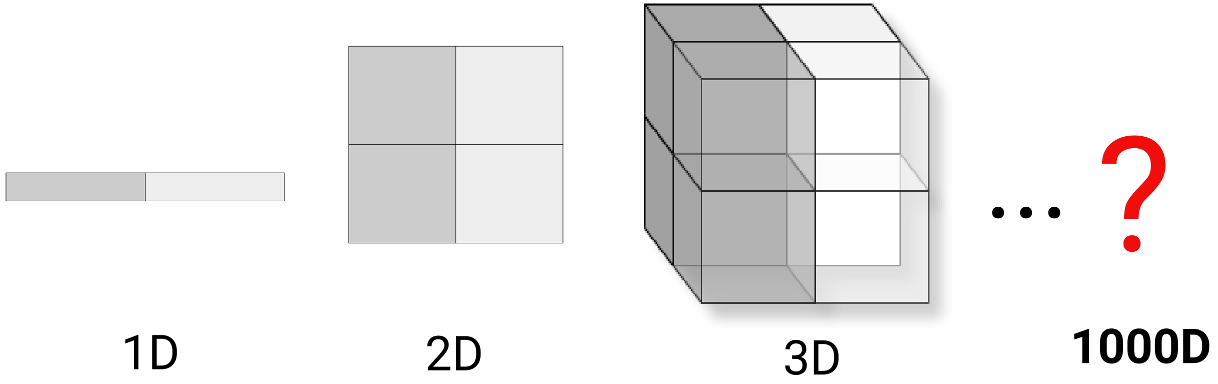
An Introduction to Hyperdimensional Computing and Its Applications

Kenny Schlegel

Chemnitz University of Technology

The talk is about...

1. What?

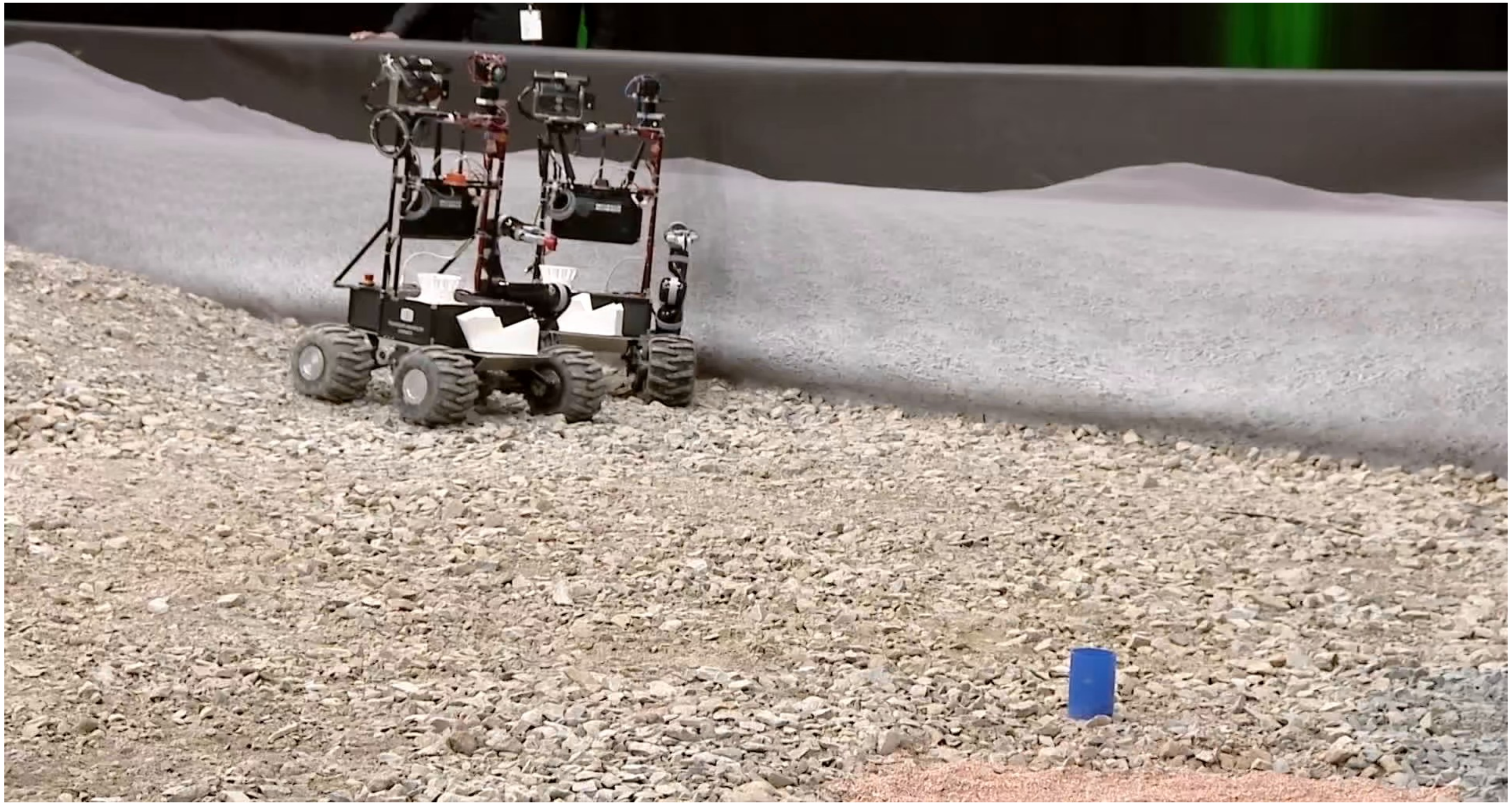


Hyperdimensional Computing

2. So What?

Example Applications

Our Professorships' topics

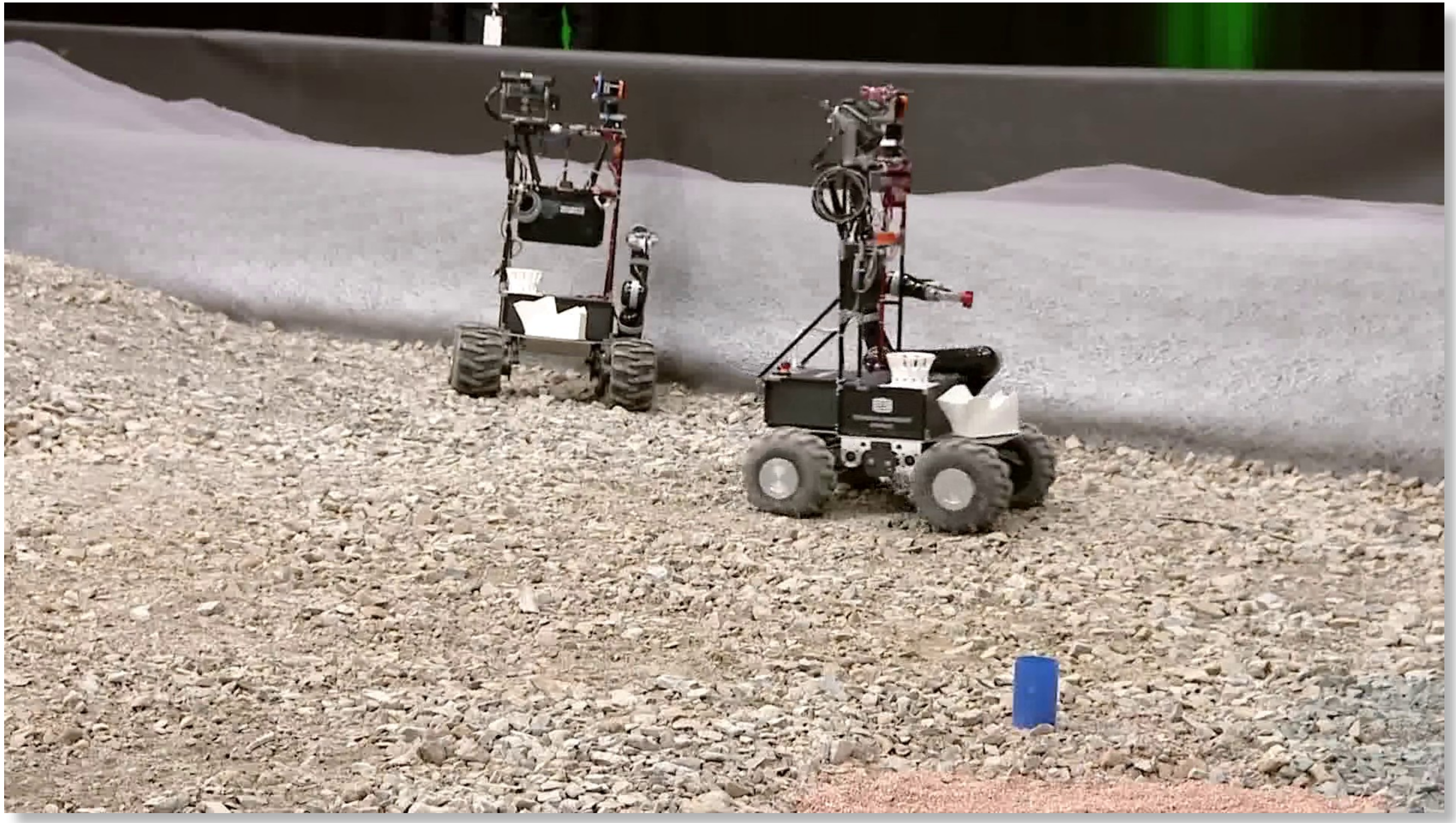


Our Professorships' topics

Robust Sensor Fusion

Multi object Tracking

Localization With GNSS



Navigation

Visual Place Recognition

Time Series Analysis

My background

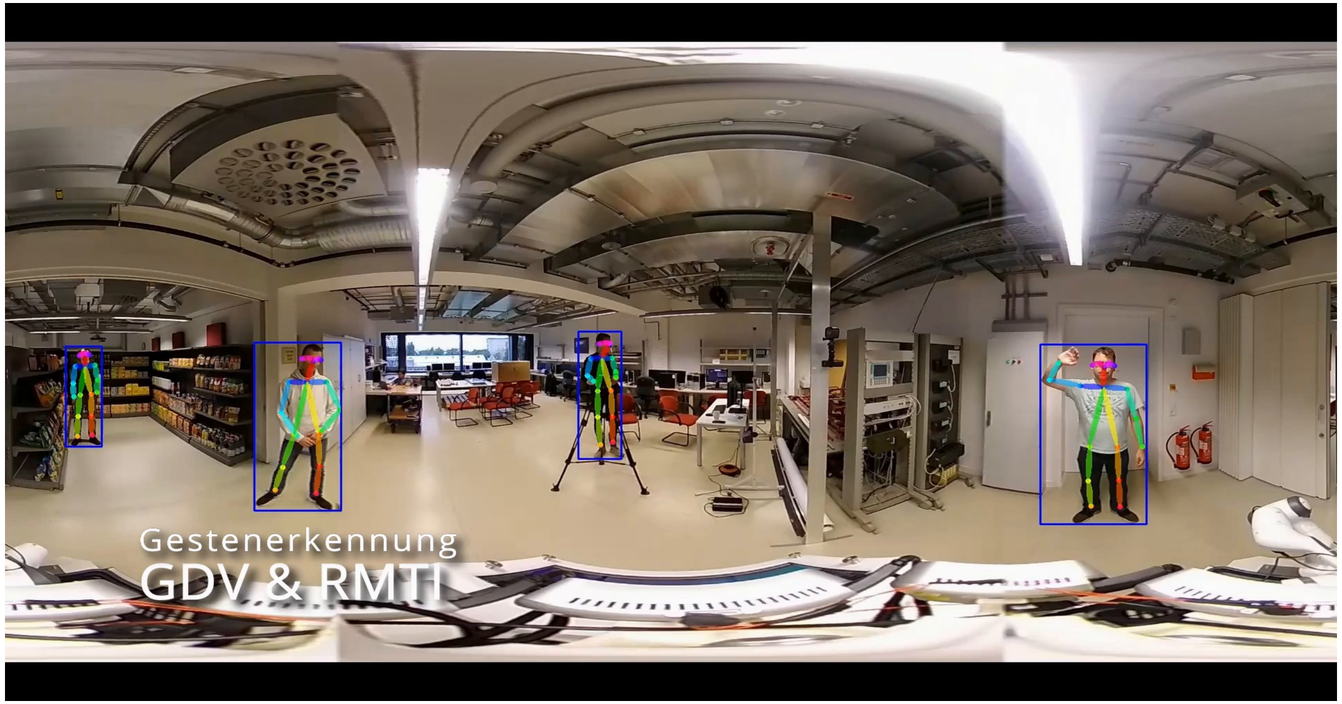
Worked in a joint research project „interaction strategies for a **shopping assistant robot**“

Our tasks:

- Environment representation
- Localization
- Navigation in dynamic environments

My background

Worked in a joint research project „interaction strategies for a **shopping assistant robot**“



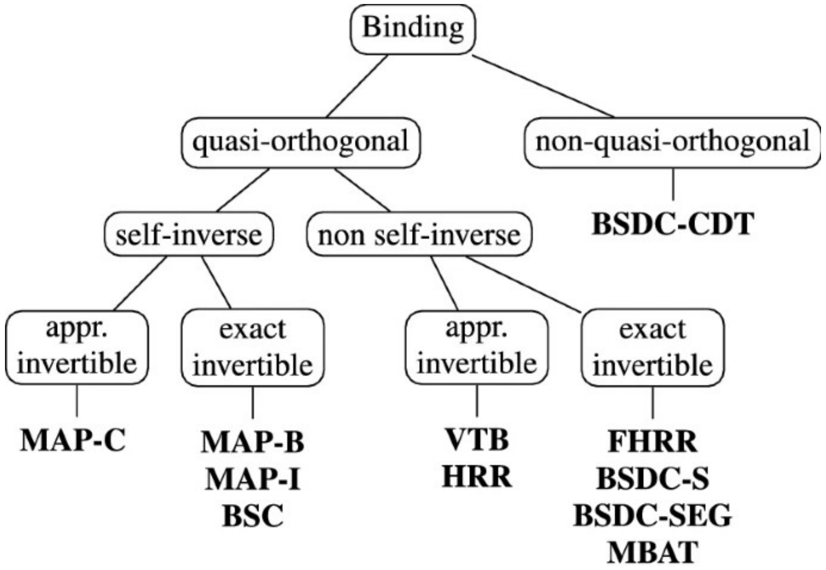
Our tasks:

- Environment representation
- Localization
- Navigation in dynamic environments

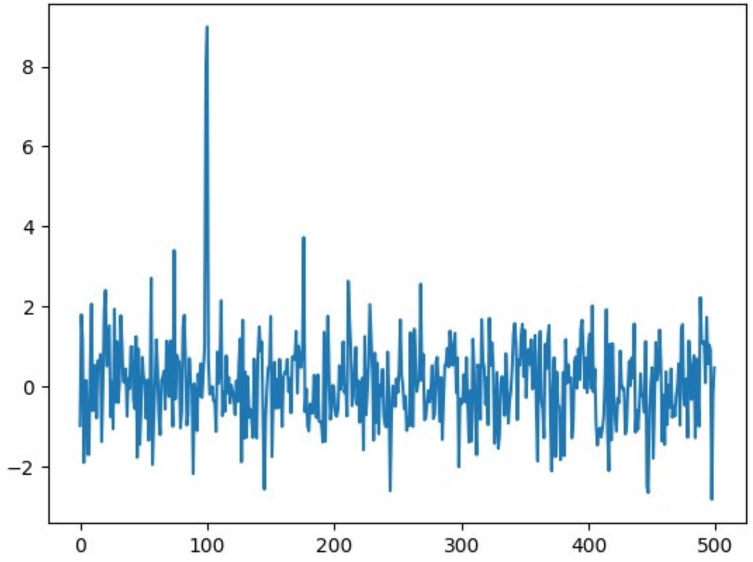
Current research

Exploring HDC/VSA for Applications in Computer Vision and Signal Processing

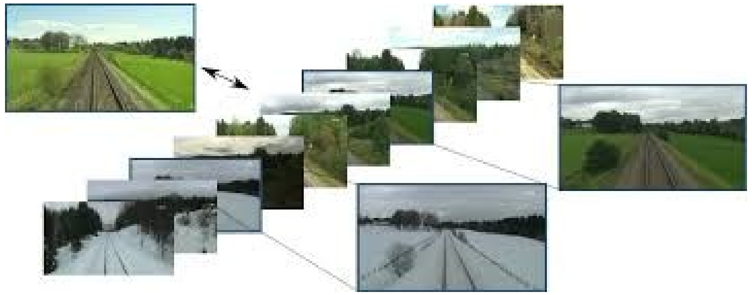
Comparison of VSAs



Time Series Analysis



Visual Place Recognition



Mainly done by my colleagues Peer Neubert and Stefan Schubert

Work on VSA / HDC and how it can be used in different tasks

Peer Neubert

Stefan Schubert

Peter Protzel

Artificial Intelligence Review
https://doi.org/10.1007/s10462-021-10110-3

A comparison of vector symbolic architectures

Kenny Schlegel[1] · Peer Neubert[1] · Peter Protzel[1]

Abstract
Vector Symbolic Architectures combine a high-dimensional vector space with a set of carefully designed operators in order to perform symbolic computations with large numerical vectors. Major goals are the exploitation of their representational power and ability to deal with fuzziness and ambiguity. Over the past years, several VSA implementations have been proposed. The available implementations differ in the underlying vector space and the particular implementations of the VSA operators. This paper provides an overview of eleven available VSA implementations and discusses their commonalities and differences in the underlying vector space and operators. We create a taxonomy of available binding operations and show an important ramification for non self-inverse binding operations using an example from analogical reasoning. A main contribution is the experimental comparison of the available implementations in order to evaluate (1) the capacity of bundles, (2) the approximation quality of non-exact unbinding operations, (3) the influence of combining binding and bundling operations on the query answering performance, and (4) the performance on two example applications: visual place- and language-recognition. We expect this comparison and systematization to be relevant for development of VSAs, and to support the selection of an appropriate VSA for a particular task. The implementations are available.

HDC-MiniROCKET: Explicit Time Encoding in Time Series Classification with Hyperdimensional Computing

Kenny Schlegel, Peer Neubert and Peter Protzel
Chemnitz University of Technology
Chemnitz, Germany
{kenny.schlegel, peer.neubert, peter.protzel}@etit.tu-chemnitz.de

***Abstract*—Classification of time series data is an important task for many application domains. One of the best existing methods for this task, in terms of accuracy and computation time, is MiniROCKET. In this work, we extend this approach to provide better global temporal encodings using hyperdimensional computing (HDC) mechanisms. HDC (also known as Vector Symbolic Architectures, VSA) is a general method to explicitly represent and process information in high-dimensional vectors. It has previously been used successfully in combination with deep neural networks and other signal processing algorithms. We argue that the internal high-dimensional representation of MiniROCKET is well suited to be complemented by the algebra of HDC. This leads to a more general formulation, HDC-MiniROCKET, where the original algorithm is only a special case. We will discuss and demonstrate that HDC-MiniROCKET can systematically overcome catastrophic failures of MiniROCKET on simple synthetic datasets. These results are confirmed by experiments on the 128 datasets from the UCR time series classification benchmark. The extension with HDC can achieve considerably better results on datasets with high temporal dependence at about the same computational effort for inference.**

***Index Terms*—time series classification, HDC, VSA, hyperdimensional computing**

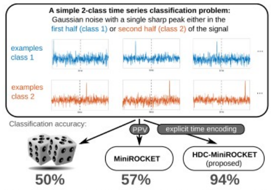


Fig. 1: MiniROCKET is a fast state-of-the-art approach for time series classification. However, it is easy to create simple datasets where its performance is similar to random guessing. The proposed HDC-MiniROCKET uses explicit time encoding to prevent this failure at almost the same computational costs.

Robotics: Science and Systems 2021
Held Virtually, July 12–16, 2021

Vector Semantic Representations as Descriptors for Visual Place Recognition

Peer Neubert, Stefan Schubert, Kenny Schlegel and Peter Protzel
Chemnitz University of Technology, Germany
{peer.neubert, stefan.schubert, kenny.schlegel, peter.protzel}@etit.tu-chemnitz.de

***Abstract*—Place recognition is the task of recognizing the current scene from a database of known places. The currently dominant algorithmic paradigm is to use (deep learning based) holistic feature vectors to describe each place and use fast vector query methods to find matchings. We propose a novel type of image descriptor, Vector Semantic Representations (VSR), that encodes the spatial semantic layout from a semantic segmentation together with appearance properties in a, for example, 4,096 dimensional vector for place recognition. We leverage operations from the established class of Vector Symbolic Architectures to combine symbolic (e.g. class label) and numeric (e.g. feature map response) information in a common vector representation. We evaluate the proposed semantic descriptor on 13 standard mobile robotic place recognition datasets and compare to six descriptors from the literature. VSR is on par with the best compared descriptor (NetVLAD) in terms of mean average precision and superior in terms of recall and worst-case average precision. This makes the approach particularly interesting for candidate selection. For a more detailed investigation, we discuss and evaluate recall integrity as additional criterion. Further, we demonstrate that the semantic descriptor is particularly well suited for combination with existing appearance descriptors indicating that semantics provide complementary information for image matching.**

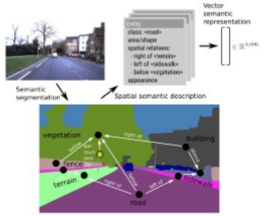


Fig. 1. A Vector Semantic Representation (VSR) is a single high-dimensional vector that combines information of semantic entities, their spatial layout, and appearance. We describe how a semantic segmentation can be used to create the entities and how operations from Vector Symbolic Architecture (VSA) can be used to encode this mixed symbolic-numeric information in a single vector that can serve as descriptor for place recognition.

