**
[-1,1]	{-1,1}	{0, 1}	sparse	~N(0,1/D)	∢∈[-π,π]
0.62	-1	0	1	-0.43	$e^{-2.65 i}$
0.81	-1	1	0	1.13	$e^{-0.36 i}$
-0.74	1	1	0	0.54	$e^{-2.47 i}$
0.82	1	0	0	0.42	$e^{2.90 i}$
0.26	-1	1	0	-0.02	$e^{-3.11 i}$
-0.80	1	0	0	0.26	$e^{1.72 i}$
-0.44	-1	0	0	-1.83	$e^{1.99 i}$
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
0.94	-1	0	0	-1.23	$e^{1.88 i}$
0.91	1	1	0	-0.46	$e^{-0.43 i}$
-0.02	-1	0	0	1.27	$e^{2.58 i}$
0.60	1	1	0	1.32	$e^{-1.99 i}$
-0.71	-1	0	1	1.83	$e^{-1.48 i}$
-0.15	1	1	0	-1.32	$e^{-2.22 i}$

Vector Symbolic Architectures (VSA)

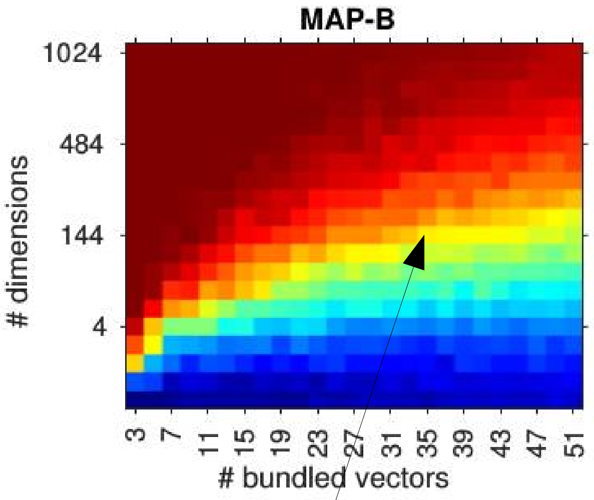
Name	elements X of vector space $\mathbb{V}$	Sim. metric	Bundling	Binding			Unbinding		
					commu-tative	asso-ciative		commu-tative	asso-ciative
MAP-C	$X \in \mathbb{R}^D$, $X \sim \mathcal{U}(-1,1)$	cosine sim.	elem. addition	elem. multipl.	✓	✓	elem. multipl.	✓	✓
HRR	$X \in \mathbb{R}^D$, $X \sim \mathcal{N}(0, \frac{1}{D})$	cosine sim.	elem. addition	circ. conv.	✓	✓	circ. corr.	x	x
VTB	$X \in \mathbb{R}^D$, $X \sim \mathcal{N}(0, \frac{1}{D})$	cosine sim.	elem. addition	VTB	x	x	transpose VTB	x	x
BSC	$X \in \{0,1\}^D$, $p(X=1) \approx 0.5$	Hamming	elem. addition with threshold	XOR	✓	✓	XOR	✓	✓
MAP-B	$X \in \{-1,1\}^D$, $p(X=1) \approx 0.5$	cosine sim.	elem. addition with threshold	elem. multipl.	✓	✓	elem. multipl.	✓	✓
BSDC-CDT	$X \in \{0,1\}^D$, $p(X=1) = 1/\sqrt{D}$	overlap	disjunction	CDT	✓	✓	-		
BSDC-S	$X \in \{0,1\}^D$, $p(X=1) = 1/\sqrt{D}$	overlap	disjunction	shifting	x	x	shifting	x	x
FHRR	$X \in \mathbb{C}^D, X = e^{i \cdot \theta}$, $\theta \sim \mathcal{U}(-\pi,\pi)$	angle distance	angles of elem. addition	elem. angle addition	✓	✓	elem. angle subtraction	x	x

Schlegel, K., Neubert, P. & Protzel, P. (2020) A comparison of Vector Symbolic Architectures. In ArXiv:2001.11797

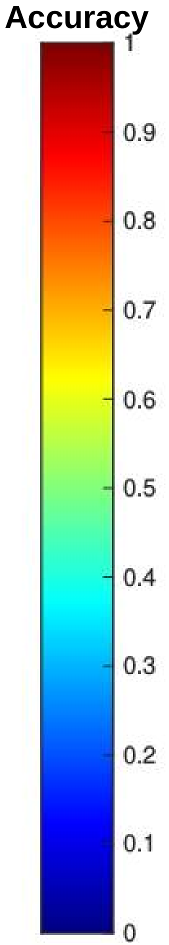
Exp 1: Bundling capacity of different VSAs



Exp 1: Bundling capacity of different VSAs



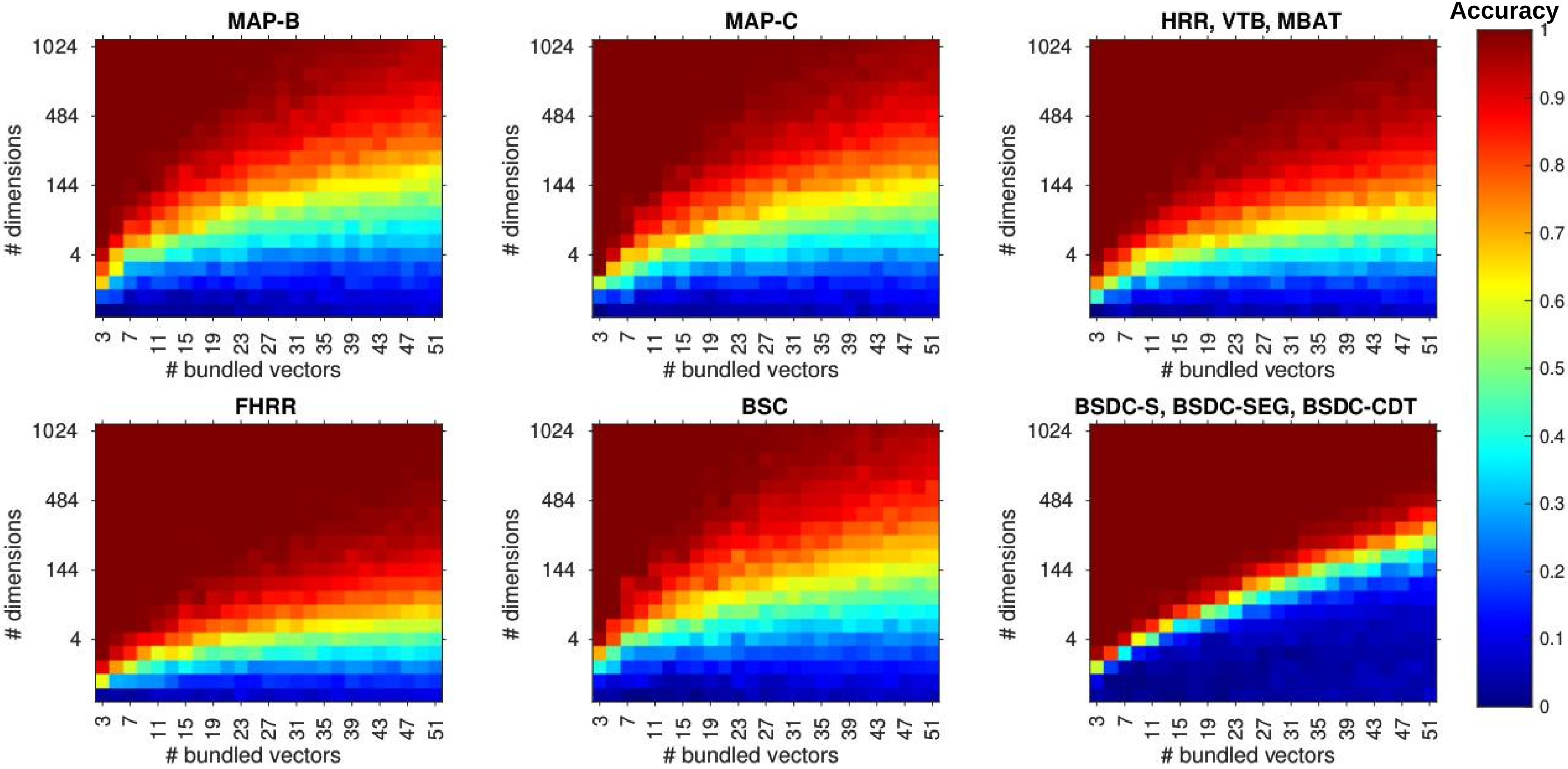
- Database with 1000 random vectors
- Query with a bundle of k vectors
- Compute accuracy using the k-NN neighbors from the database



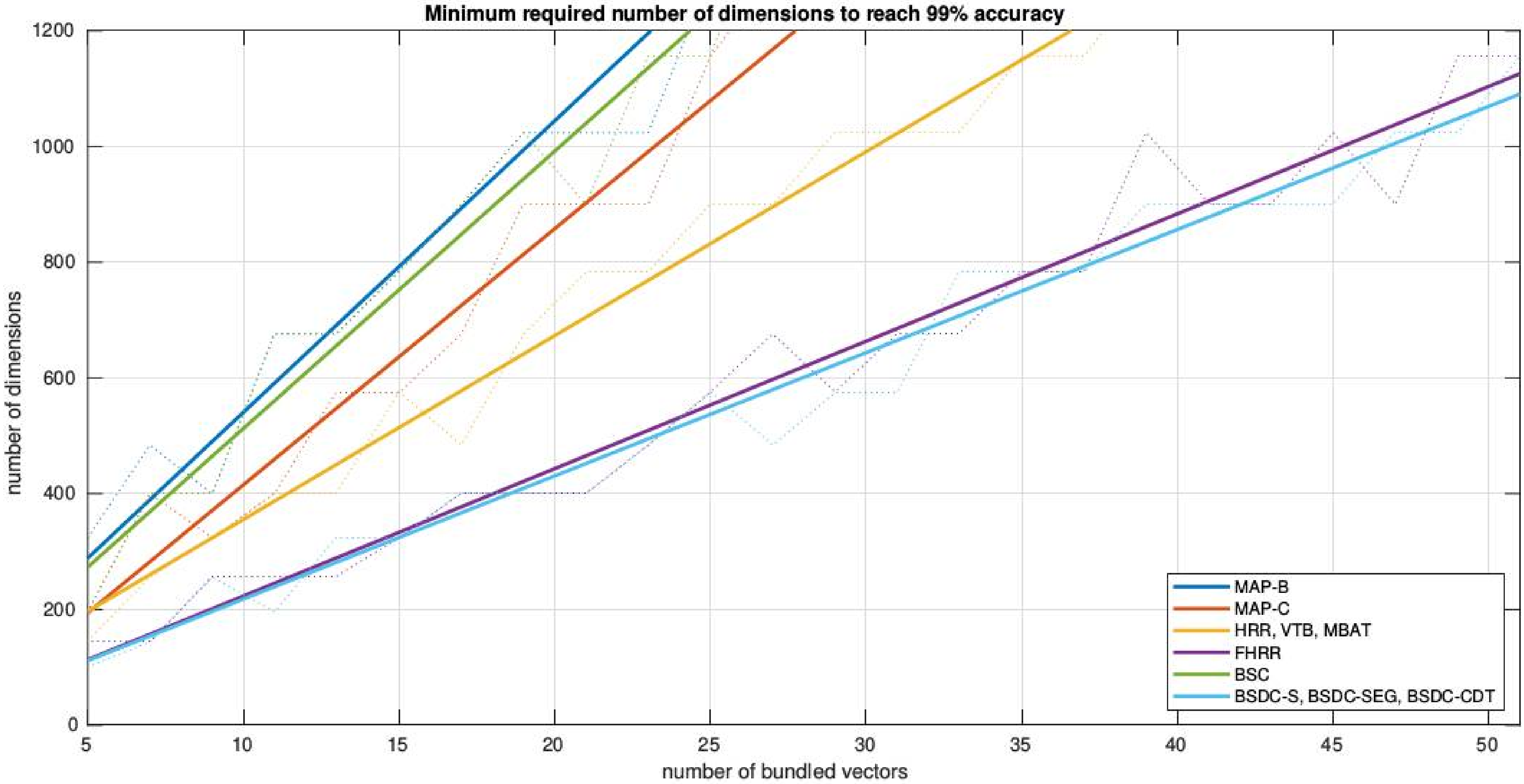
"Bundling 35 vectors in a single MAP-B vector with 144 dimensions retrieves all 35 vectors in about 60 % of all trials"

Schlegel, K., Neubert, P. & Protzel, P. (2020) A comparison of Vector Symbolic Architectures. In ArXiv:2001.11797

Exp 1: Bundling capacity of different VSAs



Exp 1: Bundling capacity of different VSAs



Schlegel, K., Neubert, P. & Protzel, P. (2020) A comparison of Vector Symbolic Architectures. In ArXiv:2001.11797

