# Fractional Binding in Vector Symbolic Architectures as Quasi-Probability Statements

**P. Michael Furlong**, Chris Eliasmith

29 September 2022

UNIVERSITY OF WATERLOO


VSAs can be a probabilistic programming language

# Benefits of the Approach

1. Vector Symbolic Architectures (VSAs) can write cognitive models that incorporate probability
    a. Provides hypotheses about uncertainty in the brain.
    b. Gaps between models and behaviour can guide future work.
2. We can code probability statements to neural networks
    a. Exploit low-power computation of neuromorphics.
    b. Apply sophisticated AI in edge computing cases.

UNIVERSITY OF WATERLOO

# Outline

1. **Preliminaries**
    a. **Kernel Density Estimators**
    b. Vector Symbolic Architectures
2. Core Idea
3. Practical Applications
4. Conclusions

# Kernel Density Estimators

From a dataset

$$\mathcal{D} = (x_1, x_2, \ldots, x_n)$$

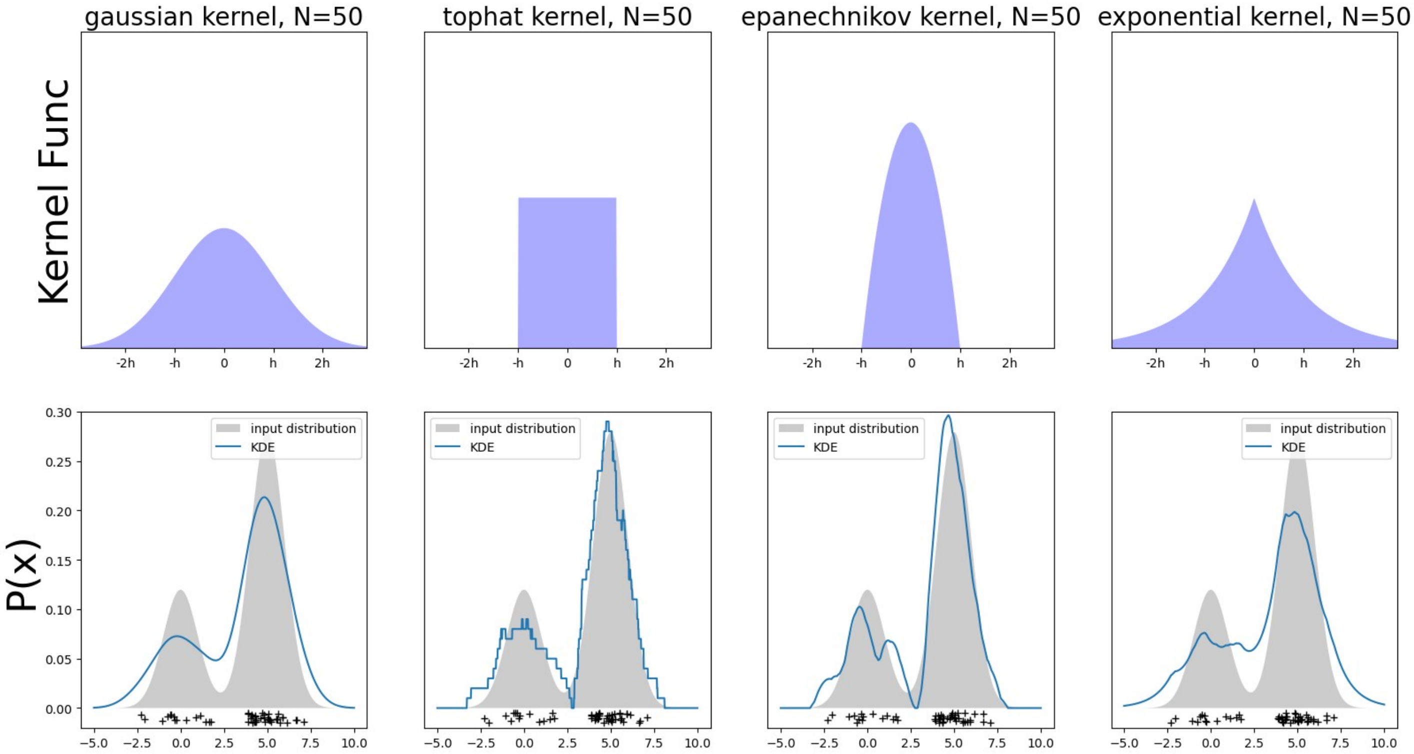
Using a kernel function

$$k_h(x, x') = k\left(\frac{\|x - x'\|}{h}\right)$$

We can estimate the probability of x

$$P_{\mathcal{D}}(X = x) = \frac{1}{nh} \sum_{x_i \in \mathcal{D}} k_h(x, x_i)$$

# Kernel Measures Similarity



# “Problems” With KDEs

The amount of memory grows linearly with the number of observations

The time to compute a probability grow linearly with # of observations

# The right feature space approximates kernels

$$k_h(\mathbf{x}, \mathbf{x}') \approx \phi_h(\mathbf{x}) \cdot \phi_h(\mathbf{x}')$$

(Rahimi & Recht, 2007)

# The Kernel Trick makes efficient Kernel Machines

$$P(X = \mathbf{x}) = \frac{1}{\quad} \sum \phi_h(\mathbf{x}) \cdot \phi_h(\mathbf{x}_i)$$

Vector Symbolic Architectures provide a neurally plausible feature space

$$P(X = \mathbf{x}) = \phi_h(\mathbf{x}) \cdot M_{\mathcal{D}}$$

# Outline

1. **Preliminaries**
   a. Kernel Density Estimators
   **b. Vector Symbolic Architectures**
2. Core Idea
3. Practical Applications
4. Conclusions