KI - Künstliche Intelligenz
https://doi.org/10.1007/s13218-019-00623-z

TECHNICAL CONTRIBUTION

An Introduction to Hyperdimensional Computing for Robotics

Peer Neubert[1] · Stefan Schubert[1] · Peter Protzel[1]

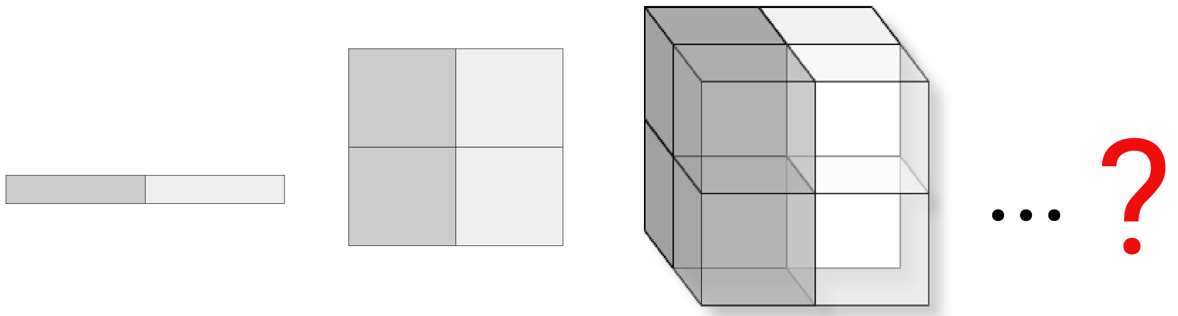
Received: 15 December 2018 / Accepted: 11 September 2019

Abstract
Hyperdimensional computing combines very high-dimensional vector spaces (e.g. 10,000 dimensional) with a set of carefully designed operators to perform symbolic computations with large numerical vectors. The goal is to exploit their representational power and noise robustness for a broad range of computational tasks. Although there are surprising and impressive results in the literature, the application to practical problems in the area of robotics is so far very limited. In this work, we aim at providing an easy to access introduction to the underlying mathematical concepts and describe the existing computational implementations in form of vector symbolic architectures (VSAs). This is accompanied by references to existing applications of VSAs in the literature. To bridge the gap to practical applications, we describe and experimentally demonstrate the application of VSAs to three different robotic tasks: viewpoint invariant object recognition, place recognition and learning of simple reactive behaviors. The paper closes with a discussion of current limitations and open questions.

Keywords Hyperdimensional computing · Vector symbolic architectures · Robotics

The talk is about...

1. What?



1D 2D 3D 1000D

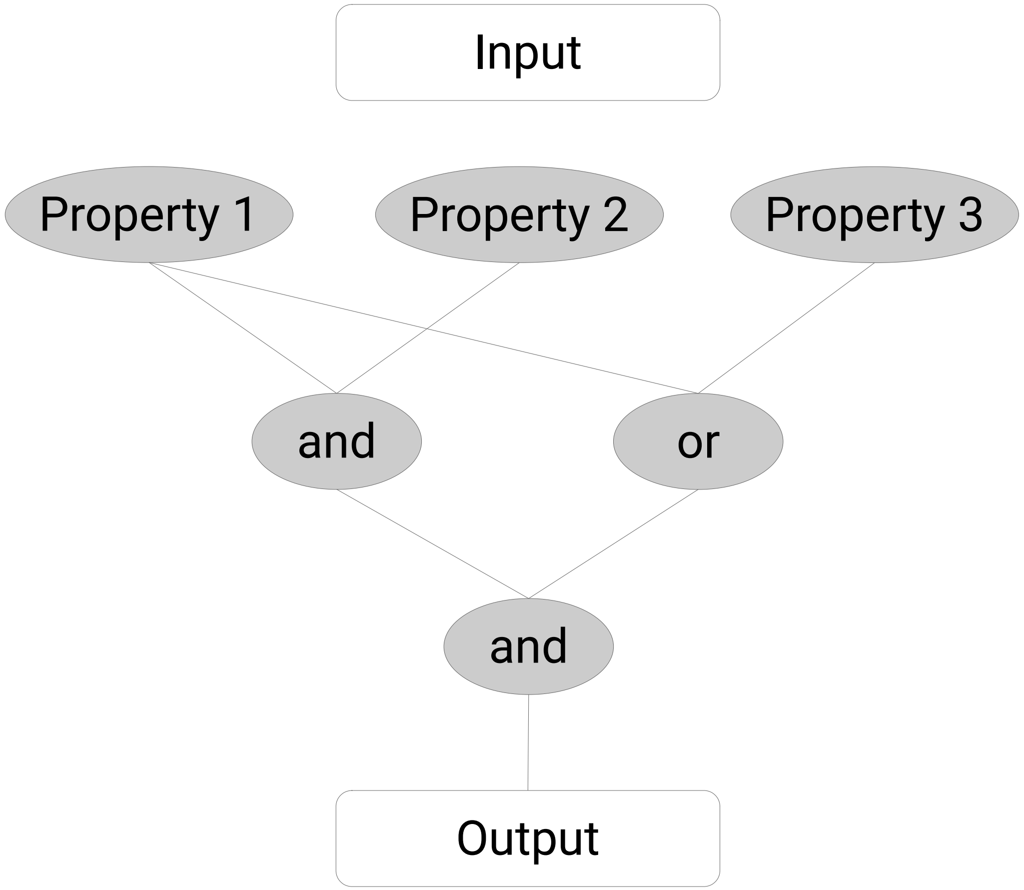
Hyperdimensional Computing

2. So What?

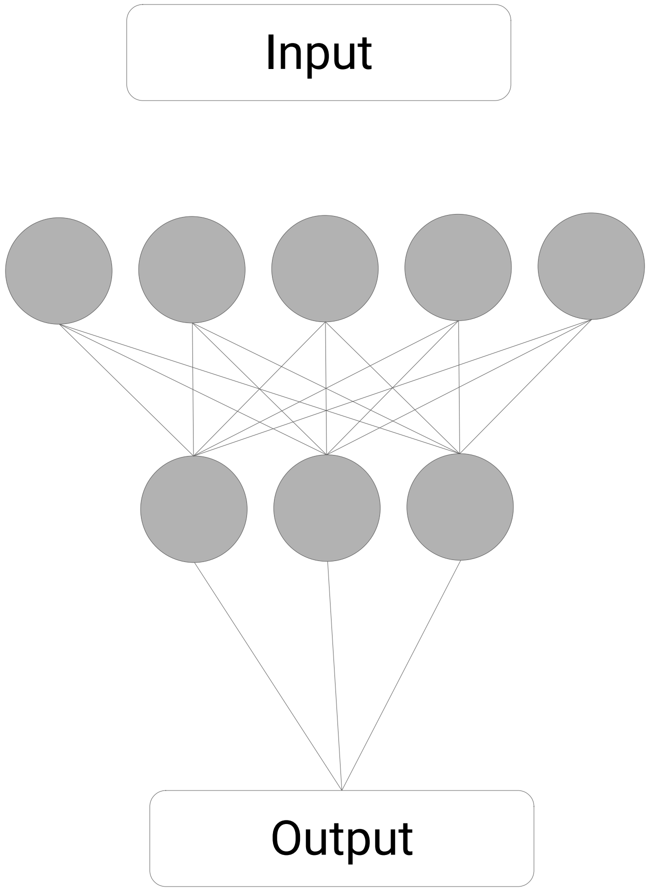
Example Applications

Motivation from the view of AI

Symbolic (Traditional AI)



Sub-symbolic (Neural Networks)



Motivation from the view of AI

	Symbolic (Traditional AI)	Sub-symbolic (Neural Networks)	HDC / VSA
Distributed Representation	X	✓	✓
Robust	X	?	✓
Learn from Data	X	✓	–
Symbolic computations	✓	X	✓
Interpretable	✓	X	✓

Table based on the HDC-Course, 2021, UC Berkeley, https://www.hd-computing.com/course-computing-with-high-dimensional-vectors

Seminal Literature and Researcher

Tony Plate

Pentti Kanerva

Ross Gayler

[1] T. A. Plate, "Holographic Reduced Representations," IEEE Trans. Neural Networks, 1995.
[2] P. Kanerva, "Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors," Cognit. Comput., 2009.
[3] R. W. Gayler, "Vector Symbolic Architectures answer Jackendoff's challenges for cognitive neuroscience," arXiv:cs/0412059 2004

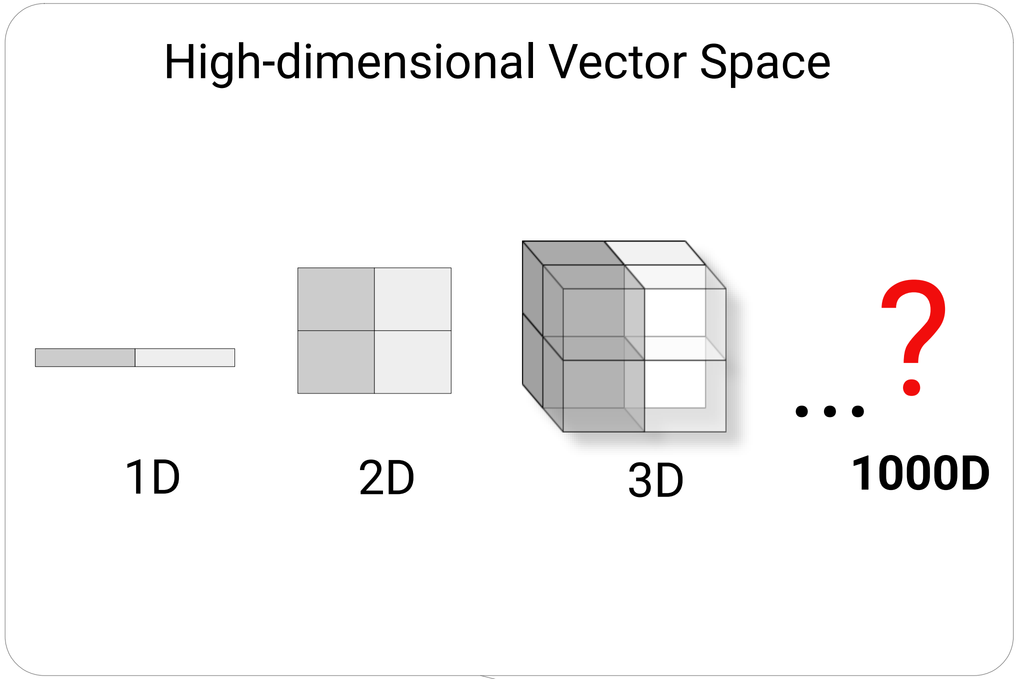
What is HDC/VSA?

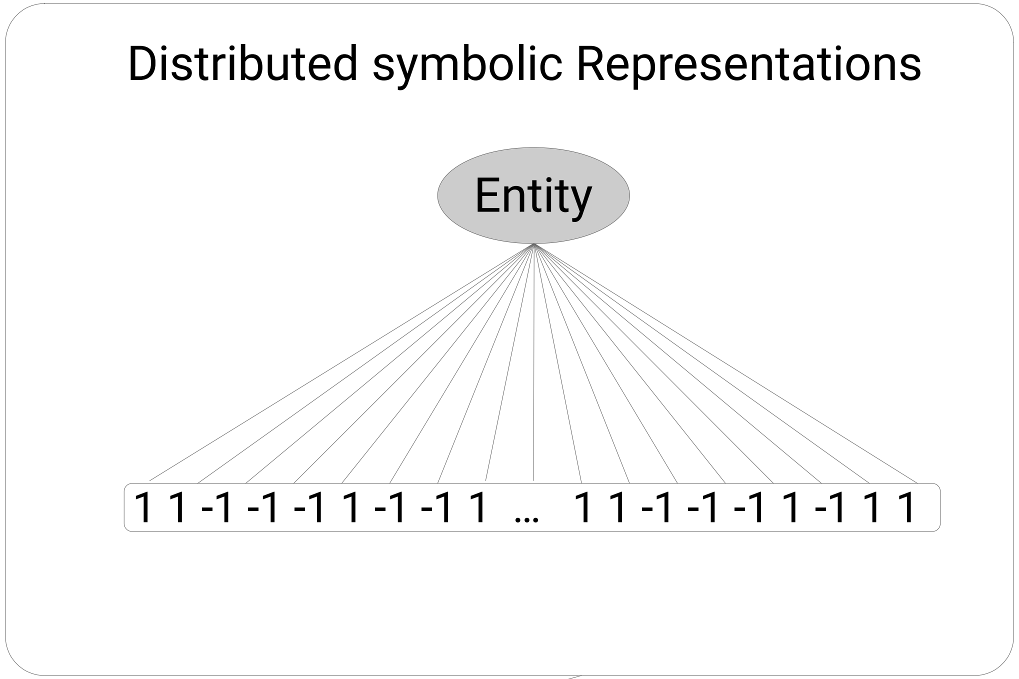
Hyperdimensional Computing (HDC)

Vector Symbolic Architectures combine a high-dimensional vector space with a set of carefully designed operators in order to perform symbolic computations with large numerical vectors.

Vector Symbolic Architecture (VSA)

Robustness comes from HD space and distributed representations



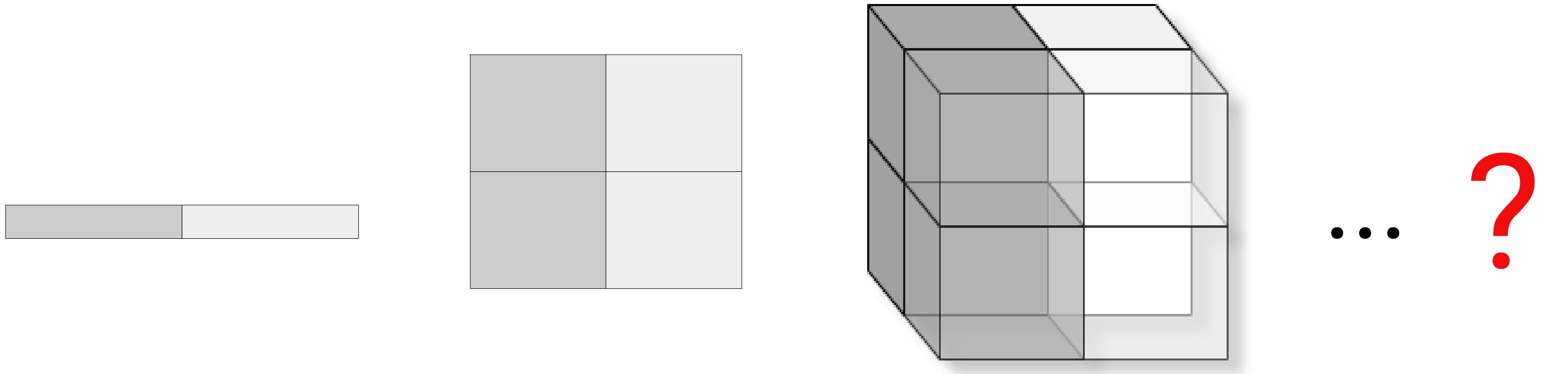


Robustness

Noise has low influence on nearest neighbor queries with random vectors

Highdimensionality - Going from 1D to 1000D

- Capacity of high-dimensional spaces grows exponentially
- For example bipolar vector space {-1, 1}:



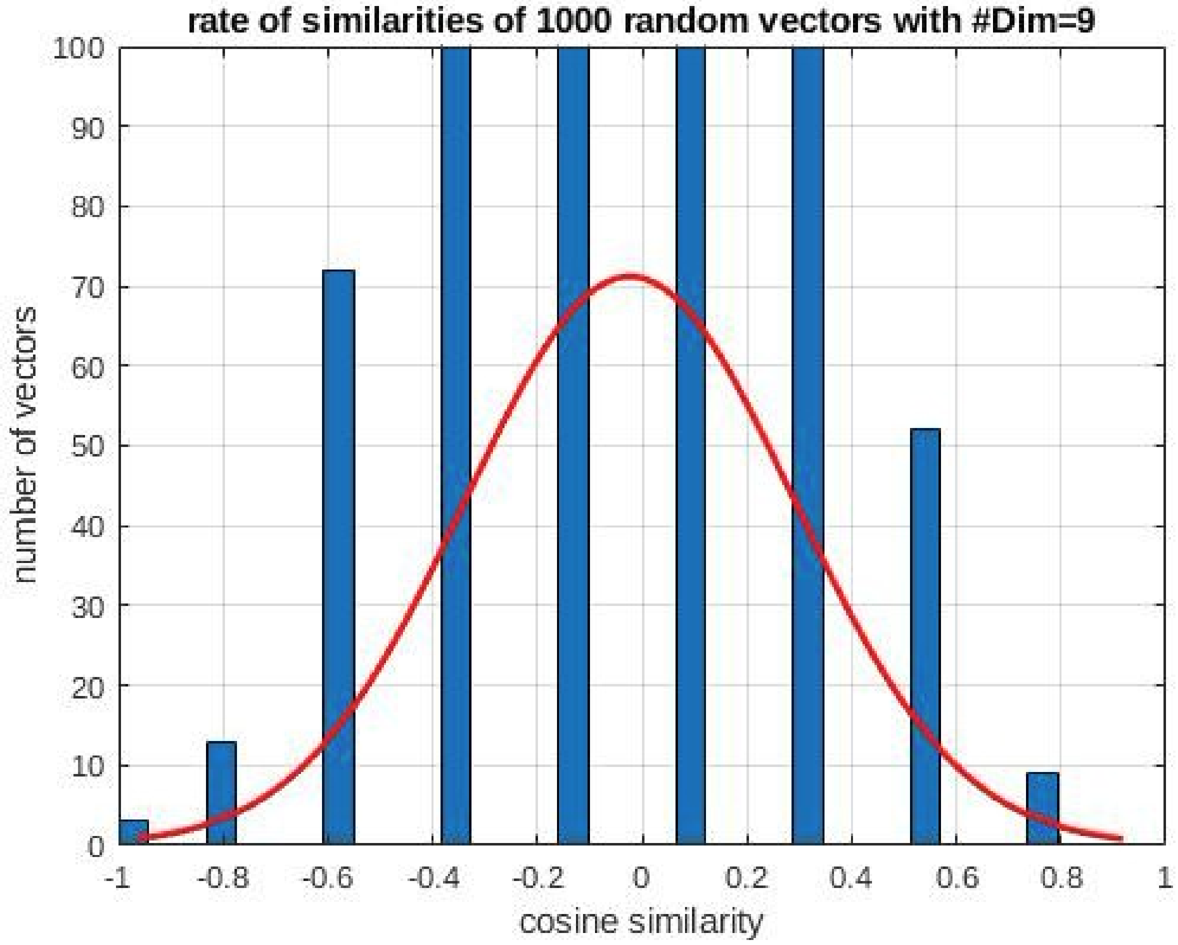
	1D	2D	3D	1000D
Capacity	**2**	**4**	**8**	**~ 10^{300}**

Atoms in the universe = 10^{80}

High-dimensional (HD) Vectors

Random High-dimensional vectors are almost orthogonal:

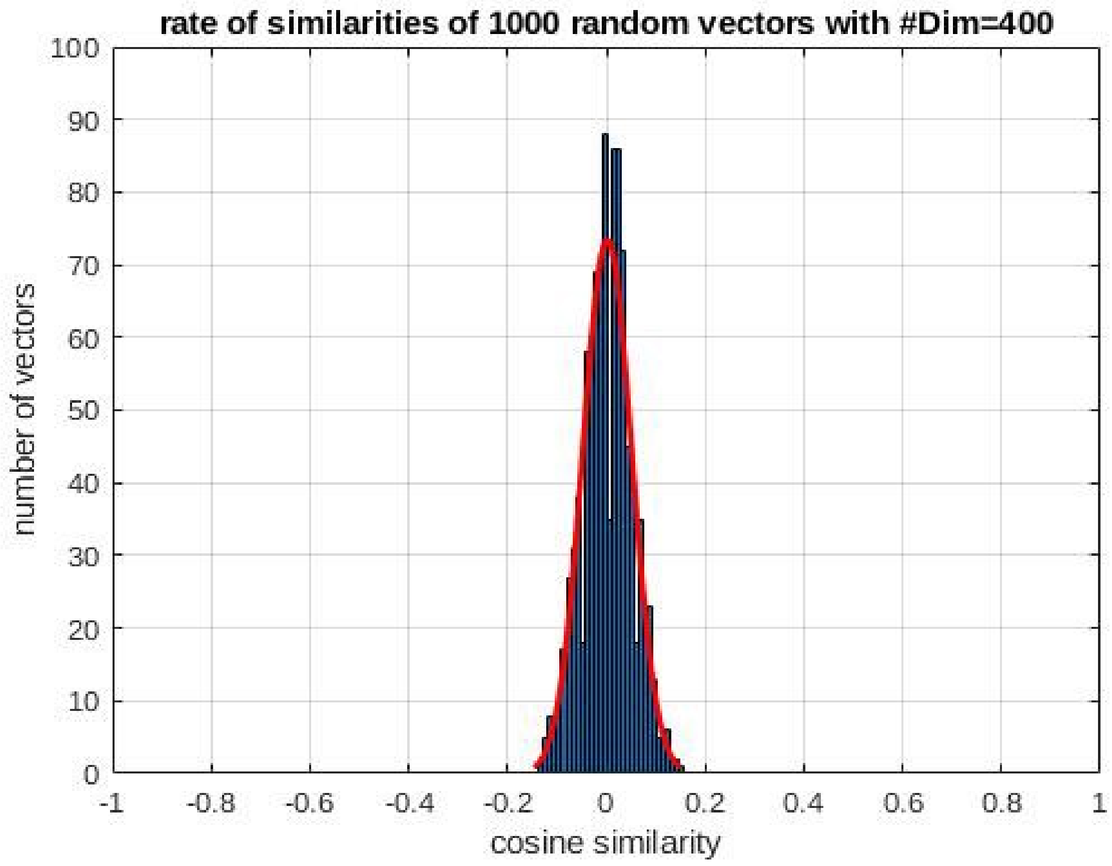
- Generation of random vectors from a high dimensional vector space
- These vectors are dissimilar in terms of cosine angle (Euclidean distance becomes meaningless in high-dimensional space)