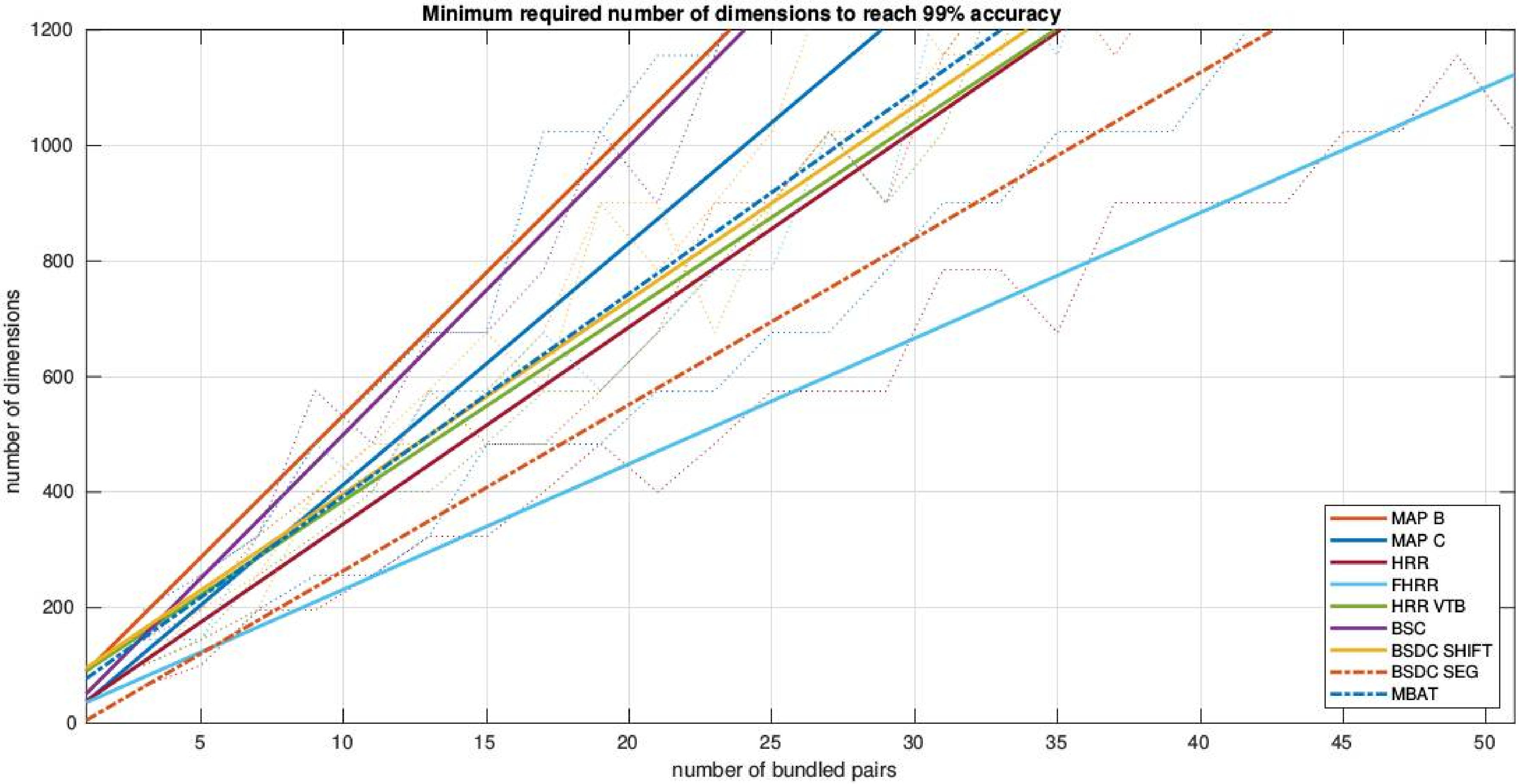
Exp 2: Role-filler query capacity (combining bundling and binding)

$$\text{USTATES} = [(\text{NAM} * \text{USA}) + (\text{CAP} * \text{WDC}) + (\text{MON} * \text{DOL})]$$

How many role-filler pairs can we bundle and still successfully query a filler using a single unbinding operation?

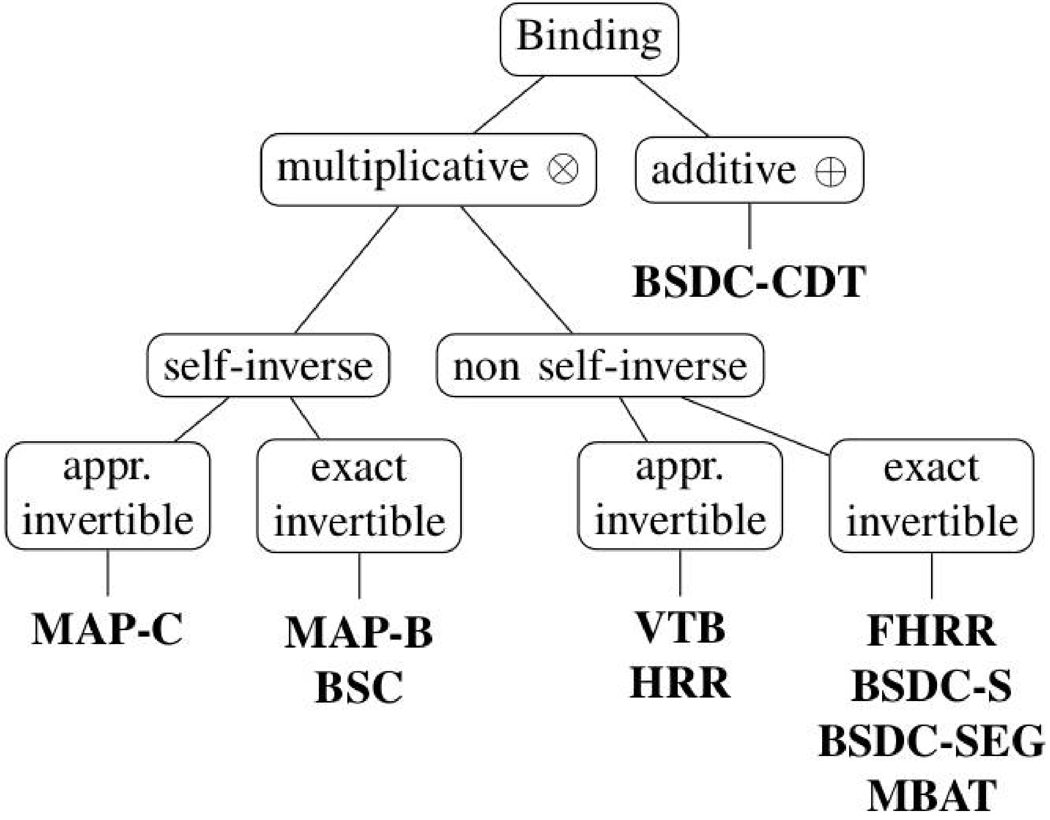
Schlegel, K., Neubert, P. & Protzel, P. (2020) A comparison of Vector Symbolic Architectures. In ArXiv:2001.11797

Exp 2: Role-filler query capacity (combining bundling and binding)



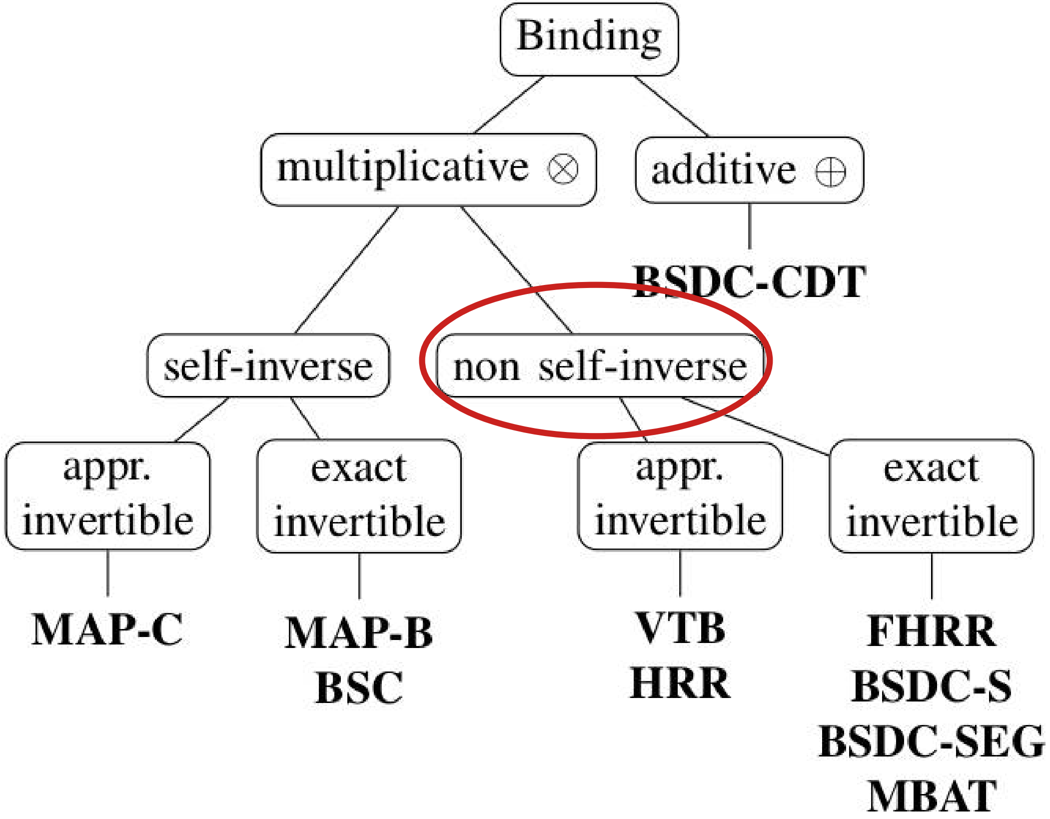
Schlegel, K., Neubert, P. & Protzel, P. (2020) A comparison of Vector Symbolic Architectures. In ArXiv:2001.11797

Taxonomy of binding implementations



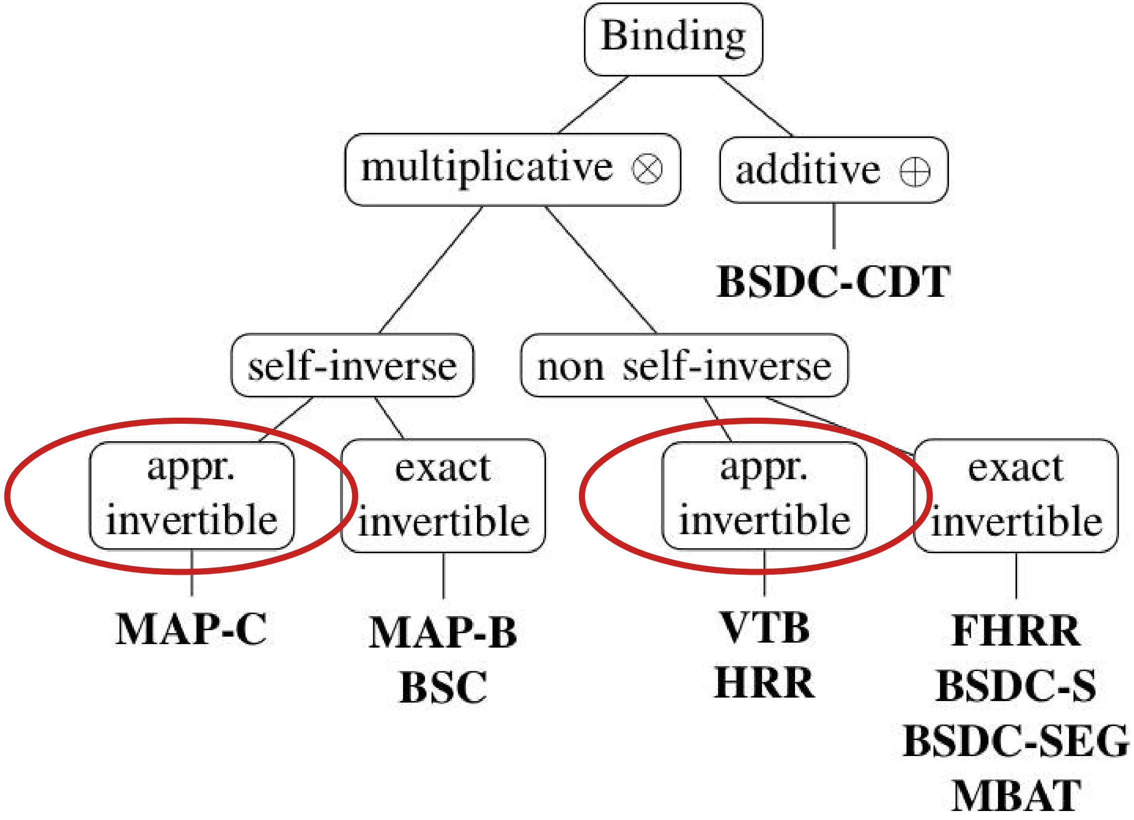
Schlegel, K., Neubert, P. & Protzel, P. (2020) A comparison of Vector Symbolic Architectures. In ArXiv:2001.11797

Taxonomy of binding implementations



Schlegel, K., Neubert, P. & Protzel, P. (2020) A comparison of Vector Symbolic Architectures. In ArXiv:2001.11797

Taxonomy of binding implementations



Schlegel, K., Neubert, P. & Protzel, P. (2020) A comparison of Vector Symbolic Architectures. In ArXiv:2001.11797

Outline

1) Introduction to VSA
   - High dimensional vector spaces and where they are used
   - Mathematical properties of high dimensional vector spaces
   - Vector Symbolic Architectures or “How to do symbolic computations using vectors spaces”

2) Available VSA implementations

3) Where do the vectors come from?

4) Demo application

5) Discussion

Where do the vectors come from?