# Vector Symbolic Architectures

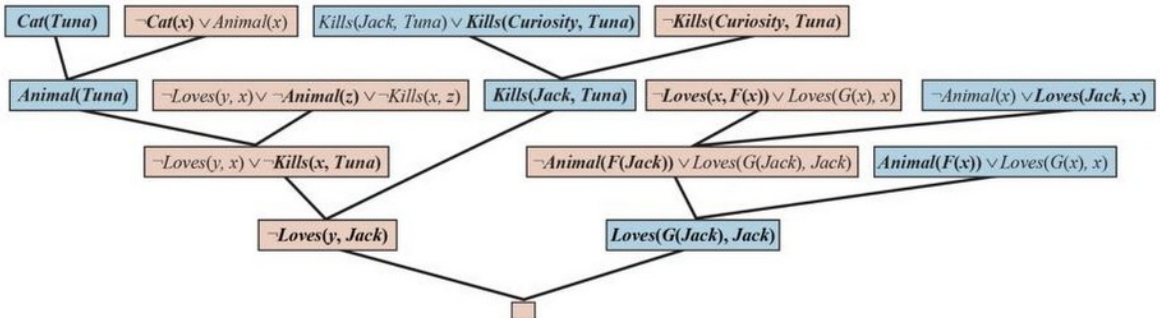
Symbolic Models

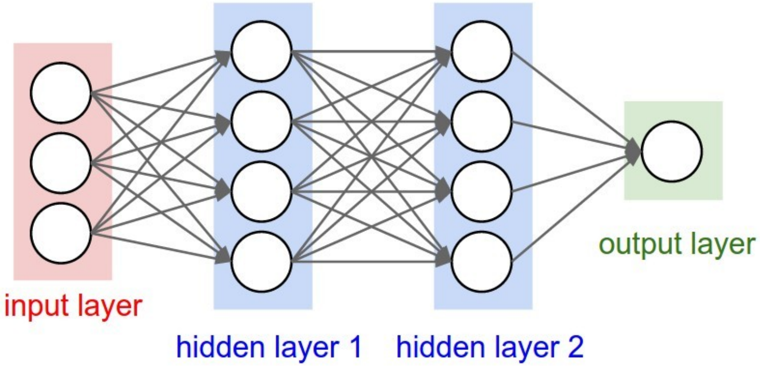
⇕

VSA Statements

⇓

Connectionist Models



# Symbols are Represented by Vectors

It is a mapping of symbols to vectors:

$$\phi : X \to \mathbb{R}^d$$

Such that

$$\forall x_1, x_2 \in X$$

$$x_1 = x_2 \implies sim(\phi(x_1), \phi(x_2)) \approx 1$$

$$x_1 \neq x_2 \implies sim(\phi(x_1), \phi(x_2)) \approx 0$$

# Operators in Vector Symbolic Architectures

| | | |
|---|---|---|
| Similarity | How similar is A to B? | $\mathrm{sim}(A, B)$ |
| Bundling | ORANGE is similar to RED and similar to YELLOW | $\mathrm{ORANGE} = \mathrm{RED} + \mathrm{YELLOW}$ |
| Binding | STOP is only similar RED and OCT, together | $\mathrm{STOP} = \mathrm{RED} \otimes \mathrm{OCT}$ |
| Unbinding | RED is the other element of STOP | $\mathrm{RED} = \mathrm{STOP} \oslash \mathrm{OCT}$ |

*A Comparison of Vector Symbolic Architectures* K. Schlegel, P. Neubert, & P. Protzel

# VSA Operator Implementations We Used

| | | |
|---|---|---|
| Similarity | Dot product | $\phi(x) \cdot \phi(y)$ |
| Bundling | Vector addition | $\phi(x) + \phi(y)$ |
| Binding | Circular Convolution | $\phi(x) \circledast \phi(y)$ |
| Unbinding | Circular Correlation/Involution | $\phi^{-1}(x) \circledast \phi(x) = 1$ |

(Plate, 2003; Eliasmith, 2013)

# Representing Integers in VSAs

$$
\begin{aligned}
\phi[n] &= \mathtt{ONE} \circledast \mathtt{ONE} \circledast \ldots \circledast \mathtt{ONE} \\
&= \bigcircledast_{i=1}^{n} \mathtt{ONE} \\
&= \mathcal{F}^{-1}\left\{\mathcal{F}\left\{\mathtt{ONE}\right\}^{n}\right\}
\end{aligned}
$$

UNIVERSITY OF WATERLOO

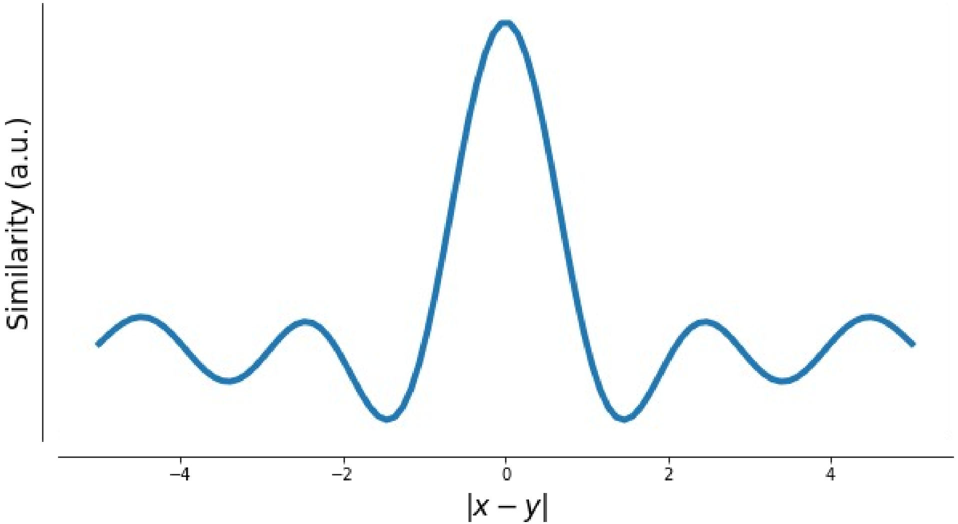
# Fractional Binding - Representing Real and Vector values

$$\phi(x) = \mathcal{F}^{-1}\left\{\mathcal{F}\left\{\mathtt{X}\right\}^{x}\right\}$$

$$\phi\left(\vec{x}\right) = \underset{i=1}{\overset{k}{\circledast}} \phi_i\left(x_i\right)$$