High-dimensional (HD) Vectors

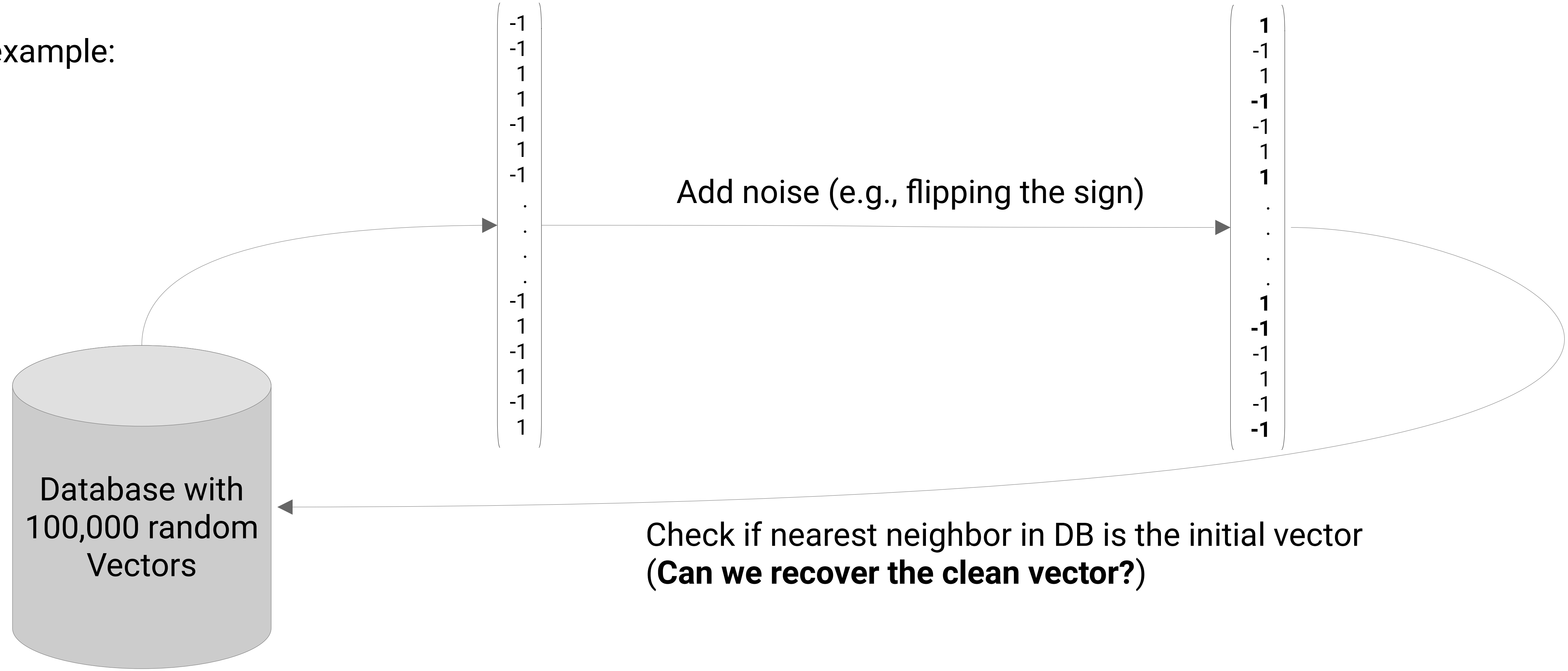
Random High-dimensional vectors are almost orthogonal:

- Generation of random vectors from a high dimensional vector space
- These vectors are dissimilar in terms of cosine angle (Euclidean distance becomes meaningless in high-dimensional space)

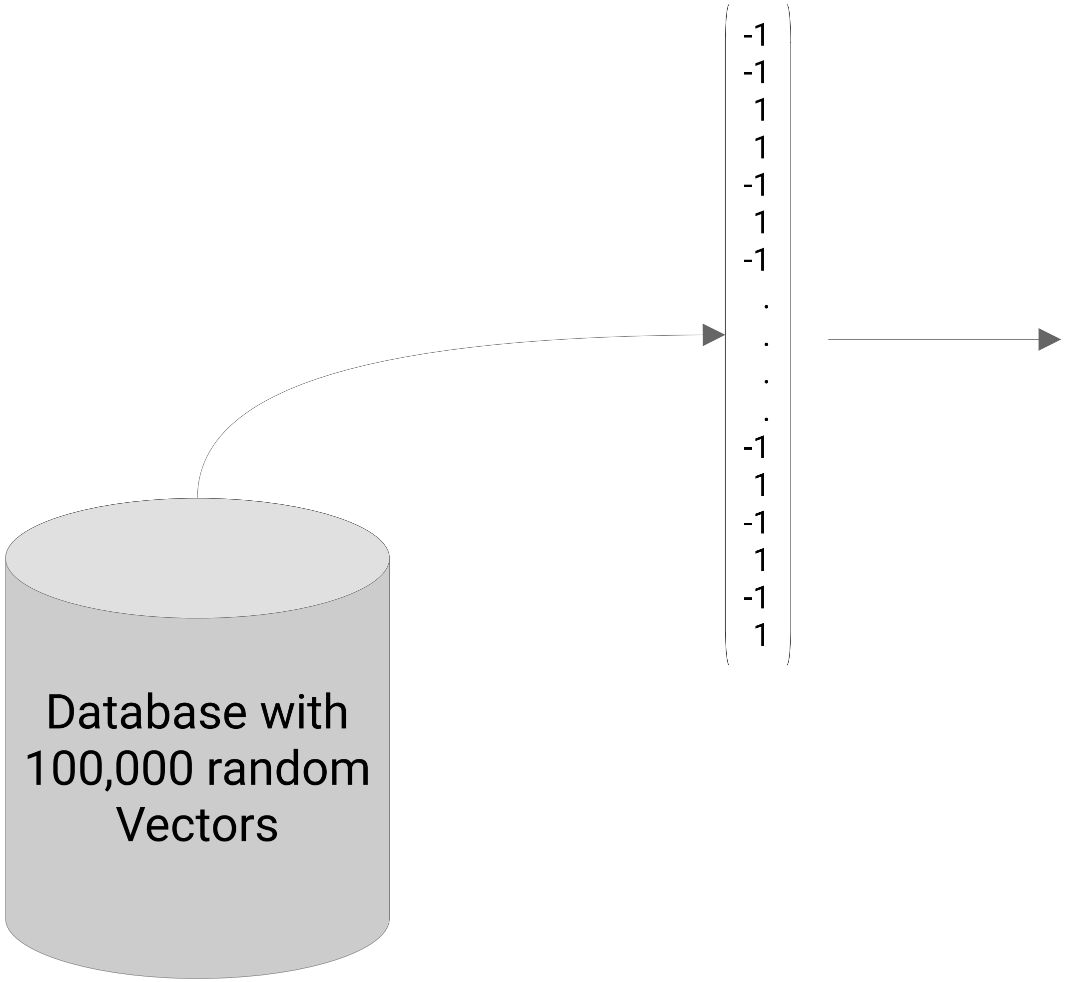


High-dimensional (HD) Vectors

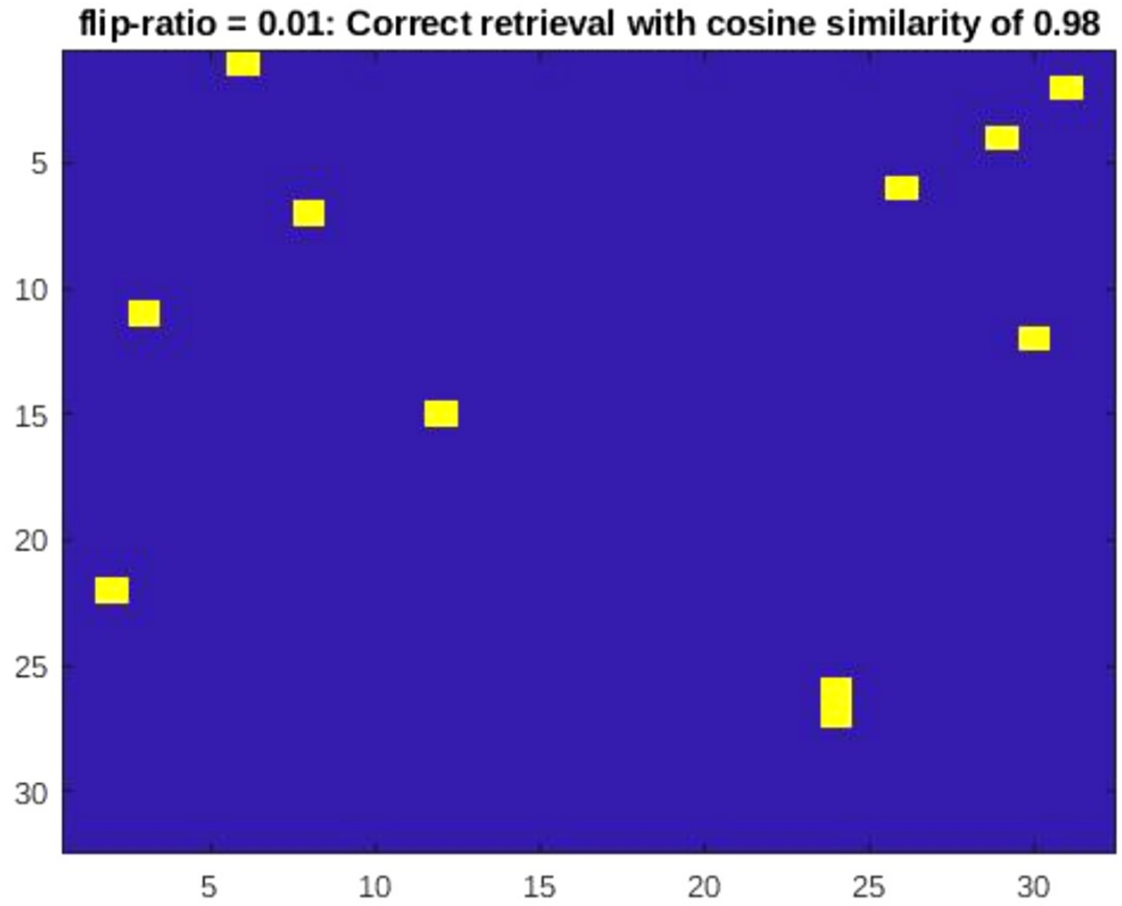
An example:



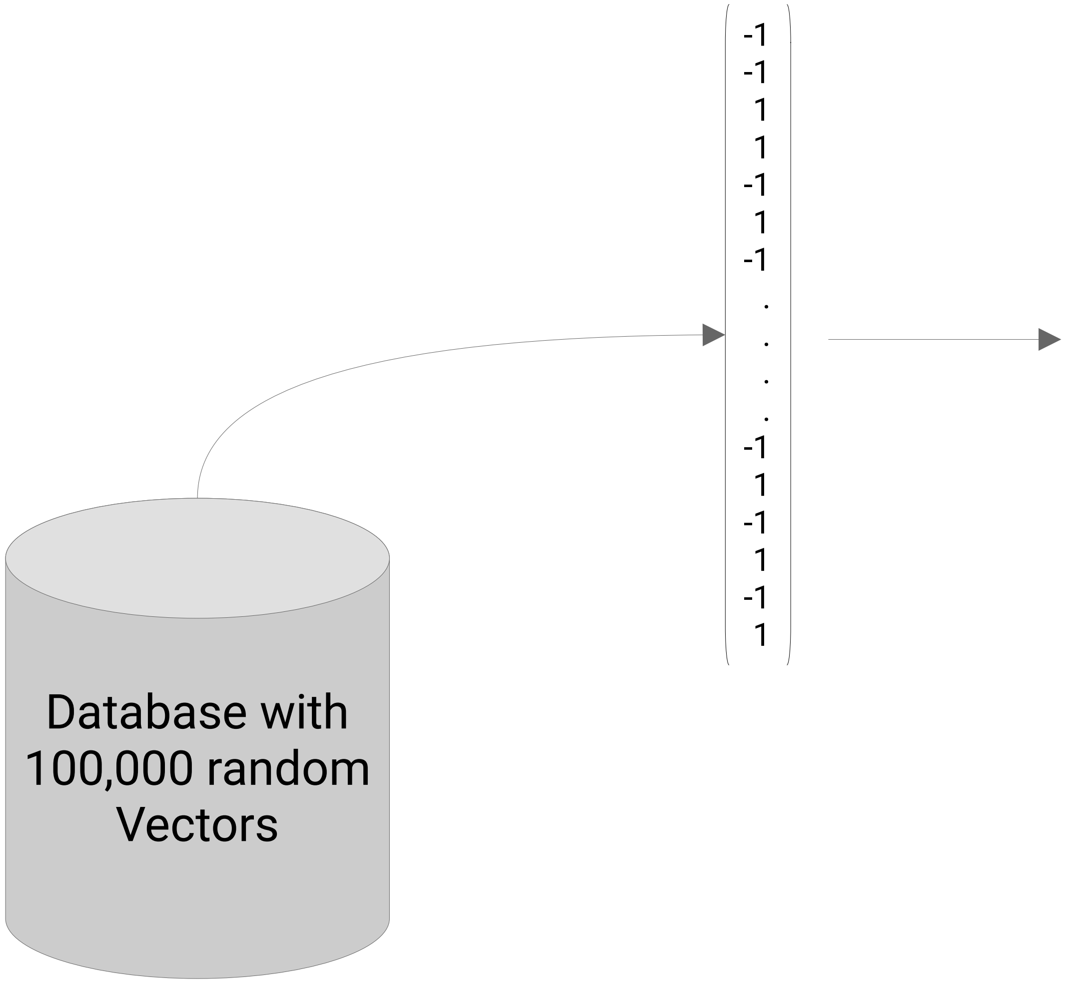
High-dimensional (HD) Vectors



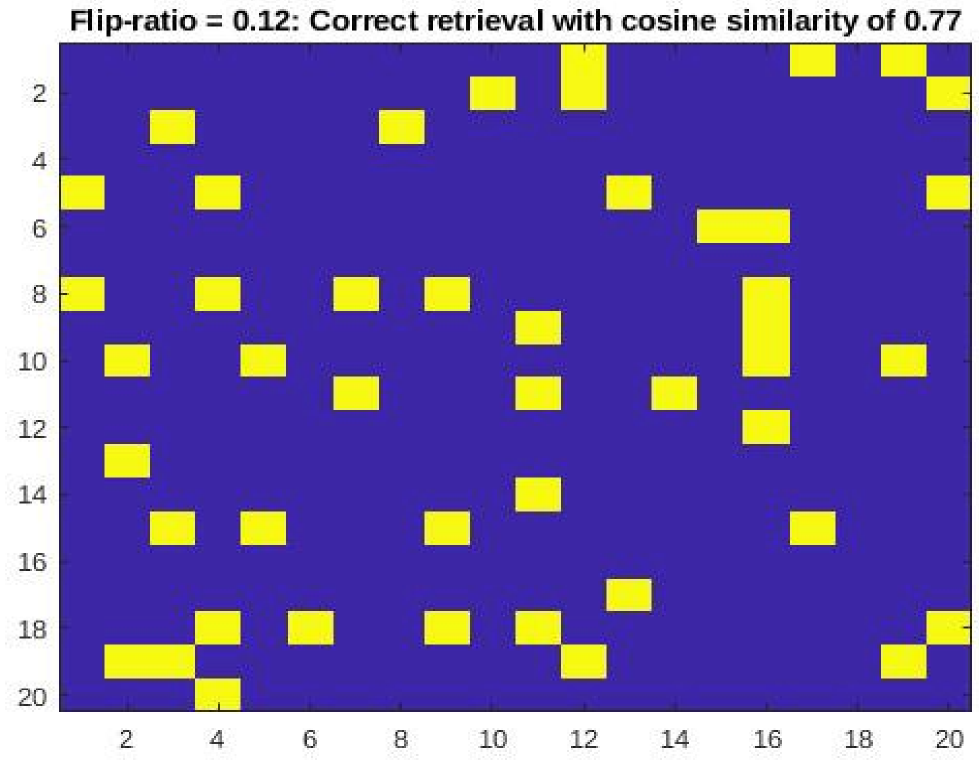
Random flips at 1024 dimensional vector



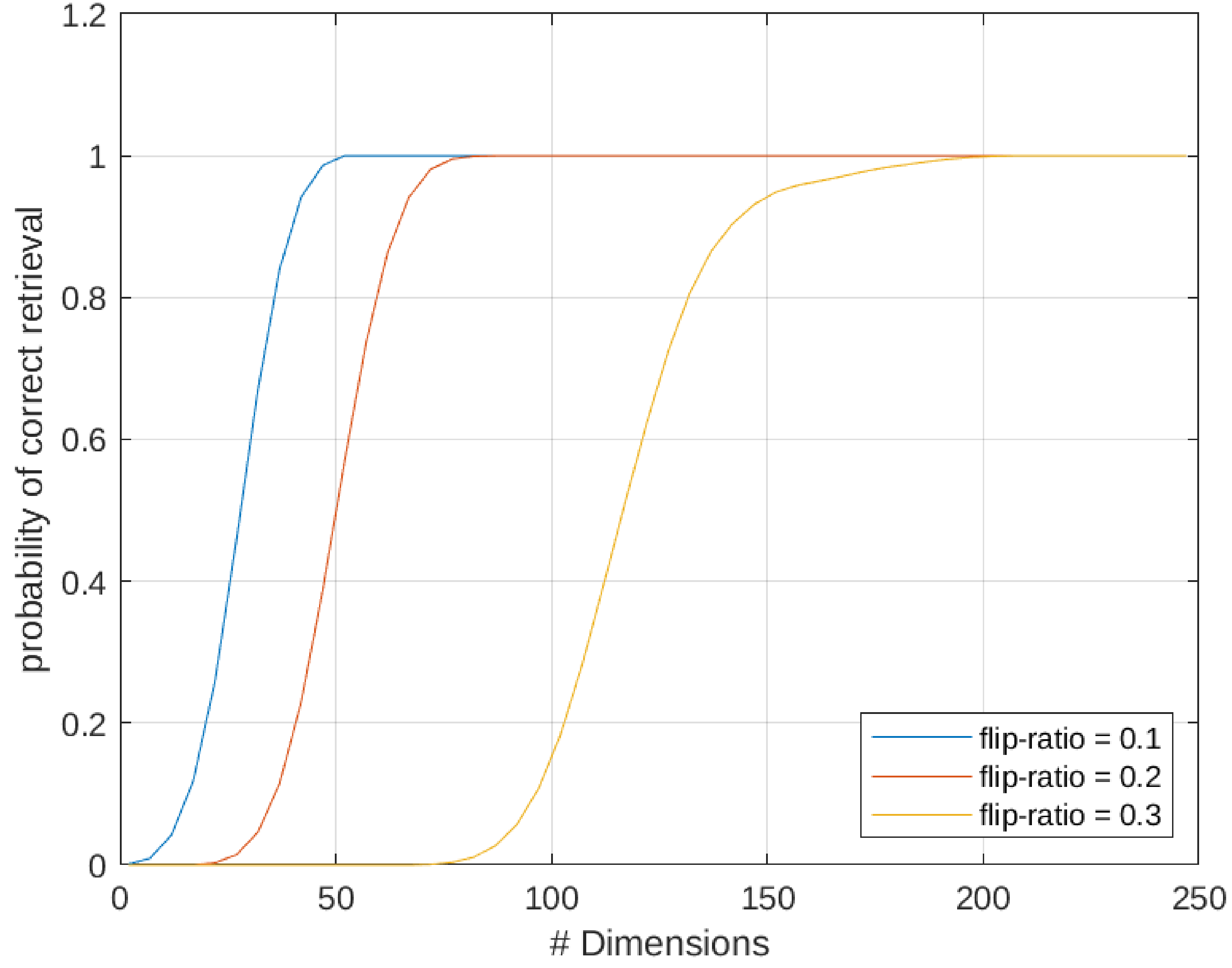
High-dimensional (HD) Vectors



Random flips at 1024 dimensional vector



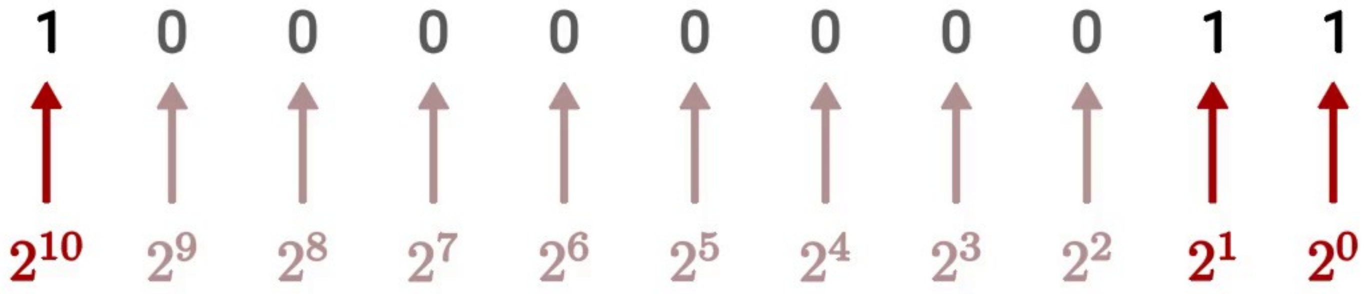
High-dimensional (HD) Vectors



Symbols as Distributed Representation to use the properties of HD space

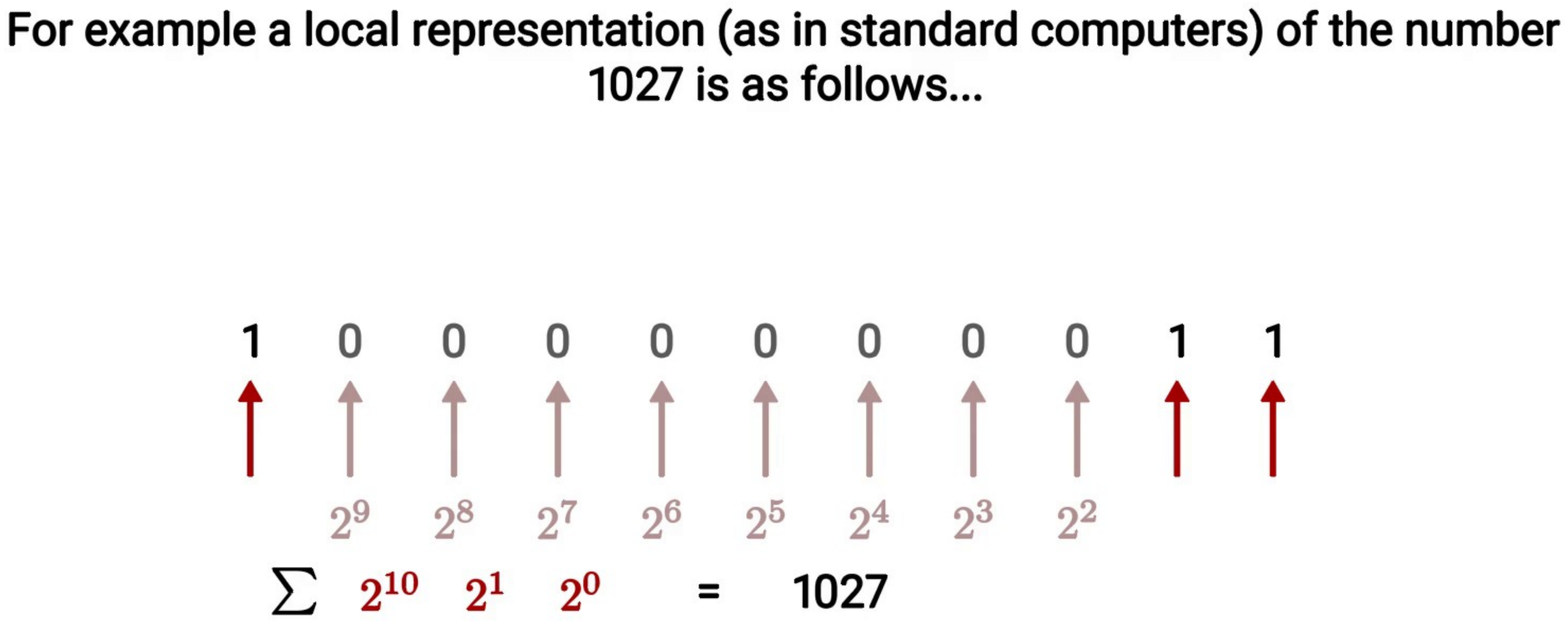
Distributed representation means that each individual value has no specific meaning unlike **local representations:**

For example a local representation (as in standard computers) of the number 1027 is as follows...



Symbols as Distributed Representation to use the properties of HD space

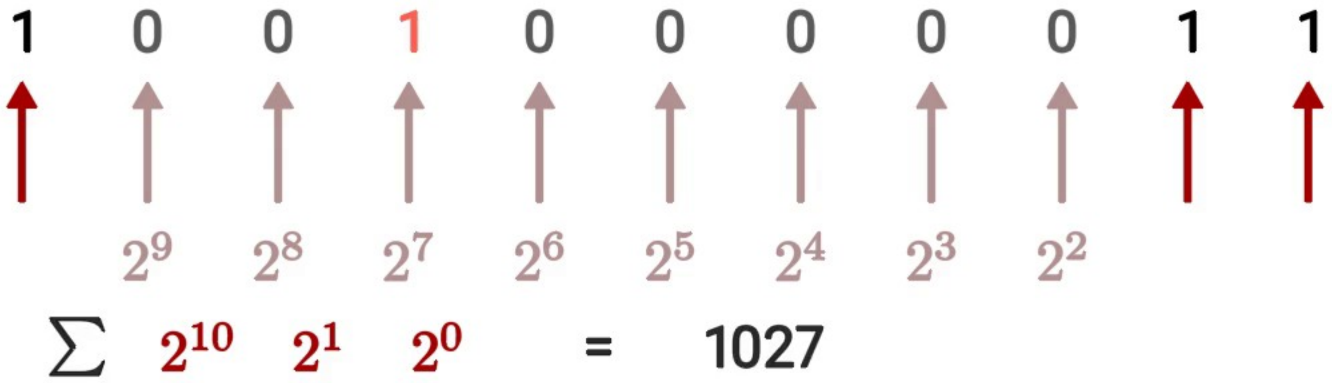
Distributed representation means that each individual value has no specific meaning unlike **local representations:**



Symbols as Distributed Representation to use the properties of HD space

Distributed representation means that each individual value has no specific meaning unlike **local representations:**

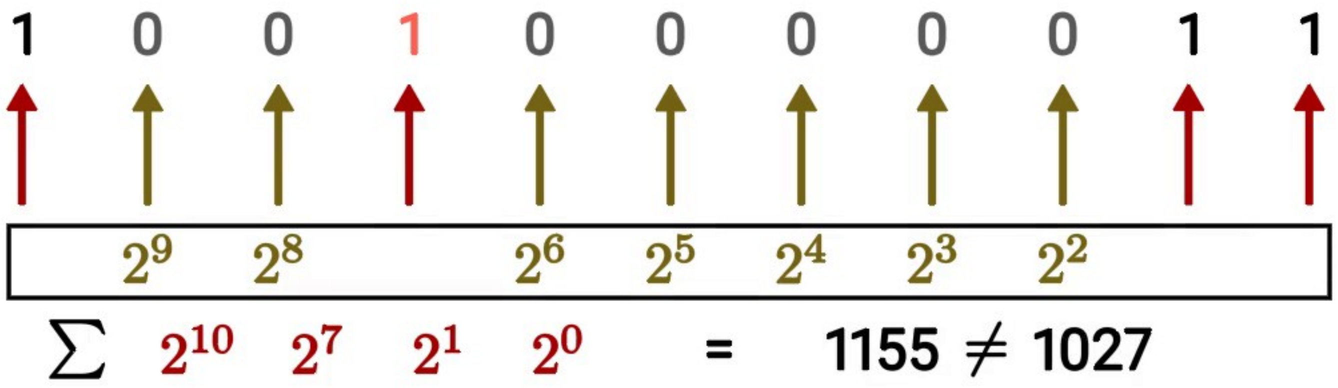
What if one bit is erroneously flipped?



Symbols as Distributed Representation to use the properties of HD space

Distributed representation means that each individual value has no specific meaning unlike **local representations:**

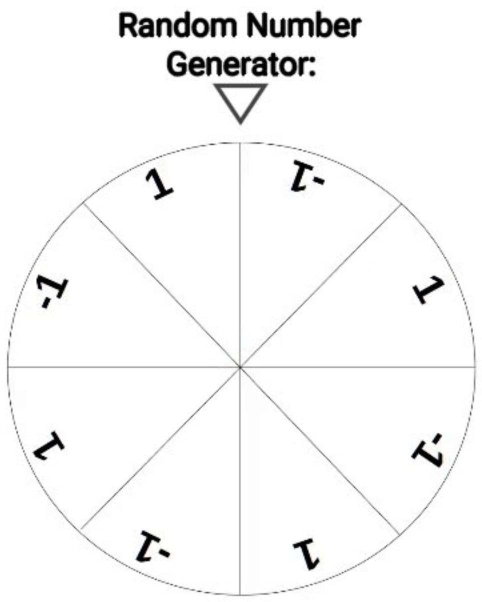
Each local bit has a specific meaning!



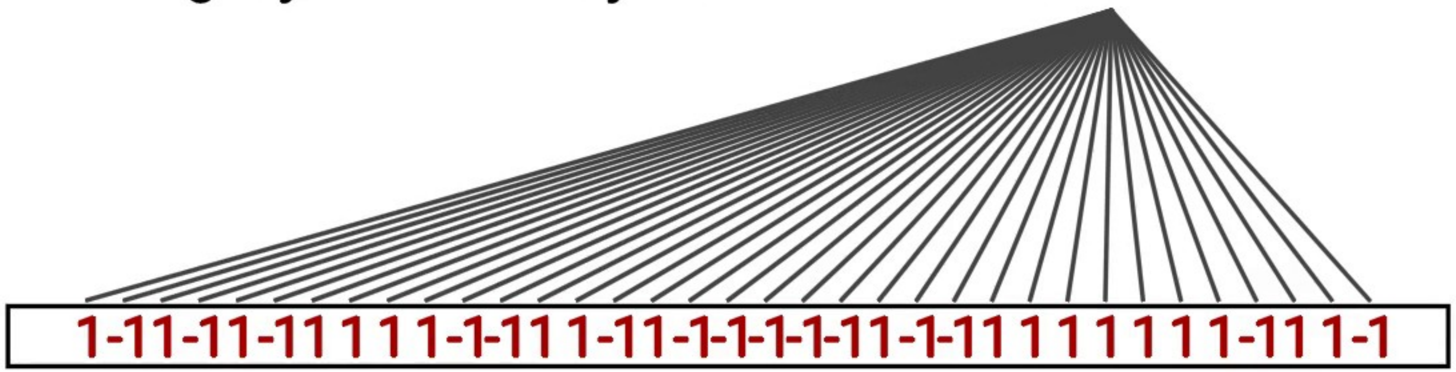
Symbols as Distributed Representation to use the properties of HD space

To overcome this local meaning HDC uses distributed representations

Information (e.g. a number) is encoded in **very large random vectors** (much larger than would be required to just distinguish the symbols)



E.g. symbolic entity for the number 1027



Random distributed representation of the symbol "1027"

Motivation and bigger Picture from the view of AI

	Symbolic (Traditional AI)	Sub-symbolic (Neural Networks)	HDC / VSA
Distributed Representation	X	✓	✓
Robust	X	?	✓
Learn from Data	X	✓	
Symbolic computations	✓	X	
Interpretable	✓	X	

Table based on the HDC-Course, 2021, UC Berkeley, https://www.hd-computing.com/course-computing-with-high-dimensional-vectors

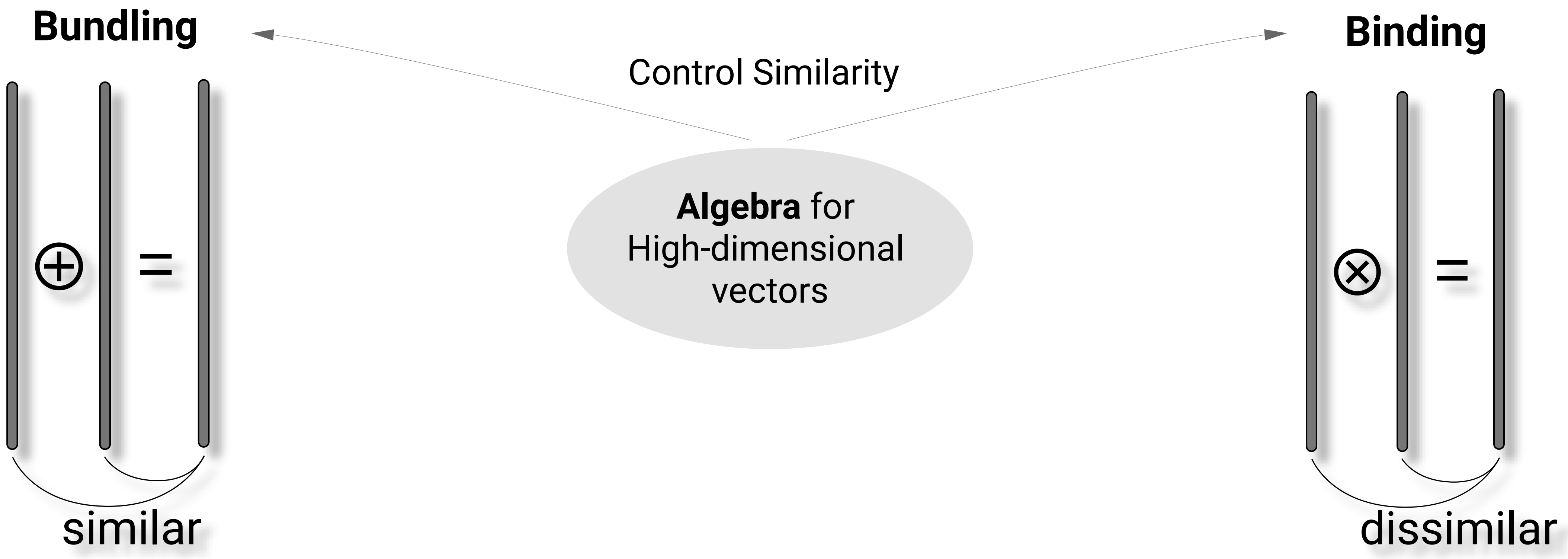
What is HDC/VSA?

Hyperdimensional Computing (HDC)

Vector Symbolic Architectures combine a high-dimensional vector space with a set of carefully designed operators in order to perform symbolic computations with large numerical vectors.

Vector Symbolic Architecture (VSA)

Operations for symbolic computation

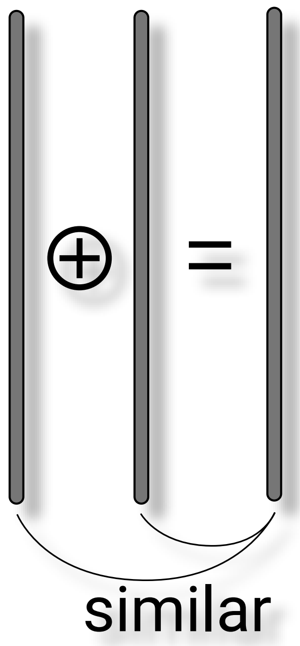


- Output is similar to each input

- Output is non-similar to each input
- invertible (⊘)
- Distributes over bundling

Operations for symbolic computation

Bundling



- Output is similar to each input

- Can be implemented as simple **element-wise addition**
- **Overlaying of vectors** (superposition)

$$\begin{pmatrix} -1 \\ -1 \\ 1 \\ 1 \\ -1 \\ 1 \\ -1 \\ \vdots \\ -1 \\ 1 \\ -1 \\ 1 \\ -1 \\ -1 \end{pmatrix} + \begin{pmatrix} -1 \\ 1 \\ 1 \\ -1 \\ 1 \\ -1 \\ -1 \\ \vdots \\ 1 \\ -1 \\ -1 \\ -1 \\ 1 \\ -1 \end{pmatrix} = \begin{pmatrix} -2 \\ 0 \\ 2 \\ 0 \\ 0 \\ 0 \\ -2 \\ \vdots \\ 0 \\ 0 \\ -2 \\ 0 \\ 0 \\ -2 \end{pmatrix}$$

0.05 0.68
0.68

Cosine similarities

Operations for symbolic computation

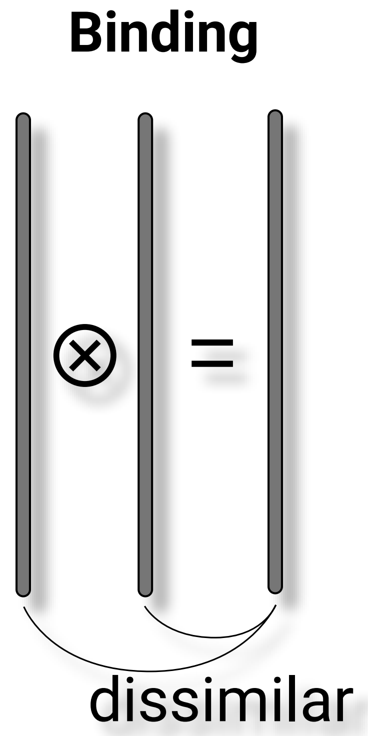
$$
\begin{array}{c}v1\\ \begin{pmatrix}-1\\-1\\1\\1\\-1\\1\\-1\\ \vdots\\-1\\1\\-1\\1\\-1\\-1\end{pmatrix}\end{array}
\otimes
\begin{array}{c}v2\\ \begin{pmatrix}-1\\1\\1\\-1\\1\\-1\\-1\\ \vdots\\1\\-1\\-1\\-1\\1\\-1\end{pmatrix}\end{array}
=
\begin{pmatrix}1\\-1\\1\\-1\\-1\\-1\\1\\ \vdots\\-1\\-1\\1\\-1\\-1\\1\end{pmatrix}
\oslash
\begin{array}{c}v1\\ \begin{pmatrix}-1\\-1\\1\\1\\-1\\1\\-1\\ \vdots\\-1\\1\\-1\\1\\-1\\-1\end{pmatrix}\end{array}
=
\begin{array}{c}v2\\ \begin{pmatrix}-1\\1\\1\\-1\\1\\-1\\-1\\ \vdots\\1\\-1\\-1\\1\\1\\-1\end{pmatrix}\end{array}
$$

0.05 0.01

0.01 1

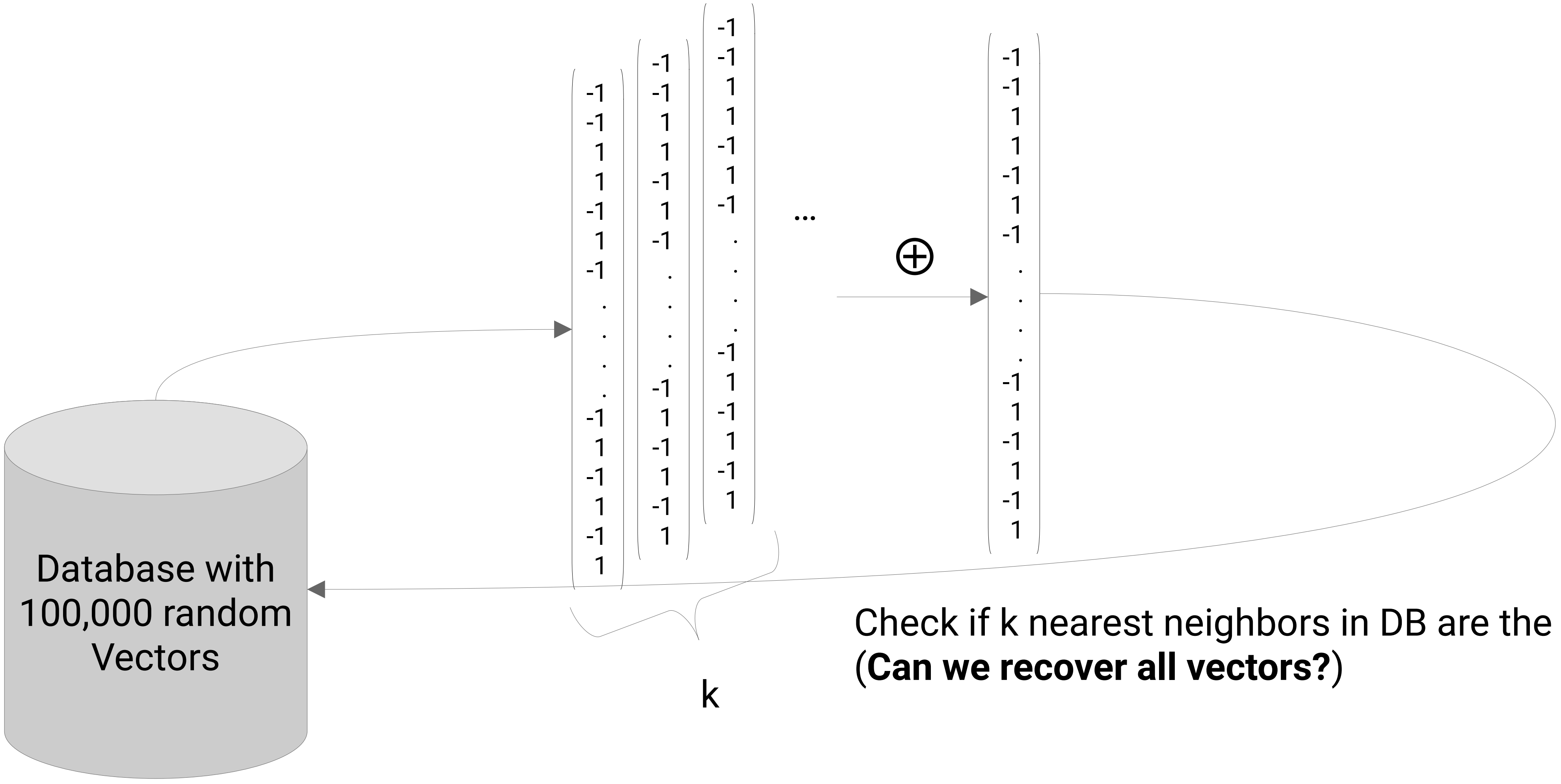
Cosine similarities

- Binding can be implemented as simple **element-wise multiplication**
- Binding is in that case self inverse v1 ⊗ v1 = I
- **Associate vectors**
- Unbinding is element-wise multiplication too (recovering)