1) Random vectors
2) Result of a VSA operation
3) Systematic encoding

Systematic encoding

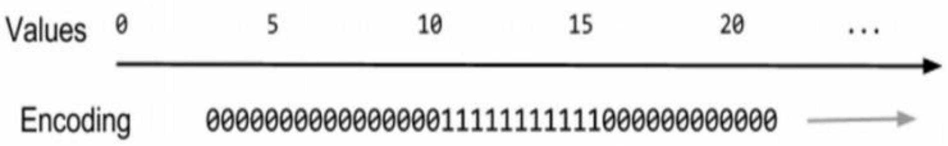
- Potential requirements
  - Distributed representation
  - Similarity preservation → trade-off with quasi-orthogonality

Systematic encoding

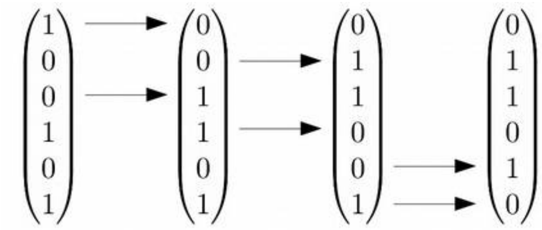
- Potential requirements
  - Distributed representation
  - Similarity preservation → trade-off with quasi-orthogonality

- Examples

Purdy (2016). Encoding Data for HTM Systems.
CoRR abs/1602.05925 (2016)

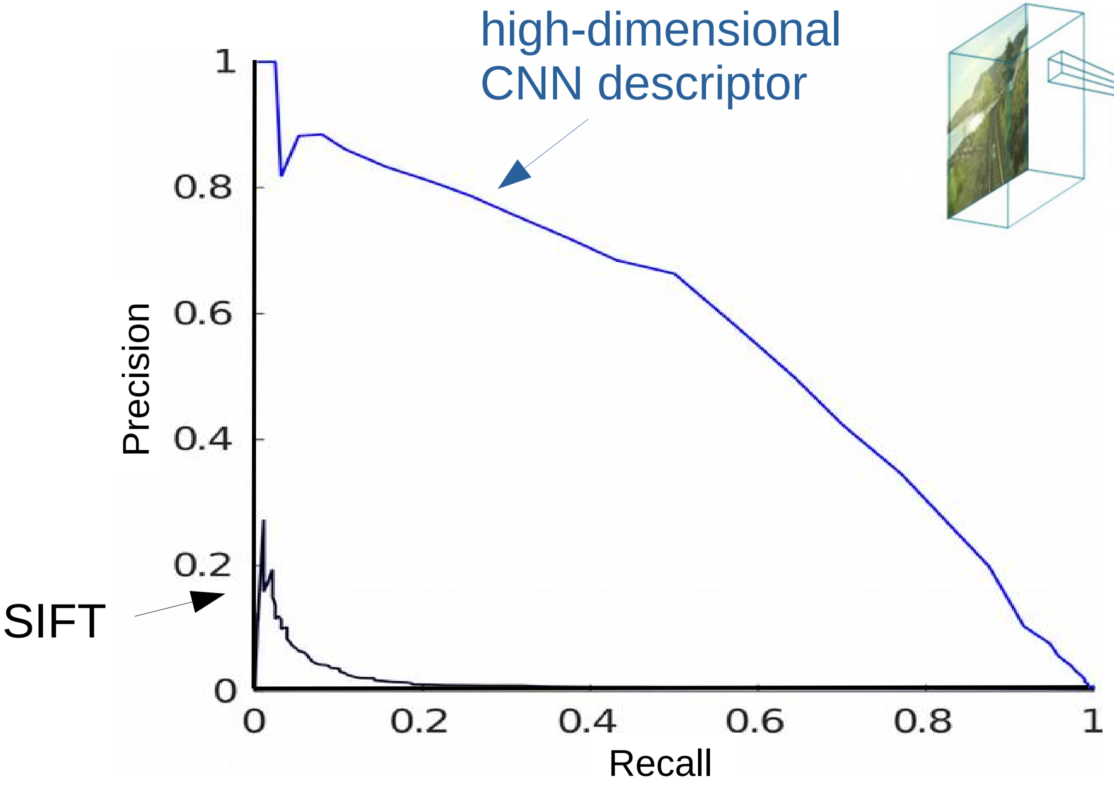


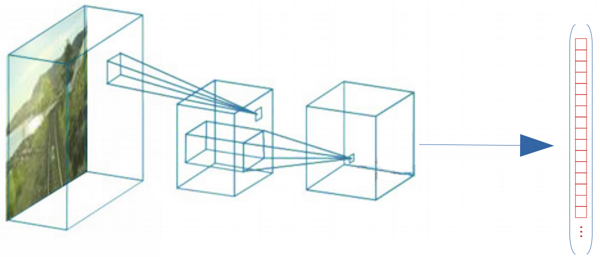
Kleyko et al. (2018). Classification and Recall With Binary Hyperdimensional Computing: Tradeoffs in Choice of Density and Mapping Characteristics. Trans. on Neural Networks and Learning Systems



Systematic encoding of images: Nordland dataset and CNN Descriptors

Systematic encoding of images: Nordland dataset and CNN Descriptors

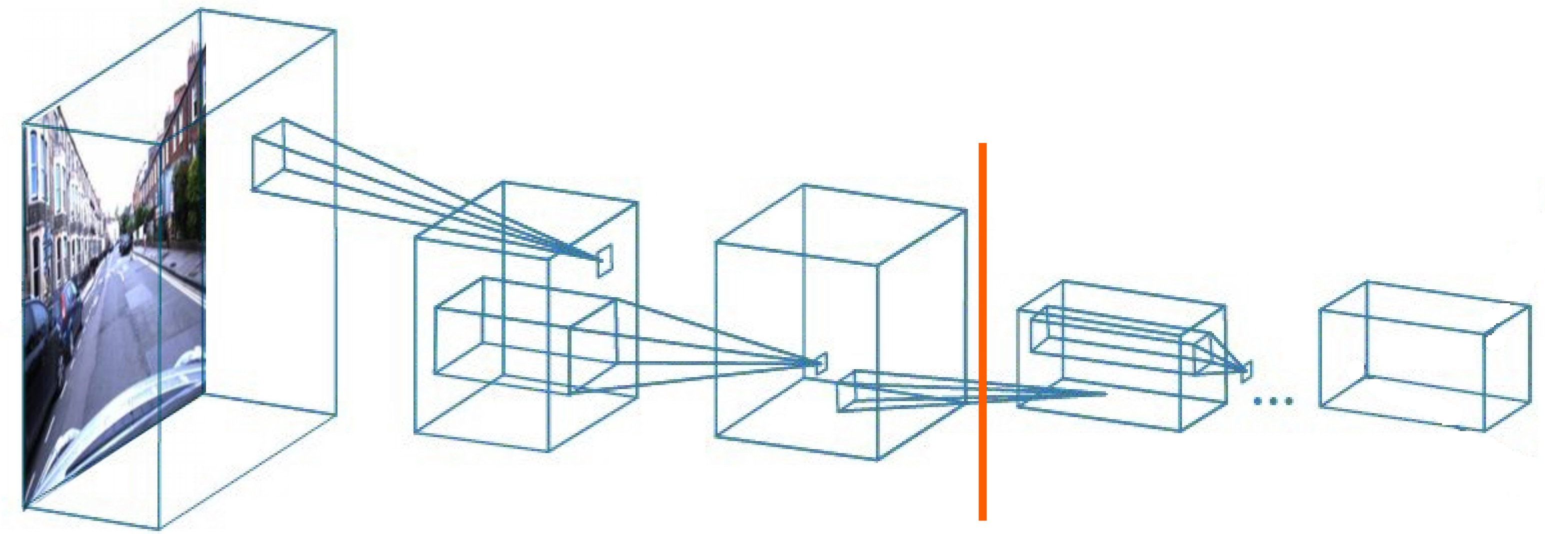




What exactly is a CNN descriptor?

- e.g., flattened output of early convolutional layer of a classification net (e.g. AlexNet)

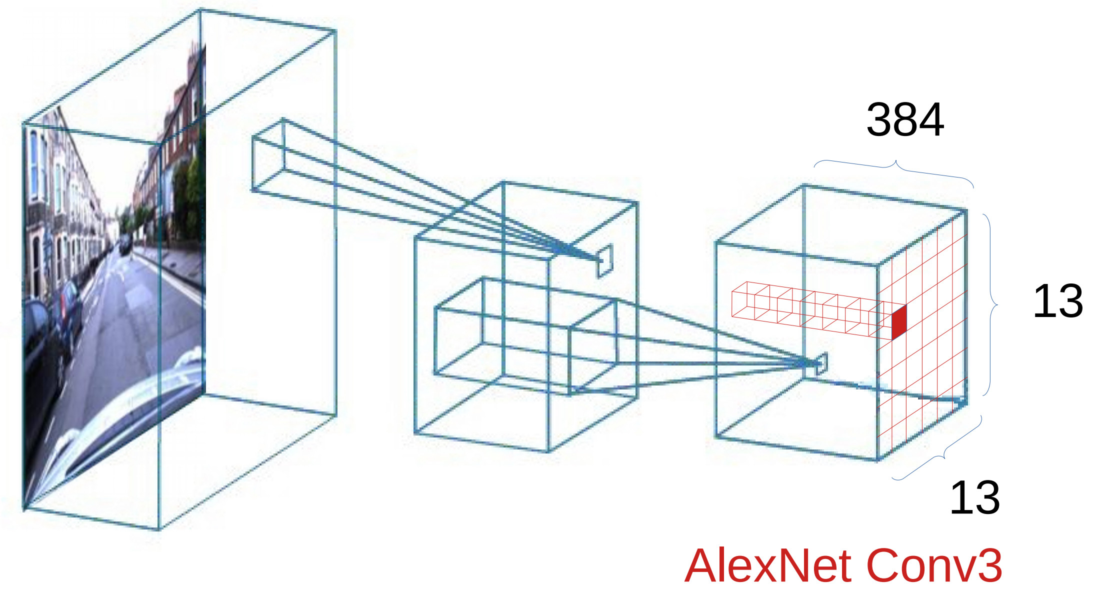
CNN



What exactly is a CNN descriptor?

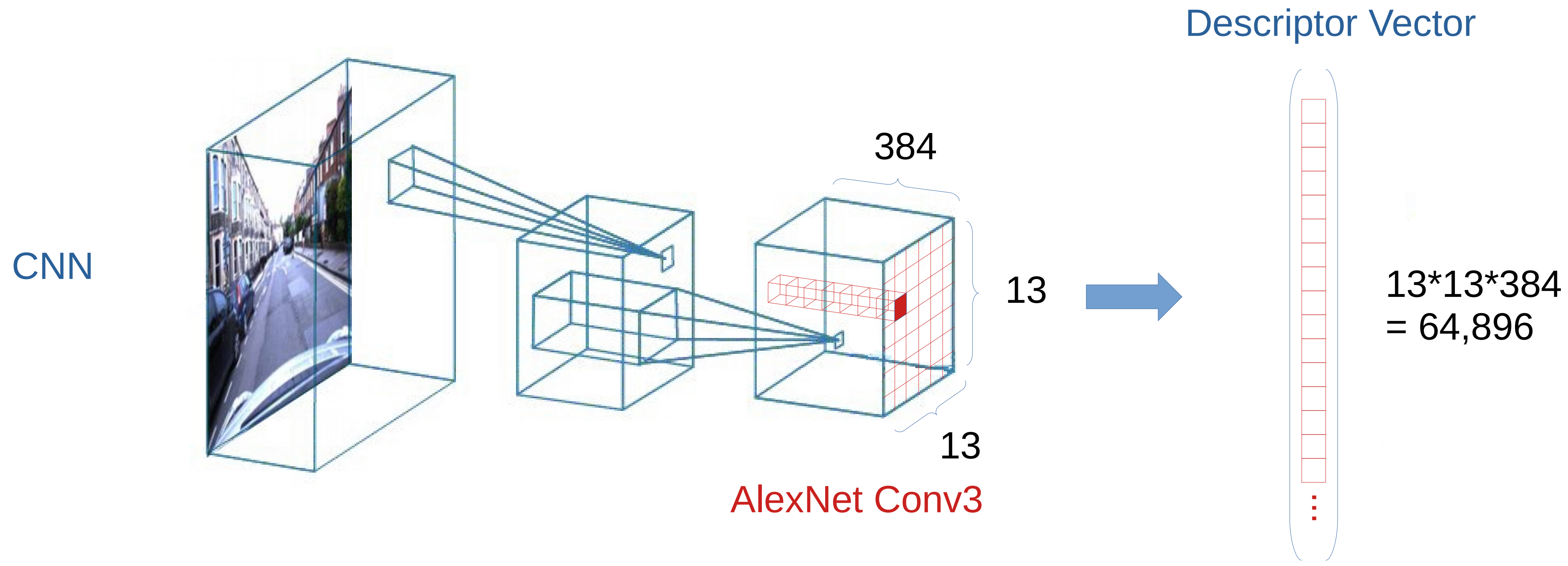
- e.g., flattened output of early convolutional layer of a classification net (e.g. AlexNet)

CNN



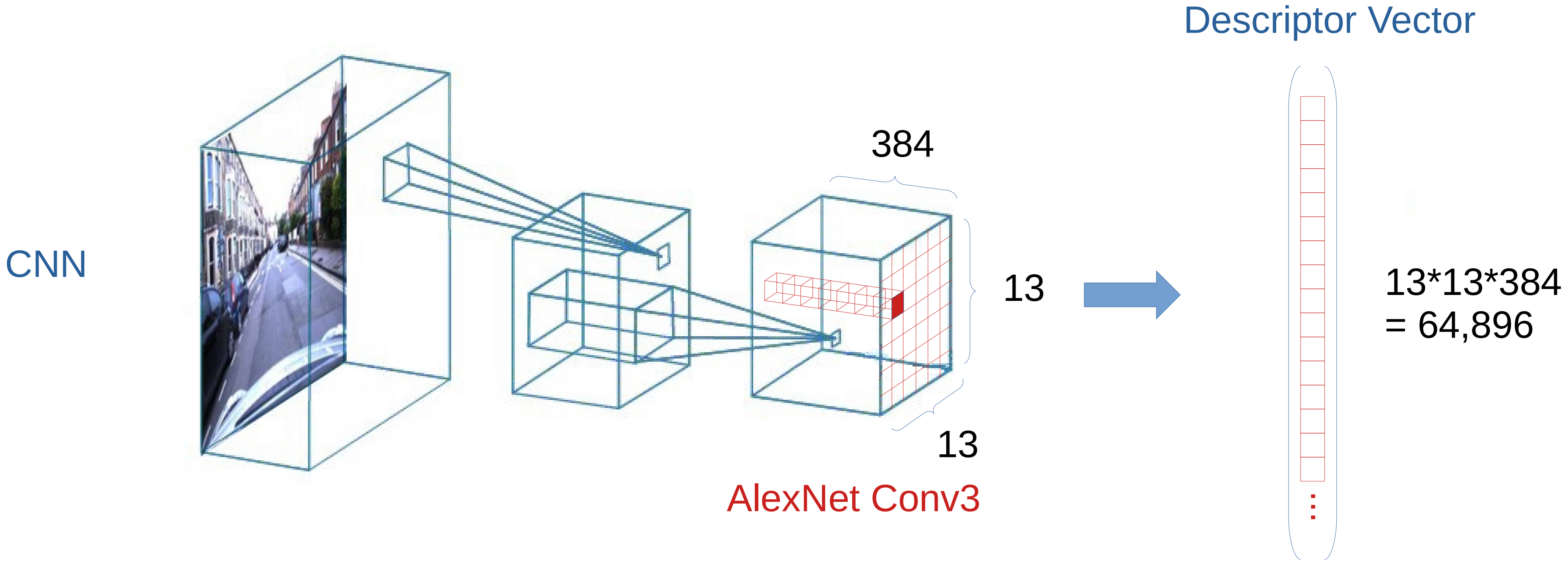
What exactly is a CNN descriptor?