(Plate, 1997; Grosman, 2016; Frady, Kanerva, Sommer, 2018; Komer & Eliasmith, 2020; Dumont & Eliasmith, 2021)

# Dot Product Provides a Usable Kernel



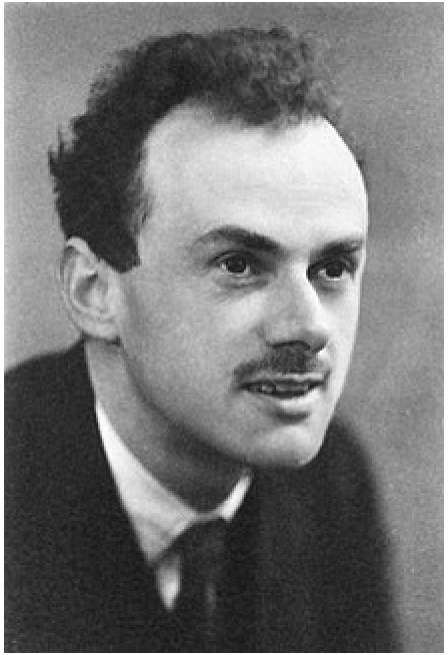

$$\phi(\mathbf{x}) \cdot \phi(\mathbf{y}) \approx \prod_{k=1}^{m} \operatorname{sinc}(|x_k - y_k|)$$

*A short letter on the dot product between rotated Fourier transforms* (2020) AR Voelker

UNIVERSITY OF WATERLOO

**"Negative energies and probabilities should not be considered as nonsense. They are well-defined concepts mathematically, like a negative of money."**

**- Paul Dirac**



# Converting to Probability

$$P_{\mathcal{D}}(X = x) = \max\left\{0, \phi(x) \cdot M_{\mathcal{D},h} - \xi\right\}$$

Glad et al, 2003

$$P_{\mathcal{D}}(X = x) = \left(\phi(x) \cdot \sum_{x_i \in \mathcal{D}} c_i \phi(x_i)\right)^2$$

Agarawal et al., 2016

$$P_{\mathcal{D}}(X = x) = \frac{1}{\eta}\left(\phi(x) \cdot M_{\mathcal{D},h}\right)^2$$

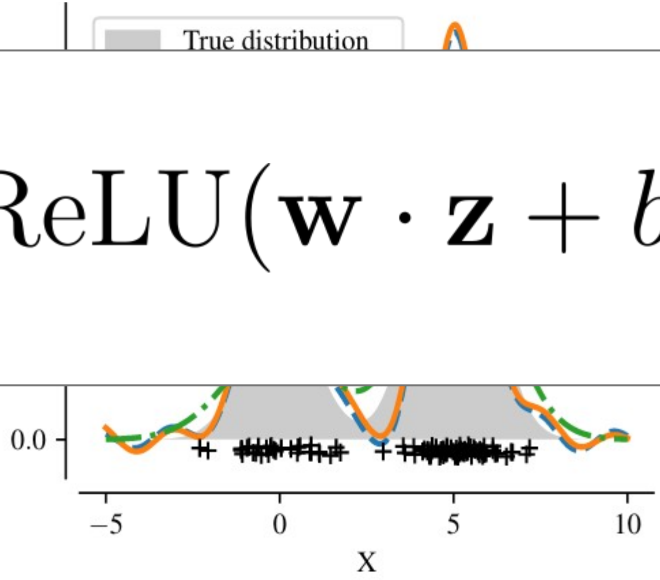
Born Rule*

UNIVERSITY OF WATERLOO

# Converting to Probability

$$P_{\mathcal{D}}(X = x) = \max\{0, \phi(x) \cdot M_{\mathcal{D},h} - \xi\}$$

Glad et al, 2003




$$\mathrm{ReLU}(\mathbf{w} \cdot \mathbf{z} + b)$$

(Frady et al, 2021, Born, 1925)

# Outline

1. Preliminaries
    a. Kernel Density Estimators
    b. Vector Symbolic Architectures
2. **Core Idea**
3. Practical Applications
4. Conclusions

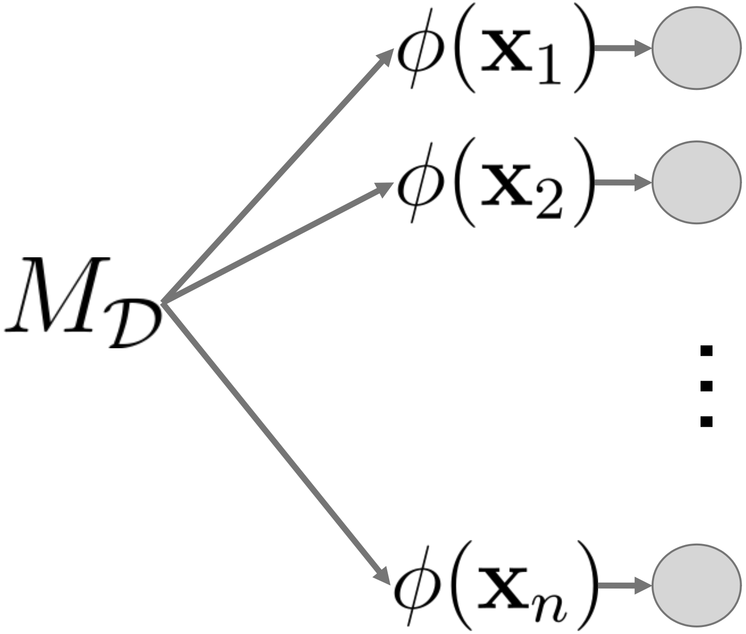
# Memory is a Latent Probability Distribution

$$M_{\mathcal{D}} = \frac{1}{n} \sum_{\mathbf{x}_i \in \mathcal{D}} \phi(\mathbf{x}_i)$$

The distribution is stored in bundles of vector symbols.