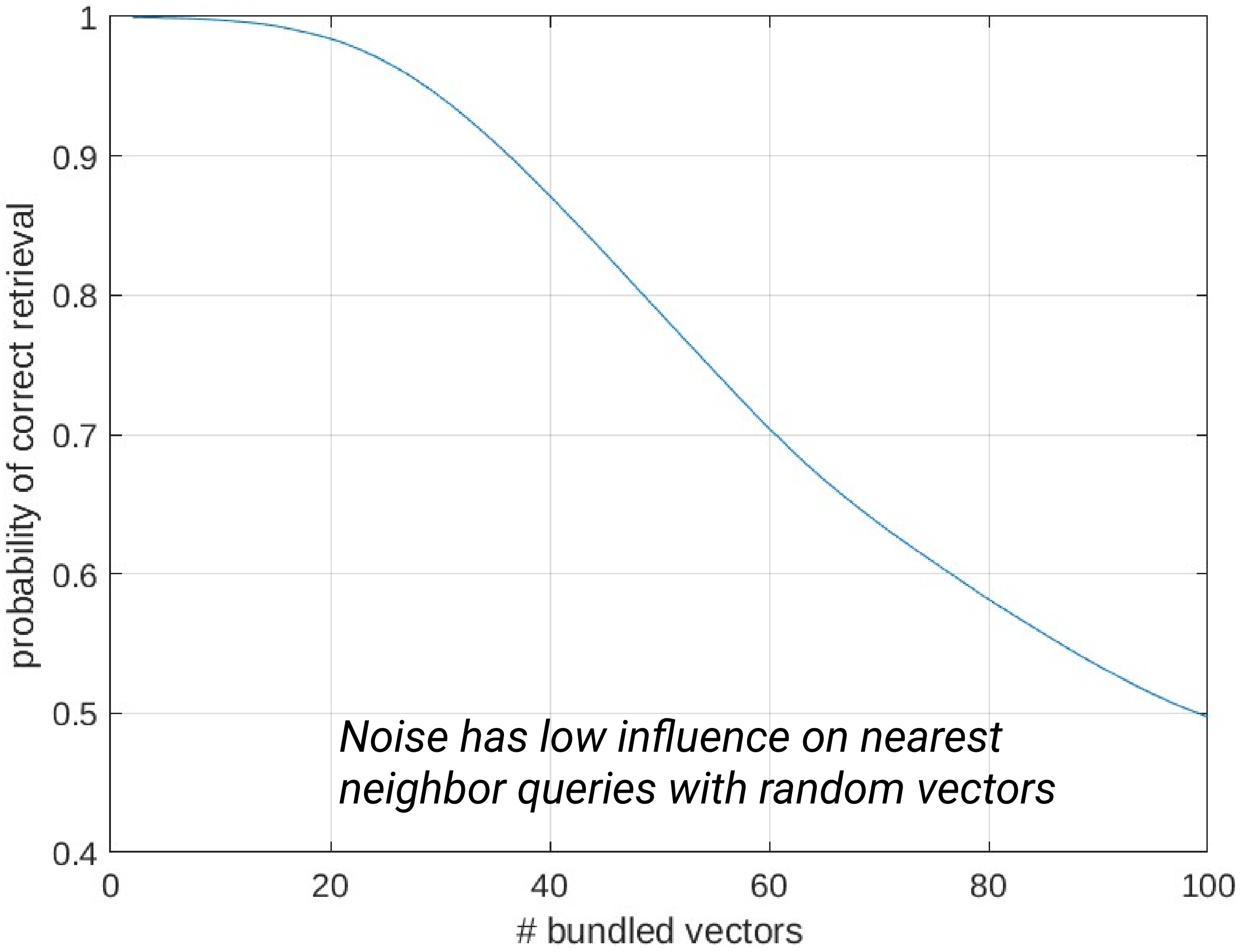
- Output is non-similar to each input
- invertible (⊘)

Bundle multiple random vectors



Check if k nearest neighbors in DB are the bundled vectors
(**Can we recover all vectors?**)

Bundle multiple random vectors (1000D)



Building Structures with HDC

A common structure for encoding information with VSA are **Bundled Role-Filler-Pairs**

Combine a scene: ***An orange ball lies on the ground.***

role		*filler*		*role*		*filler*		*role*		*filler*
Color	⊗	Orange	⊕	Object	⊗	Ball	⊕	Location	⊗	Ground

Scene Representation ⊘ Color

Asking for the color...

~~Color~~ ⊗ Orange ⊘ ~~Color~~ ⊕ Object ⊗ ***Noise*** ⊘ Color ⊕ Location ⊗ ***Noise*** ⊘ Color

A more complex Example: What is the dollar of Mexico? [1]

		role		*filler*		*role*		*filler*		*role*		*filler*	
Repr. USA	=	Country	⊗	USA	⊕	Capital	⊗	WDC	⊕	Currency	⊗	Dollar	Database
Repr. MEX	=	Country	⊗	Mexico	⊕	Capital	⊗	MXC	⊕	Currency	⊗	Peso	

Analogies = Repr. MEX ⊗ Repr. USA = ~~Country~~ ⊗ USA ⊗ ~~Country~~ ⊗ Mexico ⊕

Country ⊗ USA ⊗ Capital ⊗ MXC ⊕

...

~~Currency~~ ⊗ Dollar ⊗ ~~Currency~~ ⊗ Peso

Mapping of similar categories

Noise

Analogies = USA ⊗ Mexico ⊕ WDC ⊗ MXC ⊕ Dollar ⊗ Peso

Analogies ⊗ Dollar = USA ⊗ Mexico ⊕ WDC ⊗ MXC ⊕ ~~Dollar~~ ⊗ Peso

Noise

"What is the dollar of Mexico?" = Peso + Noise

[1] P. Kanerva, "Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors," Cognit. Comput., 2009.

Motivation and bigger Picture from the view of AI

	Symbolic (Traditional AI)	Sub-symbolic (Neural Networks)	HDC / VSA
Distributed Representation	X	✓	✓
Robust	X	?	✓
Learn from Data	X	✓	
Symbolic computations	✓	X	✓
Interpretable	✓	X	✓

Table based on the HDC-Course, 2021, UC Berkeley, https://www.hd-computing.com/course-computing-with-high-dimensional-vectors

Learning from data

Hand-crafted

Systematically encoded input data with VSA/HDC

$$\begin{pmatrix} 1 \\ -1 \\ 1 \\ -1 \\ -1 \\ -1 \\ 1 \\ . \\ . \\ . \\ . \\ -1 \\ -1 \\ 1 \\ -1 \\ -1 \\ 1 \end{pmatrix}$$

Learning is possible

- K-nearest neighbors
- Neural Networks
- Regression

...

Motivation and bigger Picture from the view of AI

	Symbolic (Traditional AI)	Sub-symbolic (Neural Networks)	HDC / VSA
Distributed Representation	X	✓	✓
Robust	X	?	✓
Learn from Data	X	✓	–
Symbolic computations	✓	X	✓
Interpretable	✓	X	✓

Table based on the HDC-Course, 2021, UC Berkeley, https://www.hd-computing.com/course-computing-with-high-dimensional-vectors

Different Implementations

Real

Bipolar

Binary

Complex

Name	Elements X of vector space $\mathbb{V}$	Initialization of an atomic vector x_i	Typical used sim. metric	Bundling	Binding		Unbinding		Ref.
					Commutative	Associative	Commutative	Associative	
MAP-B	$X \in \{-1,1\}^D$	$x_i \sim \mathcal{B}(0.5) \cdot 2 - 1$	Cosine sim.	Elem. addition with threshold	Elem. multipl. ✓	✓	Elem. multipl. ✓	✓	Gayler and Levy (2009), Kleyko et al. (2018)

Different Implementations

MATLAB Toolbox [1]

Initialization

```
% create the object of a specific VSA type
type = 'MAP_B'; % available types: 'MAP_C'; 'MAP_B'; 'MAP_I';
VSA = vsa_env('vsa',type,'dim',1024);

% add vectors to item memory (randomly chosen)
VSA.add_vector('num',100);
```

Bundling

```
bundle = VSA.bundle(v1,v2);
```

Binding

```
bound_v = VSA.bind(v1,v2);
```

Unbinding

```
r = VSA.unbind(v1,bound_v);
```

Similarity

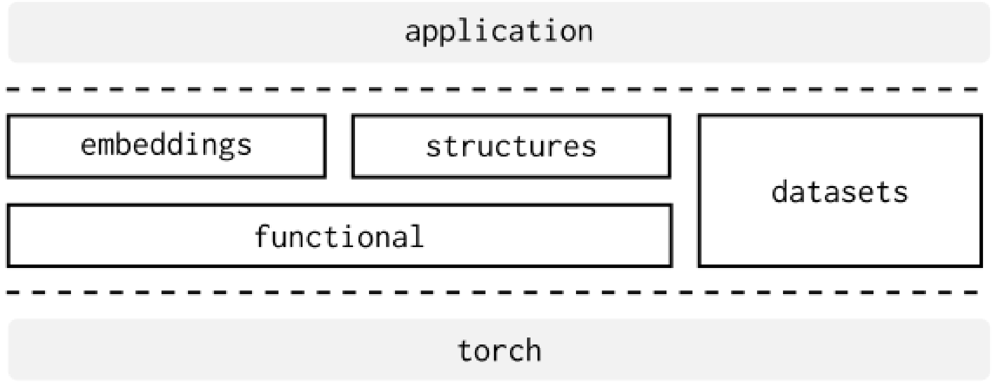
```
sim = VSA.sim(bundle,[v1 v2 v3]);
```

Clean-up

```
% finde the probe vector in item memory
[v_clean, name, s] = VSA.find_k_nearest(v,1);
```

→ Prototyping (is not a fast a possible but easy to use)

TorchHD (Pytorch) [2]



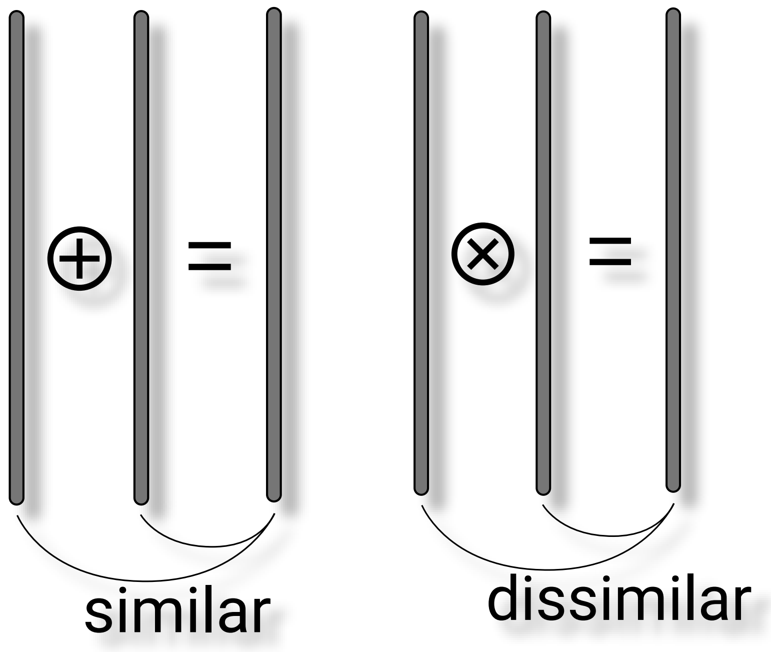
→ GPU support and many predefined functionalities

[1] K. Schlegel, P. Neubert, and P. Protzel, "A comparison of vector symbolic architectures," Artif. Intell. Rev., Dec. 2021.
[2] M. Heddes, I. Nunes, P. Vergés, D. Desai, T. Givargis, and A. Nicolau, Torchhd : An Open-Source Python Library to Support Hyperdimensional Torchhd : An Open-Source Python Library to Support Hyperdimensional Computing Research, 1. Association for Computing Machinery, 2022.

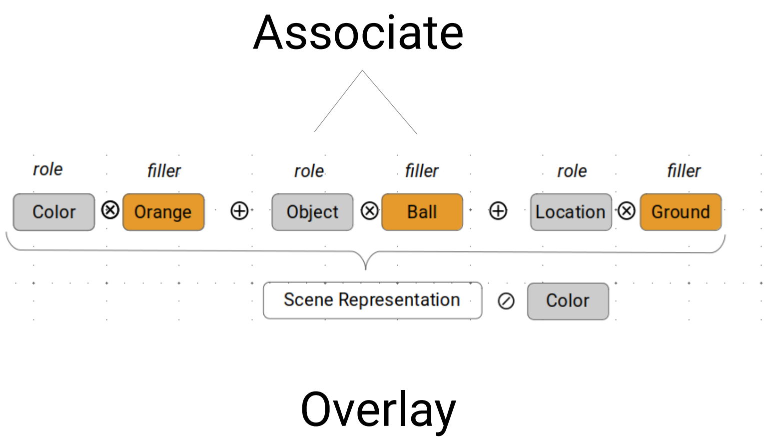
HDC/VSA is...

$$\begin{pmatrix} 1 \\ -1 \\ 1 \\ -1 \\ -1 \\ -1 \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ -1 \\ -1 \\ 1 \\ -1 \\ -1 \\ 1 \end{pmatrix}$$

...based on HD Vectors...



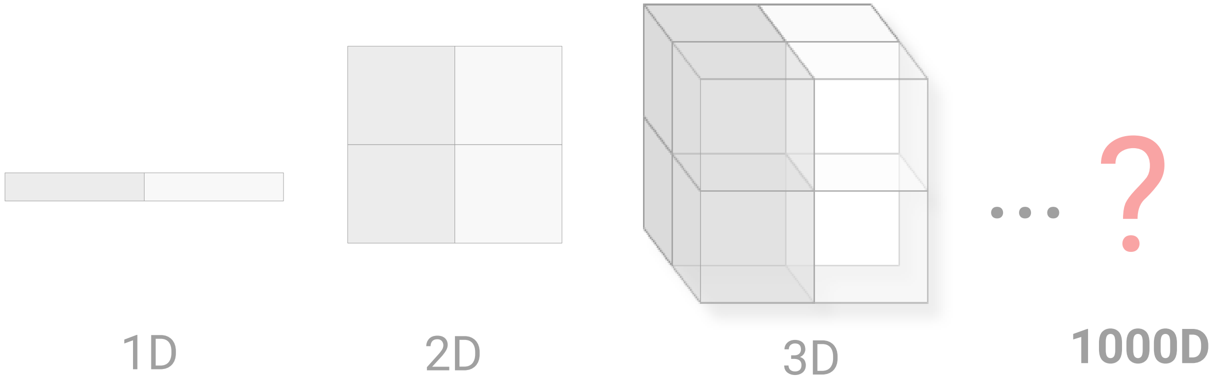
...and well defined operators...



...to systematically combine information.

The talk is about...

1. What?



High-Dimensional Computing

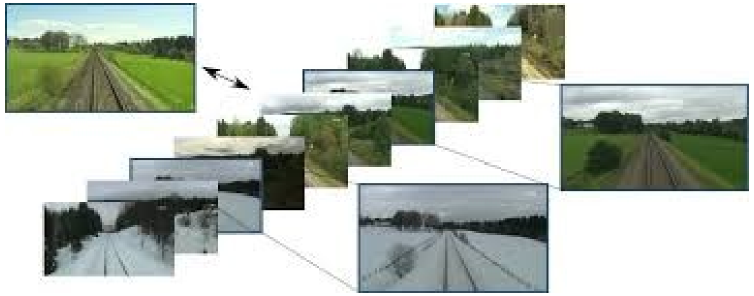
2. So What?

Example Applications

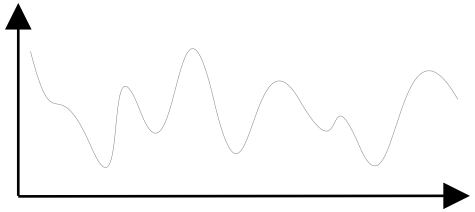
Use HDC in real applications

Most important step is converting data (information) into HD vectors

Images



Sensory data



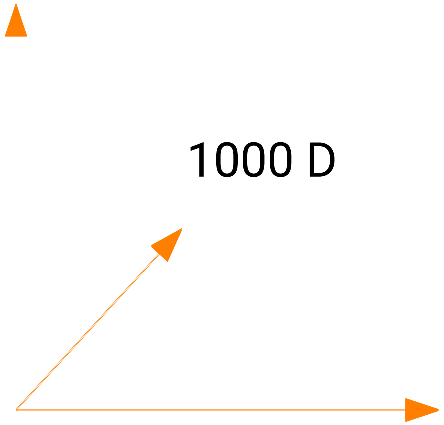
Others like unrelated entities

Random vectors for unrelated concepts

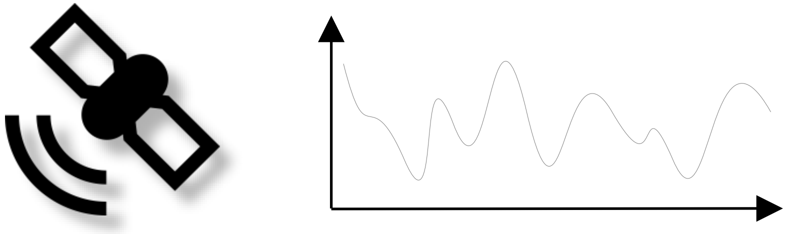
Deep-learned or **hand-crafted descriptors** (intermediate output of a neural network)