- e.g., flattened output of early convolutional layer of a classification net (e.g. AlexNet)



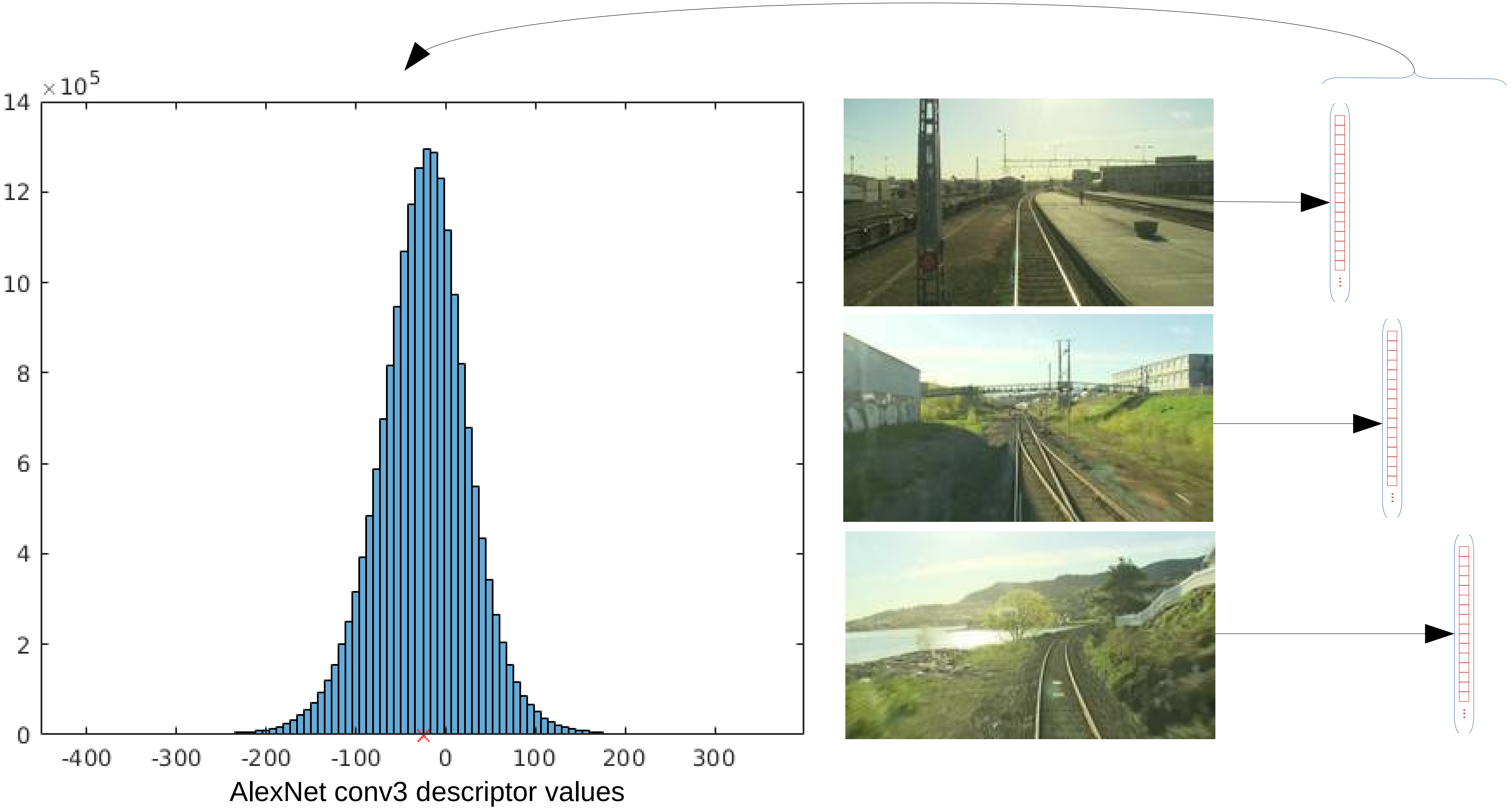
What exactly is a CNN descriptor?

- e.g., flattened output of early convolutional layer of a classification net (e.g. AlexNet)

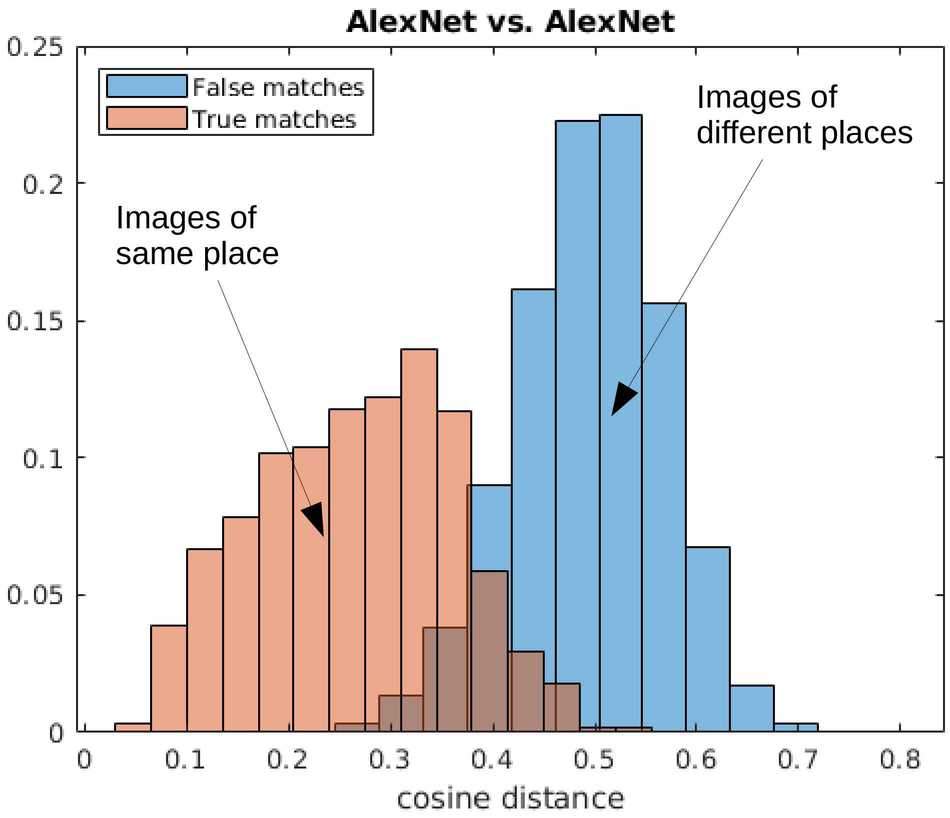


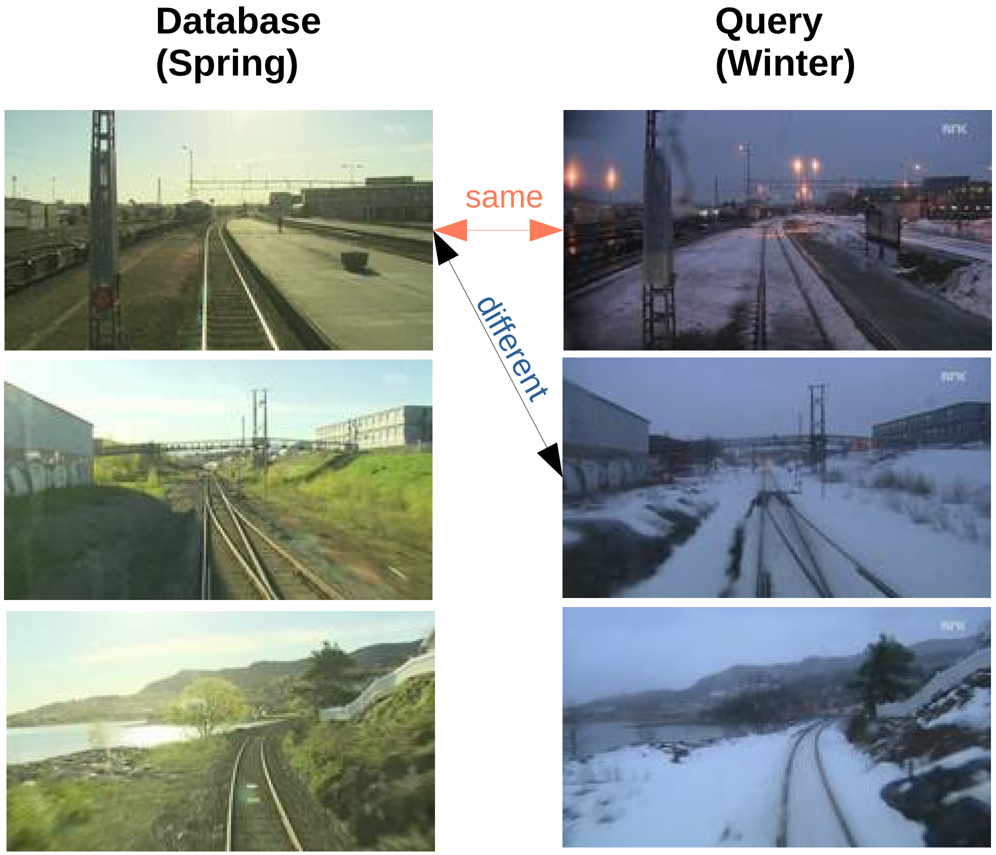
- or particularly trained descriptors for place recognition, e.g., NetVLAD (4,096 dimensions)

Statistical properties of CNN descriptors

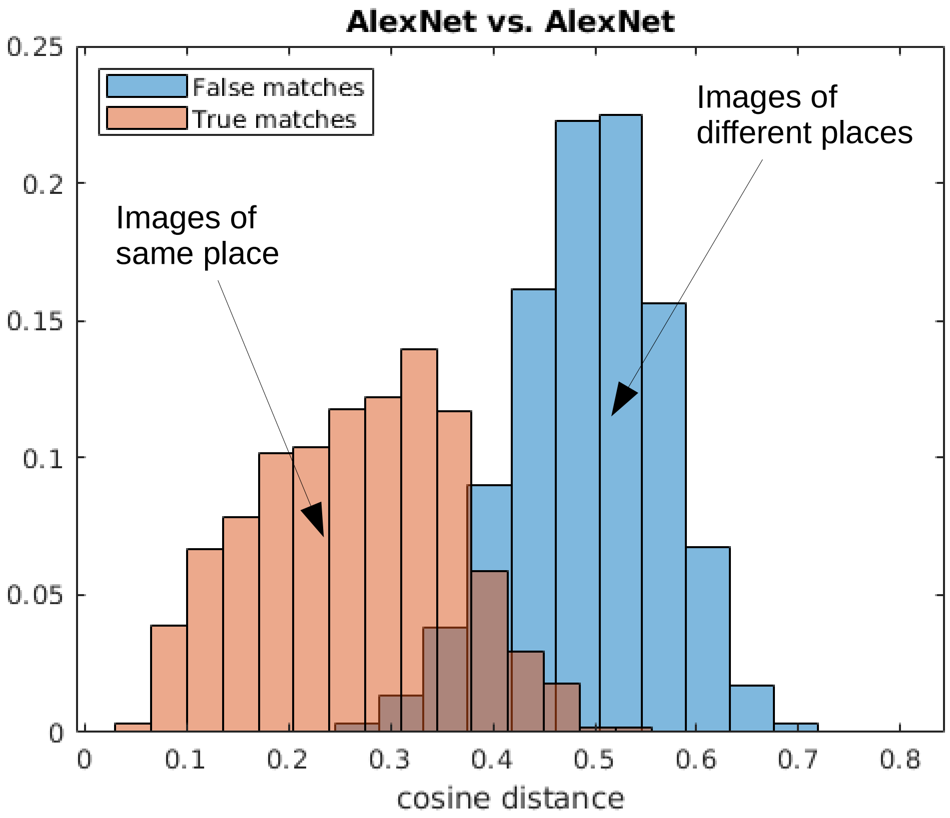


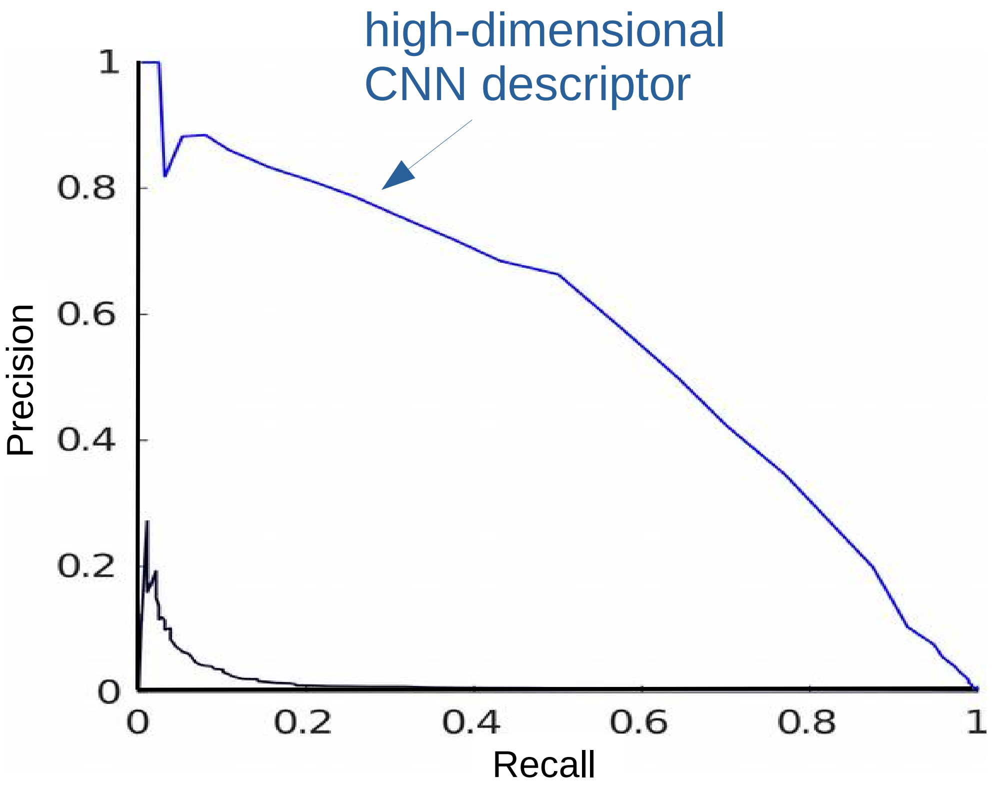
Statistical properties of CNN descriptors