We can apply manipulations to bundles to produce probabilistic statements.

# Construct Networks that Estimate Probability Distributions

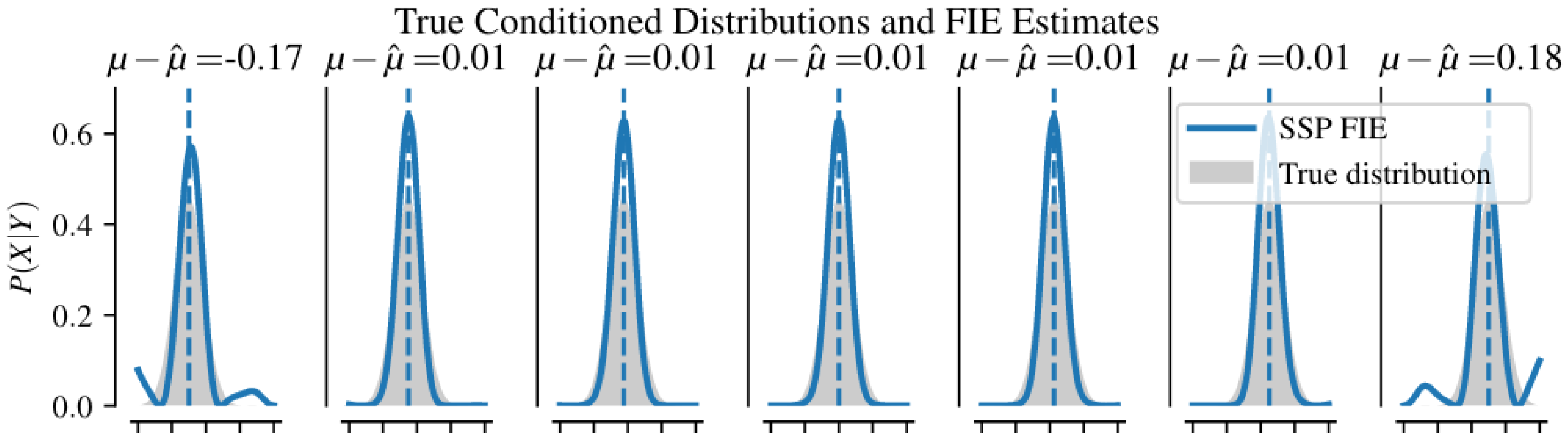



$$a_1 \approx P(X = \mathbf{x}_1 \mid \mathcal{D})$$

$$a_2 \approx P(X = \mathbf{x}_2 \mid \mathcal{D})$$

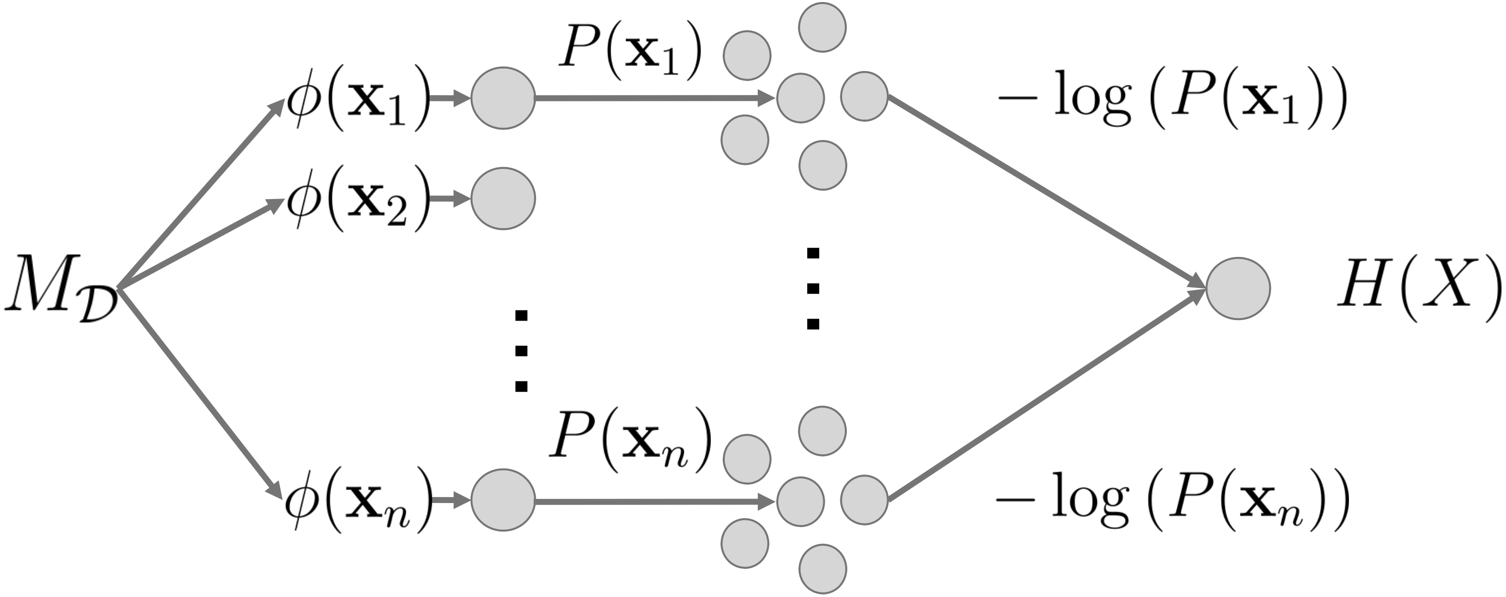
$$a_n \approx P(X = \mathbf{x}_n \mid \mathcal{D})$$