Similarity preserving encoding

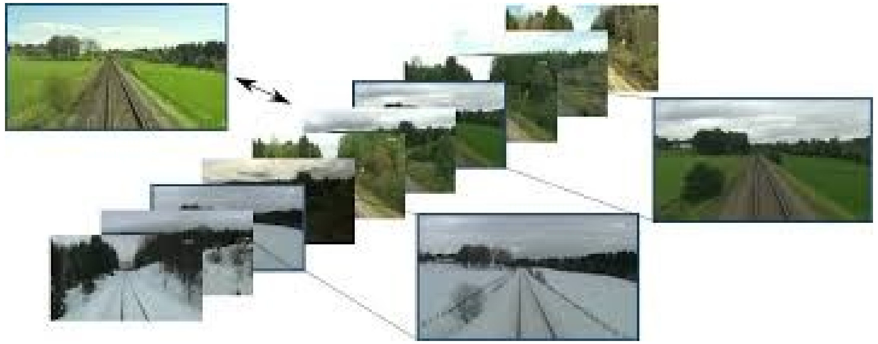
High-dimensional space



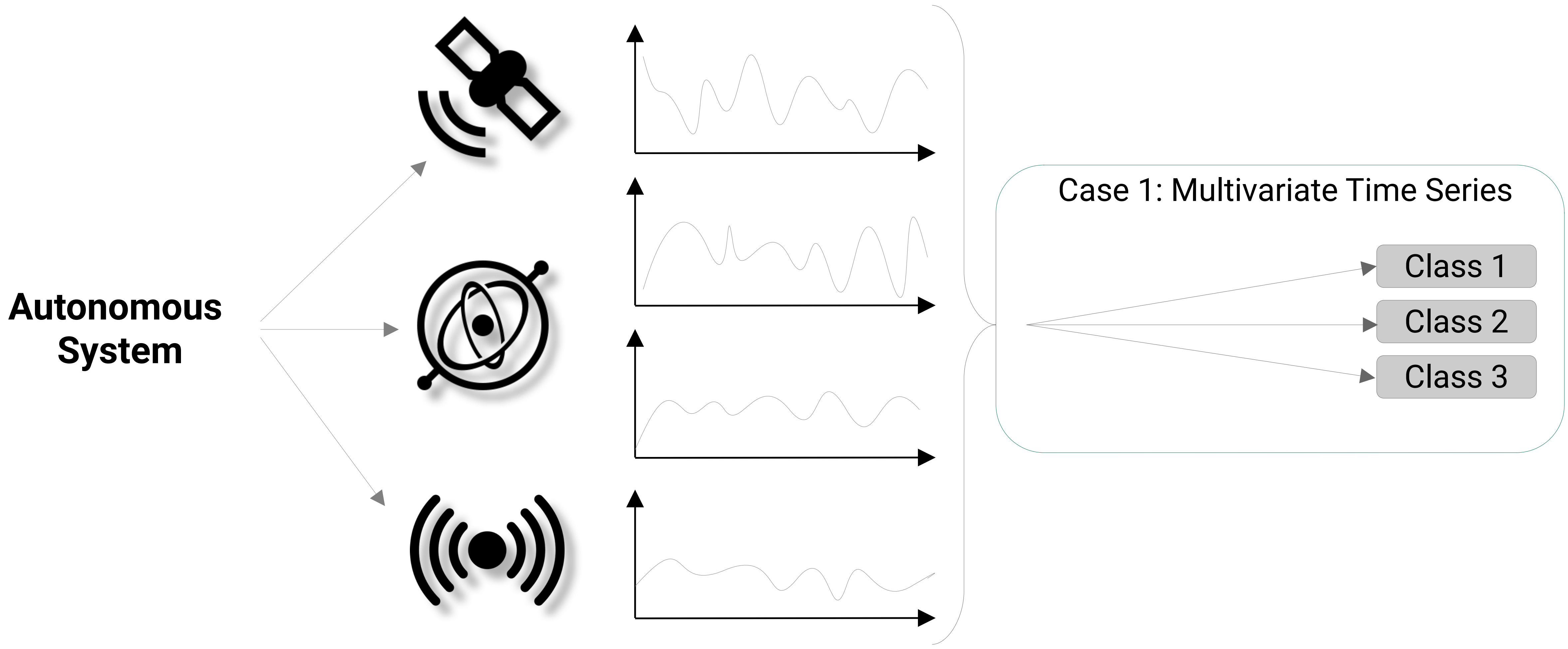
1. Time Series Classification



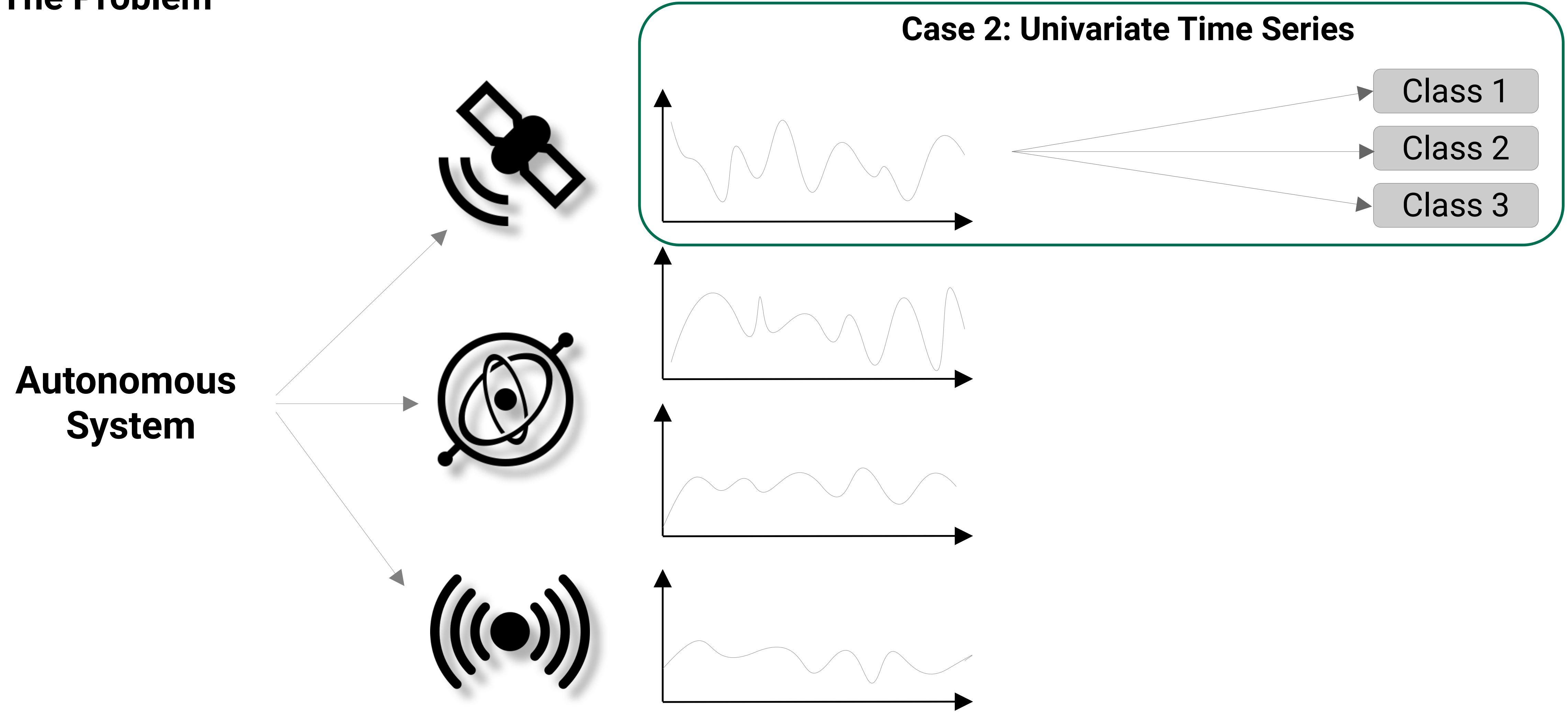
2. Visual Place Recognition



The Problem


Autonomous System
Case 1: Multivariate Time Series
Class 1
Class 2
Class 3

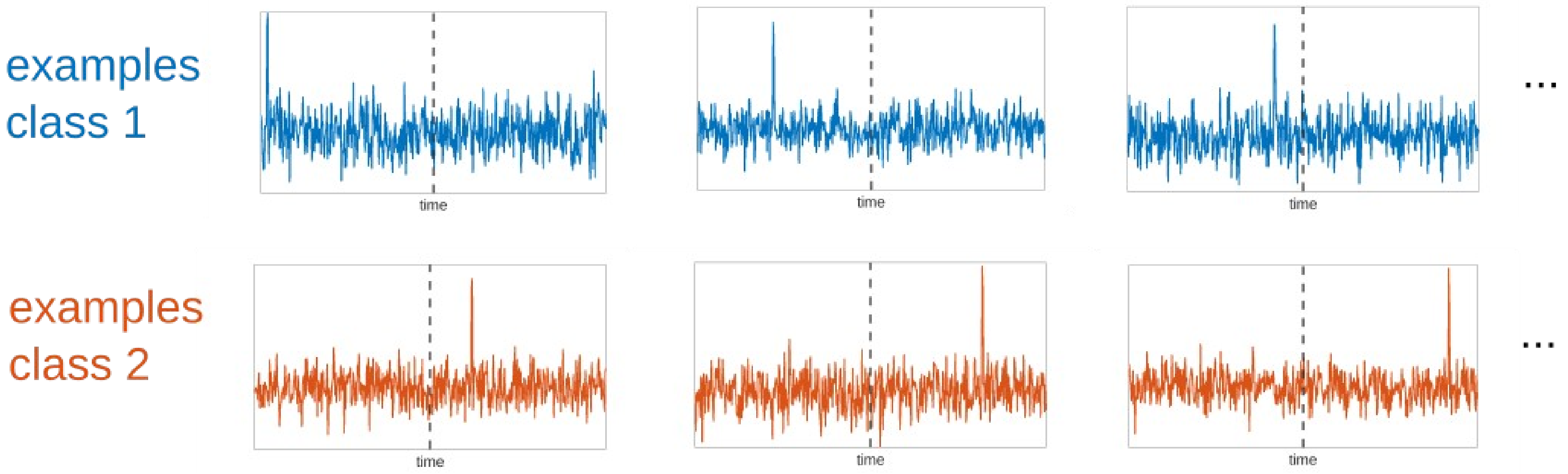
The Problem


Case 2: Univariate Time Series
Class 1
Class 2
Class 3
Autonomous System

The State-of-the-Art

One of the best perform algorithms in terms of accuracy and computing time is **MiniROCKET [1]**

A simple 2-class time series classification problem:
Gaussian noise with a single sharp peak either in the first half (class 1) or second half (class 2) of the signal



[1] A. Dempster, D. F. Schmidt, and G. I. Webb, "MiniRocket: A Very Fast (Almost) Deterministic Transform for Time Series Classification," Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., pp. 248–257, 2021.

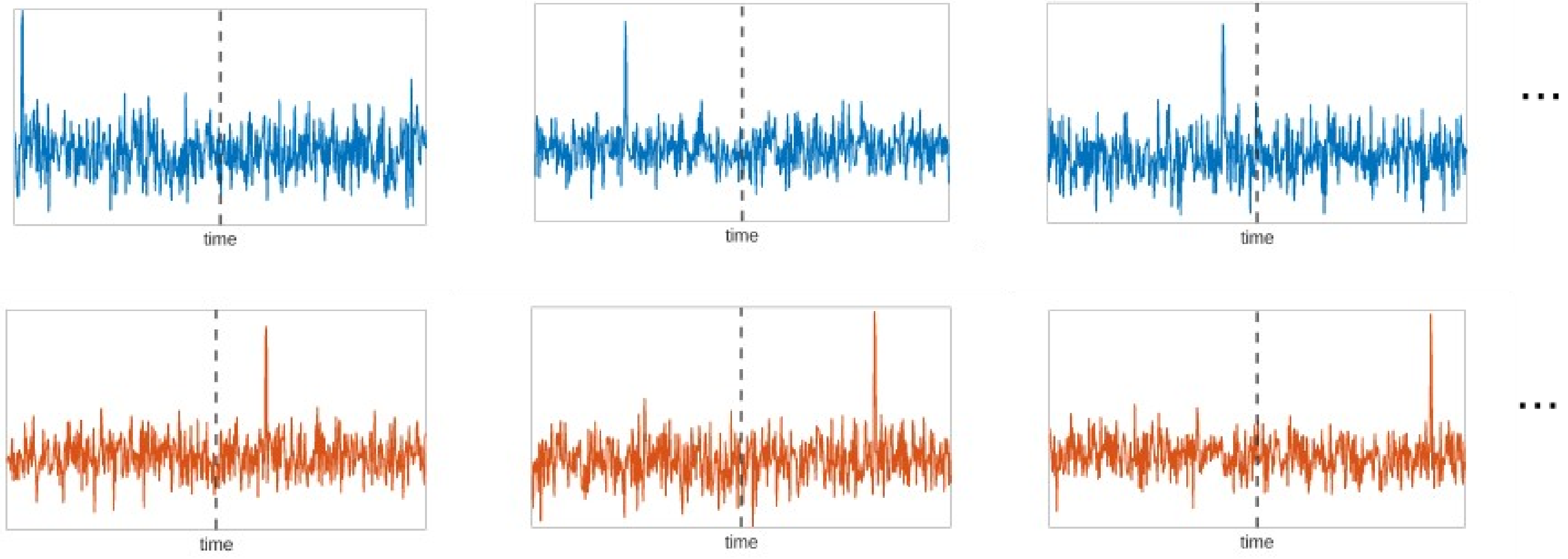
The State-of-the-Art

One of the best perform algorithms in terms of accuracy and computing time is **MiniROCKET [1]**

A simple 2-class time series classification problem:
Gaussian noise with a single sharp peak either in the first half (class 1) or second half (class 2) of the signal

examples class 1

examples class 2



Classification Accuracy of MiniROCKET:

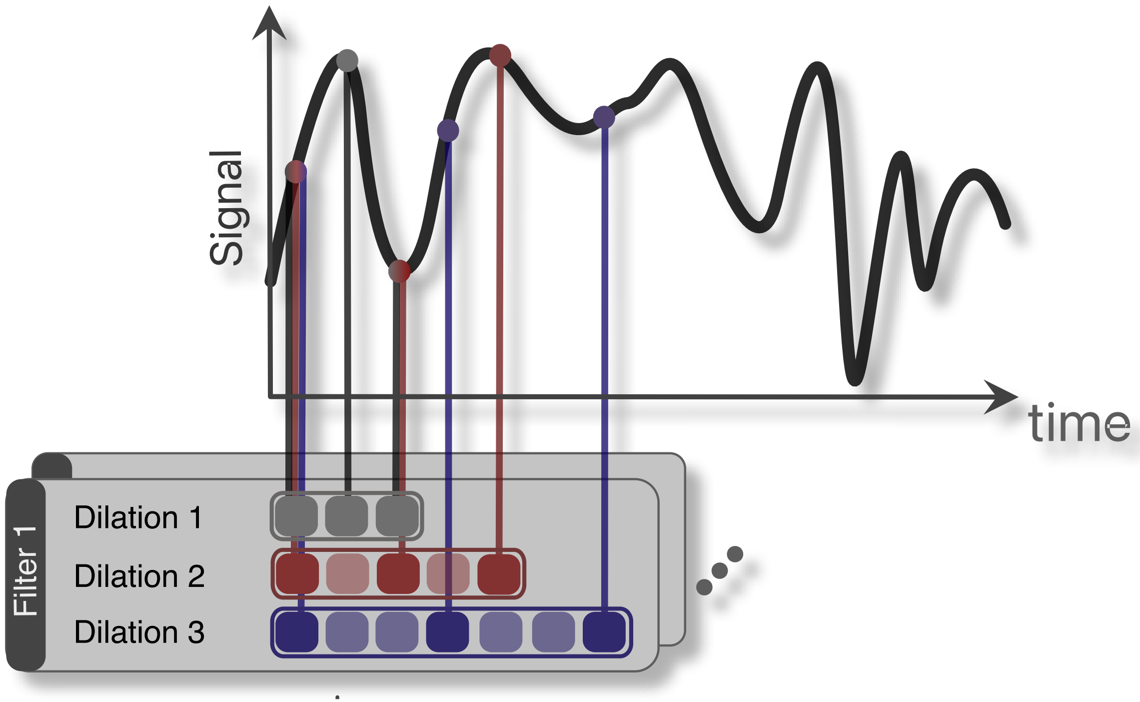
Standard Case: 67 %

Challenging Case: 56.9 %

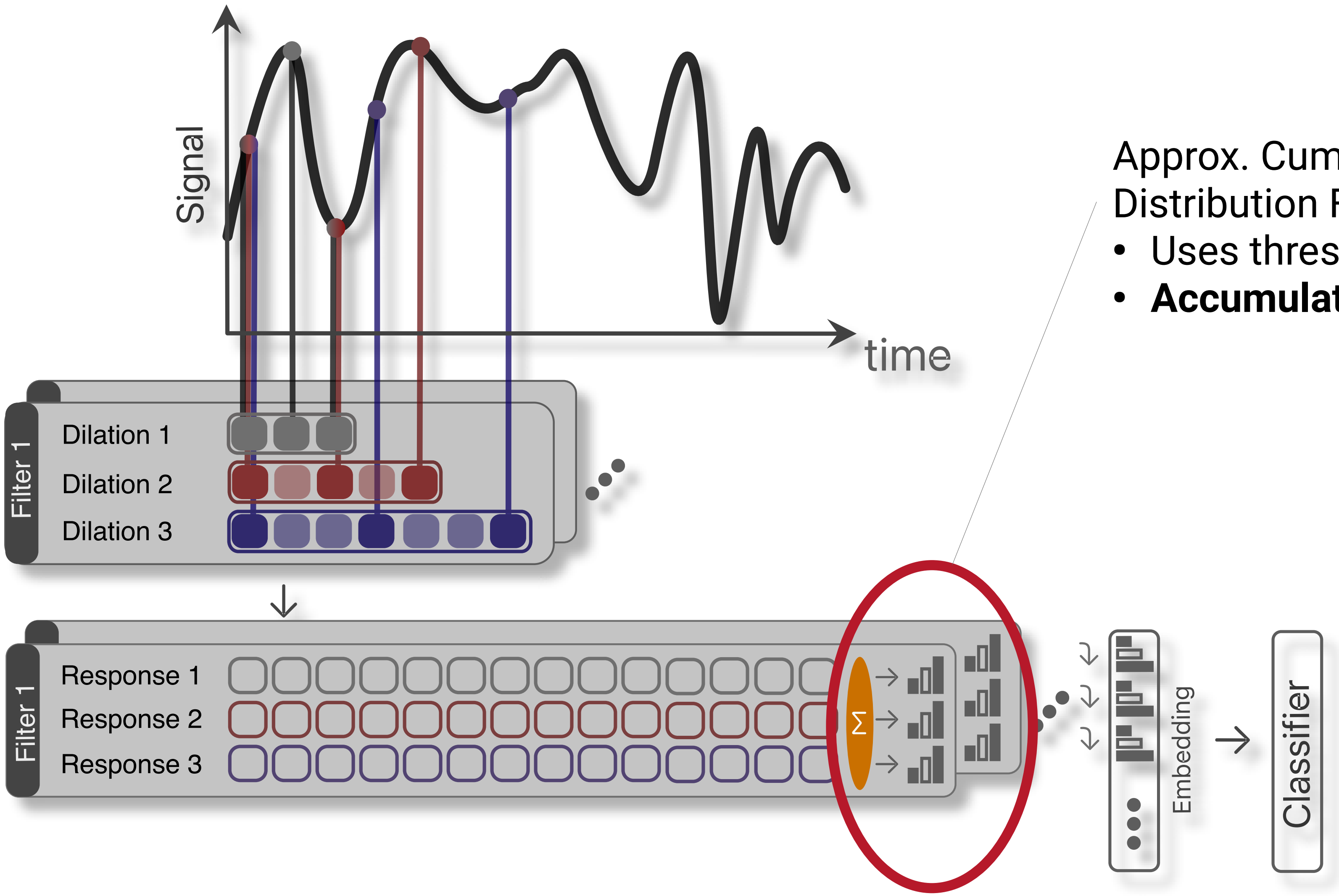
Random guess: 50%

[1] A. Dempster, D. F. Schmidt, and G. I. Webb, "MiniRocket: A Very Fast (Almost) Deterministic Transform for Time Series Classification," Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., pp. 248–257, 2021.