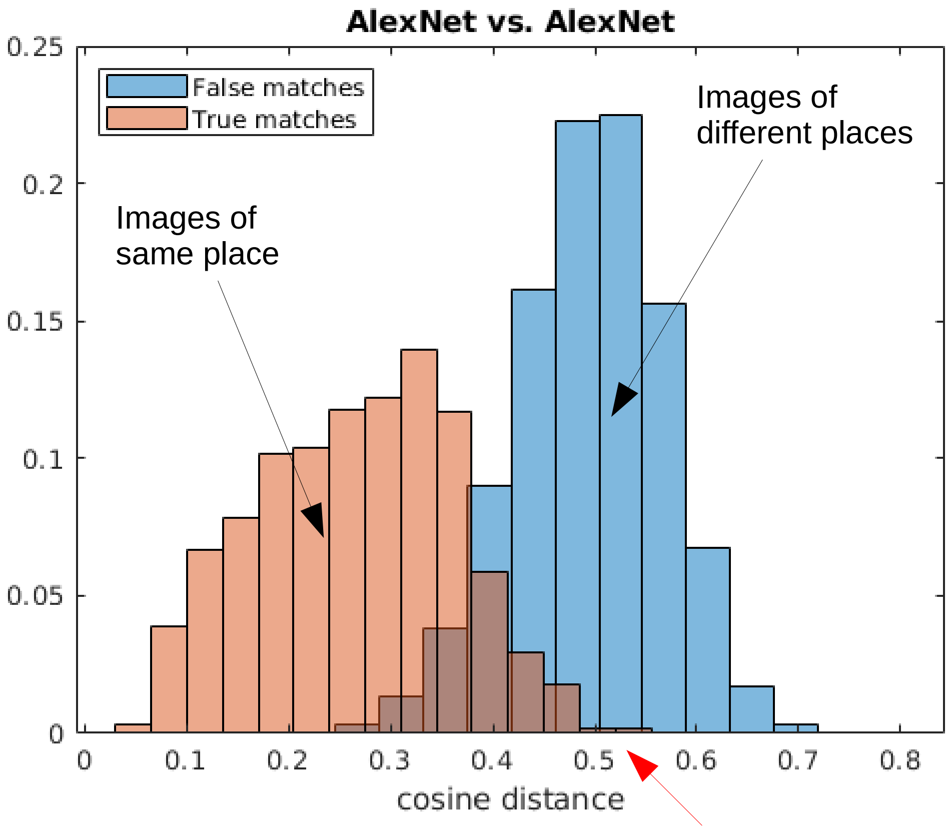


Statistical properties of CNN descriptors

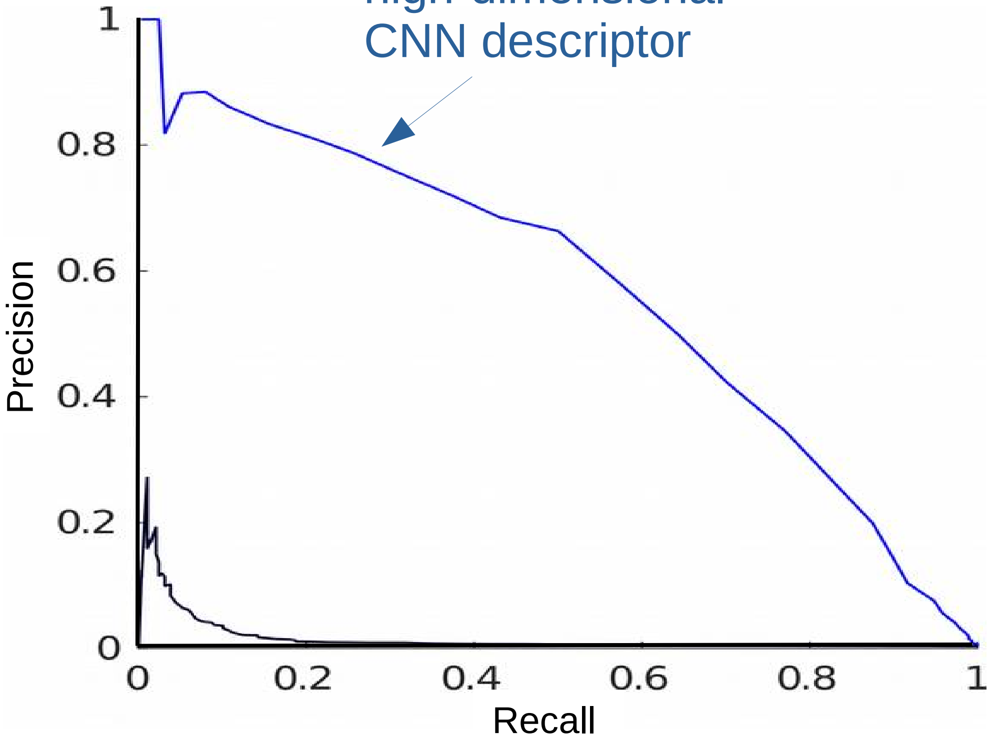




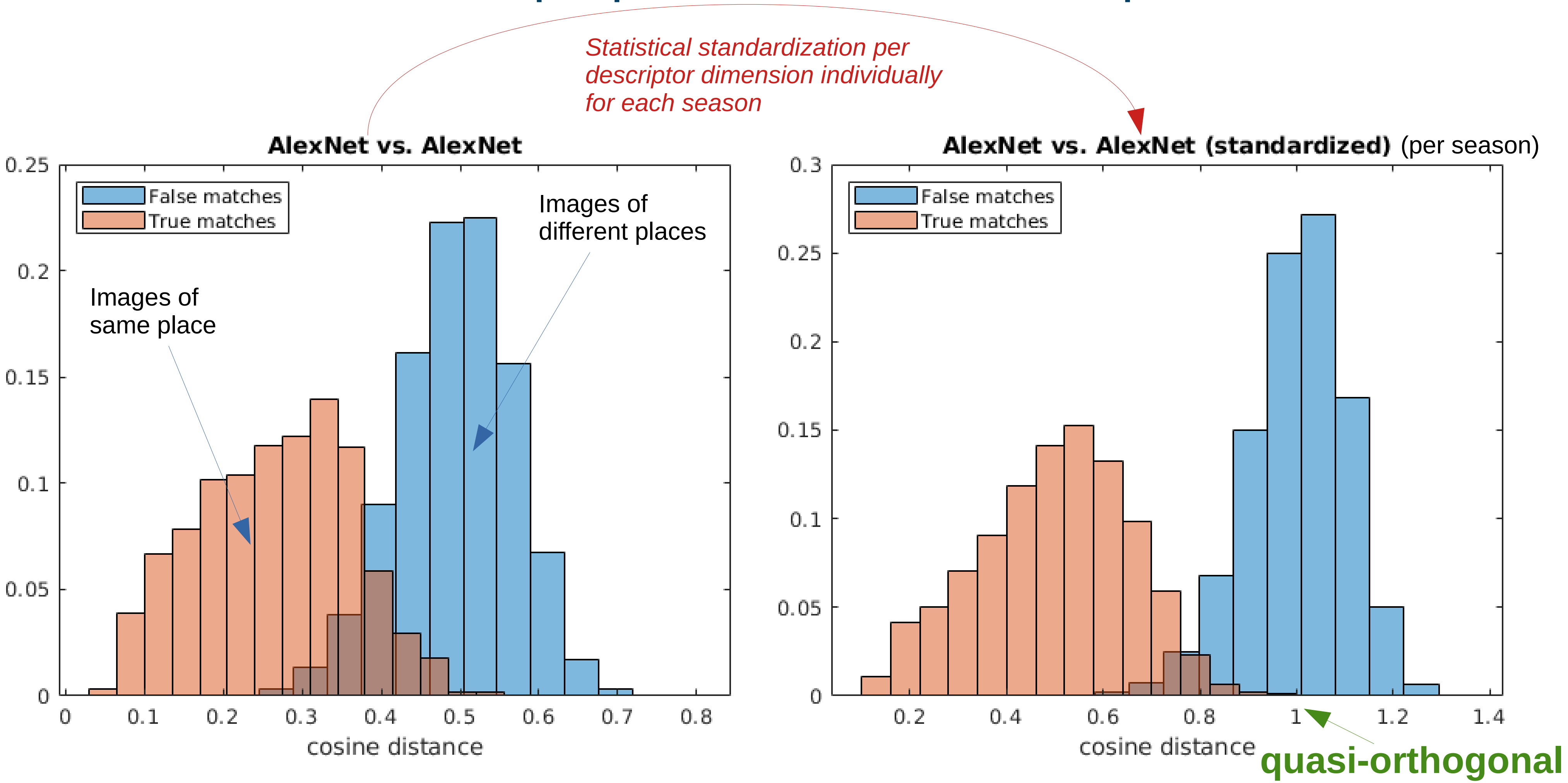
Statistical properties of CNN descriptors



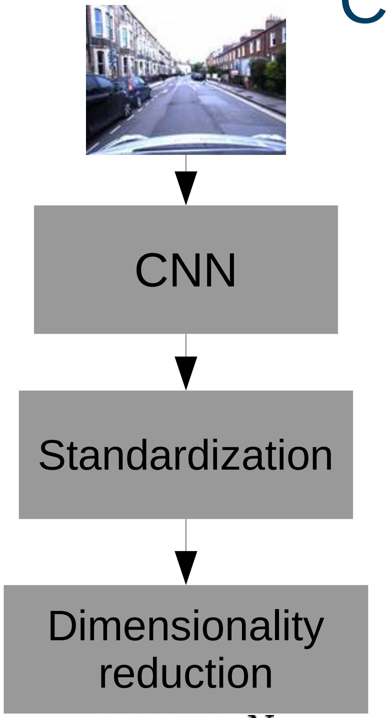
Not quasi-orthogonal!



Statistical properties of CNN descriptors



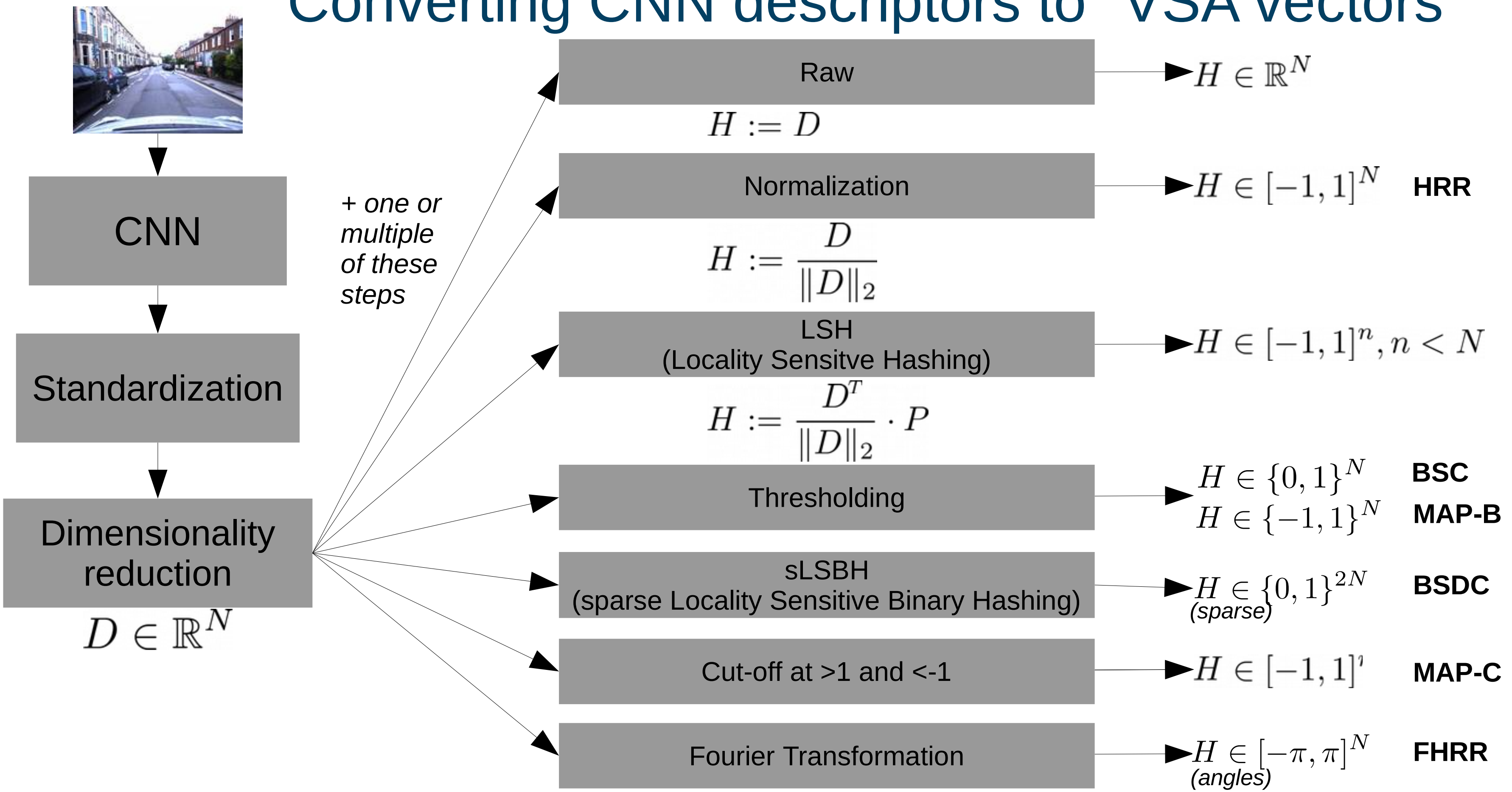
Converting CNN descriptors to “VSA vectors”



$D \in \mathbb{R}^N$

- Neubert, P., Schubert, S. & Protzel, P. (2019) A neurologically inspired sequence processing model for mobile robot place recognition. In IEEE Robotics and Automation Letters (RA-L) and Intl. Conf. on Intelligent Robots and Systems (IROS).
- Schubert, S., Neubert, P. & Protzel, P. (2020) Unsupervised Learning Methods for Visual Place Recognition in Discretely and Continuously Changing Environments. In Proc. of Intl. Conf. on Robotics and Automation (ICRA).
- Schlegel, K., Neubert, P. & Protzel, P. (2020) A comparison of Vector Symbolic Architectures. In arXiv:2001.11797

Converting CNN descriptors to “VSA vectors”



- Neubert, P., Schubert, S. & Protzel, P. (2019) A neurologically inspired sequence processing model for mobile robot place recognition. In IEEE Robotics and Automation Letters (RA-L) and Intl. Conf. on Intelligent Robots and Systems (IROS).
- Schubert, S., Neubert, P. & Protzel, P. (2020) Unsupervised Learning Methods for Visual Place Recognition in Discretely and Continuously Changing Environments. In Proc. of Intl. Conf. on Robotics and Automation (ICRA).
- Schlegel, K., Neubert, P. & Protzel, P. (2020) A comparison of Vector Symbolic Architectures. In arXiv:2001.11797

Outline

1) Introduction to VSA
   - High dimensional vector spaces and where they are used
   - Mathematical properties of high dimensional vector spaces
   - Vector Symbolic Architectures or “How to do symbolic computations using vectors spaces”

2) Available VSA implementations

3) Where do the vectors come from?