# Unbinding Induces Conditioning



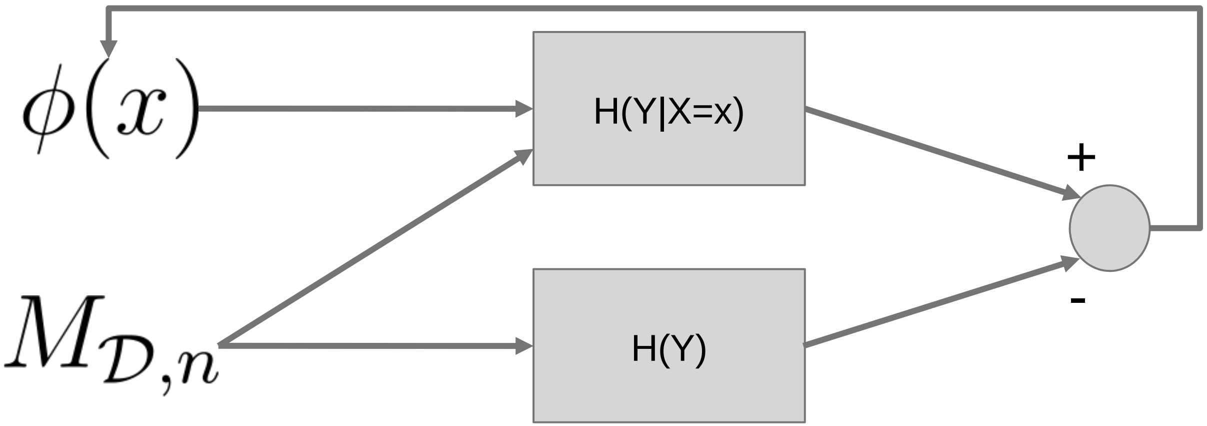

$$P(\mathrm{X}=x \mid \mathrm{Y}=y) \approx \phi_{\mathrm{X}}(x) \cdot \left[M_{\mathcal{D}} \circledast \phi^{-1}(y)\right]$$

# Entropy



$M_{\mathcal{D}}$
$\phi(\mathbf{x}_1)$
$\phi(\mathbf{x}_2)$
$\phi(\mathbf{x}_n)$
$P(\mathbf{x}_1)$
$P(\mathbf{x}_n)$
$-\log\left(P(\mathbf{x}_1)\right)$
$-\log\left(P(\mathbf{x}_n)\right)$
$H(X)$

# Mutual Information



# Other Approaches to Probabilistic Modelling Exist

Where we differ:

1. Provide a general and abstract framework for modelling probabilities
2. Draw a direct connection between cognitive models and probability statements
3. Provide network architectures for conditioning, marginalization, entropy, and mutual information

# Outline

1. Preliminaries
    a. Kernel Density Estimators
    b. Vector Symbolic Architectures
2. Core Idea
3. **Practical Applications**
4. Conclusions

# Applications: Cognitive Modelling

## 4.8 Bayesian

*Explanation target:*

- How does the mind carry out functions such as inference?

*Explanatory pattern:*

- The mind has representations for statistical correlations and conditional probabilities.
- The mind has the capacity for probabilistic computations such as applications of Bayes' theorem.
- Applying probabilistic computations to statistical representations accomplishes mental tasks such as inference.

Although Bayesian methods have had impressive applications to a wide range of phenomena, their psychological plausibility is debatable because of assumptions about optimality and computations based on probability theory.

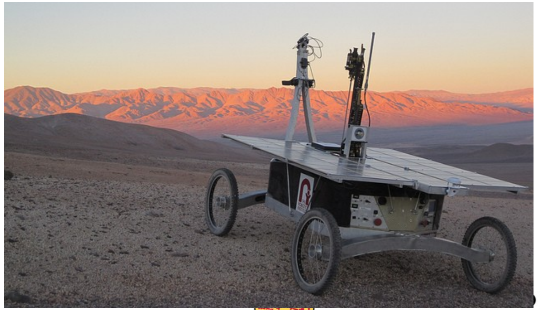
UNIVERSITY OF WATERLOO

Thagard, Paul, "Cognitive Science", The Stanford Encyclopedia of Philosophy (Winter 2020 Edition), Edward N. Zalta (ed.), https://plato.stanford.edu/archives/win2020/entries/cognitive-science/