The State-of-the-Art - MiniROCKET



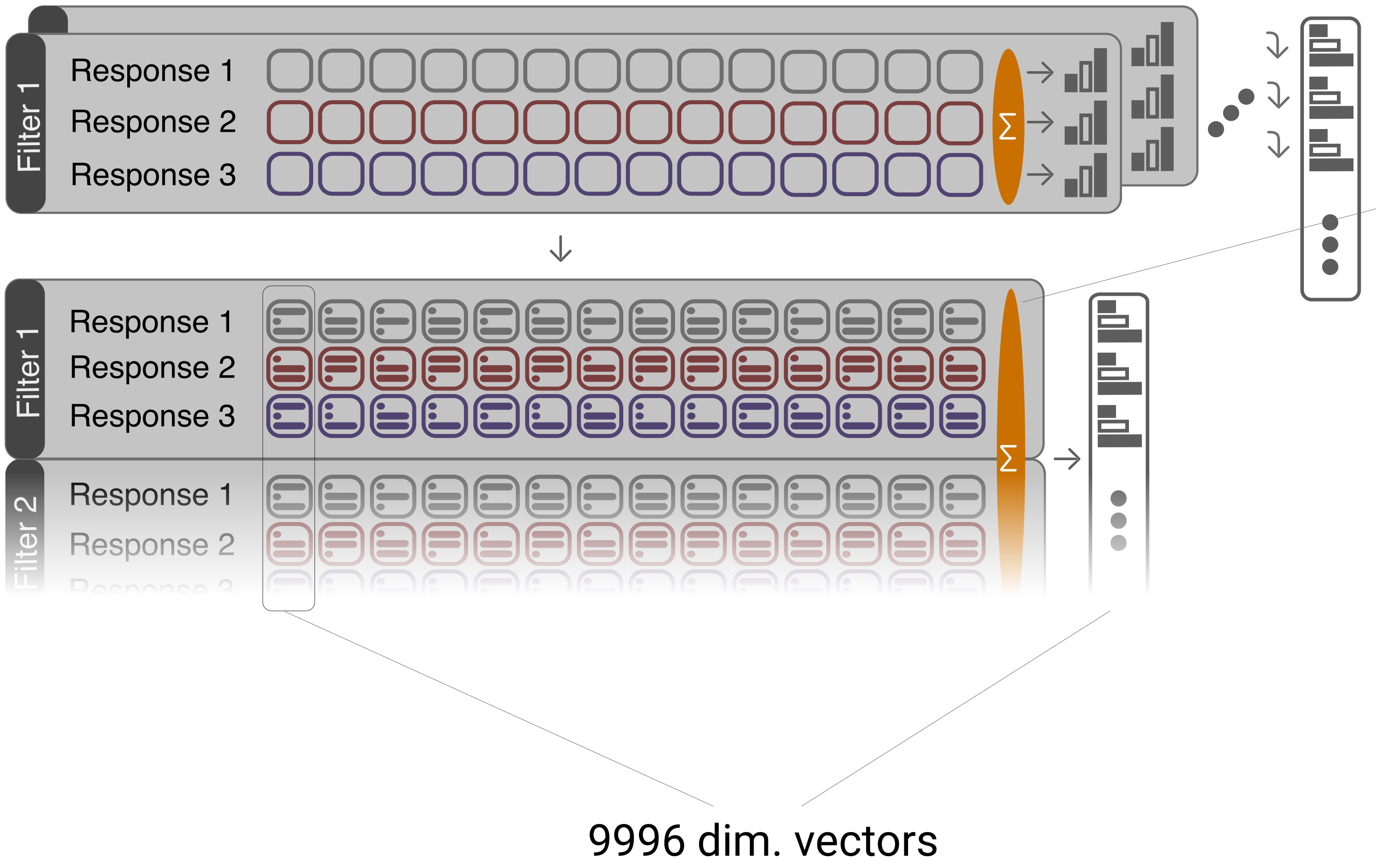
The State-of-the-Art - MiniROCKET



Approx. Cumulative Distribution Function

- Uses thresholds for the bins
- **Accumulate** these over time

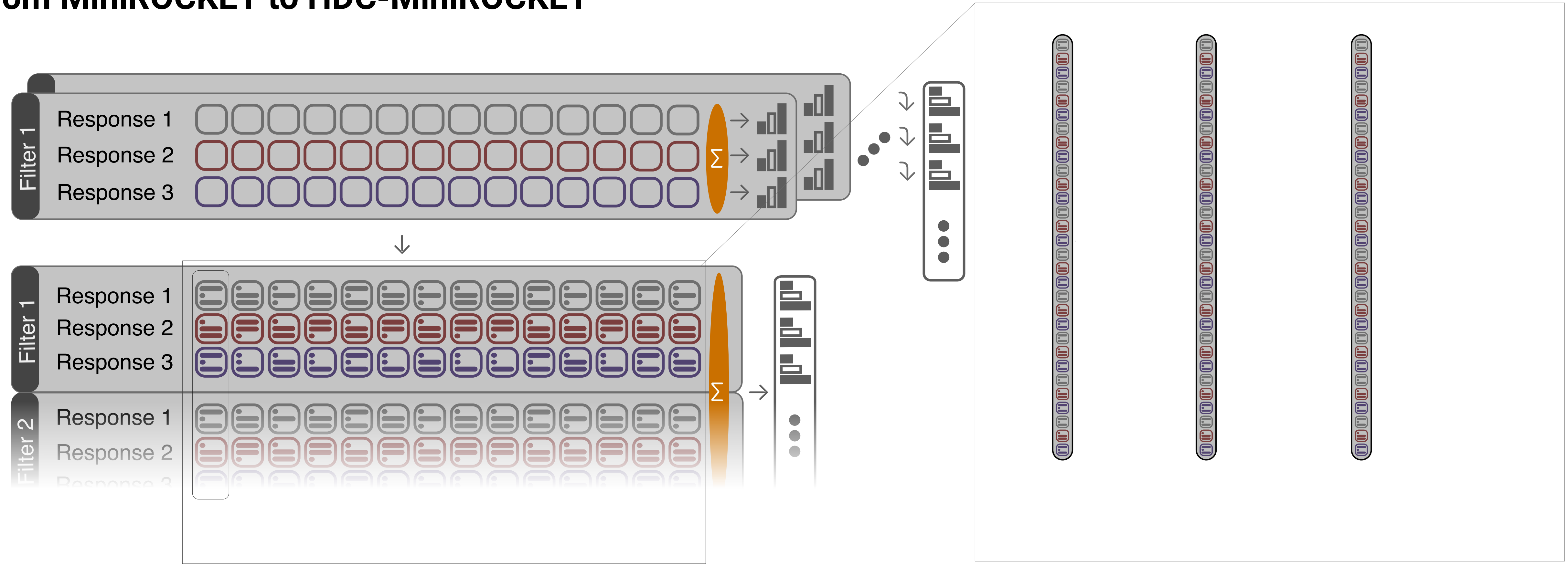
Intermediate Representation of MiniROCKET



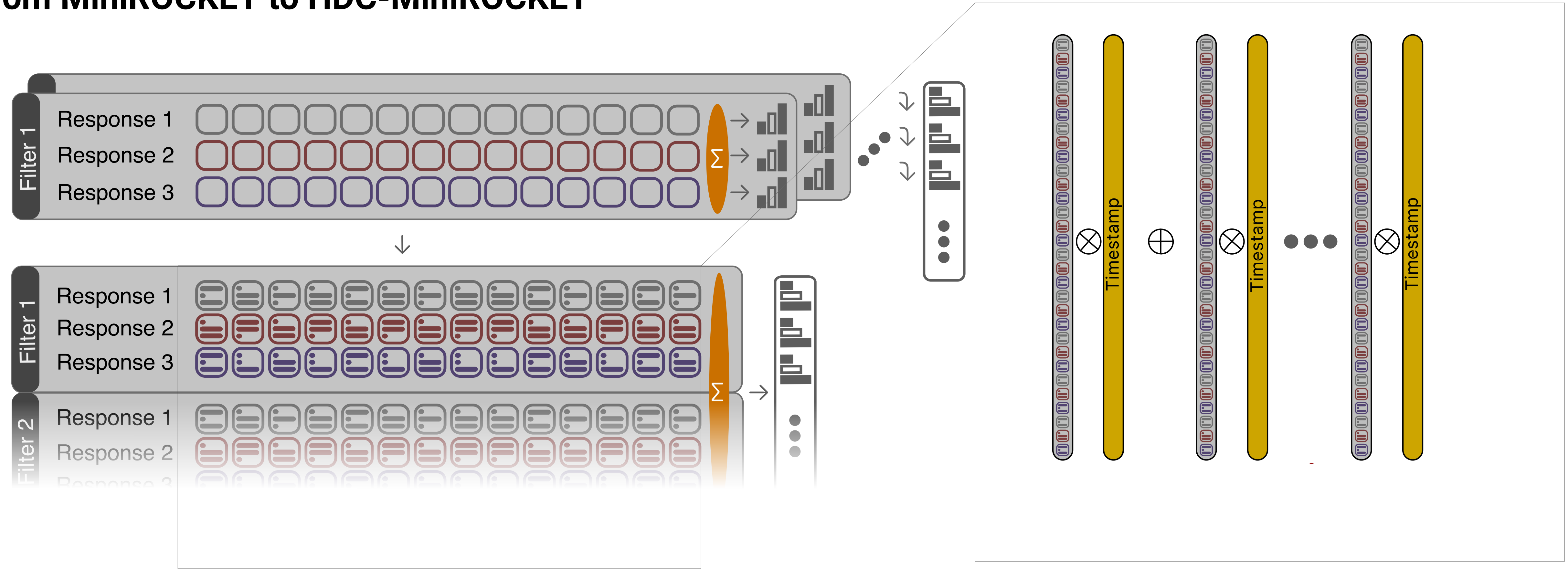
Accumulation loses information about location in time

Encode global time context
Before accumulation

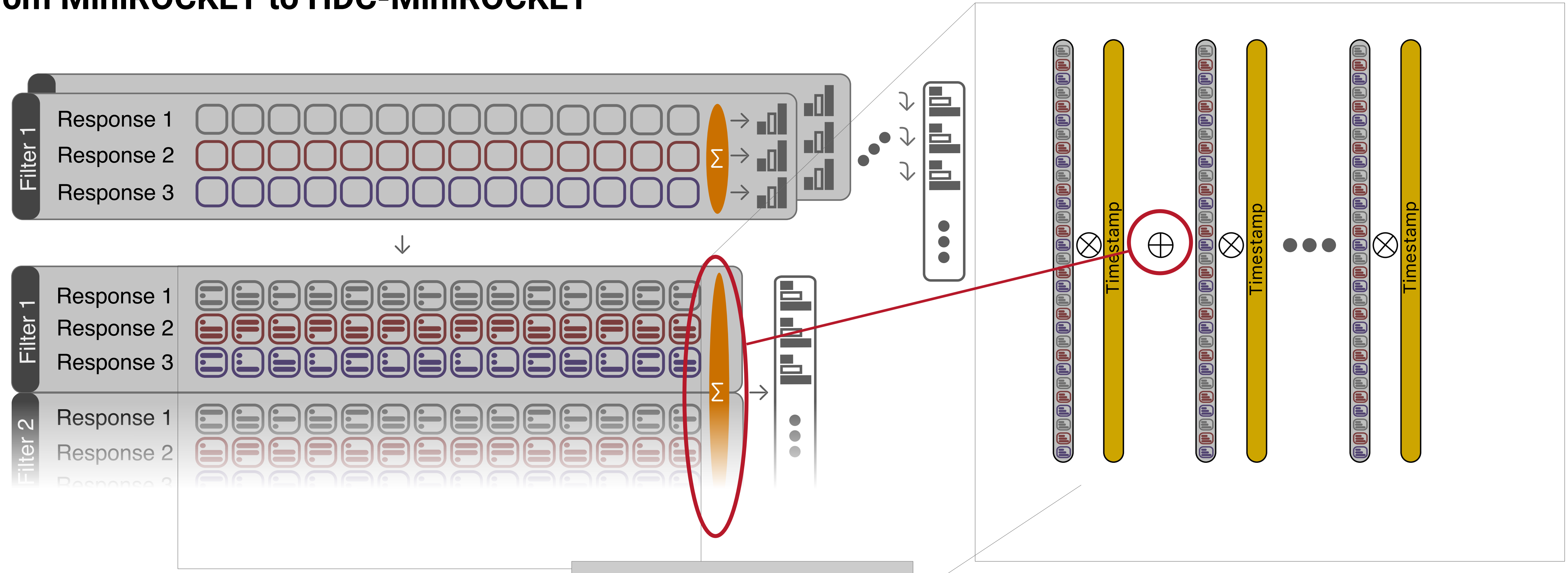
From MiniROCKET to HDC-MiniROCKET



From MiniROCKET to HDC-MiniROCKET



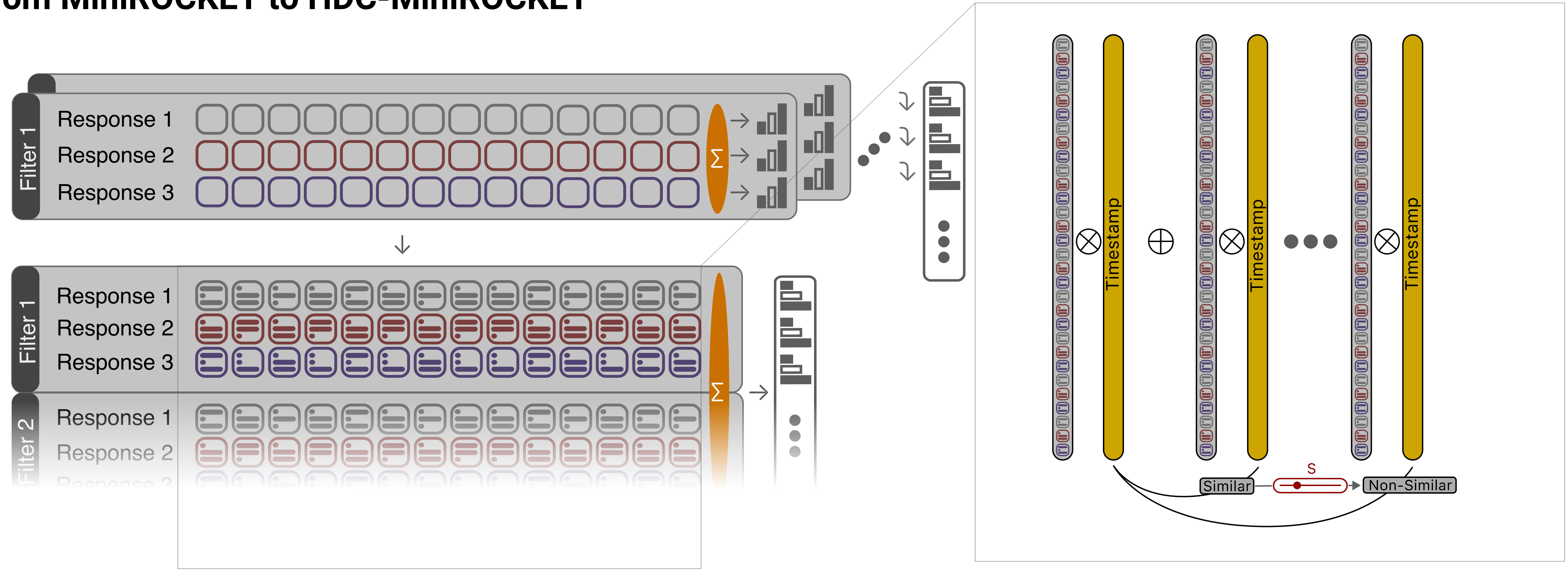
From MiniROCKET to HDC-MiniROCKET



Assign each feature Vector with its position

Bundled Role-Filler-Pairs

From MiniROCKET to HDC-MiniROCKET



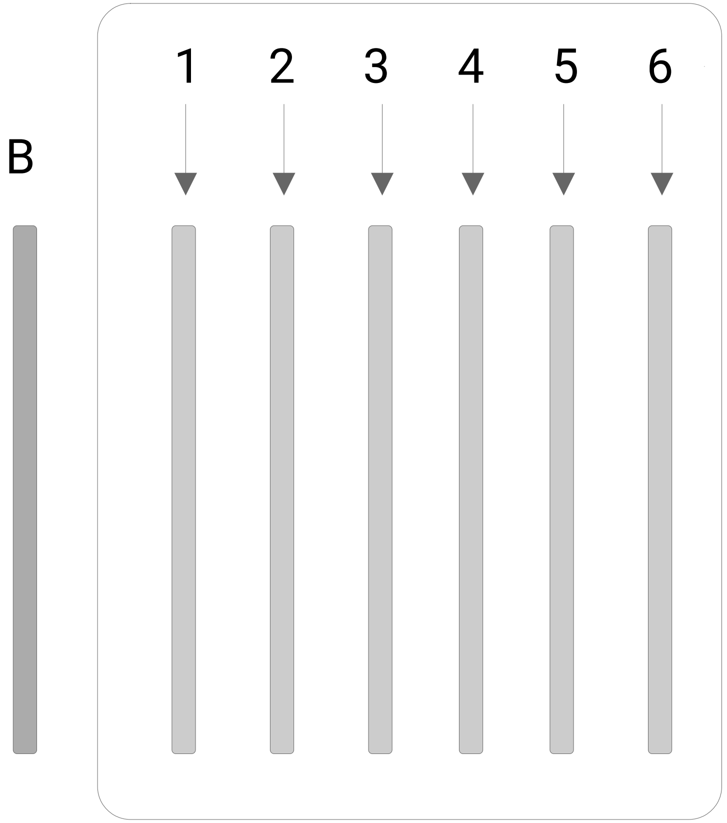
The goal is graded similarity: **nearby** timestamps should be **similar** and more **distant** timestamps should be **dissimilar**

From MiniROCKET to HDC-MiniROCKET

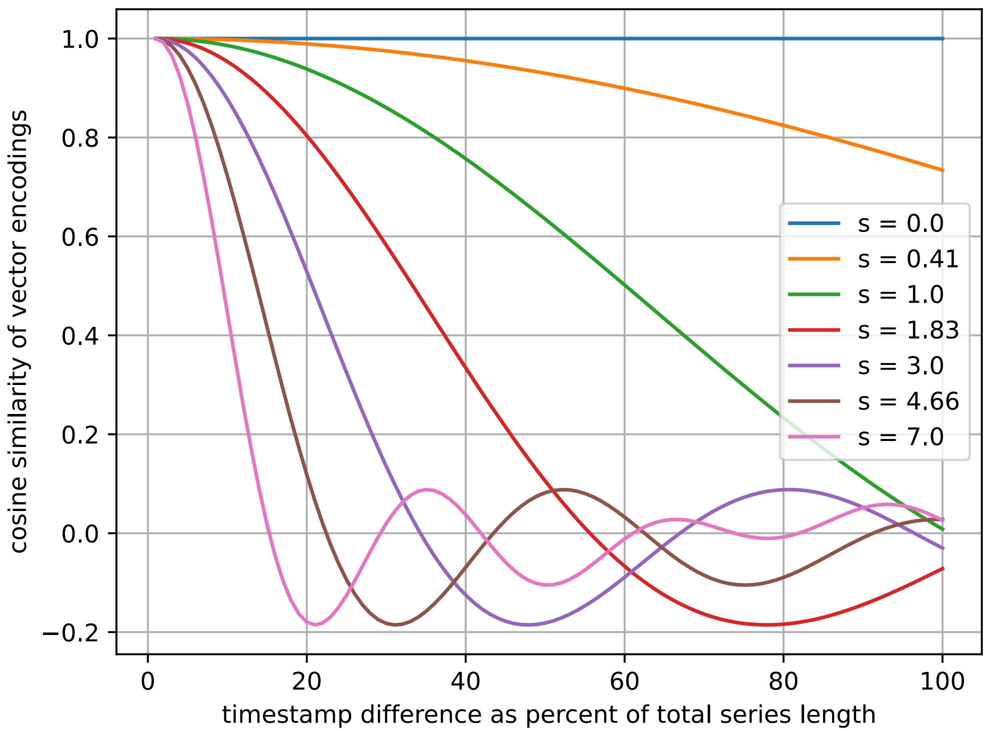
Mechanism called fractional binding creates graded similarity: $fracBind_B(x) := B^{\lambda \cdot x}$

Euclidean distance
Of scalar values x

Cosine distance
Of vector representation



Different Graded Similarities for timestamp encoding

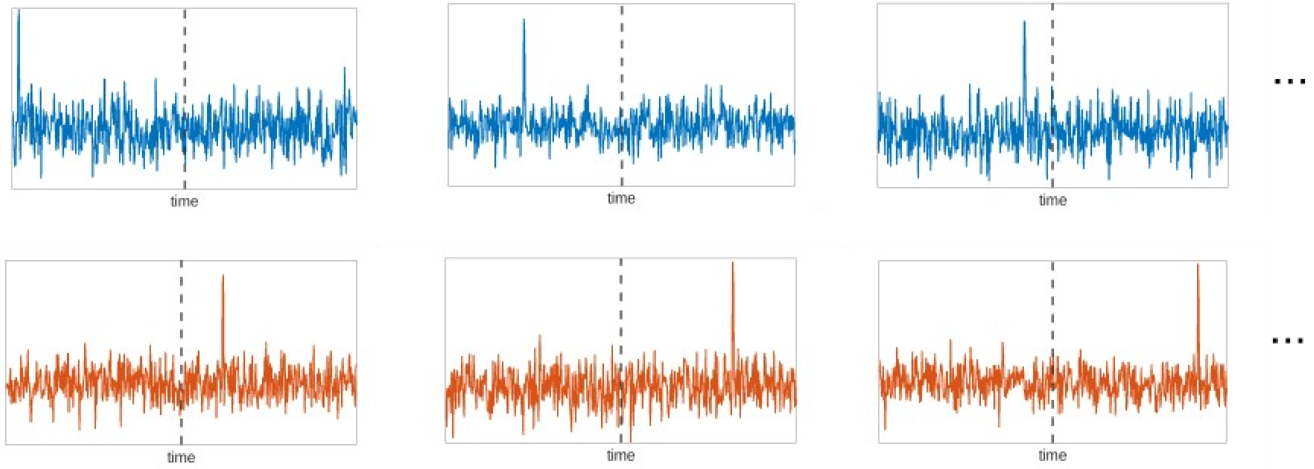


Results of HDC-MiniROCKET on synthetic Dataset

A simple 2-class time series classification problem:
Gaussian noise with a single sharp peak either in the
first half (class 1) or second half (class 2) of the signal

examples class 1

examples class 2



Classification Accuracy of MiniROCKET:

Standard Case:
67 %

Challenging Case:
56.9 %

Classification Accuracy of HDC-MiniROCKET:

Standard Case:
94 %

Challenging Case:
94 %

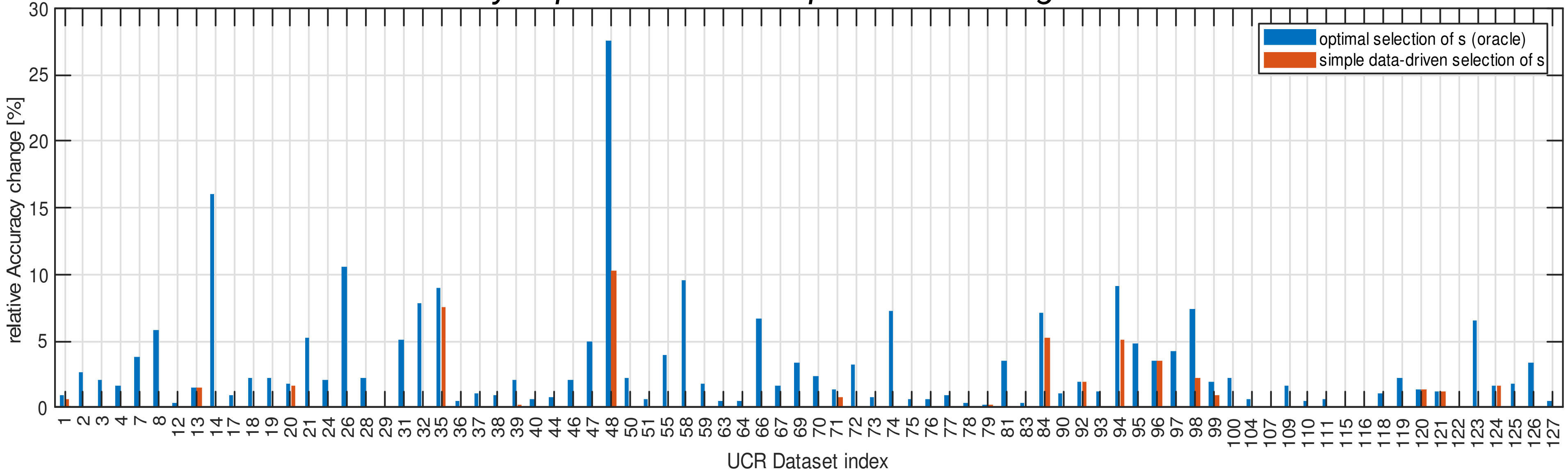
Results of HDC-MiniROCKET on Benchmark ensemble

- UCR Benchmark ensemble with 128 univariate datasets
- Since we have a hyperparameter s, we need to find the optimal s for each dataset
- Choosing the correct s is important because some datasets are getting worse with additional time encoding

Oracle (best case)

Cross validation

Relative accuracy improvements compared to the original MiniROCKET



- 17 of 128 datasets show improvements
- Maximal 27% improvement with oracle and 12% with cross val.