4) **Demo application**

5) Discussion

Some applications from the literature

- Addressing Catastrophic Forgetting in Deep Neural Networks Cheung, B, Terekhov, A, Chen, Y, Agrawal, Pt and Olshausen, B. (2019). Superposition of many models into one. Advances in Neural Information Processing (NeurIPS).
- Text classification Kleyko D, Rahimi A, Rachkovskij DA, Osipov E, Rabaey JM (2018) Classification and recall with binary hyperdimensional computing: tradeoffs in choice of density and mapping characteristics. IEEE Trans Neural Netw Learn Syst 29(12):5880–5898
- Fault detection Kleyko D, Osipov E, Papakonstantinou N, Vyatkin V, Mousavi A (2015) Fault detection in the hyperspace: towards intelligent automation systems. In: 2015 IEEE 13th international conference on industrial informatics (INDIN), pp 1219–1224.
- Analogy mapping Rachkovskij DA, Slipchenko SV (2012) Similarity-based retrieval with structure-sensitive sparse binary distributed representations. Comput Intell 28(1):106–129.
- Reinforcement learning Kleyko D, Osipov E, Gayler RW, Khan AI, Dyer AG (2015) Imitation of honey bees' concept learning processes using Vector Symbolic Architectures. Biol Inspired Cognit Arch 14:57–72
- Kanerva: "high dimensional computing LISP" Kanerva P (2014) Computing with 10,000-bit words. In: 52nd annual Allerton conference on communication, control,and computing (Allerton), pp 304–310
  - Synthesis of finite state automata Osipov E, Kleyko D, Legalov A (2017) Associative synthesis of finite state automata model of a controlled object with hyperdimensional computing. In: IECON 2017—43rd annual conference of the IEEE industrial electronics society, pp 3276–3281
  - Hyperdimensional stack machines Yerxa T, Anderson A, Weiss E (2018) The hyperdimensional stack machine. In: Proceedings of Cognitive Computing, Hannover, pp. 1–2
- Long-short term memory Danihelka I, Wayne G, Uria B, Kalchbrenner N, Graves A (2016) Associative long short-term memory. In: Balcan MF, Weinberger KQ (eds) Proceedings of ICML, PMLR vol 48., New York, pp 1986–1994.
- Predication-based Semantic Indexing (PSI) Widdows D, Cohen T (2015) Reasoning with vectors: a continuous model for fast robust inference. Logic J IGPL/Interest Group Pure Appl Logics 2:141–173
- Jackendoff Challanges of NLP Gayler RW (2003) Vector symbolic architectures answer Jackendoff's challenges for cognitive neuroscience. In: Proc. of ICCS/ASCS Int. Conf. on cognitive science, pp 133–138. Sydney,Australia
- N-gram statistics to recognize the language of a text Joshi A, Halseth JT, Kanerva P (2017) Language geometry using random indexing. In: de Barros JA, Coecke B, Pothos E (eds) Quantum interaction. Springer International Publishing, Cham,pp 265–274
- HTM for mobile robot place recognition Neubert, P., Schubert, S. & Protzel, P. (2019) A neurologically inspired sequence processing model for mobile robot place recognition. In IEEE Robotics and Automation Letters (RA-L) and presentation at IROS.

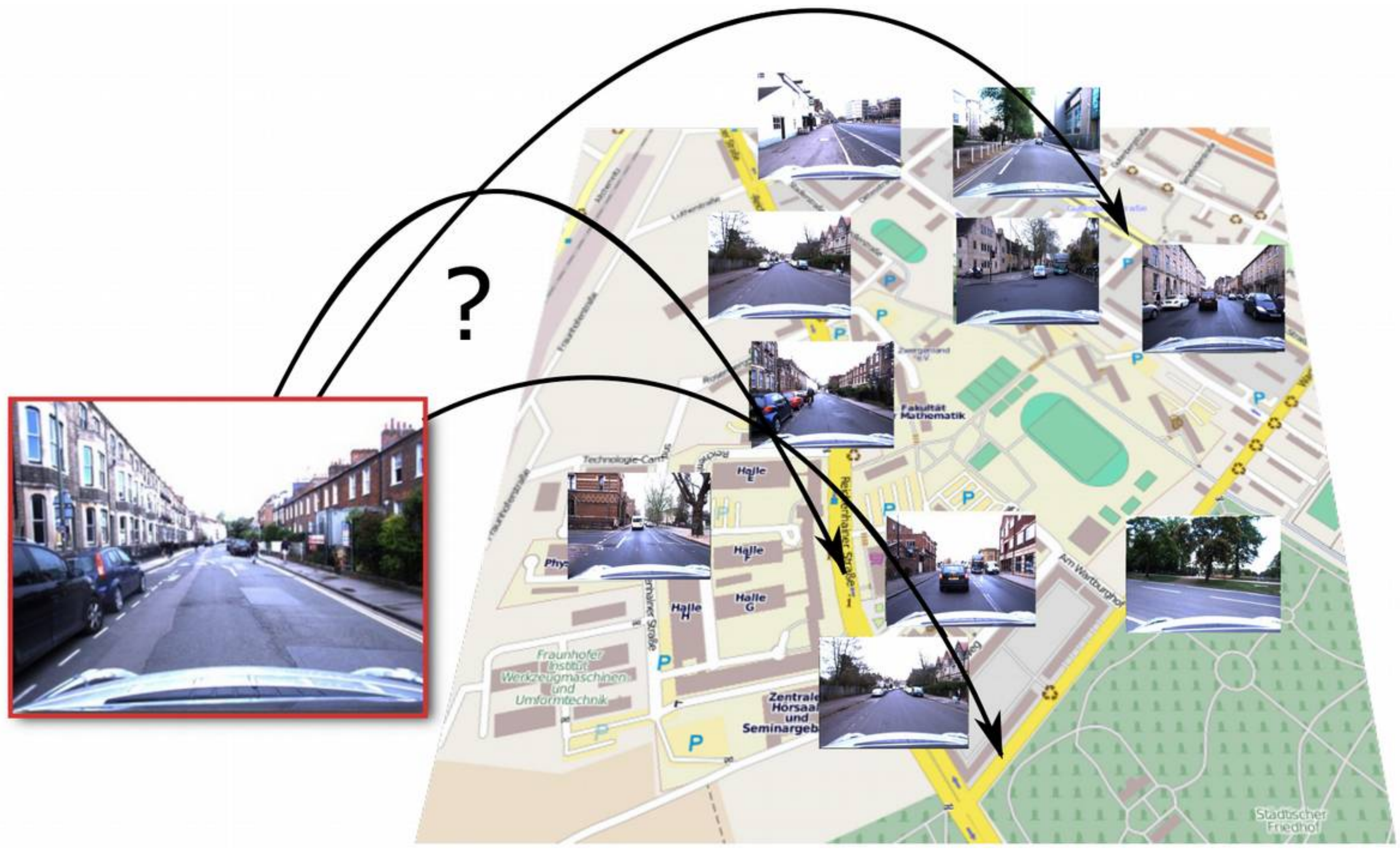
Teaser application 2: Place recognition in changing environments

The Nordland dataset – a 3000 km journey across all four seasons

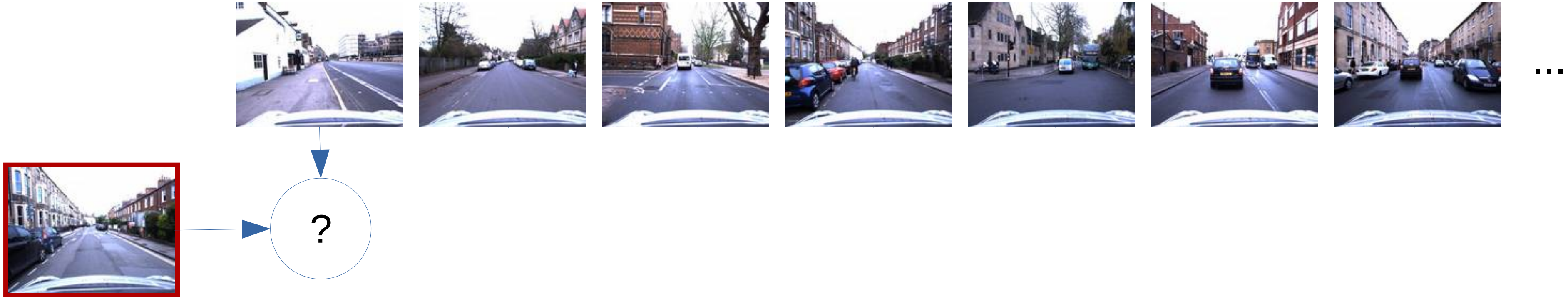
Sünderhauf, N., Neubert, P. & Protzel, P. (2013) Are We There Yet? Challenging SeqSLAM on a 3000 km Journey Across All Four Seasons. In Proc. of Workshop on Long-Term Autonomy at Int. Conf. on Rob. a. Autom. (ICRA)

http://nrkbeta.no/2013/01/15/nordlandsbanen-minute-by-minute-season-by-season/

Teaser application 2: Place recognition in changing environments

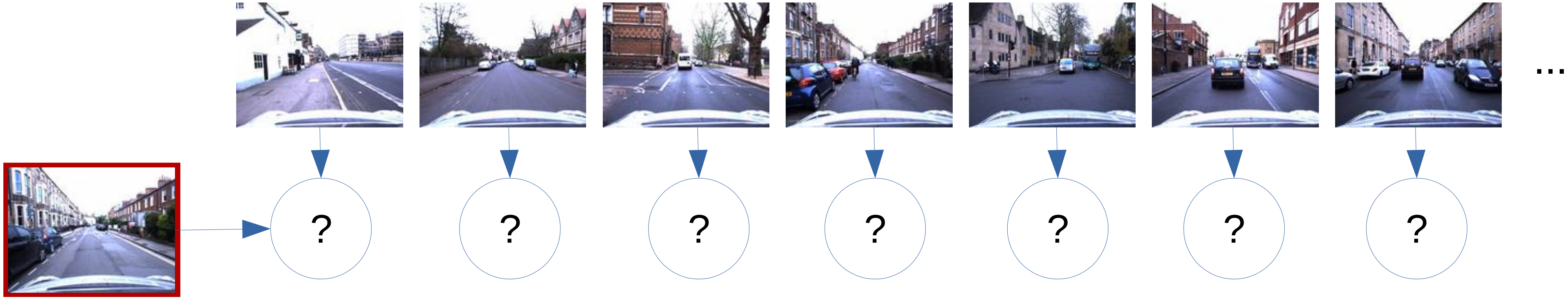


Teaser application 2: Place recognition in changing environments

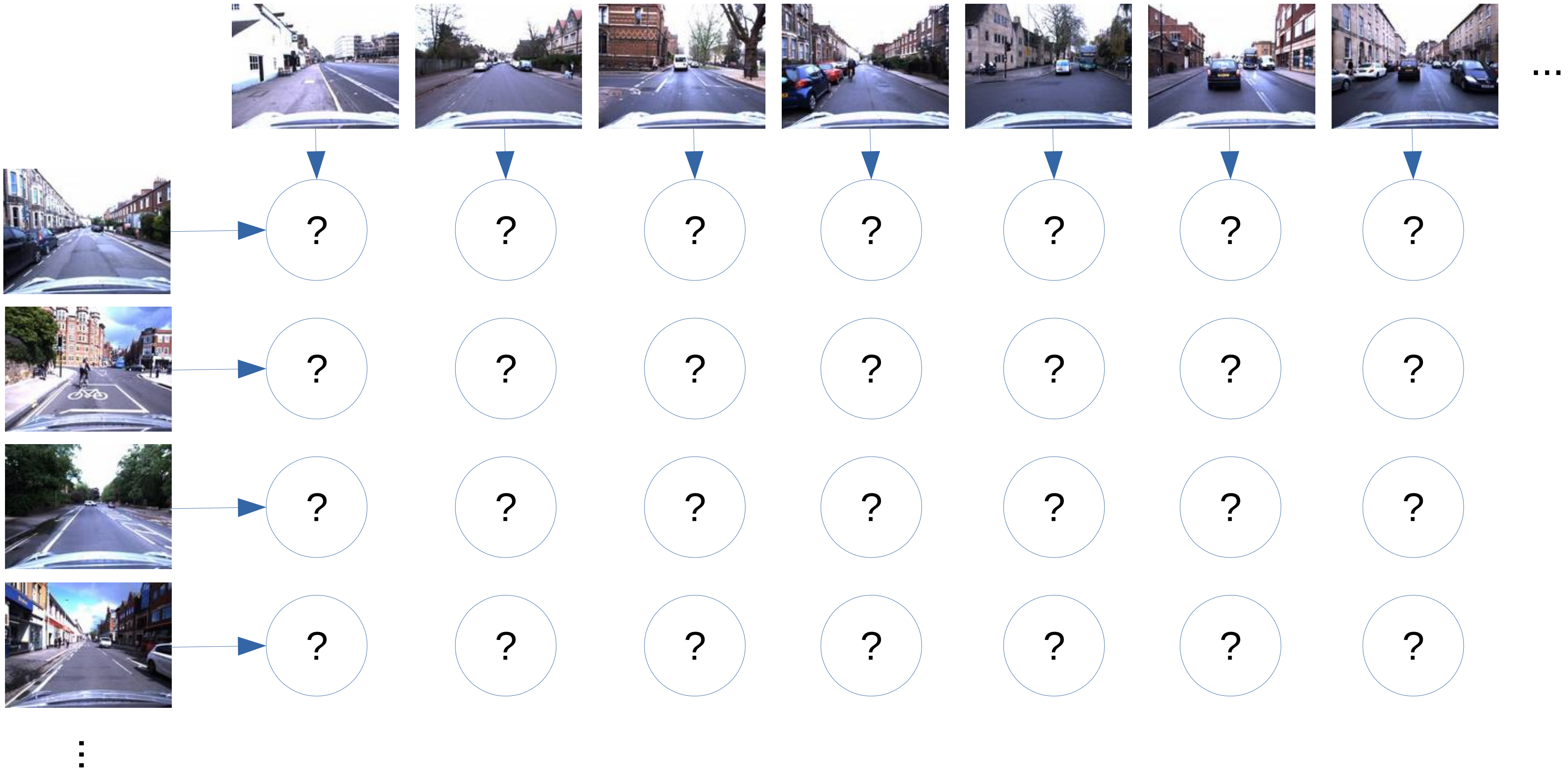


Is this the same place?