# Applications: Robotics

# NASA Ames Research Center

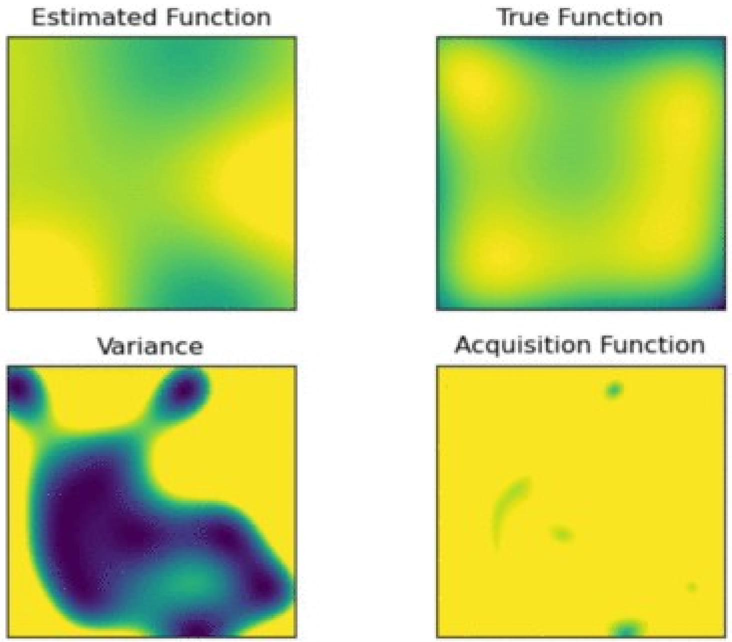
# Energy/Computing Limits Robots

# Bayesian Optimization for Exploration

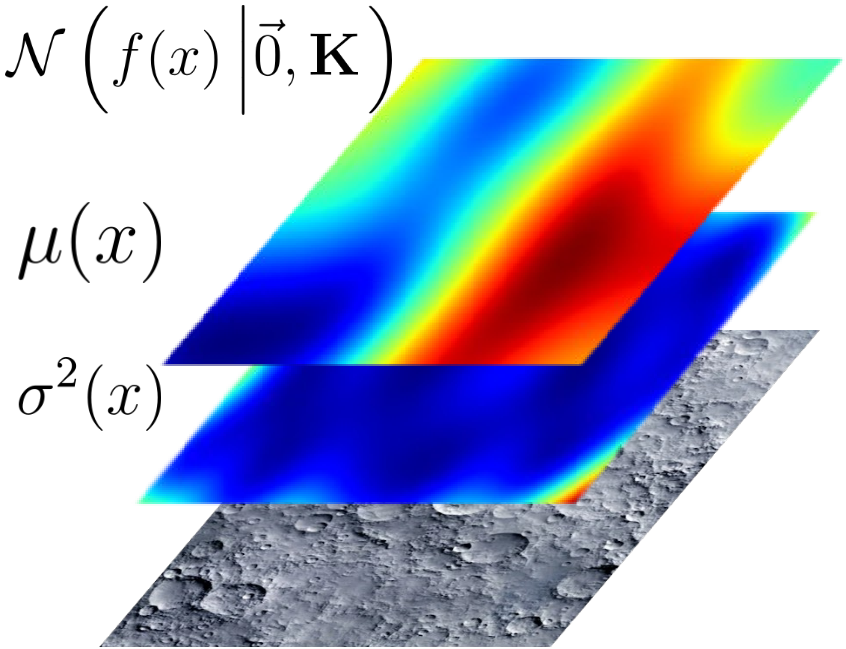
$\mu_t(x)$

$\sigma^2_t(x)$



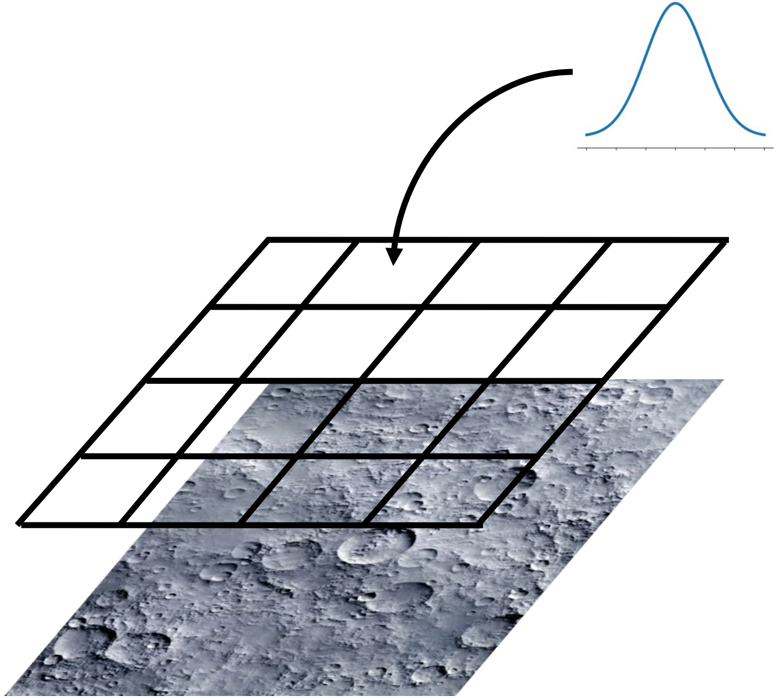

- Mutual Information (MI) is a common objective function used in exploration
- Gaussian Processes (GPs) are a convenient, but computationally intensive tool for computing MI
- We use Fractional Binding and Bayesian linear regression to approximate a GP while improving in memory and time complexity

# Where do $\mu_t(x)$ and $\sigma^2_t(x)$ come from?




Gaussian Process Regression

e.g., Contal *et al.*, 2014; Yang *et al.*, 2013; Thompson and Wettergreen, 2008

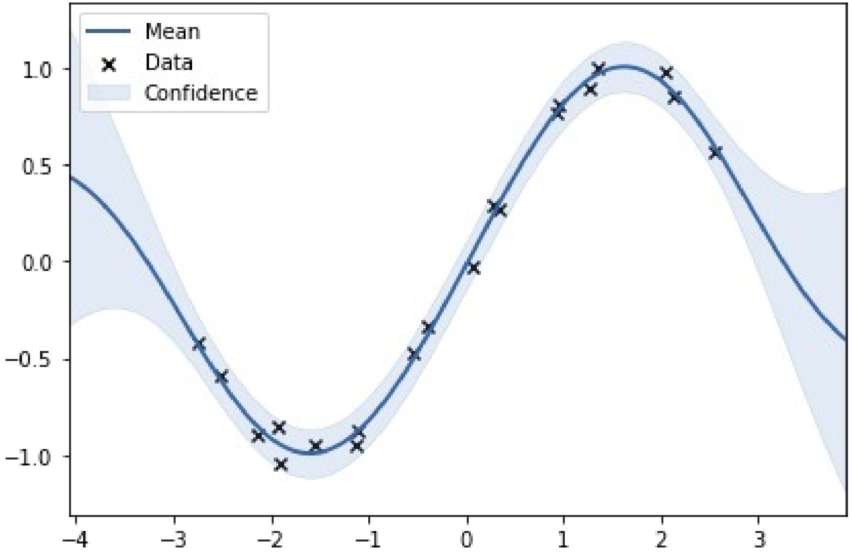


Occupancy Grids

e.g., Zhang *et al.*, 2020; Arora *et al.*, 2019; Charrow *et al.*, 2015; Bourgault *et al.*, 2002