HDC-MiniROCKET

More details in the recent paper [1]

HDC-MiniROCKET: Explicit Time Encoding in Time Series Classification with Hyperdimensional Computing

Kenny Schlegel, Peer Neubert and Peter Protzel
Chemnitz University of Technology
Chemnitz, Germany
{kenny.schlegel, peer.neubert, peter.protzel}@etit.tu-chemnitz.de

***Abstract*—Classification of time series data is an important task for many application domains. One of the best existing methods for this task, in terms of accuracy and computation time, is MiniROCKET. In this work, we extend this approach to provide better global temporal encodings using hyperdimensional computing (HDC) mechanisms. HDC (also known as Vector Symbolic Architectures, VSA) is a general method to explicitly represent and process information in high-dimensional vectors. It has previously been used successfully in combination with deep neural networks and other signal processing algorithms. We argue that the internal high-dimensional representation of MiniROCKET is well suited to be complemented by the algebra of HDC. This leads to a more general formulation, HDC-MiniROCKET, where the original algorithm is only a special case. We will discuss and demonstrate that HDC-MiniROCKET can systematically overcome catastrophic failures of MiniROCKET on simple synthetic datasets. These results are confirmed by experiments on the 128 datasets from the UCR time series classification benchmark. The extension with HDC can achieve considerably better results on datasets with high temporal dependence at about the same computational effort for inference.**

***Index Terms*—time series classification, HDC, VSA, hyperdimensional computing**

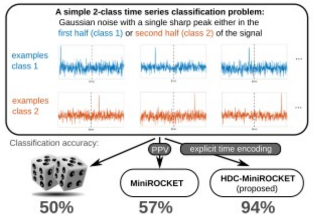


Fig. 1: MiniROCKET is a fast state-of-the-art approach for time series classification. However, it is easy to create simple datasets where its performance is similar to random guessing. The proposed HDC-MiniROCKET uses explicit time encoding to prevent this failure at almost the same computational costs.

I. Introduction

Time series classification has a wide range of applications in robotics, autonomous driving, medical diagnostic, in the financial sector, and so on. As elaborated in [1], classification of time series differs from traditional classification problems because the attributes are ordered. Hence, it is crucial to create discriminative and meaningful features with respect to the specific order in time. Over the past years, various methods for classification of univariate and multivariate time series have been proposed (for instance, [2]–[11]). Often, a high accuracy of a method comes at the cost of a high computational effort. A very noticeable exception is MiniROCKET [9] which superseded the earlier ROCKET [8] and achieves state-of-the-art accuracy at very low computational complexity. Similar to a convolutional neural network (CNN) layer, MiniROCKET applies a set of parallel convolutions to the input signal. To achieve a low runtime, two important design decisions of MiniROCKET are (1) the usage of convolution filters of small size and (2) accumulation of filter responses over time based on the *Proportion of Positive Values (PPV)*, which is a special kind of averaging. However, the combination of these design decisions can hamper the encoding of temporal variation of signals on a larger scale than the size of the convolution filters. To address this, the authors of MiniROCKET propose to use *dilated convolutions*. A dilated convolution virtually increases a filter kernel by adding sequences of zeros in between the values of the original filter kernel [12] (e.g. [-1 2 1] becomes [-1 0 2 0 1] or [-1 0 0 2 0 0 1] and so on).

The first contribution of this paper is to demonstrate that although the dilatated convolutions of MiniROCKET perform well on a series of standard benchmark datasets like UCR [13], it is easy to create datasets where classification based on MiniROCKET is not much better than random guessing. An example is illustrated in Fig. 1. There, the task is to distinguish two different classes of time series signals. Each consists of Gaussian noise and a single sharp peak either in the first half of the signal (for the first class) or in the second half of the signal (for each sample from the second class). Since this is a 2-class problem, random guessing of the class of a query signal

We want to thank Denis Kleyko from the University of California, Berkeley, for his contribution to the automatic selection of the scale parameter.

Ongoing work:

- Different similarity kernels for timestamps
- Use HDC for recent variants of MiniROCKET
- Want to extend to multivariate domain with channel combination with VSA

[1] K. Schlegel, P. Neubert, and P. Protzel, “HDC-MiniROCKET: Explicit Time Encoding in Time Series Classification with Hyperdimensional Computing,” in 2022 International Joint Conference on Neural Networks (IJCNN), 2022, no. 1, pp. 1–8.

Which place from the database is shown in the query image?

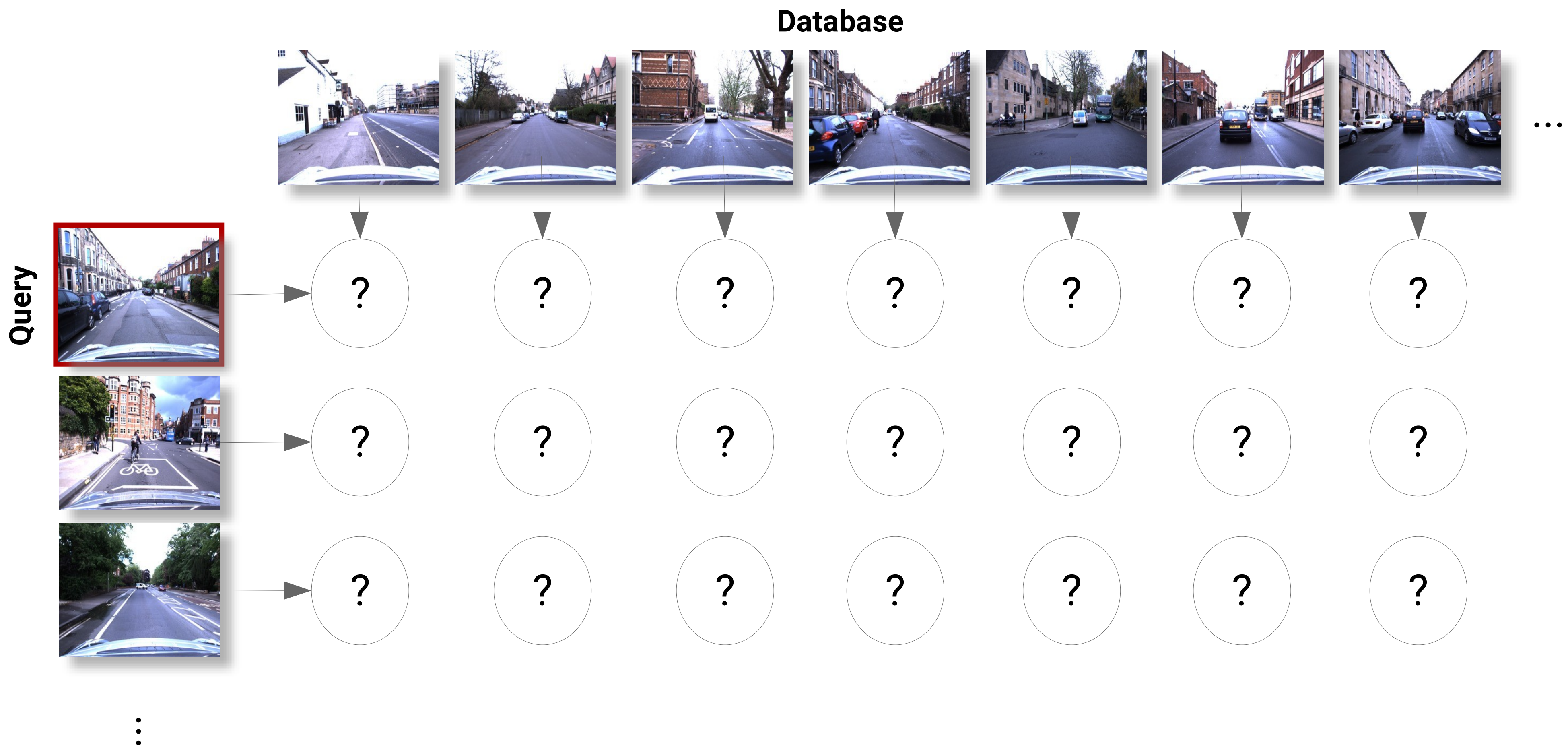
Query

Database

The role of descriptors

Is this the same place?

The role of descriptors



The role of descriptors

Holistic Descriptors

A single descriptor for the whole image

descriptor Comparisons: $\mathbf{m * n}$

Local Descriptors

A set of k local descriptors for each image
e.g. X/Y coordinate, semantic, ...

descriptor Comparisons: $\mathbf{m * n * k^2}$

The role of descriptors

Holistic Descriptors

A single descriptor for the whole image

descriptor Comparisons: **m * n**

efficient

Local Descriptors

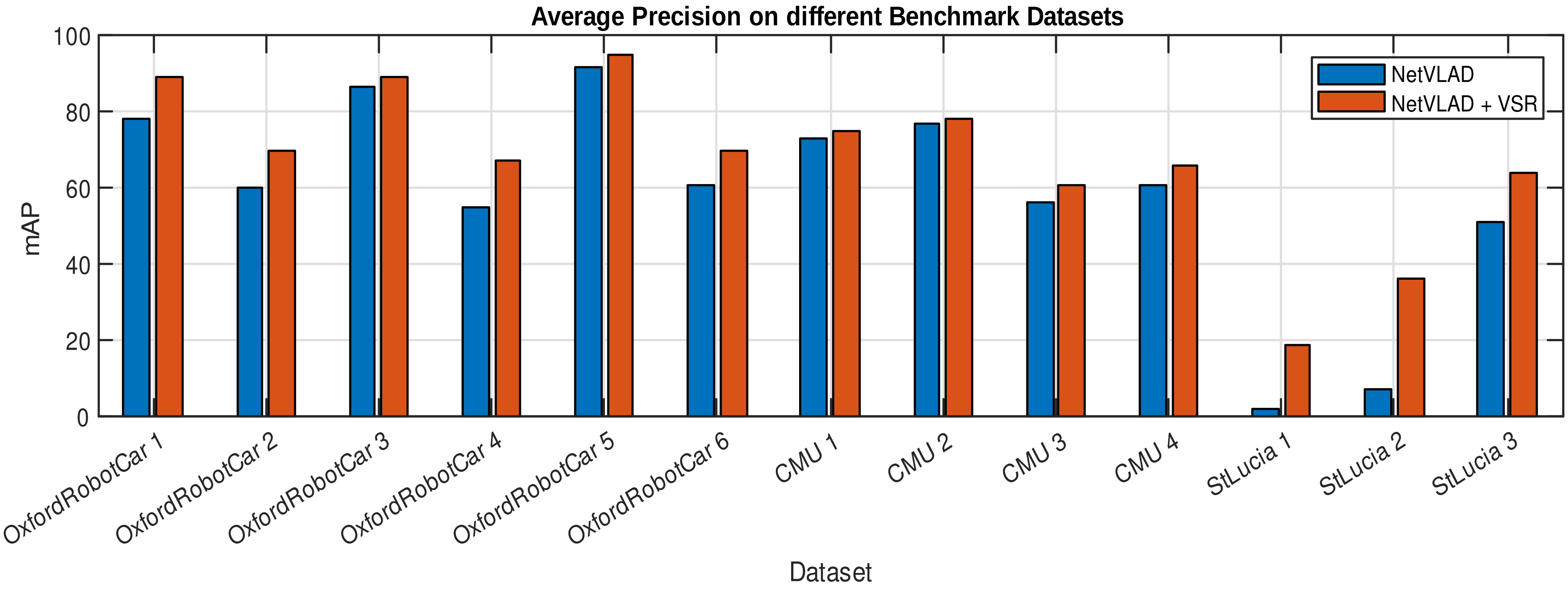
A set of k local descriptors for each image
e.g. X/Y coordinate, semantic, ...

descriptor Comparisons: $\mathbf{m * n * k^2}$

better recognition performance

not efficient

Results on Benchmark



Publications of this approach

Hyperdimensional computing as a framework for systematic aggregation of image descriptors

Peer Neubert and Stefan Schubert
Chemnitz University of Technology
firstname.lastname@etit.tu-chemnitz.de

Abstract

Image and video descriptors are an omnipresent tool in computer vision and its application fields like mobile robotics. Many hand-crafted and in particular learned image descriptors are numerical vectors with a potentially (very) large number of dimensions. Practical considerations like memory consumption or time for comparisons call for the creation of compact representations. In this paper, we use hyperdimensional computing (HDC) as an approach to systematically combine information from a set of vectors in a single vector of the same dimensionality. HDC is a known technique to perform symbolic processing with distributed representations in numerical vectors with thousands of dimensions. We present a HDC implementation that is suitable for processing the output of existing and future (deep learning based) image descriptors. We discuss how this can be used as a framework to process descriptors together with additional knowledge by simple and fast vector operations. A concrete outcome is a novel HDC-based approach to aggregate a set of local image descriptors together with their image positions in a single holistic descriptor. The comparison to available holistic descriptors and aggregation methods on a series of standard mobile robotics place recognition experiments shows a 20% improvement in average performance and $> 2\times$ better worst-case performance compared to runner-up.

1. Introduction

Image descriptors are very useful tools for recognition tasks in computer vision. Many hand-crafted and in particular deep learning based descriptors are numerical vectors with a potentially large number of dimensions, e.g. NetVLAD [1] uses 4,096-D vectors (after PCA), DELF [44] uses 1,024-D vectors (before PCA). Approaches like BoW [56], VLAD [23], or ASMK [60] aggregate the information from multiple vectors in a single holistic vector representation to reduce memory consumption and computational efforts during comparison. For example, deciding whether two images show the same place based on a set of local landmarks from each image can then be done by a single distance measure between the two aggregated vectors. Although these techniques are able to combine large numbers of descriptors in a compact vector, for certain tasks like place recognition, it is beneficial to encode additional information in the final vector representation, e.g., information about the image locations of aggregated vectors.

The central idea of this paper is to use binding and bundling of vectors as a flexible framework to combine image descriptors and additional information. The underlying technique of binding and bundling vectors is taken from a field known as hyperdimensional computing (HDC) or vector symbolic architectures (VSA). This is an established class of approaches to solve *symbolic* computational problems using mathematical operations on large numerical vectors with thousands of dimensions [25, 42]. The bundling operator $\oplus$ superposes information of a variable number of vectors in a single vector; we can think of it as some form of averaging. The binding operator $\otimes$ can, for example, express role-filler or variable-value pairs as required in symbolic processing. An important property is that the output of the operations are vectors from the same vector space. This allows to chain HDC operations and enables versatile encoding of structured data from a set of d-dimensional vectors in a single d-dimensional vector.

We will present a HDC implementation that allows the processing of existing and future (deep learning based) image descriptors in Sec. 3. This section will also describe how HDC can be used as a framework to aggregate holistic or local image descriptors and to combine them with additional information. A concrete outcome is a novel approach to create a holistic image descriptor from a set of local descriptors with image position information in Sec. 3.2.2. For example, we can create a holistic descriptor from three local descriptors L_1, L_2, L_3 with poses P_1, P_2, P_2 as simple as $(L_1 \otimes P_1) \oplus (L_2 \otimes P_2) \oplus (L_3 \otimes P_3)$. The poses serve as "roles" that are associated with landmarks as "fillers". When comparing two such holistic descriptors (e.g. based

You can also contact Peer Neubert for more detailed information in this field

Vector Semantic Representations as Descriptors for Visual Place Recognition

Peer Neubert, Stefan Schubert, Kenny Schlegel and Peter Protzel
Chemnitz University of Technology, Germany
{peer.neubert, stefan.schubert, kenny.schlegel, peter.protzel}@etit.tu-chemnitz.de

***Abstract*—Place recognition is the task of recognizing the current scene from a database of known places. The currently dominant algorithmic paradigm is to use (deep learning based) holistic feature vectors to describe each place and use fast vector query methods to find matchings. We propose a novel type of image descriptor, Vector Semantic Representations (VSR), that encodes the spatial semantic layout from a semantic segmentation together with appearance properties in a, for example, 4,096 dimensional vector for place recognition. We leverage operations from the established class of Vector Symbolic Architectures to combine symbolic (e.g. class label) and numeric (e.g. feature map response) information in a common vector representation. We evaluate the proposed semantic descriptor on 13 standard mobile robotic place recognition datasets and compare to six descriptors from the literature. VSR is on par with the best compared descriptor (NetVLAD) in terms of mean average precision and superior in terms of recall and worst-case average precision. This makes the approach particularly interesting for candidate selection. For a more detailed investigation, we discuss and evaluate recall integrity as additional criterion. Further, we demonstrate that the semantic descriptor is particularly well suited for combination with existing appearance descriptors indicating that semantics provide complementary information for image matching.**

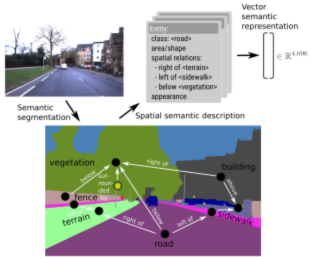


Fig. 1. A Vector Semantic Representation (VSR) is a single high-dimensional vector that combines information of semantic entities, their spatial layout, and appearance. We describe how a semantic segmentation can be used to create the entities and how operations from Vector Symbolic Architecture (VSA) can be used to encode this mixed symbolic-numeric information in a single vector that can serve as descriptor for place recognition.

I. INTRODUCTION

Visual place recognition is the task of matching a given query image to a potentially large database of known places. It is an important means for loop closure detection in SLAM and for candidate selection for 6-D pose estimation [56]. This task becomes particularly challenging when the environmental condition changes due to changing illumination, weather, or season, and/or when the size of the datasbase becomes very large. Intuitively, information about the *semantic* content of the image can help in both directions. On one hand, semantic is largely invariant of appearance chances. A snow covered tree is still a tree. Here, recent and future developments from (deep) learned models to capture semantics can be leveraged. On the other hand, to address a large-scale database, one can use the semantic gist of a scene for a coarse categorization, e.g. into urban or rural scenes (think of the seminal GIST [48] paper). After such coarse categorizations, e.g. into an urban scene, one can conduct more fine grained semantic categorization using salient semantic landmarks (e.g the Eiffel tower) or other semantic features like the architectonic style of the buildings (think of the "What makes Paris look like Paris?" paper [13]). However, an largely open question is, how can we further exploit semantics together with fine-grained appearance properties for fast image matching, e.g. how to distinguish individual urban street scenes with a high proportion of similarly looking Victorian style buildings?

In this paper we propose a novel approach to encode the spatial semantic layout of images for place recognition. An example is shown in Fig 1. The key idea is to describe the shown street scene by the semantic information that there is a sidewalk right to the street and grass terrain to the left, which in turn is followed by another sidewalk and a fence. We use a deep learning based semantic segmentation model to extract a list of semantic entities. This list includes objects with well defined shape and boundary ("things" [5], e.g. a sign), as well as amorphous background regions ("stuff" [5], e.g. terrain). Each entity is described by its semantic class, a coarse representation of its shape and location, and a list of its spatial semantic relations, e.g. "left-of <sidewalk>". This is complemented by an appearance descriptor based on salient feature map responses.

Given this list of entities, each with combined symbolic (e.g. class) and numeric (e.g. feature map response) information, the major challenge becomes to generate a descriptor that allows fast matching of these image representations. We propose

[1] P. Neubert and S. Schubert, "Hyperdimensional computing as a framework for systematic aggregation of image descriptors," Conference on Computer Vision and Pattern Recognition (CVPR), 2021
[2] P. Neubert, S. Schubert, K. Schlegel, and P. Protzel, "Vector Semantic Representations as Descriptors for Visual Place Recognition," in Robotics: Science and Systems XVII, 2021.

Take home lessons

Using high-dimensional Vectors for symbolic representation increase robustness

Algebra of HDC for systematically encoding is interpretable

HDC has potential of bridging the gap between symbolic and sub-symbolic AI

Real application can benefit from systematically encoded explicit Information (e.g. bundled role-filler-pairs)

Limitations:

- Encoding of real data into HD-vectors can be difficult
- Creating the encoding structure with HDC is still hand-crafted (expert knowledge)
- HDC theory works best with unrelated entities (random vectors) → real world problems are often not like this
- Bundling creates statistics in which frequently occurring symbols can suppress rare symbols (problem if rare events are the discriminative ones for classification)

More information about our work at:

tu-chemnitz.de/etit/proaut/vsa