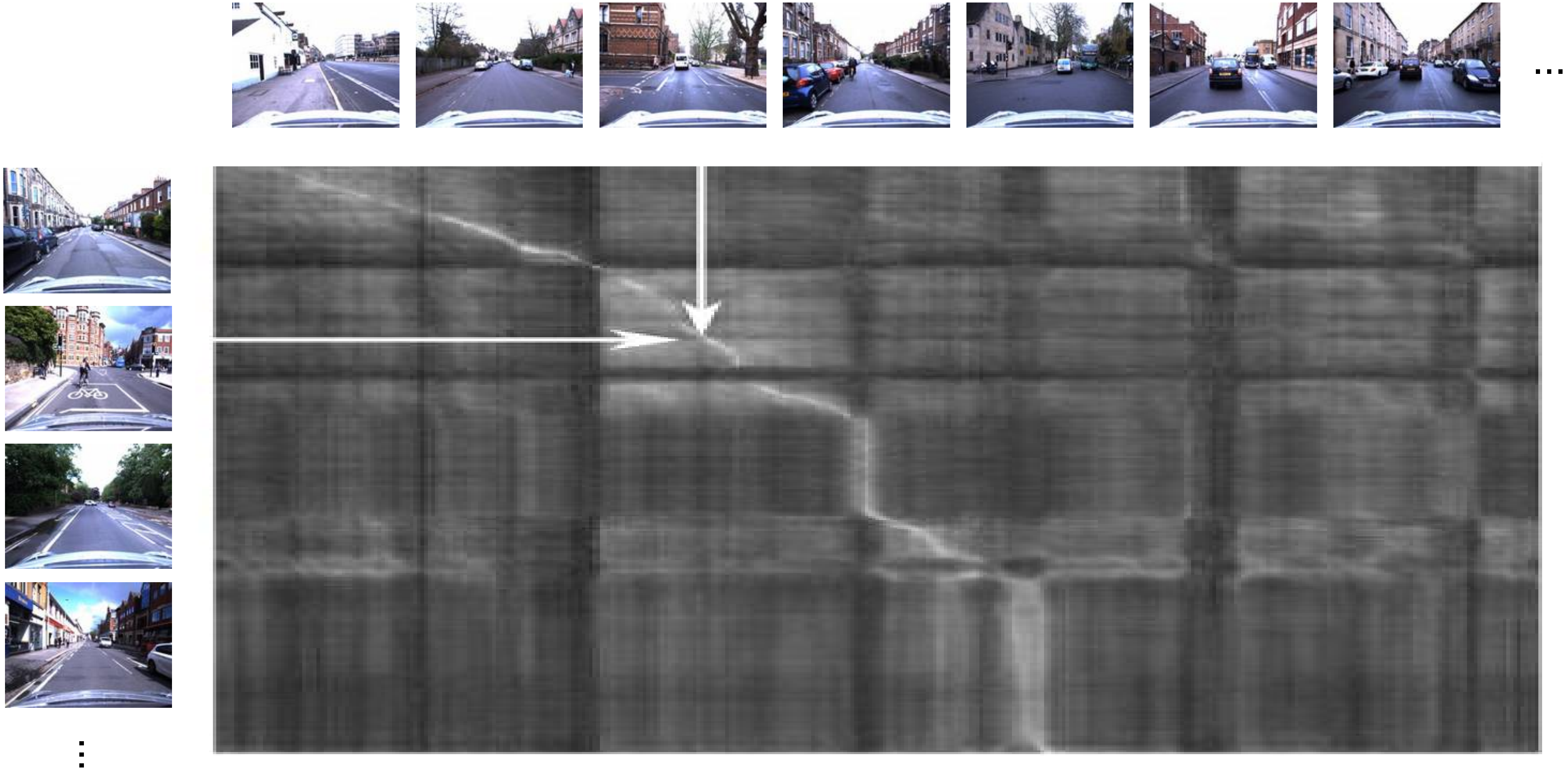
Teaser application 2: Place recognition in changing environments



Teaser application 2: Place recognition in changing environments



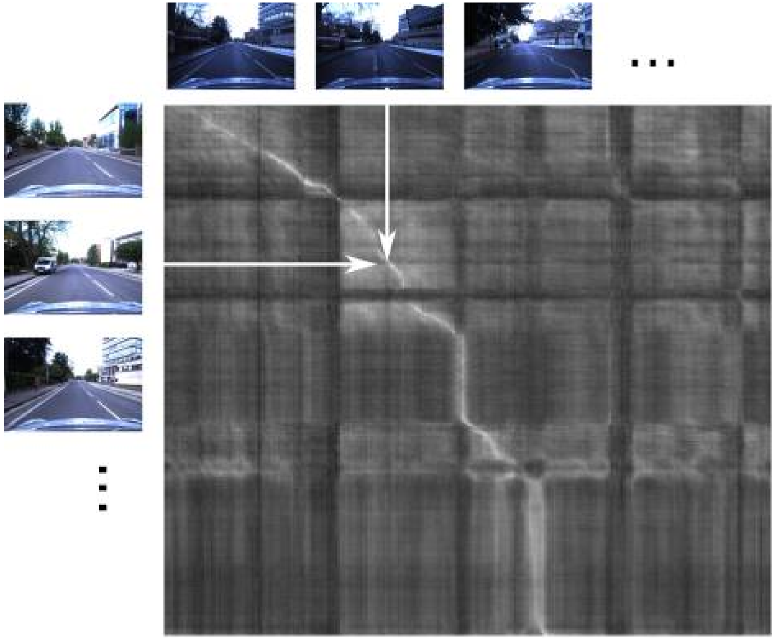
Teaser application 2: Place recognition in changing environments



Teaser application 2: Place recognition in changing environments

Deep Neural Network

$$\begin{pmatrix} 1.0 \\ 3.9 \\ -0.5 \\ . \\ . \\ . \\ 2.9 \\ -6.0 \\ 9.8 \end{pmatrix}$$

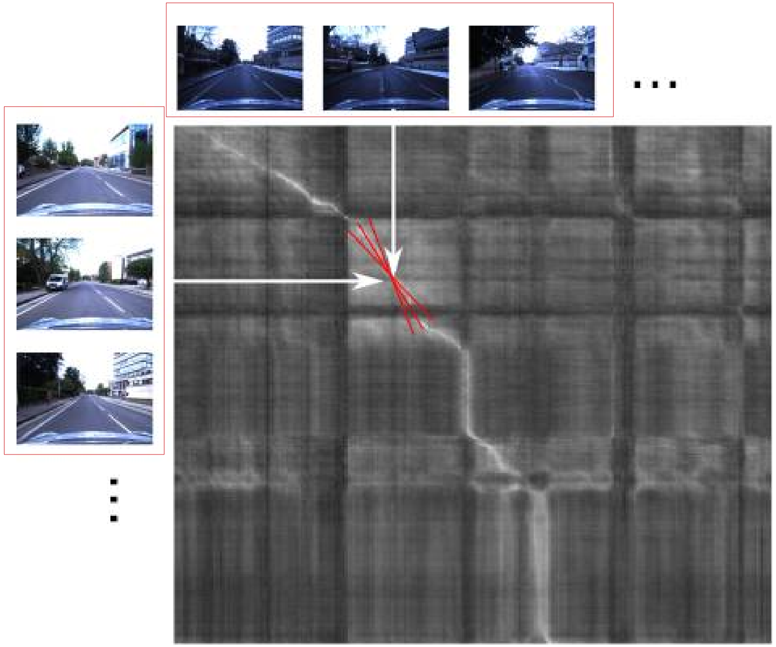


Teaser application 2: Place recognition in changing environments

Deep Neural Network

$$\begin{pmatrix}1.0\\3.9\\-0.5\\.\\.\\.\\2.9\\-6.0\\9.8\end{pmatrix}$$

Using Context:
e.g., **SeqSLAM** postprocessing



Teaser application 2: Place recognition in changing environments

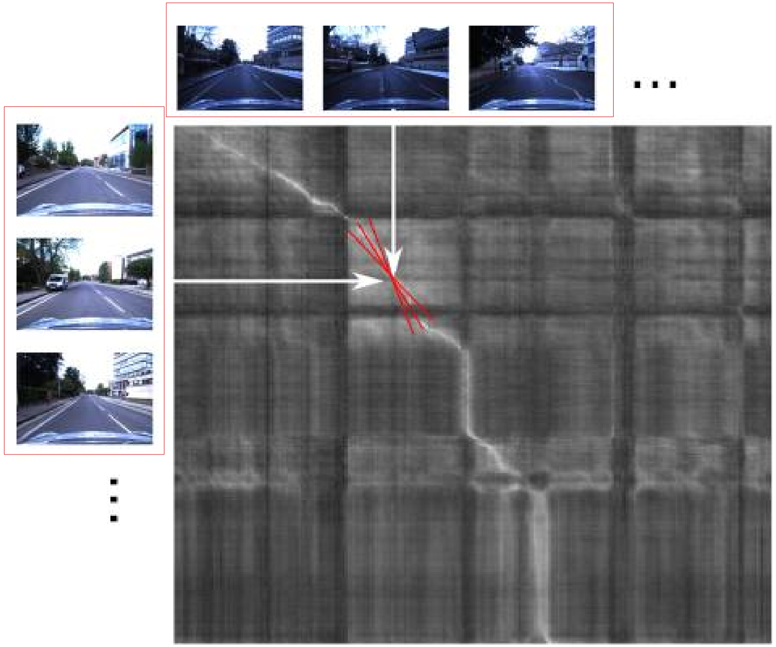
Deep Neural Network

$$\begin{pmatrix}1.0\\3.9\\-0.5\\ \vdots\\2.9\\-6.0\\9.8\end{pmatrix}$$

Some VSA magic

Using Context:

~~e.g., **SeqSLAM** postprocessing~~



Teaser application 2: Place recognition in changing environments

Simplified SeqSLAM core:

input: distance matrix D

for each summer image idx j in S

    for each winter image idx i in W

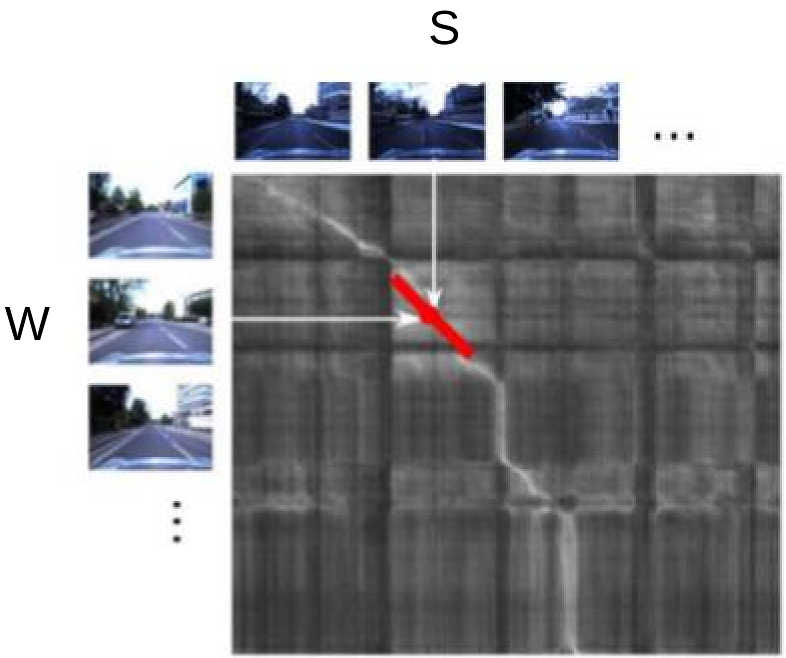
        accDist = 0

        for k=-d:1:d

            accDist = accDist + D(i+k, j+k)