UNIVERSITY OF WATERLOO

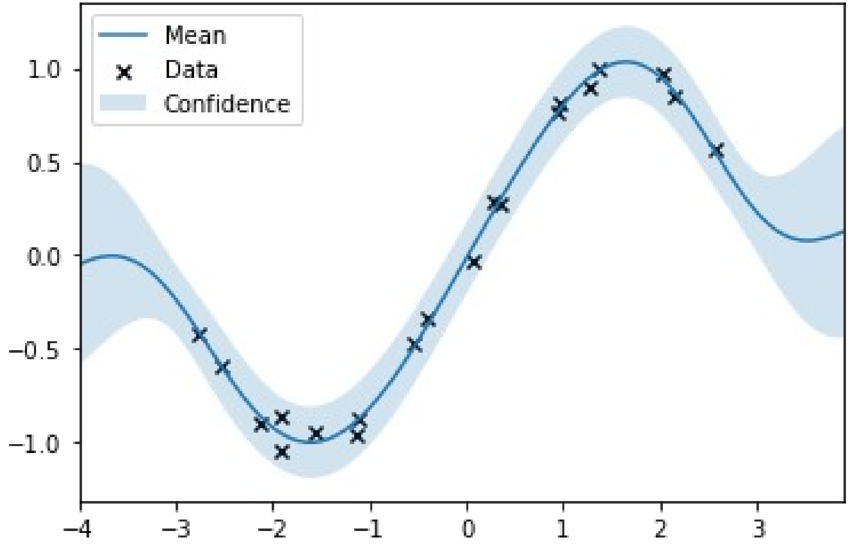
## Gaussian Process Regression




$$\mu_t(x) = \mathbf{k}_t(x)^T \mathbf{C}_t^{-1} \mathbf{y}_t$$

$$\sigma_t^2(x) = k(x, x) - \mathbf{k}^T(x) \mathbf{C}_t^{-1} \mathbf{k}(x)$$

## Bayesian Linear Regression

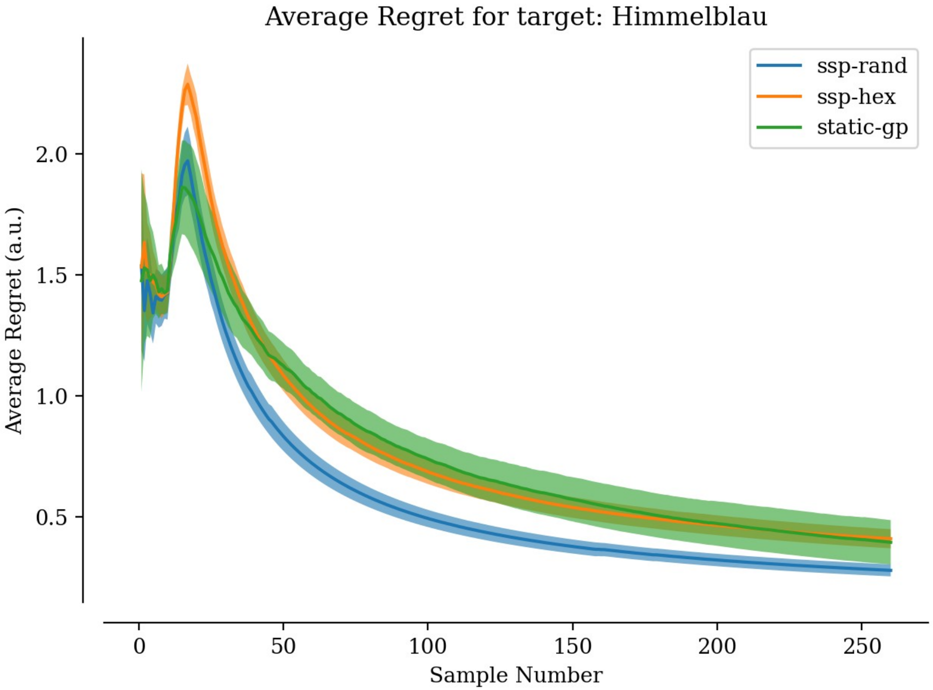


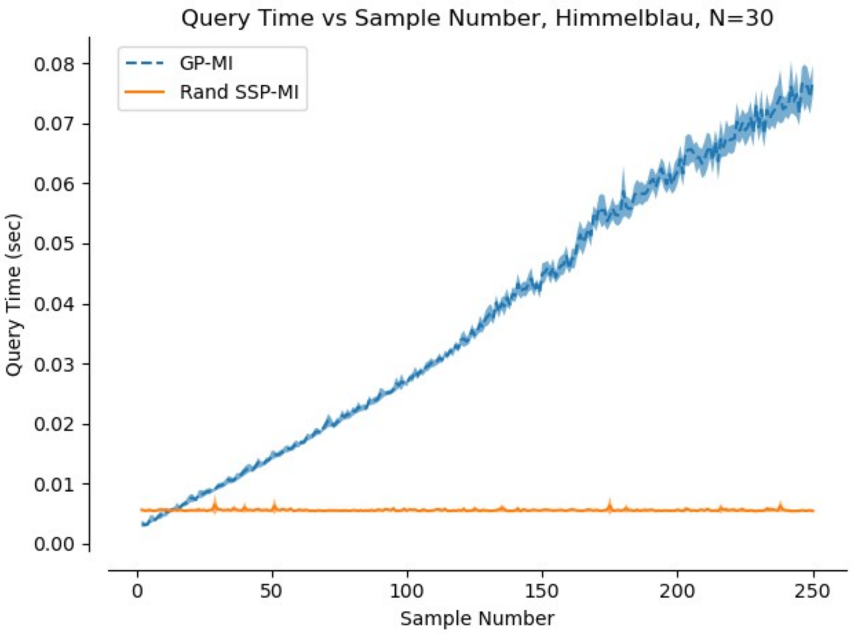

$$\mu_t(x) = \mathbf{m}_t \cdot \phi(x)$$

$$\sigma_t^2(x) = \frac{1}{\beta} + \phi(x)^T \Sigma_t \phi(x)$$

e.g. Perrone *et al.*, 2017, Harrison *et al.*, 2018, Banerjee *et al.*, 2020

# Results - Himmelblau Function




UNIVERSITY OF WATERLOO | FACULTY OF ENGINEERING

# Outline

1. Preliminaries
    a. Kernel Density Estimators
    b. Vector Symbolic Architectures
2. Core Idea of the Paper
3. Practical Applications
4. **Conclusions**

# Summary

1. Vector Symbolic Architectures (VSAs) can be a probabilistic programming language.