        R(i,j) = accDist / (2*d+1)

output: resulting distance matrix R



Teaser application 2: Place recognition in changing environments

Simplified SeqSLAM core:

input: distance matrix D

for each summer image idx j in S

for each winter image idx i in W

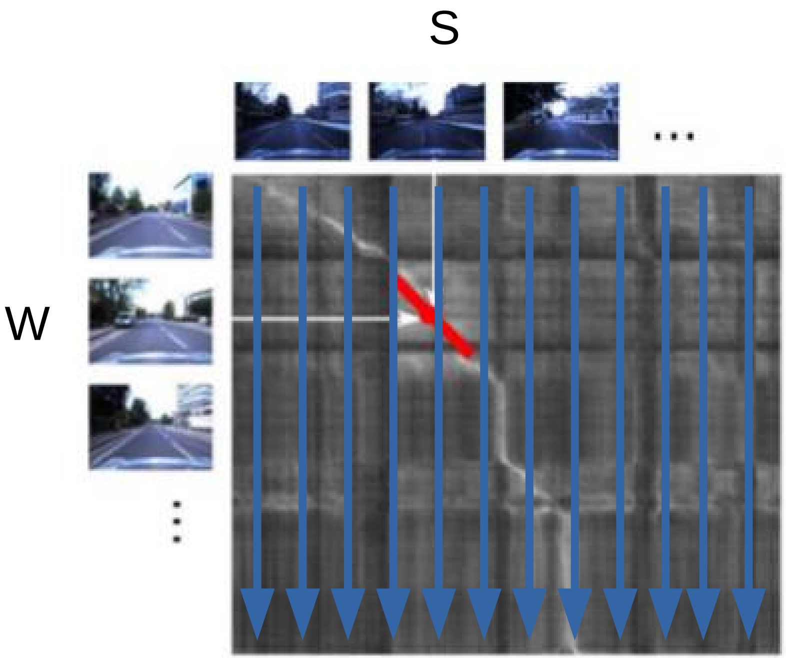
accDist = 0

for k=-d:1:d

accDist = accDist + D(i+k, j+k)

R(i,j) = accDist / (2*d+1)

output: resulting distance matrix R



Teaser application 2: Place recognition in changing environments

Simplified SeqSLAM core:

input: distance matrix D

for each summer image idx j in S

    for each winter image idx i in W

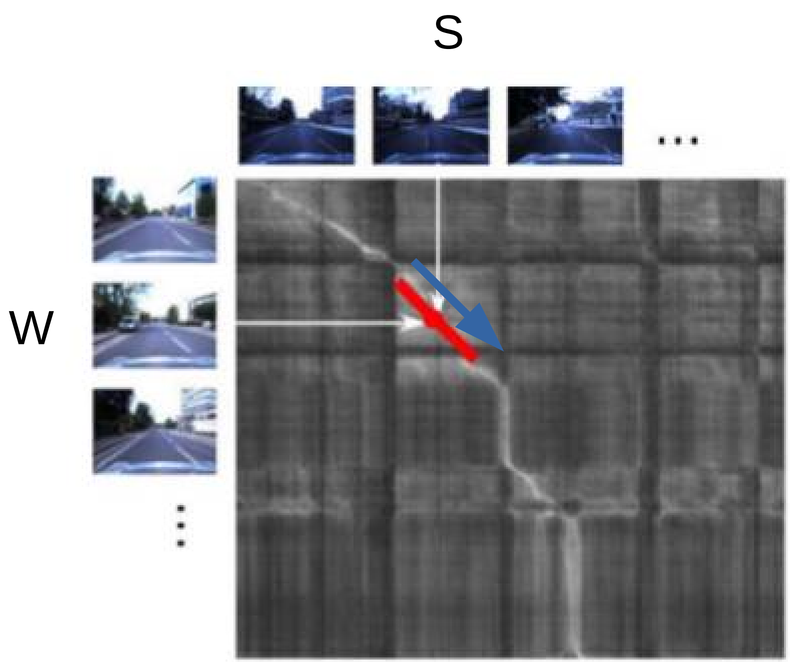
        accDist = 0

        for k=-d:1:d

            accDist = accDist + D(i+k, j+k)

        R(i,j) = accDist / (2*d+1)

output: resulting distance matrix R



Teaser application 2: Place recognition in changing environments

VSA appoach

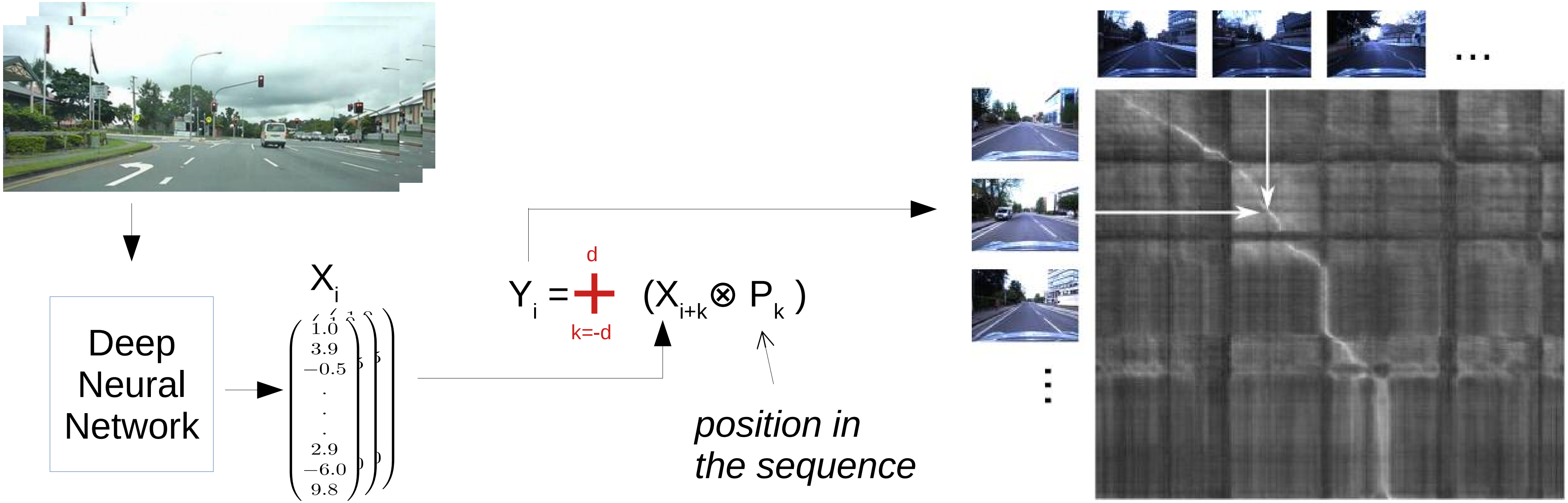
- Replace each image vector with a vector that represents the whole sequence
- Use this vector for the direct **pairwise** comparison

Neubert, P., Schubert, S. & Protzel, P. (2019) An Introduction to High Dimensional Computing for Robotics. German Journal of Artificial Intelligence

Teaser application 2: Place recognition in changing environments

VSA appoach

- Replace each image vector with a vector that represents the whole sequence
- Use this vector for the direct **pairwise** comparison



Neubert, P., Schubert, S. & Protzel, P. (2019) An Introduction to High Dimensional Computing for Robotics. German Journal of Artificial Intelligence

Teaser application 2: Place recognition in changing environments

position in the sequence

$$Y_i = \bigoplus_{k=-d}^{d} (X_{i+k} \otimes P_k)$$

Why does this work?

e.g. comparing two 2-element sequences $A = (X_{a_1} X_{a_2})$
$B = (X_{b_1} X_{b_2})$

Neubert, P., Schubert, S. & Protzel, P. (2019) An Introduction to High Dimensional Computing for Robotics. German Journal of Artificial Intelligence

Teaser application 2: Place recognition in changing environments

position in the sequence

$$Y_i = \mathop{\Large +}_{k=-d}^{d} \left(X_{i+k} \otimes P_k\right)$$

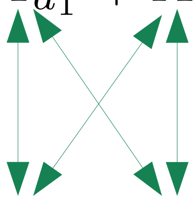
Why does this work?

e.g. comparing two 2-element sequences $A = (X_{a_1} X_{a_2})$

$B = (X_{b_1} X_{b_2})$

Without binding to position

$$Y_A = X_{a_1} + X_{a_2}$$



$$Y_B = X_{b_1} + X_{b_2}$$

Neubert, P., Schubert, S. & Protzel, P. (2019) An Introduction to High Dimensional Computing for Robotics. German Journal of Artificial Intelligence

Teaser application 2: Place recognition in changing environments

position in the sequence

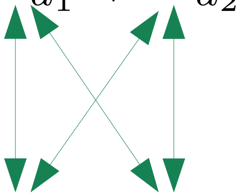
$$Y_i = \mathop{\Large +}_{k=-d}^{d} (X_{i+k} \otimes P_k)$$

Why does this work?

e.g. comparing two 2-element sequences $A = (X_{a_1} X_{a_2})$
$B = (X_{b_1} X_{b_2})$

Without binding to position

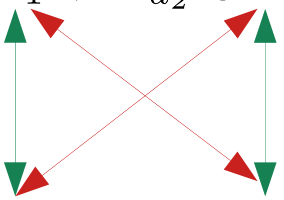
$$Y_A = X_{a_1} + X_{a_2}$$



$$Y_B = X_{b_1} + X_{b_2}$$

With binding to position

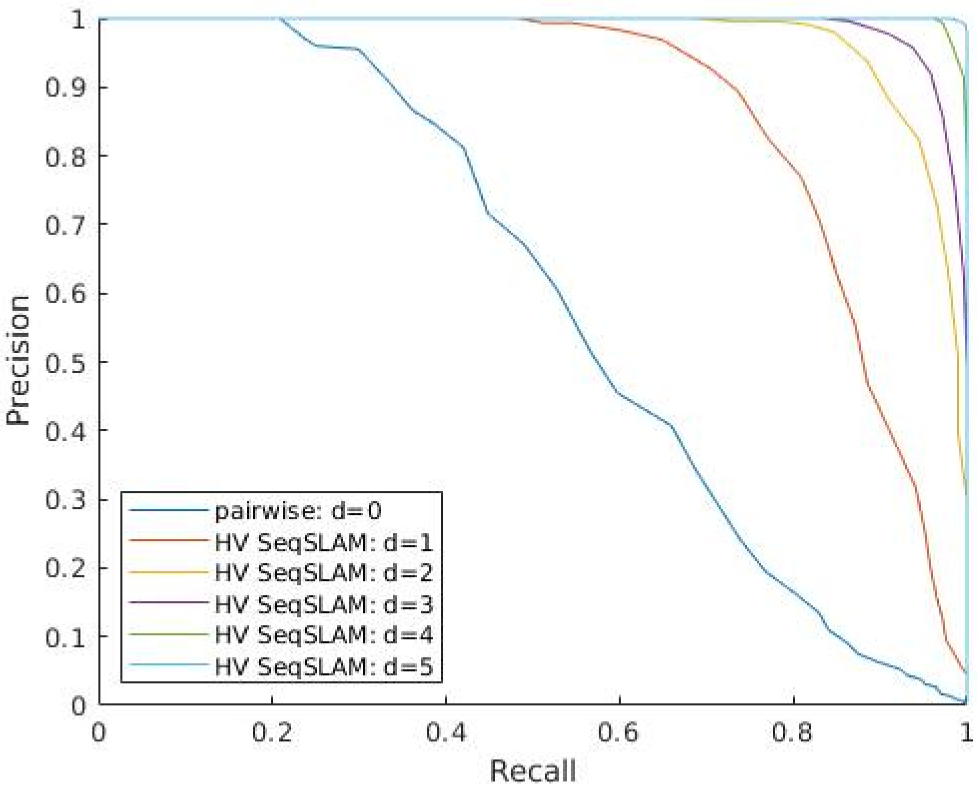
$$Y_A = X_{a_1} \otimes P_1 + X_{a_2} \otimes P_2$$



$$Y_B = X_{b_1} \otimes P_1 + X_{b_2} \otimes P_2$$

Neubert, P., Schubert, S. & Protzel, P. (2019) An Introduction to High Dimensional Computing for Robotics. German Journal of Artificial Intelligence

Hands-on



demo_hvSeqSLAM.m

Neubert, P., Schubert, S. & Protzel, P. (2019) An Introduction to High Dimensional Computing for Robotics. German Journal of Artificial Intelligence

Outline

1) Introduction to VSA
   - High dimensional vector spaces and where they are used
   - Mathematical properties of high dimensional vector spaces
   - Vector Symbolic Architectures or “How to do symbolic computations using vectors spaces”

2) Available VSA implementations

3) Where do the vectors come from?

4) Demo application

5) **Discussion**

Limitations, discussion and open questions

There is a lack of a clear definition of Vector Symbolic Architectures (i.e., in terms of axioms and theorems)

Limitations, discussion and open questions

We wish for better insights in trade-offs and capabilities

Limitations, discussion and open questions

Encoding real world data

Limitations, discussion and open questions

Easier access for non-mathematicians would be nice, as well as a structured way how to solve tasks (e.g. design patterns).

Limitations, discussion and open questions

What has to be manually designed, what can be learned?

Limitations, discussion and open questions

There are plenty of connections to other fields...

Limitations, discussion and open questions

?

Self-Test

1) What is a typical size of VSA vector space?
2) In your own words: What are the four presented properties of high-dimensional vector spaces?

The Trivial
The Bad
The Surprising
The Good

https://www.tu-chemnitz.de/etit/proaut/vsa_ecai20

3) Is the result of **bundling** A and B similar to A?
4) Is the result of **binding** A and B similar to A?
5) How many lines of code are required to implement a VSA (rough estimate)?
6) With the taxonomy of binding operators in mind, what is the requirement for the presented VSA approach to the “The Dollar of Mexico” example?
7) The place recognition demo uses sequences of CNN descriptors of camera images. Why is the binding operator important for the presented solution?

Thank you for your attention!

Peer Neubert

Kenny Schlegel

Stefan Schubert

firstname.lastname@etit.tu-chemnitz.de

https://www.tu-chemnitz.de/etit/proaut/vsa_ecai20

TECHNISCHE UNIVERSITÄT CHEMNITZ

Neubert, P., Schubert, S., Protzel, P. (2019). An Introduction to Hyperdimensional Computing for Robotics. German Journal of Artificial Intelligence Special Issue: Reintegrating Artificial Intelligence and Robotics, Springer

Schlegel, K., Neubert, P., Protzel, P. (2020) A comparison of Vector Symbolic Architectures. CoRR, abs/2001.11797