2. We can code exploit this relationship to bring sophisticated AI to edge/constrained computing via neuromorphic hardware.

UNIVERSITY OF WATERLOO

# Future Work

Exploring the difference between exact and biological implementations

$$P(X = x) = \mathrm{ReLU}\left(\frac{1}{n}\sum_{\mathbf{x}_i \in \mathcal{D}} \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}) - \xi\right)$$

What effect does biological implementation have?

# Thanks!

Nicole Dumont, Drs. Jeff Orchard, Bryan Tripp, and Terry Stewart

Supported by:

- CFI and OIT infrastructure funding
- Canada Research Chairs program,
- NSERC Discovery grant 261453,
- NUCC NRC File A-0028850
- AFOSR grant FA9550-17-1-0026
- Intel Neuromorphic Research Community Grant

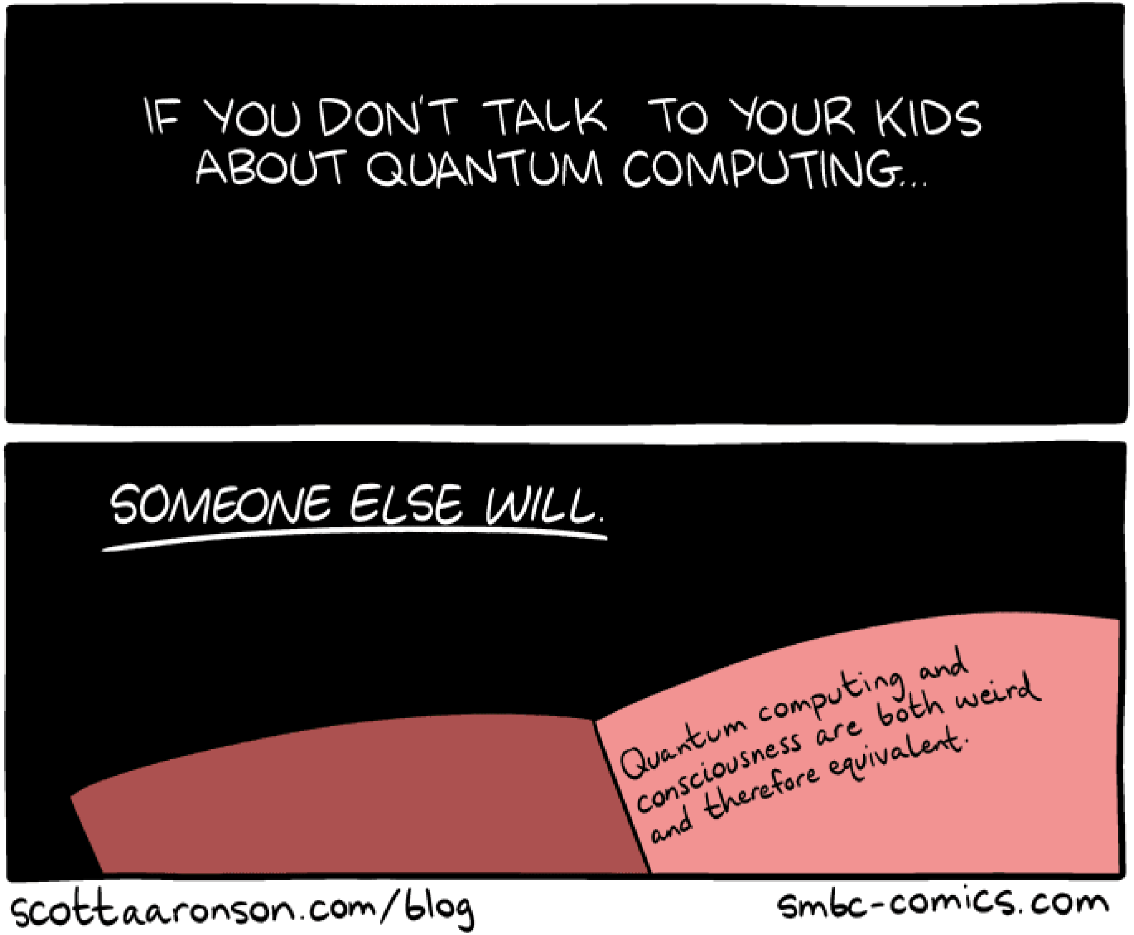
UNIVERSITY OF WATERLOO

# Questions?

michael.furlong@uwaterloo.ca

UNIVERSITY OF WATERLOO

# Backup




UNIVERSITY OF WATERLOO